drones

MDPI

Article

MFEFNet: A Multi-Scale Feature Information Extraction and Fusion Network for Multi-Scale Object Detection in UAV Aerial Images

Liming Zhou [1,2], Shuai Zhao [1,2], Ziye Wan [1,2], Yang Liu [1,2], Yadi Wang [1,2,*] and Xianyu Zuo [1,2]

[1] Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475000, China; lmzhou@henu.edu.cn (L.Z.); hedazs@henu.edu.cn (S.Z.); ziye@henu.edu.cn (Z.W.); sea@vip.henu.edu.cn (Y.L.); xianyu_zuo@henu.edu.cn (X.Z.)
[2] School of Computer and Information Engineering, Henan University, Kaifeng 475000, China
* Correspondence: yadiwang@henu.edu.cn

Abstract: Unmanned aerial vehicles (UAVs) are now widely used in many fields. Due to the randomness of UAV flight height and shooting angle, UAV images usually have the following characteristics: many small objects, large changes in object scale, and complex background. Therefore, object detection in UAV aerial images is a very challenging task. To address the challenges posed by these characteristics, this paper proposes a novel UAV image object detection method based on global feature aggregation and context feature extraction named the multi-scale feature information extraction and fusion network (MFEFNet). Specifically, first of all, to extract the feature information of objects more effectively from complex backgrounds, we propose an efficient spatial information extraction (SIEM) module, which combines residual connection to build long-distance feature dependencies and effectively extracts the most useful feature information by building contextual feature relations around objects. Secondly, to improve the feature fusion efficiency and reduce the burden brought by redundant feature fusion networks, we propose a global aggregation progressive feature fusion network (GAFN). This network adopts a three-level adaptive feature fusion method, which can adaptively fuse multi-scale features according to the importance of different feature layers and reduce unnecessary intermediate redundant features by utilizing the adaptive feature fusion module (AFFM). Furthermore, we use the MPDIoU loss function as the bounding-box regression loss function, which not only enhances model robustness to noise but also simplifies the calculation process and improves the final detection efficiency. Finally, the proposed MFEFNet was tested on VisDrone and UAVDT datasets, and the mAP0.5 value increased by 2.7% and 2.2%, respectively.

Keywords: feature extraction; multi-scale fusion; objection detection; UAV aerial images

Citation: Zhou, L.; Zhao, S.; Wan, Z.; Liu, Y.; Wang, Y.; Zuo, X. MFEFNet: A Multi-Scale Feature Information Extraction and Fusion Network for Multi-Scale Object Detection in UAV Aerial Images. *Drones* **2024**, *8*, 186. https://doi.org/10.3390/drones8050186

Academic Editor: Anastasios Dimou

Received: 30 March 2024
Revised: 6 May 2024
Accepted: 6 May 2024
Published: 8 May 2024

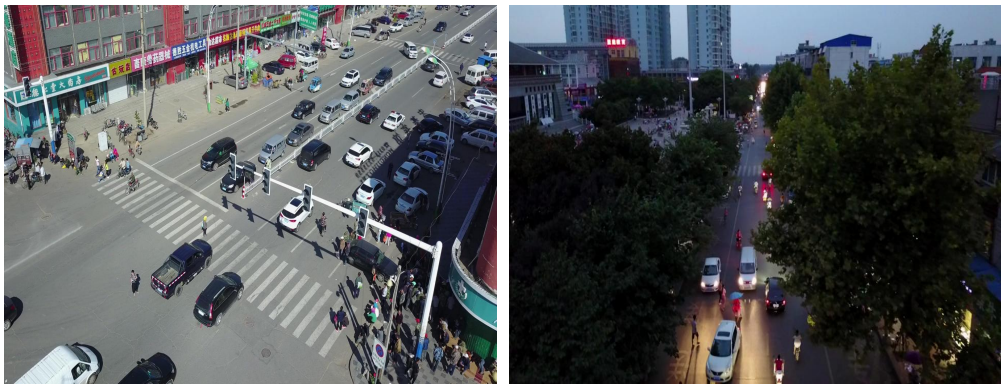
1. Introduction

Object detection holds significance in the field of computer vision, which aims to identify objects in images or videos and determine their locations and categories. Given the swift progress in deep learning neural networks and the large-scale emergence of relevant datasets [1,2], object detection algorithms have made significant strides, finding successful applications across diverse domains, such as automatic driving [3], video surveillance [4], and device detection [5]. Deep learning-based object detection methods have replaced traditional approaches as the prevailing methods. Currently, deep learning-based object detection methods can be broadly categorized into two groups, namely two-stage methods and one-stage methods.

Two-stage object detection methods play a vital role in the domain of object detection, which usually contains two main stages, namely candidate region generation and object classification localization. Classical two-stage object detection methods include R-CNN (regions with CNN features) [6], Fast R-CNN [7], Faster R-CNN [8], Mask R-CNN [9], and

Cascade R-CNN [10]. The primary benefit of the two-stage object detection methods lies in their exceptional accuracy. However, due to the two-stage processing, such object detection methods usually require more computing resources and longer processing time than one-stage methods. In contrast to two-stage object detection methods, one-stage object detection methods output the categories and locations of the objects directly from the original image without the step of the region proposal. Classic one-stage detection methods include the YOLO series [11–14], single-shot multi-box detector (SSD) [15], and RetinaNet [16]. Although these excellent detection methods have been derived for object detection so far, the object detection results of UAV images are still not satisfactory, and the task of UAV object detection still faces severe challenges.

UAVs, also known as unmanned aerial vehicles, are widely used in remote sensing mapping [17], maritime emergency rescue [18], urban inspection [19], and other relevant fields due to their convenience of use and low cost. As shown in Figure 1, UAV images mainly have the following two features. First, due to the difference in the tilt angle of UAV shooting, the object size is large in areas near the UAV camera, while the object size is small in places far away from the UAV camera, resulting in large scale changes in objects within UAV aerial images. Secondly, due to differences in the positions and heights of UAVs, drone images often contain numerous small objects that are vulnerable to interference from complex backgrounds. Therefore, detecting objects in UAV images poses a significant challenge. Many excellent researchers are working to solve the difficulties of UAV object detection. Wu et al. [20] proposed a multi-branch parallel network that utilizes multi-branch up-sampling and down-sampling to reduce information loss when the size of a feature map changes. Wang et al. [21] added an ultra-lightweight subspace attention module (ULSAM) to a path aggregation network to highlight object features. Huang et al. [22] proposed a feature-guided enhancement (FGE) module that designs two nonlinear operators to learn discriminant information. Although these methods are effective for UAV image object detection, they ignore the importance of fine-grained information in shallow feature maps.



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

Figure 1. Examples in the VisDrone dataset. (**a**,**b**) Multi-scale objects in UAV images in different scenes.

To make up for the shortcomings of previous studies and overcome the problems encountered in UAV image detection, this paper proposes a UAV object detection method named MFEFNet, which includes our proposed a spatial information extraction module (SIEM) and a global aggregation progressive feature fusion network (GAFN). First, objects in UAV images are susceptible to complex background interference. Therefore, we designed a module for extracting the location information and context relationships of objects within shallow feature maps named SIEM. The module expands the regional receptive field while maintaining the size of the feature map and weights the perceived global context features with the initial input features utilizing the residual connection. Therefore, the most useful location information for small objects is obtained, and the context information relationship around objects is constructed. Secondly, we design a global aggregation progressive feature

fusion network (GAFN) based on the adaptive feature fusion module (AFFM). The network first fuses the feature maps of different scale sizes into two mid-level feature maps. Then, the two obtained middle-level feature maps, along with the high-level feature maps abundant in semantic details, are subjected fully adaptive feature fusion. The feature maps of various scales can preserve the most effective feature information. Finally, the MPDIoU [23] loss function is used as the bounding-box regression loss function to enhance the model's precision in bounding-box localization, which makes the model simplify the calculation process of the loss function and improves the detection efficiency.

This study's main contributions are summarized as follows:

(1) We design a novel UAV image object detection method based on global feature aggregation and context feature extraction named MFEFNet , which enhances the extraction of representation information from multi-scale objects and improves the fusion efficiency of multi-scale objects.

(2) We design a spatial information extraction module (SIEM), which is mainly used for extracting the spatial information of objects and combining the residual connection to construct the long-distance feature dependency. It effectively avoids the interference of background information.

(3) To achieve efficient fusion of multi-scale features, we design a feature fusion network named GAFN, which uses the multi-structure and multi-level adaptive feature fusion module (AFFM) to adaptively learn the feature map importance of different sizes and mix features of different scales . The efficiency of multi-scale feature fusion and multi-scale object detection is considerably improved.

(4) We validate our MFEFNet on two public UAV image datasets and one public remote sensing dataset. The experimental results show that our method has better detection ability for multi-scale objects in UAV images compared with other advanced methods.

2. Related Work

2.1. Object Detection under Background Interference

One of the difficult problems in UAV object detection is that objects are susceptible to complex background interference. To solve this problem, existing deep learning-based object detection methods primarily concentrate on enhancing the feature extraction capability by focusing on increasing the network depth. However, these methods not only burden the network but also result in insufficient spatial information because the size of the deep network feature maps is relatively small. In view of the various problems that have arisen, many researchers have put forward different research methods. Qu et al. [24] proposed a detection head enhancement module (DHEM) that utilizes an attention mechanism and multi-scale feature fusion to enhance the representational information of objects. However, it only focuses on small objects and lacks attention to medium and large objects in UAV images. Wang et al. [25] proposed a novel evaluation metric, the normalized Wasserstein distance (NWD), which uses the 2D Gaussian distribution of bounding boxes to solve the sensitivity problem of small object position deviation derived from intersection over union (IoU). However, it lacks effective feature extraction and fusion mechanisms. In addition, considering the problem that small objects are easily occluded, Li et al. [26] introduced an occlusion-positioning sub-task, which, together with the object detection sub-task, constitutes an occlusion-guided multi-task network (OGMN), effectively improving the detection efficiency of occluded objects. Although these methods improve the detection accuracy of objects subject to background interference, they ignore the most important spatial information for object localization. Therefore, we design a spatial information extraction module in the shallow network to construct the context information around objects.

2.2. Multi-Scale Object Detection

Building an efficient feature pyramid network is also the key to improving the efficiency of multi-scale object detection. Liu et al. proposed the SSD [15], which uses the idea of divide and conquer. It utilizes shallow feature maps for small object detection and

deep feature maps for large object detection, which effectively reduces the computational overhead. However, due to the lack of effective feature fusion from top to bottom, the shallow feature maps lack adequate representational ability, leading to limited enhancement in small object detection performance. Subsequently, the emergence of the feature pyramid network (FPN) [27] and path aggregation network (PANet) [28] provided ideas for feature fusion from both top-down and bottom-up perspectives, realizing information exchange among feature maps of varying scales and improving the performance of small object detection. In addition, many novel and efficient feature fusion networks based on FPN and PANet have become mainstream, such as the bidirectional feature network (BiFPN) [29] and the asymptotic feature pyramid network (AFPN) [30]. However, the above feature fusion network structure leads to low fusion efficiency and feature redundancy problems. To further reduce the burden caused by the feature pyramid and improve the efficiency of feature fusion, we propose an efficient and lightweight feature fusion network inspired by AFPN. By fusing the shallower feature map a single time, not only is the unique object location information of the shallow feature maps taken into account, but the feature redundancy caused by excessive fusion is avoided.

