electronics

MDPI

Article

StarCAN-PFD: An Efficient and Simplified Multi-Scale Feature Detection Network for Small Objects in Complex Scenarios

Zongxuan Chai [1,†], Tingting Zheng [2,*,†] and Feixiang Lu [3]

[1] School of Electrical and Control Engineering, North China University of Technology, Beijing 100144, China; 21101150110@mail.ncut.edu.cn
[2] School of Economics and Management, North China University of Technology, Beijing 100144, China
[3] SUS-Baidu PaddlePaddle Intelligent Sports Technology Innovation Center, Beijing 100085, China; lufeixiang@sus.edu.cn
* Correspondence: zhengtt2019@ncut.edu.cn
† These authors contributed equally to this work.

Abstract: Small object detection in traffic sign applications often faces challenges like complex backgrounds, blurry samples, and multi-scale variations. Existing solutions tend to complicate the algorithms. In this study, we designed an efficient and simple algorithm network called StarCAN-PFD, based on the single-stage YOLOv8 framework, to accurately recognize small objects in complex scenarios. We proposed the StarCAN feature extraction network, which was enhanced with the Context Anchor Attention (CAA). We designed the Pyramid Focus and Diffusion Network (PFDNet) to address multi-scale information loss and developed the Detail-Enhanced Conv Shared Detect (DESDetect) module to improve the recognition of complex samples while keeping the network lightweight. Experiments on the CCTSDB dataset validated the effectiveness of each module. Compared to YOLOv8, our algorithm improved mAP@0.5 by 4%, reduced the model size to less than half, and demonstrated better performance on different traffic sign datasets. It excels at detecting small traffic sign targets in complex scenes, including challenging samples such as blurry, low-light night, occluded, and overexposed conditions, showcasing strong generalization ability.

Keywords: StarCAN-PFD; CAA; PFDNet; DESDetect; small object detection; complex samples

Citation: Chai, Z.; Zheng, T.; Lu, F. StarCAN-PFD: An Efficient and Simplified Multi-Scale Feature Detection Network for Small Objects in Complex Scenarios. *Electronics* **2024**, *13*, 3076. https://doi.org/10.3390/electronics13153076

Academic Editor: Beiwen Li

Received: 12 July 2024
Revised: 28 July 2024
Accepted: 31 July 2024
Published: 3 August 2024

1. Introduction

Detecting small targets with multiple scales and blurriness poses significant challenges for current recognition algorithms. Among these, road traffic sign detection algorithms are an essential research area in modern computer vision technology, aiming to automatically and accurately locate and classify traffic signs on the road. Numerous scholars have conducted in-depth research to enhance the network security [1], real-time capabilities [2], and recognition performance [3,4] of these algorithms. Compared to other object detection applications, distinguishing traffic signs is particularly challenging.

Current mainstream recognition algorithms designed by scholars are based on different colors [5], fixed symbols [6], and fixed categories [7]. However, traffic sign images are usually very small and may be affected by complex backgrounds and occlusions caused by weather and road conditions. Additionally, the rapid movement of vehicles leads to scale changes during image capture, further increasing the difficulty of recognition. Classifiers in traditional machine learning methods Traditional machine learning methods' classifiers need help with these complex issues. The emergence of deep learning methods based on convolutional neural networks (CNNs), particularly the YOLO object detection algorithm, offers solutions to these challenges. Developing an algorithm that can accurately and quickly detect small objects in complex backgrounds, occlusions, blurry images, and at different scales would provide a new research paradigm for detection in various fields.

To address the above challenges with difficult samples, some researchers have adopted explicit data augmentation techniques [8], multi-scale feature integration [9], or pyramid feature hierarchies [10] to extract features rich in scale information. For example, Mengtao et al. [11] designed the YOLO-X network using the concept of multi-branch convolutional reparameterization, which improved the network's ability to extract symbol features. Geng [12] and Zeng [13] combined high-performance networks such as Faster and EfficientViT with attention mechanisms to achieve accuracy and lightweight performance. However, these improved models often sacrifice the information processing capabilities of the feature extraction network and rely on complex manual adjustments, especially in the design for detecting small targets in autonomous driving scenarios. This paper aims to avoid the path of network complexity by proposing a simplified and clear network structure. The goal is to improve both lightweight performance and accuracy while providing a new paradigm for various fields with the proposed improved network.

In this context, this paper proposes the StarCAN-PFD network, with the specific improvements and innovations as follows: (1) Redesign the backbone network based on element-wise multiplication to achieve a balance between model accuracy and lightweight structure with a simple network design. (2) Proposing the Multi-Scale Feature Aggregation (MSFA) module and network, which captures information across three layers of scales. This mechanism overcomes the challenge of detecting small targets with varying scales through a focus-diffusion process, enhancing detection accuracy. (3) Redesign the detection head network based on shared convolution and detail-enhancing convolution to improve the ability to discern details in challenging samples, such as those that are blurry, overly dark, or occluded. This redesign significantly enhances recognition accuracy while achieving a lightweight model. In application testing, it successfully identifies difficult samples that mainstream algorithm networks typically miss, significantly reducing instances of missed detections and false detections.

The rest of this paper focuses on the design and experimental validation of the efficient detection network StarCAN-PFD. In Section 2, the rationale and relevant theories behind the proposed network are explained. Section 3 details the specific design of the network. Section 4 outlines the experimental datasets, parameter settings, and evaluation metrics. Section 5 presents the results of comparative experiments and ablation studies. Finally, Section 6 summarizes the algorithm and discusses future work.

2. Related Work

2.1. Small Object Detection

Deep learning methods for object detection are generally divided into region-based detection and single-stage detection. Region-based object detection algorithms, such as the R-CNN series [14–16], extract candidate regions from an image, classify and identify these candidates, and adjust their coordinates. However, with the increasing demand for lightweight and accurate models, single-stage detection algorithms like YOLO and SSD [17] have become more suitable for small object detection. They directly regress the position and classification probability of the target box, offering superior recognition speed compared to region-based detection.

Autonomous driving traffic recognition differs from conventional detection tasks because it often involves distant small targets, making it a small object detection task. Depending on the environment, existing definitions of small object detection are mainly categorized into relative and absolute scales. The relative scale approach defines small objects based on their relative proportion to the image. Specifically, it considers the relative area of all object instances within the same category, where the median ratio of the bounding box area to the image area ranges from 0.08% to 0.58% [18]. The absolute scale approach defines small objects based on their absolute pixel size. Typically, it refers to objects with a resolution smaller than 32 pixels by 32 pixels [19].

Traditional methods have fewer optimizations designed explicitly for small object characteristics. This, coupled with the increased detection difficulty due to the inherent

properties of small objects, leads to generally poor performance in small object detection. Traffic signs, in particular, are challenging to recognize due to their limited usable features. YOLO and SSD algorithms, along with their improved versions, have been widely applied to small object detection in complex scenarios such as traffic sign detection. Their recognition performance is now comparable to region-based detection methods. For instance, validation analysis by Zhang et al. [20] on the CCTSDB dataset found that YOLOv5 achieved an map@0.5 value 16.65% higher than the highest region-based detection method, Sparse R-CNN, and had an FPS value 115.01 higher than the highest FPS region-based method, Dynamic R-CNN. Among similar single-stage algorithms, YOLOv5's map was 27.1% higher, and FPS was 101.13 higher than SSD. Various studies, such as those by He [21], Wu [22], and Li [23], have shown that YOLO algorithms perform optimally in different datasets like TT100k (Tsinghua-Tencent 100K), SIMD (Satellite Imagery Multi-vehicles Dataset), and GRDDC2022 (Road Damage Detector). Despite the performance improvements in these algorithms, which have maximized the recognition accuracy for small object detection, the unique characteristics of targets in the traffic recognition field still result in many missed and false detections.

Numerous studies have confirmed the superiority of the YOLO framework in object recognition. Based on this, we chose the high-performance YOLOv8 as the overall framework for the design of our research algorithm.

2.2. Efficient Feature Extraction Network and Attention Mechanism

Efficient network structures aim to achieve an ideal balance between computational complexity and performance. In recent years, numerous innovative concepts have been introduced to enhance network efficiency, and these have also been applied to the field of object detection. For instance, lightweight networks based on depthwise separable convolutions, such as MobileNet [24], have been used in applications like pest [25] and disease detection [26], facial expression recognition [27], and road damage detection [28]. However, due to information loss during the ReLU operations across different dimensions, Hao's [29] introduction of Ghost, similar to depthwise separable convolution in YOLOv4, also suffers from weak feature extraction capabilities. These network algorithms may not be well-suited for detecting challenging small targets in complex environments. Other innovative networks include the lightweight EdgeViTs based on Vision Transformers (VIT) [30], FasterNet [12,31], and various heavily handcrafted designs to achieve performance results. Guang [13] incorporated the EfficientViT network concept into the YOLOv5 backbone to enhance feature extraction performance. However, the combination of multiple new modules, such as DSConv, MBConv, and EfficientViT, increased the optimization difficulty of the backbone network. While effective, these approaches often contradict the principle of pursuing simplicity and efficiency in network design and can hinder generalization research across diverse network structures. Xu et al. [32] proposed a simple yet efficient network, StarNet, combining basic convolution modules with star operations (element-wise multiplication). Unlike existing network structures, StarNet avoids complex architectures and meticulously chosen hyperparameters. It demonstrates the ability to implicitly process high-dimensional features while operating in low-dimensional spaces, resulting in excellent performance in practical applications. StarNet achieves a detection accuracy of 0.9% higher than EdgeViT-XS and operates over twice as fast. The specific structure of StarNet is shown in Figure 1.

Using lightweight feature extraction networks for object detection often results in a loss of accuracy. Introducing new network modules and attention mechanisms into the feature extraction network is a common method to improve feature extraction capability. For instance, Guang [13] achieved performance improvements by fine-tuning the positions of CBAM attention mechanisms in conjunction with EfficientViT. Li et al. [33] integrated CBAM (Convolutional Block Attention Module) into the YOLOv3 backbone, allowing the network to autonomously learn the weight of each channel, enhancing critical features while suppressing redundant ones, thereby achieving an 8.50% mAP improvement.

Given the efficiency and lightweight nature of the StarNet network, it not only achieves lightweight and accuracy improvements but also provides ample deployment space for embedding new modules to achieve higher performance enhancements.

