drones

MDPI

Article

Fine-Grained Feature Perception for Unmanned Aerial Vehicle Target Detection Algorithm

Shi Liu [1], Meng Zhu [2], Rui Tao [1,3] and Honge Ren [1,4,*]

[1] College of Computer and Control Engineering, Northeast Forestry University, Harbin 150040, China; shitou@nefu.edu.cn (S.L.); trlx20@nefu.edu.cn (R.T.)
[2] College of Information Engineering, Harbin University, Harbin 150086, China; zhumeng913@hrbu.edu.cn
[3] College of Artificial Intelligence and Big Data, Hulunbuir University, Hulunbuir 021008, China
[4] Heilongjiang Forestry Intelligent Equipment Engineering Research Center, Harbin 150040, China
* Correspondence: rhe@nefu.edu.cn; Tel.: +86-137-6680-5518

Abstract: Unmanned aerial vehicle (UAV) aerial images often present challenges such as small target sizes, high target density, varied shooting angles, and dynamic poses. Existing target detection algorithms exhibit a noticeable performance decline when confronted with UAV aerial images compared to general scenes. This paper proposes an outstanding small target detection algorithm for UAVs, named Fine-Grained Feature Perception YOLOv8s-P2 (FGFP-YOLOv8s-P2), based on YOLOv8s-P2 architecture. We specialize in improving inspection accuracy while meeting real-time inspection requirements. First, we enhance the targets' pixel information by utilizing slice-assisted training and inference techniques, thereby reducing missed detections. Then, we propose a feature extraction module with deformable convolutions. Decoupling the learning process of offset and modulation scalar enables better adaptation to variations in the size and shape of diverse targets. In addition, we introduce a large kernel spatial pyramid pooling module. By cascading convolutions, we leverage the advantages of large kernels to flexibly adjust the model's attention to various regions of high-level feature maps, better adapting to complex visual scenes and circumventing the cost drawbacks associated with large kernels. To match the excellent real-time detection performance of the baseline model, we propose an improved Random FasterNet Block. This block introduces randomness during convolution and captures spatial features of non-linear transformation channels, enriching feature representations and enhancing model efficiency. Extensive experiments and comprehensive evaluations on the VisDrone2019 and DOTA-v1.0 datasets demonstrate the effectiveness of FGFP-YOLOv8s-P2. This achievement provides robust technical support for efficient small target detection by UAVs in complex scenarios.

Keywords: unmanned aerial vehicle; small object detection; Fine-Grained Feature; YOLOv8

Citation: Liu, S.; Zhu, M.; Tao, R.; Ren, H. Fine-Grained Feature Perception for Unmanned Aerial Vehicle Target Detection Algorithm. *Drones* **2024**, *8*, 181. https://doi.org/10.3390/drones8050181

Academic Editor: Diego González-Aguilera

Received: 7 March 2024
Revised: 26 April 2024
Accepted: 27 April 2024
Published: 3 May 2024

1. Introduction

The rapid development and exceptional flexibility of unmanned aerial vehicle (UAV) technology have positioned it as a critical driver across various industries. UAV target detection, a crucial application in UAV technology, finds widespread use in scenarios such as traffic surveillance [1], marine environment monitoring [2], and fire detection [3], providing efficient means for real-time data collection and detection in these domains.

The definition of small targets in the COCO [4] dataset is based on an absolute scale, where targets with a resolution smaller than 32 pixels × 32 pixels are considered small. Another definition is based on a relative scale, where targets occupying less than 0.01% of the original image are classified as small targets. This study categorizes targets defined by both absolute and relative scales as small targets. Due to the constraints of the unmanned aerial vehicle (UAV) target detection task, most targets captured from long distances are small, with complex backgrounds often leading to scenes with dense target distributions.

Current state-of-the-art general target detection algorithms [5–7] struggle to perform well in such challenging datasets. The main reason is the targets' limited pixel information, making them susceptible to background interference, resulting in models failing to extract high-quality feature maps containing subtle features such as small targets' edges, textures, and colors. While YOLOv8s achieves a mAP_50_95 score of 44.9% on the COCO validation set, the experimental results of this study on the VisDrone2019 validation set yield only a 25.1% mAP_50_95 score. Some UAV aerial missions also require real-time detection, which requires balancing accuracy and latency in the model. Therefore, further refinement and optimization of target detection algorithms are still required in the UAV aerial photography domain to enhance their applicability and performance.

In recent years, innovative solutions have emerged, focusing on multi-scale feature fusion, data augmentation, and attention mechanisms, intending to enhance the performance and robustness of small target detection in unmanned aerial vehicles (UAVs). The Feature Pyramid Network (FPN) [8], leveraging the inherent multi-scale pyramid hierarchical structure of deep convolutional networks, constructs high-level semantic feature maps across all scales. This enhances the adaptability of target detection algorithms to scale variations, emphasizing the importance of multi-scale fusion in target detection models. Previous studies [9–12] demonstrate that well-designed fusion strategies effectively enhance the perceptual capabilities of models for small-sized targets, improving the accuracy and robustness of small target detection. However, they have not effectively balanced detection accuracy and speed.

Data augmentation techniques, incorporating geometric transformations, optical transformations, and deep learning methods [13–16], provide algorithms with richer training samples, facilitating more comprehensive learning of features representing small targets. Addressing the scarcity of small targets in images, reference [17] improves model performance by oversampling and duplicating small targets, addressing the limitations of conventional target detection algorithms. Attention mechanisms, extensively studied in prior literature [18–20], assist models in focusing more finely on target regions. References [21,22] introduces attention mechanisms tailored for small targets, demonstrating powerful inference capabilities and enhancing the accuracy of small target detection. A dual-path attention module is designed in [23], capturing spatial features with spatial attention and suppressing irrelevant information with channel attention, effectively capturing critical information of small targets while mitigating the impact of background noise. Self-attention mechanisms, capable of perceiving global information in images rather than being limited to local regions, have propelled Transformer models to significant achievements in small target detection [24,25].

In this paper, we propose a UAV small target detection algorithm named FGFA-YOLOv8s-P2, focusing on improving detection accuracy while maintaining detection speed. Due to the insufficient pixel information for the model to learn effective patterns in UAV small target detection, we comprehensively consider data processing, feature map extraction, and detection speed. As general object detection algorithms demonstrate excellent detection performance for common targets, it is reasonable to expect similar effectiveness when treating small targets as common targets. Hence, we introduce slicing-aided techniques [26] for overlapping image slicing. Aerial images exhibit complex and diverse target distributions and shapes, posing challenges to the adaptability of the feature extraction module in general object detection algorithms. In order to address this issue, previous research [27] introduced deformable convolution, which was further improved by [28,29] with satisfactory results. However, we find that [29], through depth-wise separable convolutions, inaccurately couples the learning of offsets and modulation scalar, hindering the model's ability to grasp details of small targets. Thus, we propose an improved deformable convolution by decoupling the learning of offsets and modulation scalar to enhance detection accuracy. Jierun Chen et al. [30] proposed partial convolutions to reduce redundant computations and memory access. However, experimental findings indicate that this method may reduce the accuracy of small target detection by 1.6% in mAP_50_95 on the

DOTA validation set. To meet the demand for real-time detection, we draw inspiration from partial convolutions and design stochastic partial convolution modules, achieving accelerated detection speed without compromising accuracy.

YOLOv8, the latest iteration in the YOLO series, is acclaimed for its lightweight network architecture while maintaining high precision. However, in the domain of UAV target detection, its performance is significantly compromised due to excessive downsampling. To better harness the detailed information within low-level, high-resolution feature maps, this study opts to enhance the YOLOv8s-P2 as the benchmark model. Its optimization encompasses data processing, network structure, and model lightweighting. Validation is conducted using the VisDrone2019 and DOTAv-1.0 datasets, and the enhancement effects of different improvements are systematically evaluated through ablation experiments. The FGFA-YOLOv8s-P2 achieves a mAP_50 score of 48.3% on the VisDrone2019 validation set, with 136.4 FPS on an RTX 2080Ti GPU, outperforming other advanced small target detection algorithms in both speed and accuracy. The main contributions of this paper are as follows:

(1) Introduction of slicing assistance for the first time during both training and inference phases of YOLOv8s-P2, enhancing target pixel information.
(2) Improvements in the backbone network for high-quality feature map extraction:
    a: Design of the Large Kernel Spatial Pyramid Pooling Fast module, enabling high-level feature maps to consider long-range dependencies, local dependencies, and channel adaptability, enhancing the model's understanding of complex scenes.
    b: Design the feature extraction module with deformable convolutions, decoupling the learning processes of offset and modulation scalar, enhancing target localization and adaptability to targets with different scales and shapes.
(3) A Random FasterNet Block has been designed and applied to the neck network. It introduces randomness into convolution operations and incorporates non-linear transformations by adding depth-wise convolutions followed by point-wise convolutions, enhancing the model's robustness. This approach accelerates detection speed while maintaining the original feature representation of convolution operations.

2. Related Work

2.1. Multi-Scale Feature Fusion

Improving feature fusion aims to provide more semantic information to lower-level feature maps while preventing the original responses of small targets from being overwhelmed by deeper signals. This is a nuanced consideration. Inspired by the simple yet effective design of the Feature Pyramid Network (FPN), PANet [31] enriches feature hierarchies through bidirectional pathways, enhancing deeper features with precise localization signals from the lower levels. Tan et al. [32] propose an efficient Weighted Bi-directional Feature Pyramid Network (BIFPN), introducing learnable weights for cross-scale connections to achieve advanced feature fusion. SPPNet [33] guides relationships between adjacent layers to direct appropriate features from deep to shallow layers, avoiding inconsistencies in gradient calculations between different layers. If both adjacent layers detect an object, the deep layer can provide more semantic features, optimizing simultaneously with the next layer. Shahin et al. [34] introduce the SVA module, fusing spectral saliency feature maps and spatial feature maps using an additive fusion function. This effectively utilizes enhanced spectral saliency residual layers for improved detection performance, albeit at a slower detection speed. Chai et al. [35] designed the IUDet-Fusion module, effectively addressing information loss and scale imbalances caused by the ASPP module. By incorporating global max-pooling, global average-pooling, and dilated convolutions with dilation rates of 1, 2, and 3, respectively, the feature information is enriched without sacrificing information integrity.

2.2. Data Augmentation

Appropriate data augmentation is crucial in better addressing various changes and noise in real-world scenarios. To tackle the issue of class imbalance in the training set, reference [36] extracts instances of minority classes and adds them to new images. Bosquet et al. [16] propose a novel data augmentation pipeline for small target detection. They utilize a downsampling GAN to generate large targets into more realistic small targets, placing them into existing image backgrounds. This approach addresses the challenge of insufficient small target samples for training. Refs. [26,37] enhances detection performance by reducing the representation difference between small and large targets. According to the standards of the COCO dataset, small targets account for up to 60% of the VisDrone dataset [38]. Limitations in pixel information make targets susceptible to background noise, posing significant challenges for detection tasks. This paper addresses this issue by increasing the relative size of the targets, thereby providing the model with more pixel information.