2.3. UAV Image Object Detection

In contrast to natural images, the shooting height and angle of UAV images are highly random, which leads to the inapplicability of object detection methods that perform well in natural images [31]. The objects in UAV images often display significant changes in scale and irregular arrangement, which greatly increase the difficulty of UAV image object detection. Fang et al. [32] proposed a dehazing subnetwork to detect UAV images affected by fog. However, it does not perform well under normal lighting conditions. Redmon et al. [33] proposed a CFA structure for parallel fusion of feature maps of different scales to obtain high-quality feature fusion results and used an LASPP module to expand receptive fields and maintain sensitivity to different receptive fields. However, it lacks effective extraction of objects' spatial information, resulting in weak detection ability for small objects. Leng et al. [34] proposed a Reverse-attention Exploration Module (REM) to obtain the location of challenging-to-detect objects and, through a Region-specific Context Learning Module (RCLM), to improve the feature richness of the corresponding position and improve detection efficiency. Although this method can accurately locate the key region, it lacks an effective mechanism to process the feature information of the key region. Ye et al. [19] proposed a Convolutional Multi-Head Self-Attention (CMHSA) method based on an Efficient Convolutional Transform Block (ECTB) to improve the recognition ability of occluded small objects by extracting contextual information of objects. However, due to the incomplete extraction of fine-grained information in the process of feature extraction, the false detection of occlusions increases. Lu et al. [35] proposed a hybrid model of a CNN and transformer to achieve UAV image object detection, which is helpful in enhancing the efficiency of object detection at various scales in UAV images. However, the introduction of transform increases the complexity and redundancy of the network, which is unfavorable for UAV image detection. Chalavadi et al. [36] proposed an effective network for detecting objects at multiple scales named mSODANet, which uses hierarchical extended convolution to establish contextual details from diverse objects across various scales and domains. Shen et al. [37] proposed a divide-and-conquer method based on prior information. In this method, the inertial measurement unit (IMU) is used to first calculate the object scale; then, the UAV images are divided into three parts according to the object scale for object detection. However, this method leads to the neglect of objects of other sizes in different domains.

In order to solve the problem of large changes in object scale and complex background interference in UAV aerial images, we propose a spatial information extraction module (SIEM), which strengthens the feature extraction ability of small objects and object context relationships in shallow networks by combining residual connection and context-dependent extraction modules similar to transform. In addition, we design a multi-level and multi-scale feature fusion network based on AFPN named GAFN.

3. Methods

Our main goal is to deal with the problems that objects in UAV aerial images are susceptible to background influence and large-scale variation by strengthening the spatial feature extraction ability of object detection and constructing an efficient multi-scale feature fusion network. First, we propose a spatial information extraction module named SIEM, which combines residual connection and a transform-like mechanism to establish a long-distance dependency, effectively constructing high-quality contextual feature information and avoiding excessive loss of object information at the image edge. Secondly, the extracted effective object spatial information layer is aggregated with the other three layers for cross-scale single adaptive aggregation, which fully retains the object features of each scale while avoiding feature redundancy and excessive parameters caused by excessive aggregation. Thirdly, the results of cross-scale single adaptive aggregation are integrated with deep feature maps rich in semantic object information to further preserve useful information. Figure 2 illustrates the overall structure of MFEFNet. The backbone adopts the combination of CSPDarknet53 and SIEM for feature extraction. The neck network uses a three-level global aggregation progressive feature fusion network (GAFN), which is composed of single-level adaptive aggregation and two-level progressive feature fusion for multi-scale feature fusion. The head is utilized to predict both the position and type of the object.

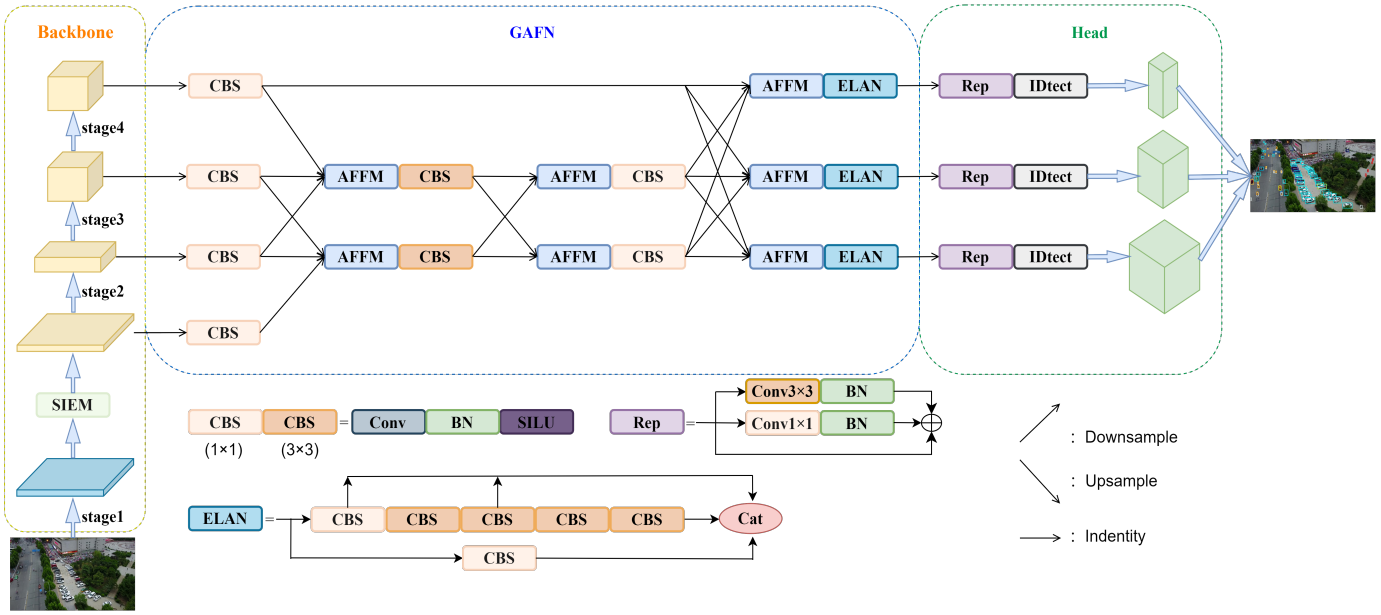


Figure 2. The overall structure of MFEFNet.

3.1. Spatial Information Extraction Module

Due to the randomness of the UAV's shooting height and angle, the surroundings of the objects in the UAV aerial image are complicated. In the process of feature extraction, the object's spatial information is lost due to the increase in the receptive field. In addition, a complex background can result in ambiguity between object edge details and background information. To solve the above problems, existing methods include reducing the interference of background information by adding object detection branches. However, this approach adds an additional burden to the network. Due to the lack of an effective object spatial information extraction mechanism, complex background information can easily interfere with objects and lead to location information loss in the feature extraction operation of the model, which affects the efficiency of object detection.

Given the above problems that object spatial information is easily lost and objects are easily disturbed by the environment, we designed a spatial information extraction module. Figure 3 shows the structure of the SIEM. In contrast traditional convolutions, we

not only use continuous 3 × 3 convolutions to expand the receptive field but also combine multi-branch sampling and transform-like mechanisms to increase the influence region and establish long-distance dependencies. Finally, the obtained results are weighted to the original feature map by residual connection to achieve small object feature extraction and context construction. To be specific, first of all, we extract local features from the input feature map in two branches to improve the feature transformation ability and enlarge the receptive field of the model. Then, the deep-extracted local features and lightly extracted local features are spliced together by the Concat module to recover the original input channel number. The calculation process is shown in Formula (1).

$$F_1 = Cat(CBS_{3\times3}(CBS_{3\times3}(CBS_{3\times3}(F_{in}))), CBS_{3\times3}(CBS_{3\times3}(F_{in}))) \quad (1)$$

where F_{in} denotes the input feature map, and F_1 denotes the output result of the first stage. Next, a 3 × 3 convolution is used to integrate the local feature information extracted from the previous layer. To improve the global perception ability, we perform global interaction calculations between the obtained features and the pixel values after further extraction and activation to acquire global contextual features. The calculation process is shown in Formula (2).

$$F_2 = CBS_{3\times3}(F_1) \times Softmax(CBS_{3\times3}(CBS_{3\times3}(F_1))) \quad (2)$$

where F_2 denotes the output of the second stage. Finally, we use a Batchnorm layer and two 1 × 1 linear mapping layers to increase feature diversity. At this point, we establish stable long-range dependencies of the input feature maps. In addition, we employ a maximum pooling layer to capture the essential feature information within the input feature map. The extracted features are weighted, together with the global context features, into the input feature map using the residual connection. The calculation process is shown in Formula (3).

$$F_{out} = Conv_{1\times1}(BN(Conv_{1\times1}(F_2))) + CBS_{3\times3}(F_1) + Maxpool_{5\times5}(F_{in}) + F_{in} \quad (3)$$

where F_{out} denotes the final output of the SIEM. Therefore, combined with the abundant object location information in shallow feature maps, our model can effectively extract object spatial location information and establish context relationships in the initial stage of the feature extraction network, especially to avoid the loss of object edge location details caused by the interference of complex backgrounds.

It is worth noting that the original intention of SIEM design is to efficiently extract the object spatial information and construct the context relationship between objects and the surrounding environment. However, the deep feature map itself loses a lot of object location information because the receptive field is too large. In addition, the number of channels in the deep network is usually 2–4 times that of the shallow network, and the effect of SIEM in the deep network is not only very limited but also increases the burden of the network. Therefore, considering that the shallow network still contains rich location information and the number of channels is small, the ideal location of the SIEM is in the shallow network. In Section 4, the experimental results of SIEM at different locations in the backbone network confirm our idea. The position behind Stage 1 in Figure 2 proved to be the most desirable. It is consistent with our original intention of designing this module.

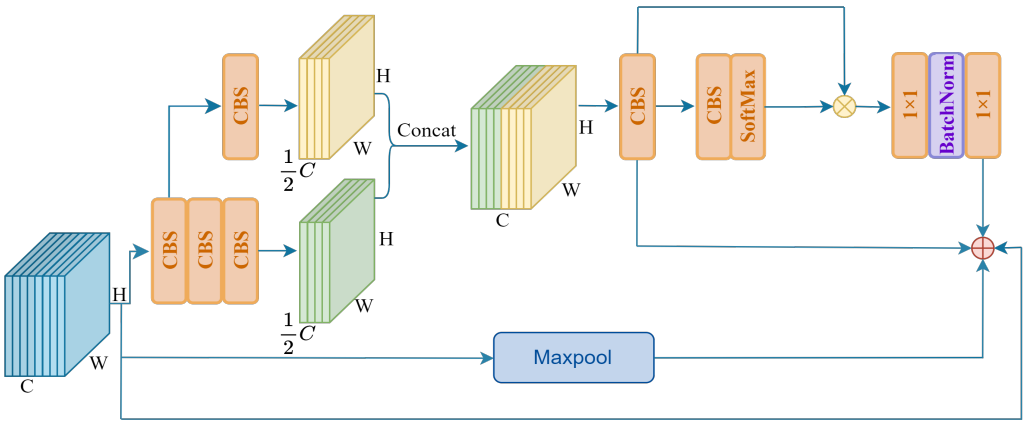


Figure 3. The structure of SIEM.

3.2. Global Aggregation Progressive Feature Fusion Network

Another notable feature of UAV images is that the object scale changes greatly. The current mainstream approach to solve this problem is to utilize a combination of bottom-up and top-down PANet. Shallow feature maps are rich in object location information, which is beneficial for regression localization tasks, but a lack of semantic information leads to insufficient classification ability. Deep feature maps are rich in object semantic information, contributing to the classification task, but a lack of location information leads to poor localization ability. Although PANet has good fusion efficiency, its high parameter number and complex redundancy feature add a significant load to the network. AFPN has become popular due to its light weight and high efficiency. Figure 4 shows the simple structure of AFPN, which adaptively fuses each feature map in the backbone network through progressive fusion. AFPN aims to reduce the semantic information difference between cross-layer feature maps and to alleviate the multi-objective information conflict in the process of feature fusion of spatial location. The fusion module it employs assigns different spatial weights to features at different levels, which enhances the importance of critical levels and mitigates the impact of contradictory information from different objects. However, AFPN ignores the importance of high-level semantic information for object classification in UAV images. Therefore, we design a more parsimonious and efficient feature fusion network inspired by AFPN named GAFN, whose simple structure is shown in Figure 5. A three-level adaptive feature fusion structure is adopted in the network. Feature maps with different scales and different feature information are screened efficiently by using an adaptive feature fusion module. Specifically, first of all, we perform an adjacent adaptive fusion of four feature maps of varying sizes from the backbone network. It is worth noting that to further improve the multi-scale object detection efficiency, feature maps rich in object location information extracted by the SIEM are also involved in the fusion network for a single fusion. Secondly, to establish the relationship between the shallowest feature map and the deepest feature map, we carry out comprehensive adaptive fusion on the two adjacent adaptive fusion results. Finally, considering the classification problem at the end of the model, some semantic information is easily lost in the first two fusion processes, so we re-add the top-level feature map rich in semantic information in the last level of global adaptive fusion.

