

Article

IDMSNet: Improved DETR Based on Multi-Scale Feature Fusion for Small Object Detection in UAV Images

Chuyu Miao ¹, Xiangbin Kong ^{1,2,*} and Zhe Luo ¹

¹ School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan 430068, China; 102410254@hbut.edu.cn (C.M.); 102410258@hbut.edu.cn (Z.L.)

² Hubei Key Laboratory for High-Efficiency Utilization of Solar Energy and Operation Control of Energy Storage System, Hubei University of Technology, Wuhan 430068, China

* Correspondence: kongxb@hbut.edu.cn

Abstract

With the rapid development of UAV technology, UAV image object detection is also facing many challenges, especially for small objects, including dense distribution, feature loss, and background interference. To address these issues, this paper proposes a novel object detection model with dynamic competitive fusion. The framework consists of three major innovations: Firstly, a dynamic competitive multi-scale feature fusion structure is adopted. Combined with a dual-path parallel attention mechanism, it ensures sufficient retention of small object features. Secondly, a small object bias mechanism is introduced. It establishes small object regions through dynamic learning and generates stronger feature responses for small targets. Thirdly, a hybrid matching strategy is introduced to provide the model with more abundant supervision signals and optimize its feature learning capability. Experiments on the VisDrone2019 and COCO2017 datasets demonstrate that, compared with various existing methods, the proposed model improves detection accuracy while maintaining comparable efficiency, and achieves a prominent performance gain especially for small objects. It verifies that the model has good generalization ability, exhibits robustness in handling small targets, and confirms the deployment potential of the proposed model in the practical application of UAV image object detection.

Keywords: unmanned aerial vehicle; small object detection; feature fusion; hybrid matching

1. Introduction

With the rapid development and popularization of unmanned aerial vehicle (UAV) technology, computer vision tasks based on UAV platforms, especially object detection, have demonstrated enormous application potential in fields such as smart cities, traffic monitoring, agricultural surveys, disaster rescue, and national defense security [1]. The UAV perspective provides us with unprecedented flexibility and a macroscopic view, but it also brings some technical challenges, which are mainly reflected in the following aspects: targets in images usually feature extremely small scales, dense distribution, and high spatial randomness, and are often affected by factors such as illumination changes and complex background interference [2].

Despite the considerable progress achieved by object detection models based on Convolutional Neural Networks (CNNs), they still fall short in processing UAV images. These models often rely on preset anchor boxes and complex manually designed components, and their performance will significantly degrade when facing extreme scale variations



Academic Editor: Fabio D'Andreagiovanni

Received: 6 July 2026

Revised: 27 July 2026

Accepted: 29 July 2026

Published: 4 August 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

particularly small objects. In recent years, Detection Transformer (DETR) has brought new breakthroughs to the field of detection by virtue of its end-to-end detection paradigm and strong ability to model global contextual information. Furthermore, Deformable DETR has effectively alleviated the problems of slow convergence and high computational cost of the original DETR model by introducing the deformable attention mechanism, which makes it a powerful candidate for processing high-resolution images.

However, standard Deformable DETR and its variants still have room for improvement when addressing the “small object dilemma” in UAV images. Its main problems are as follows: First, the features of small objects are very prone to being submerged or lost in deep networks. The model’s attention may be dominated by large-area backgrounds or prominent large objects, resulting in insufficient attention to small objects. Second, the strict one-to-one matching adopted by the conventional DETR model leads to sparse positive samples, which slows down the feature learning and optimization efficiency of the model.

To address the aforementioned issues, this paper proposes a novel object detection model IDMSNet with dynamic competitive fusion, which consists of basic components including backbone network, encoder and decoder. Unlike existing DETR variants that adopt equal-weight or static weighted feature fusion strategies, this paper innovatively designs a dynamic multi-scale feature fusion architecture with a small object bias mechanism. The introduction of this bias mechanism effectively addresses the critical problem that small-object features in UAV-borne images tend to be overwhelmed or diluted during the fusion process. Different from the strict one-to-one matching structure adopted by existing models, we introduce a hybrid matching strategy to optimize the efficiency of positive sample assignment, ultimately achieving improved detection accuracy while preserving the comparable efficiency of the model.

In summary, the main contributions of this paper are summarized as follows:

- This paper designs a novel object detection model IDMSNet. It adopts dynamic competitive multi-scale feature fusion, learnable small object bias mechanism and hybrid matching strategy, boosting the detection accuracy of UAV images especially for small objects while maintaining model inference efficiency.
- To address the issue that small-object features are prone to being overwhelmed in deep networks, this paper designs a dynamic competitive multi-scale feature fusion structure. Through a dual-path parallel attention mechanism and competitive fusion weights, it prevents the submersion of small target features and retains more complete detailed information of small objects. This paper also designs a dynamically learnable small object bias mechanism. It realizes differentiated enhancement of small target features through weight learning and dynamic threshold setting.
- To address the problem of positive sample sparsity caused by the strict one-to-one matching adopted by existing models, this paper introduces a hybrid matching strategy. Adopting the hybrid matching strategy of one-to-one branch and one-to-many branch, providing more abundant supervision signals and optimizing the efficiency of model resource utilization.
- Through experimental validation, compared with existing methods, the method proposed in this paper improves the accuracy of UAV image object detection while maintaining inference efficiency, and demonstrates advanced capabilities and good generalization ability in improving the detection accuracy of small targets.

2. Related Work

2.1. Applications of DETR in UAV Image Object Detection

At present, one-stage object detection is mainly represented by the YOLO [3] algorithm. In recent years, YOLOv6 [4] and YOLOv7 [5] have focused on edge device

adaptation and real-time inference, while YOLOv8 [6] has achieved comprehensive performance improvement. Many scholars have proposed relevant improvements on this basis [7,8]. Two-stage object detection algorithms are mainly represented by Faster R-CNN [9]. However, traditional detectors often suffer from performance limitations in the Non-Maximum Suppression (NMS) stage, which deteriorates detection results.

To address the problem of NMS, the end-to-end Transformer-based [10] model DETR has been proposed in recent years. The DETR [11] model, which eliminates the design of NMS and manual anchor boxes, provides a new idea for solving the missed detection and false detection caused by dense and overlapping targets in UAV images. However, its slow convergence and insufficient performance in small object detection limit its direct application. Deformable DETR proposed by Zhu et al. [12] focuses on a small number of key points on feature maps through the deformable attention mechanism, significantly improving computational efficiency and detection capability for tiny objects. DN-DETR [13] by Li et al. introduced a denoising training strategy, which stabilizes training and accelerates convergence by reconstructing noisy ground truth boxes, facilitating the rapid deployment of DETR models in UAV applications. With model optimization, real-time performance has become a key factor. RT-DETR proposed by Zhao et al. [14] initializes object queries from image content through IoU-aware query selection, which is more adaptable to the characteristics of uneven target distribution in UAV images, further advancing the practical deployment of DETR in the UAV field.

