



ARTICLE

An Edge-Computing-Oriented Small-Object Detection Algorithm for UAV Aerial Images

Chanchan Zhao^{1,#}, Xiaoyu Gao^{1,#}, Bao Shi^{2,*} and Ziyang Zhang¹

¹College of Intelligent Science and Technology, Inner Mongolia University of Technology, Hohhot, China

²School of Information Engineering, Inner Mongolia University of Technology, Hohhot, China

*Corresponding Author: Bao Shi. Email: kshibao@163.com

#These authors are co-first authors and contributed equally to this work

Received: 08 May 2026; Accepted: 02 July 2026; Published: 13 August 2026

ABSTRACT: Detecting objects in unmanned aerial vehicle (UAV) imagery is challenging because most targets occupy only a small number of pixels and are often distributed in crowded regions with cluttered backgrounds. For edge-side UAV applications, the detector must also remain compact enough for real-time inference on low-power computing platforms. To meet these requirements, this study develops a YOLOv11n-based small-object detector by redesigning feature extraction, cross-scale fusion, and prediction modules. In the backbone, the proposed Dual-Context Large-Small Convolution (DCLSCov) is embedded into the C3k2 structure to form C3k2-DC, allowing the network to capture broader contextual cues while retaining local details useful for small targets. In the neck, Cross-Scale Bi-Attention Fusion (CSBAF) is introduced to regulate the interaction between low-level spatial features and high-level semantic features, thereby reducing ineffective feature mixing during multi-scale fusion. In the detection head, UAV-Task Aligned Dual-Dynamic Detection Head (UAV-TAD²H) is designed to strengthen task-specific feature learning for classification and localization under scale-varying aerial scenes. Experiments on VisDrone2019 show that the proposed model reaches 0.375 mAP@0.5 and 0.230 mAP@0.5:0.95, improving the YOLOv11n baseline by 5.1 and 4.3 percentage points, respectively. The model contains 3.12M parameters and requires 12.3 giga floating-point operations (GFLOPs), while maintaining 33 frames per second (FPS) on Jetson Orin NX under TensorRT FP16 inference. Additional evaluations on PASCAL VOC2012 and BDD100K further indicate that the proposed modules can provide stable performance gains across different detection scenarios.

KEYWORDS: Edge-oriented detection; UAV aerial images; small-object detection; multi-scale feature fusion; YOLOv11n

1 Introduction

Unmanned aerial vehicle (UAV) platforms provide a flexible way to acquire visual information from low-altitude perspectives and have been widely used in traffic monitoring, emergency response, agricultural inspection, and urban management. Unlike ground-view images, aerial images usually contain objects with small pixel areas, large viewpoint changes, and dense spatial distributions. Vehicles, pedestrians, and other targets may appear in clusters or be partially occluded by buildings, trees, shadows, and motion blur. These characteristics make UAV object detection more sensitive to feature loss during downsampling and more dependent on effective multi-scale representation. At the same time, practical UAV systems often rely on embedded computing units, where memory, power consumption, and inference latency are

strictly constrained. Therefore, improving small-object detection accuracy without excessively increasing computational cost is essential for UAV edge perception.

Deep-learning-based object detection algorithms currently dominate this field and can generally be grouped into one-stage, two-stage, and end-to-end transformer-based detectors [1]. Representative one-stage detectors include the YOLO series [2–4], SSD series [5], and EfficientDet series [6]. DETR-family methods represent end-to-end transformer-based detectors [7], while representative two-stage detectors include R-CNN [8], Faster R-CNN [9], and Mask R-CNN [10]. Two-stage detectors first generate candidate regions and then perform refined classification and localization, which usually leads to higher detection accuracy. However, their complex network structures and high computational costs limit their applicability in real-time UAV scenarios. In contrast, single-stage detectors directly predict object categories and bounding boxes in a unified framework, providing faster inference speed and better suitability for edge-oriented UAV deployment.

