mathematics

MDPI

Article

Light "You Only Look Once": An Improved Lightweight Vehicle-Detection Model for Intelligent Vehicles under Dark Conditions

Tianrui Yin [1], Wei Chen [1], Bo Liu [2], Changzhen Li [3] and Luyao Du [1,*]

[1] School of Automation, Wuhan University of Technology, Wuhan 430070, China; 320973@whut.edu.cn (T.Y.); greatchen@whut.edu.cn (W.C.)
[2] Wuhan Zhongyuan Electronics Group Co., Ltd., Wuhan 430010, China; liubo@ceckc.cn
[3] School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China; changzhen.li@whut.edu.cn
* Correspondence: duluyao@whut.edu.cn; Tel.: +86-158-2715-3041

Abstract: Vehicle detection is crucial for traffic surveillance and assisted driving. To overcome the loss of efficiency, accuracy, and stability in low-light conditions, we propose a lightweight "You Only Look Once" (YOLO) detection model. A polarized self-attention-enhanced aggregation feature pyramid network is used to improve feature extraction and fusion in low-light scenarios, and enhanced "Swift" spatial pyramid pooling is used to reduce model parameters and enhance real-time nighttime detection. To address imbalanced low-light samples, we integrate an anchor mechanism with a focal loss to improve network stability and accuracy. Ablation experiments show the superior accuracy and real-time performance of our Light-YOLO model. Compared with EfficientNetv2-YOLOv5, Light-YOLO boosts mAP@0.5 and mAP@0.5:0.95 by 4.03 and 2.36%, respectively, cuts parameters by 44.37%, and increases recognition speed by 20.42%. Light-YOLO competes effectively with advanced lightweight networks and offers a solution for efficient nighttime vehicle-detection.

Keywords: intelligent vehicles; vehicle detection; lightweight network; feature pyramid network

MSC: 68T07

Citation: Yin, T.; Chen, W.; Liu, B.; Li, C.; Du, L. Light "You Only Look Once": An Improved Lightweight Vehicle-Detection Model for Intelligent Vehicles under Dark Conditions. *Mathematics* **2024**, *12*, 124. https://doi.org/10.3390/math12010124

Academic Editor: Konstantin Kozlov

Received: 16 October 2023
Revised: 13 December 2023
Accepted: 28 December 2023
Published: 29 December 2023

1. Introduction

In recent years, the global vehicle population has grown rapidly, leading to an increase in the complexity of traffic scenarios and posing significant challenges to driving safety. Intelligent driving is a key transportation innovation [1] that aims to enhance safety and efficiency. Among the various aspects of intelligent driving, vision-based vehicle-detection plays a crucial role in driving decision-making and automated control. However, multiple challenges are encountered in complex traffic environments, including variations in lighting conditions, diverse vehicle targets, and adverse weather conditions [2]. Moreover, the increasing complexities of current models result in high computational costs, and the intricate structures of vehicle-detection algorithms and their expensive hardware requirements hinder their use in edge and mobile-terminal devices [3]. Therefore, the creation of efficient and lightweight vehicle-detection algorithms is vital.

With continuous breakthroughs in artificial intelligence and computing power, target-detection algorithms have recently been the subject of rapid advancements that affect security monitoring [4,5], edge detection [6–8], autonomous driving [9–11], and pose-detection [12–14] domains. However, regardless of such advancements, detecting vehicles in low-light conditions poses a significant challenge. Existing night-condition technologies create unique obstacles, including vehicle appearances being blurred, deformed, or partially obscured, which current night-detection technologies struggle to address efficiently. Contemporary deep learning networks are powerful, but require substantial computational and

temporal resources. This presents significant obstacles for implementing these networks in real-time situations, particularly in nocturnal settings.

Notably, contemporary target-detection methods have begun shifting from traditional handcrafted models [15] to generalized deep-learning varieties. Architecturally, target detection has moved from two- to single-stage detection using multiscale feature fusion to support lightweight implementations. As such, remarkable detection performance has been achieved using publicly available datasets.

Two-stage convolutional neural networks (CNNs) divide the detection problem into candidate region generation [16] and classification activities. Representative models include the region-based CNN (R-CNN) [17], the "Fast" R-CNN [18], the "Faster" R-CNN [19], and the "Mask" R-CNN [20]. However, region candidate generation requires a significant amount of computational resources and is time-consuming, rendering such methods prohibitively costly for the real-time requirements of nighttime vehicle-detection scenarios.

In contrast, one-stage algorithms directly perform the regression and classification of candidate boxes. Typical versions include "You Only Look Once" (YOLO) models, for which there are currently eight versions [21–28]; the single-shot "multibox" detector (SSD); the deconvolutional SSD [29]; and the "Exceeding" YOLO (YOLOX) [30], among others. These algorithms complete feature sharing in a single training session, greatly improving speed. However, due to their inherent structural characteristics, one-stage detectors often suffer from class imbalances between positive and negative samples which decrease their accuracy, particularly in challenging, low-light conditions.

Lightweight networks are being designed for portability and mobility, which are needed for driving target-detection. Representative models include MobileNet [31–33], EfficientNet [34], GhostNet [35], EfficientDet [36], and YOLOv4-tiny [37]. They achieve optimal speeds, but often suffer trade-offs in terms of stability and accuracy. Although these achievements have found extensive use across diverse fields, the distinct domain of nocturnal vehicle-detection presents challenges that remain unresolved. To address these challenges in nighttime vehicle-detection, this study introduces Light-YOLO and makes a number of significant contributions to science and safety:

- We provide an efficient and accurate scale fusion attention module (SFAM) that aggregates features into a multilevel feature pyramid to enhance the accuracy of nighttime vehicle-detection. Our novel polarized self-attention-enhanced aggregated (PSEA) feature pyramid network (FPN) and its efficient pyramid-split attention module (PSA) is used to eliminate irrelevant contextual information, which helps overcome the efficiency–accuracy tradeoff.
- We provide a powerful feature-enhancement module (FEM) that mitigates the information loss caused by feature channel reductions, resulting in strong, fused multiscale feature information for accurate vehicle-detection under varying lighting conditions in mobile or edge environments.
- We leverage the lightweight EfficientNetv2 backbone network and add a "Swift" spatial pyramid pooling (SPP) layer to minimize computational resources and memory constraints and encourage portability and mobile use. The network operates efficiently while capturing comprehensive features, further prioritizing accuracy.
- We provide a stable and accurate anchor box mechanism using a finely tuned K-means clustering algorithm to detect targets with precision, even in nighttime scenarios where vehicles would otherwise appear blurred, deformed, or partially obscured. We incorporate a focal-loss mechanism to overcome the imbalance between positive and negative samples and maximize target recall and model stability.

2. Related Work

To foster computer vision advancements in a variety of scene conditions, several research teams have produced specialized datasets. Satzoda et al. [38] introduced a comprehensive annotated dataset at the Smart Safety Car Laboratory comprising over 5000 frames for evaluation and benchmarking and encompassing diverse and intricate traffic and light-

ing conditions. To enhance the quality of imagery and the challenges posed by adverse lighting conditions, Lin et al. [39] introduced an innovative approach known as AugGAN, a generative adversarial network (GAN). Their approach facilitates domain transformations and seamless transitions from day to night while preserving the integrity of image objects. Ye et al. [40] pioneered an unsupervised domain adaptation framework, based on a transformer architecture, with a focus on nocturnal aerial object tracking. Their framework generates training patches via object discovery and employs transformer-based bridging-layer columns to facilitate domain alignment, thereby enhancing tracking performance in nighttime conditions via adversarial training.

Due to the heavy computational and temporal resources required by contemporary deep-learning networks like the ones mentioned, our research addresses the most pivotal areas of improvement: multiscale feature fusion and a lightweight backbone.

2.1. Multiscale Feature Fusion

Advanced feature maps capture and track ample global data, possess an expanded receptive scope, and exhibit enhanced semantic representations. Consequently, high-level versions are used for precise target localization, whereas low-level maps provide superior spatial resolution of edges, contours, and textures. An adept target-detection model will proficiently classify targets; thus, one needs an amalgamation of multiscale feature maps for effective and balanced performance. An FPN [41] is used to fuse multiscale features into integrated maps for retention and prioritization. Notably, there are several versions, including the path aggregation network (PANet) [42], which incorporates a bottom-up fusion path; the neural architecture search (NAS)-based FPN [43]; and the bidirectional FPN (BiFPN) [44]. However, these are not effective enough for our task, because their feature channel dimensionality reductions result in feature-map information losses, and their maps accumulate extraneous contextual data unrelated to the detection task. Hence, both computational efficiency and target recognition fall short of our targeting and recognition requirements.

