

DDS-YOLO: A Detail-Enhanced and Dynamic Scale-Adaptive Network for UAV Small Object Detection

Qiang Liu

qiangliu17@126.com

Shandong University of Aeronautics

Chen Yuanze

Shandong University of Aeronautics

Niu Dexue

Shandong University of Aeronautics

Article

Keywords: UAV imagery, small object detection, detail enhancement, dynamic scale adaptation, multi-scale feature fusion

Posted Date: October 16th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11164472/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

DDS-YOLO: A Detail-Enhanced and Dynamic Scale-Adaptive Network for UAV Small Object Detection

Qiang Liu^{1*}, Chen Yuanze, Niu Dexue

1. Aerospace and Mechanical College, Shandong University of Aeronautics, 256600 Shandong, China

Qiang Liu(Corresponding author, Email: qiangliu17@126.com)

Abstract:

Small objects in unmanned aerial vehicle (UAV) imagery contain few pixels and exhibit large scale variation, dense distributions, and complex backgrounds, which weaken fine-grained representation and cross-scale fusion. This paper proposes DDS-YOLO. A Detail Enhancement Strategy (DES) integrates local structure, contextual, and high-frequency residual branches with channel-wise weighting and spatial selection to enhance boundaries and textures while suppressing background responses. A Dynamic Scale Adaptation Strategy (DSAS) aligns adjacent features and performs channel-wise scale weighting with local correction to adapt fusion to the current object distribution. A Small Object Optimization Strategy (SOOS) adds a lightweight P2 branch for high-resolution prediction, and Wise Intersection over Union version 3 (WIoU v3) improves bounding-box regression. On VisDrone2019, DDS-YOLO achieves 56.0% Precision, 44.5% Recall, 46.3% mean average precision at 0.5 (mAP0.5), and 28.6% mAP0.95, improving YOLOv11n by 9.8, 9.4, 8.9, and 6.7 percentage points, respectively. With 8.7 M parameters, 23.8 G floating-point operations (FLOPs), and 161 frames per second (FPS), the model provides a favorable accuracy–efficiency trade-off for UAV small object detection.

Keywords: UAV imagery; small object detection; detail enhancement; dynamic scale adaptation; multi-scale feature fusion

1 Introduction

1.1 Research Background

UAVs are widely used in environmental monitoring, traffic management, disaster response, and intelligent inspection because of their flexible deployment and broad-area sensing capabilities [1,2]. Object detection provides essential information for localization, scene understanding, and autonomous decision-making in these applications.

Deep learning has become the dominant approach to aerial object detection [3]. The YOLO family is particularly attractive for UAV platforms because it balances accuracy and inference speed; however, limited onboard computation and energy require strict control of model complexity [4,5].

Accurate small-object perception remains difficult because scale variation, background interference, and feature degradation jointly reduce detection stability.

1.2 Research Progress

UAV imagery is acquired from long distances and variable viewpoints, so small objects contain few pixels and weak texture, boundary, and structural cues. These cues are easily degraded during feature extraction, limiting discriminative representation [6,7].

Changes in flight altitude, viewing angle, and object distance create substantial intra-class scale variation, making consistent detection across scales difficult [8,9].

Complex backgrounds, occlusion, and dense distributions further reduce foreground–background contrast and increase false and missed detections [10,11].

Thus, UAV small object detection requires simultaneous preservation of fine details, adaptation to scale variation, and suppression of background interference.

Feature-enhancement methods use attention or dedicated modules to strengthen informative regions [12–14], but their gains are constrained when source details are scarce, and complex structures increase deployment cost.

Multi-scale methods improve cross-level interaction through architectures such as YOLO-MS, BiFPN-YOLO, and YOLO-Pika [15–18]. Most, however, use fixed fusion paths that cannot adapt feature contributions to changing object distributions.

Spatial/channel attention, multimodal fusion, and scene-adaptive learning improve robustness under occlusion and environmental variation [19–25]. Nevertheless, many methods address only one challenge and do not jointly preserve details, adapt scales, and maintain efficiency.

1.3 Research Content and Contributions

To address these limitations, this study proposes DDS-YOLO, which improves fine-grained representation, cross-scale interaction, and extremely small-object prediction within the YOLO framework.

The main contributions of this study are summarized as follows:

- (1) DES strengthens local structures, context, and high-frequency cues to preserve small-object boundaries and textures.
- (2) DSAS dynamically adjusts the contributions of adjacent feature levels to accommodate changing object-scale distributions.
- (3) SOOS introduces a lightweight P2 branch to improve high-resolution prediction of extremely small objects.

2 Related Work

2.1 YOLO-Based Detection Methods

YOLO unifies classification and bounding-box regression in a one-stage framework and is widely used for real-time detection [26–28]. Subsequent variants have improved accuracy, speed, and lightweight deployment.

Backbone and head redesign [29], lightweight architectures [30], and multi-scale fusion, dynamic assignment, and decoupled heads [31] have extended YOLO to complex scenes. However, standard YOLO models still underuse fine details and adapt poorly to the severe scale variation of UAV imagery.

2.2 Small Object Detection Methods

Small objects contain limited pixels, and their textures and boundaries are easily lost in deep networks, degrading classification and localization [32].

Existing methods therefore employ high-resolution branches and contextual or detail enhancement [33], improved assignment and losses for dense objects [34], and aerial-image-specific architectures [35]. These strategies are effective but rarely address detail loss, scale variation, and background interference within one compact framework.

2.3 Feature Fusion and Adaptive Detection Methods

Feature pyramid networks fuse hierarchical features for multi-scale detection, but fixed propagation can cause inefficient information transfer [36]. Improved fusion paths and cross-level interaction partly alleviate this problem [37].

Attention, Transformer, dynamic convolution, and dynamic weighting further support informative-region selection and adaptive representation [38,39]. Their additional complexity and limited local scale sensitivity motivate a lighter, input-dependent fusion mechanism.

3 Method

3.1 Overall Architecture of DDS-YOLO

YOLOv11n is selected as the baseline because of its favorable real-time performance. DDS-YOLO introduces targeted modifications to feature extraction, multi-scale fusion, and prediction to address detail loss, scale variation, and weak extremely small-object responses.

