drones

MDPI

Article

A High Performance Air-to-Air Unmanned Aerial Vehicle Target Detection Model

Hexiang Hao, Yueping Peng *, Zecong Ye, Baixuan Han, Xuekai Zhang, Wei Tang, Wenchao Kang and Qilong Li

School of Information Engineering, Engineering University of PAP, Xi'an 710086, China; hhx1214s@163.com (H.H.); yzc6666@yeah.net (Z.Y.); hbx911wj@163.com (B.H.); 18510004596@163.com (X.Z.); 13414667546@163.com (W.T.); 18259530623@163.com (W.K.); liqilong20020705@163.com (Q.L.)

* Correspondence: percy001@163.com

Abstract: In the air-to-air UAV target detection tasks, the existing algorithms suffer from low precision, low recall and high dependence on device processing power, which makes it difficult to detect UAV small targets efficiently. To solve the above problems, this paper proposes an high-precision model, ATA-YOLOv8. In this paper, we analyze the problem of UAV small target detection from the perspective of the efficient receptive field. The proposed model is evaluated using two air-to-air UAV image datasets, MOT-FLY and Det-Fly, and compared with YOLOv8n and other SOTA algorithms. The experimental results show that the mAP50 of ATA-YOLOv8 is 94.9% and 96.4% on the MOT-FLY and Det-Fly datasets, respectively, which are 25% and 5.9% higher than the mAP of YOLOv8n, while maintaining a model size of 5.1 MB. The methods in this paper improve the accuracy of UAV target detection in air-to-air scenarios. The proposed model's small size, fast speed and high accuracy make it possible for real-time air-to-air UAV detection on edge-computing devices.

Keywords: drone-to-drone detection; airborne vision; small object detection; deep learning

check for updates

Academic Editor: Anastasios Dimou

Received: 21 January 2025
Revised: 13 February 2025
Accepted: 14 February 2025
Published: 19 February 2025

Citation: Hao, H.; Peng, Y.; Ye, Z.; Han, B.; Zhang, X.; Tang, W.; Kang, W.; Li, Q. A High Performance Air-to-Air Unmanned Aerial Vehicle Target Detection Model. *Drones* **2025**, *9*, 154. https://doi.org/10.3390/drones9020154

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Driven by military demand and the rapid progress of UAV technology, the unmanned aerial vehicle (UAV) has been widely used in both military [1] and civilian fields [2–4], and the purchase and use of 'low, slow and small' UAVs has grown exponentially. 'Low, slow and small' UAV targets refer mainly to UAV targets that fly at low altitude (flight altitude not exceeding 1000 m), move at slow speeds (below 200 km/h) and have a small radar reflection area (below 2 m^2). The lack of comprehensive legal provisions and monitoring measures to control 'low, slow and small' drone targets has led to the emergence of misuse of 'low, slow and small' UAVs globally. The methods used in UAV detection technology mainly include acoustic detection [5], radar detection [6], radio frequency detection [7] and photoelectricity detection [8]. UAV acoustic detection systems detect targets through the frequency of sound generated by engines and propellers, and the sound pressure decays as the distance from the source increases, so acoustic detection systems have a limited range and are often ineffective in noisy environments (e.g., commercial airports). Radar, the primary tool used to detect and track conventional commercial and military aircraft, is often difficult to detect for UAV echoes and radar equipment is costly. Radio frequency (RF) detection is susceptible to background clutter, requires the constant updating of threat signal signature libraries, and is ineffective against autonomous drones. In terms of the method for detecting UAVs using sensors on UAV platforms, photoelectric detection is the most suitable. Photoelectric detection has the following advantages: First, the UAV platform has a limited load, and the visual sensor is cheap and lightweight; second, most UAV products

carry visual sensors themselves, which makes them easy to develop and realize; third, the visual sensor can provide rich information about the surrounding scene for its own UAV; and fourth, the visual sensor has a low power consumption, which is conducive to the endurance of the UAV platform. Thanks to the development of deep learning-based target detection algorithms, such as SSD [9], YOLO [10–12], VisionTransfomer [13], etc., and the maturity of countermeasure programs for countering drones with drones [14], a detection system consisting of a drone with an on-board camera and a computer that autonomously detects an intruding drone using a deep learning method has been proven to be a feasible drone detection program.

Although deep learning algorithms have solved many challenges in computer vision, there are still many problems in UAV detection tasks based on airborne vision, including that UAVs are usually imaged as weak targets in airborne vision at long distances, the background environment of airborne vision is complex and changeable and the balance between the accuracy and real-time performance of UAV target detection in airborne vision is difficult to achieve.

In order to solve the problem of UAV imaging in airborne vision with weak and small targets that are difficult to detect and have a large number of model parameters, this paper proposes an airborne vision-based UAV target detection network, ATA-YOLOv8, derived from YOLOv8. The main contributions of this paper can be summarized as follows:

- A detection layer for small targets is added. In the YOLOv8 network, the feature inputs are downsampled by the backbone (B1-B5), FPN [15] (P2-P5) and PAN [16] (N2-N4) into five scale features, respectively. Compared to the traditional method of adding P2 and N2 layers, this paper uses the B1 feature layer to enrich the features with small target information after the CSDE(cross stage detail enhance block), making the effective receptive field more dense, in favor of small target feature learning, and then introduces OmniKernel [17] to combine the features from the CSDE with the features from the P2 layer of the neck network. The OmniKernel module contains global branching, large branching and local branching, which can efficiently learn the feature representations from global to local, and ultimately improve the detection performance of small targets.
- A feature fusion module, CSP_PMSA (cross-stage-partial_partial multi-scale feature aggregation), is proposed. In this paper, we combine the ideas with GhostNet [18] and FasterNet [19] and use the efficient PartialConv to extract multi-scale feature information from the input only partially in the channels, which improves the computational efficiency. In the CSP_PMSA module, the final 1×1 convolutional layer merges the features of different scales together, and also uses residual joins to sum the input features and the processed features, effectively preserving the original information and introducing new multi-scale information, thus improving the expressive power of the model.
- The detail enhanced rep shared convolutional detection head (DERS) is proposed. The DERS uses shared convolution to realize the lightweight design of the detection head, and in order to compensate for the negative impact of shared convolution on the accuracy, it also uses the reparameterizable convolution Diverse Branch Block (DBB) [20] to introduce more learnable parameters. The reparameterizable convolution is indistinguishable from the normal convolution in the inference stage, allowing the detection head to achieve a smaller number of parameters and requiring less computation to reduce the loss of accuracy. At the same time, DERS introduces DEconv [21], which enhances the representation of detailed features of the network, and in order to solve the problem of the inconsistent scale of the target detected by each detection head, DERS uses the scale layer to scale the features.

2. Related Works

2.1. Small Object Detection

Small object detection is a challenging task. According to MS COCO [22], an object is defined as small if it occupies an area equal to or less than 32 × 32 pixels. The AI-TOD [23] benchmark refines this definition and categorizes objects as "very tiny" if they span 2 to 8 pixels, "tiny" if they span 8 to 16 pixels, and "small" if they span 16 to 32 pixels.

In recent years, many researchers have proposed various methods to solve the problem of detecting small targets in images. The main obstacle of small target detection is the loss of features in the feature extraction process. Many existing small target detection methods are based on the mainstream target detection network model improvement; the main methods are data enhancement, multi-scale feature fusion and the anchor frame mechanism. The data enhancement method is to copy, paste or add additional image data through a certain scale matching strategy for small targets in the data image, etc., which helps to improve the robustness of the model to a certain extent, and solves the problems of inconspicuous small target visualization features and little target information. For example, Kisantal et al. [24] copied and randomly pasted small objects into the image, which significantly improved the detection accuracy of small objects. Multi-scale feature fusion methods utilize the high-resolution features at the bottom of the network and the strong feature semantic information at the top, so that the small object detection accuracy can be improved to some extent. For example, FPN [15] enhances feature representation by integrating low-level details and high-level semantic features. Tan et al. [25] proposed the BiFPN (bidirectional feature pyramid network), which realizes the bidirectional cross-scale connectivity of features, where features are propagated in top-down and bottom-up paths between different layers and weighted fusion is performed. The anchor frame mechanism is designed to favor small target detection by the anchorless strategy and the combination of with and without anchor frames, which improves the detection accuracy to a certain extent. For example, the FSAF [26] is based on Retina Net and introduces two additional convolutional layers at the head of each feature map layer of the network, which are responsible for the classification and regression prediction of the anchor-free mechanism, respectively.

2.2. UAV Detection

In the UAV detection task based on airborne vision, UAV targets are usually imaged as weak and small targets. Weak and small targets generally do not have good shape and texture characteristics, with some noise effects and interference from complex backgrounds, making weak and small targets detection a difficult research task.