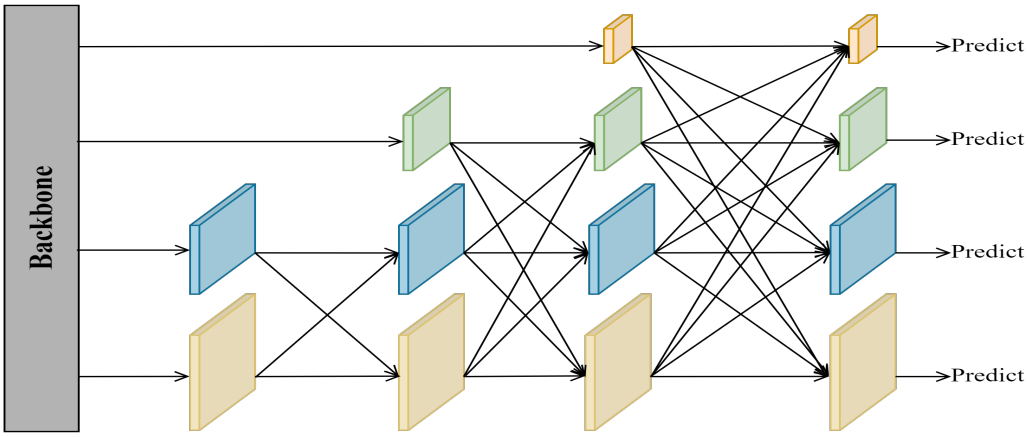


Figure 4. The simple structure of AFPN.

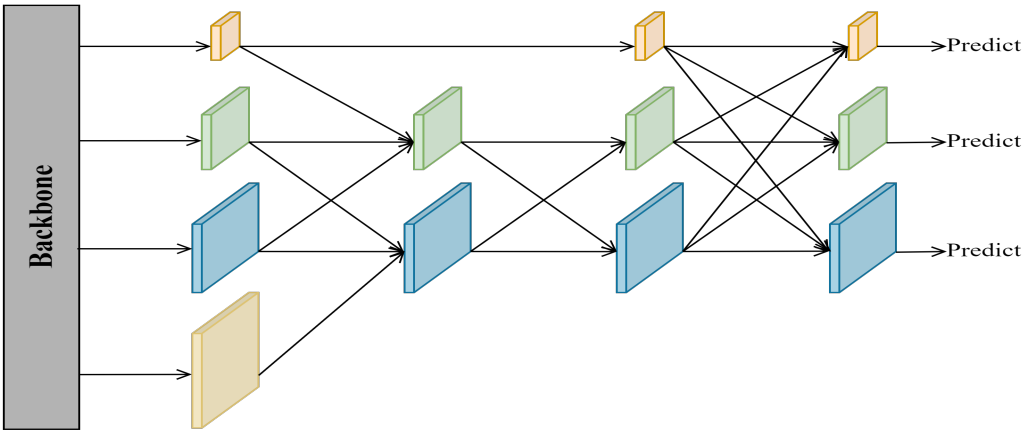


Figure 5. The simple structure of GAFN.

The adaptive feature fusion module (AFFM) plays a crucial adjective role in the whole feature fusion network; it can fuse 2–3 feature maps of varying sizes to perform feature screening. The fusion process of two adjacent dimensional feature maps is shown in Figure 6a.

Case 1: The first fusion method is the adaptive fusion of the middle-layer feature map with double up-sampling and the shallow feature map. The calculation method is shown in Formula (4).

$$F_{out} = AFFM(\gamma_1 \times F1, \gamma_2 \times Upsample_{2\times}(F2)) \quad (4)$$

Case 2: The second fusion method is the adaptive fusion of the shallow feature image with the middle-layer feature image after double down-sampling. The calculation method is shown in Formula (5).

$$F_{out} = AFFM(\gamma_1 \times Downsample_{2\times}(F1), \gamma_2 \times F2) \quad (5)$$

The fusion process of the feature maps of three adjacent dimensions is shown in Figure 6b.

Case 3: The third fusion method is the adaptive fusion of the results of the deep feature map after quadruple up-sampling and the middle feature map after double up-sampling with the shallow feature map. The calculation method is shown in Formula (6).

$$F_{out} = AFFM(\gamma_1 \times F1, \gamma_2 \times Upsample_{2\times}(F2), \gamma_3 \times Upsample_{4\times}(F3)) \quad (6)$$

Case 4: The fourth fusion method is the adaptive fusion of the results of the deep feature map after double up-sampling and the shallow feature map after double down-sampling with the middle feature map. The calculation method is shown in Formula (7).

$$F_{out} = AFFM(\gamma_1 \times Downsample_{2\times}(F1), \gamma_2 \times F2, \gamma_3 \times Upsample_{2\times}(F3)) \quad (7)$$

Case 5: The fifth fusion method is the adaptive fusion of the results of the shallow feature map after quadruple down-sampling and the middle feature map after double down-sampling with the deep feature map. The calculation method is shown in Formula (8).

$$F_{out} = AFFM(\gamma_1 \times Downsample_{4\times}(F1), \gamma_2 \times Downsample_{2\times}(F2), \gamma_3 \times F3) \quad (8)$$

where γ_1 and γ_2 are learnable weight parameters, F_1 denotes the input shallow feature map, F_2 denotes the input middle feature map, F_{out} denotes the result of adaptive fusion, F_3 denotes the input deep feature map, and γ_3 is a learnable weight parameter.

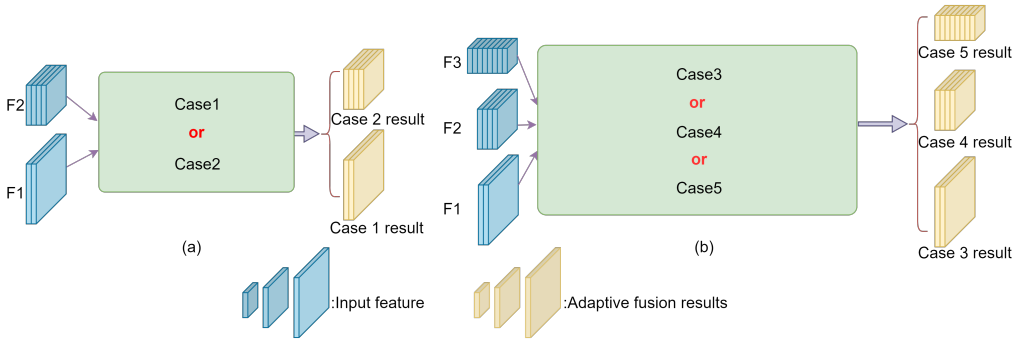


Figure 6. The fusion structure of AFFM. (**a**) represents the fusion process of two adjacent dimensional feature maps; (**b**) represents the fusion process of three adjacent dimensional feature maps.

AFFM, together with other modules, constitutes the complete GAFN, which is shown in Figure 2. CBS represents a set of convolution, batch normalization, and SiLU activation function operations. The convolution operation is mainly used to achieve feature extraction, the normalization operation is mainly used to avoid the appearance of gradient disappearance, and the SiLU activation function is mainly used to suppress the overfitting phenomenon and improve the generalization ability of the model. The ELAN module is an efficient network architecture that enables the network to learn more features and be more robust by controlling the shortest and longest gradient paths. The IDtect module is used to generate the final output of the object detection task, including steps such as bounding-box prediction, category prediction, and post processing to provide accurate object detection results.

By fusing multi-scale feature maps with adaptive weighting, the most valuable object feature information can be preserved. The AFFM proposed in this paper independently learns the weight parameters based on the information of each pixel of the input feature map, which improves the fusion efficiency of feature information at various scales and further enhances the model's detection ability for multi-scale objects. In addition, eliminating unnecessary modules is conducive to reducing the burden of the network. Our proposed GAFN significantly reduces model parameters and avoids feature redundancy.

We show the adaptive fusion steps of the three feature maps in the AFFM in Algorithm 1; the fusion of two feature maps is similar. First, we adjust the channel of $X = \{x_1, x_2, x_3, x_4\}$ after feature size matching to obtain $Y = \{y_1, y_2, y_3\}$. Secondly, the obtained feature maps are concatenated in the channel dimension to obtain F, and 1×1 convolution and softmax are used to obtain the weight feature ($W = \{w_1, w_2, w_3\}$) of the three channels. Finally, X is weighted by W, and the adaptive fusion result (L) is obtained by 3×3 convolution.

Algorithm 1 The feature fusion steps of AFFM.

Input: $\mathbf{X} = \{x_1, x_2, x_3, x_4\}$, **X** refers to three feature maps after feature size matching.
Step 1: $\mathbf{Y} = \{\}$, **Y** refers to the first intermediate feature map generated by the $CBS_{1\times1}()$ for channel adjustment. $CBS()$ represents a series of convolution operations required.
  for $i = 1$ to 3 **do**
    $y_i = CBS_{1\times}(x_i)$
    $Y.append(y_i)$
  end for
Step 2: **F** refers to the concatenation result of **Y**. $\mathbf{W} = \{\}$, **W** refers to the weight feature. $Concat()$ represents channel concatenation operation.
  $F = Concat(y_1, y_2, y_3)$
  $W = softmax(CBS_{1\times1}(F))$
  for $i = 1$ to 3 **do**
    $w_i = W[i-1, i]$
  end for
Step 3: **L** refers to the output feature map generated by adaptive fusion.
  for $i = 1$ to 3 **do**
    $l_i = w_i \times x_i$
    $L+ = l_i$
    $L = CBS_{3\times3}(L)$
  end for
Output: Return **L**.

3.3. Loss Function

The most widely used loss function for bounding-box regression is the CIoU loss function [38]. The CIoU loss function considers the overlap area between the prediction box and the truth box, along with the distance between their center points and the disparity in aspect ratios. The CIoU loss function takes into account almost all aspects affecting loss accuracy, and its definition is shown in Formulas (9)–(11):

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{9}$$

$$v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{10}$$

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{11}$$

where ρ denotes the center-point Euclidean distance between the truth box and the prediction box; c is the length of the diagonal of the smallest external rectangle of the truth box and the prediction box; v is used to calculate the consistency of the aspect ratios of the truth box and the prediction box; α is the equilibrium coefficient of ρ; w and h are the length and width of the prediction box, respectively; and w^{gt} and h^{gt} are the length and width of the truth box, respectively.

It is worth noting that most of the existing loss functions, including CIoU loss, cannot optimize the case where the prediction box and the truth box share the same aspect ratio but differ in length and width values. In addressing this issue and enhancing the accuracy of calculating the bounding-box loss, we use the MPDIoU loss function as the bounding-box loss function for MFEFNet, which is defined as Formulas (12) and (13).

$$MPDIoU = \frac{A \cap B}{A \cup B} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{12}$$

$$L_{MPDIoU} = 1 - MPDIoU \tag{13}$$

where A and B denote the truth box and the prediction box, respectively; d_1 denotes the distance of the upper-left vertices between the truth box and the prediction box; and d_2

denotes the distance of the upper-right vertices between the truth box and the prediction box. The coordinates of the top-left point and the bottom-right point can be used to uniquely identify a rectangle, and all the factors considered by the existing mainstream boundary regression loss function can be reflected in the relationship between the four vertices of a rectangle. Therefore, using the relationship between vertices to replace various complex factors can simplify the calculation process and enhance the model's prediction accuracy.

