drones

MDPI

Article

AEMFOD: Asymptotic Elimination Multi-Scale Fusion for Real-Time UAV Traffic Detection

Bo Shao [1], Yingxun Wang [2], Jiang Zhao [1] and Zhihao Cai [2,*]

[1] The School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China; xbshao@buaa.edu.cn (B.S.)
[2] The Unmanned System Research Institute, Beihang University, Beijing 100191, China
* Correspondence: czh@buaa.edu.cn

Highlights

What are the main findings?

- AEMFOD improves multi-scale and small-object detection in UAV imagery while retaining a lightweight architecture.
- AEFPN and AWCF jointly enhance feature fusion efficiency, and real-flight experiments verify the model's real-time deployment capability.

What are the implications of the main findings?

- Redundant cross-scale feature interactions can be reduced without sacrificing effective UAV object representation.
- AEMFOD provides a practical solution for real-time visual perception on resource-constrained UAV platforms.

Abstract

Object detection is a fundamental perception capability for UAV-based intelligent transportation systems, underpinning applications including traffic monitoring, incident detection, and collision warning. However, reliable detection from aerial imagery remains difficult because UAV scenes typically contain multi-scale objects, diverse viewpoints, imbalanced object distributions, and stringent onboard computational constraints. Moreover, most existing studies primarily emphasize offline accuracy improvements, often overlooking the requirements of real-time, resource-constrained deployment. To simultaneously consider detection performance and onboard efficiency, we develop an Asymptotic Elimination Multi-scale Fusion Object Detection framework, termed AEMFOD, for UAV traffic imagery. Specifically, an asymptotic elimination feature pyramid network (AEFPN) is introduced to reduce computational complexity, while the asymptotic weighted coordinate fusion (AWCF) module operates within AEFPN to strengthen the integration of multi-scale features and preserve informative low-level representations. The proposed detector is evaluated through both benchmark experiments and practical UAV flight experiments. Compared with the YOLOv5s baseline, AEMFOD achieves relative AP_{50} improvements of 4.8% on DOTA and 31.5% on VisDrone. Furthermore, embedded experiments quantify inference speed, and flight experiments yield qualitative detection samples under practical UAV operating conditions, showing competitive latency and a compact model footprint suitable for onboard traffic applications.

Keywords: object detection; unmanned aerial vehicles (UAV); multi-scale feature fusion; attention modules

Check for updates

Academic Editor: Mumtaz Karatas

Received: 9 September 2026
Revised: 27 October 2026
Accepted: 28 October 2026
Published: 30 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

UAVs have become an important mobile sensing platform for intelligent transportation systems because of their flexible deployment, broad field of view, and relatively low operating cost. UAV-based vision supports traffic monitoring, congestion and incident detection, vehicle counting, post-accident assessment, and emergency routing tasks [1–3] that directly affect traffic safety, mobility, and environmental efficiency. Accurate detection of vehicles, pedestrians, and cyclists in aerial imagery is therefore a prerequisite for downstream tasks such as traffic-state estimation, adaptive traffic management, and collision warning.

Although deep-learning detectors have achieved strong performance on conventional ground-level imagery, their direct application to UAV scenes remains problematic. This is because the characteristics of UAV platforms lead to large differences between aerial and ground-level images. Several characteristics of UAV imagery make reliable object detection difficult. (1) Diverse viewpoints. The different viewpoints of the UAV will result in the image containing various shapes, sizes, and angles of objects, which makes feature information extraction difficult. (2) Imbalanced object distribution. The background is complex and variable, and the distribution of objects, which may be sparse or dense, cannot be determined. (3) Extreme scale variation. Small and tiny objects account for a large proportion of UAV imagery, making their discriminative information vulnerable to repeated downsampling. (4) External perturbation. The images are subject to motion blur, occlusion, and illumination problems due to environmental changes and drone flights. (5) Real-time and resource constraints. There is a conflict between the high real-time processing requirements and the limited computational capacity of UAV onboard devices. These challenges jointly increase the difficulty of reliable UAV-based traffic detection.

Many researchers have proposed methods to overcome the above problems. Some focus on detecting objects in specific scenes, such as ship or vehicle detection [4], while others utilize data augmentation techniques or improve feature information [5]. Besides, some researchers introduce transformers to capture global contextual information in addition to local features [6]. Nevertheless, most of these studies focus solely on the algorithmic aspect of real-time object detection, overlooking the practical deployment of these algorithms on UAVs. Most detection algorithms run offline and locally, concentrating more on solving scale and image quality problems to enhance the detection performance without considering the limited computing capacity of UAV platforms. Few studies provide in-flight or edge-device deployment metrics (latency, FPS, power) under realistic UAV operating conditions. Therefore, this article proposes an asymptotic elimination multi-scale fusion object detection (AEMFOD) to reduce redundant computation while retaining multi-scale discriminability for practical UAV traffic monitoring. It can operate in real time with high detection accuracy on computationally constrained embedded platforms. In AEMFOD, we design an asymptotic elimination feature pyramid network (AEFPN) tailored for UAV scenarios. AEFPN enhances low-level feature representations by mitigating the influence of deep semantic information on shallow positional details, eliminating redundant features, and preserving essential information. This approach reduces the number of parameters and improves feature fusion, thereby balancing detection accuracy with computational efficiency. Complementing AEFPN, the asymptotic weighted coordinate fusion (AWCF) module is introduced to integrate neighboring features efficiently, narrow the semantic gap, and further mitigate conflicts among adjacent features. AWCF enhances multi-scale feature representation by facilitating smoother feature fusion. It effectively compensates for information loss induced by feature elimination in AEFPN, thus improving overall fusion quality. AEFPN and AWCF synergistically establish a robust, scale-aware feature hierarchy that underpins UAV target detection. This design improves information exchange between

neighboring scales and enhances the spatial information required for small object detection while maintaining a lightweight model structure.

The main contributions of this work are summarized as follows:

1. An efficient detector for onboard UAV traffic perception. We develop AEMFOD to balance detection accuracy and computational efficiency. The framework reorganizes the detection architecture to eliminate redundant computation while preserving multi-scale feature discriminability.
2. A complementary feature fusion strategy. We introduce a novel fusion hierarchy that consists of AEFPN and AWCF. AEFPN dynamically balances contributions across scales to alleviate gradient conflicts and enhance small object representation; AWCF employs cross-scale attention to harmonize feature hierarchies and mitigate intra-scale bias. Together, these modules form a compact and effective multi-scale fusion mechanism.
3. Extensive empirical validation and in-flight deployment study. AEMFOD shows consistent improvements over strong baselines on benchmark UAV datasets. Furthermore, in-flight experiments provide promising qualitative detection results under actual UAV operating conditions.

The remainder of this article is organized as follows. Section 2 discusses the related UAV detection methods. Section 3 presents the proposed AEMFOD architecture. Section 4 demonstrates the effectiveness of the proposed methodology through the dataset and flight experiments. Section 5 summarizes the main findings.

2. Related Work

2.1. Deep Learning-Based Object Detection

Deep learning has substantially reshaped object detection, with modern detectors generally following either a two-stage or a one-stage prediction paradigm. Two-stage approaches first identify candidate regions, then refine their categories and bounding boxes. R-CNN [7] established this framework, while Fast R-CNN [8] and Faster R-CNN [9] progressively improved feature sharing and proposal generation to increase efficiency. Later variants extended this formulation in different directions. Mask R-CNN [10] proposes the RoIAlign method to obtain fractional pixel coordinates of the object. Cascade R-CNN [11] uses multiple cascaded detection heads to correct the prediction region several times, thus gradually improving the detection accuracy.

One-stage detectors remove the explicit proposal-generation stage and directly perform classification and localization over dense feature maps. This design generally provides a more favorable inference speed and is therefore attractive for real-time applications. YOLO [12], the first end-to-end one-stage detection network, formulated detection as an end-to-end regression problem, dividing the image into a grid and regressing the prediction directly on each grid to obtain the class and box. SSD [13] is a fully convolutional network that improves detection accuracy by presetting the prior bounding box scales and achieves multi-scale detection by extracting feature map information from different layers. RetinaNet [14] improves the accuracy of small-scale objects through the feature pyramid. Besides, it further addressed the foreground–background imbalance in dense detection through focal loss. For deployment-oriented applications, the YOLO series stands out among one-stage detectors due to its strong balance between model complexity, detection accuracy, and inference speed.

2.2. Object Detection in UAV Imagery

UAV imagery differs considerably from conventional ground-level images because targets are often small, densely distributed, and subject to substantial scale and viewpoint

variations. Consequently, a variety of specialized detection strategies have been developed for aerial scenes. Chen et al. [15] proposed a hybrid detector RRNet by combining an anchor-free coarse detector and re-regression module, which detects the object centroid and then generates an accurate bounding box through the regression module. An adaptive resampling augmentation strategy named AdaResampling is also proposed to enhance the data for the target. Yu et al. [16] proposed the DSHNet to address the long-tailed category distribution commonly encountered in UAV datasets, which addresses the imbalance in category distribution but requires high processing power. It utilizes class-biased sampling and separate processing paths for head and tail categories, and the loss function is updated by weighting the losses of the tail and head classes. From a domain-robustness perspective, AdNet [17] introduces multi-scale adversarial learning to reduce feature discrepancies caused by changes in viewpoint, illumination, weather, and background. Bosquet et al. [18] proposed a down-sampling generative adversarial network. The DS-GAN incorporates a GAN-based object generator in conjunction to generate small object data and increase small object identification accuracy through object segmentation, inpainting, and image blending approaches. However, the GAN introduces substantial computation and additional parameters, reducing inference speed.