2.3. Attention Mechanism

The introduction of attention mechanisms aims to focus the model on crucial regions of the image, thereby enhancing detection performance. Spatial attention mechanisms deal with spatial relations by emphasizing information input from different locations, reference [18,39–41] representing different spatial attention mechanisms. The multi-head self-attention mechanism in [19] flexibly perceives information at different scales in the image, enabling the network to learn more general and robust patterns than conventional convolutions. However, it comes with a higher computational cost. Deformable convolution [27] enables the network to adaptively adjust receptive fields based on target shapes, mitigating the gap between multi-head self-attention mechanisms and conventional convolutions. Different feature channels represent distinct mapping information, and adaptive adjustment of the weights for each channel through attention facilitates the exploration of channel information. Following the introduction of channel attention [20], improvements have been made in the squeeze part [42], the excitation part [43], and both squeeze and excitation [44,45].

Building upon this previous work, researchers have proposed new methods for attention mechanisms in small target detection. Liu et al. [46] designed the context attention module (COAM) and attention enhancement module (AEM). COAM uses the technique of hypercolumn with steps in combination with the atrous spatial pyramid pooling (ASPP) module to generate an attention heat map with contextual information. AEM employs a single fully connected (FC) layer as the hidden layer to generate a channel attention matrix. This matrix undergoes Hadamard product computation with the Attention Heatmap Matrix, enhancing feature representation and mitigating the impact of background noise. Yang et al. [47] propose the Multi-Attention Residual Network, which integrates spatial attention, channel attention, and self-attention mechanisms. This comprehensive approach captures spatial structures, channel correlations, and global-local relationships, thereby improving the model's perception of targets.

3. Improved Unmanned Aerial Vehicle Target Detection Algorithm

This paper enhances target detection performance in UAV aerial images by improving the perception of detailed features in the targets. The optimization of data processing is discussed in detail in Section 3.1, while Section 3.2 focuses on enhancements to the feature extraction module of the backbone network. Simultaneously, the lightweighting of the neck network is achieved, as elaborated in Section 3.3. By integrating these key improvements, significant advancements are achieved in enhancing detection accuracy and balancing the utilization of computational resources, making the improved model more practical and efficient. Figure 1 illustrates the network architecture of the improved model, with the softened-edged dark boxes indicating the introduced enhancement modules in this paper.

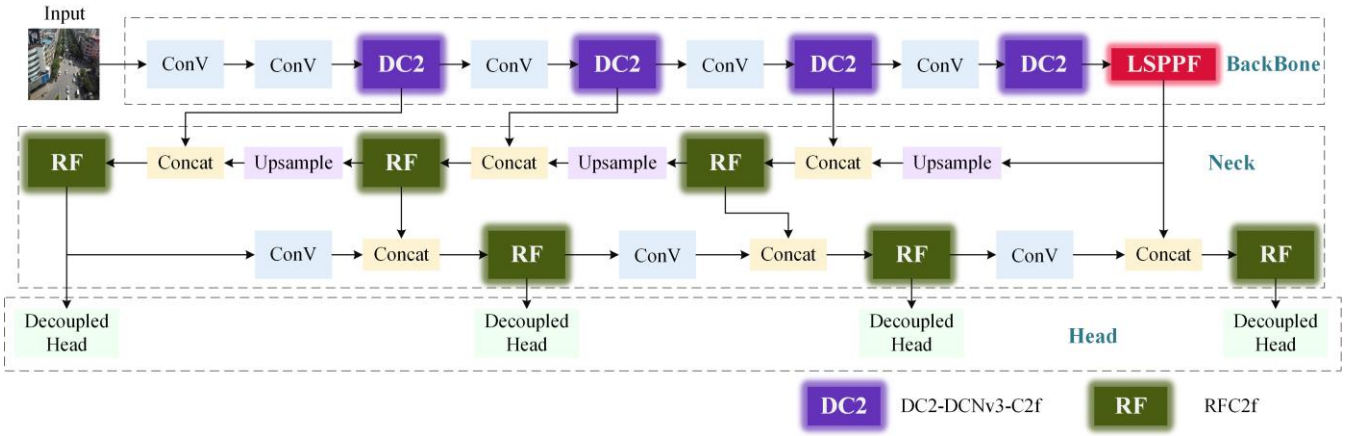

Figure 1. Network Architecture of FGFP-YOLOv8s-P2.

3.1. Data Processing

This paper draws inspiration from the idea of slicing assistance [26] to enhance the relative size of small targets. Both before training and during inference, slicing operations are performed. Before training, sliding window slicing is applied to the image with sequential overlapping based on the overlap ratio and slice size. Slices maintain aspect ratio adjustments and are input into the model with the original image for training. The same slicing method is used during inference, maintaining aspect ratio adjustments, and slices are combined with the original image for inference. The prediction results of each slice are merged with the prediction results of the entire image, and non-maximum suppression is applied for filtering. After slicing, the feature extractor can capture more feature information relative to the original image. Figure 2 provides an example of slicing operations.

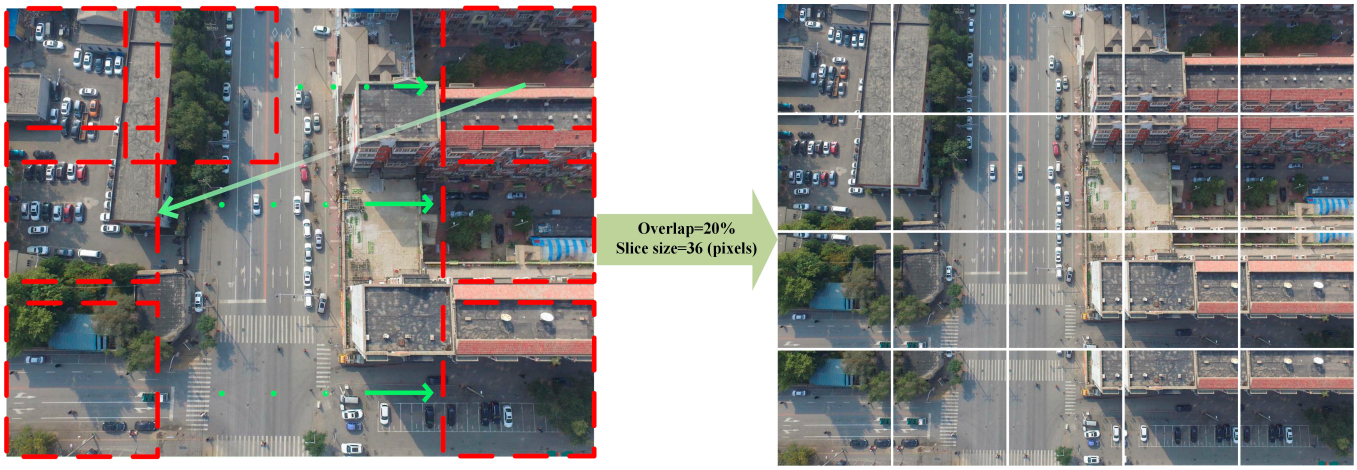

Figure 2. Example of slicing operation.

The overlap ratio and slice size are set as hyperparameters. To avoid compromising the detection performance of large targets and the detection speed, one should ensure that the slice size is not too small and that the overlap is not too considerable. Through parameter tuning on the VisDrone2019 validation set, it is found that setting the slice size to 360 and the overlap to 20% (the same for vertical and horizontal directions) has the most beneficial effect on the model. Specifically, slicing is shifted to the left or upward for the remaining images that do not meet the slice size and overlap requirements.

3.2. Backbone

3.2.1. DC2-DCNv3-C2f

YOLOv8 utilizes traditional convolutional and C2f modules for feature extraction—however, these modules' fixed receptive field size results in poor feature extraction quality for small targets. An excessively large receptive field may lead to losing details, while one too small may fail to cover the entire target. Additionally, conventional convolutions are insensitive to changes in the target's position and incapable of capturing the target's subtle boundary features. Motivated by deformable convolutions, this paper proposes the DC2-DCNv3-C2f module. This adaptive and refined information extraction approach allows for better focus on target regions, reducing background interference and improving detection accuracy.

Deformable Convnets v2 (DCNv2) is not well suited for models trained from scratch [29]. Deformable Convolution v3 (DCNv3) [29] introduces three improvements: it borrows the idea of depth-wise separable convolution, uses modulation scalar as the depth part, and employs point-wise convolution with shared projection weights between sampled points. This effectively reduces the computational load of the model. With the introduction of a multi-group mechanism, each group has independent offsets and modulation scalars, thus giving the groups different spatial aggregation patterns and enriching the diversity of feature expression. The normalization of the modulation scalar is adjusted by replacing the original sigmoid element-wise normalization with softmax normalization, leading to a more stable convergence during the training process. The formulation is expressed as follows:

$$\mathrm{y}(\mathrm{p}_0) = \sum_{\mathrm{g}=1}^{\mathrm{G}} \sum_{\mathrm{k}=1}^{\mathrm{K}} \mathrm{w}_\mathrm{g} \cdot \mathrm{x}_\mathrm{g}(\mathrm{p}_0 + \mathrm{p}_\mathrm{k} + \Delta \mathrm{p}_{\mathrm{gk}}) \cdot \mathrm{m}_{\mathrm{gk}} \tag{1}$$

where G represents the total number of groups, and K denotes the sampling count of regular grids. The modulation scalar, denoted as m_{gk}, for the kth sampling point in the gth group, is obtained through softmax normalization along the K dimension. Here, w_g represents the weights for intra-group channel-wise fully connected connections, while Δp_{gk} represents the offset of the sampled point p_k corresponding to this group. $x_g(p_0 + p_k + \Delta p_{gk})$ represents the pixel value after offsetting the k-th sampling point in the regular grid within the g-th group of feature maps. $y(p_0)$ represents a pixel value in the feature map after deformable convolution.

$$\mathrm{Offset} = \mathrm{FC}_{\mathrm{Offset}}(\mathrm{DC}_1(\mathrm{X})) \tag{2}$$

$$\mathrm{Modulation} = \mathrm{FC}_{\mathrm{modulation}}(\mathrm{DC}_1(\mathrm{X})) \tag{3}$$

$$\mathrm{Offset} = \mathrm{FC}_{\mathrm{Offset}}(\mathrm{DC}_{\mathrm{Offset}}(\mathrm{X})) \tag{4}$$

$$\mathrm{Modulation} = \mathrm{FC}_{\mathrm{modulation}}(\mathrm{DC}_{\mathrm{modulation}}(\mathrm{X})) \tag{5}$$

Formulas (2) and (3) state that DCNv3 utilizes the same depth convolution, DC_1, to generate shared feature maps for offset and modulation scalar. FC_{offset} and $FC_{modulation}$ are channel-wise fully connected operations that generate offset and modulation scalar, respectively. The model retains specific positioning information while generating offsets and considers adopting this information during modulation scalar generation. This sharing approach may lead to the model's inability to fully exploit the independence between the two tasks, making it challenging to decouple them effectively. The improved DCNv3 in this paper employs a more flexible approach, as illustrated in Figure 3. It introduces an additional depth convolution operation (enclosed in the blue dashed box) alongside the previous one. These two separate depth convolutions generate independent feature maps of x1 and x2. Subsequently, two independent point convolution operations are employed on these feature maps to generate offsets and modulation scalars separately. The two depth convolution operations in Formulas (4) and (5) are DC_{offset} and $DC_{modulation}$, respectively. The enhanced module is referred to as the DC2-DCNv3 module, which effectively addresses

the issue of information loss caused by shared feature maps, contributing to the further optimization of target positioning and shape information.