Furthermore, we use the binary cross entropy with LogitsLoss (BCEWithLogitsLoss) [39] function as the classification loss and confidence loss function for our model; its definition is shown in Formula (14).

$$L_{BCE} = \sum_{n=1}^{N} -[y_n log(\sigma(x_n)) + (1 - y_n) log(1 - \sigma(x_n))] \tag{14}$$

where N is the batch-size number, and y_n and x_n denote the label-predicted vector and truth-value vector, respectively.

Our loss function comprises three components, namely confidence loss, regression loss, and classification loss. The smaller the loss, the better the model detection effect. In the model training phase, the loss function is minimized by continuously adjusting the parameters using back-propagation. The loss function of our proposed method is shown in Formula (15).

$$\begin{aligned} Loss &= \lambda_1 Loss_{cls} + \lambda_2 Loss_{reg} + \lambda_3 Loss_{obj} \\ &= \lambda_1 Loss_{MPDIoU} + \lambda_2 Loss_{BCE} + \lambda_3 Loss_{BCE} \end{aligned} \tag{15}$$

where $Loss_{cls}$ denotes the classification loss of the model; $Loss_{reg}$ denotes the regression loss of the model; $Loss_{obj}$ denotes the confidence loss of the model; $Loss$ denotes the overall model loss; P_{cls} and T_{cls} denote the prediction class and truth class, respectively; P_{reg} and T_{reg} denote the prediction box and the truth box, respectively; P_{obj} and T_{obj} denote prediction confidence and truth confidence, respectively; λ_1, λ_2, and λ_3 are the weight parameters of the above three sub-loss functions, which are set to $\lambda_1 = 0.3$, $\lambda_2 = 0.05$, and $\lambda_3 = 0.7$, respectively.

4. Experiments

4.1. Datasets

This section introduces the datasets, evaluation metrics, and parameter settings adopted for our experiments. In addition, we introduce a series of qualitative and quantitative experimental methods to evaluate and contrast our model with other advanced methods.

In the experimental stage, we adopted two public UAV datasets.

(1) The VisDrone dataset [40] is mainly used for objection detection and image classification, with a total of 10,209 images, of which the training set consists of 6471 images, the test set consists of 3190 images, and the verification set consists of 548 images. The resolution of each image is 2000 × 1500 pixels. The images in the dataset were obtained from 14 different cities under different weather conditions, equipment conditions, and surroundings. After careful manual annotation, the dataset contained 342,391 objects and is divided into the following 10 categories: people, pedestrian, motor, awning-tricycle, bicycle, car, van, truck, tricycle, and bus. This dataset is the main dataset used in our experiments.

(2) The UAVDT dataset [41] consists of a total of 40,735 images, of which the training set consists of 24,206 images and the validation set consists of 16,529 images, each with a resolution of 1080 × 540 pixels. In contrast to the VisDrone dataset, UAVDT mainly focuses on vehicle detection tasks from the perspective of drones, including images of urban roads in different weather, different angles, and different scenes. The dataset includes only three predefined categories of vehicles, namely car, bus, and truck.

4.2. Implementation Details

The experimental setup of this paper is described as follows. The operating system of the server used in this experiment is Ubuntu 18.04.6 LTS and the GPU used in the server is an NVIDIA GeForce RTX 3090 (24G). The CPU used in the server is an Intel(R) Xeon(R) Silver 4114 CPU @2.20 GHz. In addition, we use the Python deep learning framework, where the versions of PyTorch and Python are 1.10.2 and 3.6, respectively, and the CUDA version is 11.7. In the training phase of the experiment, the initial learning rate is set to 0.010, and we use the stochastic gradient descent (SGD) optimizer with momentum. The batch size is set to 16, the weight decay coefficient is set to 0.0005, and the momentum parameter is set to 0.937. During the training and testing phases of the whole experiment, all image input sizes are 640 × 640. In the experimental tables, the units of the data are percentages, except for the parameter number indicator, which is in megabytes.

4.3. Evaluation Metrics

For a precise assessment of the proposed object detection method's performance in detection, we use Precision (P), Recall (R), Average Precision (AP), and mean Average Precision (mAP) as evaluation indexes of the model, where P stands for the ratio of the number of correctly predicted positive samples to the total number of positive samples, expressed as a percentage, and R stands for the ratio of the number of positive samples with correct prediction to the total number of positive samples, expressed as a percentage. The calculation processes of P and R are shown in Formulas (16) and (17), respectively.

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

where TP denotes the count of correctly predicted positive samples, FP denotes the count of incorrectly predicted positive samples, and FN denotes the count of incorrectly predicted negative samples. P and R are two indicators of mutual checks and balances, and AP combines them to reflect the overall ability of the model. The calculation process of AP is shown in Formula (18).

$$AP = \int_0^1 P(R)dR \tag{18}$$

where $P(R)$ denotes the precision value (P) associated with the recall value (R). mAP is the average of AP values for all categories and is recognized as the most authoritative evaluation metric to measure the quality of the current object detection method. The calculation process of mAP is shown in Formula (19).

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{19}$$

where AP_i denotes the AP value for the i-th category, and n denotes the count of categories.

4.4. Experimental Results on the VisDrone Dataset

We performed experiments using the VisDrone dataset and contrasted them with the latest object detection methods, where the number of training iterations was 300. To verify the authority of MFEFNet, COCO evaluation metrics [42] were used to evaluate the model's object detection ability at various scales; indicators include AP_S, AP_M, and AP_L. Under the COCO standard, objects with pixel values below 32 × 32 are categorized as small objects, objects with pixel values between 32 × 32 and 96 × 96 are categorized as medium objects, and objects with pixel values higher than 96 × 96 are categorized as big objects. The experimental results are shown in Table 1. According to the table, the methods proposed by us achieved the highest values of mAP0.5, mAP0.75, and mAP0.5:0.95, which reached

51.9%, 29.8%, and 29.9%, respectively, corresponding to improvements of 2.7%, 2.3%, and 2.1%, respectively, compared with the baseline YOLOv7. In addition, MFEFNet's value of AP_M also achieves the highest score of all object detection methods by reaching 41.4%, an improvement of 2.6% compared to YOLOv7. In terms of small object detection accuracy, our method achieves a 20.6% higher value than YOLOv7, which corresponds to a 2.0% improvement. On the other hand, our MFEFNet exhibits a decline in the accuracy of large object detection compared with YOLOv7 but still achieving 2.0% higher accuracy than the baseline for other indicators, which further proves that our MFEFNet has better multi-scale object detection ability. In addition, compared with CGMDet [43], which has the second highest mAP0.5 value, our MFEFNet significantly improves in all five other indicators, except the value of AP_L. Compared with the four commonly used versions of YOLOv8 [44], except that the AP_L value of MFEFNet is slightly lower than that of YOLOv8x by about 0.2%, other indicators are greatly improved. Although NWD and DMNet [45] exhibit the highest accuracy in detecting small objects and large objects, their mAP0.5 values are much smaller than that of MFEFNet proposed by us. In general, our proposed object detector has better multi-scale object detection capability and is more suitable for UAV aerial images with complex backgrounds.

Table 1. Comparison results under the COCO standard on the VisDrone dataset. "*" means our re-implemented results.

Method	mAP0.5	mAP0.75	mAP0.5:0.95	AP_S	AP_M	AP_L
Faster R-CNN [8]	40.0	20.6	21.5	15.4	34.6	37.1
Cascade R-CNN [10]	39.9	23.4	23.2	16.5	36.8	39.4
YOLOv3 [12]	31.4	15.3	16.4	8.3	26.7	36.9
RetinaNet [16]	35.9	18.5	19.4	14.1	29.5	33.7
NWD [25]	40.3	\	\	**22.2**	\	\
YOLOX [46]	45.0	26.6	26.7	17.4	37.9	45.3
YOLOv5l [47]	36.2	20.1	20.5	12.4	29.9	36.4
HawkNet [48]	44.3	25.8	25.6	19.9	36.0	39.1
QueryDet [49]	48.1	28.8	28.3	\	\	\
Edge YOLO [50]	44.8	26.2	26.4	16.3	38.7	53.1
ClusDet [51]	50.6	24.7	26.7	17.6	38.9	51.4
DMNet [45]	47.6	28.9	28.2	19.9	39.6	**55.8**
CEASC [52]	50.7	28.4	28.7	\	\	\
CDMNet [53]	49.5	29.8	29.2	20.8	40.7	41.6
CGMDet [43]	50.9	29.4	29.3	20.2	40.6	47.4
YOLOv8s [44] *	40.0	21.8	23.6	12.7	33.4	42.0
YOLOv8m [44] *	42.6	24.0	25.6	14.8	35.5	41.8
YOLOv8l [44] *	44.1	24.8	27.1	15.3	36.0	44.7
YOLOv8x [44] *	45.4	26.8	28.0	16.7	38.9	45.5
YOLOv7 [14] *	49.2	27.5	27.8	18.6	38.8	47.8
MFEFNet (Ours)	**51.9**	**29.8**	**29.9**	20.6	**41.4**	45.3

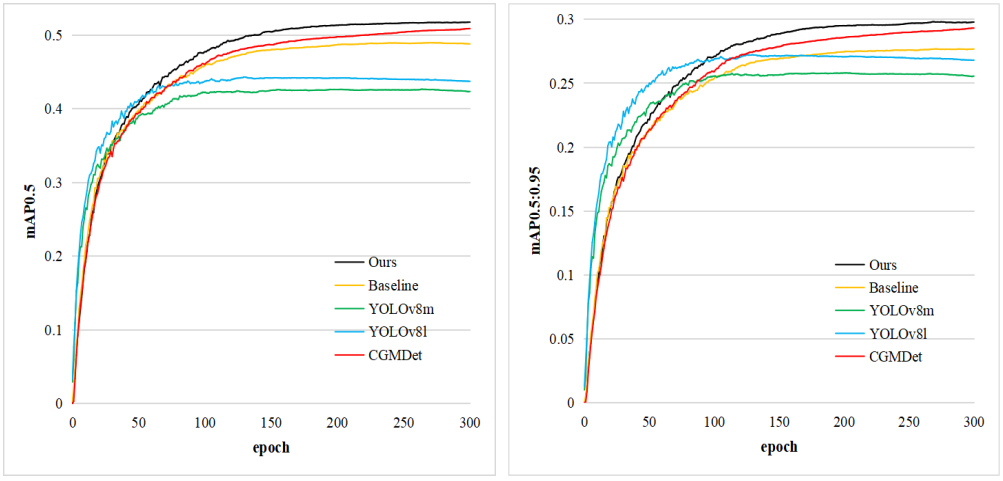
To visualize the detection capability of our MFEFNet, we select several state-of-the-art object detection methods and plot their values of mAP0.5 and mAP0.5:0.95 as the number of iterations increases. As shown in Figure 7, although YOLOv8 series algorithms have a faster convergence speed, they lack the ability of continuous learning. In contrast, our MFEFNet has a stronger learning ability and achieves a large improvement over the baseline model.

To further validate our model's effectiveness, we compared the detection precision of each category on the VisDrone dataset with some state-of-the-art detection methods. Table 2 shows the experimental results. The detection accuracy of our MFEFNet is higher than that of YOLO7 in all 10 categories, with an average of a 2.8% improvement. Compared with CGMDet, except for the two categories of car and bus, whose accuracy is slightly lower, the remaining eight categories have higher accuracy than CGMDet. Although YOLO-DCTI [54] has the best performance on various larger object types, our MFEFNet performs better on small and medium objects and outperforms other object detection

methods on larger object types. Relative to Faster-RCNN, YOLOv3, YOLOv5l, YOLOv5s-pp, and the YOLOv8 series of mainstream algorithms, our MFEFNet achieves superior accuracy values in each category.