Zhai et al. [27] proposed the YOLO-Drone network that introduces a high-resolution detection head with the pruning of redundant layers and employs the SPD-Conv and GAM Attention mechanism to solve the problem of small UAV target size and the complex environment in which it is located. Zhou et al. [28] proposed an intruder UAV detection network, designed using ResNet, an SPPS module and the Spatial Attention Module to improve the accuracy, and used the model compression technique while speeding up the inference. Cheng et al. [29] built a dataset containing more than 10,000 low-altitude UAV images and proposed a new feature extraction network and feature fusion part based on YOLOv5 to improve the detection accuracy and reduce the model complexity. Saribas et al. [30] presented a UAV detection and tracking algorithm that combines an appearance-based detector, YOLOv3, and a kernel correlation filter (KCF)-based tracker. They use the detector to enhance the tracker performance by acting as a localization initializer and self-correction mechanism whenever the tracker loses the target. However, this algorithm is mainly suitable for UAVs that account for a large number of image pixels and the YOLOv3 algorithm requires significant computational resources. Dadboud et al. [31]

proposed a YOLOv5-based approach for UAV detection and classification that integrates Mosaic data augmentation and PAN to enhance the model's ability to detect small objects. The model is trained using publicly available air-to-air datasets to increase the number of samples for small targets. Ashraf et al. [32] proposed a two-stage detector based on the segmentation method to extract the spatio-temporal information of UAVs to solve the problem of the extremely small size and complex motion of target UAVs, but the number of parameters of the method is too large to run on edge devices such as on-board computers. Transvisdrone [33] further improves the accuracy and speed of UAV detection. However, the model size of Transvisdrone is 917M, which is not affordable for edge devices with low memory capacity. Jiang et al. [34] presented the DRBD-YOLOv8 model, an enhanced version of YOLOv8 that incorporates depthwise separable convolutions, RCELAN modules, BiFPN architecture and the DN-ShapeIoU loss function to improve anti-UAV detection accuracy and efficiency while maintaining a lightweight design. Zhang et al. [35] proposed the RTSOD-YOLO model, an improved version of YOLOv8, incorporating adaptive spatial attention, redundant feature generation and scale-sequence feature fusion to enhance UAV detection accuracy and efficiency.

In summary, existing UAV detection methods cannot strike a satisfactory balance between accuracy, model size and computational cost. For example, in the case of SAG-YOLOv5s [36], on NVIDIA RTX 2070 SUPER, when the input image size is 96 × 96 pixels, the speed is only 15 fps and the model size is 15 MB. This approach is impractical for deployment on edge computing devices. Therefore, this study aims to develop a model that balances high accuracy and light weight.

3. Materials and Methods

3.1. ATA-YOLOv8

The ATA-YOLOv8 detection network proposed in this paper is improved based on the YOLOv8n model, which improves the original model's backbone, neck and head, respectively. In the YOLOv8 network, the feature inputs are divided into five classes of scale features by the backbone (B1-B5), feature pyramid networks [15] (P2-P5) and path aggregation network [16] (N2-N4), respectively. For the backbone network, a CSDE (cross stage detail enhance block) module is designed to preserve the small target features while downsampling to obtain features enriched with small target information (B1-P2 and B2-P3). For the neck network, this paper proposes a feature fusion module CSP_PMSA (cross-stage-partial_partial multi-scale feature aggregation), which adopts the efficient PartialConv idea to extract the multi-scale feature information from the input and improve the computational efficiency. New multi-scale information is introduced, which improves the expressive ability of the model. Meanwhile, OmniKernel is designed to fuse the features from layers B1 and B2 of the backbone network and P2 and P3 of the neck network, respectively. This feature fusion module contains a global branch, large branch and local branch, which can effectively learn the feature representations from global to local and, thus, ultimately improve the detection performance of small targets. For the head network, the P2 detection layer for small targets is added, the P5 layer is deleted, and we use the idea of shared convolution to design the DERS (detail enhanced rep shared convolutional detection head). The DERS uses reparameterized convolution so that the network can extract features from the images more efficiently, and the reparameterized convolution can greatly improve the parameter utilization, allowing the detection head to achieve a smaller number of parameters and less computation, improving the detection head's ability to capture details and reducing the loss of accuracy. In order to solve the problem of inconsistency in the scale of the target detected by each detection head, the features are scaled using the scale layer. The overall structure of the network is shown in Figure 1. The ATA-YOLOv8 model significantly improves the

computational efficiency while maintaining high detection accuracy through innovative network design.

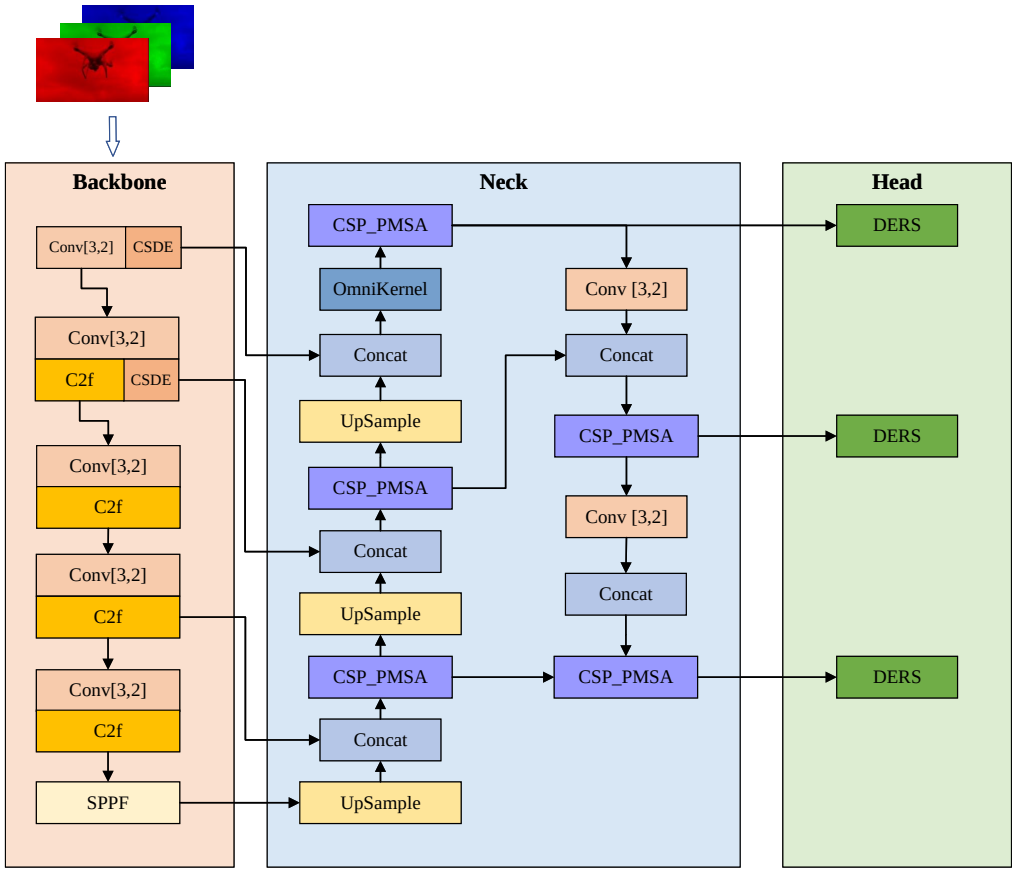


Figure 1. The framework of the ATA-YOLOv8.

3.2. Cross Stage Detail Enhance Block

The way features are extracted in a convolutional neural network is by using a convolutional layer and a pooling layer, where the convolutional layer performs a convolution operation on the input image by sliding a convolution kernel (filter) to extract local features. CNNs typically contain multiple stacked convolutional layers, each of which can have multiple convolution kernels to form a multi-channel feature map. As the depth of the network increases, the hierarchy of features that can be learned becomes more complex. However, in tasks where the image resolution is low or the objects are small, the performance of the network degrades rapidly. Because downsampling is mainly achieved through the implementation of stride convolution and pooling layers, especially at the front end of the backbone network, and when the image is blurred or the objects are small, the design of the stride convolution leads to the loss of fine-grained information, which, in turn, affects feature learning.

From the perspective of the receptive field, a smaller receptive field for the feature expression of small targets, when the receptive field is larger than the target size, i.e., the distance between two receptive field centroids is larger than the target size, the center of the receptive field may not be able to hit the target, and feature learning can not be effectively carried out. In [37], the LFFD reasonably assigns the detection tasks of different scale faces according to the relationship between the face scale and receptive field. For small-scale faces, small receptive fields are utilized for fine detection; for large-scale faces, the overall features are captured by branches of larger receptive fields. Therefore, to effectively learn the target features, it is necessary to increase the number of network layers as much as

possible before the network receptive field reaches a certain size, and to introduce more learnable parameters.

In order to solve the problem of the network's difficulty in small target feature extraction due to the use of strided convolution for downsampling, this paper designs the cross stage detail enhancement block module (CSDE), which realizes the small target by processing the feature maps of the backbone network's B1 and B2 layers that are rich in small target information and fusing them with features of the neck network's P2 and P3 layers to achieve small target detail feature enhancement, and its structure is shown in Figure 2.

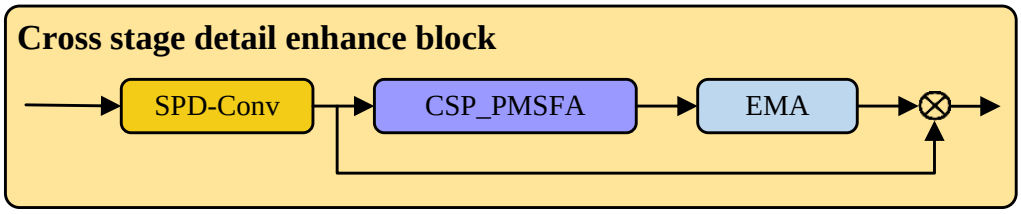


Figure 2. The structure of the cross stage detail enhance block.