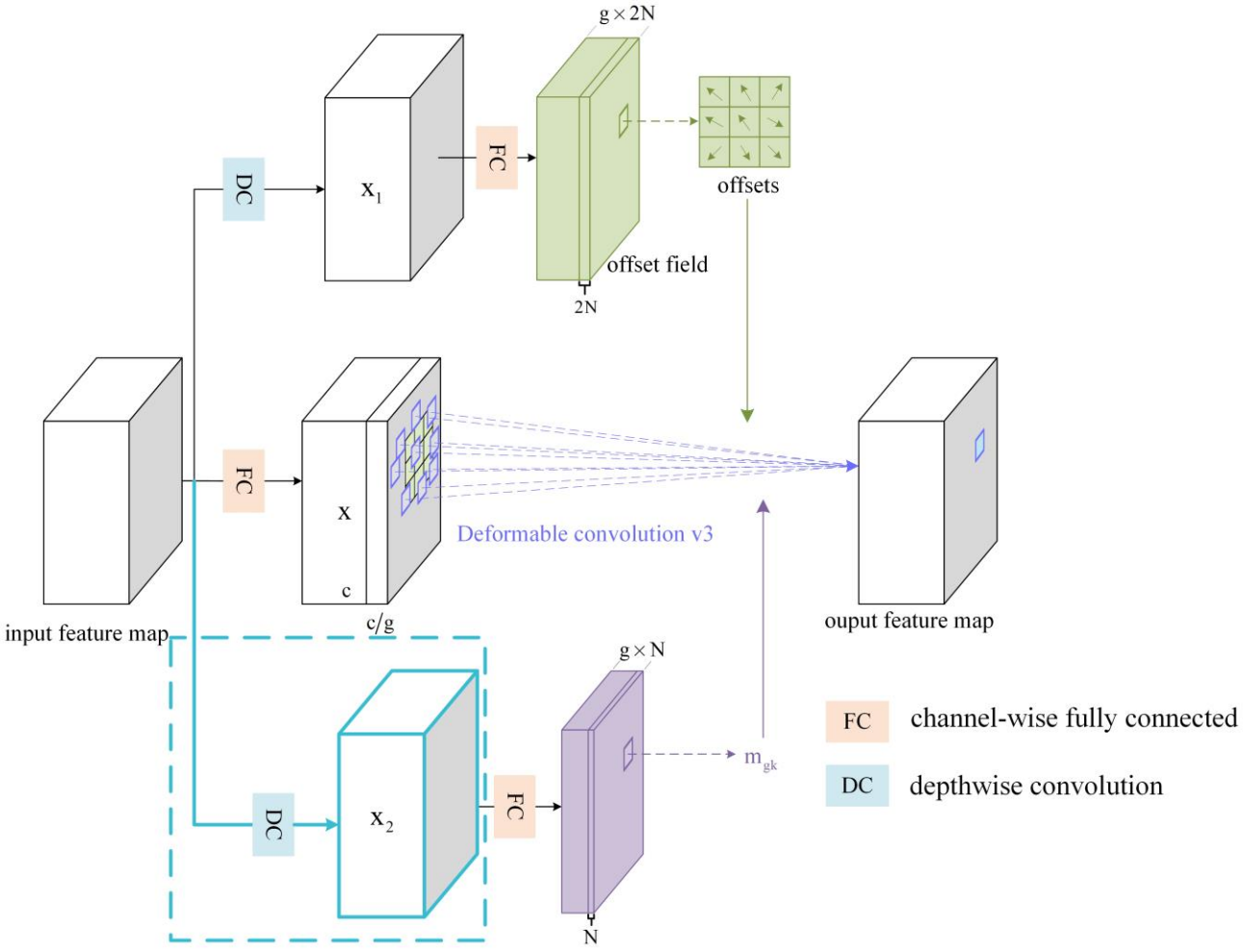

Figure 3. DC2-DCNv3 network architecture.

We integrate DC2-DCNv3 with the C2f module of the backbone network. The structure of the improved C2f module is illustrated in Figure 4b, referred to as the DC2-DCNv3-C2f module. The DC2-DCNv3-C2f module accepts images processed by slicing-aided techniques. It can extract adaptive, fine-grained information on feature maps with a larger relative area of the target and more detailed information.

3.2.2. Large Kernel Spatial Pyramid Pooling Fast Module

In object detection, attention mechanisms aid the model in selectively focusing on key information relevant to the target, such as edges, semantics, and contextual information surrounding the target. However, models struggle to attend to these critical details in small targets in UAV aerial images with complex backgrounds. Therefore, this paper proposes the Large Kernel Spatial Pyramid Pooling Fast module (LSPPF).

While the introduction of Large Kernel Attention (LKA) successfully addressed computational and memory usage issues associated with large kernel convolutions, detection performance tends to saturate with increasing kernel size. Further improvements are made by the Separable Large Kernel Attention (LSKA) [48], decomposing the large kernel into a depth-wise convolution with a small receptive field and a dilated convolution with a large receptive field. This approach more effectively captures local and long-range dependencies. LSKA cleverly divides these convolutional kernels into 1×K and K×1, reducing the growth of parameters and Floating Point Operations (FLOPs). This design allows an increase in ker-

nel size to capture global information without imposing a significant computational burden. The parameters and FLOPs for LSK and LSKA are defined as follows in the equation:

$$\text{Param}_{\text{LSK}} = (2d-1)^2 \times C + \lfloor k/d \rfloor^2 \times C + C \times C \tag{6}$$

$$\text{Param}_{\text{LSKA}} = (2d-1) \times C \times 2 + \lfloor k/d \rfloor \times C \times 2 + C \times C \tag{7}$$

$$\text{FLOPs}_{\text{LSK}} = \left((2d-1)^2 \times C + \lfloor k/d \rfloor^2 \times C + C \times C\right) \times H \times W \tag{8}$$

$$\text{FLOPs}_{\text{LSKA}} = ((2d-1) \times C \times 2 + \lfloor k/d \rfloor \times C \times 2 + C \times C) \times H \times W \tag{9}$$

K represents the size of the convolutional kernel; d denotes the dilation rate; and H, W, and C represent the input feature map's height, width, and number of channels, respectively. The design of this depth-wise convolution leads to savings in both the number of parameters and FLOPs by "$(2d-1)/2$," and at the location of dilated depth-wise convolution, the reduction in the number of parameters and FLOPs by "$\lfloor k/d \rfloor/2$".

The structure of LSKA, as shown in Figure 5a, first involves input features passing through cascaded vertical and horizontal 1D depth-wise convolutions to capture local dependencies. Next, they undergo cascaded horizontal and vertical 1D depth-wise dilated convolutions to capture long-range dependencies. Finally, point-wise convolutions are applied to capture the importance of each channel.

Figure 5b illustrates the LSPPF module, which embeds the LSKA structure from (a) into LSPPF. The input to LSKA consists of concatenated feature maps of three max-pooling layers and a convolutional layer. After LSKA's point-wise convolutions, the feature map is element-wise multiplied with the input at corresponding positions, yielding a feature map with mixed attention.

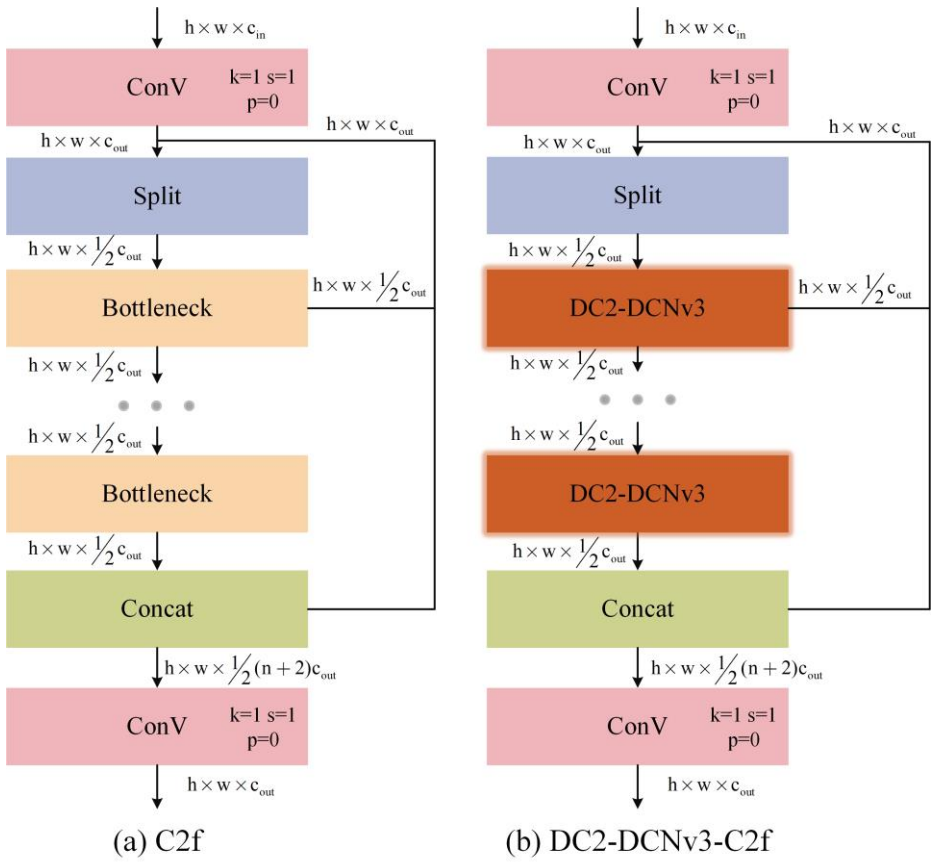

Figure 4. Architecture of C2f and DC2-DCNv3-C2f modules.