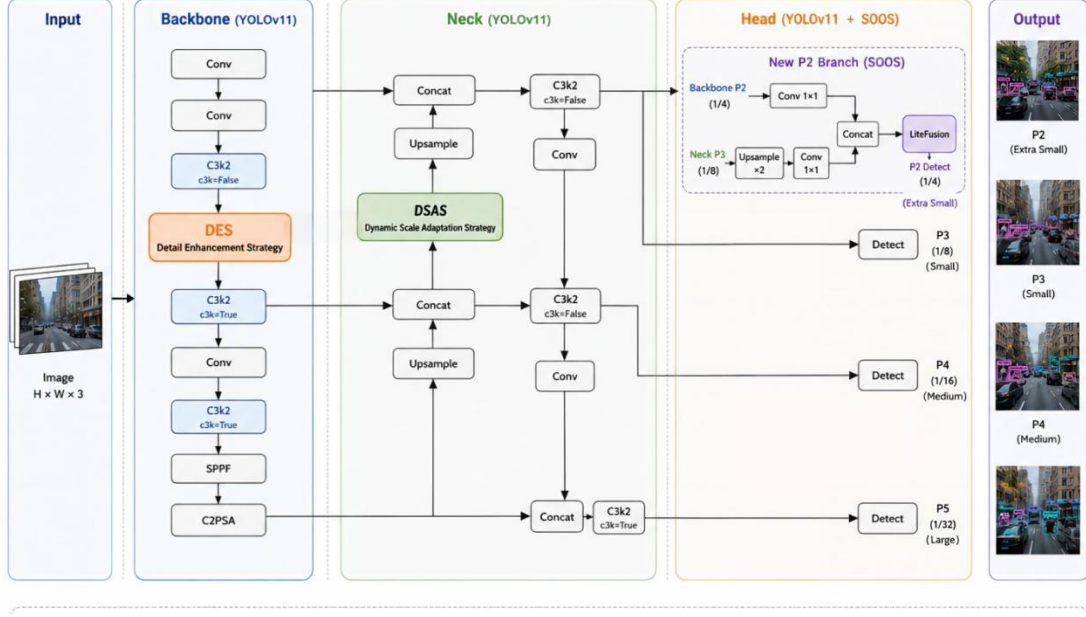


Figure 1. Overall architecture of DDS-YOLO

As shown in Figure 1, DES enhances shallow local details, DSAS dynamically fuses adjacent scales, and SOOS adds a lightweight high-resolution P2 prediction branch. The three strategies act sequentially from feature extraction to fusion and detection.

3.2 DES

3.2.1 Design Motivation

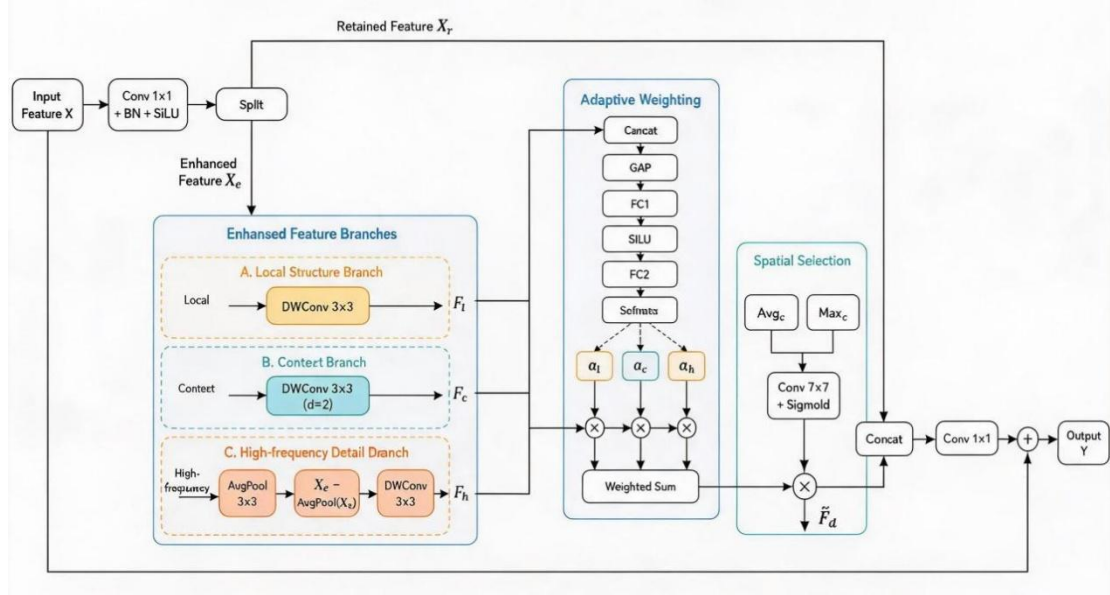


Figure 2. Structure of DES

Small UAV objects are represented mainly by boundaries, contours, and local textures, which are progressively weakened by convolution and downsampling.

Simply propagating more shallow features also transfers background noise. DES is therefore placed at the shallow-to-intermediate transition to preserve a stable branch while enhancing local, contextual, and high-frequency information.

As shown in Figure 2, the enhanced responses are adaptively weighted and spatially selected so that object-related details are strengthened without indiscriminately amplifying the background.

3.2.2 Module Structure

Let the input feature be defined as follows:

$$X \in R^{C \times H \times W} \quad (1)$$

C , H , and W denote channels, height, and width. Conv and BN denote convolution and batch normalization, respectively, while ϕ denotes the sigmoid linear unit (SiLU) activation function.

$$X' = \phi(BN(Conv_{1 \times 1}(X))) \quad (2)$$

The transformed feature X' is then divided along the channel dimension into a retained feature and an enhanced feature:

$$[X_r, X_e] = Split(X') \quad (3)$$

$Split(\cdot)$ partitions X' into the retained feature X_r and enhanced feature X_e .

X_r preserves the original information flow, whereas X_e is processed by local, contextual, and high-frequency branches:

$$F_l = DWConv_{3 \times 3}(X_e) \quad (4)$$

F_l is extracted by a 3×3 depthwise convolution to model local boundaries and textures.

$$F_c = DWConv_{3 \times 3}^{d=2}(X_e) \quad (5)$$

F_c uses a dilation rate of 2 to enlarge the receptive field and incorporate context.

The high-frequency branch subtracts a 3×3 average-pooled response from X_e and then extracts structural residuals:

$$X_h = X_e - AvgPool_{3 \times 3}(X_e) \quad (6)$$

X_h is the high-frequency residual obtained after local smoothing.

$$F_h = DWConv_{3 \times 3}(X_h) \quad (7)$$

F_h retains rapidly varying components such as boundaries, corners, and textures.

