algorithms

MDPI

Article

AFE-YOLOv8: A Novel Object Detection Model for Unmanned Aerial Vehicle Scenes with Adaptive Feature Enhancement

Shijie Wang, Zekun Zhang, Qingqing Chao and Teng Yu *

College of Electronic Information, Qingdao University, Qingdao 266071, China; 2021023782@qdu.edu.cn (S.W.); 2021020655@qdu.edu.cn (Z.Z.); 2021023773@qdu.edu.cn (Q.C.)
* Correspondence: yuteng@qdu.edu.cn

Abstract: Object detection in unmanned aerial vehicle (UAV) scenes is a challenging task due to the varying scales and complexities of targets. To address this, we propose a novel object detection model, AFE-YOLOv8, which integrates three innovative modules: the Multi-scale Nonlinear Fusion Module (MNFM), the Adaptive Feature Enhancement Module (AFEM), and the Receptive Field Expansion Module (RFEM). The MNFM introduces nonlinear mapping by exploiting the property that deformable convolution can dynamically adjust the shape of the convolution kernel according to the shape of the target, and it effectively enhances the feature extraction capability of the backbone network by integrating multi-scale feature maps from different mapping branches. Meanwhile, the AFEM introduces an adaptive fusion factor, and through the fusion factor, it adaptively integrates the small-target features contained in the feature maps of different detection branches into the small-target detection branch, thus enhancing the expression of the small-target features contained in the feature maps of the small-target detection branch. Furthermore, the RFEM expands the receptive field of the feature maps of the large- and medium-scale target detection branches through stacked convolution, so as to make the model's receptive field cover the whole target, and thereby learn more rich and comprehensive features of the target. The experimental results demonstrate the superior performance of the proposed model compared to the baseline in detecting objects of various scales. On the VisDrone dataset, the proposed model achieves a 4.5% enhancement in mean average precision (mAP) and a 5.45% improvement in average precision at an IOU threshold of 0.5 (AP50). Additionally, ablation experiments conducted on the challenging DOTA dataset showcase the model's robustness and generalization capabilities.

Keywords: object detection; deformable convolution; unmanned aerial vehicle; adaptive feature enhancement

Citation: Wang, S.; Zhang, Z.; Chao, Q.; Yu, T. AFE-YOLOv8: A Novel Object Detection Model for Unmanned Aerial Vehicle Scenes with Adaptive Feature Enhancement. *Algorithms* **2024**, *17*, 276. https://doi.org/10.3390/a17070276

Academic Editor: Frank Werner

Received: 17 May 2024
Revised: 17 June 2024
Accepted: 21 June 2024
Published: 24 June 2024

1. Introduction

In recent years, there has been significant advancement in convolutional neural network (CNN)-based object detection [1–5], driven by the continuous development in deep learning [6,7]. Object detection plays a crucial role in computer vision tasks, as it combines classification and localization, laying the foundation for various challenges such as semantic segmentation, image understanding, and object tracking. Furthermore, object detection is extensively employed in various applications such as autonomous driving, facial recognition, UAV navigation, intelligent transportation systems, and remote sensing image analysis.

In contrast to general object detection methods, our research concentrates on drone aerial object detection, an inherently more challenging task due to the presence of numerous small and multi-scale objects. Consequently, enhancing the ability of the backbone network to extract features of small objects and reinforcing the representation of small object features are the primary challenges in object detection under UAV scenes.

Current state-of-the-art (SOTA) detection networks can be classified into one-stage and two-stage models [8–10]. The two-stage detector initially generates region proposals

for unknown categories and then conducts classification and regression on these candidate regions to achieve object classification and localization, exemplified by Faster R-CNN [11], Cascade R-CNN [12], and so on.

To achieve highly efficient real-time object detection, the one-stage detector performs both regression and classification tasks simultaneously. This approach simplifies the model's complexity, leading to improved inference speed, such as in the YOLO series [13–17] and RetinaNet [18]. While these detection models have demonstrated remarkable performances on natural image datasets, their application to UAV aerial images often results in significant degradation in the detection performance, primarily due to insufficient feature detail expression.

T.-Y. Lin et al. proposed the Feature Pyramid Network (FPN) [19] to create a multi-layer feature map by incorporating deep and shallow features, which can facilitate subsequent object detection tasks. However, in scenarios where large objects coexist in close proximity to small objects, the large objects' saliency tends to overshadow the features associated with the smaller objects. This results in a weakened representation of the small objects' features, leading to a less effective detection of these objects.

To address the challenges mentioned above, this paper proposes a novel anchor-free design, the AFE-YOLOv8 model, for object detection from images captured in UAV scenes. The model introduces three innovative structures: the small-object-aware module called AFEM, salient feature enhancement module named MNFM, and the multi-scale object-matching module called RFEM as shown in Figure 1. Our AFEM draws inspiration from the detection networks of QueryDet [20] and TSODE [21], which are well established to encompass more high-level semantic information pertaining to small objects compared to shallow layers. The AFEM aims to tackle the issue of inadequate feature expression in the small-object pipeline by enhancing the feature representation of the small-object pipeline through utilizing high-level semantic information contained in the medium- and large-object pipelines' features. Furthermore, we introduce the Multi-scale Nonlinear Fusion Module (MNFM) to enhance the detection performance of small targets by capturing salient feature representations through nonlinear mapping and multi-scale fusion operations. The nonlinear mapping unit of the Nonlinear Fusion Module (NFM) is constructed using stacked deformable convolutions [22,23], which can adaptively adjust the shape of the convolution kernel, making it highly suitable as a fundamental unit for nonlinear mapping in order to extract salient features of the small target. Additionally, to further solve the issue of multi-scale object detection, we integrated the Extended Efficient Layer Aggregation Network (E-ELAN) structure from YOLOv7 [17] and designed the Receptive Field Expansion Module (RFEM). The RFEM aims to model contextual information and expand the receptive field of convolution kernels. It is strategically positioned before the large- and medium-object detection heads to ensure consistency between the receptive fields and the detection heads at varying scales.

We summarize the main contributions of this work as follows:

1. Our proposed AFEM enhances small-object feature representation and model awareness of small objects by adaptively assigning fusion weights in the feature fusion process, incorporating small-object features from various branches into the small-object pipeline.
2. The RFEM models contextual information to increase the receptive field and maintain feature diversity for the improved detection of medium and large objects. The SE branch selectively enhances useful channel features and suppresses irrelevant ones through global features, enhancing the network's feature expression capability.
3. The MNFM constructs nonlinear mapping units using deformable convolution to extract more salient features from target objects. It fuses feature maps of different scales through cascading and cross-layer connections to obtain comprehensive and rich feature representations of small targets, embedded into the backbone network to replace the C2f module.

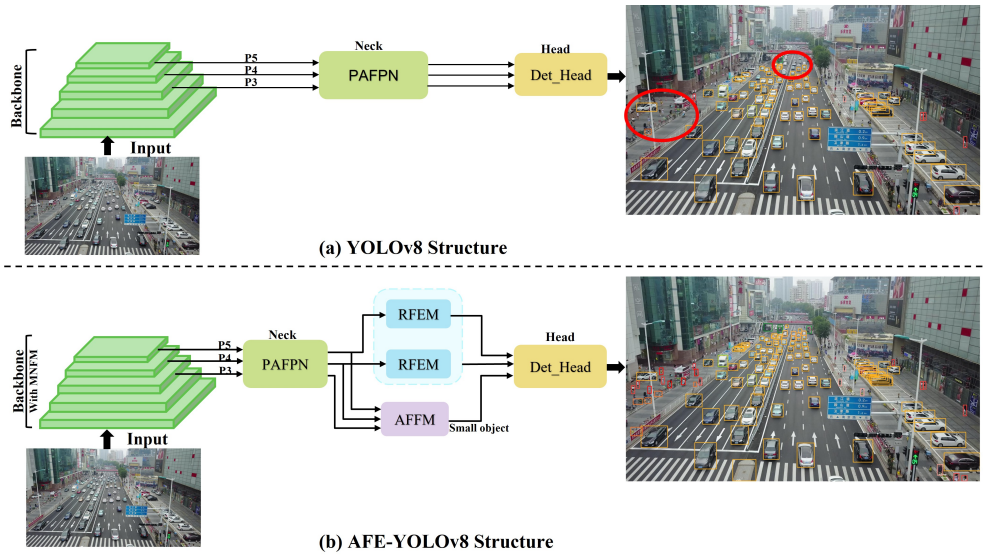


Figure 1. The pipeline of the proposed method, and the targets in the red circle are YOLOv8 missed detections. The main contributions of our approach include the AFEM, the RFEM, and the improved backbone with the MNFM.