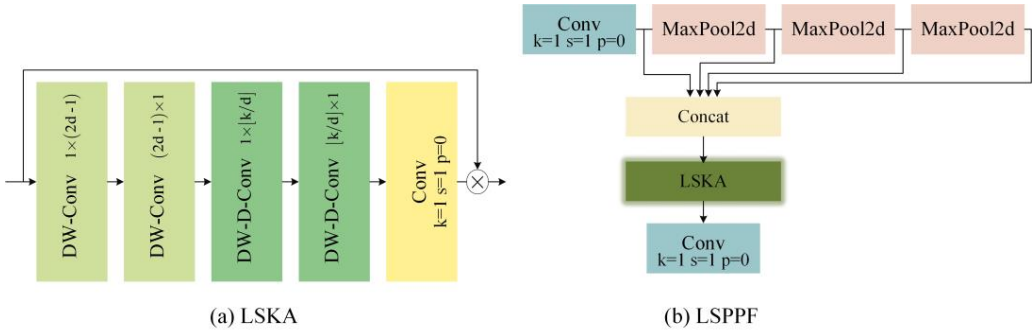

Figure 5. LSKA and LSPPF Module Structures.

The LSPPF module boasts two main advantages: first, it effectively utilizes large convolutional kernels to capture long-range dependencies in the feature map, enhancing the model's understanding of the overall image. Secondly, it restricts the number of model parameters, effectively avoiding a significant increase in computational load, thereby improving performance while maintaining model efficiency. Employing the LSPPF module in the baseline model on the Visdrone2019 dataset resulted in a 1.4% increase in mAP_50 and a 0.8% improvement in recall.

3.3. Neck

Partial convolution exploits the redundancy in the feature map by applying conventional convolution only to part of the input channels. It computes the representative values for all channels using continuous c_p channels in the feature map, leaving the rest of the channels untouched. This design significantly reduces computational workload. When the input and output have the same number of channels, the FLOPs (floating-point operations) for partial convolution are calculated as $h \times w \times k^2 \times {c_p}^2$, where h and w are the height and width of the input feature map, and k is the kernel size.

Assuming c_p represents one-fourth of all channels, the FLOPs of partial convolution are only one-sixteenth of those of standard convolution. The memory access of partial convolution is calculated as follows:

$$h \times w \times 2c_p + k^2 \times c_p^2 \approx h \times w \times 2c_p \quad (10)$$

The memory access of partial convolution is only one-fourth of standard convolution. Point convolution is placed after partial convolution to utilize information from all channels. This decoupling method leverages redundancy between filters and further reduces FLOPs. These two operators form the core components of FNet.

The proposed Random Partial Convolution (RPConV) is illustrated in Figure 6. In partial convolution, the selection of channels for regular convolution involves the first cp channels. In contrast, for random partial convolution, c_p channels are randomly chosen (Random Choice) from the input feature map for spatial feature extraction. Subsequently, the convolution result is merged with the remaining $c\text{-}c_p$ channels (Remainder) that were not selected. As different channels correspond to distinct semantic information and image structures, the random selection of channels during convolution allows for a more comprehensive learning of these features while reducing computational and memory overhead. This leads to more affluent and more diverse semantic information in the output feature map, enhancing adaptability to variations in different scenes and targets, thereby improving overall adaptability to input data.

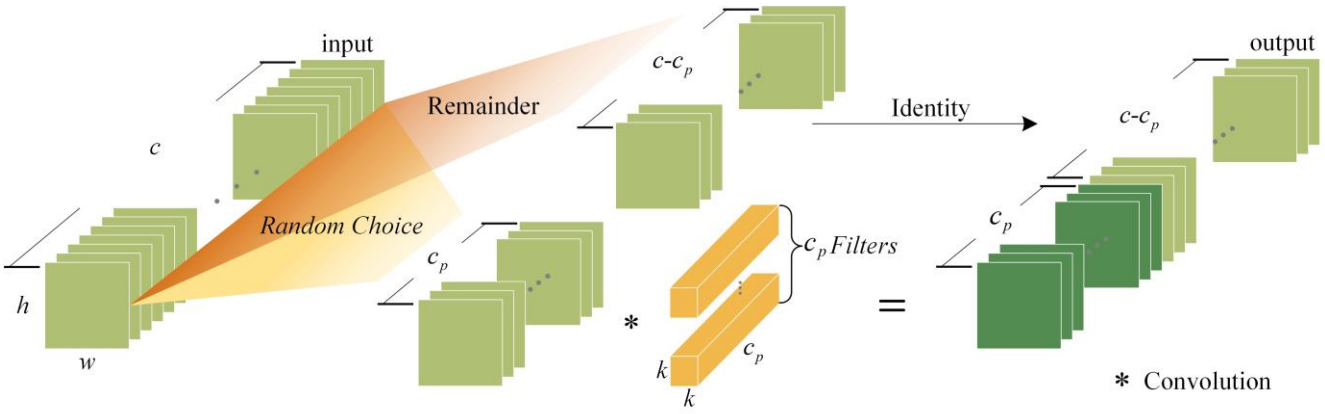

Figure 6. Random Partial Convolution Structure (RPConV).

Figure 7 illustrates the structure of the Random FasterNet Block (RFNet). The RPConV replaces the partial convolution in the FasterNet Block (FNet) [30]. A depth convolution layer was added after the first point convolution to utilize spatial information from all channels effectively. This layer continues the transformation introduced by the point convolution, introducing more non-linear transformations across channels. The processing by the depth convolution helps capture more abstract and complex feature patterns. Finally, the subsequent point convolution reduces the number of channels, allowing the network to focus on crucial features and alleviate the computational burden.

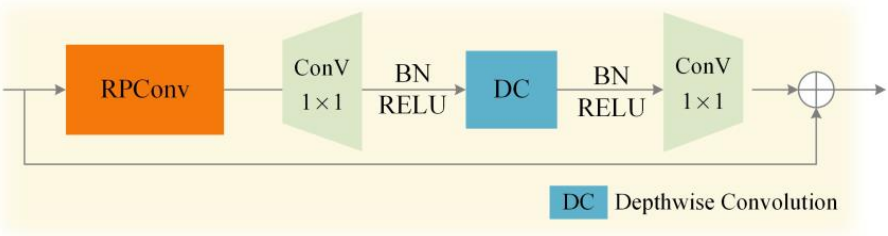

Figure 7. Illustrates the structure of the Random FasterNet Block.

The RFNet replaces the bottleneck in C2f, forming the RFC2f module. The RFC2f module is applied to the position of C2f in the neck. Experimental results demonstrate that on the DOTA validation set, the RFC2f module achieves a mAP_50 score of 69.4% and a mAP_50_95 score of 46.5%. Compared to the FC2f module (where FNet replaces the Bottleneck in C2f), the mAP_50 and mAP_50_95 scores have improved by 2.5% and 2.0%, respectively. There is no significant difference in detection accuracy compared to the baseline model, but the detection speed has increased to 170.7 FPS, a 5.8% improvement, with a 10.9% decrease in GFLOPs. This improvement is expected to be crucial in scenarios requiring real-time detection or lightweight models.

4. Experimental and Analysis

4.1. Dataset Setting

This paper selects the VisDrone2019 [49], DOTA [50], and GDUT-HWD [51] datasets to validate the effectiveness of the proposed improvement model.

VisDrone2019, collected by the AISKYEYE team at the Machine Learning and Data Mining Laboratory of Tianjin University in China, spans 14 cities across China. Thousands of kilometers geographically separate these cities and exhibit distinct terrains and urban layouts, showcasing a wide geographical diversity. The image dimensions in the VisDrone2019 dataset are approximately 2000 × 1500 pixels, with annotations for 11 object categories: pedestrians, people, bicycles, cars, trucks, vans, tricycles, tricycles with sun-

shades, buses, motorcycles, and others. The dataset comprises 6471 images in the training set, 548 in the validation set, and 1610 in the test set, totaling over 2.6 million manually annotated bounding boxes. VisDrone2019 features varying shooting heights and camera angles and diverse lighting conditions, such as insufficient light on rainy days, dim nights, intense light, and glare, posing challenges for target capture. Most objects are small, densely distributed, and prone to occlusion.

The DOTA-v1.0 dataset is a large-scale dataset for aerial image object detection, consisting of 2806 aerial images collected from different sensors and platforms. The image sizes range from 800 to 4000 pixels, with a total annotation of 188,282 instances. The dataset is divided equally into training, testing, and validation sets, each containing 15 common classes: airplane, ship, storage tank, baseball diamond, tennis court, basketball court, athletic field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer field, and swimming pool. In this study, the original rotated bounding boxes in the dataset are transformed into horizontal bounding boxes, and the images are cropped into 1024 $\times$ 1024 patches with a spacing of 200. This process results in 15,749 training images and 5297 validation images. Although the DOTA dataset has fewer instances with occlusion due to its top-down perspective, it presents challenges with densely populated areas containing numerous instances and complex backgrounds.

The GDUT-HWD dataset was collected by searching and downloading from the internet to detect whether people wear safety helmets and identify their respective colors. The dataset exhibits significant variations in scenes, lighting conditions, visual ranges, and individual poses. It includes five categories of labels: blue, white, yellow, and red safety helmets and no safety helmets, totaling 18,893 instances across 3174 images. Small objects, as defined by the COCO dataset, are the most prevalent in this dataset. In this study, the dataset was randomly divided into training and testing sets at a ratio of 7:3.

4.2. Experimental Setup

The experimental setup employed Ubuntu 16.04.1 as the operating system. It was equipped with an Intel (R) Xeon (R) Gold 5218R processor running at 2.10 GHz, 64 GB of RAM, and two GeForce RTX 2080Ti graphics processors, each with 11GB of VRAM. The CUDA version used for accelerating deep learning tasks was 11.3. PyTorch 1.11.0 and torchvision 0.12.0 were utilized for the deep learning model framework.

This study is based on Ultralytics version 8.0.202 of YOLOv8s-P2 for improvement. No pre-training weights are used for any experiments to ensure fairness in comparing models. The image input to the network in the ablation experiments is scaled equally to a width or height of 640, and the remaining portion is filled to 640 using the background. The important parameter settings for the training process are shown in Table 1.

Table 1. Experimental Parameter Settings.