3.2.3 Adaptive Detail Selection and Feature Fusion

Because scene content varies, DES generates input-dependent weights for the three branches. Their concatenated output is globally pooled to form descriptor z :

$$z = GAP([F_l, F_c, F_h]) \quad (8)$$

GAP and $[\cdot]$ denote global average pooling and channel concatenation. Two mappings followed by Softmax produce channel-wise weights α_l , α_c , and α_h :

$$[\alpha_l, \alpha_c, \alpha_h] = Softmax(W_2 \delta(W_1 z)) \quad (9)$$

W_1 and W_2 are learnable mappings and δ is the activation function; Softmax enforces Eq. (10).

$$\alpha_l + \alpha_c + \alpha_h = 1 \quad (10)$$

The weighted enhanced feature is expressed as:

$$F_d = \alpha_l \odot F_l + \alpha_c \odot F_c + \alpha_h \odot F_h \quad (11)$$

$\odot$ denotes element-wise multiplication, and F_d is the weighted detail feature.

The weights emphasize local, contextual, or high-frequency cues according to the input. A spatial map is then generated to suppress irrelevant regions:

$$A_s = \sigma(\text{Conv}_{7 \times 7}[\text{Avg}_c(F_d), \text{Max}_c(F_d)]) \quad (12)$$

A_s is obtained from channel-wise average/max pooling, a 7×7 convolution, and Sigmoid.

The detail feature after spatial selection is expressed as:

$$\tilde{F}_d = A_s \odot F_d \quad (13)$$

As reweights F_d at each spatial location; the same symbol is reused for the updated feature.

Finally, the spatially selected detail feature is concatenated with the retained branch. A 1×1 convolution is then used for channel fusion, followed by a residual connection:

$$Y = X + \text{Conv}_{1 \times 1}([X_r, \tilde{F}_d]) \quad (14)$$

Y is the *DES* output; $[X_r, \tilde{F}_d]$ denotes channel concatenation, and X provides the residual path.

The residual connection preserves semantics and gradient flow, while depthwise and pointwise convolutions limit computational overhead.

3.3 DSAS

3.3.1 Design Motivation

YOLOv11 uses fixed top-down and bottom-up fusion, although scale contributions vary with altitude, viewpoint, and object distribution. Fixed concatenation may therefore introduce weakly relevant features and background responses.

DSAS replaces key neck fusion nodes with scale alignment, channel-wise dynamic weighting, local correction, and residual fusion (Figure 3), enabling adjacent levels to contribute according to the current input.

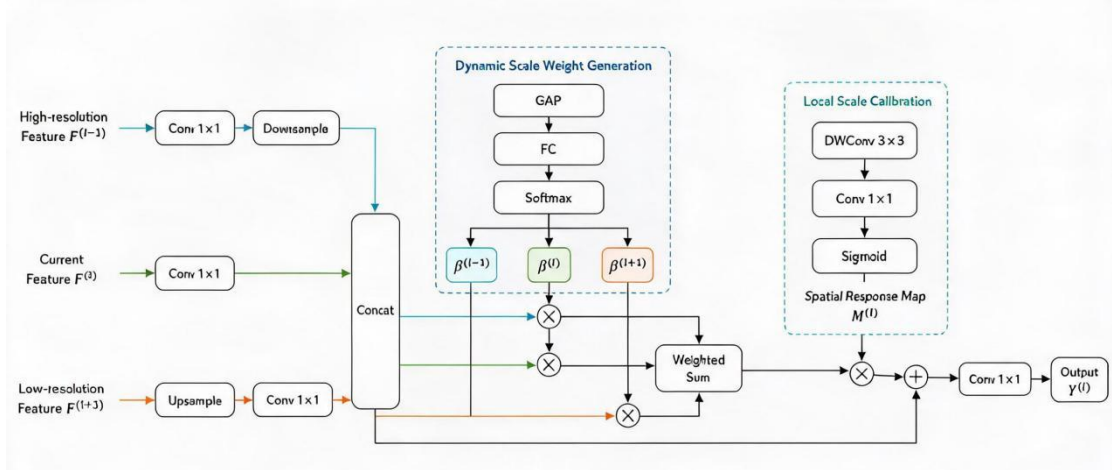


Figure 3. Structure of DSAS.

3.3.2 Multi-Scale Feature Alignment

Assume that the l -th fusion node in the neck receives three adjacent-scale features:

$$F_{l-1} \in R^{C_{l-1} \times 2H_l \times 2W_l} \quad (15)$$

$$F_l \in R^{C_l \times H_l \times W_l} \quad (16)$$

$$F_{l+1} \in R^{C_{l+1} \times \frac{H_l}{2} \times \frac{W_l}{2}} \quad (17)$$

F_{l-1} , F_l , and F_{l+1} denote the adjacent high-, current-, and low-resolution features; C_i , H_i , and W_i denote their channels and the current spatial size.

To enable cross-scale interaction, the three features are aligned to the same spatial resolution and number of channels:

$$\hat{F}_{l-1} = \text{Conv}_{3 \times 3}^{s=2}(\text{Conv}_{1 \times 1}(F_{l-1})) \quad (18)$$

$$\hat{F}_l = \text{Conv}_{1 \times 1}(F_l) \quad (19)$$

$$\hat{F}_{l+1} = \text{Conv}_{1 \times 1}(\text{Up}(F_{l+1})) \quad (20)$$

A 1×1 convolution aligns channels, a stride-2 convolution downsamples the high-resolution feature, and twofold upsampling aligns the low-resolution feature. Aligned features reuse the original symbols.

After alignment, all three features satisfy:

$$\hat{F}_i \in R^{C \times H_l \times W_l}, i \in \{l-1, l, l+1\} \quad (21)$$

C is the unified channel number.

This alignment creates a common representation space while retaining high-resolution details and low-resolution semantics.

3.3.3 Cross-Scale Dynamic Weight Allocation

The three aligned features are concatenated along the channel dimension:

$$F_m = \text{Concat}(\hat{F}_{l-1}, \hat{F}_l, \hat{F}_{l+1}) \quad (22)$$

F_m is the concatenated multi-scale feature.

A global descriptor is extracted to estimate the relative importance of the three scales:

$$z = \text{GAP}(F_m) \quad (23)$$

$$z \in R^{3C} \quad (24)$$

z is the pooled descriptor of dimension $3C$.