In addition, the YOLO series has also received much attention due to its significant real-time performance, and many related works have been proposed based on YOLO. SlimYOLOv3 [19] reduces the computational burden of YOLOv3 through network pruning, resulting in a more compact architecture and faster inference. TPH-YOLOv5 [20] incorporates transformer-based prediction heads to strengthen feature representation for drone imagery, but the added attention computation limits real-time deployment capability. VIT-YOLO [21] similarly introduces global contextual modeling through multi-head self-attention. More recently, Drone-YOLO [22] modifies YOLOv8 using RepVGG-based feature extraction and a sandwich-fusion strategy to improve small-object representation. These methods demonstrate that enhancing multi-scale or global feature interaction can improve UAV detection, but the associated computational increase remains a critical concern for onboard applications.

Overall, existing methods struggle to balance detection accuracy, inference latency, and model footprint for UAV-based traffic monitoring and ITS applications. Consequently, improving accuracy alone is insufficient; reducing FLOPs and parameters while achieving a reasonable FPS–AP trade-off is a primary problem for practical UAV deployment.

3. Proposed Method

This section presents the architecture of AEMFOD and details its main components. The framework integrates a novel scale-aware feature fusion hierarchy designed for drone applications, consisting of the asymptotic elimination feature pyramid network (AEFPN) and the asymptotic weighted coordinate fusion (AWCF) module. Together, these components optimize the multi-scale feature hierarchy, mitigate inter-scale conflicts, and enable multi-head detectors to specialize more effectively in small-object detection.

3.1. Overview

The overall architecture of AEMFOD is shown in Figure 1, following the conventional backbone–neck–head organization. We mainly improve the neck and head of the network with the backbone remaining unchanged from YOLOv5 to achieve high performance with a lightweight model. The proposed neck is constructed using AEFPN, which reorganizes multi-scale feature propagation to reduce redundant computation while preserving spatial information important for small objects. It can reduce the parameters and FLOPs while ensuring detection accuracy, which achieves the balance between accuracy and speed

compared with other feature fusion structures. Within AEFPN, the adjacent level fusion (ALF) module enables high-level semantics to complement low-level spatial details by progressively exchanging information between neighboring feature layers, while avoiding the introduction of excessive long-range interactions. The AWCF module is incorporated during hierarchical fusion to adaptively emphasize informative features across scales and reduce inconsistencies among feature levels. In addition, a high-resolution prediction branch is added to increase sensitivity to pixel-scarce tiny objects in aerial images.

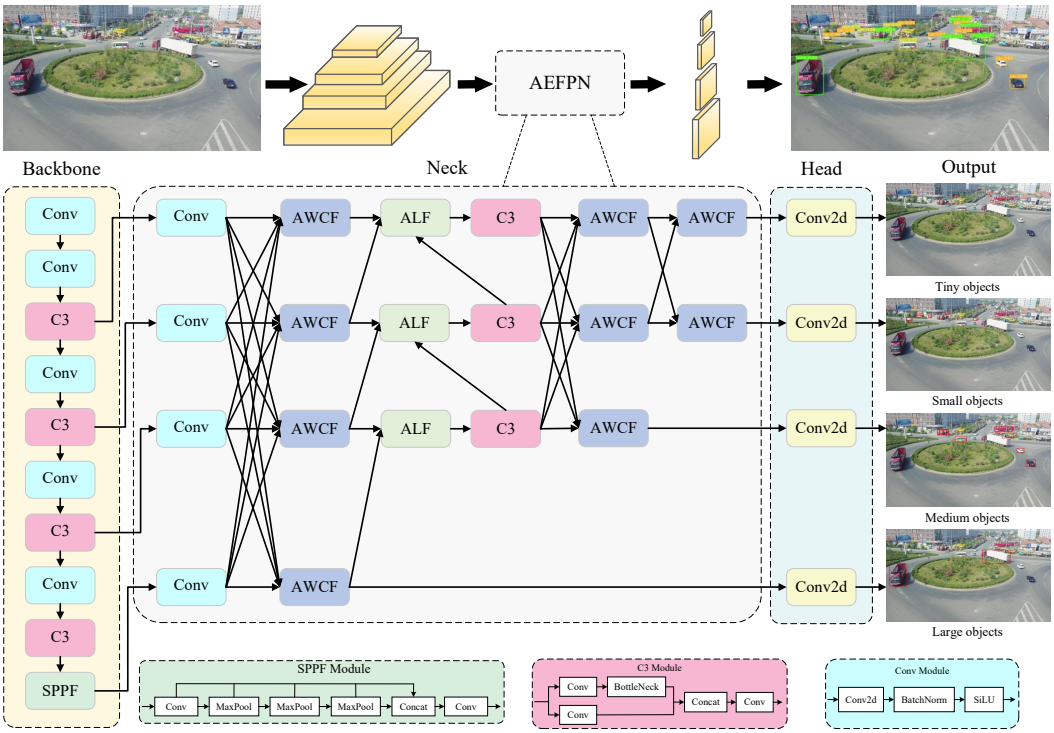


Figure 1. Architecture of AEMFOD.

3.2. Additional Prediction Head

The YOLO framework predicts objects of different sizes from three feature levels with spatial resolutions of 80 × 80, 40 × 40, and 20 × 20, which mainly correspond to small, medium, and large objects, respectively. By using multiple feature maps and prediction heads, YOLO can effectively improve the detection accuracy across different scales. However, due to changes in perspective during UAV flight, objects in the images vary widely in scale, and UAV imagery contains more tiny objects than natural scenes. This design is insufficient to handle all possible detection scenarios during UAV flight. To enhance the versatility of the UAV image detection algorithm across various scenarios, we incorporate an extra 160 × 160 prediction head for tiny objects alongside the original three detection heads. The tiny prediction head maintains low-level high-resolution features, preventing information loss during feature fusion and enhancing sensitivity to small targets. This addition increases the model's parameters and complexity but greatly boosts the performance of tiny object detection without compromising accuracy for regular-scale objects.

3.3. Asymptotic Elimination Feature Pyramid Network

Variations in UAV flight altitude lead to objects of different scales in the field of view, while the limited computing capacity of UAV platforms imposes strict constraints on model size and complexity. Traditional pyramid architectures usually propagate information repeatedly across multiple levels through both top-down and bottom-up pathways. However, this strategy tends to fuse all hierarchical features indiscriminately, which may introduce

feature redundancy and increase parameters and FLOPs. Moreover, this indiscriminate fusion can dilute essential low-level spatial details critical for small objects, posing challenges for UAV object detection.

Based on this observation, AEFPN, a lightweight asymptotic elimination feature pyramid network, is designed specifically for UAV requirements. This network reduces parameters and model complexity while maintaining detection accuracy. Inspired by AFPN's progressive fusion strategy [23], AEFPN reorganizes the fusion pathways to limit the potential dilution of shallow spatial details caused by repeated propagation of deep semantic features. Unlike AFPN, which gradually expands the fusion scope by integrating higher-level features, AEFPN transitions from extensive cross-scale interactions to increasingly restricted fusion paths. The elimination concerns subsequent fusion branches rather than the complete removal of semantic information already incorporated into the fused representations. The key innovation of AEFPN is to establish sufficient multi-scale communication in the early fusion stage and then progressively remove less useful non-adjacent high-level dependencies. In this way, low-level features can retain the spatial information required for small-object localization, while high-level semantic information is introduced in a controlled manner rather than repeatedly propagated across all levels. This structured elimination strategy reduces redundant feature interaction and lowers model complexity without completely sacrificing cross-scale information.

As illustrated in Figure 2c, AEFPN operates in three sequential stages. The feature layers at each scale are first fused to ensure the integrated utilization of multi-scale features. This initial fusion preserves comprehensive information from multiple scales. The network then prioritizes the progressive fusion of adjacent hierarchical features, which minimizes semantic gaps. This step facilitates smoother integration of cross-scale information and reduces semantic discrepancy. Finally, during the top-down propagation process, the influence of non-adjacent high-level features is gradually reduced via a structured culling mechanism. This selective elimination retains critical semantic information while discarding redundant features. This gradual transition from broad fusion to localized fusion and finally to selective elimination forms the asymptotic feature-propagation pattern of AEFPN.

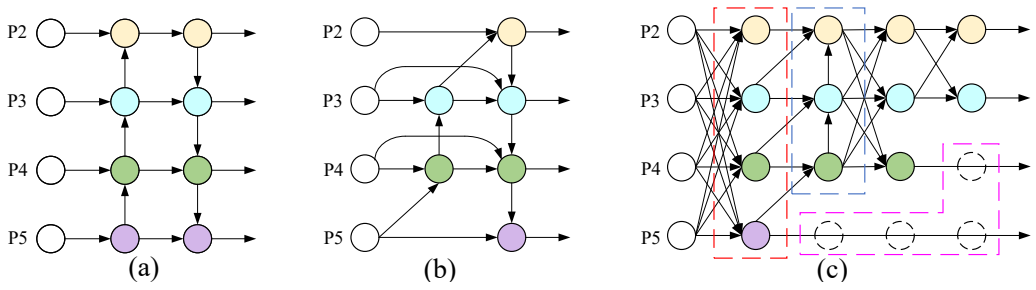


Figure 2. Feature Pyramid Network. (**a**) Path Aggregation Network (PANet). (**b**) Bi-directional Feature Pyramid Network (BiFPN). (**c**) AEFPN. P2–P5 are multi-scale features from levels 2 to 5. The red, blue, and pink rectangles denote the fully connected feature fusion, adjacent level fusion, and asymptotic feature elimination in AEFPN.