Param	Setup
Epoch	300
Batch	8
Optimizer	SGD
Initial Learning Rate.	1×10^{-2}
Final learning Rate	1×10^{-4}
NMS IoU	0.7
Base weight decay	0.0005
Close Mosaic	0
Optimizer momentum	0.937
Warmup epochs	5
Patience	50

4.3. Evaluation Metrics

To comprehensively assess the contribution of the proposed improvements to enhancing detection performance, we utilize mean average precision (mAP), precision, recall, model parameters (Params), giga floating-point operations per second (GFLOPs), model size, and frames per second (FPS) as evaluation metrics for model performance. GFLOPs measure the algorithm's complexity, where a higher value indicates a greater demand for floating-point operations. Model size represents the memory occupied by the weight file generated after model training. Precision and recall are defined by Equations (11) and (12), respectively,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

Taking pedestrians in the dataset as an example, TP represents the number of correctly classified instances as pedestrians, FP represents instances of other classes misclassified as pedestrians, and FN represents the number of pedestrians misclassified as other classes. Precision is employed to assess the accuracy of instances classified as pedestrians, indicating how accurate or precise the classification is. On the other hand, recall evaluates the ability to find correctly classified pedestrians, measuring the model's capability to identify authentic positive samples among all actual positive instances.

$$\text{AP} = \int_0^1 \text{P(r)dr} \tag{13}$$

$$\text{mAP\_50} = \frac{1}{\text{N}} \sum_{\text{i}=1}^{\text{N}} \text{AP}_\text{i}(\text{IOU}_{\text{thresh}} = 0.5) \tag{14}$$

$$\text{mAP\_50\_95} = \frac{1}{10} \sum_{\text{j}} \frac{1}{\text{N}} \sum_{\text{i}=1}^{\text{N}} \text{AP}_\text{i}(\text{IoU}_{\text{thresh}} = \text{j}) \tag{15}$$

In Equation (13), AP stands for Average Precision, the area under the Precision Recall (P-R) curve, representing the average precision for detecting the pedestrian category. mAP refers to the mean average precision calculated by averaging the AP values across all categories. In Equation (14), mAP_50 denotes the average precision across all categories at an Intersection over Union (IoU) threshold of 0.5. In Equation (15), mAP_50_95 represents the average of ten mAP values calculated for ten different IoU thresholds, denoted by j, ranging from 0.5 to 0.95 with a step size of 0.05.

4.4. Ablation Study

4.4.1. Impact of DC2-DCNv3

DCNv3 is fused with the C2f module in the same manner as shown in Figure 4b, referred to as the DCNv3-C2f module. DCNv3-C2f, DC2-DCNv3-C2f, and C2f are used as feature extraction modules for the backbone network in the benchmark models, and their performances are compared on the DOTA dataset. Table 2 presents the experimental results of the DOTA validation set.

Table 2. Influence of Different Feature Extraction Modules on the Backbone Network.

Module	Precision [%]	Recall [%]	mAP_50 [%]	mAP_50_95 [%]
C2f	75.1	65.4	69.5	46.1
DCNv3-C2f	76.7	67.9	71.0	48.2
DC2-DCNv3-C2f	77.5	68.5	72.2	49.2

Table 2 and Figure 8 reveal that the introduction of deformable convolution has enabled the model to adaptively learn receptive fields for objects of different shapes, effectively enhancing detection accuracy. The proposed DC2-DCNv3-C2f module in this

paper improves mAP_50_95 by 1.0% and recall by 0.6% compared to the DCNv3-C2f module. Furthermore, relative to the C2f module, both mAP_50_95 and recall increased by 3.1%. By decoupling the learning process of offset and modulation scalar through two depth-wise separable convolution operations, the precise determination of pixel offset positions has been achieved, facilitating an accurate assessment of the extent to which each pixel's displacement is accepted.

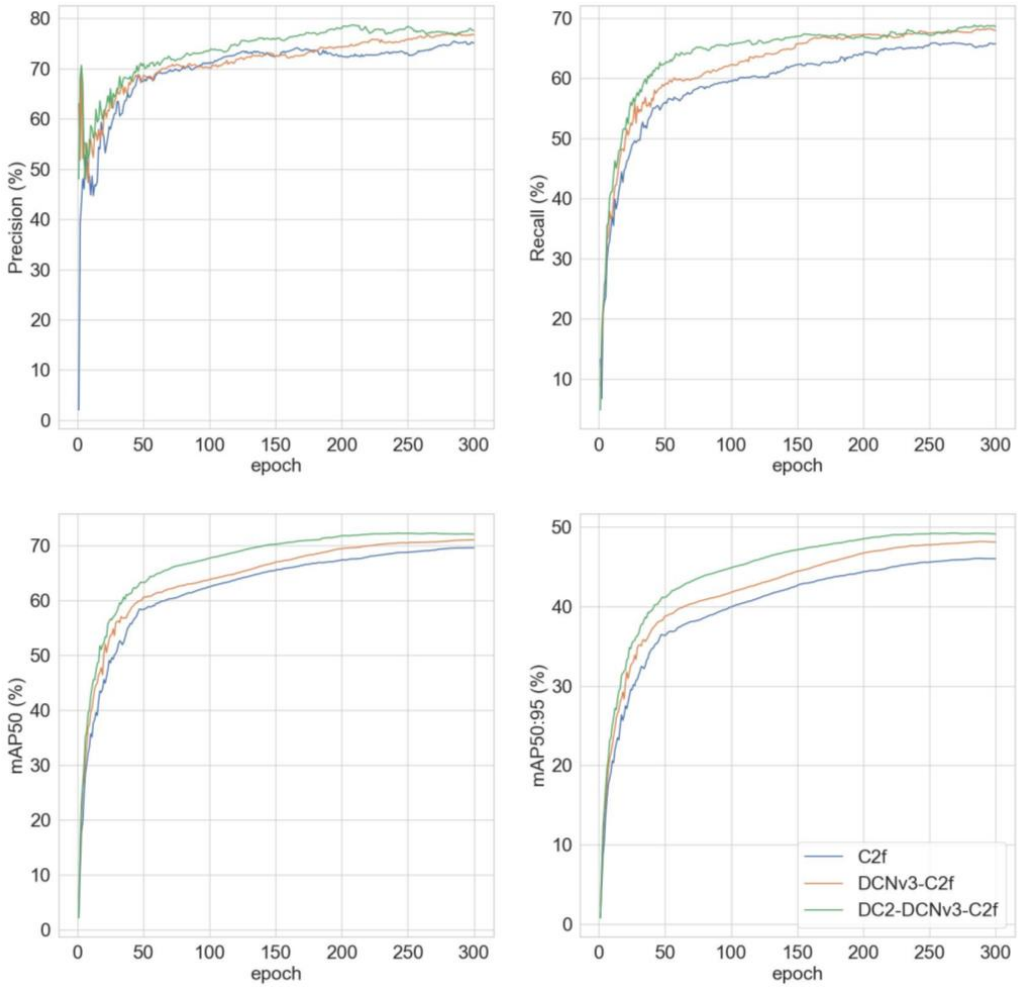

Figure 8. Influence of Different Feature Extraction Modules on the Model Training Process.

4.4.2. Impact of RFC2f

The FC2f and RFC2f modules are employed in the neck network and compared with the C2f module used in the baseline model on the DOTA dataset. The experimental results for the DOTA validation set are presented in Table 3.

Table 3. Impact of Different Lightweight Modules on the Model.

Module	Precision [%]	Recall [%]	mAP_50 [%]	mAP_50_95 [%]	GFLOPs	FPS
C2f	75.1	65.4	69.5	46.1	36.7	161.3
FC2f	73.6	63.4	66.9	44.5	32.5	175.6
RFC2f	74.9	66.0	69.4	46.5	32.7	170.7

Table 3 and Figure 9 show that the FC2f module achieves lightweighting at the expense of precision. The proposed RFC2f module introduces randomness in partial convolutions and incorporates depth-wise convolutions to accommodate the increased non-linear transformation channels resulting from the point convolutions. This enables the model to capture richer and more abstract semantic features, deepening its understanding of the image as a whole while utilizing convolutions with fewer channels. Compared to the baseline model, mAP_50 decreases by 0.1%, while mAP_50_95 improves by 0.4%. Additionally, GFLOPs are reduced by 10.9%, and FPS increases by 5.8%. Overall, the model demonstrates lightweighting without sacrificing precision, affirming the effectiveness of the RFC2f module.

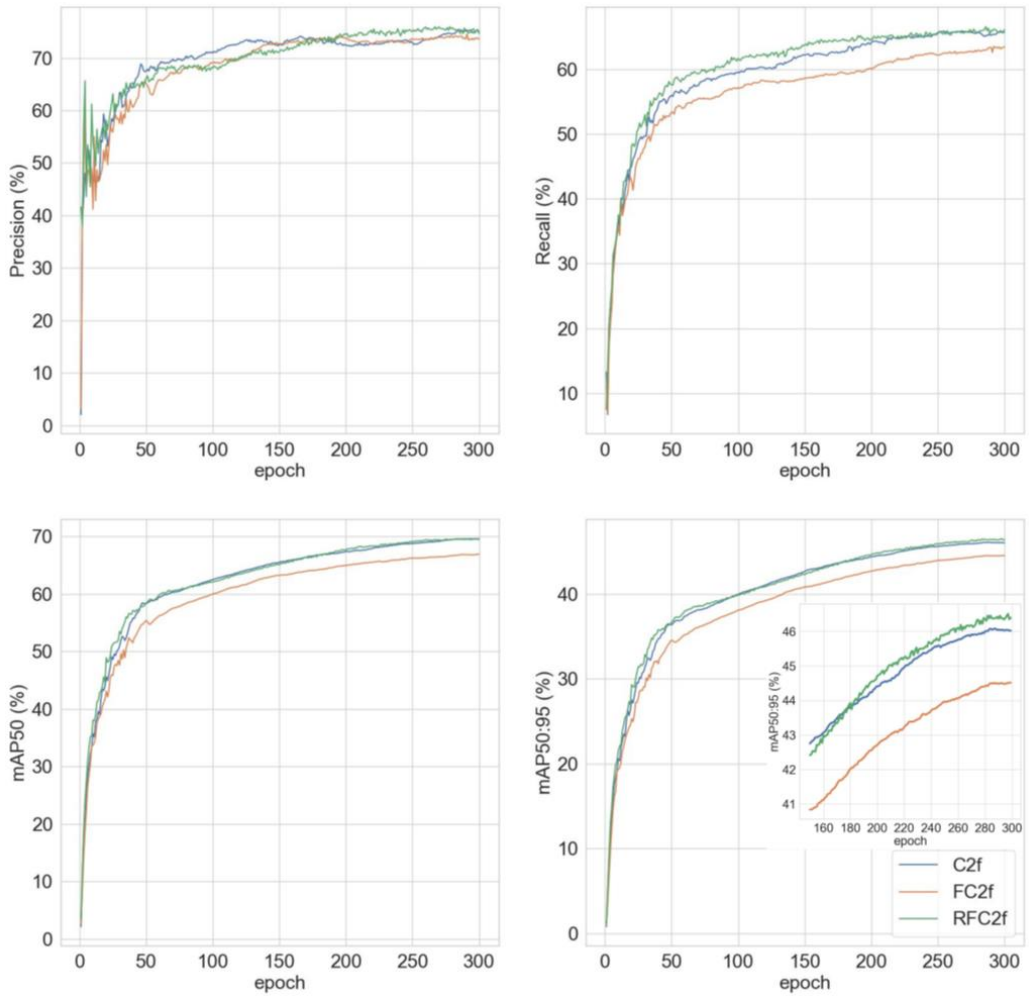

Figure 9. Impact of Different Lightweight Modules on the Training Process of the Model on the DOTA Dataset.