Table 2. Comparison of results for each category on the VisDrone dataset. "*" means our re-implemented results.

Method	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awning Tricycle	Bus	Motor	mAP0.5
Faster R-CNN [8]	37.5	19.4	13.3	71.9	42.5	42.8	19.8	18.1	58.4	34.4	35.8
YOLOv3 [12]	12.8	7.8	4.0	43.0	23.5	16.5	9.5	5.1	29.0	12.5	31.4
YOLOv5l [47]	44.4	36.8	15.6	73.9	39.2	36.2	22.6	11.9	50.5	42.8	37.4
YOLOv5s-pp [55]	51.7	39.6	19.0	82.1	44.1	36.0	26.3	14.7	55.3	48.2	41.7
YOLO-DCTI [54]	48.7	36.2	22.6	82.1	**58.2**	**60.0**	34.5	**31.4**	**72.9**	51.2	49.8
CGMDet [43]	59.7	50.7	25.4	**86.2**	53.4	47.4	37.9	20.2	66.3	61.6	50.9
YOLOv8s [44] *	43.4	33.4	13.7	79.7	44.1	37.9	28.5	15.0	58.3	45.5	40.0
YOLOv8m [44] *	47.0	36.9	16.9	81.0	47.1	40.9	31.6	17.6	57.9	49.1	42.6
YOLOv8l [44] *	46.8	37.2	18.4	81.4	49.8	42.2	34.4	18.0	62.6	49.9	44.1
YOLOv8x [44] *	49.1	38.0	19.3	82.5	50.0	44.5	34.5	18.4	65.6	51.9	45.4
YOLOv7 [14] *	57.7	48.5	22.5	84.7	52.2	45.5	38.2	20.1	62.3	59.9	49.2
MFEFNet (Ours)	**59.9**	**51.2**	**25.9**	85.8	55.3	49.4	**40.3**	23.2	65.1	**63.1**	**51.9**

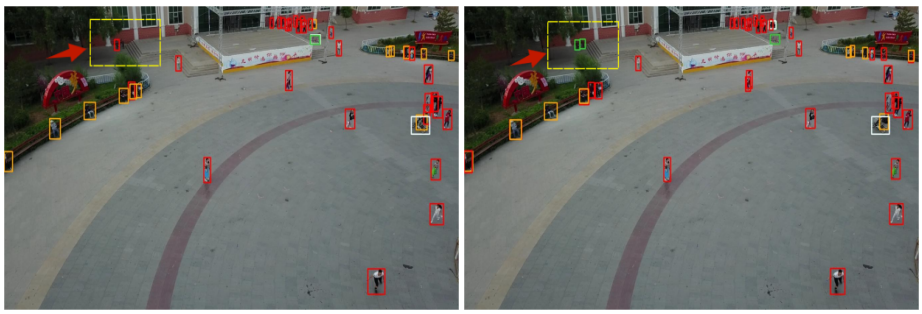


(**a**) (**b**)

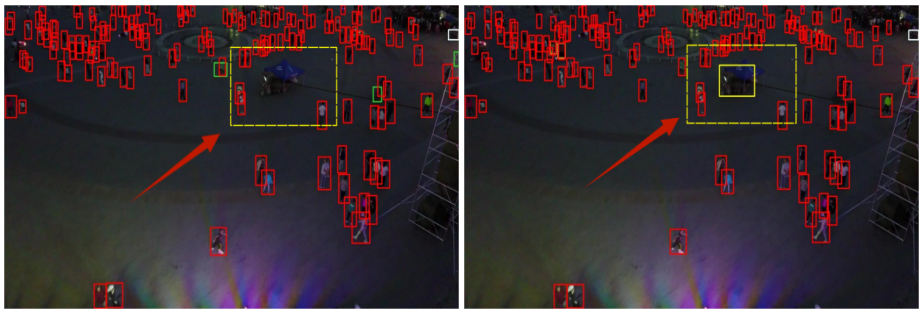
Figure 7. Comparison of mAP0.5 and mAP0.5:0.95 values on the VisDrone dataset. (**a**) represents the comparison result of mAP0.5 values; (**b**) represents the comparison result of mAP0.5:0.95 values.

Simultaneously, we present the visualization results of the baseline model and MFEFNet in different scenarios. We select five groups of representative comparison results from a large number of comparison pictures. In Figure 8, the obvious contrast part is marked with yellow dotted boxes and red arrows. As shown in Figure 8(a1), YOLOv7 incorrectly detects the top-left bicycle as a pedestrian, while our proposed MFENet accurately detects two side-by-side bicycles. In Figure 8(a2), YOLOv7 misses an awning tricycle in the dim background, but our model can still correctly detect the awning tricycle in the face of such a complex, dim background. Figure 8(a3,b3) show the detection results of an outdoor parking lot. YOLOv7 mistakenly detects the chimney on the lower-right roof as a pedestrian, while our MFENet does not produce such an error. To prove the detection performance of our model under an extremely dim and complex background, we conduct a comparative experiment, as shown in Figure 8(a4,b4). Our MFENet successfully detects a pedestrian under an extremely chaotic background at the bottom right of the picture. In addition, we show the detection results located achieved in a suburban area in Figure 8(a5,b5). YOLOv7 misdetects the roof on the right as a van and misses the awning tricycle in the middle of the picture, while our model does not produce such errors.

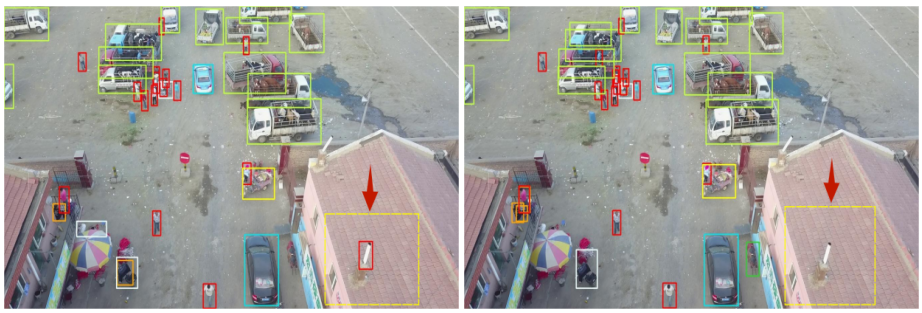
To verify our theoretical ideas about the location of the SIEM mentioned above, we place the module in four different locations in the backbone network and conduct comparative experiments. Table 3 shows the experimental results. From the experimental results, it is easy to see that the most ideal position of the SIEM is after Stage 1, which achieves the best results for each evaluation index with the fewest model parameters. Experimental results prove that the theoretical ideas we put forward above are scientific and valid.



(**a1**) (**b1**)



(**a2**) (**b2**)



(**a3**) (**b3**)

(**a4**) (**b4**)

Figure 8. *Cont.*

(**a5**) (**b5**)

Figure 8. Comparison of visualization results between YOLOv7 (**a1**–**a5**) and MFEFNet (**b1**–**b5**) on the VisDrone dataset. The obvious contrast parts is marked with yellow dotted boxes and red arrows.

Table 3. Comparative experiment using different positions of the SIEM module in MFEFNet on the VisDrone dataset.

Location	Precision	Recall	mAP0.5	mAP0.5:0.95	Param (M)
After Stage 1	**62.0**	**50.8**	**51.9**	**29.9**	**33.6**
After Stage 2	60.2	49.0	49.3	27.8	43.2
After Stage 3	61.4	46.8	48.6	27.3	68.3
After Stage 4	58.7	48.7	48.7	27.5	68.3

To confirm the efficacy of our proposed GAFN for multi-scale fusion, we designed a set of comparative experiments on feature fusion networks, and Table 4 shows the experimental results. Although PAFPN has the highest precision value, it has a low recall value and a lot of parameters. Although BIFPN has the highest recall value, it has a lower precision value and a larger number of parameters. AFPN has the lowest number of parameters, but it has the lowest mAP0.5 value. Our GAFN has the highest mAP0.5 value and mAP0.5:0.95 value, and its precision and recall values are about the same as the highest values. All in all, the GAFN can better extract multi-scale feature information and has better detection capability in UAV aerial images.

Table 4. Comparative experiment of feature fusion network on the VisDrone dataset. "*" means our re-implemented results.

Method	Precision	Recall	mAP0.5	mAP0.5:0.95	Param (M)
PAFPN [28] *	**59.1**	48.9	49.2	27.8	34.8
BIFPN [29] *	56.4	**50.4**	49.1	27.7	33.9
AFPN [30] *	57.8	49.7	48.7	27.7	**27.4**
GAFN (Ours)	57.9	50.0	**49.5**	**28.0**	31.5

In addition, a special experiment to verify the effectiveness of MPDIoU for UAV aerial image detection is presented in Table 5. The MPDIoU is compared with five mainstream boundary box regression loss calculation methods, obtaining the highest value of mAP0.5. In addition, in 5 of the 10 categories of the VisDrone dataset, MFEFNet achieves the highest value, among which pedestrians, people, awning tricycles, and motors are all small objects. The highest values of other categories are similar to ours. The above results further demonstrate the superiority of our method for detecting multi-scale objects.

In addition, in order to further demonstrate the effectiveness of our proposed SIEM, present a visual comparison of the intermediate feature map after Stage 1. Figure 9 shows the comparison results. It can be clearly seen that when we add the SIEM after Stage 1, our model has stronger spatial information extraction ability. The SIEM can not only extract more feature information but also distinguish foreground information from background information effectively and reduce the interference of background information.

Table 5. Comparison results of mainstream bounding-box regression loss with MFEFNet for each category on the VisDrone dataset. "*" means our re-implemented results.

Method	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awning Tricy-cle	Bus	Motor	mAP0.5
DIoU [38] *	59.3	50.1	24.6	85.7	54.5	49.1	40.7	21.5	**65.9**	62.8	51.4
GIoU [56] *	59.7	50.4	**26.2**	**85.9**	54.3	49.0	40.4	22.2	65.0	62.3	51.5
Focal-EIoU [57] *	59.9	50.3	25.7	85.6	54.2	**49.7**	39.4	21.5	66.9	63.0	51.6
SIoU [58] *	59.1	50.5	25.5	85.7	53.8	48.4	39.4	22.2	64.9	61.9	51.1
CIoU [38] *	59.1	50.1	25.4	**85.9**	54.6	49.4	**41.2**	22.0	66.9	62.5	51.7
MPDIoU (Ours)	**59.9**	**51.2**	25.9	85.8	**55.3**	49.4	40.3	**23.2**	65.1	**63.1**	**51.9**