Following dimensionality reduction and nonlinear mapping, channel-wise weights are generated for the three scale branches:

$$q = W_2 \delta(W_1 z) \quad (25)$$

$$[\beta_{l-1}, \beta_l, \beta_{l+1}] = \text{Softmax}_s(q) \quad (26)$$

q is the mapped descriptor; *Softmax* along the scale dimension yields channel-wise weights β_{l-1} , β_l , and β_{l+1} .

For any channel c , the three scale weights satisfy:

$$\beta_{l-1}^c + \beta_l^c + \beta_{l+1}^c = 1 \quad (27)$$

The dynamically fused scale feature is expressed as:

$$F_s = \beta_{l-1} \odot \hat{F}_{l-1} + \beta_l \odot \hat{F}_l + \beta_{l+1} \odot \hat{F}_{l+1} \quad (28)$$

F_s is the channel-wise weighted fusion, and $\odot$ denotes element-wise multiplication.

Channel-wise weighting allows high-resolution detail, current-level information, and low-resolution semantics to be emphasized according to scene content.

3.3.4 Local Scale Correction

Global weights cannot capture regional scale imbalance, so DSAS adds a lightweight spatial correction branch:

$$M_l = \sigma[\text{Conv}_{1 \times 1}(\text{DWConv}_{3 \times 3}(F_s))] \quad (29)$$

$$M_l \in R^{1 \times H_l \times W_l} \quad (30)$$

M_l is the spatial response map produced by depthwise convolution, channel compression, and Sigmoid.

The scale feature after local correction is expressed as:

$$\tilde{F}_s = M_l \odot F_s + F_s \quad (31)$$

The residual term preserves low-response regions; F_s is reused for the corrected feature.

The final output is obtained by residually combining the scale-enhanced feature with the current-level feature:

$$Y_l = \hat{F}_l + \text{Conv}_{1 \times 1}(\tilde{F}_s) \quad (32)$$

Y_l is the DSAS output, and the mapped F_s is added to the current-level residual $\hat{F}_l$.

The output retains the current spatial size and directly replaces conventional neck fusion.

DSAS uses pointwise/depthwise convolution and pooling rather than global self-attention, keeping computation suitable for real-time UAV detection.

Overall, scale alignment, dynamic weighting, and local correction jointly preserve localization details and semantic context.

3.4 SOOS

3.4.1 Design Motivation

YOLOv11 predicts at P3–P5 with strides of 8, 16, and 32. Extremely small objects may occupy only one P3 cell after downsampling, causing loss of contour and position.

SOOS therefore adds a lightweight stride-4 P2 branch that combines shallow high-resolution details with neck semantics (Figure 4).

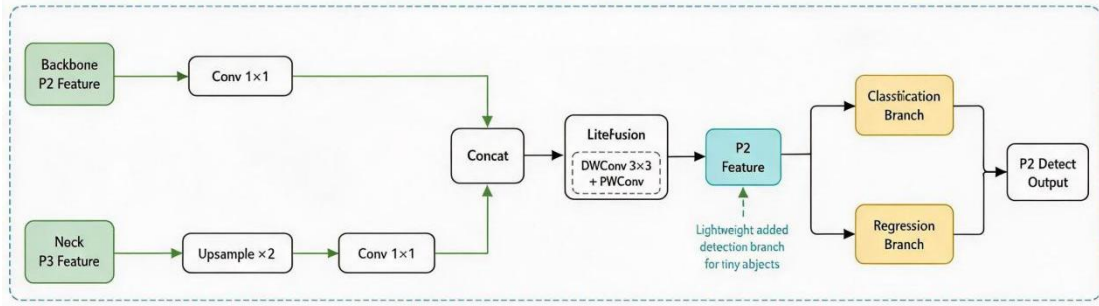


Figure 4. Structure of SOOS

3.4.2 Lightweight P2 Detection Branch

Let the P3-scale feature produced by the neck be defined as:

$$F_3 \in R^{C_3 \times \frac{H}{8} \times \frac{W}{8}} \quad (33)$$

F_3 is the P3 neck feature with C_3 channels and spatial size $H/8 \times W/8$.

The corresponding shallow P2 feature from the backbone is defined as:

$$F_2^b \in R^{C_2 \times \frac{H}{4} \times \frac{W}{4}} \quad (34)$$

F_2^b is the backbone P2 feature with C_2 channels and spatial size $H/4 \times W/4$.

The P3 feature is upsampled by a factor of two and processed by a 1×1 convolution for channel reduction:

$$\hat{F}_3 = \text{Conv}_{1 \times 1}(\text{Up}_2(F_3)) \quad (35)$$

Up_2 denotes twofold upsampling and the 1×1 convolution reduces channels; the aligned feature reuses F_3 .

The shallow P2 feature is also mapped:

$$\hat{F}_2^b = \text{Conv}_{1 \times 1}(F_2^b) \quad (36)$$

The 1×1 convolution maps F_2^b to the fusion channel space; the symbol is reused after mapping.

The two features are concatenated along the channel dimension and processed by a lightweight fusion unit to generate the P2 prediction feature:

$$F_2 = \text{LiteFusion}(\text{Concat}[\hat{F}_3, \hat{F}_2^b]) \quad (37)$$

F_2 is the fused P2 prediction feature; Concat and LiteFusion denote channel concatenation and the lightweight fusion unit.

LiteFusion combines depthwise and pointwise convolutions:

$$\text{LiteFusion}(X) = \text{Conv}_{1 \times 1}(\text{DWConv}_{3 \times 3}(X)) \quad (38)$$

LiteFusion uses depthwise convolution for local spatial modeling and pointwise convolution for channel interaction.

The fusion retains P2 resolution while supplementing shallow detail with P3 semantics.

3.4.3 Lightweight Decoupled Detection Head

The P2 branch uses separate classification and regression paths. To control high-resolution computation, both paths use depthwise separable convolutions.

The classification branch is expressed as:

$$P_2^{cls} = \text{Conv}_{1 \times 1}^{cls}[\phi(\text{PWConv}_{1 \times 1}(\text{DWConv}_{3 \times 3}(F_2)))] \quad (39)$$

The regression branch is expressed as:

$$P_2^{reg} = \text{Conv}_{1 \times 1}^{reg}[\phi(\text{PWConv}_{1 \times 1}(\text{DWConv}_{3 \times 3}(F_2)))] \quad (40)$$

P_2^{cls} and P_2^{reg} are the classification and regression outputs, with independent output layers.