In recent years, improving the balance between detection accuracy and computational complexity in UAV aerial object detection has become an important research topic. Jiang et al. [11] proposed AEM-YOLOv8s, which combines alterable kernel convolution (AKConv) and efficient multi-scale attention (EMA) within the C2f module to improve feature extraction while reducing model parameters. Wang et al. [12] proposed DAFPN-YOLO, which constructs an enhanced Drone-AFPN for adaptive multi-scale feature fusion and integrates C2f_Dual and SPPELAN modules to improve feature extraction and information aggregation in UAV-based small-object detection. Chen et al. [13] proposed MFF-YOLO, which introduces a Multi-scale Feature Fusion Strategy consisting of the Multiple Features C2f module and the Scale Sequence Feature Fusion module to strengthen cross-level feature integration for UAV aerial imagery. Liu et al. [14] designed a parallel dual-path backbone network to improve complementary feature extraction. Xu et al. [15] combined a generalized high-efficiency layer aggregation network with Sobel convolution to enhance edge feature extraction and replaced the SPPF module with an attention-based intra-scale feature interaction (AIFI) module to alleviate feature distortion caused by multi-scale pooling. Li and Li [16] introduced a new feature pyramid structure by adding a 160×160 feature map in the P2 layer for enhanced small-object feature extraction. They also proposed the CSPOK module and the ContextGuidedBlock_Down (CGBD) convolution module to improve global feature extraction and multi-scale feature fusion. Lu and Sun [17] proposed LPS-YOLO, which introduced the SKAPP module and combined E-BiFPN with OFTP to improve backbone feature transmission and fusion efficiency.

In addition to network structure optimization and feature extraction enhancement, many studies have further improved UAV object detection performance through the design of detection heads, loss functions, and post-processing strategies. Li and Chen [18] proposed MFA-YOLO for small-object detection in drone imagery, which combines a Local Feature Mapping unit, a Progressive Shared Atrous Pyramid module, and a Dynamic Decoupling Head to enhance fine-grained feature extraction, multi-scale feature integration, and adaptive task alignment. Chen et al. [19] proposed YOLO-LE, which introduces C2f-Dy and LDown modules, designs an adaptive feature fusion module, and incorporates distributed focal loss with a lightweight self-attention mechanism to improve small-object recognition and bounding-box regression accuracy.

Gao and Li [20] proposed YOLO-S3DT, which modifies YOLOv8 from the detection head, upsampling module, feature extraction module, and positive-sample assignment strategy to improve small-target detection performance in UAV images. Tu et al. [21] combined Soft-NMS with SIOU to reduce missed detections and false positives in densely distributed small-object scenarios. Ma et al. [22] proposed PSC-Head (Parameter Shared Convolution Head) to improve detection efficiency and reduce model complexity. Xue et al. [23] designed CCDHead with shared convolution to reduce parameters and adopted WIoU-V3

to improve localization performance. Zheng et al. [24] added a P2 detection head to enhance small-object detection by preserving finer spatial features.

To further clarify the architectural and functional differences between the proposed framework and recent UAV-oriented YOLO variants, a compact comparison is provided in Table 1.

Table 1: Comparison with recent UAV-oriented YOLO variants.

Method	Key Design	Main Difference from Ours
AEM-YOLOv8s [11]	AKConv + EMA + BiFPN	Focuses on YOLOv8s backbone/neck enhancement
LPS-YOLO [17]	SPDConv + SKAPP + E-BiFPN/OFTP	Focuses on lightweight feature extraction and transfer
RSP-YOLOv11n [24]	RCSOSA + SEA + P2 head	Uses attention and an extra P2 head for YOLOv11n
Ours	DCLSCConv + CSBAF + UAV-TAD ² H	Jointly optimizes backbone, neck, and detection head

Different from these methods, the proposed framework jointly improves backbone feature extraction, neck feature fusion, and task-aligned detection within a YOLOv11n-based architecture.

Existing UAV-oriented detectors have improved aerial image understanding from different perspectives, such as receptive-field enhancement, attention-based feature extraction, feature pyramid redesign, lightweight convolution, and detection-head optimization. However, many methods emphasize one part of the detection pipeline while leaving the coordination among backbone representation, neck fusion, and task-specific prediction insufficiently explored. For YOLOv11n, this issue is more prominent because its compact architecture limits redundant computation but also makes small-object features more vulnerable to information loss. Motivated by this observation, this study improves YOLOv11n through a coordinated design that enhances contextual feature extraction, regulates cross-scale feature interaction, and strengthens task-aligned prediction for UAV aerial scenes.

The main contributions of this work are summarized as follows:

- A C3k2-DC structure is constructed by embedding DCLSCConv into the YOLOv11n backbone, enabling the model to combine large-kernel contextual perception with small-kernel local aggregation for UAV small-object representation.
- A CSBAF module is developed for the neck network to perform bidirectional gated interaction between shallow spatial information and deep semantic information, reducing redundant feature fusion.
- A UAV-TAD²H detection head is designed to improve task-specific feature learning for classification and localization, thereby enhancing prediction adaptability for dense and scale-varying aerial targets.