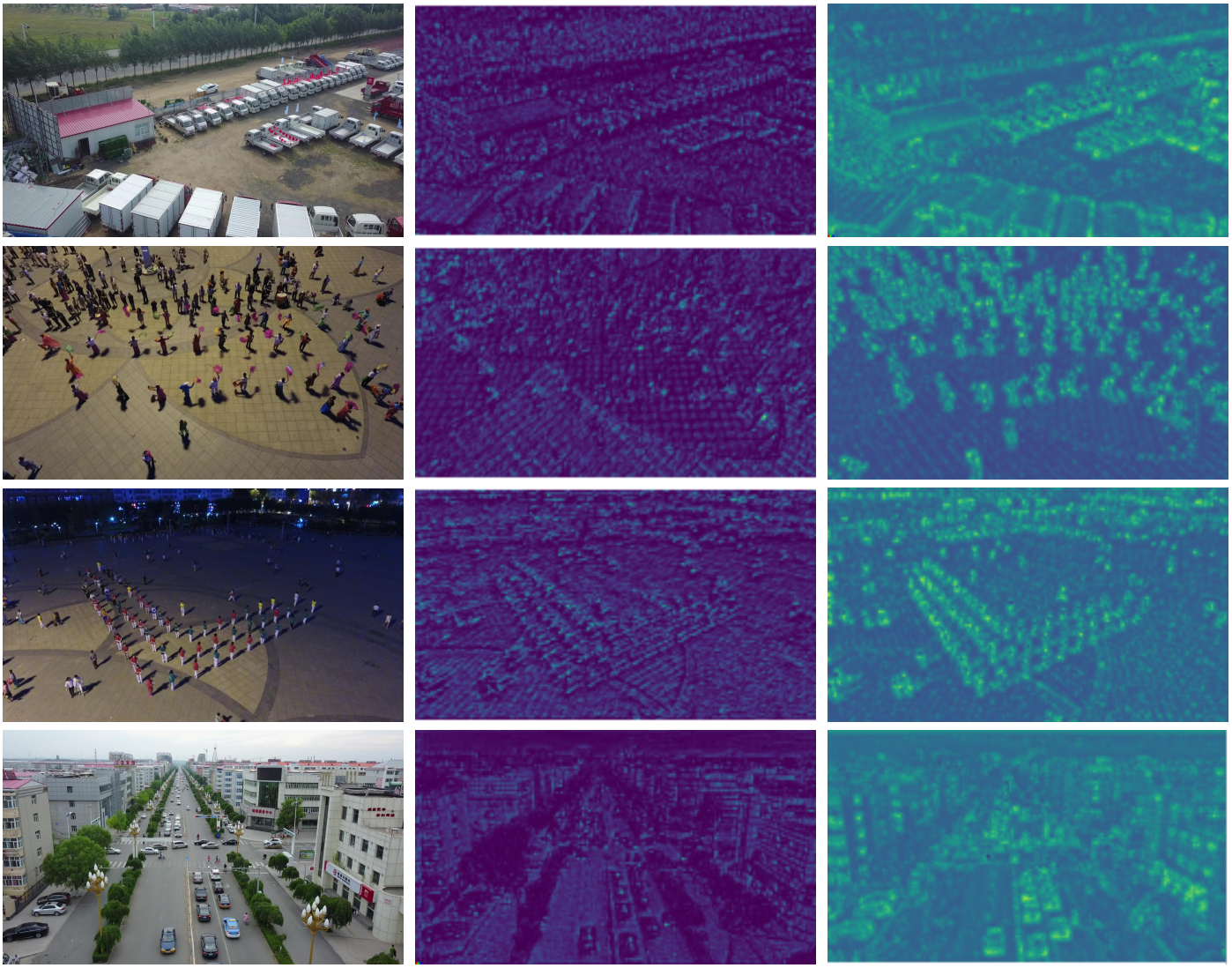


Figure 9. Intermediate feature maps. The first column is the original images; the second column is the intermediate feature map results without the SIEM; the third column is the intermediate feature map results with the SIEM.

To observe the effectiveness of our improved methods more directly, this study further uses an ablation experiment to discuss each method. The results of the ablation experiment are shown in Table 6 and analyzed as follows:

(1) SIEM: After adding the SIEM, the values of mAP0.5 and mAP0.5:0.95 on the VisDrone dataset increase by 2.2% and 1.6%, respectively. This shows that the SIEM can effectively extract the rich object representation information in the shallow backbone

network, establish the context-dependence relationship around the object, and reduce the negative impact of the background. Although the complex branch structure of the SIEM increases the number of parameters, this additional overhead is acceptable in terms of the benefits achieved.

(2) GAFN: To improve feature fusion efficiency and the detection effectiveness of multi-scale objects, we designed a feature fusion network, GAFN, based on AFFM. The addition of this structure improves the values of mAP0.5 and mAP0.5:0.95 on the VisDrone dataset by 0.3% and 0.2%, respectively. It is worth noting that this network greatly reduces the bloated degree of the original feature fusion network and the generation of redundant intermediate feature maps, resulting in a 3.3M reduction in the total number of parameters.

(3) MPDIoU: MPDIoU provides a new idea for us to design an object boundary regression loss function with simpler operation. The comparison results in Table 6 show that when CIoU is replaced by MPDIoU, the precision value of MFEFNet is improved, while the mAP0.5:0.95 value does not change. However, when we added the SIEM and GAFN to the experiment with MPDIoU, the value of mAP0.5 improves by 0.2%, and the value of precision improves by 2.7%. This proves that MPDIoU is useful for UAV image detection.

Table 6. Ablation experiments on the VisDrone dataset.

SIEM	GAFN	MPDIoU	Precision	Recall	mAP0.5	mAP0.5:0.95	Param (M)
			59.1	48.9	49.2	27.8	34.8
✓			60.0	51.3	51.4	29.4	36.9
	✓		57.9	50.0	49.5	28.0	**31.5**
		✓	59.6	48.2	49.2	27.8	34.8
✓	✓		59.3	**52.9**	51.7	**29.9**	33.6
✓	✓	✓	**62.0**	50.8	**51.9**	**29.9**	33.6

To further demonstrate the background interference resistance of our method, we use Grad-CAMTable [59] to generate heat maps of the model visualization results. Figure 10 shows the visualization results. From the first row picture, it can be seen that our method focuses on the sequential van better than YOLOv7. The second and third rows are for detecting densely distributed small objects in a complex background. It can be seen that our method can better eliminate the interference of the surrounding environment and pay more attention to the object. The fourth and fifth lines are multi-scale object detection on the road. It can be seen that YOLOv7 pays attention to both objects and unnecessary background information, while our model reduces the attention to unnecessary background information.

In order to test the effect of the values of λ_1, λ_2, and λ_3 in Formula (15) on the performance of MFEFNet, we perform a set of comparison experiments. The experimental results are shown in Table 7; precision and mAP0.5 reach the highest values when $\lambda_1 = 0.3$, $\lambda_2 = 0.05$, and $\lambda_3 = 0.7$. Recall reach the highest value when $\lambda_1 = 0.3$, $\lambda_2 = 0.03$, and $\lambda_3 = 0.7$, but precision and mAP0.5 present differences in terms of their highest values. In summary, our model achieves the best balance when $\lambda_1 = 0.3$, $\lambda_2 = 0.05$, and $\lambda_3 = 0.7$, so we choose them as the final values.

Table 7. The model's performance for different values of λ_i.

λ_1	λ_2	λ_3	Precision	Recall	mAP0.5
0.2	0.05	0.07	59.5	51.8	51.4
0.4	0.05	0.07	60.4	50.9	51.5
0.3	0.03	0.07	60.0	**51.9**	51.6
0.3	0.07	0.07	58.9	51.8	51.5
0.3	0.05	0.06	57.6	52.8	51.3

Table 7. *Cont.*

λ_1	λ_2	λ_3	Precision	Recall	mAP0.5
0.3	0.05	0.08	45.2	46.2	27.7
0.3	0.05	0.07	**62.0**	50.8	**51.9**

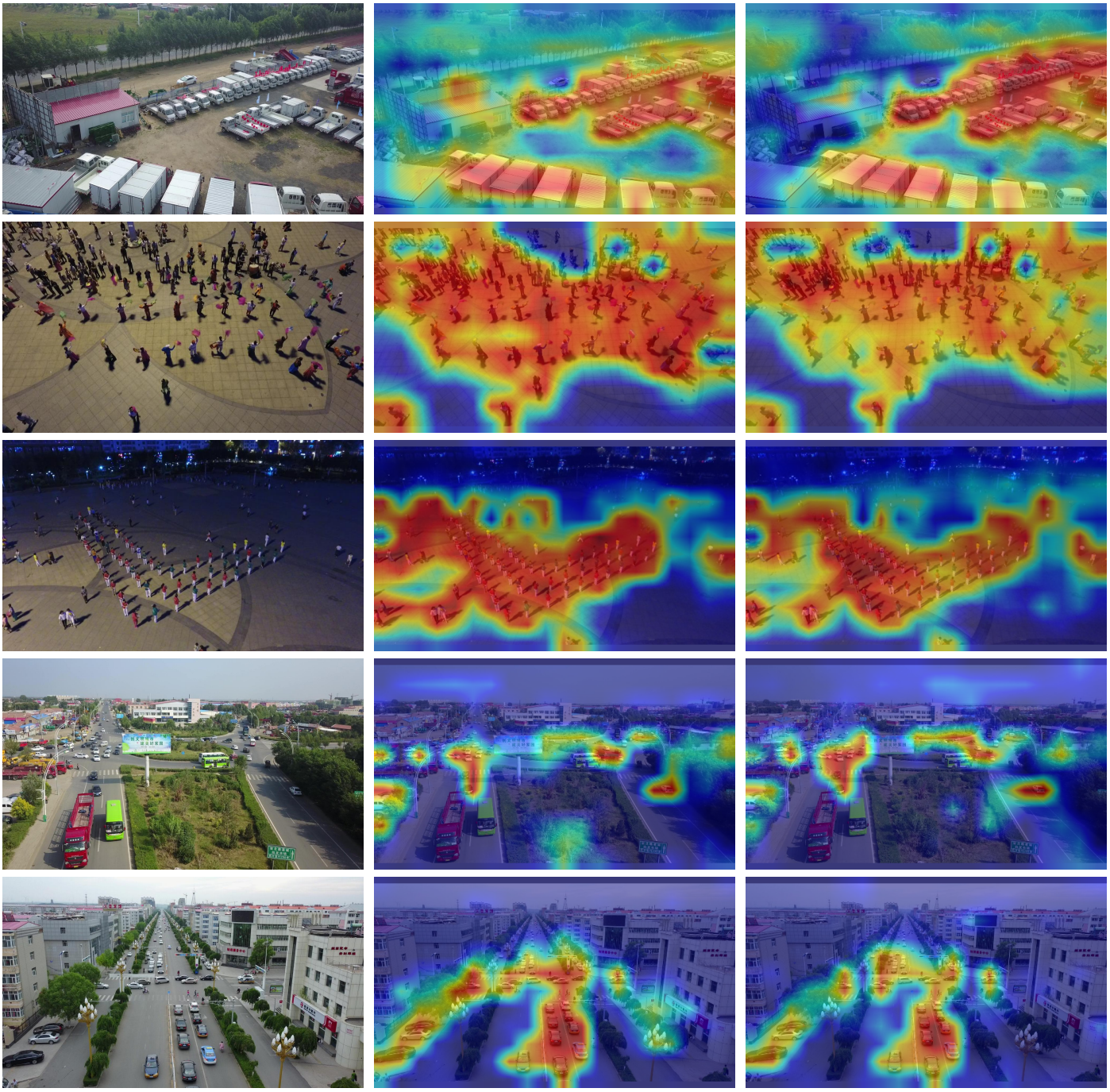


Figure 10. Visualization examples and heat maps. The first column is the original images; the second column is the visualization results of YOLOv7; the third column is the visualization results of MFEFNet.