2.2. Application of Multi-Scale Feature Fusion in DETR

In UAV image object detection, addressing the core challenges of variable target scales and the prevalence of tiny objects, multi-scale feature fusion technology has undergone continuous evolution since its foundation was laid by FPN [15]. FPN constructs a pyramid structure integrating detailed and semantic features via top-down pathways and lateral connections. Nevertheless, its single information transmission path hinders shallow details from acquiring feedback of deep semantic information. On this basis, PANet [16] adds bottom-up paths and strengthens the transmission of localization details via bidirectional information flow. However, it merely adopts simple linear feature fusion and fails to resolve feature conflicts. BiFPN [17] further introduces a learnable weighted fusion mechanism. Nevertheless, deep semantic features still participate in fusion with relatively large positive weights, leading to small object details being diluted. The newly proposed AFPN [18] adopts a progressive fusion strategy to resolve information conflicts caused by direct fusion. However, shallow feature details gradually decay during layer-wise transmission, resulting in the loss of small object information.

Meanwhile, multi-scale feature fusion networks can be integrated into different parts of the DETR model. For example, Peng et al. [19] employed a hybrid Fourier context encoder and a contribution-balanced feature fusion method, dynamically adjusting the weights of handcrafted features and deep features at different training stages to enhance the fusion effect and model generalization ability. Lu et al. [20] designed a gated linear unit with star-shaped connections for intra-scale feature interaction in the hybrid encoder, and proposed a high-low frequency feature fusion module for cross-scale feature fusion, thereby achieving powerful feature representation. MSO-DETR [21] achieves an effective balance between model lightweighting and detection accuracy on the basis of the RT-DETR framework by enhancing the feature fusion module through scale optimization. MESO-DETR [22] systematically enhances multi-scale features and edge information of the RT-DETR framework by constructing an edge enhancement module. Zhang et al. [23] introduced a context-guided spatial feature reconstruction network into the model's neck, enhancing the model's ability to perform feature fusion across different scales.

2.3. Application of Hybrid Matching

In the DETR-based object detection framework, one-to-one matching [24] is one of its core designs, aiming to achieve end-to-end detection and avoid the use of manually designed NMS. This mechanism uses the Hungarian algorithm to optimally match a fixed number of prediction results with ground truth targets, where each ground truth target is matched to only one prediction box, thereby ensuring the uniqueness of the output. However, since the number of ground truth targets in an image is much smaller than the number of prediction queries, the vast majority of queries are assigned to the “no-object” category, resulting in sparse positive samples, low training efficiency, and slow convergence.

In contrast, one-to-many matching [25] is an important improvement direction proposed to address issues such as sparse positive samples and slow training convergence caused by one-to-one matching in DETR series models. Its core idea is to allow one ground truth object to be associated with multiple prediction queries during training, thereby providing richer supervision signals and accelerating model learning. For example, FeatAug-DETR [26] proposed a method to enrich the one-to-many matching mechanism of DETR series models through feature augmentation, aiming to address the problems of slow convergence and insufficient positive sample supervision during the training of DETR. Meta-DETR [27] proposed integrating a correlation aggregation module into the encoder to address the issue of inter-class correlation neglect. Wang et al. [28] proposed the use of hybrid matching to optimize pseudo-label generation, which effectively improves the efficiency and robustness of cross-domain knowledge transfer.

In summary, compared with existing solutions, the method proposed in this paper is based on the DETR series models and does not rely on anchor boxes. To address the aforementioned problems of small object feature loss and being overwhelmed by deep semantic features, this paper adopts a dynamic competitive feature fusion method, which cooperates with small object bias mechanism to retain more small object features. Supplemented by a hybrid matching strategy, the proposed method improves model accuracy while maintaining inference efficiency.

3. Method

3.1. Network Overview

The baseline DETR series models first adopt a Convolutional Neural Network (CNN) as the backbone network to extract image features, and further enhance these features through the Transformer encoder. Subsequently, a set of learnable object queries interact with these features in the Transformer decoder, and finally, each query independently outputs a prediction. During training, bipartite graph matching uniquely assigns each prediction to its corresponding ground truth target. This enables direct loss computation and eliminates the reliance on anchor boxes as well as Non-Maximum Suppression (NMS) post-processing adopted in conventional detection methods.

Structurally, Deformable DETR also employs a CNN as the backbone network to extract multi-scale feature maps of images. It abandons the computationally intensive global attention mechanism adopted in standard DETR and introduces deformable attention within the Transformer architecture. This mechanism allows each query or feature point to interact no longer with all spatial positions but only with a sparse set of adaptive key sampling points, thereby reducing the computational complexity from quadratic to linear order. The sparse sampling significantly accelerates training convergence.

In this paper, the IDMSNet model architecture consists of basic components including the backbone, Transformer encoder and Transformer decoder. Through the design and integration of a multi-scale feature fusion structure with a small target bias learning mechanism

between the backbone network and the encoder, dynamic competitive fusion is adopted to project all features into a unified dimension, and the obtained fused features are fed into the encoder for subsequent processing. Furthermore, embeddings of dual-branch queries are generated as decoder input, and a hybrid matching strategy is introduced simultaneously. The multi-layer query features output from the decoder are then transmitted to the prediction head for subsequent classification and bounding box prediction.

The IDMSNet model structure is illustrated as Figure 1.

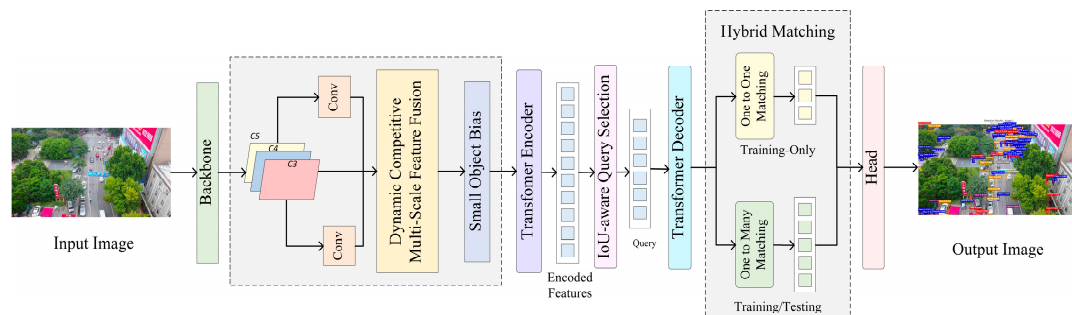


Figure 1. Model Architecture. The network adopts basic components including encoder and decoder, with ResNet50 as the backbone network. The Dynamic Competitive Multi-Scale Feature Fusion and the Small Object Bias are introduced into the model architecture, and Hybrid Matching is adopted in the decoder.

3.2. Dynamic Competitive Multi-Scale Feature Fusion

To address the aforementioned issue that small target feature information in UAV images is easily diluted and lost during conventional multi-scale feature fusion, this paper designs and embeds a dynamic competitive multi-scale feature fusion structure between the backbone network and the encoder.

In contrast, this paper proposes to generate adaptive attention weights through a dual-path parallel attention mechanism and a dynamic competitive fusion strategy, thereby ensuring that the fine-grained features of small targets are fully preserved during the fusion process. This structure enhances the model's robustness to scale-variant targets under complex backgrounds, and provides feature inputs with richer information and stronger representation capability for the subsequent encoder and detection head.

The overall structure of the proposed multi-scale feature fusion module is illustrated in Figure 2.