Figure 1 provides a high-level illustration of FPN, PANet, and BiFPN functionality. The popular EfficientNet-YOLO network incorporates the PANet structure, and EfficientDet (built on EfficientNet) leverages a BiFPN to flexibly control network size by searching for and reusing the most effective FPN blocks.

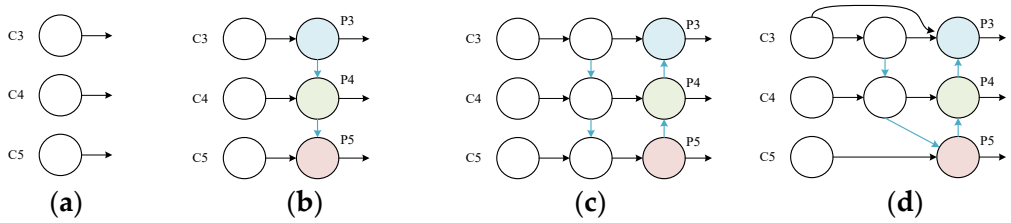


Figure 1. Various variations of the feature pyramid network: (**a**) basic starting dimension, (**b**) FPN, (**c**) PANet, and (**d**) BiFPN.

Li et al. [45] introduced the "multi-attention" FPN to address noise and background interference in vehicle-target-detection tasks via the fusion of attention information within an FPN. Gu et al. [46] presented an improved FPN for small-target vehicle detection that seamlessly integrates deeper and shallower semantic information without increasing computational costs, by using cross-scale connecting lines. Although the FPN's multiscale feature fusion has significantly advanced object detection in recent years, feature losses, inadequate small-target handling, and resource impracticalities persist.

2.2. Backbone

A seemingly straightforward deep-learning approach to providing onboard, real-time vehicle-detection would use a lightweight backbone. Hence, numerous researchers have investigated ways to apply them to general vehicle-detection. For example, Chen et al. [47] proposed an improved SSD for rapid detection using MobileNetv2 as the backbone, which approached real-time performance. With approximately 5/11 of the original model's complexity, inference speeds were improved, achieving an incredibly fast single inference time of 73 ms while sustaining impressive accuracy. To address the computational constraints of edge devices in autonomous driving scenarios, Chen et al. [48] introduced a domain-specific lightweight network that employs a DenseNet201 backbone [49] that combines its best features with YOLO, MobileNet, and online capabilities. By leveraging group convolutions and replacing some of the dense blocks with alternating blocks, model embedding was made possible while maintaining excellent speed and accuracy.

For Light-YOLO, we sought to determine the most efficient backbone. Notably, EfficientNetv2 was found to outperform MobileNet, EfficientNet, GhostNet, and others in terms of recognition accuracy and speed. Nevertheless, extant models with capabilities similar to those which we require have not been sufficiently validated under nighttime and adverse conditions. This validation is crucial to robustness and dependability. EfficientNetv2 employs NAS to determine the types of convolutional operations needed (i.e., MBConv and fused-MBConv) and calculates layer numbers, kernel sizes, and expansion ratios to maximize training speed with minimal overhead.

3. Methodology

Light-YOLO applies multiscale feature fusion with the lightweight EfficientNetv2 backbone using a stable and accurate anchor-box mechanism to strike a balance among efficiency, stability, and detection accuracy. In this section, we provide a detailed overview of its framework, algorithm, architecture, and full operation.

For the backbone design, we replaced the standard MBConv with a fused-MBConv to improve training speeds while reducing parameter increments during the early stages of model operation. Figure 2 illustrates a comparison of these structures.

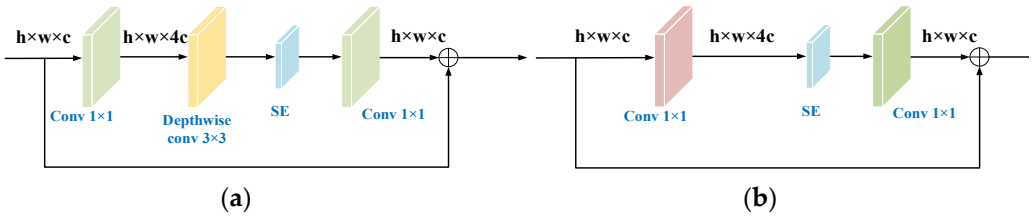


Figure 2. High-level (**a**) MBConv and (**b**) fused-MBConv architectures.

As illustrated in Figure 3, the overall Light-YOLO architecture comprises the EfficientNetv2 backbone, the PSEA-FPN, and a prediction layer. First, image standardization and background enrichment operations take place, including resizing and data augmentation. The backbone is used to extract image features at different scales, incorporating the Swift-SPP to enhance feature extraction. Subsequently, the PSEA-FPN fuses semantic and positional features. Finally, the prediction module determines the category of the target.

3.1. PSEA-FPN

As illustrated in Figure 4, feature map fusion is simplified to improve FPN efficiency. Notably, the PSEA-FPN structure comprises crucial PSA, FEM, and SFAM components.

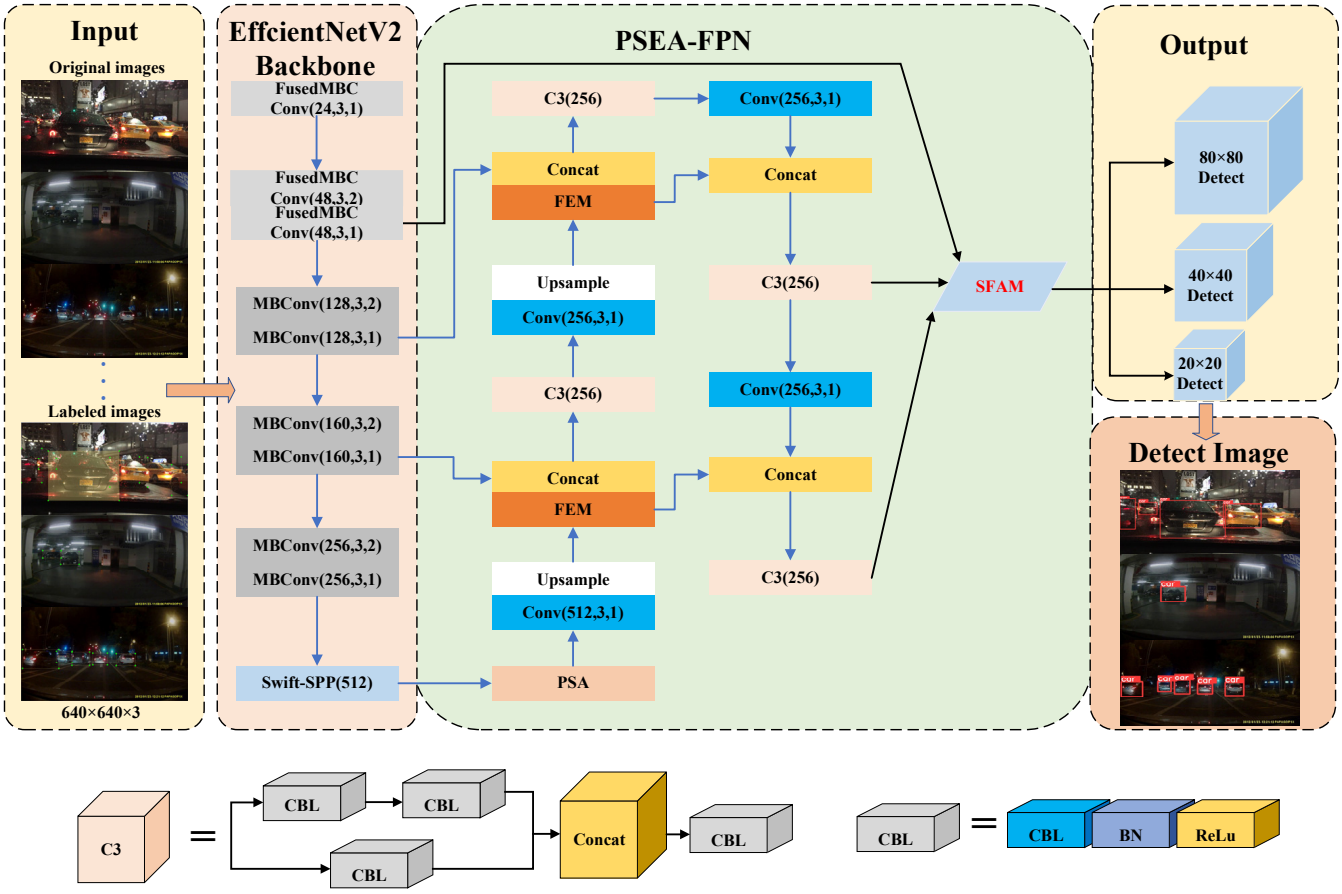


Figure 3. Overall architecture of Light-YOLO.