The CSDE module introduces SPD-Conv [38]. SPD-Conv [38] is comprised of a space-to-depth (SPD) layer followed by a nonstrided convolution layer. Suppose there is a feature map X with dimensions $S \times S \times C_1$, where S is the height and width of the feature map X, and C is the number of channels, and the SPD layer splits the feature map X into multiple sub-feature maps. Specifically, it selects pixel blocks on the feature map according to a certain step size "$scale$" and reorganizes these blocks into new sub-feature maps. For each sub-feature map f_{xy}, it is actually a sampling of a $scale \times scale$ region of the original feature map X. The SPD layer converts the original feature map X into an intermediate feature map X' with size $\frac{S}{scale} \times \frac{S}{scale} \times scale^2 C_1$. In this way, each sub-feature map is downsampled by a factor of $scale$ but retains all the relevant channel information. The feature map segmentation method enables SPD-Conv to maintain small target feature information while efficiently downsampling the feature maps. After the SPD layer, a convolutional layer with a convolutional kernel size of 1, a stride of 1 and a number of C_2 (where $C_2 < scale^2 C_1$) is added to further transform the feature map $X'(S \times S \times C_1)$ to $X''(S \times S \times C_1)$.

In order to enhance the model context information modeling ability and feature extraction ability, in this paper, we introduce the Efficient Multi-Scale Attention Module [39] (EMA). The EMA's structure is shown in Figure 3. Thus, features of different scales are captured through the parallel subnetwork structure, which helps the model to learn short-range and long-range dependencies at the same time. At the same time, the EMA introduces a cross-space learning method, which captures pixel-level pairwise relationships through matrix dot product operation and highlights the global context of all pixels, which helps to improve the semantic information of the feature map. Consider the input feature map $X \in R^{C \times H \times W}$, where C is the number of channels, H is the height and W is the width of the input tensor. The EMA module learns multi-scale spatial features through three branches, including two 1×1 branches and one 3×3 branch. Two 1D global average pooling operations are used in the 1×1 branch to encode the channel along two spatial directions, respectively. After the output of the 1×1 convolution is decomposed into two tensors, two sigmoid nonlinear activation functions are used to fit the two-dimensional binomial distribution after the linear convolution. In the 3×3 branch, only one convolution with a kernel size of 3×3 is used to obtain the multi-scale feature representation, which captures the local cross-channel interaction and expands the feature space. Then EMA implements cross-channel feature interaction by aggregating the two channel attention maps within each group. In the cross-spatial learning part, 2D global average pooling is used to encode

the global spatial information of the output of the three branches. This helps the model understand the relative importance between different channels. The outputs of the 1×1 branch and the 3×3 branch are then captured using matrix dot product to capture the interaction between the two branches to aggregate spatial information at different scales. Subsequently, the two spatial attention weights are combined via a sigmoid function to produce an output feature map for each group, the two are added together and the sigmoid function is applied and, finally, multiplied with the input feature map. This finally results in a feature map that incorporates multi-scale information, which both contains global context and retains precise spatial location information.

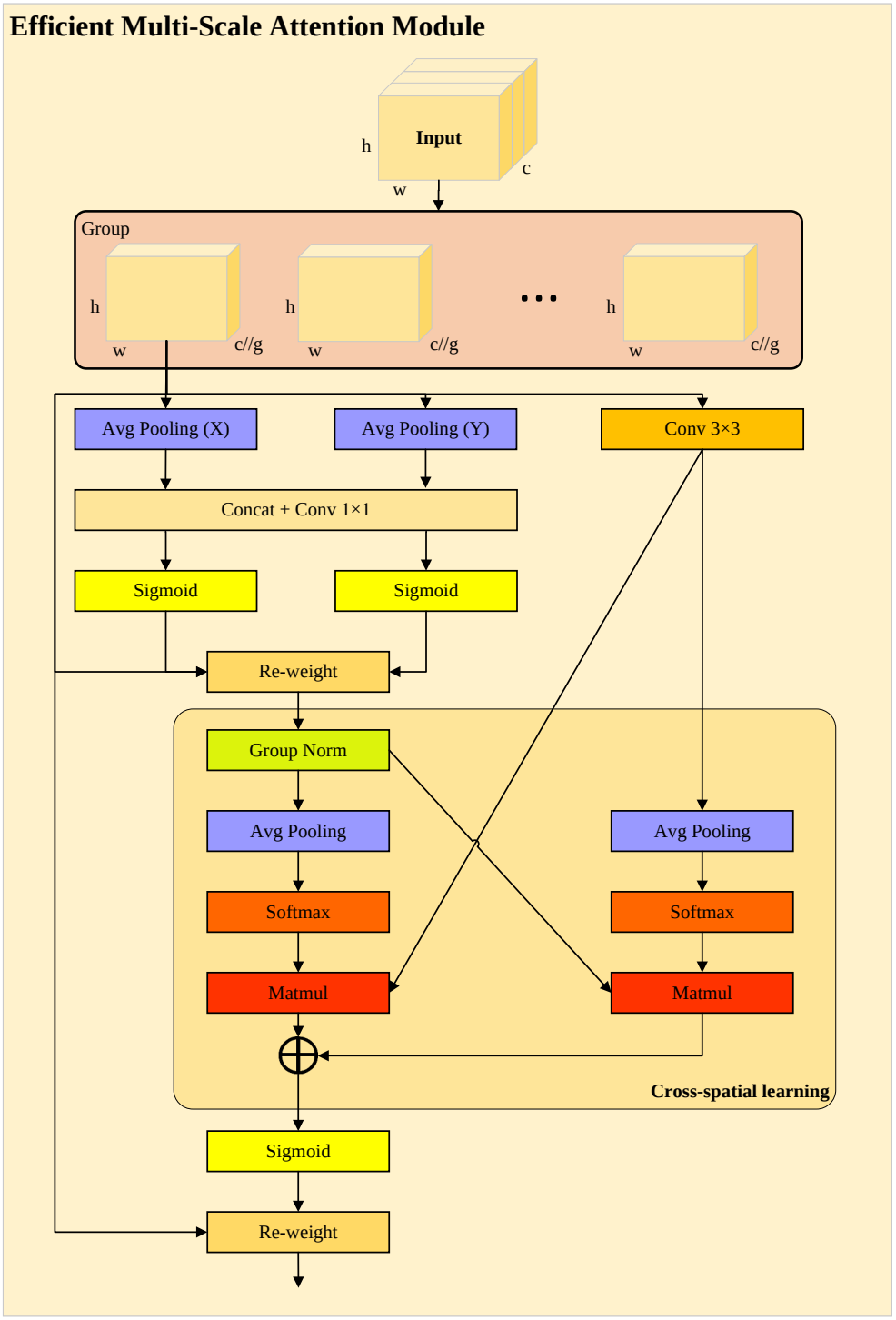


Figure 3. The structure of the Efficient Multi-Scale Attention Module.

The CSDE module retains the small target details through SPD-Conv [38], retains the precise spatial location information by applying the EMA [39] for context modeling and then multiplizes the weight information generated by the EMA [39] with the feature map of the corresponding scale to generate the output feature map.

3.3. Cross-Stage-Partial_Partial Multi-Scale Feature Aggregation

In the YOLOv8 network, the CSP (cross-stage-partial) module can fuse features from different layers, effectively combine high-level semantic information and low-level detail information and improve the detection accuracy. However, this module is designed for general scenarios, and its performance is limited in the task of UAV detection with a complex background and small target. Chen et al. [19] proposed the idea of PartialConv, which only convolves part of the feature map. PConv enhances the feature representation through partial convolutions and point conv (PWConv). The information of channels not involved in the convolution operation is retained, which helps to better capture the characteristics of small targets. PConv selects a part of the input channels for convolution operation, which reduces the amount of calculation while maintaining the spatial structure of the feature map. Compared with the traditional full convolution, PConv has a lower number of floating-point operations (FLOPs), which helps to improve the computational efficiency. Han et al. [18] proposed that there are strong correlations between feature maps in deep neural networks, i.e., there are redundant feature maps in neural networks, and some of these feature map pairs can be approximately obtained by performing cheap operations on the other. Therefore, operating on only part of the feature maps is beneficial to generate more feature maps with strong correlation and fully reveal the information in the intrinsic features. Combining the ideas of GhostNet [18] and FasterNet [19], this paper proposes a feature fusion module, CSP_PMSA (partial multi-scale feature aggregation). On the premise of ensuring the model's light weight and computational efficiency, the module extracts multi-scale feature information from the input and carries out context information modeling, so as to enhance the information of small targets and further improve the detection performance of UAV.

Figure 4 shows the module structure diagram of CSP_PMSA and PMSA. Suppose the input feature map is $X \in R^{C\times H\times W}$, where C is the number of channels, H is the height of the input tensor and W is the width of the input tensor. Firstly, after the grouped convolution of $k = 3, g = C$, X_1 and X_2 in the Figure 4 are obtained, where X_1 inputs the grouped convolution of $k = 5, g = \frac{C}{2}$, obtains the feature maps X_3 and X_4, inputs the grouped convolution of $k = 7, g = \frac{C}{4}$ and obtains X_5. Then, X_2, X_4 and X_5 are spliced along the channel dimension, and feature fusion is carried out through the convolution layer of $k = 1$, and then, the output feature is obtained by adding with the input feature map.