2. Related Works

2.1. General Object Detection

The general object detection models, which primarily comprise deep learning and traditional models, have the potential to detect multiple object classes in various scenes. In the last decade, there has been significant progress in object detection, particularly in deep learning-based models. These models typically employ CNNs to extract features and classify images, thereby achieving an accurate localization and classification of the target objects. Vision detectors based on deep learning can be broadly categorized into two categories: one-stage models and two-stage models. The two-stage models appeared earlier, mainly including R-CNN [9], Fast R-CNN [10], Faster R-CNN [11], etc. The R-CNN, initially introduced in 2014, is a CNN-based method that utilizes the selective search algorithm for candidate box extraction and performs convolution for feature extraction in both classification and regression tasks. To accelerate the detection process of the R-CNN, Girshick et al. introduced Fast R-CNN. This method introduced an ROI pooling layer, enabling the feature extraction and classification of multiple regions of interest (ROIs) in a single forward pass. Additionally, a multitask loss function was employed to optimize both classification and regression tasks, resulting in an improved detection performance. Subsequently, Faster R-CNN was introduced, incorporating the Region Proposal Network (RPN) [11] to replace the selective search algorithm. The RPN allowed for the direct learning of candidate box coordinates and categories from extracted feature maps, facilitating real-time object detection. Moreover, it effectively addresses the challenges posed by multi-scale variations, which previous algorithms struggled to handle.

The demand for real-time object detection has accelerated the rapid development of one-stage object detection. One-stage models mainly include SSD [24], RetinaNet [18], the YOLO series [13–17], etc. The SSD algorithm revolutionized the detection task by formulating it as a regression problem and incorporating a feature pyramid to enable accurate object predictions on feature maps with varying receptive fields. RetinaNet tackles the challenge of imbalanced positive and negative samples in one-stage detectors. It introduced a novel loss function called Focal Loss, which assigns appropriate weights to positive and negative samples. This innovation allows RetinaNet to achieve both faster processing times than other one-stage detectors and a higher accuracy than two-stage detectors.

The YOLO (You Only Look Once) algorithm is a real-time object detection technique that treats the entire image as the network input to predict the class and location of multiple objects directly. YOLOv2 [14] improved the detection accuracy through the incorporation of a deeper network structure, Batch Normalization, Anchor Boxes, and other techniques. Building upon YOLOv2, YOLOv3 [15] further enhanced the algorithm by introducing multi-scale detection and Darknet-53 features to improve in performance.

YOLOv4 [16] implemented new technologies such as CSPDarknet53, SPP Block, a novel neck, and an innovative head based on YOLOv3. YOLOv5 incorporates techniques such as lightweight models, data augmentation, and adaptive training strategies for faster detection and higher accuracy. The YOLOv6 model implements optimizations including the application of deeper convolutional neural networks, the addition of data augmentation techniques, and the adoption of smaller anchor boxes. Compared to previous versions, YOLOv7 [17] adopts a deeper network structure and introduced several new optimization techniques to improve the detection accuracy and speed. YOLOv8 [25] introduces an attention mechanism and dynamic convolution and specifically improves on small-object detection to address the challenges highlighted in YOLOv7.

For the problem that current detection methods lose a large amount of feature information when performing layer-by-layer feature extraction and spatial transformation operations on the input image data, YOLOv9 [26] proposes the concept of programmable gradient information (PGI), which computes the objective function by providing complete input information for the target task, so as to obtain reliable gradient information to update the model weights. Moreover, the effectiveness of PGI on lightweight models was confirmed by designing a lightweight generalized effective layer aggregation network (GELAN) based on gradient path planning. Since the detection speed and accuracy of YOLO are affected by Non-Maximum Suppression (NMS) to a greater extent and the Transformer-based target detection algorithm DETR provides an alternative to NMS, RT-DETR [27] was designed as an efficient hybrid encoder for the fast processing of multi-scale features by decoupling intra-scale interaction and inter-scale fusion to improve the detection speed. In addition, an uncertainty-minimum query selection algorithm was proposed to provide the decoder with high-quality initial queries to improve the detection accuracy. Since the current YOLOs algorithm has achieved an effective balance between the computational cost and the detection performance, performing NMS processing during the deployment of the model will affect the inference process of the model. Therefore, YOLOv10 [28] proposes a consistent dual allocation strategy for NMS-free training, which simultaneously ensures high model performance and low inference latency. Moreover, v10 fully optimizes each component of YOLO from both efficiency and accuracy perspectives to reduce the computational overhead of the model while enhancing the detection performance.

Although these current detection models have demonstrated impressive performances across diverse scene datasets, their efficacy in detecting objects within UAV aerial datasets remains limited, indicating ample room for improvement. Compounding this challenge is the existence of numerous small objects and significant variations in object sizes within UAV aerial images. The distribution of instance samples and features is imbalanced across objects of different scales, and this imbalance is exacerbated by subsequent convolution and pooling operations, resulting in the loss of crucial features.

2.2. UAV Aerial Object Detection

Object detection facilitates UAVs to quickly and accurately identify target objects and provide data support for subsequent tasks. In recent years, there has been significant research on and applications of deep learning-based methods for UAV aerial object detection, driven by the continuous advancements in deep learning technology.

The challenge of UAV aerial object detection stems from the presence of numerous small objects in the dataset, where the objects exhibit varying scales, necessitating a detection system capable of detecting objects of different sizes. To address this challenge,

Akyon et al. [29] introduced an open-source framework called Slicing-Aided Hyper Inference (SAHI) that offers a comprehensive pipeline for small-object detection. The framework incorporates slice-assisted reasoning and fine-tuning techniques. However, this two-stage detection model, which involves slicing and subsequent detection, significantly increases the detection time, making it less suitable for deployment on edge devices. For QueryDet [20], the authors proposed a design concept to improve small-object detection by introducing a small-object query mechanism. They observed that even though the deep feature map may contain relatively less information about small objects, the FPN algorithm exhibits a highly structured nature. Consequently, even on a low-resolution feature map, the approximate location of a small object can still be confidently determined. Additionally, Wang et al. [4] discovered certain correlations between different feature layers. Intuitively, one can infer that the shallow layer often retains more location-related information, while the deep layer typically contains better semantic and classification-related information.

Moreover, the difficulty of multi-scale object detection arises from the mismatch between the object's scale and the receptive field of the detection head, as well as the introduction of incomplete or redundant feature information during traditional convolutional feature extraction. The Faceboxes algorithm [30] utilizes a multi-scale prediction and joint training approach. In other words, predictions are made at different scales, and the predictions from these various scales are trained jointly. This approach involves making predictions at different scales and jointly training the predictions from these scales, thereby enhancing the algorithm's robustness and generalization ability. Similarly, the SNIP algorithm [31] is an improved version of multi-scale training, which enables the model to concentrate more on the detection of the object itself, addressing the challenges associated with multi-scale learning. Additionally, RefineDet [32] leverages the multi-layer feature map network from SSD as the RPN of Faster R-CNN, combining the strengths of both methods. Similar to FPN in feature map processing, RefineDet employs deconvolution with element-wise summation to merge deep feature maps with shallow ones, facilitating the detection of multi-scale objects.

The SOTA research models in object detection make use of dilated convolutions to extract more comprehensive features of the object by increasing the receptive field. Li et al. initially proposed the impact of the receptive fields on objects of different scales in object detection tasks and conducted a comprehensive experimental verification to validate their observations. Leveraging the benefits of dilated convolutions in expanding the receptive field, they introduced TridentNet [33], a straightforward three-branch network. TridentNet demonstrated a significant enhancement in the accuracy of multi-scale object detection. Furthermore, Zhu et al. proposed a target detection model, TPH-YOLOv5 [34], for unmanned aerial vehicle scenarios. This model is based on the YOLOv5 detection framework and integrates a small-object detection pipeline and transformer structure to address the limitations of YOLO in detecting small targets and targets with drastic size variations in drone scenarios.

3. Proposed Method

3.1. Overall Architecture

We illustrate the comprehensive network architecture of AFE-YOLOv8 in Figure 2. The MNFMs are integrated into the backbone network for salient feature extraction. Subsequent to the neck part of the small-scale branch, the AFEM further adaptively enhances the feature representation of small objects. Additionally, the RFEMs within the medium-scale and large-scale branches increase the receptive field of objects in the deep-layer feature maps, effectively resolving the scale-receptive field mismatch of the detection head. As a result, the overall algorithm improves the detection accuracy of objects across small, medium, and large scales.