Fully connected feature fusion. Feature maps contain sufficient image information after backbone feature extraction. To aggregate information from multiple scales before the network begins to restrict cross-level propagation, AEFPN conducts multi-scale feature fusion at different hierarchical feature levels, enabling each layer to contain multiple scales of feature information.

Adjacent level fusion. In the early stages of feature fusion, the difference between multi-scale features will affect the subsequent fusion. Because neighboring feature levels are more closely related than widely separated ones, the adjacent level fusion (ALF) module emphasizes information exchange between adjacent levels and suppresses less

relevant long-range interactions. The structure of ALF is shown in Figure 3. We apply a 1×1 convolution operation on the low-level features and an upsampling operation on the adjacent higher-level features before a concatenation operation. The subsequent convolution operation extracts the combined information, which contains rich details about the classification and spatial localization of the objects. The ALF module output is expressed as follows:

$$y = f^{1\times1}\left(\left[f^{1\times1}(x_1), Upsample(x_2), Upsample(x_3)\right]\right). \quad (1)$$

where x_1, x_2 and x_3 are the input feature vectors at different scales, $[.,.]$ is a concatenation operation, $f^{1\times1}$ is a convolutional layer with a kernel size of 1×1 followed by the Sigmoid Linear Unit (SiLU) activation function.

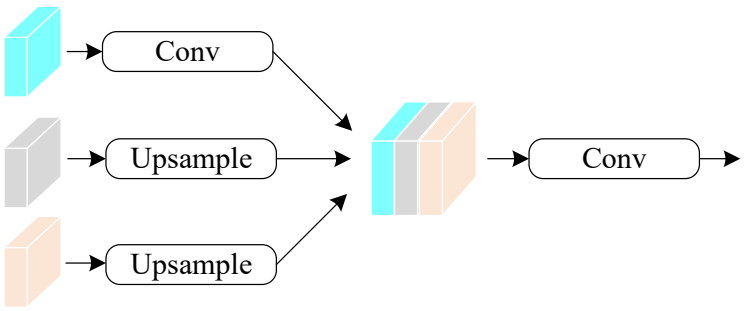


Figure 3. Architecture of the Adjacent Level Fusion module.

Asymptotic feature elimination. The low-level features have integrated sufficient high-level semantic information in the previous fusion. We gradually remove the influence of non-adjacent high-level features along the top-down path to obtain better results. Through this approach, the number of parameters and FLOPs are drastically reduced.

Unlike the traditional feature pyramid networks, the asymptotic elimination feature pyramid network proposed in this paper progressively integrates deep semantic information and shallow location information by asymptotic elimination of high-level features. This approach gradually mitigates the influence of high-level features on small object detection through the top-down pathway, thereby enhancing detection performance and reducing the model's parameters.

3.4. Asymptotic Weighted Coordinate Fusion for AEFPN

The fully connected feature fusion stage in AEFPN incorporates multi-scale features to augment the feature information. However, as features from different levels may contain contradictory information and vary in importance, balancing spatial precision with semantic richness becomes critical—especially in UAV scenarios, where small objects are prevalent. Adaptive spatial feature fusion (ASFF) combines aligned features through spatially weighted summation into a single feature map, without retaining the individual input branches as separate channel groups for subsequent recalibration. We aim to preserve complementary multi-scale responses and incorporate directional positional context during fusion. To this end, we introduce asymptotic weighted coordinate fusion (AWCF) for UAV imagery.

AWCF combines branch-level weighting with the existing coordinate attention (CA) mechanism [24] through a hierarchical fusion strategy that synergizes with AEFPN to form an integrated framework addressing UAV-specific challenges. Its weighting design draws inspiration from the weighted feature fusion method used in BiFPN [25]. Instead of the weighted summation adopted in BiFPN, AWCF concatenates the normalized weighted input features and keeps the scale-specific responses as independent channel groups before subsequent recalibration. Coordinate attention subsequently encodes horizontal and

vertical positional context to dynamically recalibrate the fused features. This combination integrates branch-level weighting and positional information into AEFPN's multi-scale fusion. This module introduces a decoupled attention mechanism that distinctly separates spatial and channel-wise recalibration, thereby reducing conflicts and inconsistencies among hierarchical features and compensating for the shortcomings of pivotal layers. Through these optimizations, AWCF effectively integrates hierarchical adaptivity with conflict suppression, achieving robust multi-scale feature fusion. Furthermore, AWCF reinforces the hierarchical representations via cross-layer feedback, which, coordinated with AEFPN, mitigates the adverse effects of asymptotic feature elimination, thus ensuring lightweight and efficient feature integration. As depicted in Figure 4, the AWCF module introduces scale-aware weighting factors to fuse input features through a weighted connection operation. This approach enables the network to learn spatial importance across the feature map and adaptively prioritize features from scales most relevant to UAV scenarios. Subsequently, a decoupled attention mechanism, implemented via coordinate attention, is applied to effectively integrate global contextual information with multi-scale local details, thereby enhancing object localization accuracy.

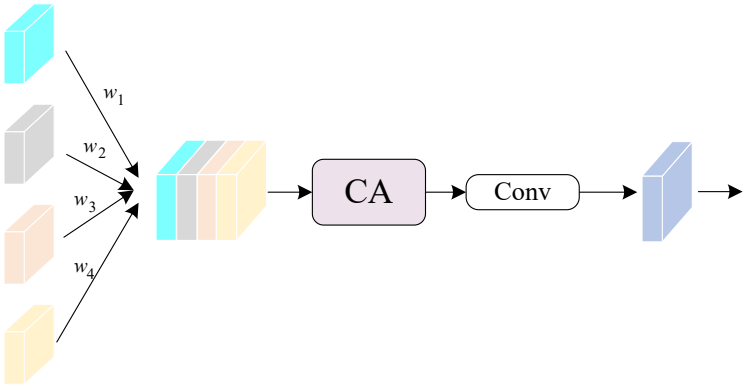


Figure 4. Architecture of the Asymptotic Weighted Coordinate Fusion Module.

Let $x_{ij}^{n\to l}$ be the feature vector at spatial location (i, j) propagated from level n to level l. The corresponding output of the AWCF module is denoted by y_{ij}^{l}, which is formulated as follows:

$$\tilde{x_{ij}}^{k} = \frac{w_k x_{ij}^{k\to l}}{\sum_{k\in l} w_k + \varepsilon} \tag{2}$$

$$y_{ij}^{l} = f^{3\times 3}\Big(\text{Attention}\big([\tilde{x}_{ij}^{1}, \tilde{x}_{ij}^{2}, \tilde{x}_{ij}^{3}, \tilde{x}_{ij}^{4}]\big)\Big) \tag{3}$$

where w_k represents the spatial weights of the features at level k, the summation $\sum_{k\in l} w_k$ denotes the sum of the weights of all input levels participating in fusion at target level l, ε is a constant to prevent data change. $f^{3\times 3}$ denotes the convolution with a 3 × 3 kernel using the sigmoid activation function.

As shown in Figure 5, two 1D feature vectors are generated along the vertical and horizontal coordinates through average pooling operations on the features of each channel. Let x_c denote the feature vector of the c-th channel. The output feature vectors z_c^h at height h and z_c^w at width w from the average pooling operations can be described as:

$$z_c^h(h) = \frac{1}{W}\sum_{1\le i\le W} x_c(h, i) \tag{4}$$

$$z_c^w(w) = \frac{1}{H}\sum_{1\le j\le H} x_c(j, w) \tag{5}$$

The average pooling operation along height and width allows for more precise coordinate information and long-range dependencies of the input feature maps in that direction. This enables the network to locate the objects of interest more precisely. To capture the inter-channel information, the results from the two directions, z^w and z^h, are concatenated before a shared 1×1 convolution function F_1 as follows:

$$f = \delta\Big(F_1\big([z^h, z^w]\big)\Big) \tag{6}$$

where δ denotes the nonlinear activation function. The intermediate representation $f \in \mathbb{R}^{\mathcal{C}/r \times (H+W)}$ incorporates horizontal and vertical encoded information, and r is the reduction ratio to control the intermediate channel dimension. Here, $\mathcal{C}$ denotes the number of input channels. The representation is then divided into $f^h \in \mathbb{R}^{\mathcal{C}/r \times H}$ and $f^w \in \mathbb{R}^{\mathcal{C}/r \times W}$. The height- and width-dependent attention weights g^h and g^w can be obtained by applying separate 1×1 convolution functions F_h and F_w to f^h and f^w, respectively, followed by a sigmoid activation function.

$$g^h = \sigma(F_h(f^h)) \tag{7}$$

$$g^w = \sigma(F_w(f^w)) \tag{8}$$

where σ represents the sigmoid activation function. Ultimately, the output feature vector $y_c(i,j)$ with coordinate attention weights will be calculated with a channel-wise broadcast multiplication operation on the original feature vector.

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{9}$$

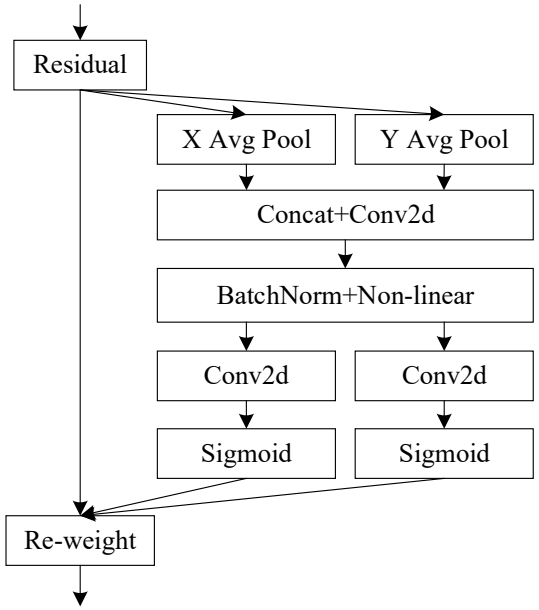


Figure 5. Architecture of the Coordinate Attention Module.