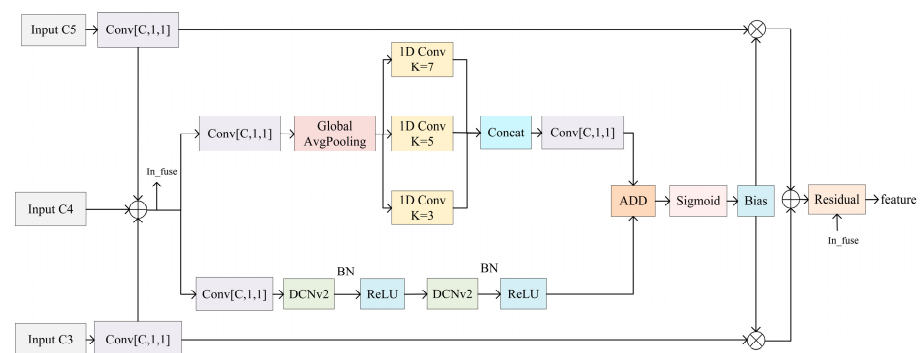


Figure 2. The Structure of Dynamic Competitive Multi-Scale Feature Fusion. It realizes multi-scale feature fusion through a dual-path parallel attention mechanism.

Firstly, the backbone network outputs C3, C4, and C5 features, which serve as the input to the multi-scale feature fusion structure. After feature alignment, preliminary

feature concatenation is performed to obtain initial fused features, which are then fed into the dual-path parallel attention branch. The spatial attention path employs deformable convolution to dynamically learn the spatial offsets of feature maps. The core idea of deformable convolution is to add learnable spatial offsets to each sampling point, enabling the convolution kernel to adaptively adjust the position of its receptive field based on the input features.

The calculation formula is presented as follows:

$$\text{DCNv2}(F) = \sum_{k=1}^K w_k \cdot F(p_k + \Delta p_k) \cdot \Delta m_k \quad (1)$$

where p_k denotes the sampling position of standard convolution, Δp_k is the learnable offset, Δm_k represents the modulation scalar, w_k is the static weight parameter, F stands for the input feature map, k denotes the sampling points of the convolution kernel, and $F(p_k + \Delta p_k)$ is the sampling from the offset position of the input feature map.

The calculation formula for the spatial attention path is given as follows:

$$C = \text{Conv}_{1 \times 1}(f) \quad (2)$$

$$Q = \delta(\text{B}(\text{DCNv2}(C))) \quad (3)$$

$$L(f) = \delta(\text{B}(\text{DCNv2}(Q))) \quad (4)$$

where f denotes the concatenated feature, $\text{Conv}_{1 \times 1}$ stands for 1×1 convolution, δ is the ReLU activation function, B denotes batch normalization, DCNv2 represents deformable convolution, Q denotes the output after the first layer processing, and $L(f)$ is the final output spatial attention weight. The spatial attention weight generated by the left channel is shown in Figure 3.

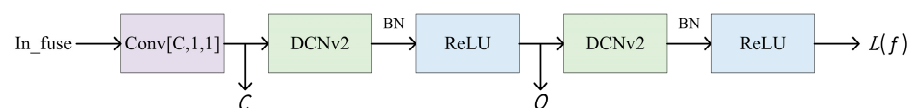


Figure 3. Left channel workflow. The output parameters in the formula are marked in the figure.

The parallel right path corresponds to the channel attention path. It performs multi-scale analysis on the features after global average pooling through three parallel groups of 1D convolutions. Among these three parallel 1D convolutions, the large kernel size is 7, the medium one is 5, and the small one is 3. The large kernel focuses on global channel interaction, making it suitable for extracting large-target features, the medium kernel balances local and global information, which is applicable to medium-scale targets, the small kernel concentrates on fine-grained channel correlation, enhancing feature sensitivity to small targets. Finally, the three groups of features are concatenated and fused through convolution to generate channel attention weights.

The calculation formula for the channel attention path is provided as follows:

$$G(f) = \text{GAP}(\delta(\text{B}(\text{Conv}_{1 \times 1}(f)))) \quad (5)$$

$$R(f) = \text{Conv}_{1 \times 1}(1\text{DConv}_{k=7}(G) \oplus 1\text{DConv}_{k=5}(G) \oplus 1\text{DConv}_{k=3}(G)) \quad (6)$$

where 1DConv denotes the 1D convolution with a kernel size of k , which slides along the channel dimension. The three convolution kernels of different scales capture channel dependencies within different ranges respectively. GAP refers to Global Average Pooling. G denotes the feature processing applied before the input to the three convolutional layers. $\oplus$ represents concatenation along the channel dimension (connecting the outputs of the

three 1D convolutions), and $R(f)$ is the channel attention weights. The right channel generates channel attention weights, as shown in Figure 4.

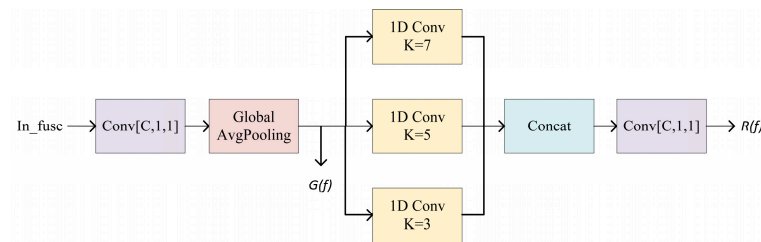


Figure 4. Right channel workflow. The output parameters in the formula are marked in the figure.

After the attention weights of the two paths are output, dynamic fusion is performed, and the adaptive fusion weight M is generated through the Sigmoid function. The formula is as follows:

$$M = \text{Sigmoid}(L(f) + R(f)) \quad (7)$$

The competitive weight S is obtained by summing the adaptive fusion weight M and the bias weight W derived from the small object bias mechanism described below. The specific parameter configurations are provided in the subsequent section. In regions where the value exceeds a predefined threshold, the calculation formula is as follows:

$$S = \text{clamp}(M + W, 0, 1) \quad (8)$$

Clamp denotes the operation that confines the weight S to the range of $[0, 1]$. Using the competitive weight S , the channel-aligned C3 and C5 features are respectively weighted to obtain the post-competition features, and the calculation formula is as follows:

$$Z = S \otimes X \oplus (1 - S) \otimes Y \quad (9)$$

where X denotes the C3 feature, Y denotes the C5 feature, and Z denotes the fused feature. $\otimes$ denotes element-wise multiplication of the corresponding elements.

Unlike existing methods, the competitive weight in this paper is constrained between 0 and 1, forming a seesaw competitive relationship between C3 and C5 features. When the model identifies that the current region contains small targets, the competitive weight approaches 1 so that output features lean more toward shallow features carrying small target information. The fused features contain more shallow feature information and less deep semantic information. Thus, more small target features are retained and the dilution of small target features by deep semantics is avoided.

Finally, the fused feature is residual-connected with the initially fused feature to enhance feature representation, thereby addressing the problem that small object feature information is prone to being overwhelmed and lost in UAV image detection.