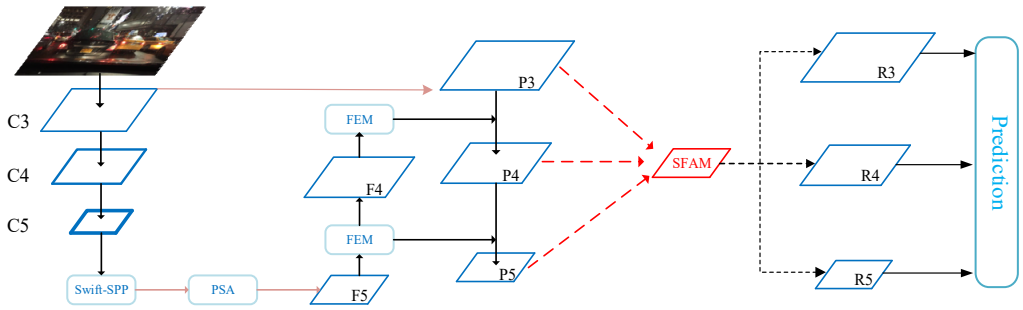


Figure 4. Structure of the PSEA-FPN.

3.1.1. Feature Fusion

As depicted in Figure 4, like the conventional PANet, our PSEA-FPN comprises top-down and bottom-up branches. We denote the output of the backbone as $\{C_3, C_4, C_5\}$, and $\{F_4, F_5\}$ is generated by the bottom-up path inside the FPN. To enhance detection efficiency, we removed node F_3, as it has only one input edge, rendering its contribution negligible. Feature map F_5 is generated from C_5 through the Swift-SPP and PSA and is fused with features from lower levels. In the top-down path, following each fusion action, the feature map expands its receptive field through the FEM and is further fused through the bottom-up path to ultimately generate $\{P_3, P_4, P_5\}$. Finally, via SFAM fusion, feature maps $\{R_3, R_4, R_5\}$ are created with dimensions of 80×80, 40×40, and 20×20, respectively, making them suitable for predictions at three scales.

3.1.2. PSA

The PSA [38] ensures that the network focuses on target objects while disregarding redundant background information. Attention mechanisms are broadly categorized into channel (e.g., squeeze-and-extraction [50] and efficient channel [51]) and spatial (self-attention [52]) types. Dual-attention mechanisms have also improved recently, with notable examples including the channel block attention module (CBAM) [53] and the dual attention module [54].

For lightweight vehicle-detection at night, we employ PSA, due to its more intricate attention mechanism, which is based on dual attention [55]. Notably, it effectively models long-range dependencies across high-resolution inputs and outputs with relatively low computational overhead. The structural diagram is shown in Figure 5.

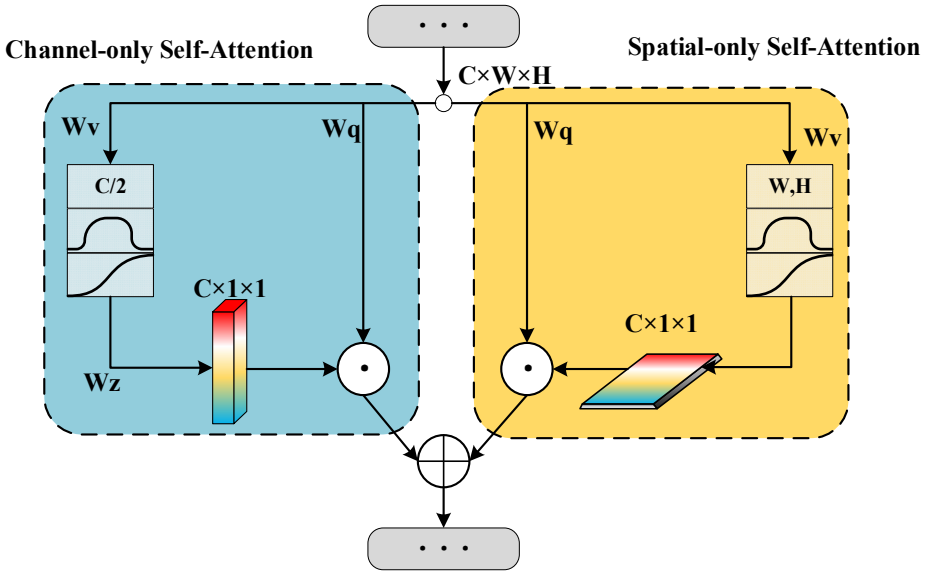


Figure 5. Polarized self-attention block.

The PSA is divided into channel and spatial branches, and the weight calculation formulas for the channel and spatial branches are presented as follows:

$$\mathbf{A}^{ch}(\mathbf{X}) = F_{SG}\left[\mathbf{W}_{z|\theta_1}\left(\sigma_1(\mathbf{W}_v(X)) \times F_{SM}\left(\sigma_2(\mathbf{W}_q(\mathbf{X}))\right)\right)\right] \tag{1}$$

$$\mathbf{A}^{sp}(\mathbf{X}) = F_{SG}\left[\sigma_3\left(F_{SM}\left(\sigma_1\left(F_{GP}(\mathbf{W}_q(\mathbf{X}))\right)\right)\right) \times \sigma_2(\mathbf{W}_v(\mathbf{X}))\right] \tag{2}$$

respectively, which are designed to maintain a high resolution.

Simultaneously, the input tensor is fully folded along the corresponding dimensions to mitigate the information loss caused by dimensionality reductions. Within the attention pathway module, the SoftMax function is applied to the smallest tensor to expand its attention scope, followed by dynamic mapping using a sigmoid function to enhance the preserved information.

Based on the results of these two branches, parallel and serial fusion approaches are formulated as follows:

$$PSA_P(\mathbf{X}) = \mathbf{Z}^{ch} + \mathbf{Z}^{sp} = \mathbf{A}^{ch}(\mathbf{X}) \odot^{ch} \mathbf{X} + \mathbf{A}^{ch}(\mathbf{X}) \odot^{sp} \mathbf{X} \tag{3}$$

$$PSA_s(\mathbf{X}) = \mathbf{Z}^{sp}\left(\mathbf{Z}^{ch}\right) = \mathbf{A}^{sp}\left(\mathbf{A}^{ch}(\mathbf{X}) \odot^{ch} \mathbf{X}\right) \odot^{sp} \mathbf{A}^{ch}(\mathbf{X}) \odot^{ch} \mathbf{X} \tag{4}$$

From the perspective of fusion methods, PSA is similar to CBAM, differing primarily in how they combine the results from the channel and spatial branches (i.e., parallel or in series). However, CBAM often employs fully connected and convolutional lay-

ers to obtain attention weights, which are not as effective for retaining knowledge. In contrast, PSA utilizes a self-attention network to derive attention weights and applies dimensionality reductions to certain maps to achieve effective long-range modeling without increasing complexity.

3.1.3. FEM

The FEM is a novel module introduced to capture receptive fields from feature maps of different scales. Its structure, illustrated in Figure 6, consists of a multibranch convolutional layer and a multibranch pooling layer. The convolutional layer is employed to capture receptive fields of varying sizes, and the pooling layer integrates information from the receptive fields of the three branches.

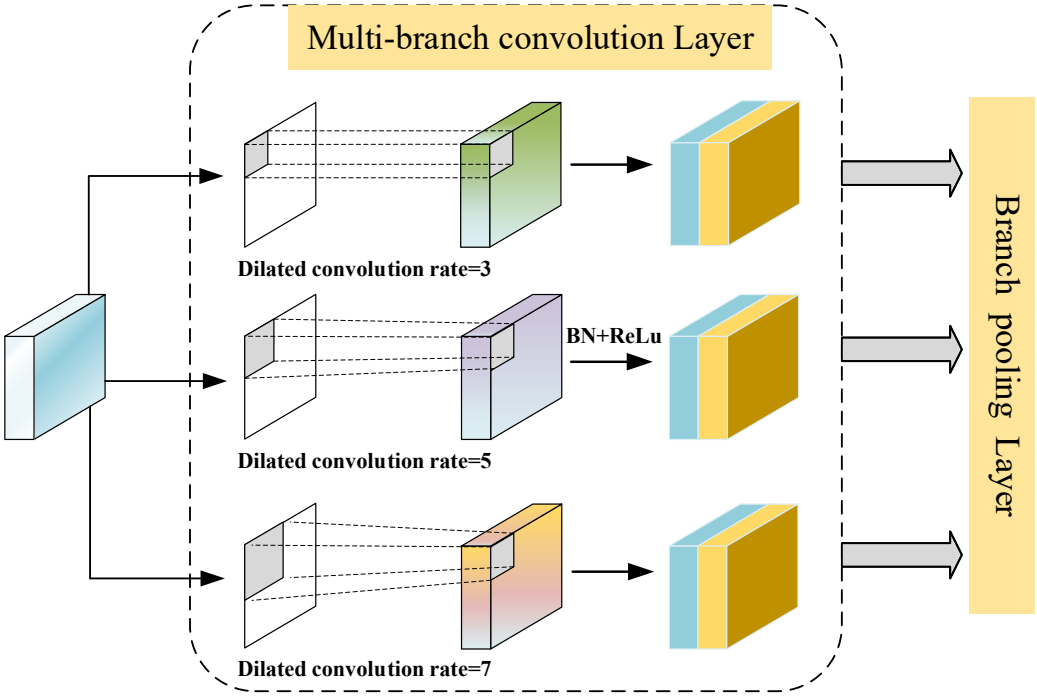


Figure 6. Structure of the feature-enhancement module.