4. Experiments

The experimental validation consists of two complementary parts. Benchmark experiments on public datasets are first conducted to quantify the detection accuracy of AEMFOD through numerical results and qualitative comparisons. Subsequently, the model is deployed on an edge-computing device for real-world UAV flight tests, where its real-time inference performance is examined under practical operating conditions.

4.1. Dataset Experiments

VisDrone2019 [26] is used as the experimental dataset. The VisDrone2019 dataset consists of 10,209 images covering pedestrians, bicycles, cars, tricycles, and other classes, totaling ten object categories. The dataset is divided into 6471 images for training, 548 for validation, 1610 for testing, and 1580 for competition. Comparisons between our proposed method and other algorithms, as well as the ablation study, are performed on this dataset.

DOTA [27] is another large-scale public dataset similar to VisDrone2019. It comprises 15 common categories and 2806 images, distributed as follows: 1411 images for training, 458 for validation, and 937 for testing. All images are captured from drones, making the DOTA dataset particularly suitable for evaluating object detectors in aerial imagery. For both VisDrone-DET2019 and DOTA, all comparison methods are trained on the official training sets and evaluated locally on the corresponding official validation sets. All reported results are from local training and evaluation. The DOTA experiments use horizontal bounding boxes (HBBs) for both training and evaluation.

4.1.1. Implementation Details and Evaluation Metrics

All dataset experiments are performed under Ubuntu 20.04 using Python 3.8, PyTorch 1.12.1, and CUDA 11.6. The computing platform consists of an Intel Xeon Silver 4310 CPU operating at 2.10 GHz and an NVIDIA RTX A6000 GPU with 48 GB of memory. For AEMFOD and the YOLOv5s baseline, training uses 300 epochs of stochastic gradient descent (SGD) with a batch size of 64, momentum of 0.937, weight decay of 0.0005, and an initial learning rate of 0.01. AEMFOD and the YOLOv5s baseline are trained from scratch under the same training settings without pretrained weights. Detection accuracy is evaluated using AP, AP_{50}, AP_{75}, AP_S, AP_M, and AP_L, whereas computational overhead is described by the parameter count and GFLOPs. AP is averaged over all object categories and IoU thresholds from 0.50 to 0.95 in steps of 0.05. AP_{50} and AP_{75} denote average precision across all categories at IoU thresholds of 0.50 and 0.75, respectively. AP_S, AP_M, and AP_L denote AP for small, medium, and large objects, respectively. The parameter count and GFLOPs indicate the model's complexity to some extent.

4.1.2. Comparison with Other Methods

Extensive comparative experiments are conducted on VisDrone2019 and DOTA datasets between AEMFOD and other representative methods. The experimental results are presented in Tables 1 and 2.

It can be seen from the results in the table that AEMFOD achieves the best results with 41.3% AP_{50} and 24.0% AP, obtaining the highest values in both metrics. Some comparison methods in Tables 1 and 2 use pretrained initialization, which may influence the observed performance differences. However, AEMFOD achieves higher AP_{50} and AP than the pretrained comparison methods. Notably, our model demonstrates improvements of 1.8% on AP_{50} and 0.5% on AP in contrast to YOLOv8s. In AP_S, which measures small-object detection performance, AEMFOD achieves 14.0%, compared with 14.4% for Drone-YOLO. On VisDrone, AEMFOD improves AP_M and AP_L from 23.5% and 28.5% for the YOLOv5s baseline to 31.8% and 39.1%, respectively. However, these values remain below the 33.3% and 42.4% achieved by YOLOv8s. AEMFOD prioritizes fine spatial detail and computational efficiency through restricted cross-scale fusion. This design may offer less benefit for medium and large objects that require broader semantic context, indicating a trade-off rather than uniform superiority across object scales. The integration of the AEFPN and AWCF modules in AEMFOD contributed to improved detection accuracy. These components significantly enhance feature integration for object detection in UAV aerial images. Moreover, AEMFOD contains only 6.3 M parameters with a computational

complexity of 16.7 G FLOPs. The parameter count and GFLOPs are lower than those of most comparison algorithms, although they are not the lowest. Our pipeline achieves higher accuracy than lightweight networks. Compared with YOLOv8s, our method achieves higher detection accuracy while reducing the parameter count by 43.8% and GFLOPs by 41.6%. Overall, AEMFOD demonstrates excellent detection accuracy while ensuring low model complexity.

Table 1. Comparisons between the AEMFOD network and various algorithms on the VisDrone-DET2019 dataset.

Methods	Size	AP_{50}(%)	AP_{75}(%)	AP(%)	AP_S(%)	AP_M(%)	AP_L(%)	Params(M)	GFLOPs(G)
SSD [13] *	512 × 512	24.9	12.1	12.9	5.6	20.5	27.9	26.4	179.4
Faster R-CNN [9] *	1333 × 800	36.2	21.7	21.1	13.4	31.6	34.7	28.2	321.6
YOLOv3-tiny [28]	640 × 640	23.4	12.4	13.0	3.7	19.8	33.5	12.1	18.9
YOLOv4s [29]	640 × 640	24.7	13.8	14.2	6.7	17.9	26.5	9.1	21.2
YOLOv5s [30]	640 × 640	31.4	15.7	16.6	9.8	23.5	28.5	7.2	16.5
YOLOX-s [31]	640 × 640	29.1	13.9	15.2	8.7	22.4	29.8	9.0	26.9
YOLOv7-tiny [32]	640 × 640	27.9	12.5	13.9	8.2	21.4	29.1	6.1	**13.3**
YOLOv8s [33]	640 × 640	39.5	23.9	23.5	13.1	33.3	42.4	11.2	28.6
YOLO11s [33]	640 × 640	38.4	21.8	21.9	13.4	32.2	39.1	9.4	21.3
YOLO26s [33]	640 × 640	38.3	21.6	21.9	13.9	30.8	43.9	9.5	20.8
TOOD [34] *	640 × 640	32.9	19.5	19.4	10.7	30.1	39.2	31.8	184.5
DINO [35] *	640 × 640	40.6	21.6	22.4	13.7	31.7	41.1	47.3	232.5
RTD [6]	640 × 640	29.6	14.1	15.3	8.1	20.6	27.1	6.4	14.5
LE-YOLO [36]	640 × 640	39.3	21.9	22.7	13.5	31.6	37.2	3.3	22.9
ADD-YOLO [37]	640 × 640	37.6	21.4	22.1	12.7	29.8	38.8	1.7	17.9
SGA-YOLO [38]	640 × 640	33.4	15.8	18.5	9.7	25.3	34.2	**1.5**	14.4
Edge-YOLO [39]	640 × 640	40.8	22.8	23.1	13.8	34.2	**48.1**	9.3	45.3
TPH-YOLOv5s [20]	640 × 640	37.2	20.3	20.8	11.3	27.5	34.0	6.3	16.6
Drone-YOLO [22]	640 × 640	**41.3**	21.3	23.3	**14.4**	31.5	36.8	10.8	37.3
AEMFOD (Ours)	640 × 640	**41.3**	**24.0**	**24.0**	14.0	31.8	39.1	6.3	16.7

* Methods use pretrained weights. The best performances are marked in bold.

The effectiveness of AEMFOD is additionally confirmed by comprehensive experiments on the DOTA dataset, as listed in Table 2. The model performed consistently on both DOTA and VisDrone datasets, confirming its applicability to aerial scene detection. Compared to current methods, AEMFOD achieved the best results, with AP_{50}, AP_{75}, and AP being 41.5%, 22.7%, and 22.8%, respectively. Besides, the AP_S of AEMFOD is not the best but reaches 6.6%, which outperforms most methods, and the AP_M achieves the best results with 18.1%. Regarding parameter count and GFLOPs, AEMFOD demonstrates comparable results to YOLOv5s and is acceptable when evaluated against YOLOv7-tiny, which exhibits the best performance. Overall, our method demonstrates excellent detection performance on UAV images.

4.1.3. Ablation Study

Ablation studies are conducted on VisDrone to quantify the contributions of the proposed modules. The YOLOv5s model serves as the baseline for comparison. We employ multiple metrics of AP_{50}, AP_{75}, AP, parameters, and GFLOPs to compare the model detection capabilities of the initial version and after adding modules progressively.

Table 2. Comparisons between the AEMFOD network and various algorithms on the DOTA dataset.