After the competitive fusion is implemented, the fused features are projected into the hidden layer of the Transformer to unify the dimensions, and then input into the Transformer encoder together with the positional encoding for global encoding. Next, the globally encoded features and two sets of query embeddings are input into the Transformer decoder, which interacts with the output features of the encoder through a cross-attention mechanism to generate query features and intermediate reference bounding boxes. These are then fed into the prediction head to generate class probabilities and bounding boxes.

3.3. Small Object Bias

Based on the above dynamic competitive multi-scale feature fusion structure, this paper innovatively introduces a small object bias mechanism to enhance the detection performance of the model for small objects. It generates additional biases for small target regions through dynamic learning. Thereby, the differential enhancement of small targets is realized during feature fusion, and the detection accuracy of small targets is improved.

A lightweight bias learning mechanism is constructed in the multi-scale feature fusion structure. After extracting the spatial response map from the high-resolution feature map C3, it is sent to the bias structure for in-depth processing. The bias learning mechanism is illustrated in Figure 5a. The spatial response map of the input C3 features is fed into the depth processing structure. After passing through convolutional layers and activation functions, the features are multiplied by a preset bias factor to obtain the small target bias weight.

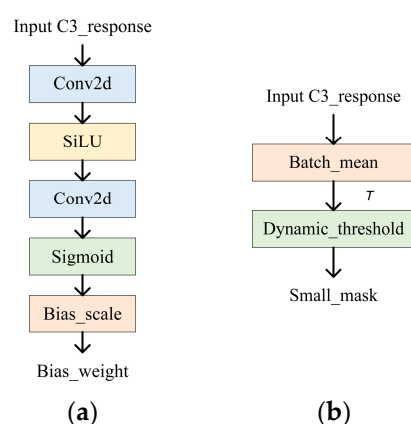


Figure 5. Figure (a) shows the structure of bias weight generation, where bias weight values are obtained through in-depth feature processing. Figure (b) presents the structure of the dynamic threshold algorithm for locating small target regions.

The formula for calculating the bias weight is defined as follows:

$$W = \beta \cdot \text{Sigmoid}(\text{Conv}_{3 \times 3}(\text{SiLU}(\text{Conv}_{3 \times 3}(r)))) \quad (10)$$

where r represents the input C3 response map, $\text{Conv}_{3 \times 3}$ denotes the two-dimensional convolution, β is the preset bias factor, and W is the obtained bias weight.

Meanwhile, the module generates a small object region mask via a dynamic threshold algorithm to ensure the specificity of regions where biases are applied, avoiding interference with large object regions. The dynamic threshold algorithm is illustrated in Figure 5b. First, the C3 response map fused with high-resolution and low-resolution features is imported, and the mean value of samples within the batch is calculated. To alleviate the statistical fluctuation across different batches, a cross-batch moving average is adopted to smooth the threshold generation process. Then, the average value is multiplied by the set threshold coefficient to obtain the dynamic threshold.

The formula for the dynamic threshold is given as follows:

$$D = T \times \text{Mean}(i) \quad (11)$$

where i denotes the fused response map of the current input sample, Mean represents the mean operation on the sample response, T is the threshold coefficient, and D is the resulting dynamic threshold.

Combined with the calculated dynamic threshold, regions with values higher than the threshold are more likely to contain small targets. Regions with response values higher than the threshold are defined as small target areas, and a corresponding mask is generated for these regions. The small target bias weights take effect in these regions. In regions where the value exceeds the threshold, the small object bias weight W is superimposed onto the aforementioned adaptive fusion weight M , yielding the bias-enhanced final fusion weight S .

This design ensures that the model enhances small object detection capabilities while retaining its original feature fusion performance. From the perspective of information theory, this mechanism increases the information weight of small object regions in feature fusion, compensates for their information attenuation in deep features, and achieves information entropy balance among multi-scale features. Compared with the global weights adopted in existing methods, the learnable dynamic bias weights can enhance the feature discrimination of small targets while maintaining computational efficiency, thereby improving the detection accuracy of small targets.

3.4. Hybrid Matching

The model introduces a dual-branch hybrid matching strategy, consisting of the original one-to-one matching branch and an additional one-to-many auxiliary matching branch. During training, an auxiliary branch for one-to-many matching is added to assign multiple prediction queries to each ground truth object, thereby providing richer supervision signals to improve model performance. The division of functions between the two branches is illustrated in Figure 6.

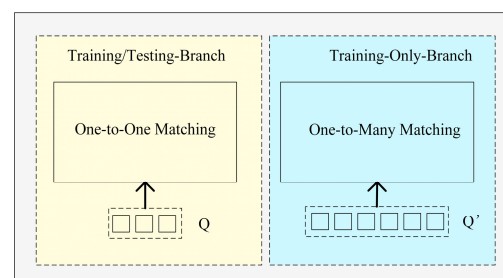


Figure 6. Hybrid branch structure. Yellow denotes the one-to-one branch, and blue denotes the one-to-many branch. The one-to-many branch is only used for training, while the one-to-one branch is applied to both training and inference.

One-to-one matching is used for end-to-end detection. DETR introduces a one-to-one matching strategy based on bipartite graph matching. The core goal of this strategy is to assign a unique prediction to each ground truth object, while all other predictions are forced to be classified as background, thereby theoretically eliminating prediction redundancy completely and abandoning NMS.

The one-to-one matching process first formulates detection as a set matching problem: the model outputs a prediction set $P = \{p_1, p_2, \dots, p_N\}$, with a fixed size of N , and the ground truth set $\hat{G} = \{\hat{g}_1, \hat{g}_2, \dots, \hat{g}_N\}$, is padded to the same size. The optimal matching correspondence is found by minimizing the matching cost, with the mathematical formula as follows:

$$\hat{\sigma} = \operatorname{argmin}_{\sigma \in \Theta_N} \sum_{i=1}^N C_{\text{match}}(\hat{g}_i, p_{\sigma(i)}) \quad (12)$$

Among them, the cost function C_{match} of a single matching pair is composed of both classification cost and localization cost, with the formula as follows:

$$C_{\text{match}}(\hat{g}_i, p_j) = 1_{\{c_i \neq \emptyset\}} \cdot \left[-\lambda_{cls} \cdot s_j(c_i) + \lambda_{L1} \cdot \|b_j - b_i^{gt}\|_1 + \lambda_{giou} \cdot \mathcal{L}_{GIoU}(b_j, b_i^{gt}) \right] \quad (13)$$

where $1_{\{c_i \neq \emptyset\}}$ denotes the indicator function, c_i represents the class label of the i -th ground truth object, $s_j(c_i)$ is the prediction probability of the model for the predicted bounding box p_j on the ground truth class c_i , b_j denotes the bounding box coordinates of the j -th predicted box, b_i^{gt} is the bounding box coordinates of the i -th ground truth object, $\mathcal{L}_{GIoU}(b_j, b_i^{gt})$ is the loss function for measuring the overlap between the two boxes.

The core idea of the one-to-many matching branch is to replicate the ground truth annotations, enabling each ground truth object to be jointly learned by multiple prediction queries, thereby enhancing the training signals, optimizing the utilization of model resources, and maintaining the recall rate.