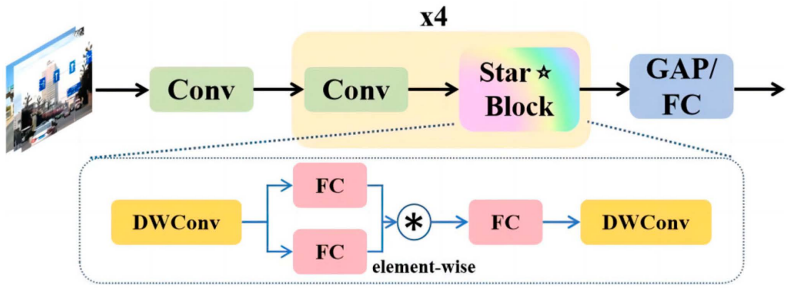


Figure 1. Structure of the StarNet network.

2.3. Recognition of Difficult Small Target Samples

For the challenging task of extracting features from difficult small targets, such as multi-scale, blurry, and occluded objects, various methods have been explored, including data augmentation [34], multi-scale feature integration [35], Feature Pyramid Network (FPN) enhancement [36] and redesign, and multi-scale convolution kernels [37]. These methods still need to adequately solve the problems when deployed in autonomous driving due to network complexity or poor real-time recognition performance.

The widely used YOLO v8 adopts the concept of multi-scale feature fusion, utilizing the structure of FPN (Feature Pyramid Networks) [38] + PAN (PANet) [39] for feature fusion. FPN is a top-down unidirectional structure that enriches the semantic information of features but overlooks localization information. To compensate for this loss, PAN adds a bottom-up path on top of FPN, combining low-level and high-level features. However, this combination can cause small-scale target features to be overshadowed by medium- and large-scale target features, increasing the risk of missing or misclassifying small-scale targets. The FPN+PAN structure achieves a complementarity of semantic and localization information, but it also increases computational complexity and may result in the loss of some original feature information after multiple upsampling and downsampling processes. This can lead to a neglect of fine details in low-level feature maps, reducing accuracy. The specific structures are shown in Figure 2. In practical application testing, these network detection systems still exhibit instances of false detections and missed detections with challenging samples, resulting in suboptimal performance in the field of autonomous driving.

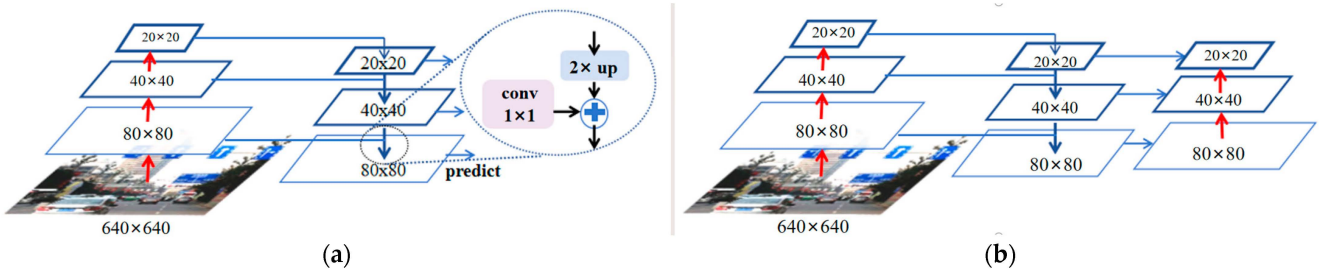


Figure 2. The neck network structures. (**a**) FPN pyramid structure (**b**) FPN+PAN pyramid structure.

This study aims to leverage the existing advantages of pyramid networks for feature enhancement. We intend to design a stronger model that preserves challenging feature details and contextual information while ensuring the model remains lightweight. This approach addresses the challenge of recognizing many difficult small target samples in autonomous driving.

3. The Proposed StarCAN-PFD Network

Detecting small traffic signs in complex autonomous driving scenarios often encounters issues such as missed detections and poor detection performance. Additionally, recognizing multi-scale objects can lead to the loss of contextual information. Although the StarNet network outperforms existing network algorithms in terms of performance and maintains a simple and clear structure, it does not address these issues. This paper redesigns an efficient multi-scale feature detail detection network, StarCAN-PFD, to achieve effective detection of small traffic signs in complex scenarios. The aim is to maintain the original network's efficiency and simplicity while improving the detection of challenging samples in autonomous driving. The overall network structure is shown in Figure 3.

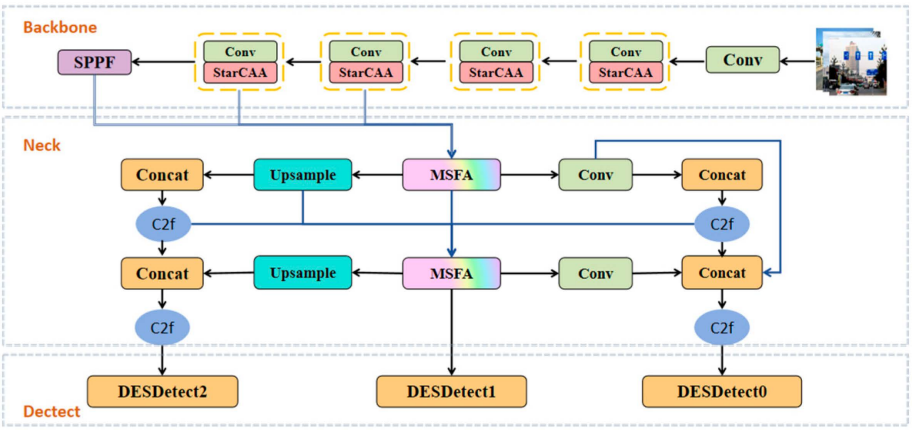


Figure 3. Structure of the StarCAN-PFD network.

In the backbone part of the network, we leverage the efficiency of star element-wise multiplication and embed the Context Anchor Attention (CAA) [40] mechanism to enhance feature extraction capability, resulting in the StarCAN network. Additionally, we propose a feature-focused diffusion pyramid structure called PFDNet for the neck, focusing on multi-scale features. Additionally, in the head, we introduce a lightweight detail-enhanced detection head (DESDetect) based on the collaborative effect of shared convolution and DESConv.

The design of StarCAN is not only easy to extend and adjust but also adaptable to various computational resources and application scenarios. Through its structure and processes, it efficiently extracts and processes image features, providing strong support for image classification tasks. In Section 5.2, we conducted ablation experiments, demonstrating its advantages and potential applications in image classification.

3.1. StarCAN BackBone

The StarCAN network structure is extremely simple, consisting of an initial Stem layer and four feature extraction stages. The image sequentially passes through the Stem layer and the four stages, progressively extracting features and outputting feature maps at each stage, thereby achieving efficient image classification.

Firstly, the 3-channel input image passes through the Stem layer, where an initial feature extraction is performed using a convolutional layer and a ReLU activation function, producing an initial feature map with 32 channels and halved spatial resolution. In each stage, downsampling is performed using a convolutional layer with a stride of 2, halving the spatial resolution of the feature map while doubling the number of channels. Each stage includes a StarCAA Block module. Each Block consists of depthwise separable convolution, a regular convolution layer, batch normalization, a ReLU activation function, the Context Anchor Attention (CAA) mechanism, and a Drop Path for random depth dropping. These Blocks progressively extract and process feature maps, converting low-level features into

high-level features, thus providing rich feature representations for the final classification task. The complete structure of the StarCAN network is shown in Figure 4.

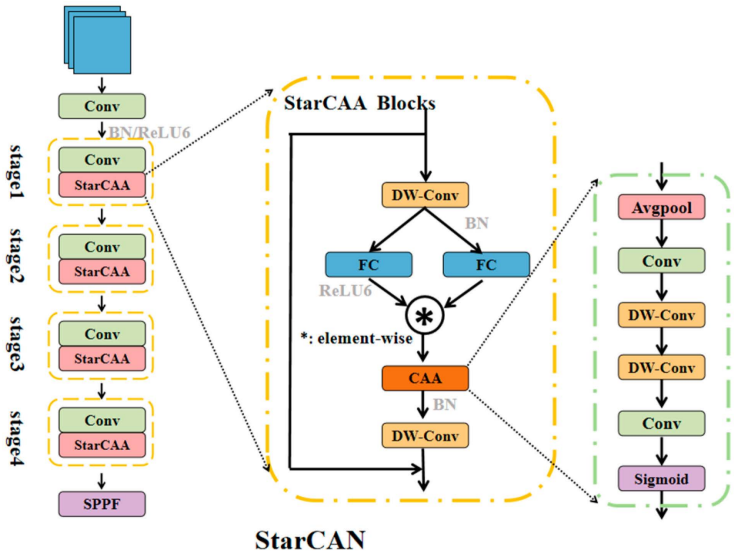


Figure 4. Structure of the StarCAN network and the StarCAA Blocks.

3.1.1. StarCAA Blocks

StarCAA Blocks are the core modules of the StarCAN network, highlighting the roles of element-wise multiplication and the CAA attention mechanism and incorporating multiple convolution operations and attention mechanisms. The process begins with a depthwise separable convolution (DWConv) to efficiently extract initial features while reducing computational complexity. Next, two parallel convolution layers transform the features, and a ReLU activation function applies nonlinear mapping. At this stage, element-wise multiplication merges the outputs of the two convolution layers, producing more representative features. Following this, another layer of depthwise separable convolution and pointwise convolution (g) further refines the feature map. The channel attention mechanism (CAA) is then introduced, adaptively recalibrating feature responses by assigning different weights to each channel, thereby highlighting important features and significantly enhancing the discriminative power of the feature map. Additionally, the introduction of Drop Path improves the network's robustness and generalization ability. Finally, the input feature map is added to the processed feature map to achieve residual connection, further enhancing network performance.

By integrating these components, the Block module aims to gradually extract and enhance relevant features, thereby improving the network's overall feature representation and classification performance. This multi-layer stacking method implicitly transforms input features into exceptionally high nonlinear dimensions while operating in a low-dimensional space.

3.1.2. The Context Anchor Attention (CAA)

The Context Anchor Attention (CAA) mechanism is integrated within the StarCAA Blocks, as illustrated in Figure 4, sharing the input parameters of the backbone network. The CAA mechanism uses global average pooling and one-dimensional depthwise separable convolution to capture relationships between distant pixels and enhance features in the central region. The specific structure of this attention module is depicted in Figure 4. The attention mechanism starts with the input feature map, which is processed through an average pooling layer to generate local region features X_{pool}:

$$X_{pool} = \text{AvgPool2d}(X, 7, 1, 3), \quad (1)$$

The pooling layer is configured with a kernel size of 7, a stride of 1, and a padding of 3. Within the average pooling layer, the input feature maps are aggregated and dimensionality-reduced. Aggregation relies on computing the mean of local areas to summarize features, smoothing the feature maps, and reducing the impact of noise, thus extracting more robust features. This approach mitigates the model's tendency to overfit local noise and details, enhancing its generalization capability, which is particularly beneficial in noisy environments such as autonomous driving. Dimensionality reduction decreases the feature map dimensions, reducing data volume and computational load, thereby improving the efficiency of subsequent convolution operations. The padding and stride settings ensure that the pooled feature map retains the same size as the input, preserving global information. This enables the model to consider global context information when processing local features. After average pooling, the feature maps contain less redundant information, reducing computational complexity and enhancing the model's efficiency. Lowering the computational resource demands of the model is crucial for the subsequent calculation of attention factors.