4.5. Experimental Results on the UAVDT Dataset

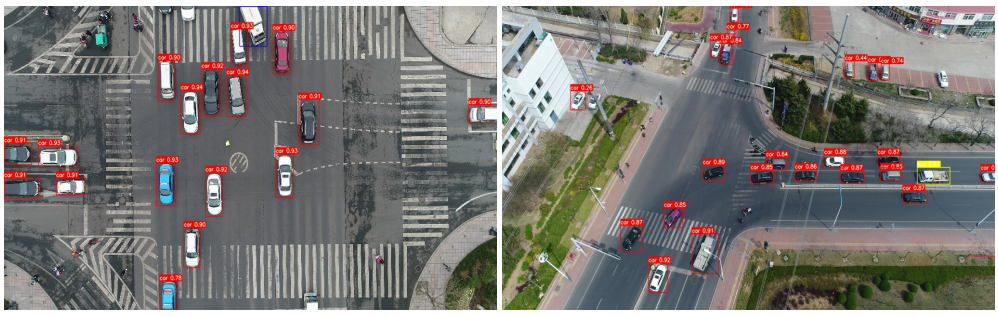
We also conduct comparative experiments on the UAVDT dataset. Table 8 shows the experimental results. Our MFEFNet has achieves excellent detection results on the UAVDT

dataset. Compared with YOLOv7, MFEFNet increases the mAP0.5 value by 2.2% and the mAP0.5:0.95 value by 1.9%. The average accuracy of the three types on the UAVDT dataset is improved by 2.7%, 1.6%, and 2.3%, respectively, compared with the baseline model. Compared to PRDet, our model has an input image size smaller than the 600 × 1000 that it used, but MFEFNet's values of mAP0.5 and mAP0.5:0.95 are 3.1% and 8.9% higher, respectively. In comparison with CFANet-s, the input picture size of MFEFNet is smaller than the 800 × 800 that it used, but our mAP0.5 value and mAP0.5:0.95 value are 2.5% and 2.2% higher, respectively. YOLOv5l also performs well on the UAVDT dataset, but our model outperforms in average accuracy and mAP values in various classes. In summary, our MFEFNet also achieves excellent detection performance on the UAVDT dataset.

Table 8. Comparison results for each category on the UAVDT dataset. "*" means our re-implemented results.

Method	Car	Truck	Bus	mAP0.5	mAP0.5:0.95
YOLOv3 [12]	30.8	3.9	26.4	36.3	20.4
YOLOX [46]	39.4	5.7	25.3	37.9	23.5
UFPMP-Det [60]	\	\	\	38.7	24.6
PRDet [34]	\	\	\	34.1	19.8
YOLOv5s [47]	78.1	13.3	45.6	45.0	26.5
YOLOv5l [47]	80.7	12.7	45.2	46.2	27.7
CFANet-s [33]	\	\	\	44.7	26.5
YOLOv7 [14] *	78.4	12.6	44.0	45.0	26.8
MFEFNet (Ours)	**81.1**	**14.2**	**46.3**	**47.2**	**28.7**

Figure 11 shows some visual inspection results for various scenarios on the UAVDT dataset. Figure 11a,b were detected on a clear day, and our model detects almost all of the objects in the images. The detection result of Figure 11c is under foggy weather; we can see that the vehicles on the main road are mostly accurately detected. Figure 11d,e are from a normal altitude level on a clear night, and Figure 11f is from a higher altitude at night; our model accurately detects the vehicles in these images.



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

(**c**) (**d**)

Figure 11. *Cont.*

(**e**) (**f**)

Figure 11. Visualization results of MFEFNet on the UAVDT dataset. All the subfigures (**a**–**f**) show the detection results in different environments.

5. Extended Experiments

We perform additional extended experiments using the DOTA-1.0 [61] dataset. The dataset contains 15 predefined categories, namely small vehicle (C1), large vehicle (C2), plane (C3), storage tank (C4), ship (C5), harbor (C6), ground track field (C7), soccer ball field (C8), tennis court (C9), swimming pool (C10), baseball diamond (C11), roundabout (C12), basketball court (C13), bridge (C14), and helicopter (C15). In the same experimental environment, we train the model using the training set and validate its performance using the validation set. The training set consists of 15,729 images, and the validation set consists of 5297 images. Table 9 shows the experimental results with state-of-the-art methods. Compared with YOLOv7, our method makes progress in the precision of nine categories of the DOTA dataset. Among the remaining six categories, except small vehicle and swimming pool, the accuracy of the remaining four categories exhibits little difference. In addition, the mAP0.5 value of our method improves by 0.8% compared with YOLOv7. However, the detection capability of our method on satellite remote sensing images is not as good as that of EAutoDet-s [62], FRIoU [63], and PCG-Net [64]. This is because MFEFNet is specifically designed for the features of UAV images, but those methods are specifically designed for the detection of satellite remote sensing images. Although our method has some shortcomings in performing satellite remote sensing image detection tasks, the mAP0.5 value of our method is only 2.1% lower than the best of them. In addition to this, we achieve the highest accuracy of plane, ship, harbor, and soccer ball field.

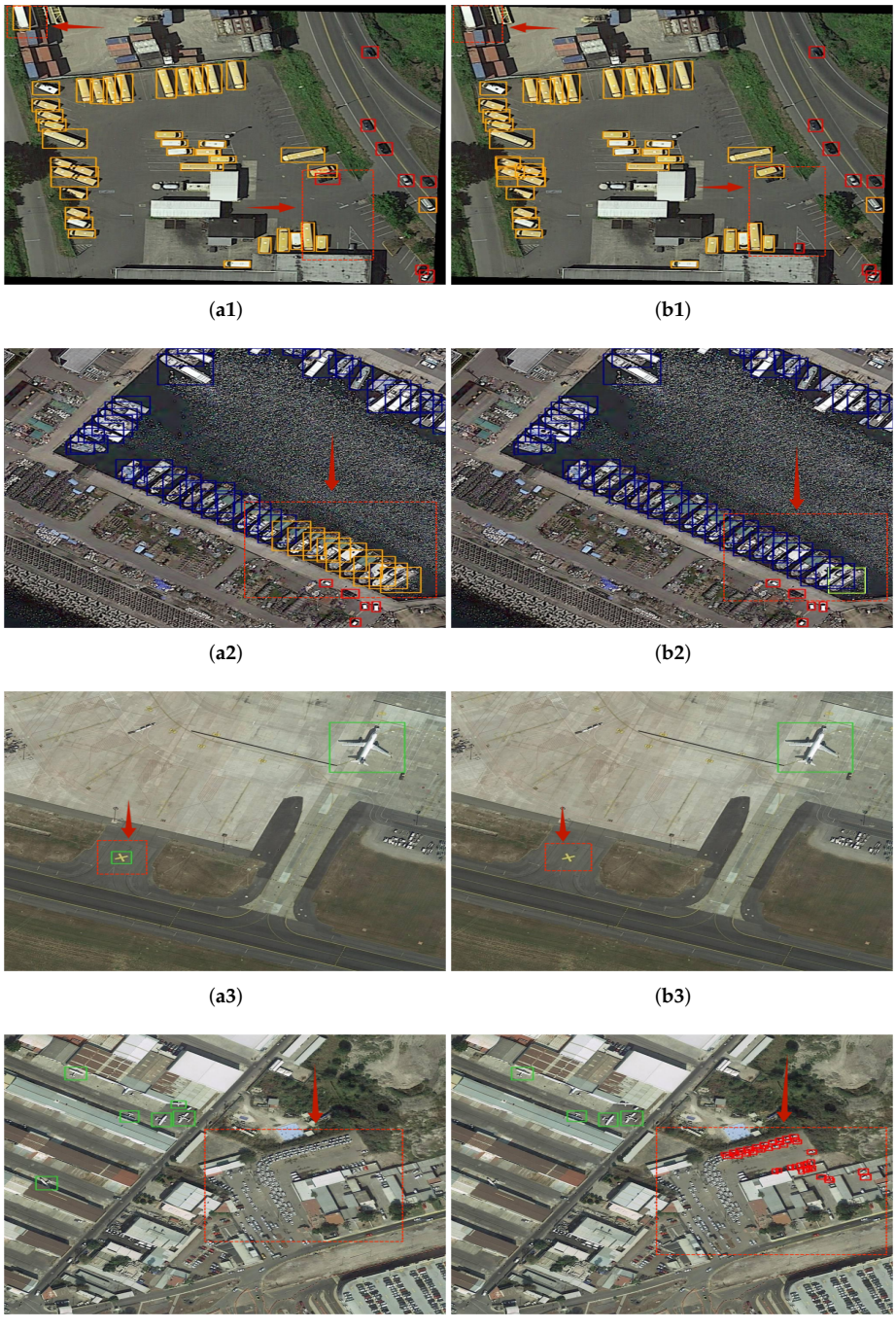
Table 9. Comparison results for each category on the DOTA-1.0 dataset. "*" means our re-implemented results.

Method	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	mAP
SASM [65]	77.3	76.0	86.4	85.7	86.7	74.0	69.9	60.1	90.9	72.2	79.0	68.2	82.6	52.5	62.4	74.9
EAutoDet-s [62]	**81.3**	84.0	88.6	**87.4**	88.2	73.7	62.0	52.5	90.8	**74.3**	**84.8**	65.7	85.8	51.0	**65.9**	75.7
FRIoU [63]	78.9	84.0	89.0	83.4	88.0	76.9	76.1	66.1	90.1	73.0	84.6	68.9	85.4	**54.5**	54.7	77.0
PCG-Net [64]	80.0	84.4	89.6	85.7	88.4	75.1	**77.2**	65.6	90.1	71.8	82.7	**69.1**	**86.1**	**54.5**	62.8	**77.6**
YOLOv7 [14] *	74.5	**88.1**	93.8	79.1	89.1	86.3	69.3	71.3	**94.9**	65.1	76.5	57.4	71.3	48.8	54.2	74.7
MFEFNet (Ours)	70.2	87.8	**94.4**	78.9	**89.3**	**86.5**	70.2	**71.6**	94.8	63.1	77.0	61.7	71.1	52.3	63.3	75.5

We show the visualization comparison results of YOLOv7 and our method in Figure 12 on the DOTA dataset. By paying attention to the red dotted boxes and red arrows in each group of pictures, we find that in Figure 12(a1,b1), YOLOv7 misdetects the cargo box in the upper left as a large vehicle and the debris in the lower right as a small vehicle. It also misses one small vehicle. However, none of our methods produces these errors. In In Figure 12(a2,b2), YOLOv7 misdetects a large number of ships in the bottom right as large vehicles, while our method only misdetects one ship at the bottom. In Figure 12(a3,b3), YOLOv7 misdetects a ground sign as a plane. In Figure 12(a4,b4), YOLOv7 fails to detect

a significant number of small vehicles in the lower-right region. The above experiments prove that our method has good generalization ability.

However, our method does not achieve precision improvements in some small object categories on the DOTA-1.0 dataset. According to the analysis, different kinds of small objects are affected by the background environment differently. Although MFEFNet can successfully extract the object location information and construct the context relationship, it is easy to lose small object sensitivity in the face of remote sensing images with a wider field of view, especially when such small objects are scattered near a building. Because of this, although our method achieves good performance on remote sensing datasets, it is better suited for object detection from the perspective of UAVs.



(**a1**) (**b1**)

(**a2**) (**b2**)

(**a3**) (**b3**)

(**a4**) (**b4**)

Figure 12. Visualization comparison results between YOLOv7 (**a1**–**a4**) and MFEFNet (**b1**–**b4**) on the DOTA-1.0 dataset. The obvious contrast parts is marked with red dotted boxes and red arrows.

6. Discussion

As can be seen from the results of the ablation experiment presented in Table 6, when the spatial information extraction module (SIEM) proposed by us is inserted into the most suitable position, the performance of our model for UAV image object detection is greatly improved. This proves that research in this direction should focus on the mining of image spatial information and context information. It can be seen from the results of the comparison experiment on the feature fusion network presented in Table 4 that popular feature fusion networks are not suitable for UAV object detection. Our proposed GAFN improves the detection performance and parameter size of the model, but there is still a lot of room for improvement in the future. Our MFEFNet has certain limitations. In an intelligent environment, UAVs, as a kind of widely used edge equipment, carryout an increasing number of tasks, and the demand for autonomous real-time detection of UAVs is increasing. However, their limited carrying capacity requirements put more demands on the model's lightweight size. Although our method reduces the size of the baseline model, it is not sufficient to meet the payload requirements of the UAV platform. Therefore, we will further reduce the burden of the model with the aim of ensuring accuracy.

7. Conclusions

To solve the object detection problem in UAV aerial images, we designed a high-precision object detection method based on the single-stage algorithm. First, we designed an innovative spatial information extraction module(SIEM), which is used to extract the location information of the object and construct the context relationship in the shallow feature map. Secondly, we designed a global aggregation progressive feature fusion network, which can efficiently fuse the location information and semantic information of the object and improve the capability to detect objects at various scales. Finally, we use MPDIoU as our bounding-box regression loss function to reduce the computational complexity of IoU loss while improving the average detection accuracy.