Methods	Size	AP_{50}(%)	AP_{75}(%)	AP(%)	AP_S(%)	AP_M(%)	AP_L(%)	Params(M)	GFLOPs(G)
SSD [13] *	512 × 512	23.9	9.9	11.2	2.3	9.3	20.6	26.4	179.2
Faster R-CNN [9] *	1333 × 800	28.3	12.1	14.3	3.3	14.5	22.8	28.2	321.6
YOLOv3-tiny [28]	640 × 640	32.2	19.0	19.2	3.8	12.9	31.3	12.1	18.9
YOLOv4s [29]	640 × 640	31.5	19.8	19.2	3.6	13.9	20.2	9.1	21.2
YOLOv5s [30]	640 × 640	39.6	21.7	20.4	5.1	16.5	30.6	7.2	16.5
YOLOX-s [31]	640 × 640	34.7	19.9	20.2	**7.6**	18.0	33.1	9	26.9
YOLOv7-tiny [32]	640 × 640	36.1	14.3	17.1	4.0	14.9	27.8	6.1	**13.3**
YOLOv8s [33]	640 × 640	40.4	19.8	21.5	6.2	15.6	30.9	11.2	28.6
TOOD [34] *	640 × 640	24.9	7.4	10.7	2.4	9.1	18.4	31.8	184.5
DINO [35] *	640 × 640	38.2	19.6	19.9	6.2	17.9	**36.1**	47.2	232.6
RTD [6]	640 × 640	34.8	15.2	17.4	5.5	14.8	23.8	6.4	14.5
LE-YOLO [36]	640 × 640	40.8	21.8	21.4	6.4	17.9	33.4	3.3	22.9
ADD-YOLO [37]	640 × 640	38.2	20.8	21.4	5.6	17.1	29.5	1.7	17.9
SGA-YOLO [38]	640 × 640	34.1	18.9	20.0	4.9	14.5	27.9	**1.5**	14.4
Edge-YOLO [39]	640 × 640	30.5	15.5	16.4	5.1	16.3	24.8	9.3	45.3
AEMFOD (Ours)	640 × 640	**41.5**	**22.7**	**22.8**	6.6	**18.1**	30.9	6.3	16.6

* Methods use pretrained weights. The best performances are marked in bold.

Table 3 compares performances incorporating different combinations of AEFPN, AWCF, and the tiny-object prediction head. Introducing the tiny-object prediction head leads to the most substantial improvement in detection precision with 20.7% on the AP_{50} index and 28.3% on the AP index. However, it also introduces increases of 2.8% and 22.4% in parameters and GFLOPs. The introduction of AEFPN to YOLOv5s significantly reduces the complexity and parameter count by 20.6% and 26.4%, although the model's accuracy slightly decreased. Without the tiny-object prediction head, adding AWCF yields relative improvements of 5.1%, 3.1%, and 12.2% in AP_{50}, AP_{75}, and AP, respectively, compared with the case without AWCF.

The performance drop in the three-head configuration may be attributed to limited use of low-level features and weakened semantic information flow across high-level fusion branches. These factors result in insufficient feature fusion, reducing model accuracy. In the four-head configuration, adding AEFPN increases AP_{50}, AP_{75}, and AP from 37.9%, 21.2%, and 21.3% to 38.3%, 21.4%, and 21.6%, respectively, while reducing the parameter count by approximately 23.0% and GFLOPs by 23.8% compared with the four-head baseline. This finding underscores the significance of the AEFPN architecture in the four-head configuration, where broader detection-scale coverage and additional high-resolution, fine-grained feature information provide a better basis for multi-scale fusion. AEFPN enhances multi-scale fusion through structural optimization, mitigating feature conflicts and reducing model complexity. These results indicate that AEFPN more effectively organizes and uses multi-scale features within this architecture, achieving a modest improvement in detection accuracy while reducing model complexity. With the tiny-object prediction head, the AWCF module improves AP_{50}, AP_{75}, and AP by approximately 7.8%, 12.1%, and 11.1%, respectively. This module enhances the quality of feature fusion by adaptively suppressing over-activation in deeper layers, typically dominant in detecting large objects, and reallocating attention toward crucial finer details. The results show that integrating AWCF into AEFPN improves detection performance, supporting the effectiveness of their collaborative design. As AWCF is designed for the AEFPN architecture, its independent contribution requires further investigation.

Table 3. Results of the ablation study.

Methods	AEFPN	AWCF	Head	AP_{50}(%)	AP_{75}(%)	AP(%)	Params(M)	GFLOPs(G)
YOLOv5s	-	-	-	31.4	15.7	16.6	7.2	16.5
YOLOv5s + AEFPN	✓	-	-	31.2	16.1	15.6	5.3	13.1
YOLOv5s + AEFPN + AWCF	✓	✓	-	32.8	16.6	17.5	5.8	14.1
YOLOv5s + Head	-	-	✓	37.9	21.2	21.3	7.4	20.2
YOLOv5s + Head + AEFPN	✓	-	✓	38.3	21.4	21.6	5.7	15.4
AEMFOD (Ours)	✓	✓	✓	**41.3**	**24.0**	**24.0**	6.3	16.7

The best performances are marked in bold.

In addition, the standalone performance of the multi-head component is approximately 3.4% lower than that of the complete AEMFOD system. It indicates that the multi-head component significantly benefits from integrating with AEFPN and AWCF, resulting in superior feature fusion. This integrated approach mitigates conflicts between cross-scale feature inconsistencies while improving the utilization of feature information. Their principal value lies in establishing a scale-aware feature hierarchy that the multi-head component can exploit efficiently. Ultimately, it is the synergy among these components that reduces model complexity while enhancing detection accuracy.

Furthermore, Gradient-weighted Class Activation Mapping (Grad-CAM) is used to visualize attention regions, providing an intuitive analysis of the effectiveness of the proposed modules. As presented in Figure 6, although the additional branch improves scale matching, the network still suffers from significant feature aliasing. The attention is diffusely distributed, with high activation responses incorrectly appearing in task-irrelevant background regions, such as the sky, roadside vegetation, and distant buildings. This phenomenon indicates that deep semantic noise interferes with small-object localization, thereby constraining effective feature fusion. After integrating AEFPN, background activations are substantially suppressed. The asymptotic elimination mechanism effectively acts as a spatial filtering process, progressively removing redundant high-level semantic dependencies that are irrelevant to small-object detection, thus "cleaning" the fused feature representations. With the further incorporation of the AWCF module, the high-activation regions become tightly concentrated on object bodies. The network exhibits precise spatial recalibration, clearly distinguishing foreground objects from the suppressed background. The results indicate that AWCF incorporates spatial cues into channel-wise recalibration, enabling the network to emphasize features associated with small objects.

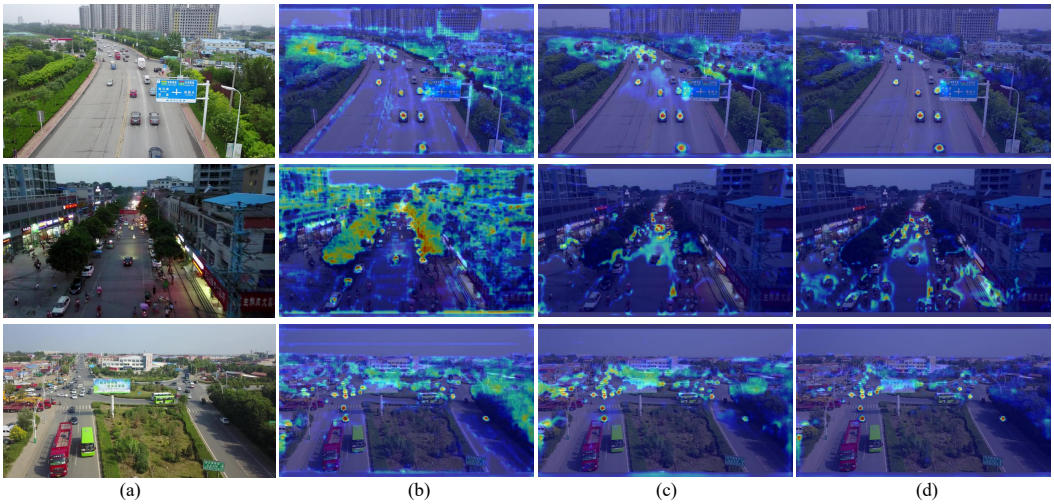


Figure 6. Heatmap visualization. (**a**) Original images. (**b**) With Head. (**c**) With AEFPN. (**d**) With AWCF.

To assess the contribution of AEFPN, we further evaluate the baseline model equipped with several alternative feature pyramid structures on the VisDrone2019 dataset. Additionally, we compare baseline models equipped with tiny prediction heads. The corresponding comparison is reported in Table 4. With the original baseline, BiFPN performs the best in AP_{50} on the baseline and baseline-4Head. Yet, its performance in AP_{75} and AP is unsatisfactory and significantly increases model parameters and complexity. AFPN achieves the lowest AP_{50} and AP_{75} among the compared FPN structures in both configurations. Compared with AFPN, AEFPN increases the AP_{50} from 30.4% to 31.2% and from 36.7% to 38.3%, while reducing the parameter count and GFLOPs. Compared with BiFPN and PAFPN, AEFPN prioritizes reducing complexity over improving detection accuracy. However, on the baseline-4Head model, AEFPN achieved 25.0% and nearly 23.0% reductions in parameters and 28.4% and 23.8% reductions in GFLOPs with comparable detection performance. This result supports the use of AEFPN in the four-head architecture, where it limits complexity while preserving comparable detection performance. By integrating these three modules, our proposed AEMFOD achieves 41.3% detection accuracy on AP_{50} with a slight increase in computational complexity. Our approach balances detection accuracy and model complexity by substantially improving detection accuracy with only a small increase in model complexity.

Table 4. Comparison of different FPN architectures.

Methods	Neck	AP_{50}(%)	AP_{75}(%)	AP(%)	Params(M)	GFLOPs(G)
Baseline	PAFPN	31.4	15.7	**16.6**	7.2	16.5
	BiFPN	**32.9**	**16.1**	16.2	7.2	16.7
	AFPN	30.4	15.0	16.0	6.2	15.4
	AEFPN	31.2	**16.1**	15.6	**5.3**	**13.1**
Baseline-4Head	PAFPN	37.9	21.2	21.3	7.4	20.2
	BiFPN	**38.6**	21.1	**21.7**	7.6	21.5
	AFPN	36.7	20.8	21.0	6.5	19.5
	AEFPN	38.3	**21.4**	21.6	**5.7**	**15.4**

The best performances are marked in bold.