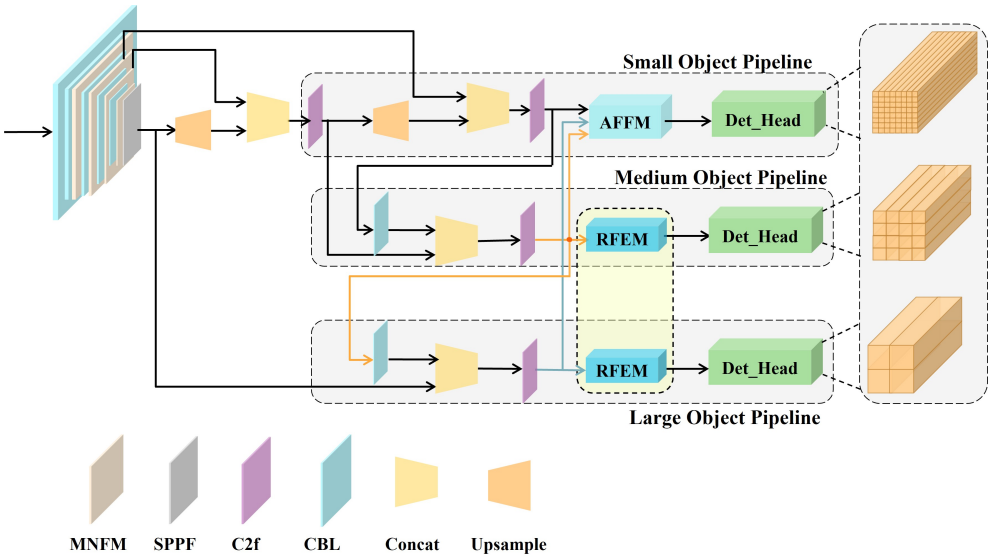


Figure 2. The overall structure of the AFE-YOLOv8 network.

As shown in Figure 3, the backbone network consists of consecutively stacked convolutional blocks and the MNFM. Among them, the convolution blocks consist of convolutions of size 3 × 3, batch normalization, and the RELU activation function, and the step size of the convolution blocks is 2, so that each time a convolution block passes through, a downsampling operation is conducted. The last part of the backbone network is the SPPF, which performs pooling operations on feature maps of different scales without changing the size of the feature maps in order to capture the multi-scale features of the target.

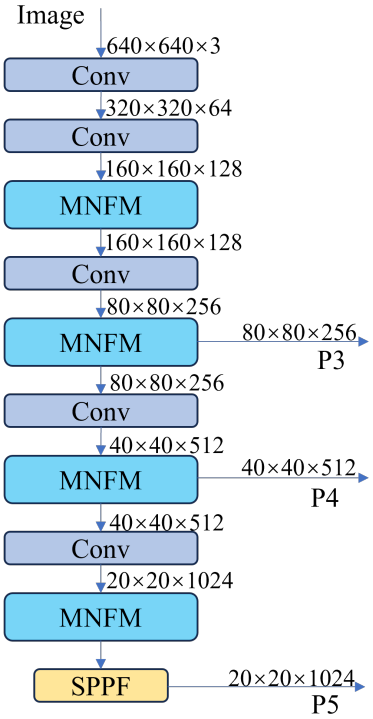


Figure 3. The architecture of the proposed backbone network.

3.2. Adaptive Feature Enhancement Module

As small-scale objects typically comprise fewer pixels, the detailed features of these small objects are often lost during the feature extraction process of the backbone network. Currently, most detection algorithms only focus on detecting small objects by utilizing

shallow feature maps in their frameworks. Although shallow feature maps contain detailed feature information about small objects, they also include unnecessary background features, leading to significant redundant computation during the detection process and a slower inference speed of the model. Wang et al. [35] and Zhuang et al. [21] proposed that deep feature maps contain more semantic information than shallow feature maps. However, deep feature maps have fewer pixels and larger receptive fields, which makes them less suitable for directly detecting small objects in the deep feature layer.

Therefore, we propose the Adaptive Feature Enhancement Module (AFEM) to enhance the features of the small-object pipeline by incorporating features from both the medium-object pipeline and the large-object pipeline. By integrating the small-object information from varying-scale-object pipelines into the feature maps of the small object pipeline, we can enrich the feature maps of the small-object pipeline with more detailed and semantic information. The underlying idea is that the contextual semantic information in the deep feature map can be utilized to guide shallow feature learning, thereby enhancing the details and semantic features of small objects through context modeling. In this manner, the AFEM enhances the saliency of features from small objects compared to those from the background, leading to improved accuracy in detecting small objects during the feature fusion process. The structure of the AFEM is illustrated in Figure 4.

Considering that deep layers predominantly contain more features from large-scale and medium-scale objects, they inherently possess less semantic information related to small objects. Consequently, it becomes imperative to assign lower weights to features from deep layers in order to enhance the features of small objects. Taking this into account, we introduce a learnable adaptive fusion weighting factor during the feature enhancement process.

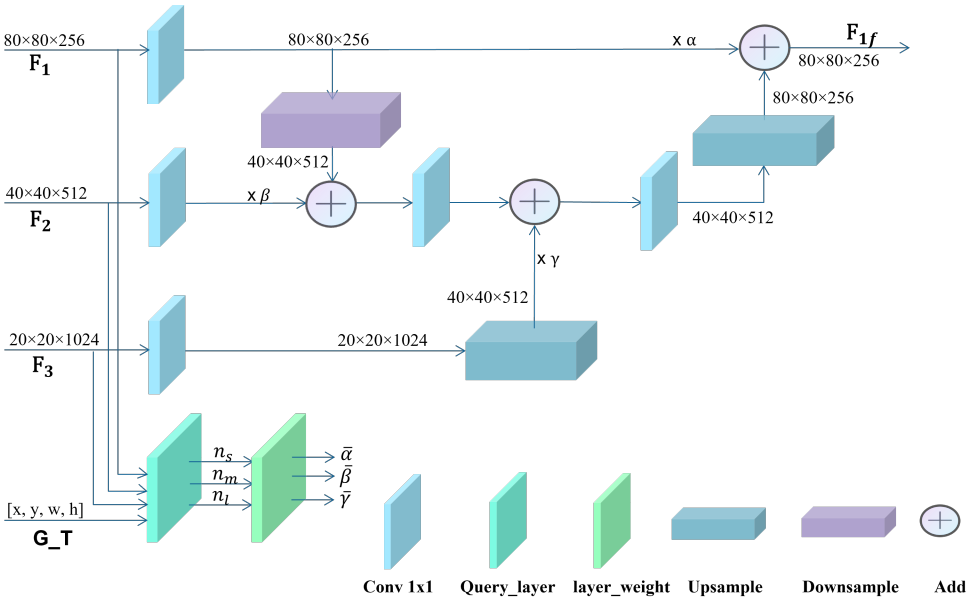


Figure 4. The architecture of the proposed Adaptive Feature Enhancement Module (AFEM).

The learnable adaptive fusion factors, denoted as α, β, and γ, are subject to the constraint that their sum is equal to 1. Specifically, α represents the contribution ratio of feature aggregation from the F_1 layer, β represents the contribution ratio of feature aggregation from the F_2 layer, and γ represents the contribution ratio of feature aggregation from the F_3 layer.

The implementation of this method consists of three steps. Firstly, we apply conv 1×1 and downsampling operations to the shallow feature map F_1, resulting in the feature map F_{1d} with the same size as the F_2 layer. Then, we multiply the F_2 layer by a learnable factor

β through a conv 1×1 operation and fuse it with F_{1d}, yielding the fused feature map F_{12}. The implementation formula is as follows:

$$F_{12} = \text{Conv}(F_2) \cdot \beta + \text{Downsample}(\text{Conv}(F_1)) \quad (1)$$

The second step involves the F_3 layer, which contains more semantic information. Convolution upsampling operations are performed, and then the results are multiplied by the learnable factor γ. The result is added to F_{12} for addition fusion, resulting in F_{23}. The formula is implemented as follows:

$$F_{23} = \text{Upsample}(\text{Conv}(F_3)) \cdot \gamma + F_{12} \quad (2)$$

The final step involves fusing the deep feature fusion result F_{23} with the shallow feature F_1; that is, F_{23} is multiplied by the learnable factor α after conv 1×1, while F_{23} is obtained from convolution upsampling operations. Finally, the two results are summed as F_{1f}. This U-shape design effectively preserves the detailed information in F_1, with minimal additional computational cost. The implementation formula is as follows:

$$F_{1f} = \text{Conv}(F_1) \cdot \alpha + \text{Upsample}(\text{Conv}(F_{23})) \quad (3)$$

3.3. Receptive Field Expansion Module

The challenge of detecting multi-scale objects lies in the inability of the feature map's receptive field to align with the scales of these objects. To deal with this problem, we propose the Receptive Field Expansion Module (RFEM), which enlarges the receptive field to accommodate multiple scales of objects. The specific implementation process is depicted in Figure 5.