One-to-many matching constructs an auxiliary supervision signal with a high recall rate by enabling each ground truth object to correspond to multiple prediction queries, so as to optimize model resource utilization. The calculation formula is as follows:

$$\sigma^* = \operatorname{argmin}_{\sigma} \sum C_{\text{match}}(y_i^k, p_{\sigma(i,k)}) \quad (14)$$

where K denotes the number of copies; i represents the ground truth object, y_i^k is the k -th copy of the i -th object. This approach significantly increases the proportion of positive samples. $p_{\sigma(i,k)}$ denotes the prediction query assigned to y_i^k according to the matching strategy, C_{match} is the cost function that measures the difference or matching degree between the ground truth object and the prediction query. $\operatorname{argmin}_{\sigma}$ indicates finding the optimal matching strategy that minimizes the total cost among all possible matching strategies σ , where σ^* is the optimal matching strategy.

In the implementation of one-to-many matching, the original annotated data is first copied K times to form an extended ground truth annotation set. During training, these copied annotations undergo global optimal matching with the prediction results output by the model, and the Hungarian algorithm is used to assign the most suitable predicted box to each copied annotation. After successful matching, the model calculates both classification loss and localization loss simultaneously, thereby providing richer supervision signals for each ground truth object. In this paper, following the configurations of existing schemes, the original annotated data is copied 6 times, and the loss weight of the one-to-many branch is set to 1, contributing equally to the overall loss as the one-to-one branch. The two branches share the decoder parameters to maintain consistency during training.

Following the aforementioned dynamic multi-scale feature fusion, the sample quality is improved. During training, the hybrid matching strategy with parallel dual branches is able to match more positive samples, thereby providing richer supervision signals for training. In contrast, only the one-to-one matching branch is employed during inference. This hybrid matching strategy introduces no additional efficiency overhead and maintains the original model efficiency. It contributes to improving detection accuracy while maintaining the model's detection efficiency. It enables the matching of more high-quality positive samples, allowing the model to learn richer feature information, including features of small objects. This strategy achieves considerable accuracy gain with modest training overhead, demonstrating excellent cost-effectiveness.

4. Experiment

4.1. Dataset Introduction

This paper conducts experiments to verify IDMSNet. It adopts the VisDrone2019 [29] dataset, a large-scale public benchmark tailored for UAV vision scenarios. It is widely used in the research of computer vision tasks such as object detection, multi-object tracking, and crowd counting. This dataset includes 10 common traffic object categories: pedestrians, people, bicycles, cars, vans, trucks, tricycles, awning-tricycles, buses, and motors. It truly

reflects the visual challenges under complex aerial perspectives. The dataset is divided into a training set, a validation set, and a test set. Among them, 6471 annotated images are used for training, 548 for validation, and 1610 for testing. All images are provided with precise bounding box annotations and category information.

We conduct a visual analysis of the object size distribution in the VisDrone2019 dataset, as illustrated in Figure 7. The object distribution is mainly concentrated in the lower-left region, indicating that most objects in the VisDrone2019 dataset are small targets. This highlights the necessity of designing a dedicated small object detection mechanism.

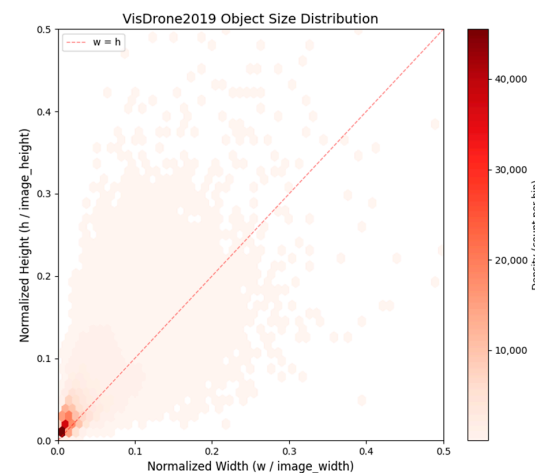


Figure 7. Object size distribution in the VisDrone2019 Dataset. It reflects the large proportion of small objects in the dataset.

4.2. Experimental Configuration

The environmental settings of this experiment are detailed in Table 1. The hardware configuration includes a Windows 10 operating system, an Intel(R) Core(TM) i9-10900X CPU (3.70 GHz), and an NVIDIA GeForce RTX 3090 24G GPU. The software environment is configured via Anaconda 3, using Python 3.8.20 as the programming language, PyTorch 2.4.1 as the deep learning framework, and CUDA 12.1.

Table 1. Configuration of Experiments.

Parameter	Set Up
CPU	Intel(R) Core(TM) i9-10900X CPU@3.70GHz
GPU	NVIDIA GeForce RTX 3090 24G
Framework	PyTorch 2.4.1
Integrated development environment	Python 3.8.20 CUDA12.1
Operating system	Windows 10

4.3. Evaluation Metrics

This paper evaluates the object detector using Precision (P), Recall (R), Average Precision (AP), and mean Average Precision (mAP). mAP is derived from the precision-recall curve and represents the average value of precision across all categories. Two mAP metrics are used: mAP@0.5:0.95 (calculated when the Intersection over Union (IoU) threshold ranges from 0.5 to 0.95) and mAP@0.5 (calculated when the IoU threshold is 0.5).

Precision (P) reflects the proportion of truly correct results among all predictions classified as positive by the model. TP (True Positive) refers to the number of samples correctly predicted as positive by the model. FP (False Positive) denotes the number of samples incorrectly predicted as positive by the model.

$$P = TP / (TP + FP) \quad (15)$$

Recall (R) reflects the proportion of all actual positive samples that are correctly identified by the model. FN (False Negative) denotes the number of positive samples incorrectly missed by the model.

$$R = TP / (TP + FN) \quad (16)$$

Average Precision (AP) is derived from the function of the precision-recall curve, with the formula as follows. Where r denotes recall, and $p(r)$ represents the precision corresponding to the recall.

$$AP = \int_0^1 p(r) dr \quad (17)$$

Mean Average Precision (mAP) is the core metric for evaluating the overall performance of multi-class detection models in the field of object detection. Where k denotes the total number of categories in the dataset, and AP_i represents the Average Precision of the i -th category.

$$mAP = \frac{1}{k} \sum_{i=1}^k AP_i \quad (18)$$

The above indicators comprehensively evaluate model performance from multiple dimensions, ensuring the objectivity and reliability of experimental results.

5. Results

5.1. Comparative Analysis of Experimental Results

To evaluate the effectiveness of the proposed method, comparative experiments are conducted on the VisDrone2019 dataset using different models under identical experimental environments and configurations. The evaluated models include Faster R-CNN, YOLOv7, YOLOv8m, YOLOv9m, YOLOv10m, YOLOv11, Efficient DETR, RT-DETR, Deformable DETR, all of which are trained under the same experimental environment.

During the training and validation phase, the weight decay coefficient is set to 0.0001, the batch size is 2 and ResNet50 is adopted as the backbone network. The training process uses an initial learning rate of 0.0001 with a total of 200 epochs. The parameter settings are shown in Table 2.

Table 2. Parameter Settings.

Parameter	Size
Epoch	200
Lr	1×10^{-4}
Backbone lr	2×10^{-5}
Batch	2
Weight decay	1×10^{-4}

Under identical experimental parameter configurations, the comparative experimental results of all evaluated models on the validation set of the VisDrone2019 dataset are presented in Table 3. Compared with existing methods, the proposed method improves the detection accuracy of UAV images.