The complete detection output consists of predictions from four scales:

$$P = \{P_2, P_3, P_4, P_5\} \quad (41)$$

P is the four-scale prediction set $\{P_2, P_3, P_4, P_5\}$.

P_2 focuses on extremely small objects, while the original P3–P5 branches remain unchanged.

For an input image of size $H \times W$, let the object width and height be w and h , respectively.

The approximate object dimensions on a feature map with stride s are expressed as:

$$w_s = \frac{w}{s}, h_s = \frac{h}{s} \quad (42)$$

w_s and h_s are the object dimensions on a feature map with stride s .

Reducing stride from 8 to 4 approximately doubles each spatial dimension and quadruples the response area, preserving more positional and contour information.

SOOS therefore improves extremely small-object prediction with limited structural and computational overhead.

3.5 Loss Function

DDS-YOLO retains the YOLOv11 classification loss and Distribution Focal Loss and replaces the bounding-box regression loss with WIoU v3.

WIoU v3 uses dynamic non-monotonic focusing to reduce the influence of very low-quality samples and emphasize samples with greater regression value [40].

This change adds no parameters or computation during inference.

4 Experiments and Results

4.1 Experimental Settings

YOLOv11n is used as the baseline because its lightweight one-stage architecture provides a suitable accuracy–efficiency reference. All comparative and ablation models use the same dataset split, input size, augmentation, training schedule, and evaluation protocol.

Experiments use VisDrone2019-DET, which contains diverse UAV platforms, urban scenes, viewpoints, illumination, occlusion, dense objects, and imbalanced categories [41].

The dataset provides 6,471 training and 548 validation images across ten categories. Because test annotations are unavailable, training uses the official training set and evaluation uses the validation set (Table 1).

Table 1. Basic Information of the VisDrone2019-DET Dataset

Subset	Number of Images	Number of Categories
Training set	6471	10
Validation set	548	10
Test set	1610	10

Images are resized with aspect-ratio preservation and padding. Training uses Mosaic, random scaling and translation, horizontal flipping, and hue-saturation-value (HSV) perturbation;

validation uses only resizing and normalization (Figure 5).



Figure 5. Examples of data augmentation methods applied to a VisDrone2019 image.

Bounding-box size, aspect-ratio, and category distributions are analyzed to characterize the prevalence of small and dense objects.

4.1.1 Implementation Details

Experiments are implemented in PyTorch with YOLOv11n pretrained weights. Models use 640×640 input, 300 epochs, stochastic gradient descent (SGD), an initial learning rate of 0.01, cosine annealing, momentum of 0.937, and weight decay of 0.0005.

Table 2. Experimental Environment and Training Parameters

Item	Configuration
Operating system	Windows 11
Deep learning framework	PyTorch
Central processing unit (CPU)	i7-13700kf
Graphics processing unit (GPU)	4080
Compute Unified Device Architecture (CUDA) version	12.6
Baseline model	YOLOv11n
Pretrained weights	Common Objects in Context (COCO) pretrained weights
Input resolution	640×640
Training epochs	300
Batch size	16
Optimizer	SGD
Initial learning rate	0.01
Momentum	0.937
Weight decay	0.0005
Bounding-box regression loss	WIoU v3

All variants use identical settings, and the best validation checkpoint is used for subsequent analyses (Table 2).

4.1.2 Evaluation Metrics

Detection accuracy is evaluated using Precision, Recall, mAP0.5, and mAP0.95; scale-specific performance is measured using average precision for small, medium, and large objects (APS, APM, and APL), respectively.

Efficiency is reported using the number of parameters (Params), FLOPs, model size, and FPS. FPS is averaged after warm-up under identical hardware, input, and batch settings.

4.2 Comparison with Existing Methods

DDS-YOLO is compared with lightweight YOLO variants trained under identical settings and representative UAV detectors using published VisDrone2019 results.

4.2.1 Comparison with Lightweight YOLO Models

Table 3. Results of Different Lightweight Detectors on the VisDrone2019 Validation Set

Model	Precision	Recall	mAP0.5	mAP0.95
YOLOv5n	40.8	29.7	31.6	17.5
YOLOv8n	43.7	32.4	34.5	19.8
YOLOv10n	44.9	33.6	35.8	20.7
YOLOv11n	46.2	35.1	37.4	21.9
DDS-YOLO	56.0	44.5	46.3	28.6

Table 3 shows that DDS-YOLO achieves 56.0% Precision, 44.5% Recall, 46.3% mAP0.5, and 28.6% mAP0.95, improving YOLOv11n by 9.8, 9.4, 8.9, and 6.7 percentage points, respectively.

The larger Recall and mAP gains indicate better coverage and localization of distant, low-resolution, and dense objects, consistent with the complementary roles of DES, DSAS, and P2 prediction.

Despite the added modules, the model retains 161 FPS and a compact size, meeting real-time requirements.

4.2.2 Comparison with UAV Small Object Detection Methods

Table 4 compares DDS-YOLO with classical and recent UAV small object detectors.

Table 4. Performance Comparison of UAV Small Object Detectors on VisDrone2019

Model	mAP0.95	mAP0.5	AP _S	AP _M	AP _L	Params	FPS
RT-DETR-R18	20.8	37.2	12.7	32.2	42.0	19.89	108
RT-DETR-R50	22.4	39.4	13.7	33.6	45.1	42.12	79
DFA-DETR	23.3	40.5	14.6	35.1	47.2	16.55	113
Drone-YOLO-N	—	38.1	—	—	—	3.1	—
Modified-YOLOv8	—	42.2	—	—	—	19.2	—
PC-YOLO11s	—	43.8	—	—	—	7.1	—
PVswin-YOLOv8s	—	43.3	—	—	—	41.8	—
EdgeYOLO-S	—	44.8	—	—	—	40.5	—
WCDB-YOLO	27.9	45.5	—	—	—	19.8	30.3
DDS-YOLO	28.6	46.3	15.4	39.1	50.4	8.7	161

DDS-YOLO exceeds DFA-DETR by 5.8 points in mAP0.5, 5.3 points in mAP0.95, and 0.8 points in APS; APS reaches 15.4%.

Its APM and APL are 39.1% and 50.4%, showing that small-object optimization does not compromise medium- or large-object detection.

Compared with WCDB-YOLO, DDS-YOLO improves mAP0.5 and mAP0.95 by 0.8 and 0.7 points while reducing Params by 11.1M and FLOPs by 36.3G.