A large number of experiments in this paper show that our proposed SIEM can effectively extract the spatial location information of the target in the aspect of feature extraction. In terms of feature fusion, our proposed GAFN can effectively fuse spatial and semantic information of four different scale feature maps in the backbone network. In general, the mAP0.5 value of our proposed MFEFNet on the VisDrone and UAVDT datasets is improved by 2.7% and 2.2%, respectively.

Although our method has achieved some progress in UAV images, there is some feature redundancy in the feature extraction stage of the network, which causes some unnecessary burdens. Therefore, we will explore the relationship between different channels in the deep feature map in the future, which will help us reduce and utilize these redundant features. In addition, we will also consider using dilation convolution, which can enlarge the receptive domain while reducing the size of the model. At the same time, we will further explore the similarity between UAV images and other images to improve the generalization ability of the model.

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

Funding: This work was supported by the National Natural Science Foundation of China (Grant nos. 62176087, 62106066), the Key Research Projects of Henan Higher Education Institutions (Grant no. 22A520019), the Key Research and Promotion Projects of Henan Province (Grant nos. 232102210009, 242102210081), and the Henan Province Science Foundation of Excellent Young Scholars (Grant no. 242300421171).

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: We sincerely thank the anonymous reviewers for the critical comments and suggestions for improving the manuscript.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
CNN	Convolutional Neural Network
R-CNN	Regions with CNN Features
SSD	Single-Shot MultiBox Detector
YOLO	You Only Look Once
SIEM	Spatial Information Extraction Module
GAFN	Global Aggregation Progressive Feature Fusion Network
MFEFNet	Multi-scale Feature information Extraction and Fusion Network
AFFM	Adaptive Feature Fusion Module
FPN	Feature Pyramid Network
PANet	Path Aggregation Network
BIFPN	Bidirectional Feature Pyramid Network
AFPN	Asymptotic Feature Pyramid Network
IOU	Intersection over Union
P	Precision
R	Recall
AP	Average Precision
mAP	mean Average Precision

References

1. Zhao, H.; Chen, J.; Wang, L.; Lu, H. ARKitTrack: A New Diverse Dataset for Tracking Using Mobile RGB-D Data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 5126–5135.
2. Cao, Y. VisDrone-DET2021: The vision meets drone object detection challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 2847–2854.
3. Zhao, T.; Ning, X.; Hong, K.; Qiu, Z. Ada3D: Exploiting the Spatial Redundancy with Adaptive Inference for Efficient 3D Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023.
4. Gan, B. Collaborative Noisy Label Cleaner: Learning Scene-aware Trailers for Multi-modal Highlight Detection in Movies. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 18898–18907.
5. Li, J.; Xu, Y.; Nie, K.; Cao, B.; Zuo, S.; Zhu, J. PEDNet: A Lightweight Detection Network of Power Equipment in Infrared Image Based on YOLOv4-Tiny. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–12. [CrossRef]
6. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierar-chies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–24 June 2014; pp. 580–587.
7. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
9. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
10. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
11. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–300 June 2016; pp. 779–788.
12. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
13. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
14. 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. *arXiv* **2022**, arXiv:2207.02696.
15. Liu, W. SSD: Single shot MultiBox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016, Proceedings, Part I 14*; Springer: Cham, Switzerland, 2016; pp. 21–37.
16. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef]

17. Zhang, H.; Sun, M.; Li, Q.; Liu, L.; Liu, M.; Ji, Y. An empirical study of multi-scale object detection in high resolution UAV images. *Neurocomputing* **2021**, *421*, 173–182. [CrossRef]
18. Zhang, L.; Zhang, N.; Shi, R.; Wang, G.; Xu, Y.; Chen, Z. SG-Det: Shuffle-GhostNet-Based Detector for Real-Time Maritime Object Detection in UAV Images. *Remote Sens.* **2023**, *15*, 3365. [CrossRef]
19. Ye, T.; Qin, W.; Zhao, Z.; Gao, X.; Deng, X.; Ouyang, Y. Real-Time Object Detection Network in UAV-Vision Based on CNN and Transformer. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–13. [CrossRef]
20. Wu, Q.; Zhang, B.; Guo, C.; Wang, L. Multi-Branch Parallel Networks for Object Detection in High-Resolution UAV Remote Sensing Images. *Drones* **2023**, *7*, 439. [CrossRef]
21. Wang, X.; He, N.; Hong, C.; Wang, Q.; Chen, M. Improved YOLOX-X based UAV aerial photography object detection algorithm. *Image Vis. Comput.* **2023**, *135*, 104697. [CrossRef]
22. Huang, S.; Ren, S.; Wu, W.; Liu, Q. Discriminative features enhancement for low-altitude UAV object detection. *Pattern Recognit.* **2024**, *147*, 110041. [CrossRef]
23. Ma, S.; Xu, Y. MPDIoU: A Loss for Efficient and Accurate Bounding Box Regression. *arXiv* **2023**, arXiv:2307.07662.
24. Qu, J.; Tang, Z.; Zhang, L.; Zhang, Y.; Zhang, Z. Remote Sensing Small Object Detection Network Based on Attention Mechanism and Multi-Scale Feature Fusion. *Remote Sens.* **2023**, *15*, 2728. [CrossRef]
25. Wang, J.; Xu, C.; Yang, W.; Yu, L. A normalized Gaussian Wasser-stein distance for tiny object detection. *arXiv* **2021**, arXiv:2110.13389
26. Li, X.; Diao, W.; Mao, Y.; Gao, P.; Mao, X.; Li, X.; Sun, X. OGMN: Occlusion-guided multi-task network for object detection in UAV images. *ISPRS J. Photogramm. Remote Sens.* **2023**, *199*, 242–257. [CrossRef]
27. Lin, T.-Y.; Dollar, 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.
28. 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.
29. 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.
30. Yang, G.; Lei, J.; Zhu, Z.; Cheng, S.; Feng, Z.; Liang, R. AFPN: Asymptotic Feature Pyramid Network for Object Detection. In Proceedings of the 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, HI, USA, 1–4 October 2023.
31. Zhu, P. Detection and tracking meet drones challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7380–7399. [CrossRef]
32. Fang, W.; Zhang, G.; Zheng, Y.; Chen, Y. Multi-Task Learning for UAV Aerial Object Detection in Foggy Weather Condition. *Remote Sens.* **2023**, *15*, 4617. [CrossRef]
33. Zhang, Y.; Wu, C.; Guo, W.; Zhang, T.; Li, W. CFANet: Efficient Detection of UAV Image Based on Cross-Layer Feature Aggregation. *IEEE Trans. Geo. Remote Sens.* **2023**, *61*, 5608911. [CrossRef]
34. Leng, J.; Mo, M.; Zhou, Y.; Gao, C.; Li, W.; Gao, X. Pareto Refocusing for Drone-View Object Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 1320–1334. [CrossRef]
35. Lu, W. A CNN-Transformer Hybrid Model Based on CSWin Transformer for UAV Image Object Detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 1211–1231. [CrossRef]
36. Chalavadi, V.; Jeripothula, P.; Datla, R.; Ch, S.B. MSODANet: A network for multi-scale object detection in aerial images using hierarchical dilated convolutions. *Pattern Recognit.* **2022**, *126*, 108548. [CrossRef]
37. Shen, H.; Lin, D.; Song, T. Object Detection Deployed on UAVs for Oblique Images by Fusing IMU Information. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 6505305. [CrossRef]
38. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-iou loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
39. Sun, Z.; Leng, X.; Lei, Y.; Xiong, B.; Ji, K.; Kuang, G. BiFA-YOLO: A Novel YOLO-Based Method for Arbitrary-Oriented Ship Detection in High-Resolution SAR Images. *Remote Sens.* **2021**, *13*, 4209. [CrossRef]
40. Du, D. 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, 28–29 October 2019; pp. 213–226.
41. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The unmanned aerial vehicle benchmark: Object detection and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 370–386.
42. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, D.; Zitnick, C.L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014, Proceedings, Part V 13*; Springer: Cham, Switzerland, 2014; pp. 740–755.
43. Zhou, L.; Liu, Z.; Zhao, H.; Hou, Y.-E.; Liu, Y.; Zuo, X.; Dang, L. A Multi-Scale Object Detector Based on Coordinate and Global Information Aggregation for UAV Aerial Images. *Remote Sens.* **2023**, *15*, 3468. [CrossRef]
44. Ultralytics. YOLOv8. Available online: https://github.com/ultralytics/ultralytics (accessed on 1 January 2023).
45. Li, C.; Yang, T.; Zhu, S.; Chen, C.; Guan, S. Density Map Guided Object Detection in Aerial Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, New Seattle, WA, USA, 14–19 June 2020; pp. 737–746.
46. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
47. Ultralytics. Yolov5. Available online: https://github.com/ultralytics/yolov5 (accessed on 18 June 2022).

48. Lin, H.; Zhou, J.; Gan, Y.; Vong, C.; Liu, Q. Novel up-scale feature aggregation for object detection in aerial images. *Neurocomputing* **2020**, *411*, 364–374. [CrossRef]
49. Yang, C.; Huang, Z.; Wang, N. QueryDet: Cascaded Sparse Query for Accelerating High-Resolution Small Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13658–13667.
50. Liu, S.; Zha, J.; Sun, J.; Li, Z.; Wang, G. EdgeYOLO: An Edge-Real-Time Object Detector. In Proceedings of the 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 24–26 July 2023; Volume 6, pp. 7507–7512.
51. Li, Z. Aerial Image Object Detection Method Based on Adaptive ClusDet Network. In Proceedings of the IEEE 21st International Conference on Communication Technology (ICCT), Tianjin, China, 13–16 October 2021; pp. 1091–1096.
52. Du, B.; Huang, Y.; Chen, J.; Huang, D. Adaptive Sparse Convolutional Networks with Global Context Enhancement for Faster Object Detection on Drone Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 13435–13444.
53. Duan, C.; Wei, Z.; Zhang, C.; Qu, S.; Wang, H. Coarse-grained density map guided object detection in aerial images. In Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 2789–2798.
54. Min, L.; Fan, Z.; Lv, Q.; Reda, M.; Shen, L.; Wang, B. YOLO-DCTI: Small Object Detection in Remote Sensing Base on Contextual Transformer Enhancement. *Remote Sens.* **2023**, *15*, 3970. [CrossRef]
55. Xu, H.; Zheng, W.; Liu, F.; Li, P.; Wang, R. Unmanned Aerial Vehicle Perspective Small Target Recognition Algorithm Based on Improved YOLOv5. *Remote Sens.* **2023**, *15*, 3583. [CrossRef]
56. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
57. Zhang, Y.-F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and Efficient IOU Loss for Accurate Bounding Box Regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]
58. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740.
59. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 618–626.
60. Huang, Y.; Chen, J.; Huang, D. UFPMP-Det: Toward accurate and efficient object detection on drone imagery. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 1026–1033.
61. Xia, G.-S. 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.
62. Wang, X.; Lin, J.; Zhao, J.; Yang, X.; Yan, J. EAutoDet: Efficient Architecture Search for Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 668–684.
63. Qian, X.; Wu, B.; Cheng, G.; Yao, X.; Wang, W.; Han, J. Building a bridge of bounding box regression between oriented and horizontal object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5605209. [CrossRef]
64. Niu, T.; He, X.; Chen, H.; Qing, L.; Teng, Q. Semantic and geometric information propagation for oriented object detection in aerial images. *Appl. Intell.* **2024**, *54*, 2154–2172. [CrossRef]
65. Hou, L.; Lu, K.; Xue, J.; Li, Y. Shape-adaptive selection and measurement for oriented object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 3974–3983.