Next, the pooled feature map X_{pool} undergoes a 1×1 convolution operation, resulting in an intermediate feature map X_1.

$$X_1 = \mathrm{Conv}_1(X_{pool}), \tag{2}$$

Subsequently, the intermediate feature map X_1 sequentially passes through depthwise separable convolutions in the horizontal and vertical directions to capture contextual information from different orientations, generating the feature map X_v.

$$X_h = \mathrm{DWConv}_{1\times k_h}(X_1),\ X_v = \mathrm{DWConv}_{k_v\times 1}(X_h), \tag{3}$$

Typically, depthwise separable convolutions (DWConv) decompose a standard convolution into two simpler operations: depthwise convolution and pointwise convolution. We define the parameter count of a standard convolution as $O\left(K^2 \cdot C_{in} \cdot C_{out} \cdot H \cdot W\right)$. The computational complexity of a depthwise separable convolution is $O\left(K^2 \cdot C_{in} \cdot H \cdot W + C_{in} \cdot C_{out} \cdot H \cdot W\right)$. Here, K represents the kernel size, C_{in} and C_{out} are the numbers of input and output channels, respectively, and H and W are the height and width of the feature map. This design significantly reduces the number of parameters, especially when C_{in} and C_{in} are large. The parameter count after decomposing into depthwise and pointwise convolutions is much lower than that of standard convolutions, thus reducing the model's parameter count and lowering the risk of overfitting. Depthwise separable convolutions can significantly improve the model's inference and training speed while maintaining similar or even higher accuracy.

The two depthwise separable convolutions in the CAA attention mechanism are designed to be lightweight while recognizing long-distance pixel correlations. Unlike traditional $k \times k$ 2D depthwise convolutions, these use a pair of 1D depthwise convolution kernels to achieve a similar effect as standard large-kernel depthwise convolutions, thus reducing both parameters and computational load. The reconfigured horizontal (kh) and vertical (kv) kernel sizes capture relevant information in the horizontal and vertical directions of the input feature map, respectively. This operation extracts edge and shape information in different directions of the feature map more effectively, enhancing the ability to establish relationships between distant pixels without significantly increasing the computational cost due to the dual depthwise separable convolution design.

Next, the feature map X_v undergoes a second 1×1 convolution operation to extract high-level contextual features further, resulting in an enhanced feature map X_2. This enhanced feature map is then passed through a Sigmoid activation function to generate the attention factor A. Finally, the input feature map is multiplied element-wise by the attention factor A, producing the enhanced output feature map Y. By weighting the original input

feature map, important features are enhanced while unimportant features are suppressed, thereby improving the model's ability to focus on crucial parts.

$$X_2 = \mathrm{Conv2}(X_v),\ A = \sigma(X_2),\ Y = A \odot X, \tag{4}$$

Through the aforementioned multi-stage processing and weighting mechanism, the CAA module effectively integrates multi-scale contextual information, significantly enhancing the overall performance of the neural network in various computer vision tasks.

3.2. Multi-Scale Focus and Diffusion Pyramid Network

To address the issues of frequent upsampling and downsampling in YOLO networks, which lead to the loss of original feature information and poor multi-scale feature performance, we were inspired by the PKINet module designed by Xin et al. [40] for recognizing remote sensing images with large-scale variations and complex backgrounds. We proposed the Pyramid Focus and Diffusion Network (PFDNet) for small traffic sign detection. PFDNet employs a focus–diffusion–focus mechanism to propagate features with rich contextual information across different detection scales, helping to capture more contextual information and improve detection performance.

3.2.1. Multi-Scale Feature Aggregator Block

We designed the parallel Multi-Scale Focus Aggregator (MSFA) Block to accept feature inputs from three backbone network layers: P5, P4, and P3. The module initially defines three input channels (inc) to receive feature maps of different scales. The first convolution layer sequence includes an upsampling layer and a 1×1 convolution layer. The second convolution layer uses a scaling factor e to choose between a 1×1 convolution or retaining the original input. We introduced the ADown (downsampling) [41] module for the third downsampling convolution layer, which reduces information loss compared to the conventional 2×2 convolution module. For the input feature maps, X_1, X_2, and X_3, the specific definitions are as follows:

$$X_1' = \mathrm{Conv}_1(\mathrm{Upsample}(X_1)),\ X_2' = \mathrm{Conv}_2(X_2),\ X_3' = \mathrm{Conv}_3(\mathrm{ADown}(X_3)), \tag{5}$$

As shown in Figure 5, the ADown module uses an average pooling strategy for downsampling. The downsampled feature map is split into two parts along the channel dimension. The second part undergoes additional max pooling, followed by convolution, batch normalization, and activation operations. The processed parts are then concatenated along the channel dimension, allowing the output to retain more contextual information.

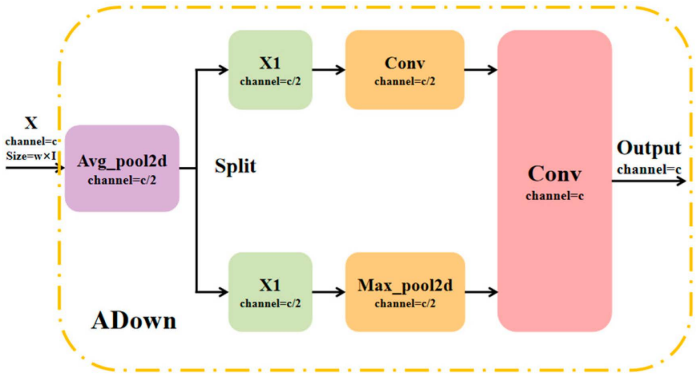


Figure 5. Structure of the ADown model.

To fully utilize these multi-scale features, we apply a set of parallel depthwise convolutions to the concatenated feature map to capture multi-scale contextual information.

The three processed feature maps are then concatenated along the channel dimension. The function is defined as follows:

$$X_{cat} = \text{Concat}(X'_1, X'_2, X'_3), \tag{6}$$

The concatenated feature map is processed using depthwise convolution layers with multiple different kernel sizes and then summed. Finally, the summed feature map is further processed through a pointwise convolution layer and added to the original feature map to generate the final output feature map. Figure 6 illustrates the structure of the PWConv (The pointwise convolution).

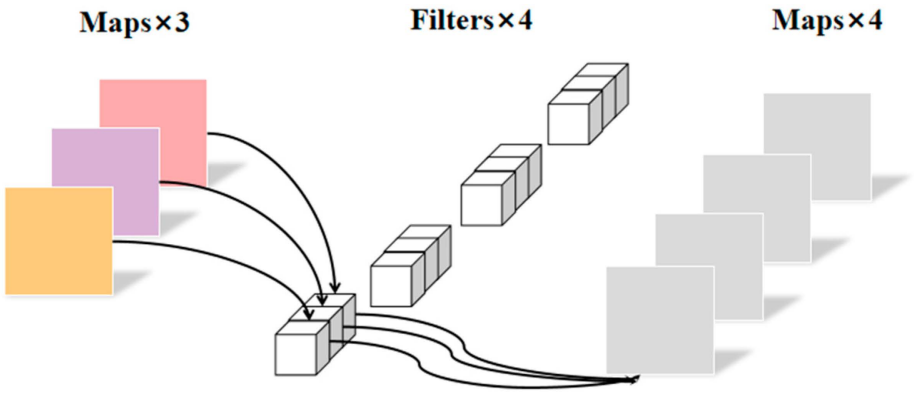


Figure 6. Structure of the PWConv.

The specific structural design of the MSFA Block is illustrated in Figure 7. According to this structure, the MSFA extracts features from the input feature map through multi-scale feature fusion and depthwise convolution (DWConv) operations. By combining multi-scale features with pointwise convolutions, it captures feature information at different scales, thereby enhancing the network's expressive capabilities.

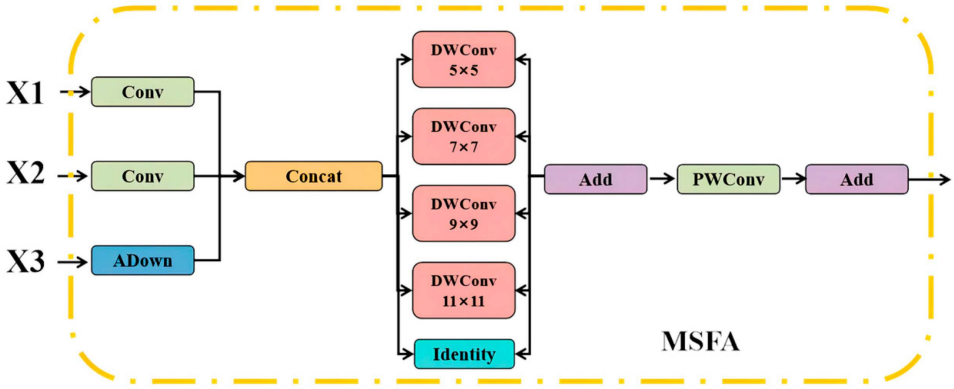


Figure 7. Structure of the MSFA Block.

The definitions of the MSFA Block are as follows:

$$\text{MSFA}(x) = x + \text{PWConv}\left(\sum_{i=1}^{4} \text{DWConv}_i(x)\right) \tag{7}$$

where x represents the input feature map, DWConv_i represents the depthwise separable convolution operation, and PWConv represents the pointwise convolution operation.

3.2.2. PFDNet

We designed the Pyramid Focus and Diffusion Network (PFDNet) based on the MSFA Block. The goal of this pyramid network is to achieve multi-scale feature extraction and fusion.

Firstly, the FocusFeature module processes three feature maps of different scales through upsampling, downsampling, and convolution operations. Parallel depthwise convolutions are used to capture multi-scale information, generating a comprehensive feature map. This architecture ensures that each scale's feature map contains rich contextual information by performing multiple focus and diffusion operations. This enhances the feature representation capability, thereby improving the accuracy of object detection and classification. The specific pyramid network design is shown in Figure 8.