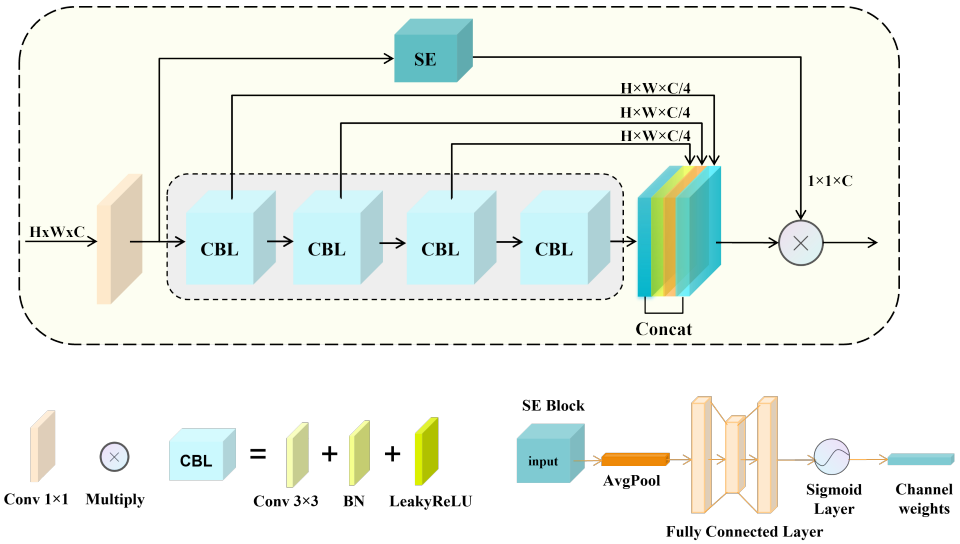


Figure 5. The architecture of the proposed Receptive Field Expansion Module (RFEM).

The motivation for this module derives from the architecture of the E-ELAN of YOLOv7. Through the expansion of the receptive field, we integrate features from different layers to gather more comprehensive gradient flow information, allowing the network to achieve improved accuracy while maintaining reasonable processing time. To achieve the objective of enhancing the receptive field, we employ stacked convolution layers in the main branch. Through skip connections, the feature layers with varying receptive fields are concatenated together, thereby enhancing the diversity of multi-receptive field features.

In our approach, we employ 3×3 convolutions for all convolution layers, and each layer generates an output with C/4 channels. These convolution blocks are stacked in a cascaded manner, allowing subsequent layers to reuse computations from the preceding

layers. This strategy not only reduces computational overhead but also effectively expands the receptive field. The incorporation of an ELAN-like structure ensures the accuracy and efficiency of our model while further enhancing the feature extraction capabilities for multi-scale objects. The output of the RFEM structure is

$$y_{\text{cat}} = \text{Concat}\{\text{CBL}(x), \text{CBL}(\text{CBL}(x)), \ldots\} \tag{4}$$

To selectively emphasize important channels and suppress less important ones, we introduce a Squeeze-and-Excitation (SE) branch, which begins by conducting global average pooling on the feature map layer, which yields the global feature vector G with dimensions $1 \times 1 \times C$. Subsequently, the global feature G is passed through a two-layer fully connected bottleneck structure. This compression step reduces the number of channels to 14 of the original count, followed by an expansion step that restores the number of channels back to the original value. The resulting weights represent the importance of each channel in the feature maps, and each channel is multiplied by its corresponding weight. The process of the SE module can be summarized as follows:

$$w = \text{Sigmoid}(\text{Liner}(\text{Liner}(\text{AvgPool}(x)))) \tag{5}$$

The entire module is positioned in front of the detection heads for medium-scale and large-scale objects, and the output through the RFEM is

$$y_{out} = w \cdot y_{cat} \tag{6}$$

3.4. Multi-Scale Nonlinear Fusion Module

There exists a large number of small objects and many other objects with multi-scales in the UAV dataset, resulting in the poor detection performance of YOLOv8 [25]. At the same time, the C2f module is not capable of learning global salient features because the conventional convolutions only aggregate features in the neighborhood. Therefore, we introduce the Multi-scale Nonlinear Fusion Module (MNFM), as shown in Figure 6, which combines multi-scale fusion with nonlinear mapping, which helps to obtain a more salient feature representation of the target. The MNFM employs a cascaded and cross-layer connection approach to achieve the fusion of feature maps at different scales while utilizing the Nonlinear Fusion Module for nonlinear mapping to obtain more prominent target features. Specifically, the MNFM first divides the input feature map into two parts: lower-level detail features and higher-level semantic features. Then, it performs a concatenation operation between the lower-level feature map and the higher-level feature map, merging them into a more comprehensive feature representation. This approach retains detailed information from the lower-level while introducing higher-level semantic and contextual information. At the same time, the NFM Bottleneck enhances the saliency of target features through variable convolutions and nonlinear mapping operations.

The specific implementation formula of the Multi-scale Nonlinear Fusion Module is as follows:

$$y_{\text{MNFM}} = \text{Conv}(\text{Concat}\{\text{Split}(x), \text{NFM}(x), \text{NFM}(\text{NFM}(x)), \ldots\}) \tag{7}$$

The NFM achieves the nonlinear mapping of features through multiple branches and skip connections. Specifically, the process involves passing the feature map through different mapping branch units, where each mapping branch unit consists of two cascaded deformable convolutions [22,23]. And then, the results from different mapping branch units are integrated together. The specific implementation formula of the Nonlinear Fusion Module is as follows:

$$y_{\text{NFM}} = x + \text{dconv}(\text{dconv}(x)) + \text{dconv}(\text{dconv}(x)) + \text{dconv}(\text{dconv}(x)) + \ldots \tag{8}$$

Due to the following characteristics of deformable convolutions, it is highly suitable to utilize them as nonlinear mapping units. Deformable convolutions possess the capability of long-range modeling and can dynamically modify the shape of convolution kernels while interacting with local or global features through the adjustment of offset parameters. In addition, it also has the ability of adaptive spatial aggregation. When aggregating different features from convolutions, the offsets and weights are learnable and vary according to the input. For a given input $x \in R^{C \times H \times W}$ and a pixel point p_0, the implementation process of deformable convolution is as follows:

$$y(p_0) = \sum_{k=1}^{K} w_k m_k x(p_0 + p_k + \Delta p_k) \tag{9}$$

where K represents the number of sampling points, w_k is the projection weight of the corresponding sampling points, m_k represents the modulation scalar of the corresponding sampling points, and $\triangle p_k$ represents the offset of the corresponding sampling points.

To a certain extent, the NFM bottleneck can compensate for the shortcomings of the backbone network in extracting features from small objects and objects with multiple scales. Its mapping branch units incorporate multiple sets of mechanisms, which have a more significant global modeling ability and are more suitable for retaining the diversity of features during feature fusion.

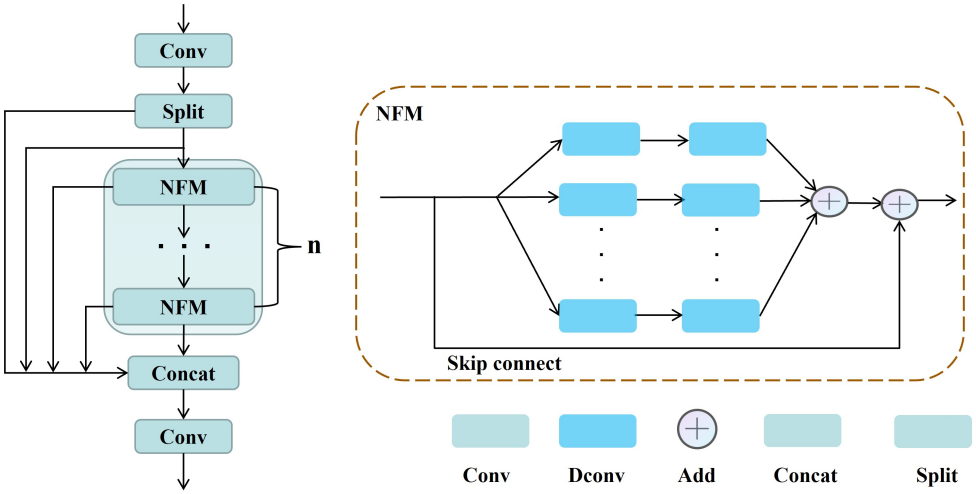


Figure 6. On the left is the MNFM block, and on the right is our NFM Bottleneck.

In a word, the MNFM integrates nonlinear transformations and multi-scale feature fusion into the backbone to enhance the salient features of small targets, thereby improving the detection performance on small targets.