4.4.3. Impact of Integrated Enhancements

In this study, we propose four enhancements to the baseline model, aiming to enhance further the performance of small target detection in unmanned aerial vehicle (UAV) images. These four improvements include (a) Slicing Assistance for dataset augmentation, (b) replacing SPPF with LSPPF, (c) adopting the DC2-DCNv3-C2f module for the backbone network, and (d) employing the RFC2f module for the neck network. To quantitatively analyze the necessity of each enhancement, we conducted ablation experiments on the VisDrone2019 dataset, and the results of the test set are presented in Tables 4 and 5.

Table 4. The ablation study of each improvement point based on the baseline model was conducted on the VisDrone test set.

Model	Precision [%]	Recall [%]	mAP_50 [%]	mAP_50_95 [%]
Baseline	48.6	37.3	36.2	20.8
a	49.9	38.6	38.0	21.7
b	49.7	38.1	37.6	21.4
c	50.1	38.9	38.3	21.8
d	48.5	37.6	36.1	20.9
a + b + c	51.6	39.0	40.9	23.0
a + b + d	50.6	38.8	38.5	22.3
a + b + c + d (our)	51.4	39.2	40.7	23.1

Table 5. Impact of Enhancement Point (d) on Model Lightweight Design.

Model	GFLOPs	FPS	Params [M]
Baseline	36.7	137.0	10.6
a + b + c	36.1	128.4	11.9
d	32.7	148.2	9.2
a + b + c + d (our)	31.8	136.4	10.4

Analysis of Experimental Results in Tables 4 and 5:

1. Slicing Assistance (a): During both model training and inference stages, the use of slicing assistance resulted in a 1.3% increase in recall, a 1.8% increase in mAP_50, and a 0.9% increase in mAP_50_95 relative to the baseline model. This demonstrates that slicing-aided, by enhancing the relative size of targets, effectively alleviated the issue of missed detections, leading to the most notable improvement in detection accuracy among the four enhancement points.

2. Integration of LSPPF with the Last Output Layer of the Backbone Network (b): Fusion of the last output layer of the backbone network with LSPPF resulted in a 1.4% increase in mAP_50 and a 0.6% increase in mAP_50_95. This indicates that the LSKA attention mechanism effectively enhanced the focus on crucial information in higher-level feature maps, improving detection accuracy.

3. Utilizing DC2-DCNv3-C2f Module in the Backbone Network (c): The use of the DC2-DCNv3-C2f module in the backbone network led to a 1.6% improvement in recall, a 2.1% improvement in mAP_50, and a 1.0% improvement in mAP_50_95. This module enhanced the extraction of feature maps by the backbone network to better reflect detailed information of various target classes, thereby improving the quality of feature maps.

4. Application of RFC2f Module in the Neck Network (d): While there was not a significant change in accuracy, the RFC2f module resulted in a 10.9% reduction in GFLOPs, an 8.2% increase in FPS, and a 13.2% decrease in Params. This substantial reduction in computational load accelerated the model's image processing speed, making the model more lightweight.

For the Baseline + a + b + c model, GFLOPs decreased by 1.6%, FPS decreased by 6.3%, Params increased by 12.3%, mAP_50 increased by 4.7%, and mAP_50_95 increased by 2.2%. The design of cascade convolution and DC2-DCNv3 allows the model to obtain the benefits of large kernel convolutions with adaptive receptive fields and also reduces the model computation to a certain extent. However, the need to merge results from multiple slices during slicing-aided inference led to a 6.3% decrease in FPS. After integrating enhancement point (d), the decline in inference speed was alleviated, with the FPS of our model decreasing only by 0.4% compared to the baseline model.

The proposed model in this study achieved a 2.8% improvement in precision, 1.9% improvement in recall, 4.5% improvement in mAP_50, and an 2.3% improvement in mAP_50_95 while reducing GFLOPs by 13.4%, Params by 1.9%, and FPS by only 0.4%. This enhanced detection accuracy and achieved lightweight model design, meeting the

real-time and accuracy requirements of unmanned aerial vehicle (UAV) target detection tasks. Specific detection samples from selected scenes are showcased in Figure 8, focusing on enlarging positions that are difficult to observe closely.

The four images in Figure 10 contain six target categories: car, truck, bus, pedestrian, motorcycle, and people (some of the hard-to-see areas are shown enlarged). The proposed improved model has enhanced the detection capability of small targets across various detection scenarios. However, targets with residual shadows, such as the motorcycle in the bottom right corner of Figure 10a, and situations involving tiny and densely packed targets pose significant challenges to the model, leading to inevitable instances of missed detections. From a high-altitude perspective, the pixel information captured for trucks and buses appears similar, resulting in potential misidentifications. In Figure 10d, a portion outlined in the box contained four buses, with the baseline model erroneously detecting two of them, while the improved model correctly identified all buses. This indicates that the approach presented in this paper has mitigated this issue.

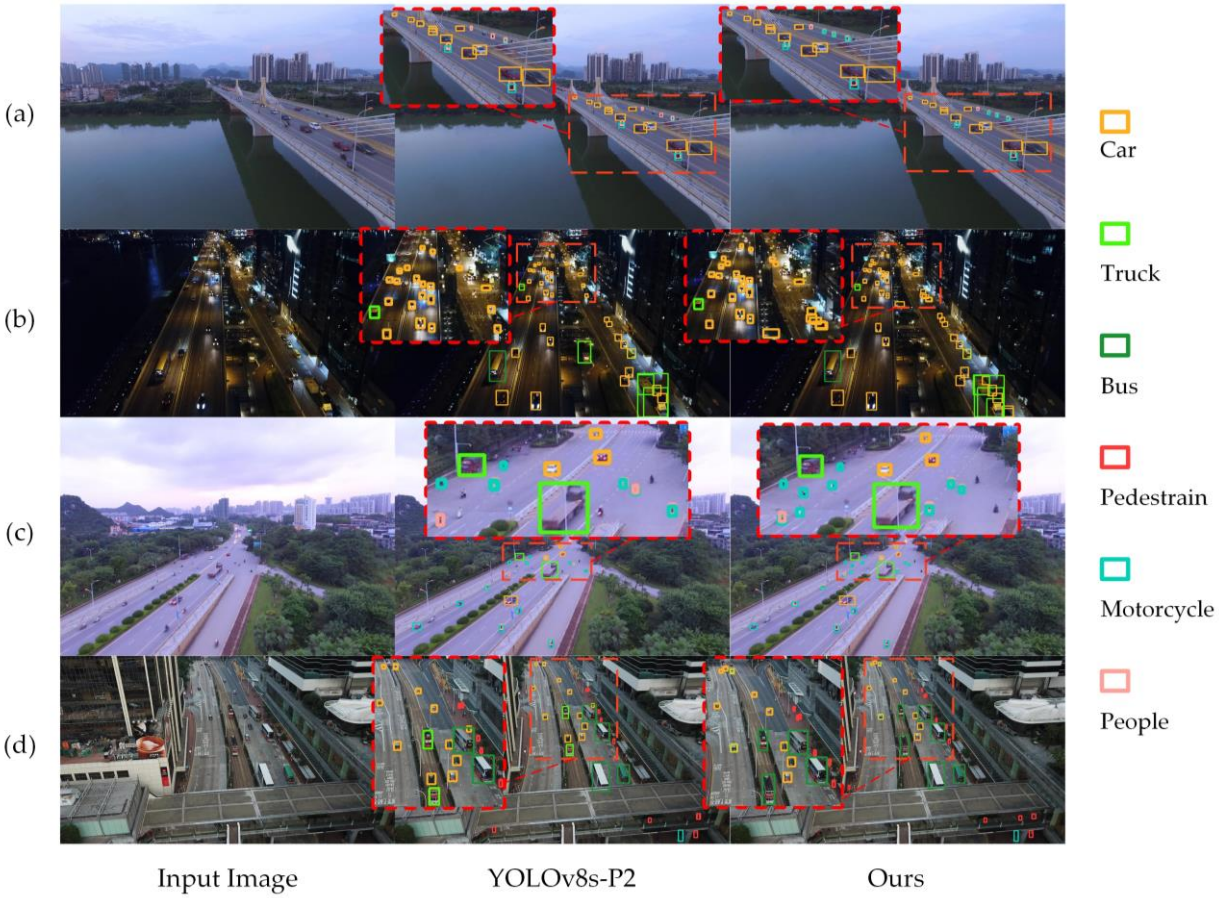

Figure 10. Exemplary Detection Results on the VisDrone2019 Test Set. (**a**) Insufficient lighting, small targets. (**b**) At night, small targets. (**c**) Insufficient lighting, small targets. (**d**) Complex background, insufficient lighting, small targets.

Figure 11 illustrates the detection results of the baseline model and the proposed enhanced model on the GUDT-HWD test dataset. The target categories for the three images are none, yellow, red, and white. The proposed enhanced model improves detection performance in drone scenarios and demonstrates excellent detection performance in helmet detection tasks. This indicates that the proposed enhanced model exhibits good robustness and applicability.