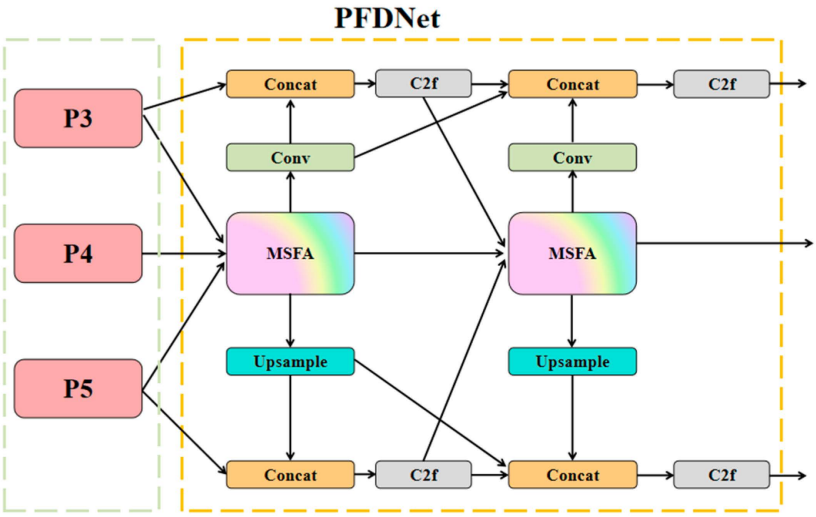


Figure 8. Structure of the PFDNet.

3.3. DESDetect

Target images captured in dynamic driving and natural scenes often exhibit significant visual quality degradation or color distortion [42], compromising performance in advanced visual tasks such as small target detection. Existing solutions [43] usually come with high computational costs and substantial GPU memory usage. Therefore, developing efficient and accurate detection models suitable for resource-constrained environments is crucial for current target detection applications. In traffic sign recognition, traditional detection heads often suffer from information loss during transmission due to the small and variable nature of traffic signs. This lack of information sharing between detection heads can affect the final detection rate. To address these issues, we propose a novel lightweight shared convolution detection head, DESDetect-Head (Detail-Enhanced Conv Shared Detect Head).

The DESDetect-Head is built on the foundation of Group Normalization Convolution (GNConv), which has been shown to reduce computational complexity while maintaining efficient processing and improving detection capability. We further utilize GNConv to enhance the detection head's classification and localization abilities. However, traditional GNConv methods have limitations in capturing fine-grained details essential for precise target localization. To overcome this, we introduce DEConv (Detail-Enhanced Convolution) [44] as a key enhancement component in the detection head.

DEConv integrates multiple parallel convolution layers, including Central Difference Convolution (CDC), Angular Difference Convolution (ADC), Horizontal Difference Convolution (HDC), and Vertical Difference Convolution (VDC). In DEConv, conventional convolution is used to obtain intensity-level information, while differential convolutions enhance gradient-level information. By integrating traditional local descriptors into the convolution layer, the DEConv output can be obtained by simply adding the learned features together.

Deploying five parallel convolutional layers will undoubtedly increase the number of parameters and inference time. Chen noted that when multiple 2D kernels of the same size, stride, and padding are applied to the same input and their outputs are summed to obtain the final output, the corresponding positions of these kernels can be added to

yield an equivalent kernel for the final output. Based on this characteristic, they simplified the parallel convolution deployment to a single standard convolution. The output Fe_{out}, with the same computational cost and inference time, is directed to a regular convolution layer. In the backpropagation phase, the kernel weights of the five parallel convolutions are updated separately using the chain rule of gradient propagation. During the forward propagation phase, the kernel weights of the parallel convolutions are fixed, and the transformed kernel weights are calculated by summing the corresponding positions. This method allows for the acceleration of training and testing processes during the forward propagation phase. For details on the reparameterization steps, refer to Figure 9.

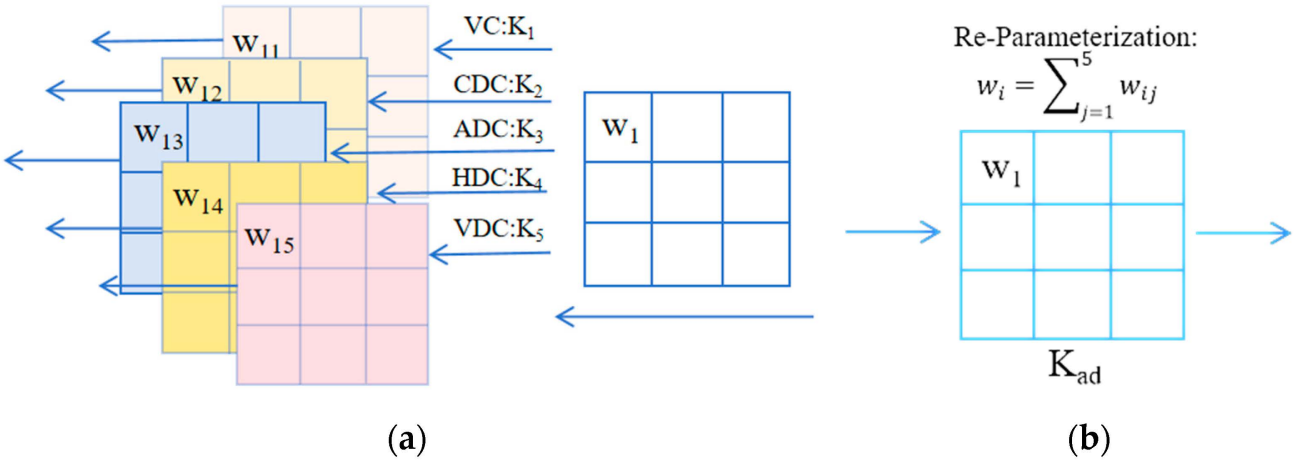


Figure 9. Reparameterization of the concrete implementation process. (**a**) Backpropagation process; (**b**) forward propagation reparameterization.

We can simplify this process into the following formula:

$$Fe_{out} = DEConv(Fe_{in}) = \sum_{i=1}^{5}(Fe_{in} * K_i) = Fe_{in} * \left(\sum_{i=1}^{5} K_i\right) = Fe_{in} * K_{ad} \quad (8)$$

In Formula (8), K_i represents the i-th convolution kernel in DEConv, corresponding to the five specific convolutions mentioned above. Fe_{in} represents the input features, and Fe_{out} represents the output features. The equivalent convolution kernel K_{ad} is the sum of all parallel convolution kernels. $*$ represents a convolution operation. Compared to ordinary convolution layers, DEConv can extract richer features while maintaining the same parameter scale and not introducing additional computational cost and memory burden during the inference phase.

This extends the capabilities of standard convolution by encoding detailed spatial relationships and edge information into the feature map, significantly enhancing the representation capability of the detection head. The specific structural design of this detection head is shown in Figure 10.

The DESDetect-Head achieves synergistic optimization by combining the advantages of shared convolution with the enhanced feature extraction capabilities of DEConv. Shared convolution excels in handling global information and reducing computational complexity, while DEConv plays a crucial role in capturing fine-grained information and enhancing feature representation. This synergy enables the detection head to recognize better and locate complex traffic sign features, ensuring superior generalization across different object categories and environmental conditions. Additionally, the use of DEConv helps mitigate the issue of inconsistent target scales often encountered in shared convolution architectures, ensuring robust detection performance for varying target sizes.

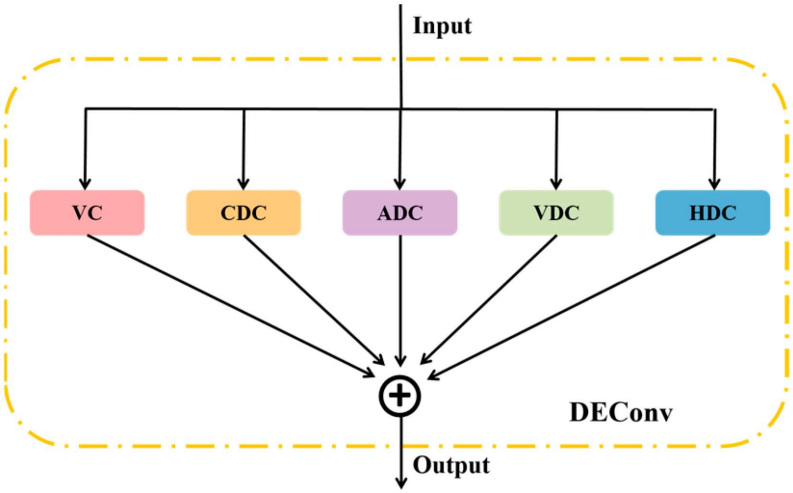


Figure 10. Structure of the DEConv Block here.

By integrating the strengths of shared convolution and DEConv, the DESDetect-Head significantly improves the accuracy and reliability of traffic sign recognition. This method enhances the model's ability to discriminate complex object features and promotes its generalization in diverse environments, offering significant practical value. The structure of the DESDetect-Head is shown in Figure 11. However, adding the DEConv structure on top of the shared convolution will inevitably increase the model's complexity. We will conduct module experiments and discuss the results in Section 5.3.3.

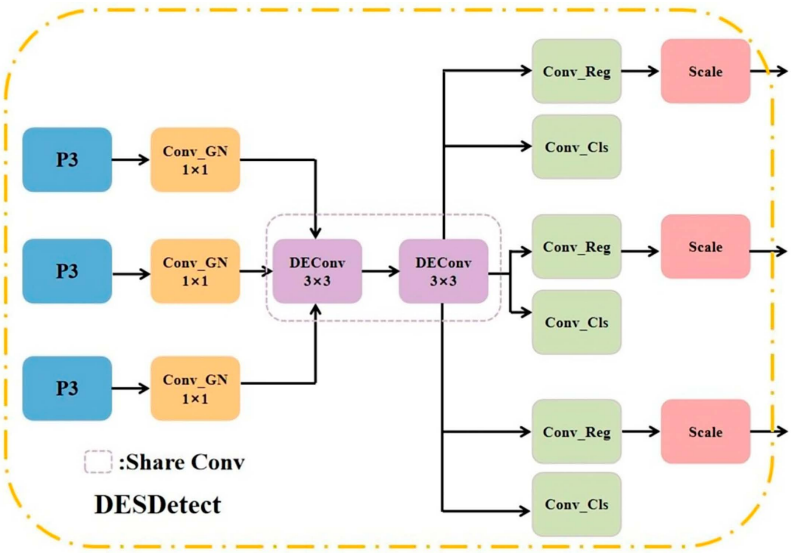


Figure 11. Structure of the DESDetct.

4. Experimental Design

4.1. Experimental Dataset

In this study, we selected the CCTSDB 2021 [20] public dataset as the primary experimental data. The rationale for selecting this dataset is that autonomous driving application scenarios are often complex and variable. Current mainstream traffic-related datasets rarely focus on difficult samples, such as those under adverse weather conditions. Based on this, Zhang enhanced the original CCTSDB 2017 dataset by adding data from special weather conditions and replacing many simple samples with difficult ones to simulate complex road environments better, closely aligning with real-world applications. They specifically collected and categorized samples under different weather conditions, including sunny, cloudy, night, rain, foggy, and snow. Among these samples, there are many specific exam-

ples with blurring and occlusion, making the dataset more complex and experimentally valuable compared to others.