It also improves mAP0.5 by 1.5 points over EdgeYOLO-S using about one-fifth of its parameters and computation, and by 2.5 points over PC-YOLO11s with 1.6 M additional parameters.

4.2.3 Accuracy–Efficiency Analysis

DDS-YOLO uses 8.7 M parameters and 23.8 G FLOPs. Relative to DFA-DETR, these are reduced by 47.4% and 51.7%; compared with WCDB-YOLO, it is faster and reduces parameters and FLOPs by 56.1% and 60.4%.

The gain therefore does not rely on large-scale expansion: DES and DSAS use lightweight convolutions and weight generation, while LiteFusion and the decoupled P2 head control high-resolution cost.

Overall, DDS-YOLO provides a favorable accuracy–efficiency trade-off for real-time UAV small object detection.

4.3 Ablation Studies

Ablation studies use YOLOv11n and the settings in Section 4.1 to assess the contributions of DES, DSAS, SOOS, and WIoU v3.

4.3.1 Ablation Study of the Overall Model

Table 5 reports the progressive integration of the four components.

DES produces the largest single gain, increasing mAP0.5 from 37.4% to 41.8% and APS by 1.9 points, confirming the importance of shallow detail preservation.

DSAS improves mAP0.5 and mAP0.95 by 3.3 and 2.6 points, while SOOS contributes more strongly to Recall and APS through high-resolution prediction.

Table 5. Ablation Results of the Main Components in DDS-YOLO

DES	DSAS	SOOS	WIoU v3	Precision	Recall	mAP0.5	mAP0.95	Params
				46.2	35.1	37.4	21.9	2.6
√				50.8	39.6	41.8	25.1	4.1
	√			49.5	38.7	40.7	24.5	3.5
		√		48.7	39.1	40.2	24.0	4.2
√	√			53.0	41.8	44.0	27.0	5.0
√		√		52.4	42.2	43.8	26.8	5.7
	√	√		51.5	41.4	42.9	26.1	5.1
√	√	√		55.3	43.8	45.6	28.0	8.7
√	√	√	√	56.0	44.5	46.3	28.6	8.7

Combining DES, DSAS, and SOOS yields 45.6% mAP0.5; WIoU v3 further raises it to 46.3% without inference-time overhead, indicating complementary effects.

4.3.2 Structural Ablation of DES

Table 6. Ablation Results of the Internal Components of DES

LS	CB	HFD	AW	SS	mAP0.5	mAP0.95	APs	Params
					37.4	21.9	11.8	2.6
√					38.5	22.8	12.5	2.9
√	√				39.6	23.7	12.9	3.3
√	√	√			40.7	24.4	13.2	3.7
√	√	√	√		41.4	24.9	13.5	4.0
√	√	√	√	√	41.8	25.1	13.7	4.1

DES comprises local structure (LS), context (CB), high-frequency detail (HFD), adaptive

weighting (AW), and spatial selection (SS); Table 6 reports their progressive ablation.

LS improves mAP0.5 by 1.1 points, and adding CB raises it to 39.6%, showing that local boundaries and surrounding context are complementary.

HFD raises APS to 13.2%, while AW adds 0.7 points in mAP0.5 by adapting branch contributions.

SS provides a smaller but consistent gain by suppressing background responses, and the complete DES performs best.

Table 7 further compares DES insertion positions.

Table 7. Results of DES at Different Insertion Positions

Insertion Position	mAP0.5	mAP0.95	APs	Params	FLOPs
After the shallow P2/4 feature	41.3	24.8	13.5	4.0	9.4
P2–P3 transition	41.8	25.1	13.7	4.1	9.8
P3–P4 transition	40.9	24.4	13.1	4.1	8.9
Dual deployment at P2 and P3	42.0	25.2	13.8	5.6	13.3

The P2–P3 transition provides the best accuracy–complexity balance. Dual-position deployment adds only 0.2 points in mAP0.5 but substantially increases cost, so it is not used.

4.3.3 Structural Ablation of DSAS

DSAS consists of multi-scale feature alignment (MFA), dynamic scale weighting (DSW), local scale correction (LSC), and residual correction (RC); Table 8 reports their progressive ablation.

Table 8. Ablation Results of the Internal Components of DSAS

MFA	DSW	LSC	RC	mAP0.5	mAP0.95	APs	Params	FLOPs
				37.4	21.9	11.8	2.6	6.5
√				38.0	22.5	12.2	2.9	7.3
√	√			40.1	24.0	13.0	3.3	8.2
√	√	√		40.5	24.3	13.2	3.5	8.6
√	√	√	√	40.7	24.5	13.3	3.5	8.7

MFA increases mAP0.5 from 37.4% to 38.0%, whereas DSW further raises it to 40.1%, indicating that input-dependent scale weighting is the main contributor rather than simple feature alignment. LSC and RC provide additional gains through local correction and stable residual information.

4.3.4 Structural Ablation of SOOS

Table 9 compares direct P2 addition, C3k2 fusion, and the lightweight SOOS configuration.

Adding P2 improves APS by 1.4 points. C3k2 gives the highest single-module accuracy but increases computation to 20.0 G and reduces speed to 129 FPS.

Table 9. Comparison of Different SOOS Configurations

Detection-Head Configuration	mAP0.5	mAP0.95	APs	Params	FLOPs	FPS
Original P3–P5 heads	37.4	21.9	11.8	2.6	6.5	186
Directly adding a P2 head	39.7	23.6	13.2	4.8	18.1	132
P2 with C3k2 fusion	40.4	24.1	13.6	5.2	20.0	129
P2+LiteFusion	40.1	23.9	13.4	4.4	15.6	162
P2+LiteFusion+lightweightdecoupled head	40.2	24.0	13.5	4.2	14.5	164

LiteFusion with the lightweight decoupled head is only 0.2 points lower in mAP0.5, but saves 5.5 G FLOPs and reaches 164 FPS; it is therefore adopted.

4.3.5 Comparison of Bounding-Box Regression Losses

Table 10 compares Complete-IoU (CIoU), SCYLLA-IoU (SIoU), and WIoU v3 on the complete structural model.

Table 10. Results of Different Bounding-Box Regression Losses

Bounding-Box Loss	Precision	Recall	mAP0.5	mAP0.95	APs
CIoU	55.3	43.8	45.6	28.0	15.1
SIoU	55.5	44.0	45.8	28.2	15.2
WIoU v3	56.0	44.5	46.3	28.6	15.4

WIoU v3 performs best across the reported metrics without increasing inference complexity, so it is used in the final model.