The multi-branch convolutional layer is composed of dilated convolution, batch normalization, and rectified linear unit (ReLU) activation functions. Each branch in the multibranch convolutional layer employs dilated convolutions with the same kernel size, 3×3. However, they differ in their dilation rates, d, which we set to 1, 3, and 5 in this study. Doing so expands the receptive field and captures more contextual information, which is expressed as follows:

$$\begin{aligned} r_1 &= d \times (k-1) + 1 \\ r_n &= d \times (k-1) + r_{n-1} \end{aligned} \tag{5}$$

where k and r_i represent the convolution kernel size and dilation rate, respectively, and d denotes the convolution stride.

The multi-branch pooling layer combines information from different parallel branches. During training, we employ an averaging operation to balance the contributions of the parallel branches. The equation is as follows:

$$y_p = \frac{1}{B}\sum_{i=1}^{B} y_i \tag{6}$$

where y_p represents the output of the multibranch pooling layer, and B represents the number of parallel branches. We set B to three in this case.

The FEM employs dilated convolutions to adaptively learn different receptive fields from various feature maps, depending on the different vehicle features detected at night, thereby enhancing the accuracy of multiscale object detection.

3.1.4. SFAM

The goal of the SFAM is to aggregate multi-level multiscale features into a multi-level feature pyramid. The first step involves a channel-wise summation of features of the same scale, resulting in an aggregated channel representation denoted as $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_i]$. Here, $\mathbf{X}$ represents the feature maps at different scales, denoted as $\mathbf{X}_i = Concat\left(\mathbf{X}_i^1, \mathbf{X}_i^2, \cdots, \mathbf{X}_i^L\right) \in \mathbb{R}^{W_i \times H_i \times C}$. Thus, each scale in the aggregated feature pyramid contains features of the same scale from different layers.

The second step introduces a channel-wise attention mechanism to excite features, focusing on channels that provide the greatest detection assistance. This leverages SENet, where the squeeze stage channel information is generated using global pooling. To fully capture channel dependencies, the subsequent excitation step employs two fully connected layers to learn the attention mechanism.

$$\mathrm{s} = \mathbf{F}_{ex}(\mathbf{z},\mathbf{W}) = \sigma(\mathbf{W}_2\delta(\mathbf{W_1 z})) \tag{7}$$

Among these, σ represents the ReLU operation, δ represents the sigmoid, $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{r} \times C}$, $\mathbf{W}_2 \in \mathbb{R}^{\frac{C}{r} \times C}$, and r is the reduction ratio, where r = 16 in our experiment. The final output is obtained by reweighting input X using activation s, as follows:

$$\tilde{\mathbf{X}}_i^c = \mathbf{F}_{scale}(\mathbf{X}_i^c, s_c) = s_c \cdot \mathbf{X}_i^c \tag{8}$$

where each element in $\tilde{\mathbf{X}}_i = \left[\tilde{\mathbf{X}}_i^1, \tilde{\mathbf{X}}_i^1, \cdots, \tilde{\mathbf{X}}_i^C\right]$ is rescaled for enhancement or weakening. A summary structural diagram of the SFAM is shown in Figure 7.

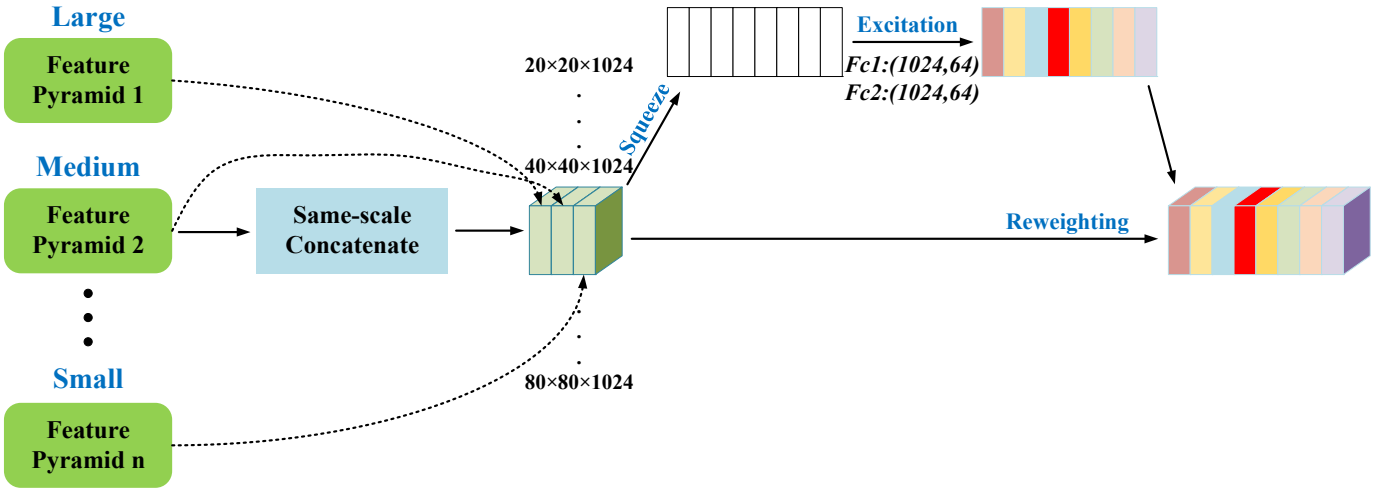


Figure 7. Illustration of the scale-wise feature aggregation module.

3.2. Swift-SPP

Spatial pyramids employ pooling layers of different kernel sizes to capture receptive fields of various scales, and subsequently fuse features to enrich the information in the feature maps. Considering the real-time requirements and the need for high detection speed in vehicle detection tasks, we applied the Swift-SPP to improve inference speeds (Figure 8).

Swift-SPP employs a multibranch parallel structure, eliminating the repetitive operations of the contemporary SPP and significantly improving operational speeds. It also replaces the pooling structure with a convolutional structure with a kernel size of 5×5 and a stride of one. The three parallel convolution operations have receptive fields equivalent to those of convolutions with sizes of 5×5, 9×9, and 13×13. This design not only enhances the network's detection speed, but it also enriches the information in the feature maps, thereby strengthening the network's feature-extraction capability.

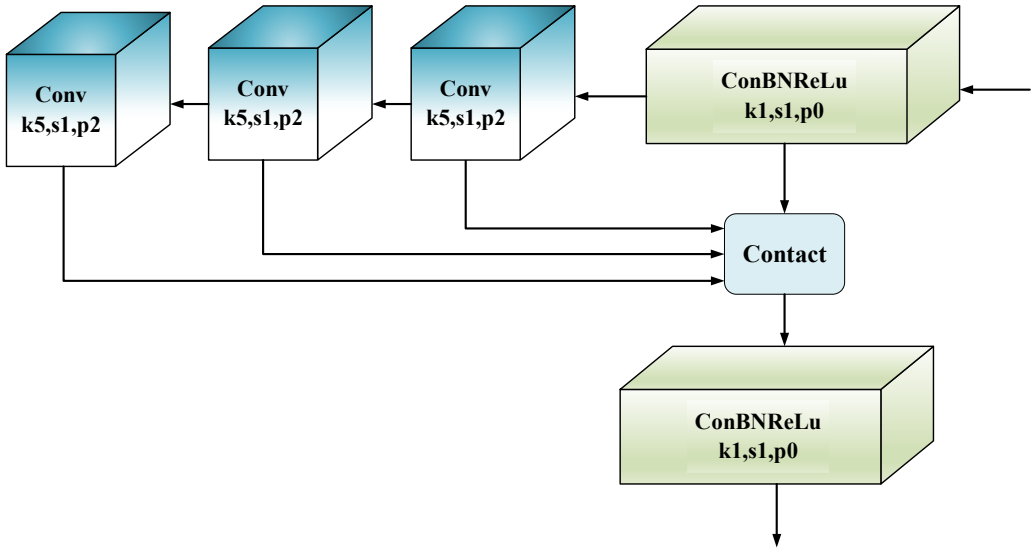


Figure 8. Structure of the Swift-SPP.