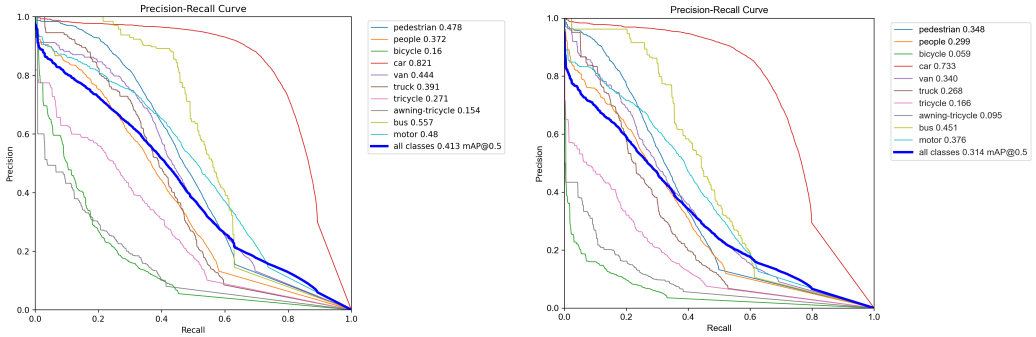
Figure 7 compares the Precision–Recall (P–R) curves of YOLOv5s and AEMFOD on VisDrone2019. Besides the P–R curves for individual object categories in the dataset, the legend displays the respective AP_{50} of each class. The results in Figure 7 illustrate that the proposed AEMFOD outperforms YOLOv5s in detection precision across all object categories, especially for the bicycle, truck, tricycle and bus categories. The AP_{50} for trucks is improved by up to 46% and for tricycle nearly 63%. The smallest improvement is for cars, at 12%. The pedestrian and people categories, which are more numerous in the dataset, show improvements of 37% and nearly 24%, respectively. The performance gains of AEMFOD are associated with its aggregation of features at multiple resolutions and its incorporation of positional context from low-level feature maps, further supporting the effectiveness of the proposed modules.

To evaluate the applicability of proposed designs beyond YOLOv5s, we integrate AEFPN, AWCF, and the tiny-object prediction head into YOLOv8s and YOLO11s while retaining their native backbones and basic feature-extraction modules. The model training settings remain consistent with the experiments described above. Similarly, no pre-trained weights are used. As shown in Table 5, incorporating the proposed modules increases AP_{50} from 39.5% to 43.6% and AP from 23.5% to 24.9% for YOLOv8s, while reducing the parameter count from 11.2 M to 8.5 M. For YOLO11s, AP_{50} increases from 38.4% to 42.7% and AP from 21.9% to 24.5%, with the parameter count decreasing from 9.4 M to 8.3 M.

However, GFLOPs increase for both architectures, and AP_L decreases from 42.4% to 37.9% on YOLOv8s and from 39.1% to 37.9% on YOLO11s. On YOLOv8s, AP_{75} also decreases from 23.9% to 22.4%. In contrast, AP_M increases from 33.3% to 33.8% on YOLOv8s and from 32.2% to 33.5% on YOLO11s. AEFPN's progressive restriction of selected deeper-level fusion branches may limit the deep semantic context available for large-object detection. The drop in AP_{75} on YOLOv8s also indicates performance trade-offs at the stricter IoU threshold. These results show that the proposed design is equally applicable to newer YOLO architectures, although improvements are not consistent across all metrics.

Table 5. Applicability to Other YOLO Architectures.

Methods	AP_{50}(%)	AP_{75}(%)	AP(%)	AP_S(%)	AP_M(%)	AP_L(%)	Params(M)	GFLOPs(G)
YOLOv8s [33]	39.5	23.9	23.5	13.1	33.3	42.4	11.2	28.6
YOLOv8s + AEMFOD modules	43.6	22.4	24.9	14.9	33.8	37.9	8.5	32.1
YOLO11s [33]	38.4	21.8	21.9	13.4	32.2	39.1	9.4	21.3
YOLO11s + AEMFOD modules	42.7	22.0	24.5	14.1	33.5	37.9	8.3	26.4



(a) (b)

Figure 7. Precision–Recall curve obtained by (**a**) AEMFOD and (**b**) YOLOv5s.

4.1.4. Edge Deployment Comparison with TensorRT

AEMFOD is compared with other lightweight detectors on the NVIDIA Jetson Xavier NX onboard computer to validate its effectiveness in edge deployment scenarios. To assess the proposed detector from both performance and deployment perspectives, we report its detection accuracy along with model size and inference speed. The FPS is calculated using inference time during validation on the VisDrone dataset, excluding preprocessing and non-maximum suppression (NMS). All models are converted to TensorRT engines with FP16 precision and evaluated on the VisDrone dataset with a batch size of 1 and input image size of 640 × 640. The engine storage size is also included as a comparison metric.

The results are shown in Table 6. When deployed with FP16 TensorRT on the Jetson NX platform, LE-YOLO achieves the smallest model size among all methods; however, its inference speed is relatively low. Edge-YOLO attains competitive detection accuracy but exhibits relatively low FPS. Drone-YOLO and AEMFOD have the same AP_{50}, while AEMFOD achieves higher AP and FPS. AEMFOD achieves higher detection accuracy than TPH-YOLOv5s, although its inference FPS is lower. As shown in Table 6, AEMFOD achieves the highest AP and joint-highest AP_{50} among the compared methods while maintaining competitive inference FPS. Moreover, the AP_{50} and AP metrics remain consistent with those obtained from the PyTorch evaluation, indicating that the proposed architecture is robust to reduced-precision computation and deployment optimization.

Figure 8 illustrates qualitative detection results of YOLOv5, LE-YOLO, and the proposed AEMFOD under FP16 precision in representative UAV-based traffic scenarios with varying flight altitudes, illumination conditions, and traffic densities. In high-altitude

scenes with a large number of small-scale vehicles (a2–c2), YOLOv5 exhibits evident missed detections and localization errors, particularly in dense traffic regions. LE-YOLO improves small-object detection performance; however, it still suffers from incomplete detections and bounding box overlaps in highly congested areas. AEMFOD achieves more accurate and complete detections, effectively preserving small and densely distributed vehicles. Similar situations also occur in lower-altitude scenarios (a1–c1). AEMFOD shows favorable detection performance in complex urban traffic scenes. In the illustrated bright-light (a3–c3) and nighttime (a4–c4) scenes, YOLOv5 and LE-YOLO exhibit missed detections and lower detection confidence, while AEMFOD provides more complete detection results. Overall, AEMFOD demonstrates favorable detection performance across the depicted drone traffic scenarios.

Table 6. Comparison results with TensorRT.

Methods	Size (MB)	AP_{50}(%)	AP(%)	FPS
YOLOv5s	17.2	31.4	16.6	**64.9**
YOLOv8s	24.9	39.6	23.7	49.0
LE-YOLO	**12.0**	39.1	21.7	37.5
Edge-YOLO	27.1	40.6	23.1	26.6
TPH-YOLOv5s	16.8	37.2	20.7	55.2
Drone-YOLO	26.3	**41.3**	23.3	27.0
AEMFOD (Ours)	20.6	**41.3**	**23.8**	45.2

FPS $= 1000/\bar{t}_{\text{infer}}$. The best performances are marked in bold.