The dataset includes three categories of traffic signs: indicative, prohibitory, and warning signs. It provides a comprehensive public data source for traffic sign recognition research. The images exhibit highly variable brightness and weather conditions, including night, snowy days, rainy days, evenings, and foggy conditions. The dataset contains 17,856 images of traffic signs, with nearly 40,000 traffic signs in total. These are classified into three main categories: warning signs, indicative signs, and prohibitory signs. The dataset serves as a paradigm data source for small target detection in multi-scale, complex environments with various backgrounds, occlusions, and blurriness. The resolution ranges from 1000 × 350 to 1024 × 768. We divided the dataset into training, validation, and testing sets with a ratio of 7:2:1.

To further validate the network's performance, we also selected the TT100k, RoadSign, and GTSDB datasets as auxiliary validation sets. These four datasets are all public and contain real-world images of traffic signs. The annotated traffic sign targets meet the conditions for small target detection, with TT100k, RoadSign, and GTSDB focusing on small sample target detection, facilitating the evaluation of the model under different conditions. The input size for the model was uniformly set to 640 × 640 across all datasets.

4.2. Experimental Environment and Parameters

The hardware environment for this experiment is based on an RTX 4090 GPU and a 16 vCPU AMD EPYC 9654 96-Core Processor CPU for training and testing. We used PyTorch 1.8.1 as the deep learning framework, with CUDA version 11.3 and Python version 3.8.

During the training phase, to better compare model performance and avoid the incomparability caused by differences in pre-trained weights, we used the Stochastic Gradient Descent (SGD) optimizer. For data feature extraction, we performed online augmentation of the data images based on parameters to expand the research samples and increase data diversity, thus improving the model's generalization ability. The experimental parameters and data augmentation settings are shown in Table 1.

Table 1. Experimental and data augmentation parameters.

Experimental Parameters	Value	Data Augmentation Parameters	Value
Initial Learning Rate	0.01	hsv_h (Hue Adjustment)	0.015
Minimum Learning Rate	0.0001	hsv_s (Saturation Adjustment)	0.7
Epoch	300	hsv_v (Brightness Adjustment)	0.4
Batch size	16	Translate (Translation Range)	0.1
Momentum	0.937	Scale (Scaling Range)	0.5
Weight Decay	0.0005	Mosaic (Mosaic Probability)	1
Works	8	Erasing (Random Erasing Probability)	0.4

4.3. Eval

In this study, ensuring detection accuracy, speed, and a light weight in the model is crucial for small target detection applications. We use precision (P), recall (R), and mean Average Precision (mAP) to measure the detection accuracy of the algorithm. Based on the binary classification metrics, the confusion matrix is shown in Table 1. We use frames per second (FPS) to measure the detection speed of the algorithm. Evaluation metrics include the number of parameters (parameters) and giga floating-point operations per second (GFLOPS), which measure the execution time of the model. The number of parameters also assesses the model's size and complexity. Precision (P) measures the accuracy of the model, with the intersection over union (IoU) set to 0.7. This means that the model's prediction is considered correct if the IoU between the ground truth and the predicted box

is greater than 0.7. The specific setting of the IOU value directly impacts the detection rate and accuracy. In typical image recognition experiments, IOU is seldom examined in detail. However, since autonomous driving systems must accurately recognize and detect traffic signs in various complex environments to ensure safety and reliability, both false negatives and false positives can lead to irreparable accidents. Therefore, this study aims to improve recognition accuracy by increasing the IOU value for typical tasks to 0.7. This higher threshold might increase false negatives, but the proposed algorithm strives to overcome this challenge and achieve simultaneous improvements in detection rate and accuracy. Detailed experimental and prediction results demonstrating these improvements are provided in Sections 5 and 6.

The calculation formula is given in Equations (9)–(12).

$$\mathrm{IOU} = \frac{\mathrm{area}(\mathrm{B_P} \cap \mathrm{B_T})}{\mathrm{area}(\mathrm{B_P} \cup \mathrm{B_T})} \tag{9}$$

where $\mathrm{B_P}$ is the predicted box, and $\mathrm{B_T}$ is the ground truth box.

Precision (P), recall (R), and mean Average Precision (mAP) are calculated using the formulas shown in Equations (9)–(11).

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{10}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{11}$$

$$\mathrm{mAP} = \frac{\sum \mathrm{AP}}{\mathrm{N_{class}}} \tag{12}$$

where TP (True Positive) is the number of correct positive predictions, TN (True Negative) is the number of correct negative predictions, FP (false positive) is the number of incorrect positive predictions, and FN (false negative) is the number of incorrect negative predictions. The AP value is the area under the P-R curve, AP represents the sum of AP values for all categories, and $\mathrm{N_{class}}$ represents the total number of categories.

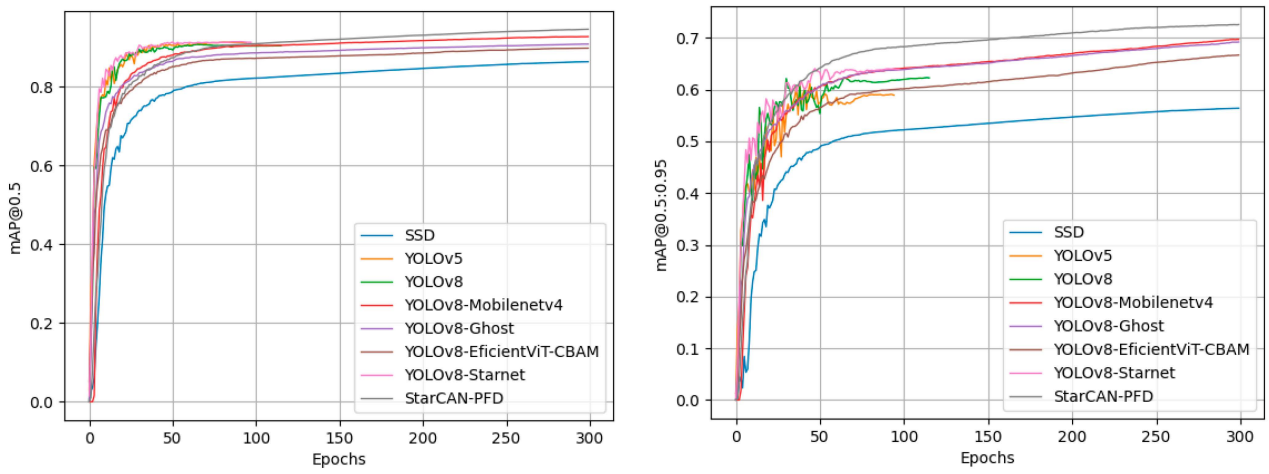
5. Experimental Results and Analysis

5.1. Algorithm Performance Comparison

To evaluate the detection performance of our algorithm, we compared StarCAN-PFD with SSD, YOLOv5, YOLOv8, YOLOv8-MobilenetV4, YOLOv8-Ghost, and YOLOv8-EfficientViT. To highlight the advantages of StarCAN's simplified network, we validated these methods with different backbone networks on the CCTSDB dataset. Additionally, we generated P-R curves to display the effectiveness of the algorithms on this dataset graphically. The specific results are shown in Figure 12.

Based on the comparative experimental results in Table 2, we found that the StarCAN-PFD algorithm has the highest mAP@0.5, precision, and recall compared to other object detection networks. Specifically, StarCAN-PFD improves mAP@0.5 by 9.9%, 4.0%, and 4.0% over SSD, YOLOv5, and YOLOv8, respectively. Compared to other innovative network backbones, StarCAN-PFD (ours) shows an increase in evaluation precision by 1.8%, 2.8%, and 3.8%.

In terms of model size, the StarCAN-PFD model is less than half the size of other algorithms, demonstrating its lightweight nature. The FPS of StarCAN-PFD is comparable to YOLOv8, indicating that it improves accuracy while maintaining sufficient inference speed to meet application requirements.



(**a**) (**b**)

Figure 12. Model P-R curve analysis. (**a**) Comparison of multiple models' mAP@0.5 results; (**b**) comparison of multiple models' mAP@0.5:0.95 results.

Table 2. Comparative experimental results.

Algorithm	Backbone	mAP@0.5	mAP@0.5:0.95	P	R	FPS	Size
SSD	VGG16	0.846	0.564	0.863	0.821	92.3	13.2
YOLOv5	CPSDarknet53	0.905	0.622	0.93	0.827	246.3	5.0
YOLOv8	CPSDarknet	0.905	0.611	0.921	0.825	279.4	5.9
YOLOv8-mobilenetv4	MobileNetv4	0.927	0.697	0.929	0.879	279.6	11.2
YOLOv8-Ghost	GhostNet	0.907	0.692	0.911	0.876	274.0	3.6
YOLOv8-EficientViT-CBAM	EficientViT	0.897	0.668	0.908	0.854	67.7	8.4
YOLOv8-StarNet	Starnet	0.912	0.639	0.930	0.836	334.6	6.0
StarCAN-PFD(Ours)	StarCAN	0.945	0.723	0.946	0.879	278.6	2.9

5.2. Ablation Study

To validate the performance of our algorithm, we conducted ablation experiments on various modules, including StarCAN, PFDNet, and DESDetect head. In the table below, a "√" indicates that the method was used in the model, and a "—" indicates that the method was not used. We used the YOLOv8 network as the baseline model for comparison. As shown in Table 3, each improvement contributes to an increase in detection accuracy, demonstrating the scientific validity and effectiveness of the proposed methods.

Table 3. Ablation experiment results.

Model	StarCAN	PFDN	DESDect	mAP@0.5	P	R	FPS	Params/10^6	GLOPs
0	—	—	—	0.905	0.921	0.825	279.4	3.01	8.1
1	√	—	—	0.924	0.942	0.851	322.6	2.21	6.5
2	—	√	—	0.912	0.941	0.827	274.9	3.04	9.4
3	—	—	√	0.937	0.944	0.865	312.0	2.36	6.5
4	√	√	—	0.927	0.946	0.848	264.9	1.73	6.2
5	√	—	√	0.935	0.938	0.871	270.9	1.57	4.9
6	—	√	√	0.942	0.946	0.879	252.4	2.36	7.4
7	√	√	√	0.945	0.962	0.882	278.6	1.20	4.7

In the ablation experiments, Group 0 uses the YOLOv8 network. Group 1 utilized the efficient star-shaped feature extraction network, StarCAN, which significantly improved accuracy and inference speed. According to the results, mAP@0.5 increased by 1.9%, precision (P) improved by 1%, recall increased by 2.4%, parameters decreased by 26.6%,

and GFLOPS reduced by 19.8%. Compared to the baseline model, its size is reduced, making the lightweight effect more apparent, and FPS increased by 15.5%. This validates the performance and value of the efficient and simple star-shaped network.