3.3. Sample Equalization (SE)

Class imbalances consistently pose challenges to object-detection accuracy. SSDs are known for their speed, but they often suffer from lower accuracy, due to the fundamental issue of class imbalance. Employing an anchor-based mechanism results in the generation of thousands of candidate boxes from a single feature map, with only a small fraction of these containing potential targets (i.e., positive samples), whereas the rest are considered negative samples. Negative samples are usually easy to distinguish and do not contribute significantly to the training process. However, when there are too many negative samples and they dominate the loss function, the training process tends to focus excessively on them, overshadowing the positive ones and leading to substantial loss.

To address these issues, we employ a novel focal-loss function derived from the standard cross-entropy loss. However, it is modified to address our research problem. The specific form is as follows:

$$\mathbf{CE}(p, y) = \begin{cases} -\log(p) & if\ y = 1 \\ -\log(1-p) & otherwise \end{cases} \tag{9}$$

where y is 1 or -1, representing the foreground and background, respectively. The value range of p is in (0, 1), which reflects the probability of the model predicting a positive outcome. Function p is defined as

$$p_t = \begin{cases} p & if\ y = 1 \\ 1-p & otherwise \end{cases} \tag{10}$$

By combining Equations (9) and (10), a simplified formula can be obtained as follows:

$$\mathbf{CE}(p, y) = \mathbf{CE}(p_t) = -\log(pt) \tag{11}$$

To solve the problem of imbalanced positive and negative samples, modulation and weight factors from the cross-entropy loss structure are introduced to help distinguish samples. The focal loss formula is as follows:

$$\mathbf{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(pt) \tag{12}$$

where modulation factor $(1-p_t)^{\gamma}$ is used to reduce the loss contribution of easily distinguishable samples (i.e., foreground or background). The larger the p_t, the easier the sample is to distinguish and the smaller the modulation factor. α_t is used to adjust the proportion between positive and negative sample losses, where α_t is the foreground category, and $1-\alpha_t$ is the corresponding background category.

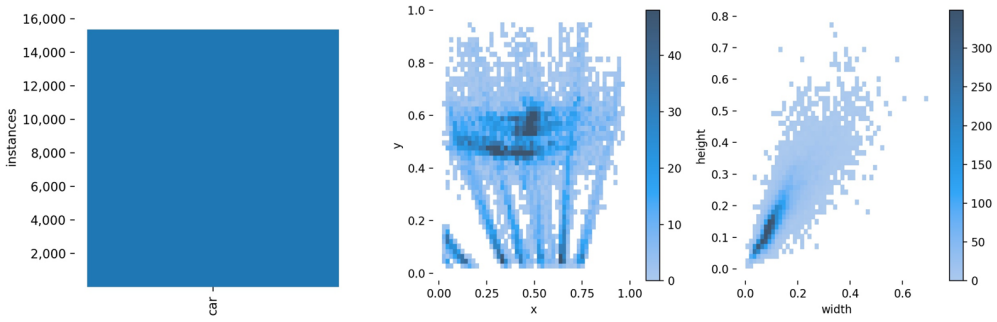
4. Experiments

4.1. Dataset

Our dataset consisted of 10,000 images obtained by extracting 6000 original frames from the classic and diverse BDD100K large-scale autonomous driving video dataset, focusing on nighttime driving scenes, and by capturing nighttime dashcam video frames, which resulted in 4000 more images. As shown in Figure 9, our self-assembled dataset contains a wide variety of nighttime driving scenarios, and its size promises good adaptability and robustness.

Figure 9. Nighttime vehicle dataset based on BDD100k and dashcam footage.

Our experimental dataset meets the requirements for diversity in terms of scenes, shapes, and lighting conditions, and it is suitable for training deep-learning networks. During model training, we employed data augmentation strategies to expand the dataset. The attributes are visualized in Figure 10. In Figure 10a, it can be seen that the dataset contains more than 14,000 labels. Figure 10b displays the central coordinate positions of the objects, with darker colors indicating higher label concentrations at those positions. Figure 10c illustrates the sizes of the objects, revealing that our dataset contains a good variety.



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

Figure 10. Dataset attribute visualization results: (**a**) number of labels, (**b**) locations of objects, (**c**) sizes of objects.

We partitioned the dataset into training and validation sets in an 80–20% ratio. Using a Python3.8 environment, we employed the open-source LabelImg1.8.6 tool for the manual annotation of regions encompassing target objects. Consequently, we produced the corresponding positional information files in an .XML format. For our experiment, we designated the objects as "cars." Subsequently, we standardized the .XML positional files and transformed them into .TXT files to facilitate nocturnal vehicle labeling.

4.2. Experimental Environment

Our experiments were conducted using Python 3.8 and the PyTorch11.0 framework. The development platform of was a 64-bit Linux system, and the processor was an Intel(R) Core(TM) i9-11900K CPU. To enhance training efficiency, an NVIDIA GeForce RTX 3090 GPU with CUDA 11.3 and CuDNN 10.0 were employed for graphics acceleration, facilitated through BAIDU AutoDL cloud server resources. Additionally, stochastic gradient descent was used to control the loss reduction, the batch size was set to 128, the initial learning rate was 0.01, and we used 200 training epochs.

4.3. Evaluation Criterion

We used standard precision (P), recall (R), average precision (AP), mean AP (mAP), number of parameters ($Params$), and speed (fps) to assess performance and accuracy. Higher values of P and R indicate higher detection accuracy, and mAP measures the overall model performance, which reflects the efficacy of training. Compared with P and R, mAP provides a more comprehensive estimation of algorithmic performance. In this experiment, we used mAP@0.5 and mAP@0.5:0.95 to provide a comprehensive evaluation.

$$P = \frac{T_P}{T_P + F_p} = \frac{T_p}{alldetections} \tag{13}$$

$$R = \frac{T_P}{T_P + F_N} = \frac{T_P}{allgroundtruths} \tag{14}$$

$$AP = \int_0^1 P_i(R_i) dR_i \tag{15}$$

$$mAP = \frac{1}{n} \sum_{j=1}^{n} AP_j \tag{16}$$

Because mAP reflects only the model accuracy, we tracked the number of model parameters required and inference speeds achieved. The fps measure reflects the algorithm's execution speed.

$$\text{FPS} = frameNum \ / \ elapsedTime \tag{17}$$

where *elapsedTime* represents a fixed period, and *frameNum* represents the number of frames processed within that period.

4.4. Ablation Experiment

4.4.1. Experimental Benchmark

Because our model is an improvement of the EfficientNetv2-YOLOv5, the latter serves as the baseline for our ablation experiments. Its metrics are shown in Figure 11. In Figure 11, with an increase in the number of training epochs, the values of the mAP@0.5 and mAP@0.5:0.95 scores gradually rise, whereas the loss values decrease. Model training attained relative stability after ~100 epochs, and the final training round consisted of 200 epochs, resulting in a mAP@0.5 score of 90.28%, a mAP@0.5:0.95 score of 42.07%, a box_loss of 0.0397, and an obj_loss of 0.0623, respectively. Thus, it is clear that there is room for improvement with Efficientnetv2-YOLOv5.