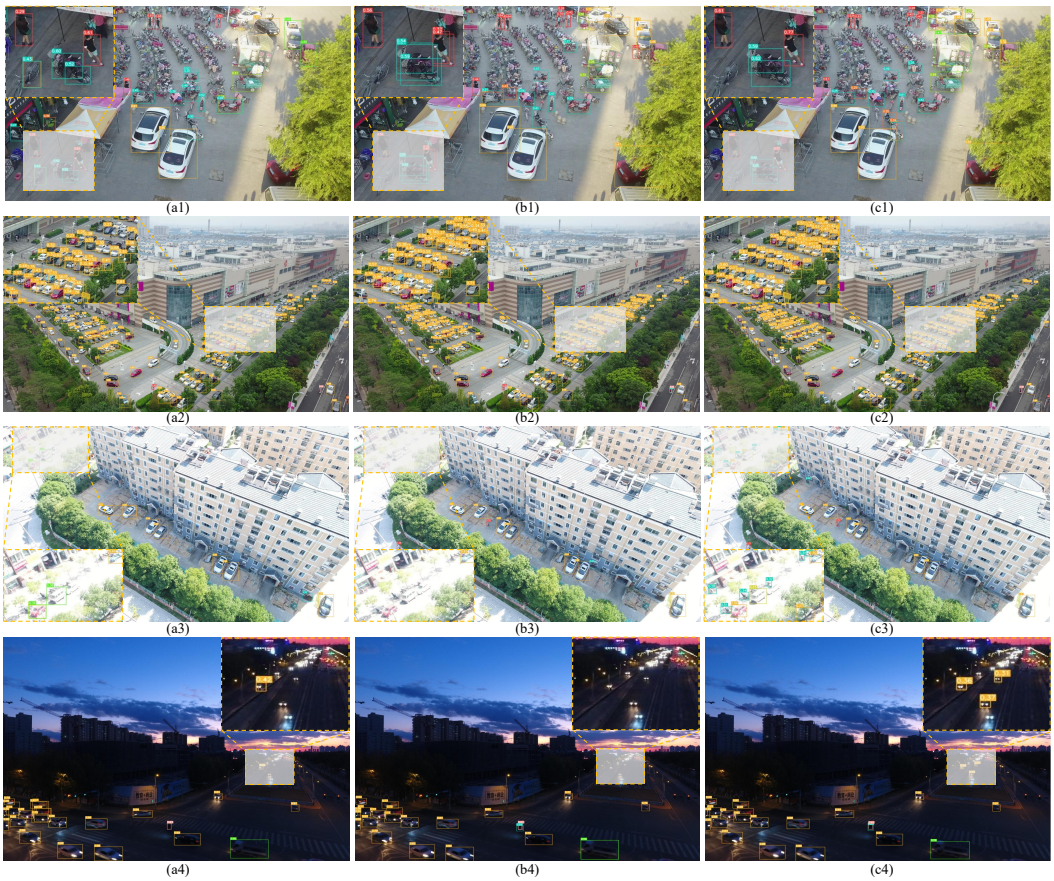


Figure 8. Comparison of visualizations on VisDrone-2019 dataset. a1-a4 denotes the detection results of YOLOv5. b1-b4 denotes the detection results of LE-YOLO. c1-c4 denotes the detection results of AEMFOD. (a1, b1, c1) A lower-altitude traffic scene; (a2, b2, c2) a higher-altitude scene with densely distributed vehicles; (a3, b3, c3) a bright-light scene; (a4, b4, c4) a nighttime scene. Yellow dashed rectangles and connecting lines identify selected regions and their corresponding magnified views.

4.2. Flight Experiments

To demonstrate airborne real-time performance and provide qualitative detection examples, we conducted a single flight test with the autonomous quadrotor shown in Figure 9. The quadrotor weighs approximately 1.5 kg and has a diameter of 410 mm. The flight was conducted at an approximate altitude of 30–60 m. Control commands are sent to the motors through a Pixhawk V4 flight control board, and the detection algorithm runs on an NVIDIA Jetson Xavier NX onboard computer. Image acquisition is performed using a G1-OV OS12D40 monocular camera. The monocular camera records 1280 × 720 images at 30 FPS. Its horizontal and vertical fields of view are 125° and 69°, respectively. Onboard inference is executed on the Jetson NX, which integrates a 384-core NVIDIA Volta™ GPU with 48 Tensor Cores and a six-core NVIDIA Carmel Arm® v8.2 64-bit CPU. The device provides up to 21 TOPS (INT8) of peak AI performance and 8 GB of RAM. NVIDIA JetPack 4.4.1, CUDA 10.2, cuDNN 8.0, and Ubuntu 18.04 LTS constitute the software environment. Before deployment, the model is converted to TensorRT for inference acceleration.



Figure 9. Platforms in flight experiments. (**a**) Quadrotor. (**b**) NVIDIA Jetson Xavier NX.

On computationally constrained embedded devices, the inference speed of AEMFOD, YOLOv5s, and YOLOv8s is evaluated using frames per second (FPS) under FP16 and FP32 precision. The FPS is measured for a single flight image using inference time only, excluding preprocessing and NMS. Table 7 reports that the proposed AEMFOD outperforms YOLOv8s in FPS under FP32 precision. The detection speed of AEMFOD is 1.4 frames/s faster than that of YOLOv8s. YOLOv5s achieves the highest speed under both FP32 and FP16 precision, while its accuracy is the lowest. Compared with YOLOv5s and YOLOv8s, AEMFOD exhibits lower inference speed under FP16. However, this trade-off in inference speed is coupled with improved detection accuracy, providing a practical balance between accuracy and inference latency for onboard UAV detection.

Table 7 demonstrates a significant discrepancy in FPS between YOLOv5s and AEMFOD under FP16 and FP32 precision. To further investigate the effects of AEFPN and AWCF on real-time performance, we conduct ablation studies across different precision levels. In Table 8, the experimental results indicate that reducing model precision from FP32 to FP16 results in a 143.3% enhancement in real-time performance for the baseline model. However, with the integration of AEFPN, this improvement is reduced to 74.2%. Furthermore, the addition of AWCF achieves an 88.3% increase in FPS. The primary factor contributing to this diminished speed gain is the fully connected feature fusion mechanism employed in AEFPN, which introduces multi-scale features with significant semantic discrepancies, thereby reducing the speed improvement when transitioning from FP32 to FP16. The introduction of AWCF enables the model to focus on valid features, diminishing

the influence of features with significant semantic gaps on feature fusion. It alleviates the reduction in real-time performance improvement to some extent.

Table 7. Test results on the embedded platform.

Precision	Model	FPS
FP32	YOLOv5s	28.4
	YOLOv8s	19.2
	AEMFOD	20.6
FP16	YOLOv5s	69.1
	YOLOv8s	45.9
	AEMFOD	38.8

FPS $= 1000/t_{\text{infer}}$.

The qualitative detection results obtained during flight are compared to illustrate the performance of the evaluated methods in the presented scenes. Figure 10 shows the detection results, where objects are marked with rectangles of different colors and accompanied by confidence scores. As shown in Figure 10(a1–a4), YOLOv5 exhibits noticeable missed detections and inaccurate localization in crowded scenes, particularly for small and partially occluded objects. Specifically, in Figure 10(a2,a3), only a few distant trucks are detected, while multiple background structures are incorrectly classified as pedestrians. YOLOv8 demonstrates improved detection performance compared with YOLOv5, especially in terms of localization accuracy and object recall. However, as illustrated in Figure 10(b2,b4), YOLOv8 still suffers from incomplete detections and redundant bounding boxes. In contrast, AEMFOD consistently achieves more complete and accurate detection results. It effectively detects small and densely distributed targets while maintaining clearer object boundaries and fewer false negatives. This advantage is particularly evident in complex scenes with severe occlusion and large perspective distortion. The flight results provide qualitative examples of onboard detection, while quantitative detection accuracy is evaluated on the public datasets.

Table 8. Results of the ablation study under different precisions.

Methods	Precision	FPS
Baseline	FP32	28.4
	FP16	69.1(↑143.3%)
Baseline+AEFPN	FP32	22.5
	FP16	39.2(↑74.2%)
AEMFOD	FP32	20.6
	FP16	38.8(↑88.3%)

5. Conclusions

This work develops AEMFOD, a high-accuracy and resource-efficient object detector for UAV-based traffic monitoring. AEMFOD improves detection accuracy significantly over the YOLOv5s baseline while retaining a compact architecture. Embedded inference measurements and the flight demonstration further support its feasibility for onboard UAV detection. An additional prediction head is incorporated to strengthen low-level information. The asymptotic elimination feature pyramid network (AEFPN) enhances spatial detail preservation through a structured elimination process, effectively reducing FLOPs and parameters. Within AEFPN, the asymptotic weighted coordinate fusion (AWCF) module employs a cross-scale attention mechanism to harmonize the feature hierarchy, mitigate inter-scale conflicts, and improve feature fusion. Together, AEFPN and AWCF

establish a robust, scale-aware feature hierarchy that maximizes the benefits of the multi-head strategy. As a result, the network strengthens the representation of small targets by reducing redundant cross-scale interactions while keeping the model structure lightweight.