The partial multi-scale feature aggregation (PMSA) module extracts feature information of multiple scales from the input, and only performs it in part of the channels, thereby improving the computational efficiency and reducing the amount of calculation. The final 1×1 convolution layer fuses features of different scales together. At the same time, the residual connection is used to add the input features and the processed features, which effectively retains the original information and introduces new multi-scale information, so as to improve the expression ability of the model.

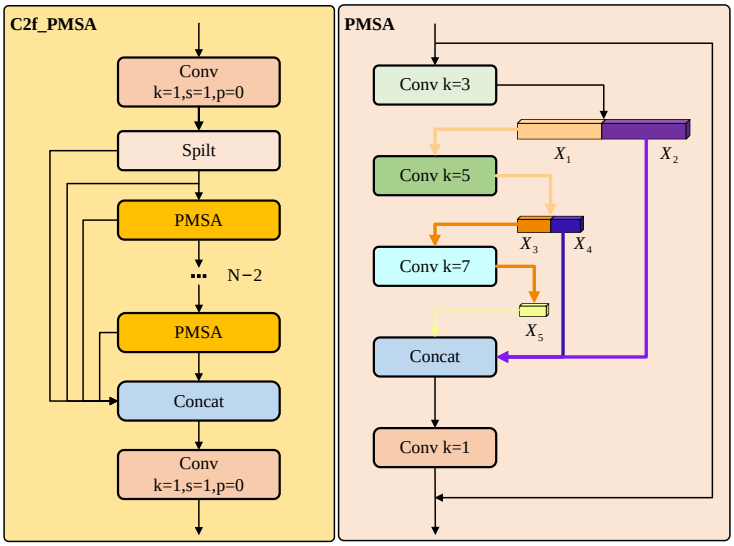


Figure 4. The structure of cross-stage-partial_partial multi-scale feature aggregation.

3.4. Omni-Kernel Module

In the UAV detection task based on airborne vision, the image of the UAV is a small target. When the target moves along the relative direction of the camera, the target will be blurred or shaken. Such motion blur will cause image quality degradation and make UAV more difficult to detect. The purpose of image restoration is to restore clear images from low-quality images. The core of the transformer's outstanding performance in the field of image enhancement is its global perception ability and strong remote dependence modeling ability. Cui et al. [17] proposes an effective convolutional network for image restoration. The OKM module is proposed on the CNN method, which can make the model have global perception ability. It is a solution to use the context relationship to model to improve the detection effect of small objects. Therefore, this study introduces the Omni-Kernel Module (OKM) in the neck network, which fuses the features from the B1 layer of the backbone network and the N2 layer of the neck network. It can effectively learn the feature representation from global to local, and finally improve the detection performance of small targets. Its structure is shown in Figure 5.

Suppose the input feature is $X \in \mathbb{R}^{C\times H\times W}$. After the convolution processing of $k = 1$, the feature is input into three branches, namely, the local branch, large-scale branch and global branch. Then, the results of the three branches are fused by addition, and the feature fusion is performed by another convolution with $k = 1$. In the large-scale branch, depthwise separable convolutions with kernel sizes of 63 × 63, 63 × 1 and 1 × 63 are used to obtain contextual features. The global branch consists of a dual-domain channel attention module (DCAM) and a frequency-based spatial attention module (FSAM). Given the input feature $X_{Gobal} \in \mathbb{R}^{C\times H\times W}$, the DCAM first applies frequency channel attention (FCA) to X_{Gobal}:

$$X_{\mathrm{FCA}} = \mathcal{IF}(\mathcal{F}(X_{Global}) \otimes W_{1\times 1}^{\mathrm{FCA}}(\mathrm{GAP}(X_{Global}))) \quad (1)$$

where $\mathcal{IF}$ and $\mathcal{F}$ are the fast Fourier transform and its inverse operation; X_{FCA}, $W_{1\times 1}^{\mathrm{FCA}}$ and GAP denote the output of FCA, the output of 1 × 1 convolutional layer and the output of global average pooling, respectively. $\otimes$ denotes the element-wise multiplication operation. The global features are effectively refined by Fourier transform processing. After global

feature extraction in the spectral domain, the obtained features are further sent to the spatial channel attention module (SCA):

$$X_{\mathrm{DCAM}} = X_{\mathrm{FCA}} \otimes W_{1\times1}^{\mathrm{SCA}}(\mathrm{GAP}(X_{\mathrm{FCA}})) \tag{2}$$

where X_{DCAM} is the output of $DCAM$. Then, a frequency-based attention module is applied in the spatial dimension to refine the spectral features:

$$X_{\mathrm{FSAM}} = \mathcal{IF}(\mathcal{F}(W_{1\times1}^{1}(X_{\mathrm{DCAM}})) \otimes W_{1\times1}^{2}(X_{\mathrm{DCAM}})) \tag{3}$$

Partial branches extract local features by using depthwise separable convolutions.

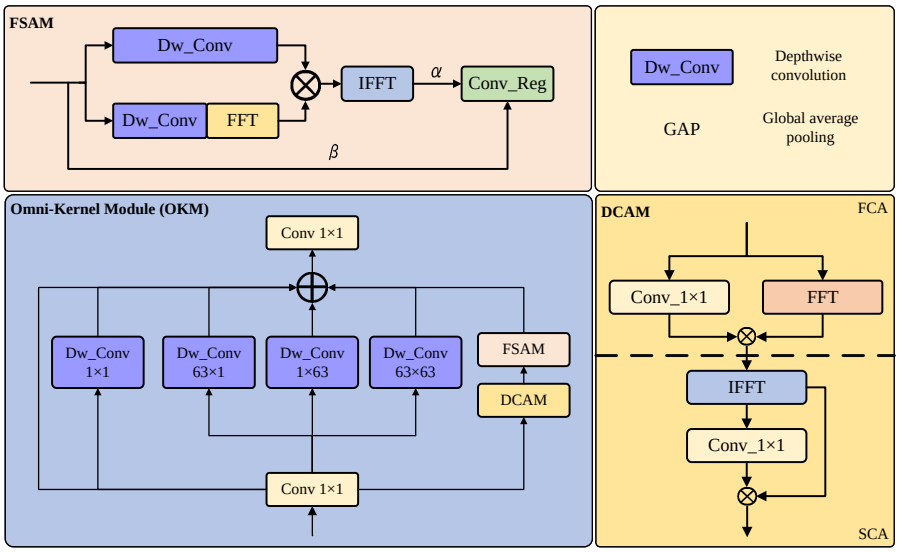


Figure 5. The structure of Omni-Kernel Module.

3.5. Detail Enhanced Rep Shared Convolutional Detection Head

In order to improve the accuracy of UAV small target detection, the traditional approach is to add a P2 detection layer to improve the detection ability of small targets, but at the same time, it will also bring a series of problems, such as excessive calculation after adding the P2 detection layer, and longer post-processing time. Compared with only adding the P2 detection layer, this paper removes the P5 detection head, and uses the idea of shared convolution to design the Detail Enhanced Rep Shared Convolutional Detection Head (DERS). Its structure is shown in Figure 6. In In order to reduce the impact of shared convolution adopted to achieve light weight, DERS uses reparameterized convolution Diverse Branch Block [20] (DBB) to introduce more learnable parameters to improve the detection accuracy of weak and small UAV. The reparameterized convolution can greatly improve the parameter utilization, which is the same as the ordinary convolution in the inference stage, so that the detection head can achieve fewer parameters and less calculation, improve the detail capture ability of the detection head and reduce the loss of accuracy. At the same time, DEConv [21] is introduced, so that the network can extract the detailed features of small targets more effectively. In order to cope with the problem of inconsistent object scales detected by each detection head, the scale layer is used to scale the features.

In the literature [40], Tian et al. experimentally demonstrated that group normalization can improve the performance of detection head localization and classification, and the proposed FCOS algorithm improves the detection performance of small targets when group normalization is used in the detection head. Additionally, adding group normalization

(GN) to the convolutional layers (except the final prediction layer) makes the training process more stable; it reduces the internal covariate shift (ICS) by dividing the channels into groups and normalizing each group to improve the stability and performance of training. In DERS, the convolution before the parameter sharing module uses GN as the normalization method.

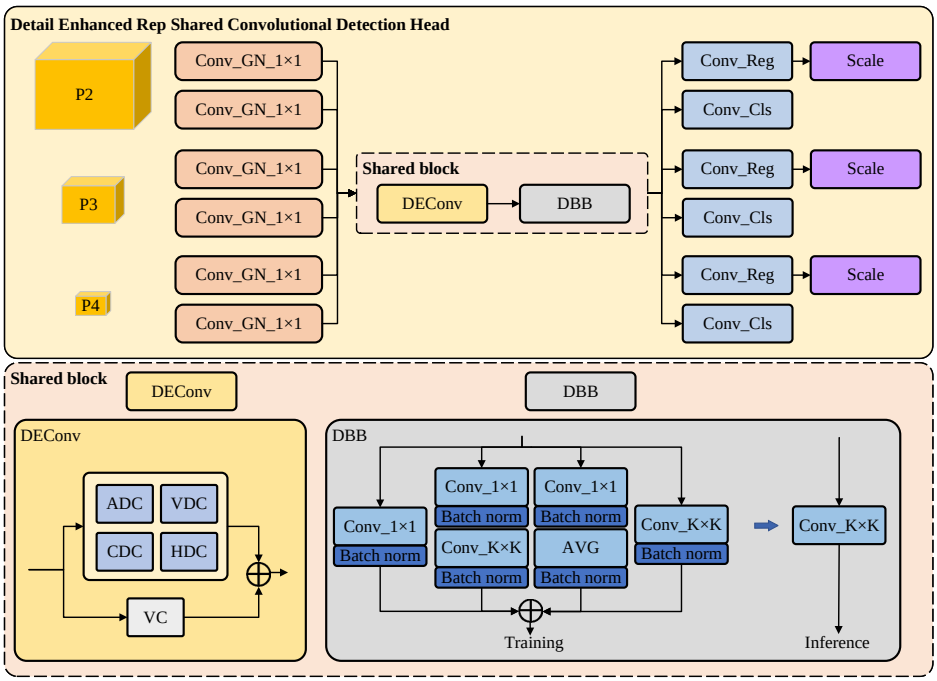


Figure 6. The structure of the Detail Enhanced Rep Shared Convolutional Detection Head. The shared block in this figure shows the structure. VC, ADC, CDC, VDC and HDC represent the five parallel convolution layers, which are the standard convolution, angle differential convolution, center differential convolution, vertical differential convolution and horizontal differential convolution, respectively.