In Group 2, we independently validated the parallel pyramid network. Adding the PFDN pyramid network resulted in a slight improvement in mAP@0.5 by 0.7%, precision by 2%, and recall by only 0.2%. Compared to the high computational cost of the FPN+PAN pyramid network, the PFDN pyramid network, with the MSFA module, introduced additional upsampling, downsampling, and multiple parallel depthwise convolution operations, significantly increasing parameters and computation. Parameters increased by 0.9% and GFLOPS by 16.0%, with minimal change in FPS. Increasing the model complexity contradicts our initial goal. We aim to achieve better results through the combination with other modules.

In Group 3, we evaluated DESDetect and found it contributed the most significant performance improvement. Due to the lightweight effect of shared convolution, both parameters and GFLOPS decreased, while DEAConv enhanced the feature detail processing capability. With the synergistic effect of DEAConv and shared convolution, mAP@0.5 increased by more than 3.2%. FPS increased by 11.8%, achieving improvements in model accuracy, speed, and lightweight characteristics. We will further analyze its contributions to accuracy and lightweight performance in Section 5.3.3, focusing on the role of the detection head's module structure.

In Group 4, we added the PFDN pyramid network to the Group 1 model, resulting in a 2.2% increase in mAP@0.5 compared to the baseline model. Both parameters and GFLOPS decreased significantly, with the model size reduced by 40.7%. The modular design allowed for sharing depthwise separable convolution parameters, enabling multi-feature fusion to share information across different scales. This reduced the computational load of each convolution operation, further lowering overall parameters and computational costs. In Group 5, using DESDetect on the StarCAN backbone network, mAP@0.5 increased by nearly 3% compared to the baseline model, with a 47.8% reduction in parameters, a 39.5% reduction in GFLOPS, and a model size compression of nearly 40%. In Group 6, combining PFDN and DESDetect, we observed a strong synergistic effect, with mAP@0.5 increasing by 3.7% compared to the baseline model.

Group 7 represents our proposed StarCAN-PFD network, achieving an mAP@0.5 of 94.5%, which is a four percentage point improvement over the baseline model. Parameters and computational load decreased by 61.1% and 41.9%, respectively, with the model size reduced by half. This demonstrates the effectiveness of the overall network and individual improvements in detecting small traffic targets in complex scenarios.

5.3. Module Comparison Experiment

5.3.1. Comparison of Backbone Network Structure

To better illustrate the recognition accuracy and computational load of the StarCAA feature extraction network, this study compares it with the main feature extraction networks.

The specific results of the parameter count and computational load comparison are shown in Figure 13. Based on the results, we found that our StarCAA backbone feature network has nearly half the FLOPs and parameters of YOLOv5, YOLOv8, EfficientViT, and MobileNetV4. Compared to the Ghost network, the total FLOPs increased by approximately 0.428G, and the computational load increased by 28.412k. However, in the experiments shown in Tables 2 and 3, the mAP@0.5 improved by 1.7%, significantly enhancing detection accuracy. After redesigning the StarNet network, we found that adding a minimal amount of computational load resulted in improved accuracy. Comparing the results in Tables 2 and 3, we observed a 1.2% improvement in mAP@0.5, with the FPS remaining almost unchanged. Regarding the accuracy comparison of the backbone networks, we found that among the latest improved algorithms for these backbone networks, StarCAN achieved the highest precision with an mAP@0.5 value of 0.924. Therefore, our StarCAN backbone network can effectively balance accuracy and computational efficiency.

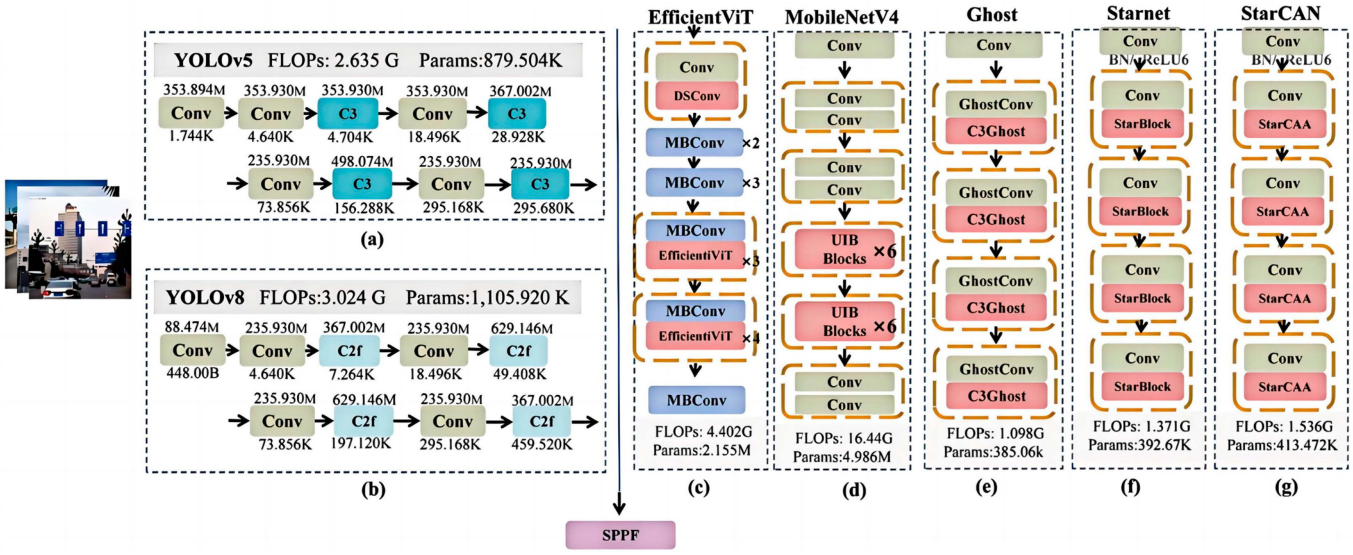


Figure 13. Main backbone network FLOPs and parameters. (**a**,**b**) show the FLOPs and parameters of the backbone extraction networks for YOLOv5 and YOLOv8, respectively. (**c**–**e**) illustrate the FLOPs and parameters of the leading innovative networks. (**f**) The original Starnet feature extraction network. (**g**) presents the computational complexity of the StarCAN network proposed in this study, based on Starnet.

5.3.2. Attention Mechanism Experiment

In this study, we added an attention mechanism to the core Star Block of the efficient star-shaped StarNet network. Unlike traditional studies that add attention layers to the Backbone or head, we chose to modify a single module within the efficient network rather than increasing the overall complexity. We conducted comparative experiments on the CCTSDB dataset to evaluate the effectiveness of adding attention mechanisms in different positions.

Based on the results in Table 4, we found that in this experiment, adding the CAA attention mechanism in the backbone and head was less effective than the proposed StarCAA Block scheme. This proves the effectiveness of our method. Adopting simple structural adjustments in a more streamlined network may be more advantageous than extensive complex manual tuning. Further in-depth research and experiments are required to provide reliable and generalizable conclusions.

Table 4. Comparison of attention mechanism positions.

Attention Position	mAP@0.5	Params/10^6	GLOPs
Starblock	0.924	2.21	6.5
Backbone	0.920	2.30	6.7
Head	0.916	2.69	8.1

5.3.3. DESDetect Head Experiment

The DESDetect head proposed in this study is designed to enhance the discrimination ability of complex object features by incorporating DEConv convolution on top of LSCD's shared convolution. We conducted comparative experiments by embedding DEAConv and shared convolution into the original Detect head separately.

Based on the results in Table 5, we found that adding only the Shared Convolution module increased mAP@0.5 by 0.9%, reduced parameters by 26.6%, and reduced GFLOPS by 24.6% compared to the baseline model. Using only the DEConv module increased mAP@0.5 by 2.6%, but both parameters and GFLOPS slightly increased. Combining shared convolution and DEConv achieved the best performance, with mAP@0.5 reaching 0.937, indicating the superior performance of the DESDetect head in detecting small traffic targets.

Table 5. DEAConv comparison experiment.

Module	mAP@0.5	Params/10^6	GLOPs
Shared Conv	0.914	2.21	6.5
DEConv	0.931	3.10	8.4
Shared Conv + DEConv	0.937	2.69	6.5

Comparing the contributions of individual modules, Shared Conv significantly improved the model's lightweight nature but provided only a slight increase in accuracy. In contrast, DEConv enhanced long-distance information and central features, achieving the highest accuracy improvement among all experiments, thereby validating the detailed explanation of its role provided earlier in this paper. This edge information extraction and feature enhancement capability showed a stronger advantage in the autonomous driving field, which often involves many difficult samples. The combined use of DEConv and shared convolution significantly mitigated the increase in model complexity that typically accompanies accuracy improvements. This combination enhanced feature extraction capability by dynamically adjusting anchors and strides, allowing the detection head to adapt to different input shapes and using DFL layers to decode bounding boxes, further improving localization accuracy and enhancing the model's ability to detect complex object features.

5.4. Performance on Other Datasets

To test the generalization capability of StarCAN-PFD on different traffic small object datasets, we selected three datasets, TT100k, GTSDB, and Roadsign, for comparative validation.

Based on the results in Table 6, we found that compared to the baseline model, StarCAN-PFD significantly improved accuracy, especially on the TT100k and GTSDB datasets, which lack more challenging difficult samples. In the context of autonomous driving road sign detection, our algorithm demonstrated excellent generalization ability in detecting various target objects in different traffic environments. These results validate its effectiveness in detecting traffic targets in complex scenarios, providing the potential for further generalization studies in other complex environments.

Table 6. Results on Different Datasets.

Dataset	YOLOv8			StarCAN-PFD		
	mAP@0.5	P	R	mAP@0.5	P	R
TT100k	0.875	0.894	0.795	0.912	0.907	0.834
GTSDB	0.744	0.905	0.674	0.801	0.946	0.700
Roadsign	0.866	0.920	0.809	0.908	0.919	0.878

6. Discussion

6.1. Further Analysis of Difficult Sample Recognition Performance

The proposed StarCAN-PFD will demonstrate generalized detection performance when dealing with multi-scale and adverse weather conditions in practical applications. We validated this advantage on the CCTSDB dataset. In the IoU threshold range of 0.5 to 0.95, to achieve more precise detection results, we set the intersection over union (IoU) threshold to 0.7. Our experimental results indicate that increasing the IoU threshold in this manner does not reduce the detection rate. On the contrary, due to our algorithm's enhanced ability to recognize difficult samples, the detection rate actually improved. The specific detection accuracy was 94.6%, with a recall rate of 87.9% and an average precision (mAP@0.5) of approximately 0.942. This performance is significantly superior to mainstream object detection algorithms.