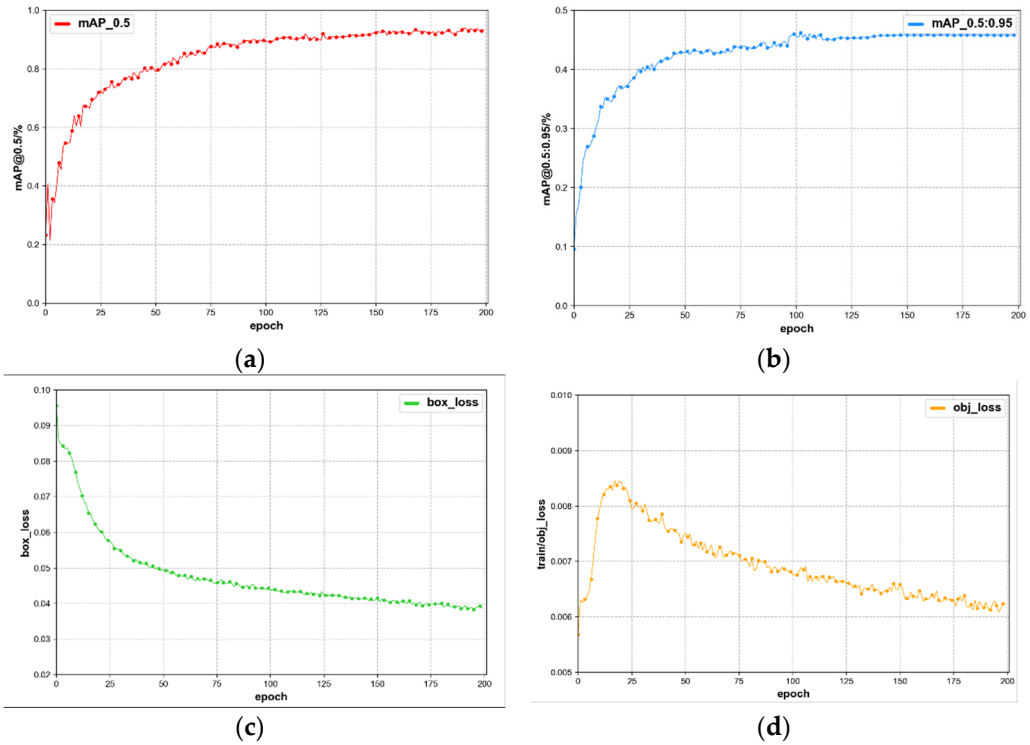


Figure 11. Training results of Efficienctnetv2-YOLOv5: (**a**) mAP@0.5, (**b**) mAP@0.5:0.95, (**c**) box_loss, and (**d**) obj_loss.

4.4.2. PSEA-FPN Internal Structure Validity Verification

The PSEA-FPN is a novel feature pyramid structure that includes several internal improvements. The model incorporates four small improvements that are sequentially removed or added for ablation testing: the F3 node, PSA, FEM, and SFAM (Table 1).

As shown in Table 1, Experiment 0 represents the full Efficientnetv2-YOLOv5 baseline. In Experiment 01, which involved deleting the F3 node, there was a slight decrease in the mAP@0.5 and mAP@0.5:0.95 scores, but there was a noticeable increase in detection speed (i.e., from 84.74 to 115.51 fps). In Experiment 06, where the PSA was added to Experiment 01, there were increases in the mAP@0.5 and mAP@0.5:0.95 scores of 1.30 and 0.62%, respectively. Experiment 02, which included the PSA module, showed improvements in both mAP@0.5 and mAP@0.5:0.95 scores, while also reducing the parameter count. In Experiment 04, the SFAM module was introduced, resulting in a significant improvement in both mAP@0.5 and mAP@0.5:0.95 scores, although there was a slight decrease in fps. Experiment 05 is an extension of Experiment 04, with the addition of the PSA module. The results demonstrate an increase in both the mAP@0.5 and the mAP@0.5:0.95 scores,

while effectively reducing parameters. Experiments 08 and 07 improved upon Experiments 06 and 02 by utilizing the FEM method, which resulted in increased mAP@0.5 and mAP@0.5:0.95 scores. Finally, Experiment 09 incorporated all improvements, including the PSEA-FPN and SFAM, atop Experiment 08. For Experiment 09, the mAP@0.5 and mAP@0.5:0.95 scores were 92.72 and 44.76%, respectively, making them 2.44 and 2.69% higher than the baseline. The detection speed also reached 102.31 fps, an increase of 17.57 fps over the baseline.

The visualization results of the internal improvements given by the PSEA-FPN are shown in Figure 12a. Although adding the PSA and FEM slightly increased the model parameter count, they significantly enhanced recognition accuracy. In summary, combining these four improvements not only improved the detection speed, but also significantly enhanced detection accuracy.

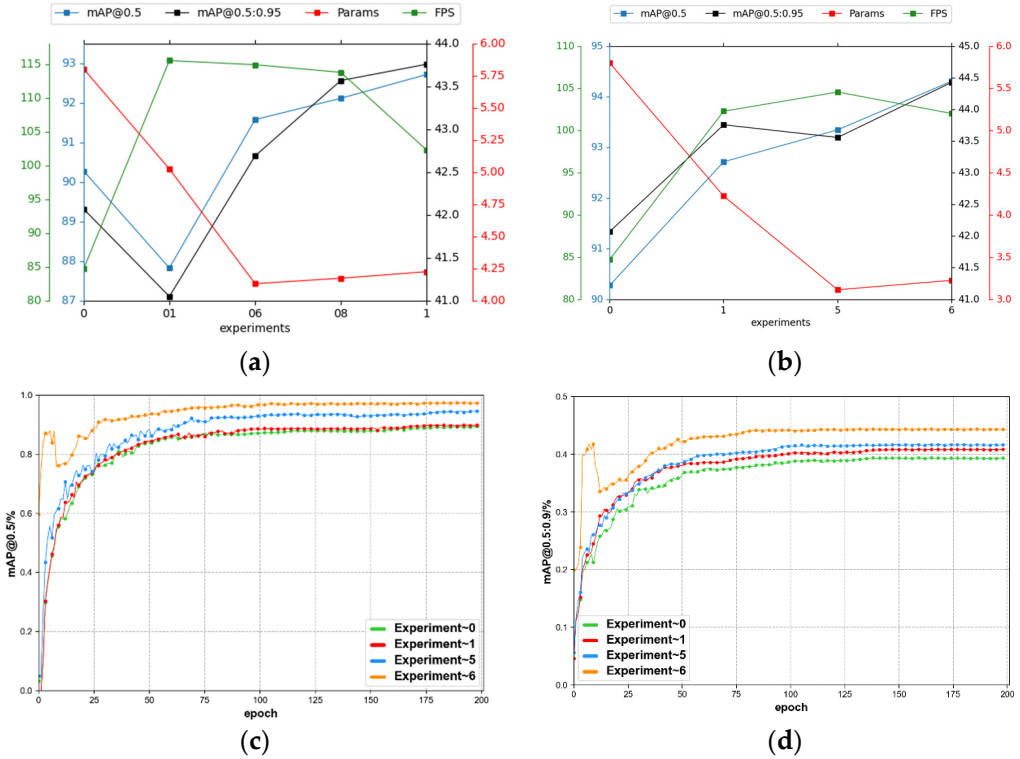


Figure 12. Visualization of ablation-experiment results: (**a**) ablation experiment for PSEA-FPN improvement points, (**b**–**d**) ablation experiment for light-YOLO improvement points.

Table 1. PSEA-FPN internal structure validation ablation experiments.

	Delete F3	PSA	FEM	SFAM	mAP@0.5/%	mAP@0.5:0.95/%	Params (M)	FPS/(f.sˆ-1)
0	×	×	×	×	90.28	42.07	5.803	84.74
01	√	×	×	×	87.83	41.05	5.026	115.51
02	×	√	×	×	90.58	42.69	4.922	80.65
03	×	×	√	×	88.91	41.64	5.975	78.12
04	×	×	×	√	90.54	42.39	5.885	84.49
05	×	√	×	√	91.89	42.83	5.263	81.87
06	√	√	×	×	91.58	42.69	4.133	114.89
07	×	√	√	×	91.79	42.77	5.117	80.43
08	√	√	√	×	92.12	43.57	4.175	113.77
09	√	√	√	√	92.72	44.76	4.226	102.31

4.4.3. Validation of the Effectiveness of Light-YOLO Improvement Points

PSEA-FPN represented the first improvement, Swift-SPP was the second, and SE was the third. The results are listed in Table 2. Experiment 0 used the Efficientnetv2-YOLOv5 baseline. Experiments 1–3 introduced single improvement factors to the baseline to demonstrate the effectiveness of each addition. The results of Experiments 1–3 show that each improvement point (i.e., PSEA-FPN, Swift-SPP, and SE, in that order), led to improvements in the *mAP* score. The most significant improvement was provided by the PSEA-FPN, which increased the mAP@0.5 from 90.28 to 92.72%, a gain of 2.24%. It also reduced the number of model parameters from 5.803 M to 4.226 M and improved the detection speed from 84.74 to 102.31 fps. Swift-SPP, with its parallel structure, reduced model complexity, resulting in a 13.08 fps increase in detection speed compared with the baseline. It also achieved a mAP@0.5 score of 90.91%, which is a 0.63% improvement. SE improved the mAP@0.5 score by 1.94% while maintaining the strong detection speed.

Experiments 4–6 were combinations of improvement points, still using Efficientnetv2-YOLOv5 as the baseline. The results show that Experiment 4, which added the Swift-SPP strategy to Experiment 3, increased the mAP@0.5 score by 2.41% over the baseline, while achieving a certain degree of lightweight performance and fps increase. Experiment 5, a combination of the PSEA-FPN and Swift-SPP, achieved a *mAP* score of 93.35%, a 3.07% improvement over the baseline, with a 19.84 fps increase in detection speed. Experiment 6, which added the SE to Experiment 5, achieved a mAP@0.5 score of 94.31%, a 4.03% improvement, with a detection speed of 102.04 fps, an increase of 17.30 fps over the baseline.