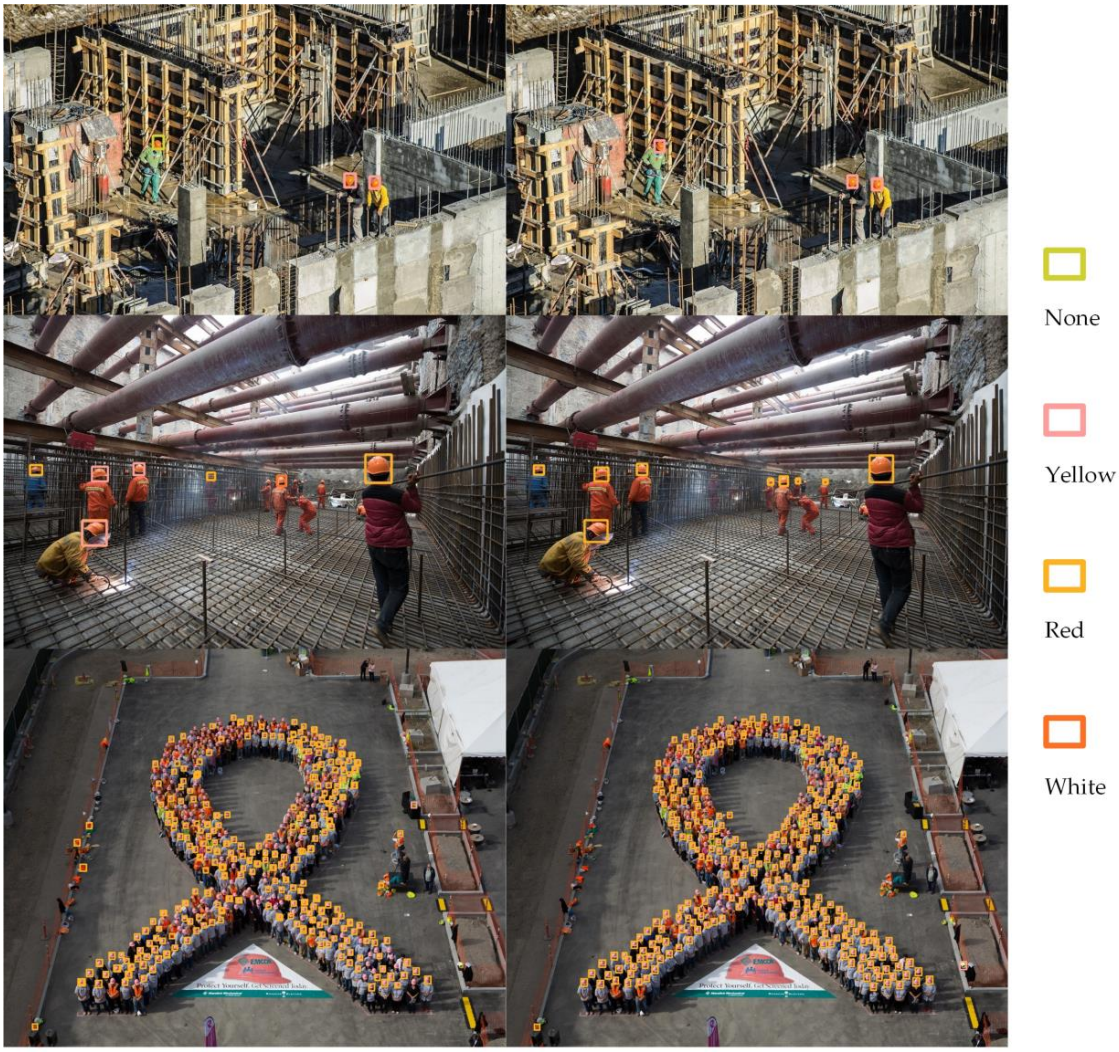

Figure 11. Exemplary Detection Results on the GUDT-HWD Test Set.

Missed detection refers to instances where objects are present but detected as background. The miss detection rate for a certain category is calculated as the number of missed targets divided by the total number of targets in the dataset. Table 6 indicates that the improved model exhibits a decreasing trend in missed detection rates across all categories, highlighting its universality across different categories. The most significant decrease in missed detection rate is observed in trucks, followed by motorcycles, cars, and pedestrians. These three categories have many instances in the dataset, and the targets are tiny, indicating that the proposed improvement method effectively mitigates the problem of missed detections for small targets.

Table 6. Miss detection rate for Various Object Categories in the VisDrone Validation Set.

	Pedestrian	**People**	**Bicycle**	**Car**	**Van**	**Truck**	**Tricycle**	**Awning -Tricycle**	**Bus**	**Motorcycle**
Baseline	0.69	0.85	0.9	0.26	0.67	0.63	0.75	0.86	0.51	0.64
Our Method	0.65	0.82	0.86	0.24	0.64	0.56	0.71	0.84	0.49	0.59
Decrease [%]	(5.8)	(3.5)	(4.4)	(7.7)	(4.5)	(11.1)	(5.3)	(2.3)	(3.9)	(7.8)

Utilizing Grad-CAM to visualize attention regions in the network contributes to interpreting the model's performance. As observed in Figure 12, the improved model

exhibits more comprehensive attention to small targets in the lower layers compared to the baseline model. For instance, distant cars in the background of the elevated bridge in image (a), cars under the intersecting overpass in image (b), black and white cars under the tree in image (c), and cars in the parking lot in image (d) all receive more effective attention in the lower layers of the improved model compared to the baseline. In the higher layers of the improved model, attention to the targets is more extensive, with orange or even deep red coverage in most regions containing targets. In contrast, the baseline model effectively attends to close-range targets but fails to cover orange or even deep red regions for distant targets. This indicates that the proposed improved model effectively prevents information loss during the downsampling of small targets, demonstrating clear advantages in handling small targets and overall image understanding.

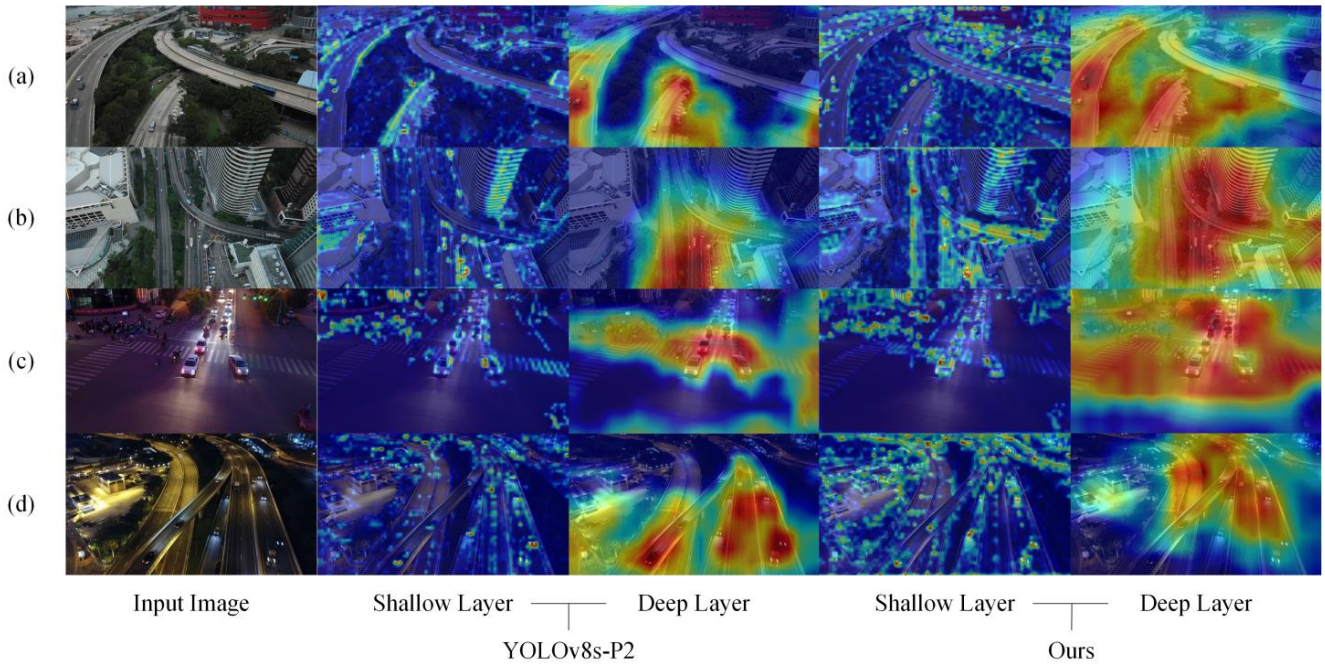

Figure 12. Grad-CAM Visualization of Shallow and Deep Layers in the Backbone Network. (**a**) Insufficient lighting, small targets. (**b**) Insufficient lighting, small targets. (**c**) At night, small targets. (**d**) At night, small targets.

4.5. Comparative Study

To further validate the performance advantages of the proposed method for small target detection in unmanned aerial vehicles (UAVs), a comparative study was conducted against state-of-the-art detection models in the field, along with the relatively larger-scale YOLOv8m as a benchmark. The performance of each model on the VisDrone2019 validation set is presented in Table 7.

Table 7. Performance Comparison of Various Detection Algorithms.

Methods	Image Size	mAP_50_95	mAP_50	FPS	GPU
Our	640 × 640	29.8	48.3	136.4	GeForce RTX 2080Ti
YOLOv8m	640 × 640	26.8	43.8	75.0	GeForce RTX 2080Ti
MFFSODNet [52]	640 × 640	-	45.5	70	TITAN RTX
FE-YOLOv5 [53]	640 × 640	21.0	37.0	-	GeForce RTX 2080Ti
AVS-YOLO [54]	416 × 640	22.19	43.4	31.8	GeForce RTX 2080Ti
FPN+SARSA+TDA+LSRN [55]	600 × 1000	25.8	51.5	1.3	TITAN Xp
MMF-YOLO [56]	640 × 640	-	42.2	-	GeForce RTX 3080Ti
Li et al. [57]	640 × 640	-	42.2	167.0	GeForce RTX 3090Ti
HR-FPN [58]	1024 × 1024	-	50.8	23.9	GeForce RTX 3090

Table 7 indicates that our improved model achieves a 4.0% higher mAP_50_95 compared to the two-stage improved model mentioned in [55]. Additionally, compared to the larger-scale YOLOv8m model, our model demonstrates faster detection speed and higher accuracy. It exhibits a noticeable advantage in accuracy over models based on the YOLO series [53,54,56,57]. As the targets in the input images are relatively larger than those in the original images, the DC2-DCNv3-C2f module of the backbone network adapts to capture detailed features of objects of various shapes and sizes. Moreover, the final output layer learns mixed attention features, enabling our model to perceive the details of small targets. Although the mAP50 of [58] is slightly higher than that of our improved model, its GPU is superior to ours, with only 23.9 FPS, about one-sixth of that of the improved model in this paper. Overall, our improved algorithm demonstrates excellent detection performance for UAV aerial images, exhibiting good performance in speed and accuracy. However, there is room for improvement in detecting situations where targets are mutually occluded.

5. Conclusions

This paper proposes an aerial image detection model based on YOLOv8s-P2, overcoming the challenges of detecting small, densely distributed, and poorly illuminated objects in images captured by unmanned aerial vehicles (UAVs). While meeting real-time detection requirements, the model significantly improves accuracy. Introducing the slicing-aided technique at the data level increases the relative size of small targets in images, providing richer pixel information for the model. Additionally, the proposed Large Kernel Spatial Pyramid Pooling Fast module aims to enhance the model's focus on critical regions in the overall context, improving sensitivity to global semantic structures. The cascaded convolution kernel design of LSPPF introduces no additional computational burden.

Conventional convolution with shared parameters tends to extract coarse-grained information for fine-grained targets. The DC2-DCNv3-C2f module is introduced to address this limitation, leveraging the advantages of deformable convolution. By decoupling the learning process of offset and modulation scalar, the module allows the model to better adapt to different shapes of objects, focusing on target areas and effectively reducing interference from background noise. Lastly, considering the limited computational resources of embedded devices on UAVs, the RFNet module is proposed. By exploiting the redundancy of feature maps, it randomly selects part of the channels for convolution, reducing computational and memory access costs while enriching feature representation. The first point convolution introduces more non-linear transformation channels, capturing more abstract spatial features with deep convolution, and the second point convolution reduces dimensionality, allowing the network to focus on more critical features. Without compromising model accuracy, this design provides greater flexibility in adapting to the limitations of embedded devices.

Experiments demonstrate that our proposed method significantly reduces the missed detection rate of small targets in UAV aerial images, even in low-light conditions at night, accurately detecting small and dense targets. Compared to advanced UAV object detection algorithms, our model achieves an excellent balance between speed and accuracy. This research contributes to advancing UAV object detection algorithms, offering a substantial solution to the problem of detecting small targets in aerial images. However, the study has limitations; the experimental data were only validated on the VisDrone2019 and DOTA datasets. Future work should include verifying the model's improvement on more UAV aerial datasets to further confirm its generality and robustness.