In Figure 14, we conducted prediction experiments on foggy, night, snowy, blurry, and overexposed samples. According to the experimental results, we found that StarCAN-PFD demonstrated relatively stable prediction results under these five complex conditions, offering superior overall detection performance. Although, in specific scenarios, YOLOv8

achieved the highest accuracy for blurred samples and the GhostNet achieved the highest accuracy for foggy conditions, both had significant issues with missed detections, leading to poorer overall performance. Under various conditions, StarCAN-PFD consistently maintains high accuracy and successfully identifies more challenging samples. This stability in achieving high accuracy is particularly evident in situations prone to missed detections, such as at night and in snowy conditions. Therefore, our algorithm is highly suitable for challenging and difficult environments.

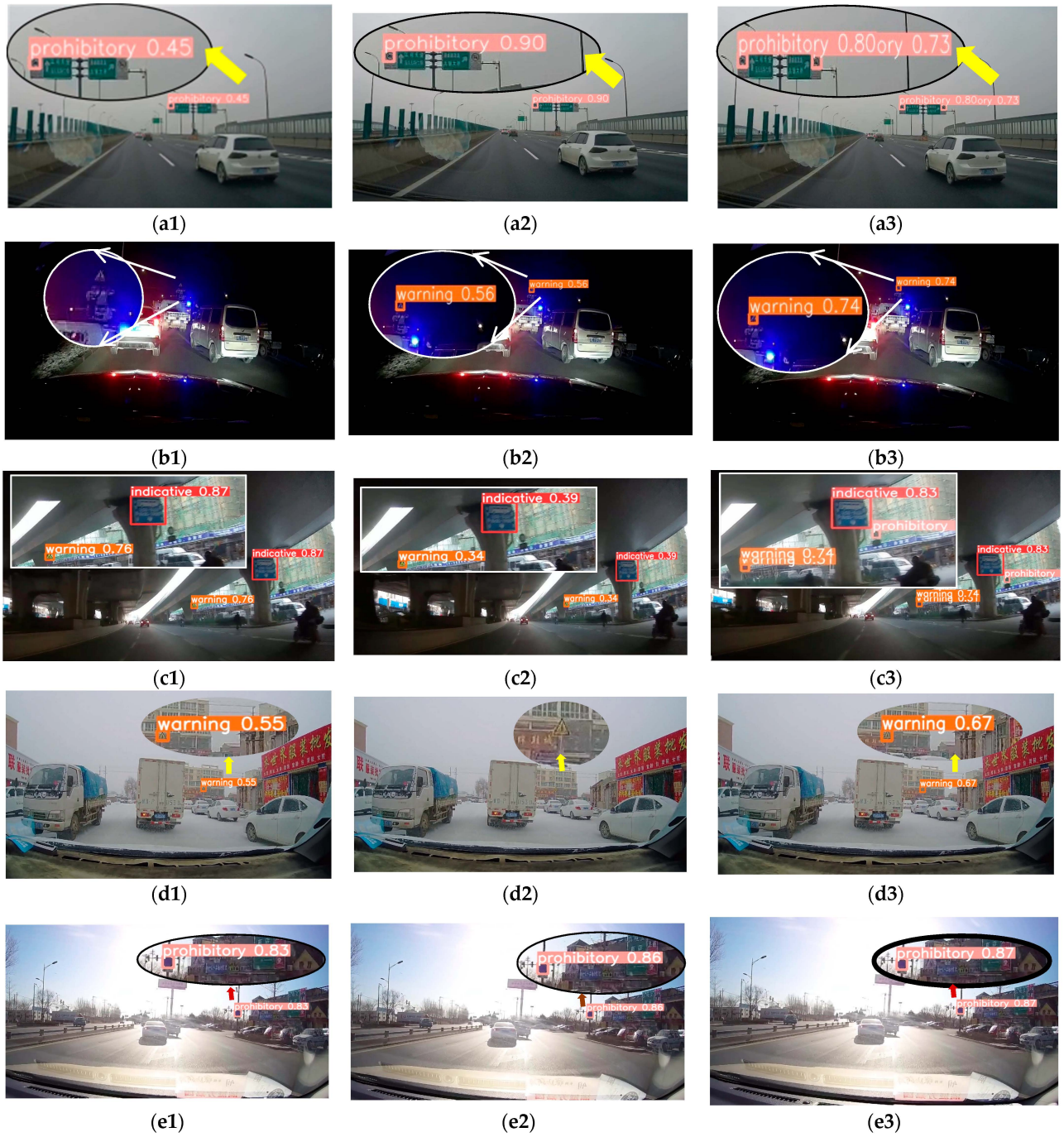


Figure 14. Structure of the DESDetect. (**a**–**e**) show the results under different conditions: foggy, night, blurry, snowy, and sunny days. Groups 1, 2, and 3 represent the visual prediction results of YOLOv8, YOLOv8-Ghost, and our StarCAN-PFD algorithm, respectively.

Figure Group A (Foggy Samples): We observed that both YOLOv8 and YOLOv8-Ghost missed detections, only identifying the left traffic sign. In contrast, the StarCAN-PFD model successfully detected all relevant signs. For accuracy in recognition, YOLOv8-Ghost achieved the highest accuracy in identifying the traffic sign on the left side of the figure. The Ghost backbone network might have higher accuracy in foggy conditions with an increased IoU threshold. However, due to its higher miss detection rate, the overall detection performance could be better. Additionally, the accuracy of 0.80 achieved by our model is sufficient to meet application requirements.

Figure Group B (Night Samples): In the night samples, YOLOv8 failed to recognize the distant, low-brightness, blurry warning sign. Our method, compared to YOLOv8-Ghost, identified it with an 18% higher confidence score.

Figure Group C (Blurry Samples): In the blurry samples under a highway overpass, both YOLOv8 and YOLOv8-Ghost missed the rightmost prohibition sign and the second warning sign on the left. In contrast, our model successfully detected these signs, demonstrating their practical value despite not achieving the highest detection accuracy. While its specific accuracy is slightly lower than that of YOLOv8, its stable performance meets application requirements. YOLOv8-Ghost, on the other hand, showed poor performance with low accuracy in detecting blurry samples.

Figure Group D (Snow Samples): In the snow samples, YOLOv8-Ghost had missed detections, while our algorithm achieved a 12% higher confidence score compared to YOLOv8.

Figure Group E (High-Exposure Sunny Samples): In the high-exposure sunny samples, all three algorithms correctly detected the prohibitory sign, with our algorithm showing the highest detection confidence.

These experiments demonstrate that the proposed StarCAN-PFD presents an efficient algorithmic network without overly complex structures or extensive manual tuning. The performance validation in various complex scenarios also provides more room for other researchers to generalize and improve network algorithms.

6.2. Limitations and Challenges of the Study

Although ablation and comparative experiments have demonstrated the superiority of our algorithm, autonomous driving systems still face challenges in real-world applications due to varying lighting conditions, weather changes, traffic sign styles, and hardware deployments. These factors can all affect model performance. Like other studies, the design of certain structures in this experiment involves trade-offs between accuracy and efficiency. These potential issues require detailed discussion and further research.

In the specific design of the algorithm network, we developed the StarCAN feature extraction network. By simply adding the CAA structure, we sacrificed a small amount of space to achieve a significant improvement in accuracy. This balance between efficiency and accuracy is a common challenge faced by researchers. Although the added parameters and computational load are relatively low, future research should explore more simplified adjustments.

The same issues are evident in the designs of PFDNet and DESDetect. For challenging samples in complex environments and multi-scale variations, we must maintain the model complexity to overcome these challenges. Although we simplified the model through parameter sharing in the backbone network, shared convolutions, and reparameterization, achieving significant reductions in parameters and computational load compared to mainstream algorithms, these experiments are based on fixed research environments. Future work must test hardware deployments to verify their generalization superiority across different hardware devices. Additionally, during the algorithm design process, the adjustments in these schemes need to fully align with the principle of StarCAN, which aims to achieve both accuracy and efficiency improvements with the simplest structure. This area requires collaborative exploration by more researchers to identify the most efficient network structure paths.

The design of DESDetect has potential limitations. While the combination of multi-level convolution and shared convolution enhances feature extraction capabilities and lightweight characteristics, the added convolutional layers and parameters may slow down inference speed during hardware deployment and pose a risk of overfitting. Dynamic adjustments of anchors and strides increase detection flexibility and accuracy but also carry the risk of slowing down inference speed. Despite showing stable and favorable results on the validation and test sets in our experiments, DESDetect requires further extensive data research to improve system reliability due to the complexity of autonomous driving environments. In summary, although DESDetect demonstrates good performance, its practical application requires careful consideration of potential performance fluctuations caused by environmental changes.

The dataset used in this study is one of the latest and most comprehensive datasets for autonomous driving under complex weather conditions. Additionally, we performed data augmentation to increase the data volume further. However, we must consider that the complexity of environmental scenarios requires more extensive real-time experiments. In some challenging conditions, such as extreme weather, unusual lighting conditions, or highly cluttered scenes, there may still be a need for greater diversity. Moreover, the dataset requires more detailed classification. We plan to collaborate with more researchers in the future to expand the traffic sign samples in these complex scenarios. This will ensure a more thorough evaluation of our algorithm and verify its applicability to a wide range of real-world conditions.

Addressing these limitations and challenges is crucial for further improving the applicability and reliability of the algorithm in autonomous driving and other complex recognition tasks. Future research will focus on the issues mentioned above, combining breakthrough discoveries from other scholars to conduct deeper investigations. This will help in developing more robust and efficient models.

7. Conclusions

To address issues such as occlusion, blur, and low-light conditions in complex road scenes, as well as the challenges of small-scale target missed detections and multi-scale sample variations, this study combines the YOLOv8 framework with the StarNet network to design a new efficient algorithm. This new network can accurately identify challenging samples.

In the backbone feature extraction network, we designed a new framework based on the simple and efficient concept of StarNet. We introduced the CAA attention mechanism to construct the StarCAA module, allowing for improved accuracy in a lightweight deployment. Compared to existing complex handcrafted networks, this feature extraction network reduces computational and parameter costs by half compared to mainstream backbones like CPSDarknet and EfficientViT while maintaining high performance. Due to its simplicity, it exhibits excellent generalization capabilities. In the neck, we designed the PFDNet (Pyramid Focus and Diffusion Network) pyramid network, which preserves more contextual information through focus and diffusion mechanisms. The shared convolution and integrated DEAConv convolution modules in the detection head work synergistically to enhance the information processing capabilities of complex samples, reducing missed detections of challenging samples such as blurry and low-light conditions. The overall network structure achieves optimal lightweight results through shared parameters.