Visualizations of the results of the combination experiments are shown in Figure 12c,d, where the changes in detection accuracy (mAP@0.5 and mAP@0.95) are clearly and incrementally demonstrated. These results verify the effectiveness of the Light-YOLO model and highlight its improvements over the baseline.

Table 2. Effects of different experimental schemes on model performance.

	PSEA-FPN	Swift-SPP	SE	mAP@0.5/%	mAP@0.5:0.95/%	Params (M)	FPS/(f.sˆ-1)
0	×	×	×	90.28	42.07	5.803	84.74
1	√	×	×	92.72	43.76	4.226	102.31
2	×	√	×	90.91	41.83	3.929	97.82
3	×	×	√	92.22	42.41	3.962	93.47
4	×	√	√	92.69	42.17	3.954	95.25
5	√	√	×	93.35	43.56	3.116	104.58
6	√	√	√	94.31	44.43	3.228	102.04

4.5. Comparison with Other Classic Algorithms

Most extant studies on vehicle detection do not use lightweight models; hence, comparisons were made with the most prominent lightweight networks (i.e., YOLOX_s, YOLOv7-tiny, EfficientDet-D1, and varying combinations of MobileNetv3-YOLOv5, GhostNet-YOLOV5, and MobileNetv2-SSD). The results are listed in Table 3.

EfficientDet comprises a series of scalable and efficient object detectors, ranging from EfficientDet-D1 to EfficientDet-D7. The models gradually increase in accuracy as their real-time performance decreases. The fastest, EfficientDet-D1, was selected for comparison with Light-YOLO. From Table 2, it is evident that Light-YOLO has a significant advantage in terms of detection accuracy, with a mAP@0.5 score of 4.99% and a mAP@0.5:0.95 of 4.55%. Although our model complexity was slightly higher, Light-YOLO achieved significantly higher detection speeds.

Although YOLOv5–YOLOv7, v7 being the most advanced, are widely used for object detection, they are not considered lightweight algorithms. However, YOLOX_s, which is based on YOLOv5s, is. The results show that Light-YOLO outperformed YOLOX_s in terms of detection speed and accuracy. YOLOv7 has a simplified version (i.e., YOLOv7-tiny), which has a faster detection speed than Light-YOLO; however, it lagged behind mAP@0.5 and mAP@0.5:0.95 scores by 8.29 and 5.91%, respectively.

The results of the variant combinations show that GhostNet-YOLOV5 had lower *mAP* and overall accuracy than Light-YOLO, in addition to higher model complexity. MobileNetv3-YOLOv5 and MobileNetv2-SSD had accuracy levels similar to Light-YOLO, but they lagged behind in recognition speed, by 8.49 fps.

The visualization results mAP@0.5, mAP@0.5:0.95, *Params*, and fps are presented in Figure 13. Figure 14 shows a comparison between Light-YOLO and the other lightweight algorithms in terms of recognition speed and accuracy. Overall, Light-YOLO demonstrated a significant competitive advantage in all comprehensive metrics.

Table 3. Performance comparison between Light-YOLO and other lightweight algorithms.

Methods	Size	mAP@0.5/%	mAP@0.5:0.95/%	Params (M)	FPS/(f.sˆ-1)
Light-YOLO	640	94.31	44.43	3.23	102.04
EfficientDet-D1	640	89.32	39.88	6.63	76.75
improvement	-	+4.99	+4.55	−51.28%	+32.95%
YOLOX_s	640	86.02	38.52	8.96	60.35
improvement	-	+8.29	+5.91	−63.95%	+69.08%
YOLOv7-tiny	640	84.17	36.74	6.02	136.67
improvement	-	+10.14	+7.69	−46.35%	−25.34%
MobileNetV3-YOLOv5	640	91.87	42.15	5.03	85.47
improvement	-	+2.44	+2.28	−35.78%	+19.39%
GhostNet-YOLOV5	640	90.31	41.17	2.04	98.29
improvement	-	+4	+3.26	+58.33%	+3.82%
MobileNetV2-SSD	640	93.43	43.47	3.63	93.55
improvement	-	+0.88	+0.96	−11.02%	−9.08%

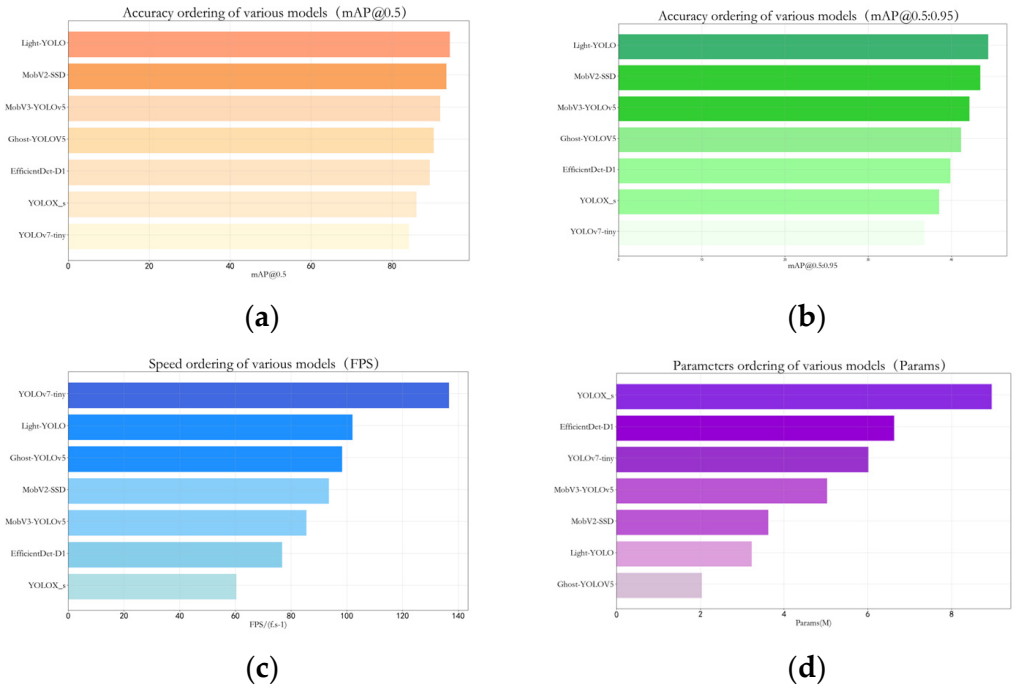


Figure 13. Metric visualizations of lightweight models: (**a**) mAP@0.5, (**b**) mAP@0.5:0.95, (**c**) Params, (**d**) FPS.

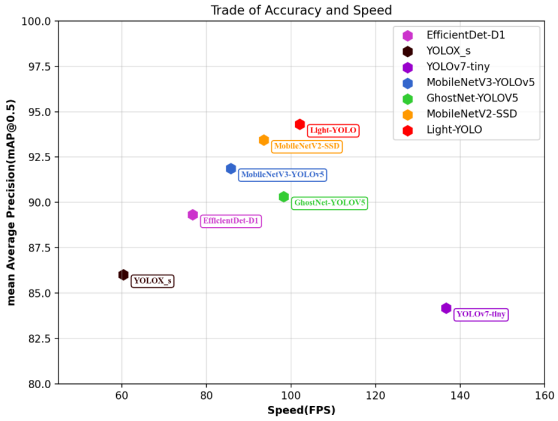


Figure 14. Trade-offs between accuracy and speed.

4.6. Comparison of Effects

To further validate the applicability of Light-YOLO in nighttime scenarios, a set of images was selected from our dataset for before-and-after comparisons of performance. As shown in Figure 15, with Nighttime Image I, three vehicles were detected. Both the Efficienctnetv2-YOLOv5 and Light-YOLO models recognized all vehicle objects, but Light-YOLO showed a higher confidence level. In Nighttime Image II, Efficienctnetv2-YOLOv5 produced false positives, due to lighting and shadow interference. In contrast, Light-YOLO demonstrated good robustness and higher confidence. For Nighttime Images III and IV, it becomes apparent that, in densely packed nighttime scenes, Efficienctnetv2-YOLOv5 struggles with accurate vehicle localization, compared with Light-YOLO, which performs better.

The detection results shown in Figure 15 illustrate the effectiveness of the improvements made in this study.

Figure 15. Comparison of effects before and after algorithm improvement: (**a**) original images; (**b**) detection effect of the algorithm before improvement (Efficienctnetv2-YOLOv5); (**c**) detection effect of the algorithm after improvement (Light-YOLO).