4. Experimental Results and Analysis

4.1. Datasets and Parameter Settings

In this paper, we use MOT-FLY [41] and Det-Fly [42] datasets; the total number of images of MOT-FLY [41] reaches 11186, the image resolution is 1920 × 1080 pixels and each image contains 1 to 3 micro Uavs. MOT-FLY [41] covers different scenes, backgrounds, UAV motion states and distances. The dataset contains four different types of environment settings: sky (11.36 %), plain (12.26 %), village (35.78 %) and city (40.58 %). In addition, different viewpoints (top view (37.82 %), forward view (32.76 %), down view (29.43 %)) and lighting conditions (day, night and dusk) are considered in this dataset. Its original annotation format is mot format, which is converted into a suitable format for the detection task in this paper. Regarding Det-Fly [42] data, the total number of images is more than 13,000 and each image is 3840 × 2160 pixels. Compared with the existing air-to-air datasets, the design of the dataset is more systematic, covering different actual scenes, different viewpoints, relative distances, flight heights and brightness conditions. The relative distance of the target UAV is from 10 m to 100 m, and the flying altitude is from 20 m to 110 m. Since the brightness condition is also an important factor for UAV detection, the time of data collection is at different time periods of the day. The dataset also covers some challenging scenes, for example, bright light, motion blur and partial object occlusion.

In this paper, the original dataset preset ratio is used to divide the training, validation and test sets. For hardware setup, the experiments in this paper are performed based on the ubantu20.04 server with Intel Core i9-11900KF, Nvidia GeForce RTX 2080Ti 11G VRAM

and 32GB RAM. The Nvidia GeForce RTX 2080Ti is a highly arithmetic card that can be used to run deep learning algorithms at a relatively low cost. We used PyTorch 1.10.0 and Torchvision 0.11.1 as deep learning frameworks, improved based on Ultralytics YOLOv8 version 8.2.50 and trained with official pre-training weights. Choosing the appropriate batch size and learning rate is a very important hyperparameter tuning task in deep learning, and directly affects the training efficiency and convergence performance of the model. Considering the influence of GPU memory and other factors, we experimentally verify different combinations to find the most suitable parameters for the current task. Table 1 details the main parameter settings used during training.

Table 1. The main parameter settings.

Parameters	Setup
Epochs	300
Batch size	8
Workers	8
Imgsz	640
Optimizer (SGD)	0.7
Initial learning rate	1×10^{-2}
Final learning rate	1×10^{-4}
Momentum	0.937
Mosaic	1
Warmup epochs	10
IoU	0.7

4.2. Ablation Experiment

In order to verify the effectiveness of the improved modules on model detection performance, this paper conducted a series of ablation experiments using motfly to systematically verify the effectiveness of each module. This paper uses YOLOv8n as the base model and evaluates the effects of different modules on detection under the same experimental conditions, and the results of the relevant ablation experiments are shown in Table 2.

Table 2. Ablation experiments.

Model	mAP50	P	R	mAP0.95	Params (10^6)	GFLOPs
YOLOv8n	0.699	0.883	0.589	0.461	3.01	8.2
−P5	0.675	0.863	0.543	0.436	1.5	6.4
+P2	0.901	0.953	0.813	0.604	2.01	15.2
A	0.922	0.95	0.845	0.621	2.13	15.8
B	0.923	0.957	0.849	0.635	1.63	12.9
C	0.926	0.965	0.841	0.639	1.59	13.3
D	0.921	0.959	0.836	0.617	1.6	10.5
A + B	0.925	0.967	0.842	0.618	2.05	13.3
A + C	0.928	0.967	0.856	0.624	2.01	13.8
A + D	0.926	0.945	0.854	0.637	2.11	13.6
B + C	0.927	0.963	0.845	0.629	2.01	14.2
B + D	0.93	0.963	0.851	0.627	2.14	14.5
C + D	0.927	0.976	0.839	0.625	1.9	13
B + C + D	0.934	0.954	0.858	0.633	2	12.8
A + C + D	0.934	0.972	0.843	0.643	2.14	13.5
A + B + D	0.931	0.965	0.846	0.633	2.12	13.7
A + B + C	0.931	0.965	0.846	0.633	2.15	14.8
A + B + C + D	0.949	0.964	0.876	0.644	2.13	12.3

As shown in Table 2, A stands for CSDE(cross stage detail enhance block), B stands for CSP_PMSA(cross-stage-partial_partial multi-scale feature aggregation), C stands for OmniKernel and D stands for DERS(detail enhanced rep shared convolutional detection head). In this paper, model performance is evaluated using the metrics of precision, recall, mAP50, mAP95, parameters and GFLOPs. Figure 7 highlights the mAP50 and GFLOPs. Each group of experiments has improved the model performance, and the experimental results are analyzed next.

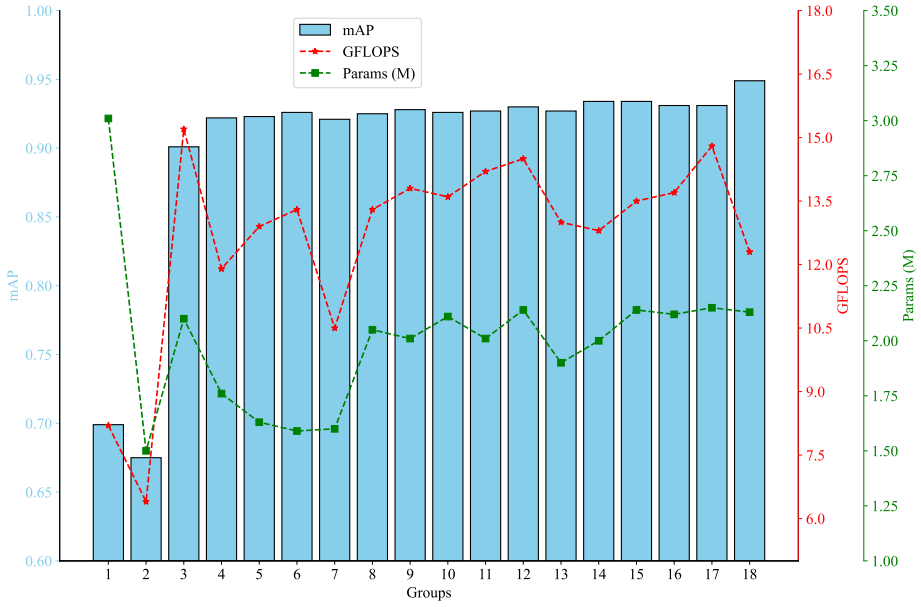


Figure 7. The mAP50, GFLOPs and parameters of ablation experiments.

The mAP50 of the YOLOv8n model is 0.699, but its computational amount is 8.2GFLOPs, because the UAVs in MOT-FLY dataset are mostly small targets and the background complexity of the scene accounted for a large proportion, so the YOLOv8n official algorithm is not good. After subtracting the P5 layer, which is the detection layer for large targets, the mAP50 decreases to 0.675, and the computational amount decreases to 6.4GFLOPs, and the number of model parameters decreased by 50%. On this basis, the model adds the P2 detection layer for small targets, and the mAP50 increases to 0.901, which greatly improves the detection accuracy, which is due to the fact that the small target detection layer can effectively learn the small target features, but the model parameter number and computation volume both increase, and the number of parameters increases to 2.01 M, and the computation volume is 11.5GFLOPs. It can be seen from the results of Experiment A that the CSDE module downsampled the B1 and B2 layers of the backbone network, obtained the feature map rich in small target information and fused the features with the N2 and N3 layers of the neck network to realize the enhancement of small target detail features.

As can be seen from Experiments 4–7, the mAP is improved by 2.1%, 2.2%, 2.5% and 2% (compared to Experiment 3, on which all the following comparisons are based) and the number of parameters is reduced by 0.25 M, 0.38 M, 0.42 M and 0.41 M, respectively, with the addition of the corresponding modules, thus verifying the effectiveness of the designed modules for accuracy improvement and lightweighting. The mAP95 of Experiment 2 and Experiment 3 were improved by 3.1% and 3.5%, respectively, highlighting the superior ability of the design of the CSP_PMSA, OKM module with a large convolutional kernel in enhancing the contextual associations between distant pixels and optimizing the representation of the central features, especially when dealing with the detection of small targets in

complex environments and demonstrating excellent localization accuracies. In summary, the design of the CSP_PMSA, OKM module not only enhances the computational efficiency, but also significantly improves the small target detection performance of the model.