2 Improved Model

The proposed model is designed around the main information flow of YOLOv11n, including backbone feature extraction, neck feature fusion, and detection-head prediction. In UAV aerial images, small targets are easily weakened after repeated downsampling, while dense backgrounds may introduce irrelevant responses into multi-scale fusion. Moreover, the compact design of YOLOv11n leaves limited redundancy for directly increasing network depth or adding high-resolution detection layers. Therefore, instead of simply enlarging the baseline model, this work introduces targeted modifications at three key stages of the detection pipeline.

The backbone is enhanced to preserve both contextual and local cues, the neck is redesigned to filter and exchange cross-scale information more selectively, and the detection head is adjusted to improve the coordination between classification and localization. The overall architecture of the proposed model is shown in Fig. 1.

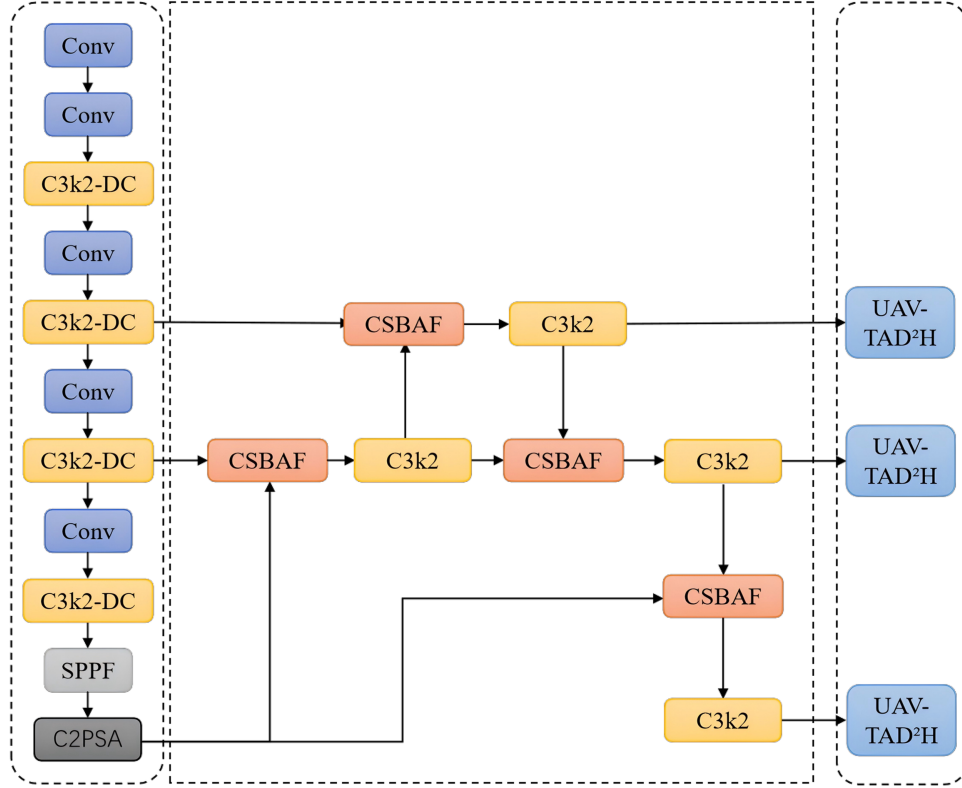


Figure 1: Overall architecture of the proposed YOLOv11n-based detection model.

In this architecture, DCLSCov is embedded into the C3k2 structure to build the C3k2-DC module for multi-scale feature representation. CSBAF is then used in the neck to guide feature exchange between adjacent pyramid levels. Finally, UAV-TAD²H processes the enhanced P3, P4, and P5 features through task-aware branches, allowing the detector to better adapt to dense and scale-varying UAV targets.

2.1 Dual Context C3k2-DC for Multi-Scale Feature Representation

Small objects in UAV images often lack sufficient texture and shape cues after several convolutional downsampling stages. A backbone module for this task should therefore perceive a wider surrounding context while still retaining local structures around small targets. The LSNet design offers an efficient strategy for combining large-range perception with small-region feature aggregation. Inspired by the “See Large, Focus Small” strategy in LSNet [25], this study designs DCLSCov to introduce dual-context perception into the C3k2 structure of YOLOv11n. The structure of the proposed DCLSCov module is shown in Fig. 2.