4.4 Visual Analysis of Detection Results

Three representative VisDrone2019 validation images compare DDS-YOLO with YOLOv11n under multi-scale road traffic, weak pedestrians, and distant dense objects. Enlarged regions highlight local differences.

4.4.1 Detection Results in an Urban Road Scene with Multi-Scale Objects

Figure 6 shows an urban road scene with strong scale variation.

Both models detect clear foreground objects, but DDS-YOLO recovers more small vehicles and pedestrians in the middle and distant regions.

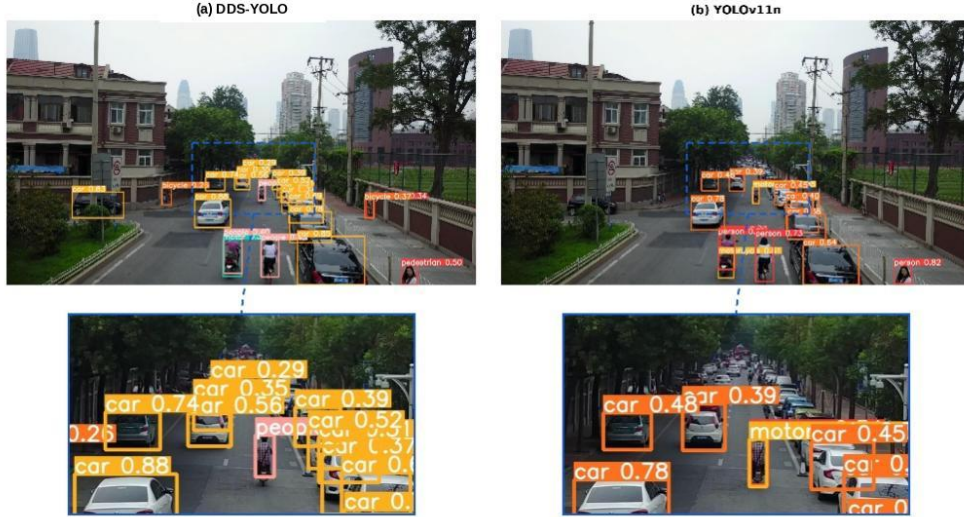


Figure 6. Comparison of DDS-YOLO and YOLOv11n in an urban road scene with multi-scale objects.

The enlarged region confirms denser and more complete predictions by DDS-YOLO, whereas YOLOv11n misses several low-resolution objects.

This result is consistent with improved detail preservation and cross-scale fusion.

4.4.2 Detection Results for Weak Pedestrians in Complex Backgrounds

Figure 7 shows weak pedestrians affected by reflections, occlusion, and complex textures.

YOLOv11n detects the salient vehicles but misses several nearby pedestrians with incomplete contours and low contrast.

DDS-YOLO produces separate boxes for more adjacent pedestrians, indicating stronger local discrimination.

DES strengthens local and contextual cues, while the P2 branch provides a higher-resolution prediction space.



Figure7. Comparison of detection results for weak pedestrian objects in a complex background.

4.4.3 Detection Results for Distant and Densely Distributed Traffic Objects

Figure 8 presents distant, dense traffic objects under strong perspective distortion.



Figure8. Detection comparison for distant dense objects.

Both models detect nearby vehicles, but DDS-YOLO maintains more continuous responses in the middle and distant regions, including smaller motorcycles.

The enlarged area shows more complete coverage and better separation of adjacent targets than YOLOv11n.

The result is consistent with adaptive scale fusion and high-resolution P2 prediction.

4.4.4 Overall Analysis

Across the three scenes, DDS-YOLO improves coverage of distant vehicles, weak pedestrians, and dense small targets while retaining stable detection of large foreground objects.

These qualitative trends agree with the quantitative and ablation results: DES enhances details, DSAS improves cross-level fusion through input-dependent scale weighting, and SOOS strengthens high-resolution prediction.

5 Conclusion

This paper proposes DDS-YOLO for UAV small object detection by integrating DES, DSAS, SOOS, and WIoU v3 into YOLOv11n. DES enhances local, contextual, and high-frequency cues; DSAS performs input-dependent cross-scale fusion; SOOS adds a lightweight P2 prediction

branch; and WIoU v3 improves regression supervision.

On VisDrone2019, DDS-YOLO achieves 56.0% Precision, 44.5% Recall, 46.3% mAP_{0.5}, and 28.6% mAP_{0.95}, improving YOLOv11n by 9.8, 9.4, 8.9, and 6.7 percentage points, respectively. Ablation and visual analyses confirm complementary gains for distant, weak, and dense small objects. With 8.7 M parameters, 23.8 G FLOPs, and 161 FPS, the model maintains a favorable accuracy–efficiency balance for real-time UAV perception.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Shandong Provincial Natural Science Foundation (grant number: ZR2022ME099) and Doctoral Funding Program of Shandong University of Aeronautics (grant number: 2023Y26).

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request

References

- [1] Tang G, Ni J, Zhao Y, et al. A survey of object detection for UAVs based on deep learning[J]. Remote Sensing, 2024, 16(1): 149.
- [2] Sun N, Zhao J, Shi Q, et al. Moving target tracking by unmanned aerial vehicle: A survey and taxonomy[J]. IEEE Transactions on Industrial Informatics, 2024, 20(5): 7056-7068.
- [3] Zhao C, Liu R W, Qu J, et al. Deep learning-based object detection in maritime unmanned aerial vehicle imagery: Review and experimental comparisons[J]. Engineering Applications of Artificial Intelligence, 2024, 128: 107513.
- [4] Shakya B, El-Gayar O F, Kaabi J, et al. Small-object detection at the edge: A Pareto-efficient benchmark of lightweight YOLO models on UAV and overhead datasets[J]. IEEE Access, 2026, 14: 528-548.
- [5] Fan Q, Li Y, Deveci M, et al. LUD-YOLO: A novel lightweight object detection network for unmanned aerial vehicle[J]. Information Sciences, 2025, 686: 121366.
- [6] Hou T, Leng C, Wang J, et al. MFEL-YOLO for small object detection in UAV aerial images[J]. Expert Systems with Applications, 2025, 291: 128459.
- [7] Xie B, Wang Y, Han M, et al. Density-guided two-stage small object detection in UAV images[J]. Expert Systems with Applications, 2026, 297: 129346.
- [8] Xu Z, Wang T, Skidmore A K, et al. A review of deep learning techniques for detecting animals in aerial and satellite images[J]. International Journal of Applied Earth Observation and Geoinformation, 2024, 128: 103732.
- [9] Liu F, Zheng Q, Tian X, et al. Rethinking the multi-scale feature hierarchy in object detection transformer (DETR)[J]. Applied Soft Computing, 2025, 175: 113081.
- [10] Li X, Diao W, Mao Y, et al. OGMN: Occlusion-guided multi-task network for object detection in UAV images[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2023, 199: 242-257.
- [11] Luo X, Ren J, Su L, et al. SRA-YOLO: Enhancing tiny person detection on construction sites via Riemannian metric coordinate attention[J/OL]. Expert Systems with Applications, 2026,

324: 132525.