As shown in Table 3, the proposed model achieves superior mAP performance in comparative experiments. Specifically, it reaches an mAP@0.5:0.95 of 28.3% and an mAP@0.5 of 43.8%. Compared with conventional anchor-based detectors, most DETR-series models achieve better detection accuracy for UAV images, with higher mAP than traditional one-stage and two-stage detectors. Building upon this foundation, the proposed model achieves further performance gains. Compared with Efficient DETR, our method improves mAP@0.5:0.95 by 5.6% and mAP@0.5 by 7.4%. Compared with RT-DETR, the proposed

model increases mAP@0.5:0.95 by 0.8% and mAP@0.5 by 0.7%. Compared with the baseline Deformable DETR, mAP@0.5:0.95 is improved by 1.2% and mAP@0.5 by 1%. In the final five epochs, the mean mAP@0.5:0.95 approaches the peak value of 28.3 with a variance of 0.004, and the results consistently exceed those of the baseline model, demonstrating the stability and effectiveness of our improvement. The YOLO series algorithms have an advantage in speed, whereas Deformable DETR is primarily optimized for accuracy. Compared with the baseline model Deformable DETR, the accuracy is effectively improved with a slight increase in GFLOPs and a comparable FPS, which verifies the effectiveness of the proposed strategy in maintaining inference efficiency. Overall, the proposed model achieves enhanced accuracy and robustness in UAV image detection tasks.

Table 3. Results of Different Models on the VisDrone2019 Validation. Results superior to other models are marked in the table.

Methods	mAP@0.5:0.95	mAP@0.5	FPS	GFLOPs
Faster R-CNN [9]	16.5	33.0	37	118
YOLOv7 [5]	23.0	41.1	160	105
YOLOv8m [6]	25.5	42.1	153	78.7
YOLOv9m [30]	26.1	43.2	89	76.3
YOLOv10m [31]	26.5	43.7	121	126
YOLOv11 [32]	24.9	41.5	151	86.9
Efficient DETR [33]	22.7	36.4	79	159.3
RT-DETR [14]	27.5	43.1	60	136
Deformable DETR [12] (Baseline)	27.1	42.8	19	162
Ours	28.3	43.8	19	167

To further verify the effectiveness of the experimental results and the superiority of the proposed method, we select some of the most competitive models, newer models, and the baseline model from the results for further comparison with our proposed method on the VisDrone2019 test set. The results on the VisDrone2019 test set are shown in the following Table 4:

Table 4. Comparison with the Baseline Model on the VisDrone2019 Test Set.

Methods	mAP@0.5:0.95	mAP@0.5	P	R
YOLOv8m	24.1	40.8	51.7	40.6
YOLOv9m	24.4	41.4	52.7	41.2
YOLOv10m	25.2	41.6	53.4	41.3
YOLOv11	23.6	40.4	50.3	40.1
RT-DETR	26.5	41.4	55.2	41.3
Deformable DETR	25.8	41.3	54.9	41.4
Ours	27.4	42.1	56.0	41.6

As can be seen from Table 4, on the VisDrone2019 test set, compared with the baseline model Deformable DETR, the proposed model achieves improvements of 1.6% in mAP@0.5:0.95, 0.8% in mAP@0.5, 1.1% in Precision, and 0.2% in Recall. Compared with other models, our proposed model achieves improvements across all metrics and obtains the best results. This proves the validity of the experimental results, as well as the effectiveness and advancement of the proposed model in UAV image detection.

Comparing the small object detection results of the proposed model with other methods on the COCO2017 dataset [34] verifies the generalizability of our approach in small target detection tasks.

As can be seen from Table 5, on the COCO2017 dataset, the proposed model achieves an AP@0.5:0.95 of 44.3% and an AP@0.5 of 62.0%. The proposed model achieves a higher AP@0.5:0.95 than other competing methods. Although its AP@0.5 is slightly lower than

that of DN-DETR, it still outperforms the baseline Deformable DETR. As clearly observed in Table 5, the proposed model achieves a notably higher AP for small objects than other comparative methods, with an improvement of 2.5% over the baseline Deformable DETR and 6% over DN-DETR. Meanwhile, it brings only a slight increase in GFLOPs and hardly introduces extra computational overhead. This further validates the effectiveness and generalizability of the proposed model for small object detection, demonstrating its strong generalization capability in small target detection tasks.

Table 5. Comparison of Various Models on the COCO2017 Dataset. Results superior to other models are marked in the table.

Methods	AP@0.5:0.95	AP@0.5	APs	GFLOPs
Faster R-CNN [9]	37.9	58.8	22.4	28
DAB-DETR [35]	38.0	60.3	19.2	194
DN-DETR [13]	43.3	62.4	22.9	193
DN-Deformable DETR [13]	43.4	61.9	24.8	175
Deformable DETR [12]	43.8	61.6	26.4	162
(Baseline)				
Ours	44.3	62.0	28.9	167

5.2. Ablation Experiment

To verify the effectiveness of the method proposed in this paper, ablation experiments are conducted on the VisDrone2019 dataset. These experiments aim to verify the role of each structure in the small target detection of UAV images, and analyses from multiple dimensions are carried out to confirm the rationality and effectiveness of the proposed method.

The experimental results are illustrated in Table 6.

Table 6. Ablation Experimental Results. $\checkmark$ indicates that the structure is included, and $\times$ indicates that it is not included.

Baseline	Dynamic Competitive Multi-Scale Feature Fusion	Small Object Bias	Hybrid Matching	mAP@0.5:0.95	mAP@0.5	P	R	FPS	GFLOPs
$\checkmark$	$\times$	$\times$	$\times$	27.1	42.8	55.1	41.6	19	162
$\checkmark$	$\times$	$\times$	$\checkmark$	27.3	43.0	55.3	42.3	19	162
$\checkmark$	$\checkmark$	$\times$	$\times$	27.9	43.4	55.9	42.0	18	166
$\checkmark$	$\checkmark$	$\checkmark$	$\times$	28.2	43.6	56.2	42.2	18	167
$\checkmark$	$\checkmark$	$\times$	$\checkmark$	28.1	43.5	56.0	42.6	18	166
$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	28.3	43.8	56.4	42.8	19	167

As shown in Table 6, taking Deformable DETR as the baseline model, with only the hybrid matching strategy incorporated, the model achieves improvements in both mAP@0.5:0.95 and mAP@0.5 without increasing inference computational overhead, along with a 0.7% gain in recall. The introduction of the dynamic competitive multi-scale feature fusion structure increases mAP@0.5:0.95 by 0.8% and mAP@0.5 by 0.6%. With the integration of both the dynamic competitive multi-scale feature fusion structure and the hybrid matching strategy, mAP@0.5:0.95 is improved by 1.0% and mAP@0.5 by 0.7%. On the basis of dynamic competitive multi-scale feature fusion, after introducing the small object bias mechanism, mAP@0.5:0.95 is increased by 0.3% and mAP@0.5 by 0.2%. With the further integration of hybrid matching, mAP@0.5:0.95 increases by 0.1% and mAP@0.5 increases by 0.2%. Compared with the baseline model, the complete model improves mAP@0.5:0.95 by 1.2%, mAP@0.5 by 1.0%, precision by 1.3%, and recall by 1.2%.