Ablation experiments on the backbone network, pyramid network, detection head, and modules demonstrate the effectiveness of the proposed method. Further comparative experiments with high-performance network algorithms also verify the superior performance of our algorithm. Compared to YOLOv8, our algorithm shows a 4% improvement in mAP@0.5, reduces the model size to less than half of mainstream models, and exhibits better performance across different traffic application datasets. This algorithm achieves precise detection of traffic signs in complex autonomous driving scenarios while maintaining generalization capabilities.

There is still room for improvement in the proposed model. Future work will consider further tuning of additional modules within the simple network structure to enhance detection speed and generalization capabilities without compromising the existing detection performance. This will enable the model to be applied not only to autonomous driving target detection in complex environments but also to explore its advantages in other fields through experimental investigations.

Author Contributions: Conceptualization and methodology, Z.C. and T.Z.; software, Z.C.; validation, Z.C. and T.Z; formal analysis, Z.C. and T.Z.; investigation, Z.C. and T.Z; resources, Z.C.; data curation, Z.C.; writing—original draft preparation, Z.C. and T.Z.; writing—review and editing, Z.C. and F.L.; visualization, Z.C.; supervision, T.Z.; project administration, T.Z.; funding acquisition, T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Yuxiu Innovation Project of NCUT (Project No. 2024NCU-TYXCX211).

Data Availability Statement: The data sources used in this article are CCTSDB, TT100K, GTSDB, and Roadsign. Specific data can be obtained by contacting the corresponding author.

Acknowledgments: We thank Xuecheng Wang for his guidance and support on the project.

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

References

1. Abuadbba, A.; Rhodes, N.; Moore, K.; Sabir, B.; Wang, S.; Gao, Y. DeepiSign-G: Generic Watermark to Stamp Hidden DNN Parameters for Self-contained Tracking. *arXiv* **2024**, arXiv:2407.01260.
2. Barodi, A.; Bajit, A.; Zemmouri, A.; Benbrahim, M.; Tamtaoui, A. Improved deep learning performance for real-time traffic sign detection and recognition applicable to intelligent transportation systems. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 249294472. [CrossRef]
3. Trappey, A.J.; Shen, O.T. A universal traffic sign detection system using a novel self-training neural network modeling approach. *Adv. Eng. Inform.* **2024**, *62*, 102674. [CrossRef]
4. Bao, D.; Gao, R. YED-YOLO: An object detection algorithm for automatic driving. *Signal Image Video Process.* **2024**, 1–9. [CrossRef]
5. Agrawal, S.; Chaurasiya, R.K. Ensemble of SVM for accurate traffic sign detection and recognition. In Proceedings of the 1st International Conference on Graphics and Signal Processing, Singapore, 24–27 June 2017; pp. 10–15.
6. Ren, X.; Zhi, M. An overview of traffic sign detection and recognition algorithms. In Proceedings of the Thirteenth International Conference on Graphics and Image Processing (ICGIP 2021), Kunming, China, 18–20 August 2021; pp. 618–626.
7. Yazdan, R.; Varshosaz, M. Improving traffic sign recognition results in urban areas by overcoming the impact of scale and rotation. *ISPRS J. Photogramm. Remote. Sens.* **2021**, *171*, 18–35. [CrossRef]
8. Chen, Y.; Zhang, P.; Li, Z.; Li, Y.; Zhang, X.; Meng, G.; Xiang, S.; Sun, J.; Jia, J. Stitcher: Feedback-driven data provider for object detection. *arXiv* **2020**, arXiv:2004.12432.
9. Lin, Z.; Ji, K.; Leng, X.; Kuang, G. Squeeze and Excitation Rank Faster R-CNN for Ship Detection in SAR Images. *IEEE Geosci. Remote. Sens. Lett.* **2018**, *16*, 751–755. [CrossRef]
10. Wang, G.; Zhuang, Y.; Chen, H.; Liu, X.; Zhang, T.; Li, L.; Dong, S.; Sang, Q. FSoD-Net: Full-scale object detection from optical remote sensing imagery. *IEEE TGRS* **2021**, *60*, 1–18. [CrossRef]
11. Huang, M.; Wan, Y.; Gao, Z.; Wang, J. Real-time traffic sign detection model based on multi-branch convolutional reparameterization. *J. Real-Time Image Process.* **2023**, *20*, 57. [CrossRef]
12. Geng, H.; Liu, Z.; Jiang, J.; Fan, Z.; Li, J. Embedded road crack detection algorithm based on improved YOLOv8. *J. Comput. Appl.* **2024**, *44*, 1613.
13. Zeng, G.; Wu, Z.; Xu, L.; Liang, Y. Efficient Vision Transformer YOLOv5 for Accurate and Fast Traffic Sign Detection. *Electronics* **2024**, *13*, 880. [CrossRef]
14. Xu, X.; Zhao, M.; Shi, P.; Ren, R.; He, X.; Wei, X.; Yang, H. Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN. *Sensors* **2022**, *22*, 1215. [CrossRef] [PubMed]
15. Bi, X.; Hu, J.; Xiao, B.; Li, W.; Gao, X. IEMask R-CNN: Information-Enhanced Mask R-CNN. *IEEE Trans. Big Data* **2022**, *9*, 688–700. [CrossRef]
16. Chen, M.; Yu, L.; Zhi, C.; Sun, R.; Zhu, S.; Gao, Z.; Ke, Z.; Zhu, M.; Zhang, Y. Improved faster R-CNN for fabric defect detection based on Gabor filter with Genetic Algorithm optimization. *Comput. Ind.* **2022**, *134*, 103551. [CrossRef]
17. Bai, D.; Sun, Y.; Tao, B.; Tong, X.; Xu, M.; Jiang, G.; Chen, B.; Cao, Y.; Sun, N.; Li, Z. Improved single shot multibox detector target detection method based on deep feature fusion. *Concurr. Comput. Pract. Exp.* **2021**, *34*, e6614. [CrossRef]
18. Krishna, H.; Jawahar, C.V. Improving small object detection. In Proceedings of the 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), Nanjing, China, 26–29 November 2017; pp. 340–345.

19. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
20. Zhang, J.; Zou, X.; Kuang, L.-D.; Wang, J.; Sherratt, R.S.; Yu, X. CCTSDB 2021: A more comprehensive traffic sign detection benchmark. *Hum.-Centric Comput. Inf. Sci.* **2022**, *12*, 23. [CrossRef]
21. He, X.; Li, T.; Yang, Y. Improved traffic sign detection algorithm based on improved YOLOv8s. *J. Comput. Electron. Inf. Manag.* **2024**, *12*, 38–45. [CrossRef]
22. Wu, T.; Dong, Y. YOLO-SE: Improved YOLOv8 for remote sensing object detection and recognition. *Appl. Sci.* **2023**, *13*, 12977. [CrossRef]
23. Li, S.; Shi, T.; Well, F. Improved road damage detection algorithm of YOLOv8. *Comput. Eng. Appl.* **2023**, *59*, 165–174. [CrossRef]
24. 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.
25. Bi, C.; Wang, J.; Duan, Y.; Fu, B.; Kang, J.-R.; Shi, Y. MobileNet Based Apple Leaf Diseases Identification. *Mob. Netw. Appl.* **2020**, *27*, 172–180. [CrossRef]
26. Nan, Y.; Ju, J.; Hua, Q.; Zhang, H.; Wang, B. A-MobileNet: An approach of facial expression recognition. *Alex. Eng. J.* **2022**, *61*, 4435–4444. [CrossRef]
27. Wang, W.; Li, Y.; Zou, T.; Wang, X.; You, J.; Luo, Y. A Novel Image Classification Approach via Dense-MobileNet Models. *Mob. Inf. Syst.* **2020**, *2020*, 7602384. [CrossRef]
28. Guo, G.; Zhang, Z. Road damage detection algorithm for improved YOLOv5. *Sci. Rep.* **2022**, *12*, 15523. [CrossRef]
29. Hao, J.; Yang, J.; Han, S.; Wang, Y. YOLOv4 highway pavement crack detection method using Ghost module and ECA. *J. Comput. Appl.* **2023**, *43*, 1284.
30. Pan, J.; Bulat, A.; Tan, F.; Zhu, X.; Dudziak, L.; Li, H.; Tzimiropoulos, G.; Martinez, B. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 294–311.
31. Chen, J.; Kao, S.-h.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, don't walk: Chasing higher FLOPS for faster neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
32. Ma, X.; Dai, X.; Bai, Y.; Wang, Y.; Fu, Y. Rewrite the Stars. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 5694–5703.
33. Li, J.; Wang, H.; Xu, Y.; Liu, F. Road Object Detection of YOLO Algorithm with Attention Mechanism. *Front. Signal Process.* **2021**, *5*, 9–16. [CrossRef]
34. Shamsolmoali, P.; Zareapoor, M.; Chanussot, J.; Zhou, H.; Yang, J. Rotation Equivariant Feature Image Pyramid Network for Object Detection in Optical Remote Sensing Imagery. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 1–14. [CrossRef]
35. Zhou, Y.; Yang, X.; Zhang, G.; Wang, J.; Liu, Y.; Hou, L.; Jiang, X.; Liu, X.; Yan, J.; Lyu, C. Mmrotate: A rotated object detection benchmark using pytorch. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 7331–7334.
36. Zhang, W.; Jiao, L.; Li, Y.; Huang, Z.; Wang, H. Laplacian Feature Pyramid Network for Object Detection in VHR Optical Remote Sensing Images. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 1–14. [CrossRef]
37. Zhang, K.; Bello, I.M.; Su, Y.; Wang, J.; Maryam, I. Multiscale depthwise separable convolution based network for high-resolution image segmentation. *Int. J. Remote. Sens.* **2022**, *43*, 6624–6643. [CrossRef]
38. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
39. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
40. Cai, X.; Lai, Q.; Wang, Y.; Wang, W.; Sun, Z.; Yao, Y. Poly kernel inception network for remote sensing detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 27706–27716.
41. Wang, C.; Yeh, I.; Liao, H. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv* **2024**, arXiv:2402.13616.
42. Tan, R.T. Visibility in bad weather from a single image. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, 23–28 June 2008; pp. 1–8.
43. Guo, C.-L.; Yan, Q.; Anwar, S.; Cong, R.; Ren, W.; Li, C. Image dehazing transformer with transmission-aware 3d position embedding. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5812–5820.
44. Chen, Z.; He, Z.; Lu, Z.-M. DEA-Net: Single Image Dehazing Based on Detail-Enhanced Convolution and Content-Guided Attention. *IEEE Trans. Image Process.* **2024**, *33*, 1002–1015. [CrossRef] [PubMed]