3.5. Loss Function

In this chapter, we introduce a novel loss function named AF_Loss, which plays a crucial role in optimizing the learnable adaptive fusion factor proposed in the AFEM. The implementation details of AF_Loss are presented below:

$$AF\_Loss = \frac{1}{3}\left((\alpha - \bar{\alpha})^2 + (\beta - \bar{\beta})^2 + (\gamma - \bar{\gamma})^2\right) \tag{10}$$

where α, β, and γ represent the learnable adaptive fusion factors of our AFEM. $\overline{\alpha}$, $\overline{\beta}$, and $\overline{\gamma}$ are determined by the number of different-size objects on various layers in each image, which represent the contribution weights of the F_1, F_2, and F_3 feature layers, respectively, for the feature enhancement of small objects. The layer weight calculation formula is as follows:

$$\overline{\alpha} = \frac{e^{n_s}}{e^{n_s} + e^{-n_m} + e^{-n_l}} \tag{11}$$

$$\overline{\beta} = \frac{e^{-n_m}}{e^{n_s} + e^{-n_m} + e^{-n_l}} \tag{12}$$

$$\overline{\gamma} = \frac{e^{-n_l}}{e^{n_s} + e^{-n_m} + e^{-n_l}} \tag{13}$$

In the equations provided above, n_s, n_m, and n_l are determined by searching for objects within their respective receptive fields in the ground truth, based on the varying receptive fields of the detector. These variables represent the numbers of small-scale objects, medium-scale objects, and large-scale objects, respectively.

The specific query process is as follows: our AFE-YOLOv8 structure uses downsampling 5 times, and the final output feature maps are F_1, F_2, and F_3, with sizes of 80 × 80, 40 × 40, and 20 × 20, respectively. The F_1 feature map has a small receptive field and is responsible for detecting targets with a size less than 32 × 32, and the F_2 feature map is responsible for detecting targets with dimensions greater than 32 × 32 but less than 96 × 96, and the F_3 feature map has a large receptive field and is responsible for detecting targets with a size greater than 96 × 96. The query layer is responsible for counting the number of targets at different scales. The targets smaller than 32 × 32 are denoted as n_s, while the targets larger than 32 × 32 and less than 96 × 96 are denoted as n_m; the targets larger than 96 × 96 are denoted as n_l.

As part of our proposed approach, the AF_Loss serves as a critical element within the overall loss function, which is primarily responsible for fine-tuning the learnable adaptive fusion factors in the AFEM. The total loss function comprises three key components: classification loss, regression loss, and AF_Loss.

$$\text{Loss} = \lambda_1 \text{Loss}_{cls} + \lambda_2 \text{Loss}_{\text{reg}} + \lambda_3 \text{AF\_Loss} \tag{14}$$

The parameter λ represents the weight of different loss functions in the total loss, where λ_1 represents the weight of the classification loss, with a value of 0.5; λ_2 represents the weight of the regression loss, with a value of 1.5; and λ_3 represents the weight of the fusion loss proposed in this article, with a value of 0.8 (the λ values in the equation were optimized using the cosine annealing algorithm). The classification loss is calculated using binary cross-entropy loss:

$$\text{Loss}_{cls} = -(y \log(p(x)) + (1 - y) \log(1 - p(x))) \tag{15}$$

And the regression loss is utilized as a Distribution Focal loss:

$$\text{Loss}_{\text{reg}} = \text{DFL}(S_i, S_{i+1}) = -((y_{i+1} - y) \log(S_i) + (y - y_i) \log(S_{i+1})) \tag{16}$$

Intuitively, $S_i = \frac{y_{i+1} - y}{y_{i+1} - y_i}$, $S_{i+1} = \frac{y - y_i}{y_{i+1} - y_i}$, and the DFL allows the network to focus on the values near the target y more quickly, increasing their probability.

4. Experiments

In this chapter, we first introduce the commonly used datasets VisDrone [36] and DOTA [37] for the UAV scenario in Section 4.1. Next, we compare our model with the current mainstream detection model to highlight the advantages of our improved model in Section 4.2. Then, we present the ablation study in Section 4.3, in which the new modules proposed in this paper were gradually added to the baseline model, aiming to clearly indicate the improvement brought by each module. Moreover, we analyze the effects of different loss functions in Section 4.4. Finally, we discuss the operational efficiency of the model in Section 4.5.

4.1. Datasets and Evaluation Metrics

4.1.1. Dataset Analysis

Our experiments against the SOTA methods were mainly conducted on the VisDrone2019 dataset, which includes 10,209 UAV aerial images (6471 for training, 548 for verification, and 3190 for testing). The top ten categories of interest, namely pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle were used for training and testing. The main difficulty for detection is that there exists a large number of small objects, and the object scales of the same class vary dramatically according to the drone's altitude. The distribution of target instance samples in the VisDrone dataset is shown in Figure 7. In addition, to further evaluate the generalization ability and robustness of the proposed model, we conducted the ablation experiments on the VisDrone2019 dataset as well as on the DOTA dataset. The DOTA dataset contains 2806 aerial images from different sensors and platforms with 15 categories and 188,282 objects. The distribution of target instance samples in the DOTA dataset is shown in Figure 8.

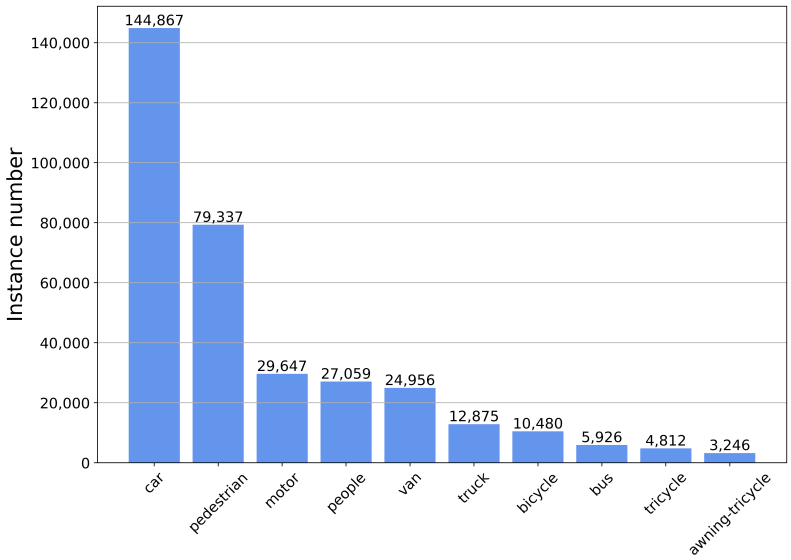


Figure 7. The distribution of instance samples for each category in the VisDrone dataset.

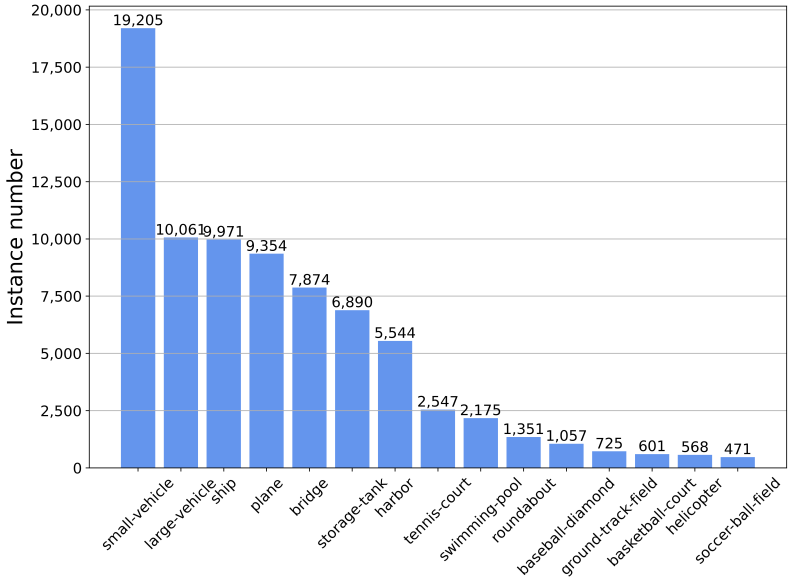


Figure 8. The distribution of instance samples for each category in the DOTA dataset.

4.1.2. Evaluation Metrics

The main evaluation metric used is the mean average precision (mAP). In our experiments, the mAP was calculated by averaging the areas under the precision–recall (PR) curves across 10 IoU threshold values in the range of [0.50: 0.95) for all categories. AP50 represents the average detection accuracy of all categories at an IoU threshold of 0.5. Additionally, to accurately assess the model's detection performance on small and multi-scale objects, we employed the evaluation metrics AP_S, AP_M, and AP_L from the COCO dataset [38] as our additional evaluation metrics. Among them, AP_S represents the average detection accuracy of small targets (less than 32×32), AP_M represents the average detection accuracy of medium targets (greater than 32×32 and less than 96×96), and AP_L represents the average detection accuracy of large targets (greater than 96×96).