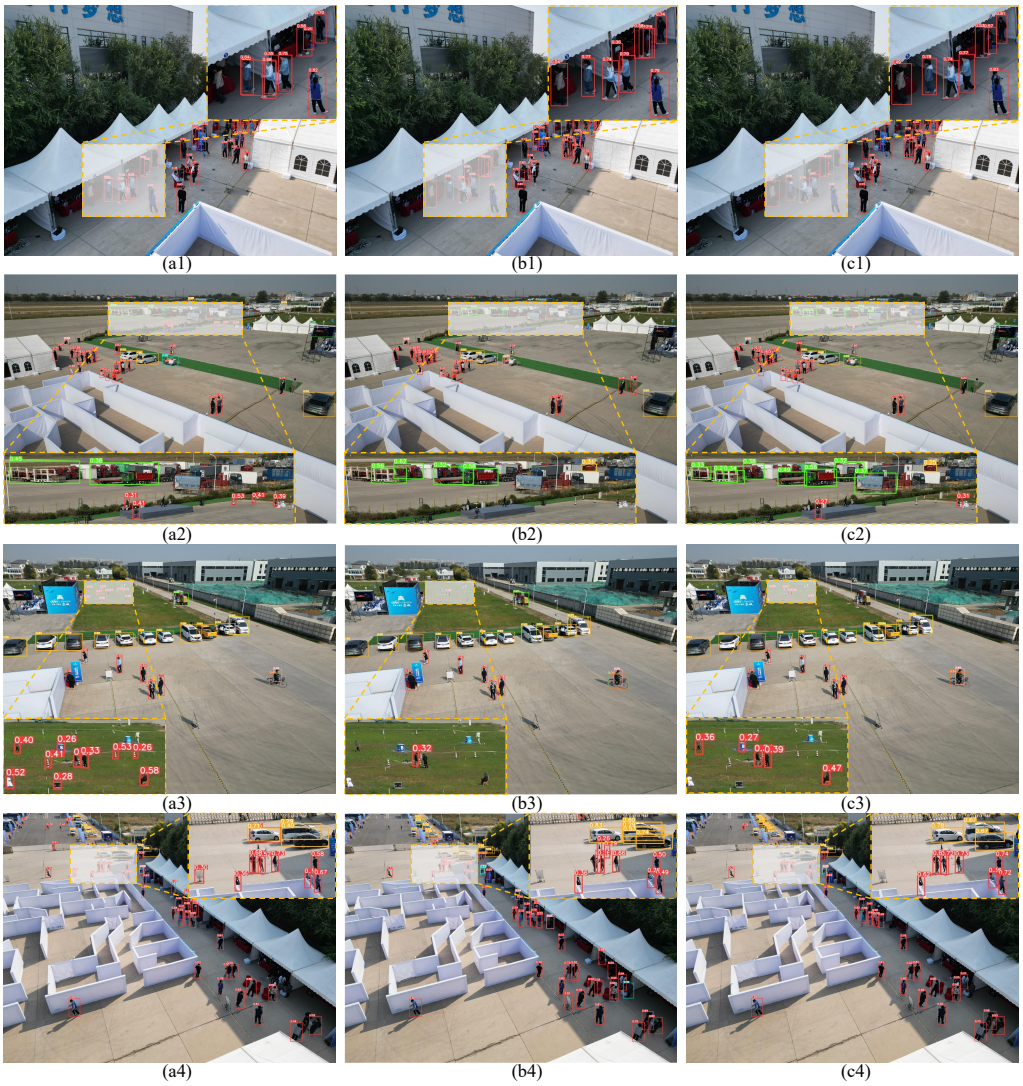


Figure 10. Visualization of the detection results. (a1-a4), (b1-b4), and (c1-c4) show the detection results of YOLOv5, YOLOv8, and AEMFOD, respectively. Rows 1-4 correspond to four representative scenes from the flight test. Yellow dashed rectangles and connecting lines identify selected regions and their corresponding magnified views.

Evaluation on VisDrone2019 and DOTA confirms the detection performance of the proposed architecture. AEMFOD delivers a 31.5% relative increase in AP_{50} on VisDrone and 4.8% on DOTA compared to YOLOv5s. Moreover, the flight experiments illustrate onboard operation and qualitative detection performance in the evaluated scenes. The results also show more complete detection of small, densely distributed, and partially occluded objects in the evaluated scenes.

Overall, AEMFOD establishes a practical compromise among detection accuracy, inference latency, and model compactness, making it ideal for deployment on drones in practical scenarios. For intelligent transportation applications, robust aerial detection enables scalable traffic monitoring from UAVs. Accurate aerial perception can support traffic-monitoring applications such as congestion assessment, collision warning, and incident response. Although AEMFOD performs well in the illustrated bright-light and nighttime scenes, these observations do not demonstrate robustness to drastic illumi-

nation changes. Robustness to light changes and motion blur requires further investigation. In the future, data augmentation can be considered to improve performance under these conditions. Furthermore, the complementary properties of different sensors equipped on the UAV can be utilized to achieve more robust and accurate detection in complex environments.

Author Contributions: Conceptualization, B.S.; methodology, B.S.; validation, B.S. and J.Z.; resources, Z.C.; data curation, J.Z.; writing-original draft preparation, B.S.; writing-review and editing, J.Z. and Y.W.; funding acquisition, Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology on Aircraft Control Laboratory under grant 11300LB2022103007.

Data Availability Statement: The dataset used in this study is available from the corresponding author upon reasonable request.

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

References

1. Yue, T.; Huang, H.; Song, B.; Chen, Y.; Chen, Z. MSF-YOLO: A multi-scale feature enhancement network for tiny fire spot detection in UAV forest images. *Drones* **2026**, *10*, 662. https://doi.org/10.3390/drones10090662.
2. Sirma, A.; Plastropoulos, A.; Tang, G.; Zolotas, A. E'CHIT: Identity-stable operator-centric UAV tracking for disaster response. *Drones* **2026**, *10*, 637. https://doi.org/10.3390/drones10080637.
3. Liu, Y.; Xu, C.; Xiao, K.; Wu, Y.; Li, Z. GPOD: Geographic priors and object detection for candidate-guided target localization in city-scale UAV vision-and-language navigation. *Drones* **2026**, *10*, 458. https://doi.org/10.3390/drones10060458.
4. Ying, Z.; Zhou, J.; Zhai, Y.; Quan, H.; Li, W.; Genovese, A.; Piuri, V.; Scotti, F. Large-scale high-altitude UAV-based vehicle detection via pyramid dual pooling attention path aggregation network. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 14426–14444. https://doi.org/10.1109/TITS.2024.3396915.
5. Wu, D.; Jiang, S.; Zhao, E.; Liu, Y.; Zhu, H.; Wang, W.; Wang, R. Detection of *Camellia oleifera* fruit in complex scenes by using YOLOv7 and data augmentation. *Appl. Sci.* **2022**, *12*, 11318. https://doi.org/10.3390/app122211318.
6. Ye, T.; Qin, W.; Zhao, Z.; Gao, X.Z.; 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. https://doi.org/10.1109/TIM.2023.3241825.
7. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587. https://doi.org/10.1109/CVPR.2014.81.
8. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. https://doi.org/10.1109/ICCV.2015.169.
9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
10. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. https://doi.org/10.1109/ICCV.2017.322.
11. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into high quality object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162. https://doi.org/10.1109/CVPR.2018.00644.
12. 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; pp. 779–788. https://doi.org/10.1109/CVPR.2016.91.
13. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.-Y.; Berg, A.C. SSD: Single shot MultiBox detector. In *Computer Vision–ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; Volume 9905, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.
14. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007. https://doi.org/10.1109/ICCV.2017.324.

15. 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 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 100–108. https://doi.org/10.1109/ICCVW.2019.00018.
16. Yu, W.; Yang, T.; Chen, C. Towards resolving the challenge of long-tail distribution in UAV images for object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 5–9 January 2021; pp. 3258–3267.
17. Zhang, R.; Newsam, S.; Shao, Z.; Huang, X.; Wang, J.; Li, D. Multi-scale adversarial network for vehicle detection in UAV imagery. *ISPRS J. Photogramm. Remote Sens.* **2021**, *180*, 283–295. https://doi.org/10.1016/j.isprsjprs.2021.08.002.
18. Bosquet, B.; Cores, D.; Seidenari, L.; Brea, V.M.; Mucientes, M.; Del Bimbo, A. A full data augmentation pipeline for small object detection based on generative adversarial networks. *Pattern Recognit.* **2023**, *133*, 108998. https://doi.org/10.1016/j.patcog.2022.108998.
19. Zhang, P.; Zhong, Y.; Li, X. SlimYOLOv3: Narrower, faster and better for real-time UAV applications. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 37–45. https://doi.org/10.1109/ICCVW.2019.00011.
20. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on Transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, QC, Canada, 11–17 October 2021; pp. 2778–2788.
21. Zhang, Z.; Lu, X.; Cao, G.; Yang, Y.; Jiao, L.; Liu, F. ViT-YOLO: Transformer-based YOLO for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, QC, Canada, 11–17 October 2021; pp. 2799–2808. https://doi.org/10.1109/ICCVW54120.2021.00314.
22. Zhang, Z. Drone-YOLO: An efficient neural network method for target detection in drone images. *Drones* **2023**, *7*, 526. https://doi.org/10.3390/drones7080526.
23. Yang, G.; Lei, J.; Zhu, Z.; Cheng, S.; Feng, Z.; Liang, R. AFPN: Asymptotic Feature Pyramid Network for Object Detection. *arXiv* **2023**, arXiv:2306.15988. https://doi.org/10.48550/arXiv.2306.15988.
24. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 13713–13722. https://doi.org/10.1109/CVPR46437.2021.01350.
25. 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 (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790. https://doi.org/10.1109/CVPR42600.2020.01079.
26. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Ling, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y.; et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 213–226. https://doi.org/10.1109/ICCVW.2019.00030.
27. Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.J.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 3974–3983. https://doi.org/10.1109/CVPR.2018.00418.
28. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767. https://doi.org/10.48550/arXiv.1804.02767.
29. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934. https://doi.org/10.48550/arXiv.2004.10934.
30. Ultralytics. YOLOv5. Available online: https://github.com/ultralytics/yolov5 (accessed on 8 September 2026).
31. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO series in 2021. *arXiv* **2021**, arXiv:2107.08430. https://doi.org/10.48550/arXiv.2107.08430.
32. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00721.
33. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO. Available online: https://github.com/ultralytics/ultralytics (accessed on 8 September 2026).
34. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-aligned one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 3490–3499. https://doi.org/10.1109/ICCV48922.2021.00349.
35. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.-Y. DINO: DETR with improved DeNoising anchor boxes for end-to-end object detection. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.

36. Yue, M.; Zhang, L.; Huang, J.; Zhang, H. Lightweight and efficient tiny-object detection based on improved YOLOv8n for UAV aerial images. *Drones* **2024**, *8*, 276. https://doi.org/10.3390/drones8070276.
37. Ye, Q.; Ma, M.; Zhao, X.; Duan, B.; Wang, L.; Ma, D. ADD-YOLO: An algorithm for detecting animals in outdoor environments based on unmanned aerial imagery. *Measurement* **2025**, *242*, 116019. https://doi.org/10.1016/j.measurement.2024.116019.
38. Ge, P.; Wan, M.; Qian, W.; Xu, Y.; Kong, X.; Gu, G.; Chen, Q. SGA-YOLO: A lightweight real-time object detection network for UAV infrared images. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 22432–22446. https://doi.org/10.1109/TITS.2025.3616525.
39. Liu, S.; Zha, J.; Sun, J.; Li, Z.; Wang, G. EdgeYOLO: An edge-real-time object detector. In Proceedings of the 42nd Chinese Control Conference (CCC), Tianjin, China, 24–26 July 2023; pp. 7507–7512. https://doi.org/10.23919/CCC58697.2023.10239786.

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