It can be concluded that all modules and structures proposed in this work can further improve the performance of the model in UAV image detection. The proposed method

achieves higher detection accuracy with basically unchanged model parameters, which verifies the effectiveness of each designed component as well as the proposed approach.

5.3. Visualization Result Analysis

The visualized detection results of the proposed model and the baseline model in UAV image detection are shown in Figure 8.

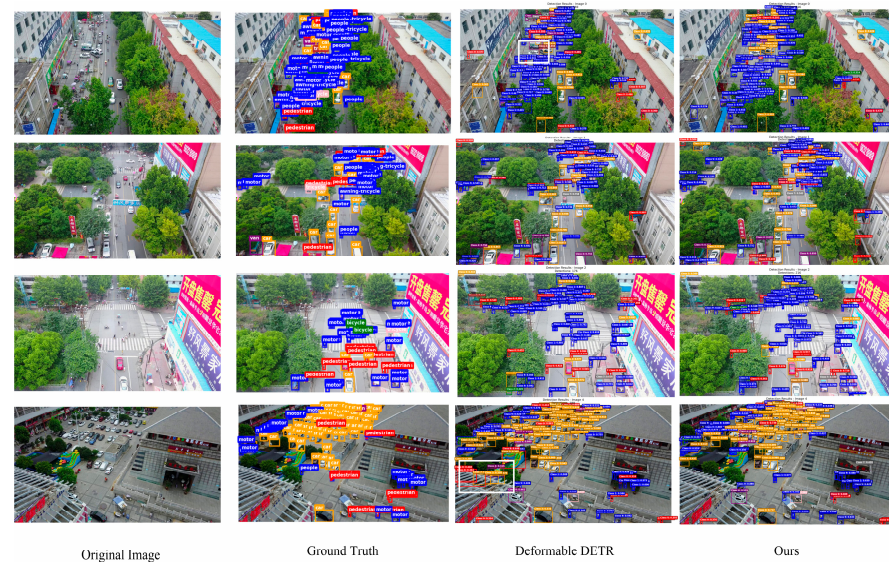


Figure 8. Comparison of Visualization Results. In comparison, the proposed method achieves fewer detection errors.

As can be seen from the comparison figures, although Deformable DETR effectively covers most targets, it still has errors in detecting small objects and some overlapping or occluded targets, which have been marked in Figure 8. For example, in the first figure, Deformable DETR has missed detections when some vehicles or pedestrians are occluded or overlapped. In the fourth figure, there are detection errors in areas occluded by vegetation or buildings. These have all been marked with white boxes in the figure, as shown in Figures 8 and 9.

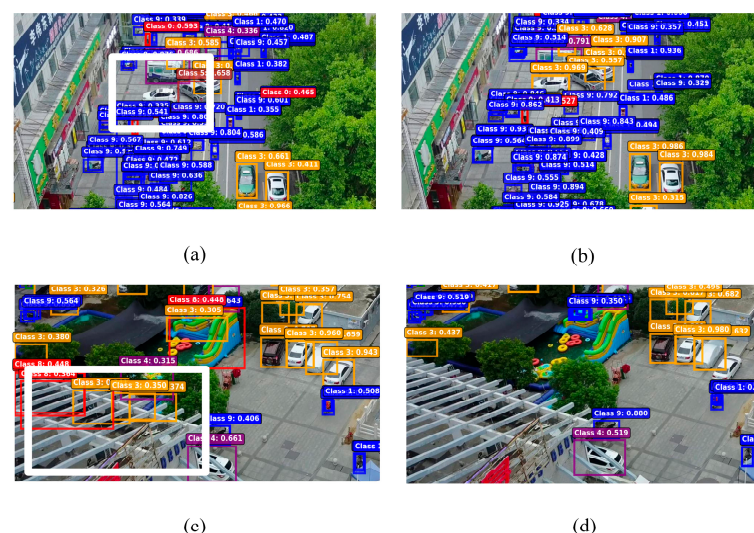


Figure 9. Visual marks. The marked areas in Figure (a) represent missed targets of the baseline model, while the proposed model in Figure (b) has no missed targets. The marked areas in Figure (c) are false detections of the baseline model, and the proposed model in Figure (d) shows no false detections.

It can be seen from Figure 9 that the Deformable DETR suffers from missed detections and false detections, while the proposed model avoids such errors and achieves more accurate detection results. In summary, the detection results of the proposed model are more consistent with the ground truth. It still maintains strong coverage capability in dense small object areas and demonstrates advanced detection performance when dealing with complex scenarios such as dense small objects or vegetation coverage.

6. Conclusions

This paper proposes an innovative competitive fusion model for UAV image object detection, aiming to address the issues of small object feature loss and training efficiency in UAV image detection tasks.

Its key innovations include the design of a dynamic competitive multi-scale feature fusion structure to realize dynamic competitive fusion of multi-scale features. It also develops a small object bias mechanism to enhance small target features on the basis of original feature fusion. Additionally, it introduces a hybrid matching strategy to optimize model resource utilization efficiency. These innovations aim to improve the detection accuracy of UAV images while maintaining computational efficiency. Extensive experiments conducted on the VisDrone2019 dataset demonstrate that the proposed model achieves promising performance in UAV image detection and small object detection tasks, with superior performance across various evaluation metrics. Experimental results on small objects from the COCO2017 dataset verify that the improved method for small target detection in this paper possesses effectiveness and universality. It exhibits excellent generalization ability.

Despite the promising detection performance achieved by our method, certain limitations remain to be addressed. For instance, its generalization capability under adverse weather conditions requires further validation, and additional lightweight optimization is still needed for deployment on edge devices with more restricted computational resources. These limitations also constitute important directions for our future work.

In future research, we will focus on improving the proposed model in terms of lightweight design and theoretical analysis, with specific improvement directions as follows. First, we develop a lightweight backbone network or model architecture to reduce the number of model parameters. Second, we extend the idea of the proposed model from static images to video sequences to realize UAV target detection for consecutive frames [36]. Third, to address the generalization issue under adverse weather conditions, we explore the integration of dehazing methods as a potential preprocessing step with our detection model as a future direction for improvement [37,38]. In addition, we plan to extend our method to other small object detection tasks, such as power line or insulator detection, to further validate its applicability in other UAV vision scenarios. We will continue to conduct research on improving detection accuracy while effectively balancing computational efficiency and real-time performance, so as to enhance the practicality and robustness of the model in complex real-world scenarios.

Author Contributions: Conceptualization, X.K.; Methodology, C.M.; Validation, C.M.; Formal analysis, X.K.; Investigation, C.M.; Resources, X.K.; Data curation, Z.L.; Writing—original draft, C.M.; Writing—review & editing, C.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by [Hubei Province Science and Technology Innovation Talent Program], grant number [2023DJC093].

Data Availability Statement: The dataset used in this study is derived from publicly available datasets. The VisDrone2019 dataset can be accessed at <https://aiskyeye.com/>. The COCO2017 dataset can be accessed at <https://cocodataset.org/>.

Acknowledgments: The authors wish to thank the editor and reviewers for their valuable suggestions.