DCLSCov contains three functional parts. The first part, MultiScaleDLKP, extracts contextual information through two depthwise convolution branches with 7×7 and 11×11 kernels. These branches provide complementary mid-range and long-range responses for aerial targets with different apparent sizes. The second part, SKDA, uses 3×3 local patches and aggregates them according to the adaptive weights generated by MultiScaleDLKP, which helps recover fine local details that are important for small-object discrimination.

The third part recalibrates channel responses through channel attention so that informative feature channels are emphasized while redundant activations are suppressed. The dual-context dynamic weight-generation process in DCLSCov is defined as follows:

$$w_i = P_{MLS}(x_i; K_1, K_2) = PW(DW_{(K_1 \times K_1)}(z_i) + DW_{(K_2 \times K_2)}(z_i)), \quad z_i = PW_{(\frac{C}{2})}(x_i), K_1 = 7, \quad K_2 = 11 \quad (1)$$

where x_i denotes the input feature at spatial position i , $PW_{(\frac{C}{2})}(\cdot)$ denotes pointwise convolution used to reduce the channel dimension to $\frac{C}{2}$, and $DW_{(K_1 \times K_1)}(\cdot)$ and $DW_{(K_2 \times K_2)}(\cdot)$ denote depthwise convolutions with kernel sizes K_1 and K_2 , respectively. In this work, $K_1 = 7$ and $K_2 = 11$. The generated weight w_i is then used for small-kernel dynamic aggregation. Here, P_{MLS} denotes the multi-scale large-small dynamic weight-generation operation.

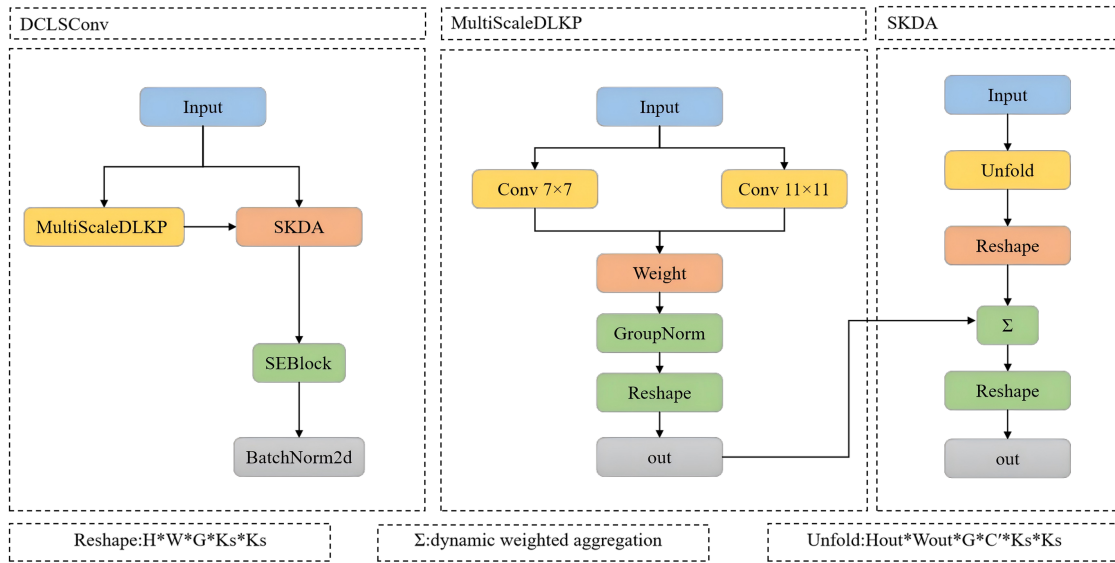


Figure 2: Structure of the proposed DCLSCov module.

In YOLOv11n, the original C3k2 module has limited capability in extracting discriminative features for UAV aerial small-object detection. To improve multi-scale feature representation, the proposed DCLSCov module is integrated into the C3k2 structure to form the improved C3k2-DC module, as shown in Fig. 3. By combining multi-scale large-kernel contextual perception, small-kernel dynamic aggregation, and channel attention enhancement, the proposed module improves the backbone network's ability to capture both global contextual information and fine-grained local details across different object scales.

Compared with fixed-receptive-field convolutions, the improved C3k2-DC module provides more flexible feature representation by coupling mid- and long-range contextual modeling with local dynamic aggregation, making it more suitable for UAV small objects under complex backgrounds and scale variations.