While this study has achieved satisfactory results in the current experimental environment, further validation is required on detection terminals. Future work will focus on deploying the algorithm in real-world scenarios and testing it on drone hardware to verify its real-time processing capability and detection accuracy.

Author Contributions: Conceptualization, H.R.; Funding acquisition, M.Z. and H.R.; Methodology, H.R.; Resources, H.R.; Software, S.L. and M.Z.; Validation, S.L. and M.Z.; Visualization, S.L.; Writing—original draft, R.T. and H.R.; Writing—review & editing, R.T. and H.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received support from the Natural Science Foundation of Heilongjiang Province (Grant No. LH2020F040), the Young Doctoral Research Initiation Fund Project at Harbin University, titled "Research on Wood Recognition Methods Based on Deep Learning Fusion Model" (Project No. 2022ZCZJCG022), the Self-funded project at Harbin Science and Technology Plan for Research on Computer Vision Recognition Technology of Wood Species Based on Transfer Learning Fusion Model (Project No. ZC2022ZJ010027), and the Fundamental Research Funds for the Central Universities (Grant No. 2572017PZ10).

Data Availability Statement: VisDrone2019 Dataset: The VisDrone2019 dataset used in this study is publicly available and can be accessed from the official GitHub repository: https://github.com/VisDrone/VisDrone-Dataset (accessed on 18 November 2023), DOTA-v1.0 Dataset: The DOTA-v1.0 dataset used in this study is publicly accessible and can be obtained from the official repository: https://captain-whu.github.io/DOTA/dataset.html (accessed on 26 November 2023), GDUT-HWD Dataset: The GDUT-HWD dataset used in this study is publicly accessible and available from the official repository: https://github.com/wujixiu/helmet-detection (accessed on 20 April 2024).

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

1. Butilă, E.V.; Boboc, R.G. Urban traffic monitoring and analysis using unmanned aerial vehicles (uavs): A systematic literature review. *Remote Sens.* **2022**, *14*, 620. [CrossRef]
2. Vasilopoulos, E.; Vosinakis, G.; Krommyda, M.; Karagiannidis, L.; Ouzounoglou, E.; Amditis, A. A comparative study of autonomous object detection algorithms in the maritime environment using a UAV platform. *Computation* **2022**, *10*, 42. [CrossRef]
3. Talaat, F.M.; ZainEldin, H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput. Appl.* **2023**, *35*, 20939–20954. [CrossRef]
4. 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; Proceedings, Part V 13, 2014. pp. 740–755.
5. 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, 18–22 June 2023; pp. 7464–7475.
6. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
7. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer International Publishing: Cham, Switzerland, 2020; pp. 213–229.
8. 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.
9. Guan, W.; Zou, Y.X.; Zhou, X. Multi-scale object detection with feature fusion and region objectness network. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2596–2600.
10. Zeng, N.; Wu, P.; Wang, Z.; Li, H.; Liu, W.; Liu, X. A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3507014. [CrossRef]
11. Deng, C.; Wang, M.; Liu, L.; Liu, Y.; Jiang, Y. Extended feature pyramid network for small object detection. *IEEE Trans. Multimed.* **2021**, *24*, 1968–1979. [CrossRef]
12. Sun, W.; Dai, L.; Zhang, X.; Chang, P.; He, X. RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring. *Appl. Intell.* **2022**, *52*, 8448–8463. [CrossRef]
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, NV, USA, 3–6 December 2012.
14. Wang, S. An augmentation small object detection method based on NAS-FPN. In Proceedings of the 2020 7th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 18–20 December 2020; pp. 213–218.
15. Ali-Gombe, A.; Elyan, E. MFC-GAN: Class-imbalanced dataset classification using multiple fake class generative adversarial network. *Neurocomputing* **2019**, *361*, 212–221. [CrossRef]

16. Bosquet, B.; Cores, D.; Seidenari, L.; Brea, V.M.; Mucientes, M.; Del Bimbo, A. A full data augmentation pipeline for small object detection based on generative adversarial networks. *Pattern Recognit.* **2023**, *133*, 108998. [CrossRef]
17. Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J.; Cho, K. Augmentation for Small Object Detection. In Proceedings of the 9th International Conference on Advances in Computing and Information Technology (ACITY 2019), Sydney, Australia, 21–22 December 2019; Aircc Publishing Corporation: Chennai, India, 2019; pp. 119–133.
18. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; p. 30.
20. 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.
21. Lim, J.-S.; Astrid, M.; Yoon, H.-J.; Lee, S.-I. Small object detection using context and attention. In Proceedings of the 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Republic of Korea, 13–16 April 2021; pp. 181–186.
22. Zhan, W.; Sun, C.; Wang, M.; She, J.; Zhang, Y.; Zhang, Z.; Sun, Y. An improved Yolov5 real-time detection method for small objects captured by UAV. *Soft Comput.* **2022**, *26*, 361–373. [CrossRef]
23. Lu, X.; Ji, J.; Xing, Z.; Miao, Q. Attention and feature fusion SSD for remote sensing object detection. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 5501309. [CrossRef]
24. Fang, Y.; Liao, B.; Wang, X.; Fang, J.; Qi, J.; Wu, R.; Niu, J.; Liu, W. You only look at one sequence: Rethinking transformer in vision through object detection. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 26183–26197.
25. Zhang, J.; Xia, K.; Huang, Z.; Wang, S.; Akindele, R.G. ETAM: Ensemble transformer with attention modules for detection of small objects. *Expert Syst. Appl.* **2023**, *224*, 119997. [CrossRef]
26. Akyon, F.C.; Altinuc, S.O.; Temizel, A. Slicing aided hyper inference and fine-tuning for small object detection. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 966–970.
27. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
28. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9308–9316.
29. Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 22 August 2023; pp. 14408–14419.
30. 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, 18–19 June 2023; pp. 12021–12031.
31. 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.
32. 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.
33. Hong, M.; Li, S.; Yang, Y.; Zhu, F.; Zhao, Q.; Lu, L. Sspnet: Scale selection pyramid network for tiny person detection from uav images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 8018505. [CrossRef]
34. Shahin, A.I.; Almotairi, S. SVA-SSD: Saliency visual attention single shot detector for building detection in low contrast high-resolution satellite images. *PeerJ Comput. Sci.* **2021**, *7*, e772. [CrossRef] [PubMed]
35. Chai, E.; Chen, L.; Hao, X.; Zhou, W. Mitigate the scale imbalance via multi-scale information interaction in small object detection. *Neural Comput. Appl.* **2024**, *36*, 1699–1712. [CrossRef]
36. Ruiz-Ponce, P.; Ortiz-Perez, D.; Garcia-Rodriguez, J.; Kiefer, B. Poseidon: A data augmentation tool for small object detection datasets in maritime environments. *Sensors* **2023**, *23*, 3691. [CrossRef]
37. Li, J.; Liang, X.; Wei, Y.; Xu, T.; Feng, J.; Yan, S. Perceptual generative adversarial networks for small object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1222–1230.
38. Wan, X.; Yu, J.; Tan, H.; Wang, J. LAG: Layered objects to generate better anchors for object detection in aerial images. *Sensors* **2022**, *22*, 3891. [CrossRef]
39. Mnih, V.; Heess, N.; Graves, A. Recurrent models of visual attention. In Proceedings of the Advances in Neural Information Processing Systems 27 2014, Montreal, QC, Canada, 8–13 December 2014.
40. Jaderberg, M.; Simonyan, K.; Zisserman, A. Spatial transformer networks. In Proceedings of the Advances in Neural Information Processing Systems 28 2015, Montreal, QC, Canada, 7–12 December 2015.
41. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Vedaldi, A. Gather-excite: Exploiting feature context in convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 31 2018, Montreal, QC, Canada, 3–8 December 2018.
42. Gao, Z.; Xie, J.; Wang, Q.; Li, P. Global second-order pooling convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3024–3033.

43. 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.
44. Lee, H.J.; Kim, H.E.; Nam, H. Srm: A style-based recalibration module for convolutional neural networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1854–1862.
45. Yang, Z.; Zhu, L.; Wu, Y.; Yang, Y. Gated channel transformation for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11794–11803.
46. Liu, X.; Leng, C.; Niu, X.; Pei, Z.; Cheng, I.; Basu, A. Find small objects in UAV images by feature mining and attention. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 6517905. [CrossRef]
47. Yang, L.; Zhong, J.; Zhang, Y.; Bai, S.; Li, G.; Yang, Y.; Zhang, J. An improving faster-RCNN with multi-attention ResNet for small target detection in intelligent autonomous transport with 6G. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 7717–7725. [CrossRef]
48. Lau, K.W.; Po, L.M.; Rehman, Y.A.U. Large Separable Kernel Attention: Rethinking the Large Kernel Attention Design in CNN. *Expert Syst. Appl.* **2024**, *236*, 121352. [CrossRef]
49. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
50. Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
51. Wu, J.; Cai, N.; Chen, W.; Wang, H.; Wang, G. Automatic detection of hardhats worn by construction personnel: A deep learning approach and benchmark dataset. *Autom. Constr.* **2019**, *106*, 102894. [CrossRef]
52. Jiang, L.; Yuan, B.; Du, J.; Chen, B.; Xie, H.; Tian, J.; Yuan, Z. MFFSODNet: Multi-Scale Feature Fusion Small Object Detection Network for UAV Aerial Images. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 5015214. [CrossRef]
53. Wang, M.; Yang, W.; Wang, L.; Chen, D.; Wei, F.; KeZiErBieKe, H.; Liao, Y. FE-YOLOv5: Feature enhancement network based on YOLOv5 for small object detection. *J. Vis. Commun. Image Represent.* **2023**, *90*, 103752. [CrossRef]
54. Ma, Y.; Chai, L.; Jin, L.; Yu, Y.; Yan, J. AVS-YOLO: Object detection in aerial visual scene. *Int. J. Pattern Recognit. Artif. Intell.* **2022**, *36*, 2250004. [CrossRef]
55. Deng, S.; Li, S.; Xie, K.; Song, W.; Liao, X.; Hao, A.; Qin, H. A global-local self-adaptive network for drone-view object detection. *IEEE Trans. Image Process.* **2020**, *30*, 1556–1569. [CrossRef] [PubMed]
56. Zhang, Q.; Zhang, H.; Lu, X. Adaptive Feature Fusion for Small Object Detection. *Appl. Sci.* **2022**, *12*, 11854. [CrossRef]
57. 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]
58. Chen, Z.; Ji, H.; Zhang, Y.; Zhu, Z.; Li, Y. High-Resolution Feature Pyramid Network for Small Object Detection on Drone View. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *34*, 475–489. [CrossRef]