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

References

1. Tan, L.; Lv, X.; Wang, G.; Lian, X. UAV Image Object Recognition Method Based on Small Sample Learning. *Multimed. Tools Appl.* **2023**, *82*, 26631–26642. [\[CrossRef\]](#)
2. Muzammul, M.; Algarni, A.; Ghadi, Y.; Assam, M. Enhancing UAV Aerial Image Analysis: Integrating Advanced SAHI Techniques with Real-time Detection Models on the VisDrone Dataset. *IEEE Access* **2024**, *12*, 21621–21633. [\[CrossRef\]](#)
3. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 779–788.
4. Fang, Y.; Guo, X.; Chen, K.; Zhou, Z.; Ye, Q. Accurate and Automated Detection of Surface Knots on Sawn Timbers Using YOLO-V5 Model. *Bioresources* **2021**, *16*, 5390–5406. [\[CrossRef\]](#)
5. Wang, C.; Bochkovskiy, A.; Liao, H. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 7464–7475.
6. Reis, D.; Kupec, J.; Hong, J. Real-Time Flying Object Detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972.
7. Ma, C.; Fu, Y.; Wang, D.; Guo, R.; Zhao, X. YOLO-UAV: Object Detection Method of Unmanned Aerial Vehicle Imagery Based on Efficient Multi-Scale Feature Fusion. *IEEE Access* **2023**, *11*, 126857–126878. [\[CrossRef\]](#)
8. Liu, H.; Duan, X.; Chen, H.; Lou, H.; Deng, L. DBF-YOLO: UAV Small Targets Detection Based on Shallow Feature Fusion. *IEEE Trans. Electr. Electron. Eng.* **2023**, *18*, 605–612. [\[CrossRef\]](#)
9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Vaswani, A.; Shazeer, N.; Parmar, N. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems 30: 31st Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Curran Associates: Red Hook, NY, USA, 2017; pp. 5998–6008.
11. Carion, N.; Massa, F.; Synnaeve, G. End-to-End Object Detection with Transformers. In *Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020*; Springer: Cham, Switzerland, 2020; Volume 12346, pp. 213–229.
12. Zhu, X.; Su, W.; Lu, L. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv* **2020**, arXiv:2010.04159.
13. Li, F.; Zhang, H.; Liu, S. DN-DETR: Accelerate DETR Training by Introducing Query Denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022*; IEEE: Piscataway, NJ, USA, 2022; pp. 13619–13627.
14. Zhao, Y.; Lv, W.; Xu, S. 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*; IEEE: Piscataway, NJ, USA, 2024; pp. 16965–16974.
15. Lin, T.; Dollár, P.; Girshick, R. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 2117–2125.
16. Liu, S.; Qi, L.; Qin, H. Path Aggregation Network for Instance Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018*; IEEE: Piscataway, NJ, USA, 2018; pp. 8759–8768.
17. Tan, M.; Pang, R.; Le, Q. EfficientDet: Scalable and Efficient Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020*; IEEE: Piscataway, NJ, USA, 2020; pp. 10781–10790.
18. Yang, G.; Lei, J.; Zhu, Z. AFPN: Asymptotic Feature Pyramid Network for Object Detection. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, HI, USA, 1–4 October 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 2187–2192.
19. Peng, Y.; Tao, Z.; Zhang, J. KDFF-DETR: Knowledge-Driven Feature Fusion DETR for Spaceborne Infrared Ship Detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2025**, *18*, 17638–17653. [\[CrossRef\]](#)
20. Lu, X.; Zhan, G.; Wu, W.; Zhang, W.; Wu, X.; Han, C. Van-DETR: A Lightweight Real-Time Object Detection Model with Enhanced Feature Fusion. *Vis. Comput.* **2025**, *41*, 4221–4238. [\[CrossRef\]](#)
21. Li, J.; Hua, Y.; Xue, M. MSO-DETR: A Lightweight Detection Transformer Model for Small Object Detection in Maritime Search and Rescue. *Electronics* **2025**, *14*, 2327. [\[CrossRef\]](#)
22. Zhou, S.; Cai, Y.; Zhang, Z.; Yin, J. MES-DETR: An Improved RT-DETR Algorithm for Steel Surface Defect Detection. *Electronics* **2025**, *14*, 2232. [\[CrossRef\]](#)
23. Zhang, Q.; Zhang, S.; Qu, C.; Guo, X. A Key Points Recognition and Deformation Correction Method for a Pointer Instrument Based on an Improved RT-DETR Model. *Meas. Sci. Technol.* **2025**, *36*, 095417. [\[CrossRef\]](#)

24. Wang, Y.; Zhang, X.; Yang, T.; Sun, J. Anchor DETR: Query Design for Transformer-Based Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 22 February–1 March 2022*; AAAI Press: Washington, DC, USA, 2022; pp. 2567–2575.
25. Jia, D.; Yu, H.; Lin, W. DETRs with Hybrid Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–14.
26. Fang, R.; Gao, P.; Zhou, A.; Cai, Y.; Liu, S.; Dai, J. FeatAug-DETR: Enriching One-to-Many Matching for DETRs with Feature Augmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 6402–6415. [\[CrossRef\]](#)
27. Zhang, G.; Luo, Z.; Cui, K.; Lu, S.; Xing, E. Meta-DETR: Image-Level Few-Shot Detection with Inter-Class Correlation Exploitation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12831–12843. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Wang, X.; Suo, J.; Li, Y.; Gao, M.; Jiang, P.; Dai, P. Hybrid Matching Teacher Framework for Cross-Domain Visual Detection Transformer. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 13538–13551. [\[CrossRef\]](#)
29. Du, D.; Zhu, P.; Wen, L. VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27 October–2 November 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 1569–1578.
30. Wang, C.; Yeh, I.; Liao, H. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
31. Wang, A.; Chen, H.; Liu, L. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
32. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725.
33. Yao, Z.; Ai, J.; Li, B.; Zhang, C. Efficient DETR: Improving End-to-End Object Detector with Dense Prior. *arXiv* **2021**, arXiv:2104.01318.
34. Lin, T.; Maire, M.; Belongie, S. Microsoft COCO: Common Objects in Context. In *Proceedings of the 13th European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2014; pp. 740–755.
35. Liu, S.; Li, F.; Zhang, H. DAB-DETR: Dynamic Anchor Boxes Are Better Queries for DETR. *arXiv* **2022**, arXiv:2201.12329.
36. Keawboontan, T.; Thammawichai, M. Toward Real-Time UAV Multi-Target Tracking Using Joint Detection and Tracking. *IEEE Access* **2023**, *11*, 65238–65254. [\[CrossRef\]](#)
37. Zhang, S.; Zhang, X.; Shen, L.; Wan, S.; Ren, W. Wavelet-Based Physically Guided Normalization Network for Real-Time Traffic Dehazing. *Pattern Recognit.* **2026**, *172*, 112451. [\[CrossRef\]](#)
38. Liu, Y.; Yan, Z.; Chen, S.; Ye, T.; Ren, W.; Chen, E. NightHazeFormer: Single Nighttime Haze Removal Using Prior Query Transformer. *arXiv* **2023**, arXiv:2305.09533. [\[CrossRef\]](#)

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