As can be seen from Experiments 8–17, after adding the corresponding modules, the mAP50 is improved, but the parameter quantities and GFLOPs are unstable. In Experiment 18, compared with baseline, the mAP is improved by 25%, th emAP95 is improved by 18.3%, the parameter quantity is reduced by 0.88 M and the GFLOPS are reduced by 2.9, which achieves the accuracy improvement while lightweighting the model and enhances the model's ability to detect the small targets. The CSDE module retains the small target information while downsampling the model. The CSDE module retains small target information while performing downsampling, while using EMA to process multi-scale information, enhancing the model's ability to model using global contextual information and feature extraction. CSP_PMSA extracts multi-scale feature information from inputs for contextual information modeling while ensuring the model's lightweighting and computational efficiency to enhance small target information and further improve the UAV's detection performance. The Omni-Kernel Module (OKM) can efficiently learn feature representations from global to local to improve the detection performance of small targets. The Detail Enhanced Rep Shared Convolutional Detection Head (DERS) uses the idea of shared convolution to make the number of parameters smaller and less computationally intensive, but improves the ability to enhance the detection head detail capture capability.

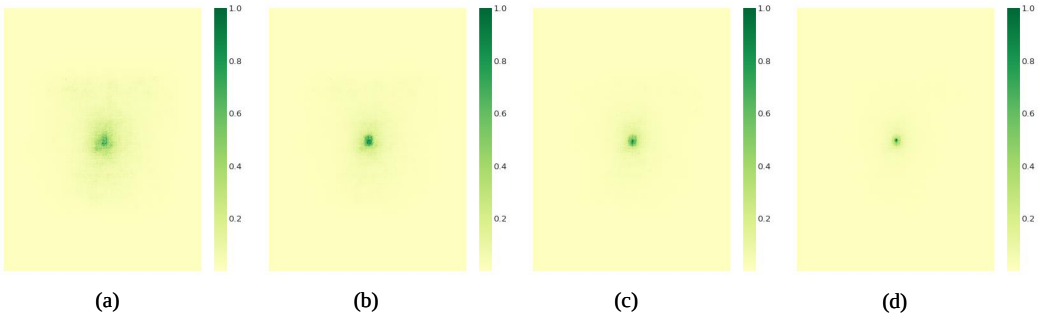
In order to investigate the enhancement effect of the CSDE module on different scale features, this paper applies CSDE to the backbone network layers of B1, B2 and B3 scales as well as to B1 and B2 layers at the same time, and analyzes the experimental effects. The experimental details are shown in Table 3.

Table 3. Enhancement of CSDE's target detection performance at different scales.

Model	mAP50	P	R	Params	GFLOPs
+P2	0.901	0.953	0.813	2.01	15.2
B1	0.916	0.948	0.826	2.03	15.4
B2	0.91	0.942	0.815	2.08	15.6
B3	0.897	0.946	0.795	2.16	16.3
B1+B2	0.922	0.95	0.845	2.13	15.8

When the CSDE module is added to the B1 or B2 layer, the effect is improved, the mAP50 is increased by 1.5% and 0.9%, respectively, and the number of parameters and computation amount are increased to some extent, which is due to the fact that some learnable parameters in the CSDE module effectively retain the small target features. When adding the CSDE module to the B3 layer, there is a decrease in accuracy and a larger rise in the number of parameters, mainly due to the higher number of channels in this layer after adding the CSDE module, and, therefore, the number of parameters is also higher. When applying CSDE to layers B1 and B2 at the same time, the mAP50 improves by 2% compared to before adding the module. In order to more intuitively reflect the detection effect of CSDE on UAV targets, this paper calculates and visualizes the effective receptive fields for the 13th, 17th, 18th and 21st layers of the network, as shown in Figure 8. The dark part represents the effective receptive field area. Figure 8a, the 13th layer of the network, corresponds to the P3 layer, i.e., the feature map of the neck network with a downsampling multiple of 3; Figure 8b, the 17th layer of the network, is the feature layer of the B2 layer after downsampling and feature enhancement by the CSDE module and fusion with layer 13; Figure 8c, the 18th layer of the network, corresponds to the P2 layer, i.e., the feature map of the neck network with a downsampling multiple of 2; and Figure 8d, the 21st layer

of the network, is the feature map of the B2 layer after downsampling by the CSDE module and feature enhancement.



(a) (b) (c) (d)

Figure 8. Effective receptive fields of the 13th, 17th, 18th and 21st layers of the network, which corresponds respectively to (**a**–**d**).

From Figure 8, we can intuitively see that compared with Figure 8a, the center region of the receptive field in Figure 8b is darker; compared with Figure 8c, the center region in Figure 8d is darker, and the effective receptive field is more focused on the center region, which proves that the CSDE module in this paper can focus the effective receptive field on the center region of the anchor point. The more concentrated effective receptive field can capture the detailed features of small targets better, such as texture, shape, etc., which is more conducive to the expression of small target features. When detecting weak and small targets, the small receptive field can perceive pixel-level differences, which means that the network can perform finer-grained feature extraction on the image to locate the small targets more accurately, and thus reduce the phenomena of false detection (misjudging the background or non-target area as a target) and missed detection (failing to detect the small targets that are actually present) due to the large receptive field. Based on the above experiments, this paper concludes that CSDE is most effective when applied to B1 and B2 layers.

4.3. Parallel Testing

The core technology of UAV target detection algorithms based on airborne vision is target detection algorithms based on computer vision. Target detection algorithms can be divided into traditional target detection algorithms and deep learning target detection algorithms; traditional target detection algorithms tend to extract target features by manual design, while deep learning target detection algorithms generally learn how to extract the features of the target adaptively through the model. With the rapid development of GPU parallel computing technology, the development of deep learning algorithms is extremely rapid, and deep learning target detection algorithms have far exceeded the accuracy of traditional target detection algorithms. Therefore, this paper focuses on the comparison of deep learning target detection algorithms.

In order to verify the superiority of the algorithms, the YOLOX, YOLOv5 and YOLOv10n algorithms are tested on the MOT-FLY dataset in this paper, and the experimental results are shown in Table 4.

Table 4. Parallel testing on the MOT-FLY dataset.

Model	mAP	P	R	Params	GFLOPs
YOLOXL	0.716	0.693	0.346	99	282.03
YOLOv10n	0.708	0.857	0.589	2.71	8.4
Ours(ATA-YOLOv8)	0.949	0.964	0.876	2.13	12.3

On the Det-Fly dataset, we compare the proposed algorithm with YOLOv7, YOLOv8n, VDTNet and EDGS-YOLOv8, and the results are shown in Table 5.

Table 5. Parallel testing on the Det-Fly dataset

Model	mAP50	P	R	Model Size	GFLOPs
YOLOv7	0.888	0.881	0.865	9.7	-
YOLOv8n	0.905	0.943	0.855	6.3	8.1
VDTNet	0.948	0.921	0.949	3.9	-
EDGS-YOLOv8	0.934	0.914	0.872	4.23	-
Ours(ATA-YOLOv8)	0.964	0.98	0.924	5.1	12.3

As shown in Tables 4 and 5, ATA-YOLOv8 maintains a small model size while having the highest accuracy. Compared to the YOLOv8n algorithm, the mAP50 is improved and the number of model parameters is reduced. This demonstrates the improvements of our ATAYOLOv8 model across all metrics compared to the original model. Analyzing the technical aspects, the proposed algorithm effectively enhances the feature extraction. The CSDE module improves the feature representation of micro UAVs by preserving the small target details through SPD-Conv, focusing the receptive field with the center, improving the localization capability, reducing the background noise interference and using the EMA for global context modeling. OmniKernel enhances the detection of small targets in complex scenes by integrating features from backbone and neck layers (B1 and P2) through multi-branch processing (global, large and local). CSP_PMSA reduces computation while preserving spatial details through partial multi-scale feature aggregation that can be exploited with channel segmentation and concatenation. DERS reduces computation through shared convolution with reparameterized DBB and DEconv to reduce parameters and FLOPs without loss of accuracy, making it lighter (5.1 MB) and faster. ATA-YOLOv8 enhances robustness in complex environments, with adaptability to motion blur, occlusion and varying light/angles, which is critical for reliable performance in UAV swarm coordination, obstacle avoidance or surveillance missions. Meanwhile, the algorithm reduces FP, and context modeling and multi-scale feature extraction reduces the misclassification of background objects (e.g., trees, clouds). Moreover, the algorithm is scalable and flexible, valid for datasets such as MOT-FLY and Det-Fly, and it can be adapted to various UAV types and scenarios (civil/military).

4.4. Visualization Experiments

To visualize the effectiveness of the module proposed in this paper, the model detection layer feature maps were analyzed using HiResCAM to generate a visual heat map, as shown in Figure 9.