5. Conclusions

This paper has proposed the Light-YOLO lightweight target-detection algorithm, designed for nocturnal vehicle-detection and mobility. This model was built on the EfficientNetv2-YOLOv5s baseline and incorporates a multitude of enhancements, including PSEA-FPN, Swift-SPP, and focal loss. The empirical findings demonstrate that Light-YOLO yields substantial performance improvements in nocturnal vehicle-detection tasks over the benchmark. The ultimate *mAP* score increased from 90.28 to 94.31%, concomitantly reducing the parameter count by 44.37% and augmenting the recognition speed by 20.42%. Additionally, internal validation experiments show that the incorporation of the four improvements within PSEA-FPN boosts *mAP* by 1.84%, while increasing the frame rate by 34.26%, proving the efficacy of the proposed internal structure. Comparisons between different lightweight networks illustrate that the Light-YOLO outperforms YOLOv7-tiny in *mAP* by 10.14% while using 46.35% fewer parameters, and it outperforms GhostNet-YOLOV5, with a 4% increase in *mAP* and a 3.82% improvement in frame rate. These outcomes underscore the capacity of Light-YOLO to considerably increase the precision of nocturnal vehicle-detection while preserving real-time operationality, thus promoting its pragmatic application potential.

In future investigations, we will evaluate the efficacy of this algorithm for detecting smaller targets during nocturnal conditions. Furthermore, the lightweight attributes of the model must be enhanced. Subsequently, we will embark on investigations of other lightweight models, seeking to enhance their suitability for real-time vehicle-target-detection in nocturnal settings.

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

Funding: This research was funded by Key Research and Development Program of Hubei Province (No. 2023BAB052) and the Hubei Province Technological Innovation Major Project (No. 2019AAA025).

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

Conflicts of Interest: Author Bo Liu was employed by the Wuhan Zhongyuan Electronics Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The Wuhan Zhongyuan Electronics Group Co., Ltd. had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Yin, T.; Chen, W.; Du, L.; Xiao, Z.; Tong, B.; Su, Z. Improved Crossing Pedestrian Detection Model for Intelligent Vehicles in Complex Traffic Scenes. In Proceedings of the 2023 7th International Conference on Transportation Information and Safety (ICTIS), Xi'an, China, 4–6 August 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1625–1630.
2. Buch, N.; Velastin, S.A.; Orwell, J. A review of computer vision techniques for the analysis of urban traffic. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 920–939. [CrossRef]
3. Chen, C.; Liu, B.; Wan, S.; Qiao, P.; Pei, Q. An edge traffic flow detection scheme based on deep learning in an intelligent transportation system. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1840–1852. [CrossRef]
4. Lin, Y.; Deng, L.; Chen, Z.; Wu, X.; Zhang, J.; Yang, B. A real-time ATC safety monitoring framework using a deep learning approach. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 4572–4581. [CrossRef]
5. Kim, E.; Ryu, H.; Oh, H.; Kang, N. Safety monitoring system of personal mobility driving using deep learning. *J. Comput. Des. Eng.* **2022**, *9*, 1397–1409. [CrossRef]
6. Muntarina, K.; Mostafiz, R.; Khanom, F.; Shorif, S.B.; Uddin, M.S. MultiResEdge: A Deep Learning-Based Edge Detection Approach. *Intell. Syst. Appl.* **2023**, *20*, 200274. [CrossRef]

7. Wang, X.; Wang, S.; Guo, Y.; Hu, K.; Wang, W. Coal gangue image segmentation method based on edge detection theory of star algorithm. *Int. J. Coal Prep. Util.* **2023**, *43*, 119–134. [CrossRef]
8. Chen, G.; Zhang, G.; Yang, Z.; Liu, W. Multi-scale patch-GAN with edge detection for image inpainting. *Appl. Intell.* **2023**, *53*, 3917–3932. [CrossRef]
9. Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; et al. Planning-oriented autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 17853–17862.
10. Zheng, X.; Lu, C.; Zhu, P.; Yang, G. Visual Multitask Real-Time Model in an Automatic Driving Scene. *Electronics* **2023**, *12*, 2097. [CrossRef]
11. Miraliev, S.; Abdigapporov, S.; Kakani, V.; Kim, H. Real-Time Memory Efficient Multitask Learning Model for Autonomous Driving. *IEEE Trans. Intell. Veh.* **2023**, 1–12. [CrossRef]
12. Li, X.; Zhou, Z.; Wu, J.; Xiong, Y. Human posture detection method based on wearable devices. *J. Healthc. Eng.* **2021**, *2021*, 8879061. [CrossRef]
13. Ogundokun, R.O.; Maskeliūnas, R.; Damaševičius, R. Human posture detection using image augmentation and hyperparameter-optimized transfer learning algorithms. *Appl. Sci.* **2022**, *12*, 10156. [CrossRef]
14. Ogundokun, R.O.; Maskeliūnas, R.; Misra, S.; Damasevicius, R. A novel deep transfer learning approach based on depth-wise separable CNN for human posture detection. *Information* **2022**, *13*, 520. [CrossRef]
15. Joachims, T. Making Large-Scale SVM Learning Practical. *Tech. Rep.* **1998**, *8*, 499–526. [CrossRef]
16. Park, M.J.; Ko, B.C. Two-step real-time night-time fire detection in an urban environment using Static ELASTIC-YOLOv3 and Temporal Fire-Tube. *Sensors* **2020**, *20*, 2202. [CrossRef] [PubMed]
17. Chen, C.; Liu, M.Y.; Tuzel, O.; Xiao, J. R-CNN for small object detection. In Proceedings of the Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; Revised Selected Papers, Part V 13. Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 214–230.
18. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
20. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
21. Shenoda, M. Real-time Object Detection: YOLOv1 Re-Implementation in PyTorch. *arXiv* **2023**, arXiv:2305.17786.
22. Sang, J.; Wu, Z.; Guo, P.; Hu, H.; Xiang, H.; Zhang, Q.; Cai, B. An improved YOLOv2 for vehicle detection. *Sensors* **2018**, *18*, 4272. [CrossRef]
23. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
24. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
25. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
26. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
27. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
28. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones* **2023**, *7*, 304. [CrossRef]
29. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
30. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
31. 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.
32. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
33. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
34. Tan, M.; Le, Q. Efficientnetv2: Smaller models and faster training. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 10096–10106.
35. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.

36. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
37. Tong, B.; Chen, W.; Li, C.; Du, L.; Xiao, Z.; Zhang, D. An Improved Approach for Real-Time Taillight Intention Detection by Intelligent Vehicles. *Machines* **2022**, *10*, 626. [CrossRef]
38. Ahmed, M.; Seraj, R.; Islam, S.M.S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics* **2020**, *9*, 1295. [CrossRef]
39. Lin, C.T.; Huang, S.W.; Wu, Y.Y.; Lai, S.-H. GAN-based day-to-night image style transfer for nighttime vehicle detection. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 951–963. [CrossRef]
40. Ye, J.; Fu, C.; Zheng, G.; Paudel, D.P.; Chen, G. Unsupervised domain adaptation for nighttime aerial tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8896–8905.
41. 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.
42. 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.
43. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7036–7045.
44. Chen, J.; Mai, H.S.; Luo, L.; Chen, X.; Wu, K. Effective feature fusion network in BIFPN for small object detection. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 9–22 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 699–703.
45. Li, X.; Men, F.; Lv, S.; Jiang, X.; Pan, M.; Ma, Q.; Yu, H. Vehicle detection in very-high-resolution remote sensing images based on an anchor-free detection model with a more precise foveal area. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 549. [CrossRef]
46. Gu, Z.; Zhu, K.; You, S. YOLO-SSFS: A Method Combining SPD-Conv/STDL/IM-FPN/SIoU for Outdoor Small Target Vehicle Detection. *Electronics* **2023**, *12*, 3744. [CrossRef]
47. Chen, Z.; Guo, H.; Yang, J.; Jiao, H.; Feng, Z.; Chen, L.; Gao, T. Fast vehicle detection algorithm in traffic scene based on improved SSD. *Measurement* **2022**, *201*, 111655. [CrossRef]
48. Chen, L.; Ding, Q.; Zou, Q.; Chen, Z.; Li, L. DenseLightNet: A light-weight vehicle detection network for autonomous driving. *IEEE Trans. Ind. Electron.* **2020**, *67*, 10600–10609. [CrossRef]
49. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
50. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
51. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
52. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [CrossRef]
53. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
54. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
55. Liu, H.; Liu, F.; Fan, X.; Huang, D. Polarized self-attention: Towards high-quality pixel-wise regression. *arXiv* **2021**, arXiv:2107.00782.