4.1.3. Implementation Details

Our models were trained and tested using the NVIDIA GEFORCE RTX2080Ti GPU with 11 GB memory. The data augmentation part uses the default initialization parameters of YOLOv8, and the training period for all experiments was set to 100 epochs. In addition, due to the limited physical memory on the GPU, we set the batch size to 8 for training. The same training and testing configurations were used for all datasets. In addition, validation was conducted after each training phase to select the model with the best performance on the validation set. Finally, the best model was evaluated on the test set.

4.2. Comparison with the SOTA Methods

4.2.1. Quantitative Comparisons

To verify the effectiveness of our proposed model, we compared it with six state-of-the-art UAV object detection approaches on the VisDrone dataset. The methods we compared with are RRNet [39], DPNet-ensemble [36], SMPNet [40], YOLOv8 [25], DPNetV3 [40], YOLOv9 [26], PP-YOLOE [41], YOLOv10 [28], RT-DETR [27], TPH-YOLOv5 [34], and CZ Det [42]. The scores of these methods are presented in Table 1.

Table 1. Quantitative comparisons with the SOTA methods on the VisDrone dataset.

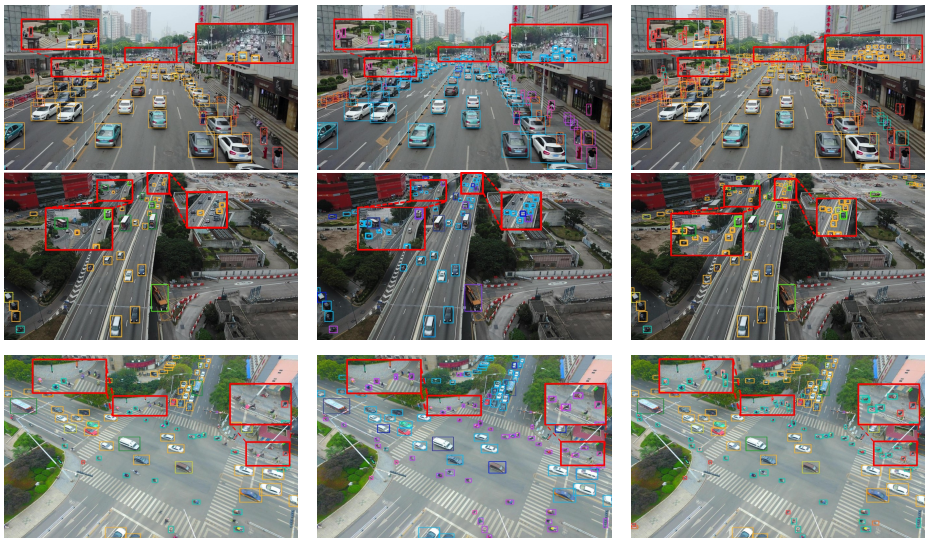
Method	mAP	AP50
RRNet	29.13	55.82
DPNet-ensemble	31.22	57.36
SMPNet	34.63	\
YOLOv8	35.24	60.92
DPNetV3	36.37	61.15
YOLOv9	37.08	62.36
PP-YOLOE	37.14	62.83
YOLOv10	38.42	63.69
RT-DETR	38.84	64.76
TPH-YOLOv5	39.18	\
CZ Det	39.36	65.83
AFE-YOLOv8 (ours)	**39.75**	**66.37**

Table 1 presents the test results of the current mainstream detection model on the VisDrone2019 dataset and compares the detection results of our AFE-YOLOv8 model with those of other models.

4.2.2. Qualitative Comparisons

We conducted comparison experiments on the VisDrone and DOTA datasets to demonstrate the effectiveness of AFE-YOLOv8, and we visualize the detection results in Figures 9 and 10. From the visualization of the detection results, it is evident that our AFE-YOLOv8 model outperforms other models in detecting small objects and multi-scale objects. For instance, in Figures 9 and 10, the objects enclosed within the red rectangles

represent false negatives obtained by other models, while our model successfully detected these missed objects.



(**a**) YOLOv8 (**b**) TPH-YOLOv5 (**c**) AFE-YOLOv8 (Ours)

Figure 9. Visualization of the detection results of the three models on the VisDrone dataset, where the red rectangular boxes show enlarged displays of the target dense areas.

(**a**) YOLOv8 (**b**) TPH-YOLOv5 (**c**) AFE-YOLOv8 (Ours)

Figure 10. Visualization of the detection results of the three models on the DOTA dataset, where the red rectangular boxes show enlarged displays of the small-target dense areas.

In Figure 9, we visualize the detection results of YOLOv8, TPH-YOLOv5, and AFE-YOLOv8 (our method) on the VisDrone dataset. In sharp contrast to YOLOv8 and TPH-YOLOv5 object detection methods, our method exhibits a superior detection performance for dense small objects and multi-scale objects.

Compared to the VisDrone dataset, the DOTA dataset exhibits significant variation in object scales and higher scene complexity. To evaluate the robustness and generalization of our method, we conducted an additional qualitative comparison experiment on the DOTA dataset and visualized the detection results, as shown in Figure 10. The comparison demonstrates that thanks to the inclusion of the RFEM in our method, our approach effectively detects multi-scale objects and maintains reliable robustness in complex scenes.

4.3. Ablation Studies

To further showcase the effectiveness of the proposed methods, we conducted several ablation experiments on both the VisDrone and DOTA datasets. These experiments aimed to evaluate the performances of the individual modules of our methods.

4.3.1. Ablation Analysis of All Components

We selected YOLOv8 as the baseline model and conducted experiments for the following cases: (1) baseline; (2) baseline with the MNFM; (3) baseline with the MNFM and AFEM; (4) baseline with the MNFM, AFEM, and RFEM. We performed the ablation experiments on the VisDrone and DOTA datasets, and the results are presented in Tables 2 and 3.

Table 2. Ablation comparison of model performance improvement on the VisDrone dataset.

Model	mAP	AP50	AP_S	AP_M	AP_L
YOLOv8 (Baseline)	35.24	60.92	28.33	44.52	63.12
YOLOv8 + MNFM	36.58	62.83	31.02	46.83	65.55
YOLOv8 + MNFM + AFEM	38.43	64.16	32.85	46.81	65.46
YOLOv8 + MNFM + AFEM + RFEM (ours)	**39.75**	**66.37**	**33.14**	**51.27**	**69.35**

Based on the detection performance on the VisDrone dataset, our model showed an improvement of 4.51% in mAP compared to the baseline. The detection performance metric for small objects AP_S improved by 4.81%. Additionally, the detection performance metrics for medium-scale objects AP_M and large-scale objects AP_L showed improvements of 6.75% and 6.23%, respectively.

Table 3. Ablation comparison of model performance improvement on the DOTA dataset.

Model	mAP	AP50	AP_S	AP_M	AP_L
YOLOv8 (Baseline)	63.85	85.47	50.12	77.23	91.31
YOLOv8 + MNFM	64.68	86.29	53.45	83.25	93.22
YOLOv8 + MNFM + AFEM	66.15	87.83	56.18	83.28	93.15
YOLOv8 + MNFM + AFEM + RFEM (ours)	**67.29**	**88.38**	**56.32**	**84.41**	**96.15**

From the evaluation results on the DOTA dataset, our model showcased a remarkable enhancement in mAP of 3.44% compared to the baseline. Notably, the detection performance of small objects with high difficulty AP_S witnessed a substantial improvement of 6.2%. Additionally, the detection performance metrics AP_M and AP_L, which measure the performance for medium- and large-scale objects, respectively, both show commendable enhancements of 7.18% and 4.84%.