In Figure 9, we compare the proposed model with the YOLOv8n baseline model, and the selected comparison scenes include the scene where the UAV is a small target, the scene with complex background and the scene with low light. Figure 9b represents the original YOLOv8 algorithm, which has the problems of identifying the background as a target, focusing on a large area, containing too much background and failing to detect all the existing targets. The original YOLOv8 algorithm is easily disturbed by similar objects around it, its feature expression ability is not strong and its ability to detect small targets is poor. In the first row of Figure 9b, the YOLOv8 algorithm incorrectly identifies backgrounds such as tree branches as drones, and the CSDE module solves this problem. This is because the EMA attention mechanism in the CSDE module enhances the robustness of the model to illumination changes and complex background interference, which illustrates the superiority of our model in identifying small targets and complex

environments. The third row of Figure 9 shows the reduction in the missed detection rate, which is attributed to the small object detection layer as well as the CSDE module, which effectively preserves the small object detail features. As can be seen in Figure 9d, CSP_PMSA and the Omni-Kernel Module enhance the modeling ability of the model for the context, which is reflected in the modeling of the position relationship between the fuselage and the arm as well as the position relationship between the UAV and the background, effectively improving the positioning ability of the model for the UAV. Figure 9e shows the ATA-YOLOv8 algorithm proposed in this paper. The dark area is concentrated in the fuselage and arm part of the UAV, covering almost the entire UAV, which effectively enhances the target feature extraction, improving the detection performance.

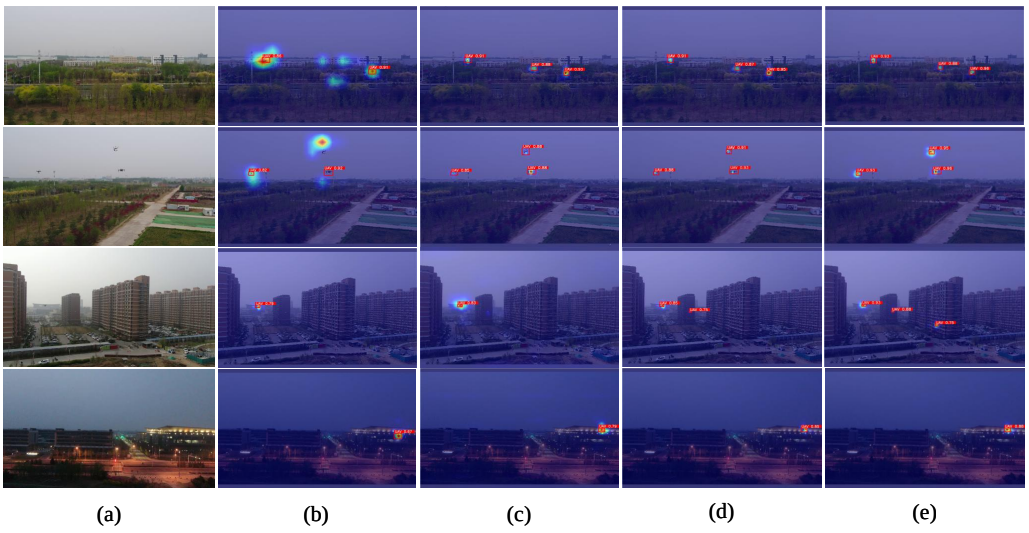


Figure 9. The heat map using HiResCAM, where (**a**) is the original image; (**b**) is the heatmap of the YOLOv8 model; (**c**) is the heatmap of the YOLOv8 + CSDE model; (**d**) is the heatmap of the YOLOv8 + CSDE + CSP_PMSA + OmniKernel model; (**e**) is the heatmap of ATA-YOLOv8 model.

To reflect the enhancement of the effective receptive field by ATA-YOLOv8, this paper visualizes the effective receptive fields inputted to the detection head. The first row in Figure 10 is the effective receptive field visualization plot of the original YOLOv8, and the darker areas in the plot correspond to the effective receptive field sizes of the feature maps when the downsampling multiplicity is 8, 16 and 32, respectively. The second row is the effective receptive field visualization of ATA-YOLOv8, which corresponds to the effective receptive field size of the feature maps when the downsampling multiplicity is 4, 8 and 16, respectively. The design of adding the N2 layer is more favorable to the small target detection task, and the smaller effective receptive field is more focused on the detection of small objects. At the same scale, such as with the N3 and N4 layers of the two models, the effective receptive field map of ATA-YOLOv8 is obviously more focused on the center region than that of YOLOv8n. In the UAV detection task, the features of ATA-YOLOv8 are crucial for distinguishing the UAV from the background. Focusing the receptive field on the center can extract these key features more prominently, enhance the discriminative ability of the features and improve the accuracy of the detection.

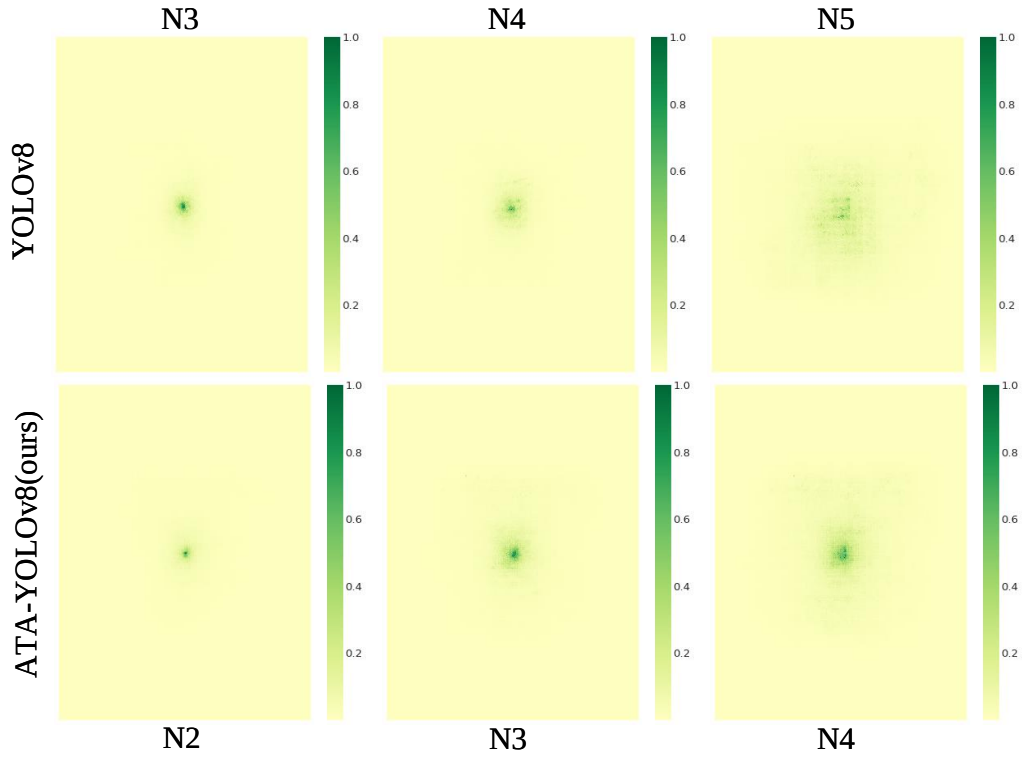


Figure 10. Visualization of the network's effective receptive fields.

5. Conclusions

Aiming at the problem that existing UAV detection methods in air-to-air scenarios are unable to achieve a satisfactory balance between the accuracy of small UAV target detection, model size and computational cost, this paper proposed the ATA-YOLOv8 UAV detection network to improve the UAV detection effect by the cross stage detail enhance block (CSDE), Omni-Kernel Module (OKM), cross-stage-partial partial multi-scale feature aggregation (CSP_PMSA) and Detail Enhanced Rep Shared Convolutional Detection Head (DERS). The CSDE module preserves the small target details through SPD-Conv, and after CSP_PMSA feature fusion, the EMA is applied for contextual modeling, which also preserves the precise spatial location information, and then, the weight information generated by EMA attention is multiplied with the feature maps of the corresponding scales to generate the output feature maps. For the neck network, a feature fusion module, CSP_PMSA, is proposed, which employs the efficient PartialConv to extract multi-scale feature information from the input, improves the computational efficiency, effectively preserves the original information and introduces new multi-scale information, thus improving the model's expression ability. Meanwhile, the Omni-Kernel Module (OKM) is designed. The OKM contains a global branch, a large branch and a local branch, which can efficiently learn the feature representations from global to local, thus improving the performance of the detection of the small targets. In order to reduce the negative impact of shared convolution taken to achieve lightweighting, DERS is designed using the idea of shared convolution, while using reparameterizable convolution, by introducing more learnable parameters. At the same time, DERS introduced DEconv, so that the network can more effectively learn from the features, improving the detection head detail capturing ability and reducing the loss of accuracy. In order to cope with the problem of inconsistency in the scale of the target detected by each detection head, the features are scaled using the scale layer.

UAV air-to-air detection algorithms also face a number of practical challenges they are deployed in real-world scenarios: Images captured by UAVs usually have special viewing angles, lighting conditions, and background noise, which cause significant variations

in the appearance and size of the target objects and increase the difficulty of detection. Low-altitude images often contain many dense small targets, which are easy to ignore especially when shooting at long distances, while the occlusion problem in dense scenes also increases the complexity of detection. Due to the limitation of hardware resources, many detection algorithms reduce the size of high-resolution images (e.g., downsizing to 320 × 320 or 640 × 640), which may result in small targets being missed in the detection process. UAVs often have limited computational power, which restricts the deployment of complex models and requires finding a balance between model accuracy and computational efficiency. UAV detection tasks usually require real-time processing, which means that the detection algorithms must complete their reasoning in a short period of time, which places high demands on the efficiency of the models.