- [12] Islam M S B, Chowdhury M E H, Hasan-Zia M, et al. VisioDECT: A novel approach to drone detection using CBAM-integrated YOLO and GELAN-E models[J]. *Neural Computing and Applications*, 2025, 37: 20181-20204.
- [13] Xu S, Huang W, Wang D, et al. Automatic pine wilt disease detection based on improved YOLOv8 UAV multispectral imagery[J]. *Ecological Informatics*, 2024, 84: 102846.
- [14] Usman M T, Khan H, Rida I, et al. Lightweight transformer-driven multi-scale trapezoidal attention network for saliency detection[J]. *Engineering Applications of Artificial Intelligence*, 2025, 155: 110917.
- [15] Chen Y, Yuan X, Wang J, et al. YOLO-MS: Rethinking multi-scale representation learning for real-time object detection[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025, 47(6): 4240-4252.
- [16] Doherty J, Gardiner B, Kerr E, et al. BiFPN-YOLO: One-stage object detection integrating bi-directional feature pyramid networks[J]. *Pattern Recognition*, 2025, 160: 111209.
- [17] Liu Y, Zhao J, Xu C, et al. YOLO-Pika: A lightweight improved model of YOLOv8n incorporating Fusion_Block and multi-scale fusion FPN and its application in the precise detection of plateau pikas[J]. *Frontiers in Plant Science*, 2025, 16: 1607492.
- [18] Wei B, Hu Y, Yuan D, et al. Underwater concrete crack detection of dams via CycleGAN-based data enhancement and optimized multi-scale YOLO11[J]. *Construction and Building Materials*, 2026, 514: 145497.
- [19] Chu H, Liu H, Zhuo J, et al. Occlusion-guided multi-modal fusion for vehicle-infrastructure cooperative 3D object detection[J]. *Pattern Recognition*, 2025, 157: 110939.
- [20] Mueller T G, Buckner M A. UAV-based remote sensing of bee nesting aggregations with computer vision for object detection[J]. *Journal of Applied Ecology*, 2026, 63(1): e70285.
- [21] Bárbulo Barrios D, Valente J, van Langevelde F. Monitoring mammalian herbivores via convolutional neural networks implemented on thermal UAV imagery[J]. *Computers and Electronics in Agriculture*, 2024, 218: 108713.
- [22] Kuang W, Ho H W, Zhou Y, et al. A comprehensive review on tree detection methods using point cloud and aerial imagery from unmanned aerial vehicles[J]. *Computers and Electronics in Agriculture*, 2024, 227: 109476.
- [23] Zheng C, Bai G, Meng Y, et al. Open-pit mine occlusion object detection for unmanned transport vehicles[J]. *Engineering Applications of Artificial Intelligence*, 2025, 159: 111436.
- [24] Li W, Zhao W, Du Y. Large-scale wind turbine blade operational condition monitoring based on UAV and improved YOLOv5 deep learning model[J]. *Mechanical Systems and Signal Processing*, 2025, 226: 112386.
- [25] He M, Liang Q, Deng X, et al. MFI-YOLO: Multi-fault insulator detection based on an improved YOLOv8[J]. *IEEE Transactions on Power Delivery*, 2024, 39(1): 168-179.
- [26] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]. *IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, 2016: 779-788.
- [27] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger[C]. *IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, 2017: 6517-6525.
- [28] Bochkovskiy A, Wang C Y, Liao H M. YOLOv4: Optimal speed and accuracy of object detection[PP/OL]. 2020, arXiv: 2004.10934.

- [29] Wang C Y, Bochkovskiy A, Liao H M. YOLOv7:Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver,2023: 7464-7475.
- [30] Wang C Y, Yeh I H, Liao H Y M. YOLOv9: Learning what you want to learn using programmable gradient information[C] European Conference on Computer Vision. Cham: Springer, 2024: 1-21.
- [31] Yue M, Zhang L, Huang J, et al. Lightweight and efficient tiny-object detection based on improved YOLOv8n for UAV aerial images[J]. Drones, 2024, 8(6): 276.
- [32] Kim M, Jeong J, Kim S. ECAP-YOLO: Efficient channel attention pyramid YOLO for small object detection in aerial image[J]. Remote Sensing, 2021, 13(23): 4851.
- [33] Tong K, Wu Y, Zhou F. Recent advances in small object detection based on deep learning: A review[J]. Image and Vision Computing, 2020, 97: 103910.
- [34] Wu J, Pan Z, Lei B, et al. FSANet: Feature-and-spatial-aligned network for tiny object detection in remote sensing images[J]. IEEE Transactions on Geoscience and Remote Sensing, 2022, 60: 5630717.
- [35] Li C, Yang T, Zhu S, et al. Density map guided object detection in aerial images[C] Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020: 190-191.
- [36] Zeng S, Yang W, Jiao Y, et al. SCA-YOLO: A new small object detection model for UAV images[J]. The Visual Computer, 2024, 40(3): 1787-1803.
- [37] Tan M, Pang R, Le Q V. EfficientDet: Scalable and efficient object detection[C]Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 10781-10790.
- [38] Zhang B, Zhang Y. UAV small object detection algorithm based on dynamic feature aggregation and hierarchical attention mechanism[J]. IEEE Access, 2025, 13: 178277-178292.
- [39] Yang Y, Gao X, Wang Y, et al. VAMYOLOX: An accurate and efficient object detection algorithm based on visual attention mechanism for UAV optical sensors[J]. IEEE Sensors Journal, 2023, 23(11): 11139-11155.
- [40] Zhao Y, Lv W, Xu S, et al. DETRs beat YOLOs on real-time object detection[C]Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 16965-16974.
- [41] Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, 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, Seoul, Republic of Korea, 27–28 October 2019.