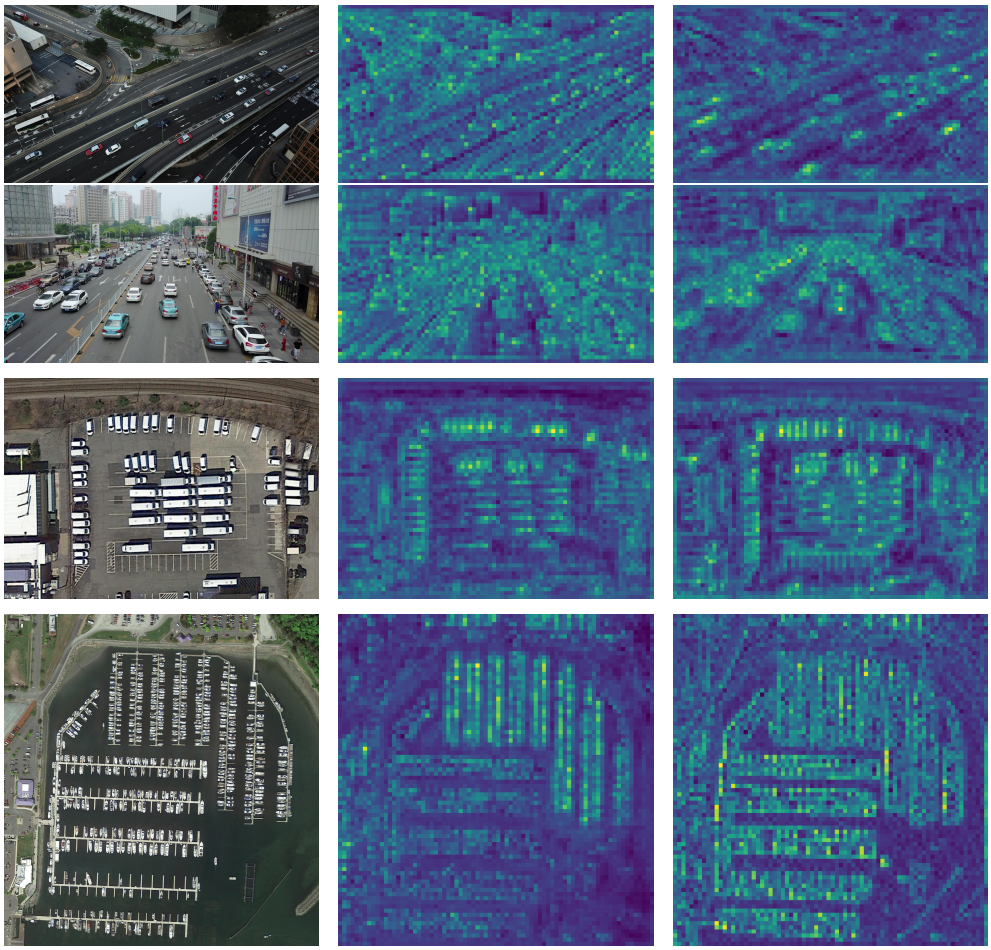
4.3.2. Ablation study on MNFM

To investigate the improvement of the MNFM on the detection accuracy, we conducted ablation experiments on the VisDrone and DOTA datasets. The experimental results, as shown in Table 4, provide insights into the improvement achieved by our method.

Table 4. Ablation comparison with and without the MNFM on the VisDrone and DOTA datasets.

Dataset	MNFM	mAP	AP50	AP_S	AP_M	AP_L
VisDrone	**w/o**	35.24	60.92	28.33	44.52	63.12
	w/	**36.58**	**62.83**	**31.02**	**46.83**	**65.55**
DOTA	**w/o**	63.85	85.47	50.12	77.23	91.31
	w/	**64.68**	**86.29**	**53.45**	**83.25**	**93.22**

The aforementioned quantitative ablation experiments demonstrated that the MNFM improves the mAP of the baseline model. In adaptively adjusting the scale of the convolution kernels, the MNFM is able to extract more useful features from small, medium, and large objects, thereby enhancing the overall performance of the model. To visually observe and compare the effectiveness of feature extraction with and without the MNFM, we generated visualizations of the feature maps. The qualitative comparison result is shown in Figure 11.



(**a**) Input (**b**) With/o MNFM (**c**) With MNFM

Figure 11. (**a**) represents the input image (from the VisDrone and DOTA datasets), (**b**) represents the feature map without the MNFM, and (**c**) represents the feature map with the MNFM.

Obviously, based on the visualization results, it is evident that the backbone network is capable of extracting more valid and salient features of the objects after integrating the MNFM. Therefore, it can be concluded that the MNFM plays a critical role in the feature extraction stage.

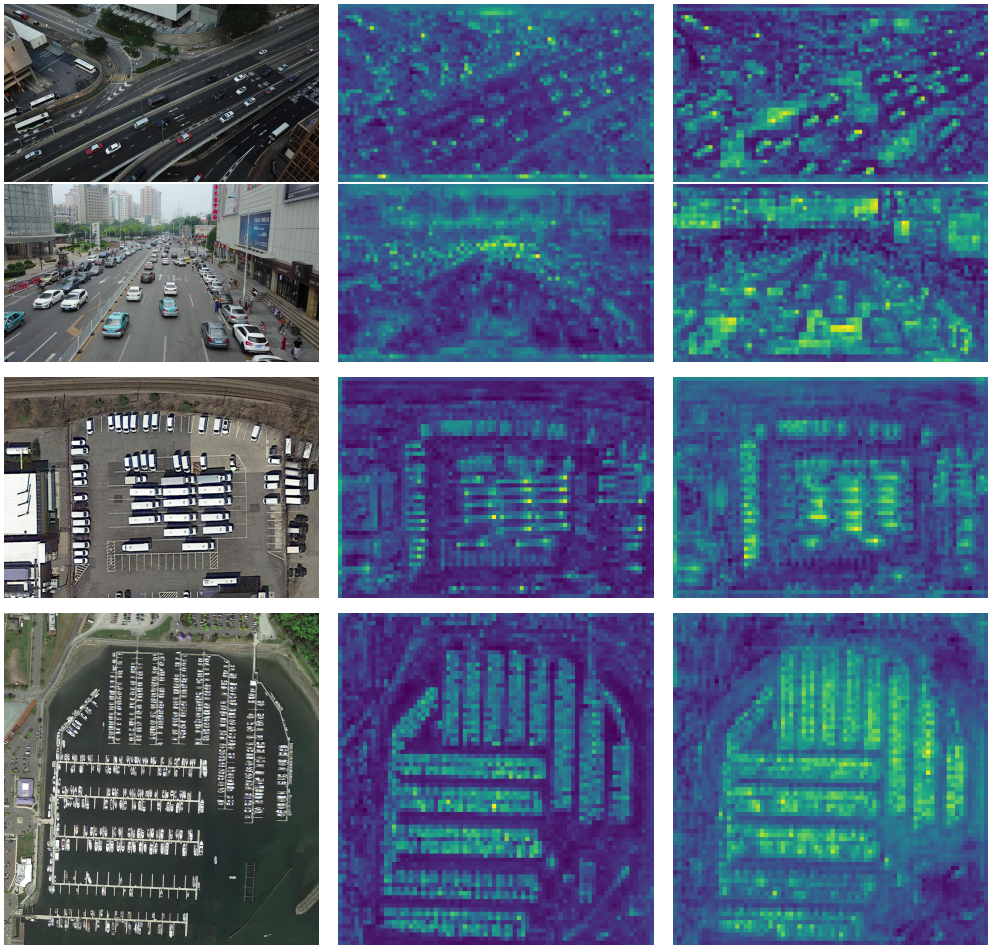
4.3.3. Ablation study on AFEM

We conducted ablation experiments on both the VisDrone and DOTA datasets to evaluate the impact of the AFEM on enhancing the features of small objects. The experimental results are presented in Table 5.

Table 5. Ablation comparison with and without the AFEM on the VisDrone and DOTA datasets.

Dataset	AFEM	mAP	AP50	AP_S	AP_M	AP_L
VisDrone	**w/o**	35.24	60.92	28.33	44.52	63.12
	w/	**36.86**	**63.22**	**31.57**	**44.55**	**63.10**
DOTA	**w/o**	63.85	85.47	50.12	77.23	91.31
	w/	**64.76**	**86.31**	**52.66**	**77.22**	**91.33**

As demonstrated in the quantitative ablation experiments presented in Table 5, the AFEM exhibits improvements in the mAP of small-object detection. By adaptively fusing features from different object pipelines, the AFEM effectively preserves more features of small-object pipelines, resulting in a more significant improvement in the detection performance of small objects. To visually observe the impact of the AFEM in feature extraction, we compare the feature maps with and without AFEM in Figure 12. The visualization results clearly show that the addition of the AFEM enriches the feature details within the small-object pipeline. This confirms that the module effectively enhances the feature maps for small objects.



(**a**) Input image (**b**) With/o AFEM (**c**) With AFEM

Figure 12. (**a**) represents the input image (from the VisDrone and DOTA datasets), (**b**) represents the feature map without the AFEM, and (**c**) represents the feature map with the AFEM.

4.3.4. Ablation study on RFEM

To demonstrate that the RFEM can improve the detection performance of medium-scale and large-scale objects by enlarging the receptive field, we conducted ablation experiments on both the VisDrone and DOTA datasets, and the experimental results are shown in Table 6.

Table 6. Ablation comparison with and without the RFEM on the VisDrone and DOTA datasets.

Dataset	RFEM	mAP	AP50	AP_S	AP_M	AP_L
VisDrone	**w/o**	35.24	60.92	28.33	44.52	63.12
	w/	**37.43**	**64.02**	**28.38**	**48.11**	**66.20**
DOTA	**w/o**	63.85	85.47	50.12	77.23	91.31
	w/	**65.06**	**87.15**	**50.24**	**83.72**	**94.82**

As evident from the experimental results, the RFEM improves the overall detection accuracy of our model. This is achieved by leveraging global information modeling to expand the receptive field, thereby boosting the model's performance.

4.4. Loss Function

We visualize the curve of the loss function during the training process in Figure 13, which shows that our loss curve decreases slowly as it approaches 100 epochs, and the training process was stopped in time to avoid overfitting the model.