In the future, UAV air-to-air detection algorithms will be more widely used in military and civilian fields. The UAV air-to-air detection algorithm can be applied to UAV cluster coordination and autonomous obstacle avoidance, realizing autonomous formation flight and collision avoidance in dynamic environments, which is especially suitable for logistics and distribution, agricultural plant protection and other scenarios requiring dense formation. Combined with the proposed cascade multi-target tracking algorithm, it can further support the multi-aircraft cooperative tasks in complex airspace. In sensitive areas such as airports and military bases, the algorithm can detect intruding malicious UAVs in real-time, and combined with the tracking algorithm, a fast-response countermeasure system can be constructed.

In summary, the ATA-YOLOv8 UAV detection model proposed in this paper achieves the highest accuracy on the Det-Fly and MOT-FLY datasets with a mAP50 of 0.964 and 0.949, respectively, which is 5.9% and 35.8% higher than that of the original YOLOv8n algorithm's mAP. At the same time, the size of the model is only 5.1MB, which makes it possible to deploy it on low-storage-space edge devices. It also maintains a better performance in real-time and achieves improved accuracy, making is easy to deploy the application in embedded systems and providing a feasible solution for UAV target detection based on airborne vision.

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

Funding: This work was supported by the Applied Research Advancement Project in Engineering University of PAP [grant no. WYY202304]; Research and Innovation Team Project in Engineering University of PAP [grant no. KYTD202306]; Graduate Student Funded Project [grant no. JYWJ2023B003].

Data Availability Statement: The data used in this analysis are publicly available, and access is provided in the text.

Acknowledgments: The authors would like to thank all coordinators and supervisors involved and the anonymous reviewers for their detailed comments that helped to improve the quality of this article.

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

References

1. Pan, L.; Song, C.; Gan, X.; Xu, K.; Xie, Y. Military Image Captioning for Low-Altitude UAV or UGV Perspectives. *Drones* **2024**, *8*, 421. [CrossRef]
2. Ahirwar, S.; Swarnkar, R.; Bhukya, S.; Namwade, G. Application of Drone in Agriculture. *Int. J. Curr. Microbiol. Appl. Sci.* **2019**, *8*, 2500–2505. [CrossRef]

3. Raivi, A.M.; Huda, S.M.A.; Alam, M.M.; Moh, S. Drone Routing for Drone-Based Delivery Systems: A Review of Trajectory Planning, Charging, and Security. *Sensors* **2023**, *23*, 1463. [CrossRef] [PubMed]
4. Hassanalian, M.; Abdelkefi, A. Classifications, applications, and design challenges of drones: A review. *Prog. Aerosp. Sci.* **2017**, *91*, 99–131. [CrossRef]
5. Yang, B.; Matson, E.T.; Smith, A.H.; Dietz, J.E.; Gallagher, J.C. UAV Detection System with Multiple Acoustic Nodes Using Machine Learning Models. In Proceedings of the 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 25–27 February 2019; pp. 493–498. [CrossRef]
6. Yuan, S.; Yang, Y.; Nguyen, T.H.; Nguyen, T.M.; Yang, J.; Liu, F.; Li, J.; Wang, H.; Xie, L. MMAUD: A Comprehensive Multi-Modal Anti-UAV Dataset for Modern Miniature Drone Threats. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 2745–2751. [CrossRef]
7. Xiao, Y.; Zhang, X. Micro-UAV Detection and Identification Based on Radio Frequency Signature. In Proceedings of the 2019 6th International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2–4 November 2019; pp. 1056–1062. [CrossRef]
8. Wang, B.; Li, Q.; Mao, Q.; Wang, J.; Chen, C.L.P.; Shangguan, A.; Zhang, H. A Survey on Vision-Based Anti Unmanned Aerial Vehicles Methods. *Drones* **2024**, *8*, 518. [CrossRef]
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV, Amsterdam, The Netherlands, 11–14 October 2016; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerland, 2016; pp. 21–37.
10. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
11. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
12. Wang, C.Y.; Yeh, I.H.; Mark Liao, H.Y. Yolov9: Learning what you want to learn using programmable gradient information. In Proceedings of the European Conference on Computer Vision, Nashville, TN, USA, 11–15 June 2025; Springer: Berlin/Heidelberg, Germany, 2025; pp. 1–21.
13. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
14. Goppert, J.M.; Wagoner, A.R.; Schrader, D.K.; Ghose, S.; Kim, Y.; Park, S.; Gomez, M.; Matson, E.T.; Hopmeier, M.J. Realization of an Autonomous, Air-to-Air Counter Unmanned Aerial System (CUAS). In Proceedings of the 2017 First IEEE International Conference on Robotic Computing (IRC), Taichung, Taiwan, 10–12 April 2017; pp. 235–240. [CrossRef]
15. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
16. 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.
17. Cui, Y.; Ren, W.; Knoll, A. Omni-Kernel Modulation for Universal Image Restoration. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 12496–12509. [CrossRef]
18. 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.
19. 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 (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
20. Ding, X.; Zhang, X.; Han, J.; Ding, G. Diverse branch block: Building a convolution as an inception-like unit. In Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10886–10895.
21. Chen, Z.; He, Z.; Lu, Z.M. DEA-Net: Single Image Dehazing Based on Detail-Enhanced Convolution and Content-Guided Attention. *IEEE Trans. Image Process.* **2024**, *33*, 1002–1015. [CrossRef] [PubMed]
22. 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*; Springer: Berlin/Heidelberg, Germany, 2014, pp. 740–755.
23. Wang, J.; Yang, W.; Guo, H.; Zhang, R.; Xia, G.S. Tiny Object Detection in Aerial Images. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 3791–3798. [CrossRef]
24. Kisantal, M. Augmentation for Small Object Detection. *arXiv* **2019**, arXiv:1902.07296.
25. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787. [CrossRef]
26. Zhu, C.; He, Y.; Savvides, M. Feature selective anchor-free module for single-shot object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 840–849.

27. Zhai, X.; Huang, Z.; Li, T.; Liu, H.; Wang, S. YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection. *Electronics* **2023**, *12*, 3664. [CrossRef]
28. Zhou, X.; Yang, G.; Chen, Y.; Li, L.; Chen, B.M. VDTNet: A High-Performance Visual Network for Detecting and Tracking of Intruding Drones. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 9828–9839. [CrossRef]
29. Cheng, Q.; Wang, Y.; He, W.; Bai, Y. Lightweight air-to-air unmanned aerial vehicle target detection model. *Sci. Rep.* **2024**, *14*, 2609. [CrossRef] [PubMed]
30. Saribas, H.; Uzun, B.; Benligiray, B.; Eker, O.; Cevikalp, H. A hybrid method for tracking of objects by UAVs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019; pp. 563–572.
31. Dadboud, F.; Patel, V.; Mehta, V.; Bolic, M.; Mantegh, I. Single-Stage UAV Detection and Classification with YOLOV5: Mosaic Data Augmentation and PANet. In Proceedings of the 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Washington, DC, USA, 16–19 November 2021; pp. 1–8. [CrossRef]
32. Ashraf, M.W.; Sultani, W.; Shah, M. Dogfight: Detecting Drones From Drones Videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7067–7076.
33. Sangam, T.; Dave, I.R.; Sultani, W.; Shah, M. TransVisDrone: Spatio-Temporal Transformer for Vision-based Drone-to-Drone Detection in Aerial Videos. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 6006–6013. [CrossRef]
34. Jiang, P.; Yang, X.; Wan, Y.; Zeng, T.; Nie, M.; Liu, Z. DRBD-YOLOv8: A Lightweight and Efficient Anti-UAV Detection Model. *Sensors* **2024**, *24*, 7148. [CrossRef] [PubMed]
35. Zhang, S.; Yang, X.; Geng, C.; Li, X. A Reparameterization Feature Redundancy Extract Network for Unmanned Aerial Vehicles Detection. *Remote Sens.* **2024**, *16*, 4226. [CrossRef]
36. Lv, Y.; Ai, Z.; Chen, M.; Gong, X.; Wang, Y.; Lu, Z. High-Resolution Drone Detection Based on Background Difference and SAG-YOLOv5s. *Sensors* **2022**, *22*, 5825. [CrossRef] [PubMed]
37. He, Y.; Xu, D.; Wu, L.; Jian, M.; Xiang, S.; Pan, C. LFFD: A Light and Fast Face Detector for Edge Devices. *arXiv* **2019**, arXiv:1904.10633.
38. Sunkara, R.; Luo, T. No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects. In Proceedings of the Machine Learning and Knowledge Discovery in Databases, Turin, Italy, 18–22 September 2023; Amini, M.R., Canu, S., Fischer, A., Guns, T., Kralj Novak, P., Tsoumakas, G., Eds.; Springer: Cham, Switzerland, 2023; pp. 443–459.
39. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient Multi-Scale Attention Module with Cross-Spatial Learning. In Proceedings of the ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp.1–5. [CrossRef]
40. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: A Simple and Strong Anchor-Free Object Detector. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1922–1933. [CrossRef] [PubMed]
41. Chu, Z.; Song, T.; Jin, R.; Jiang, T. An Experimental Evaluation Based on New Air-to-Air Multi-UAV Tracking Dataset. In Proceedings of the 2023 IEEE International Conference on Unmanned Systems (ICUS), Hefei, China, 13–15 October 2023; pp. 671–676. [CrossRef]
42. Zheng, Y.; Chen, Z.; Lv, D.; Li, Z.; Lan, Z.; Zhao, S. Air-to-Air Visual Detection of Micro-UAVs: An Experimental Evaluation of Deep Learning. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1020–1027. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.