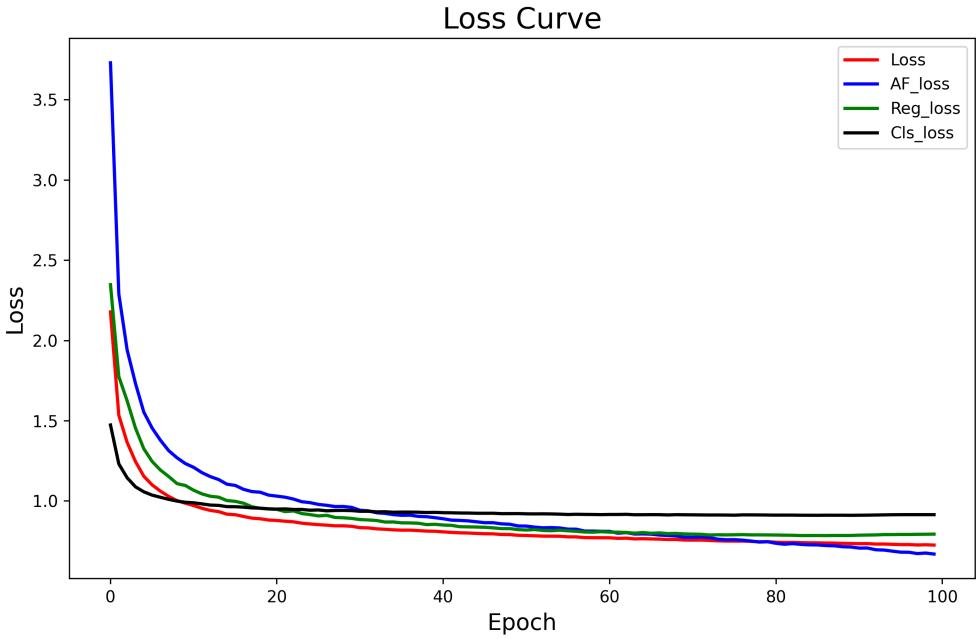


Figure 13. Loss curves for different loss functions during training process.

In Figure 13, the total loss is represented by a red curve, the AF_Loss by a blue curve, the Reg_Loss by a green curve, and the categorical loss Cls_Loss by a black curve. The total loss is obtained by multiplying the individual loss components by the corresponding loss weights λ. In observing the trend of the total loss function curves, it can be seen that the total loss of the model continues to decrease under the effect of different loss functions, and the model gradually converges.

4.5. Running Efficiency

In order to explore the performance advantages of the framework proposed in this article, we added relevant experiments mainly to explore the performance comparison of our model compared to the baseline model in terms of average detection accuracy (mAP),

parameter quantity (Params), computational complexity (FLOPs), and inference speed (Speed). The experimental results are shown in Table 7.

Table 7. Quantitative comparisons with baseline models on the NVIDIA 2080Ti platform.

Model	mAP	Paras (M)	FLOPs (B)	Speed (ms)
YOLOv8-s (Baseline)	35.24	11.2	28.6	3.9
AFE-YOLOv8-s	39.75	13.5	32.4	8.2

Through the analysis of the experimental results, it can be seen that the system in this article has significantly improved the average detection accuracy compared to the baseline system, with an increase of 4.51% compared to the baseline model. However, the parameter and computational complexity of the model have slightly increased, and the inference speed has decreased compared to the baseline system. However, it can still meet the requirements of real-time detection. In future research, we will improve the model toward a more lightweight direction, further reducing the number of parameters and computation while minimizing the detection accuracy of the model, in order to accelerate the inference speed of the model.

5. Conclusions

This paper presented a novel object detection method, AFE-YOLOv8, for drone aerial scenes. The proposed method builds upon the widely used YOLOv8 structure, which suffers from a significant limitation in extracting features of small objects, thereby limiting its performance in detecting massive small objects. To overcome this limitation, we introduced the Adaptive Feature Enhancement Module (AFEM), a small-object feature enhancement module that aggregates the features of small objects contained in different object pipelines. This is achieved by adaptively assigning learnable fusion factors for different object pipeline features. Additionally, we incorporated a Multi-scale Nonlinear Fusion Module (MNFM) in the backbone network to capture more small-object-related feature representations. To further boost the performance, we also proposed the Receptive Field Expansion Module (RFEM) to increase the receptive field for large- and medium-scale objects. Based on the above innovative structural designs, our AFE-YOLOv8 achieved a significantly improved detection accuracy in drone aerial scene object detection. Finally, We conducted comprehensive quantitative evaluations and qualitative analyses to validate the effectiveness of our proposed method.

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

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Kim, B.; Lee, J.; Kang, J.; Kim, E.S.; Kim, H.J. Hotr: End-to-end human-object interaction detection with transformers. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
2. Vielzeuf, V.; Lechervy, A.; Pateux, S.; Jurie, F. Centralnet: A multilayer approach for multimodal fusion. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
3. Zhu, C.; He, Y.; Savvides, M. Feature selective anchor-free module for single-shot object detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
4. Wang, X.; Zhang, S.; Yu, Z.; Feng, L.; Zhang, W. Scale-equalizing pyramid convolution for object detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.

5. Zang, S.; Chen, M.; Ai, Z.; Chi, J.; Yang, G.; Chen, C.; Yu, T. Texture-aware gray-scale image colorization using a bistream generative adversarial network with multi scale attention structure. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106094. [CrossRef]
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
7. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
8. Liu, L.; Ouyang,W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikinen, M. Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [CrossRef]
9. Dai, J.; Li, Y.; He, K.; Sun, J. *R-fcn: Object Detection via Region-Based Fully Convolutional Networks*; Curran Associates Inc.: New York, NY, USA, 2016.
10. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
11. 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]
12. 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 (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
13. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
14. Redmon, J.; Farhadi, A. Yolo9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
15. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
16. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
17. Wang, C.Y.; Bochkovskiy, A.; Liao, H. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
18. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *42*, 318–327. [CrossRef] [PubMed]
19. Lin, T.Y.; Dollar, 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; IEEE Computer Society: Piscataway, NJ, USA, 2017.
20. Yang, C.; Huang, Z.; Wang, N. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022.
21. Zhuang, J.; Qin, Z.; Yu, H.; Chen, X. Task-specific context decoupling for object detection. *arXiv* **2023**, arXiv:2303.01047.
22. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
23. Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. *arXiv* **2022**, arXiv:2211.05778.
24. Berg, A.C.; Fu, C.Y.; Szegedy, C.; Anguelov, D.; Erhan, D.; Reed, S.; Liu, W. Ssd: Single shot multibox detector. In *Computer Vision—ECCV 2016*; Springer: Cham, Switzerland, 2016.
25. Varghese, R.; Sambath, M. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 18–19 April 2024; pp. 1–6.
26. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv* **2024**, arXiv:2402.13616.
27. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974.
28. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. *arXiv* **2024**, arXiv:2405.14458.
29. Akyon, F.C.; Onur Altinuc, S.; Temizel, A. Slicing aided hyper inference and fine-tuning for small object detection. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022.
30. Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; Li, S.Z. Faceboxes: A cpu real-time face detector with high accuracy. In Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, USA, 1–4 October 2017.
31. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A.A.A.; Yogamani, S.; Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 4909–4926. [CrossRef]
32. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-shot refinement neural network for object detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
33. Li, Y.; Chen, Y.; Wang, N.; Zhang, Z. Scale-aware trident networks for object detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.

34. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021.
35. Wang, P.; Xue, D.; Zhu, Y.; Sun, J.; Yan, Q.; Yoon, S.E.; Zhang, Y. Take a prior from other tasks for severe blur removal. *arXiv* **2023**, arXiv:2302.06898.
36. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Liu, Z.M. Visdrone-det2019: The vision meets drone object detection in image challenge results. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019.
37. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. Dota: A large-scale dataset for object detection in aerial images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
38. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014*; Springer: Cham, Switzerland, 2014.
39. Chen, C.; Zhang, Y.; Lv, Q.; Wei, S.; Wang, X.; Sun, X.; Dong, J. Rrnet: A hybrid detector for object detection in drone-captured images. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 100–108.
40. Cao, Y.; He, Z.; Wang, L.; Wang, W.; Yuan, Y.; Zhang, D.; Zhang, J.; Zhu, P.; Van Gool, L.; Han, J.; et al. Visdrone-det2021: The vision meets drone object detection challenge results. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Virtual, 11–17 October 2021.
41. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y.; et al. Pp-yoloe: An evolved version of yolo. *arXiv* **2022**, arXiv:2203.16250.
42. Meethal, A.; Granger, E.; Pedersoli, M. Cascaded zoom-in detector for high resolution aerial images. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 17–24 June 2023.