2.2 Cross-Scale Bi-Attention Fusion for Feature Fusion Optimization

In the original YOLOv11n neck, feature maps from different pyramid levels are mainly fused through upsampling and concatenation. Although this operation is simple and efficient, it treats channels and spatial responses with nearly equal importance after resolution alignment. In UAV aerial images, weak small-object cues from shallow layers may be overwhelmed by background responses, while high-level semantic features may not provide sufficient guidance for dense local regions. To make cross-scale fusion more selective,

this study proposes CSBAF, which introduces bidirectional attention gates between low-level and high-level features before final concatenation. The improved PAN-FPN-CSBAF structure is shown in Fig. 4.

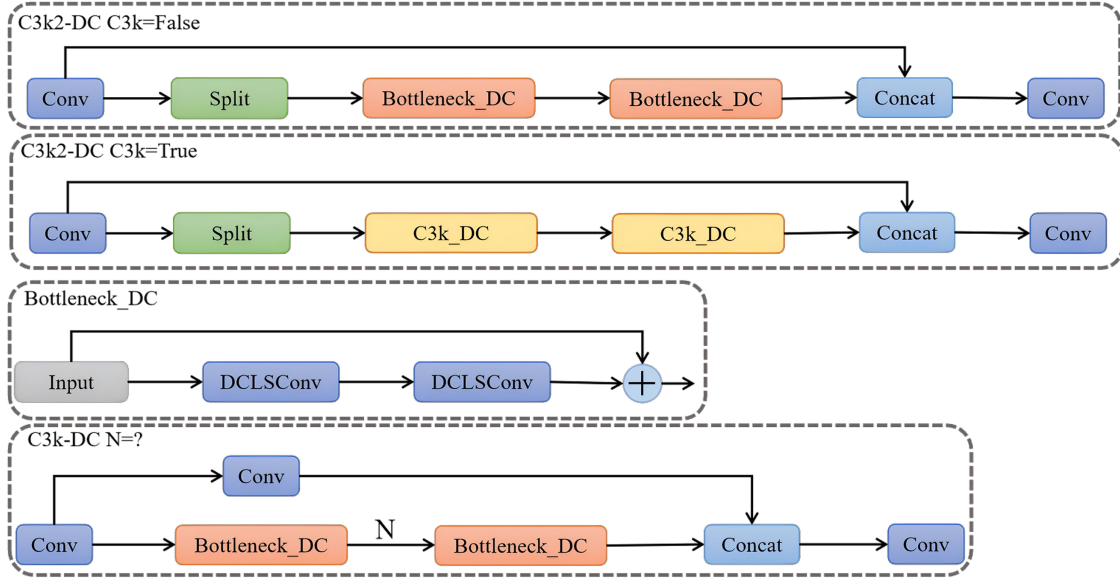


Figure 3: Structure of the proposed C3k2-DC module.

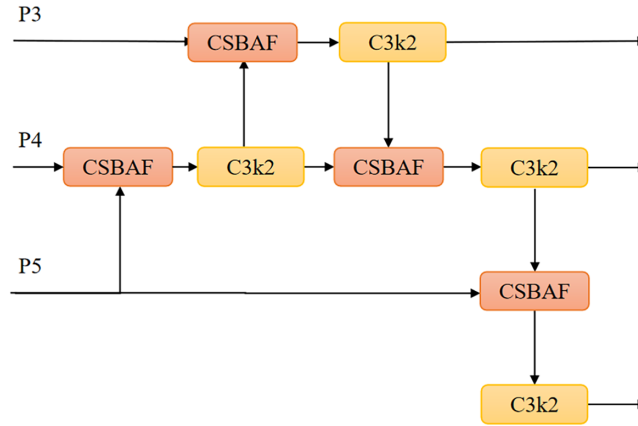


Figure 4: Structure of the proposed PAN-FPN-CSBAF network.

The internal structure of CSBAF is shown in Fig. 5. The module first aligns the channels of the input low-level and high-level features using 1×1 convolutions. Then, attention gates are generated to control the information retained from each scale. The low-level branch receives semantic guidance from the high-level branch to suppress irrelevant spatial details, whereas the high-level branch obtains spatial compensation from the low-level branch to enhance local target cues. After bidirectional interaction, the enhanced high-level features are upsampled and concatenated with the enhanced low-level features, followed by a 3×3 convolution to obtain the fused output.

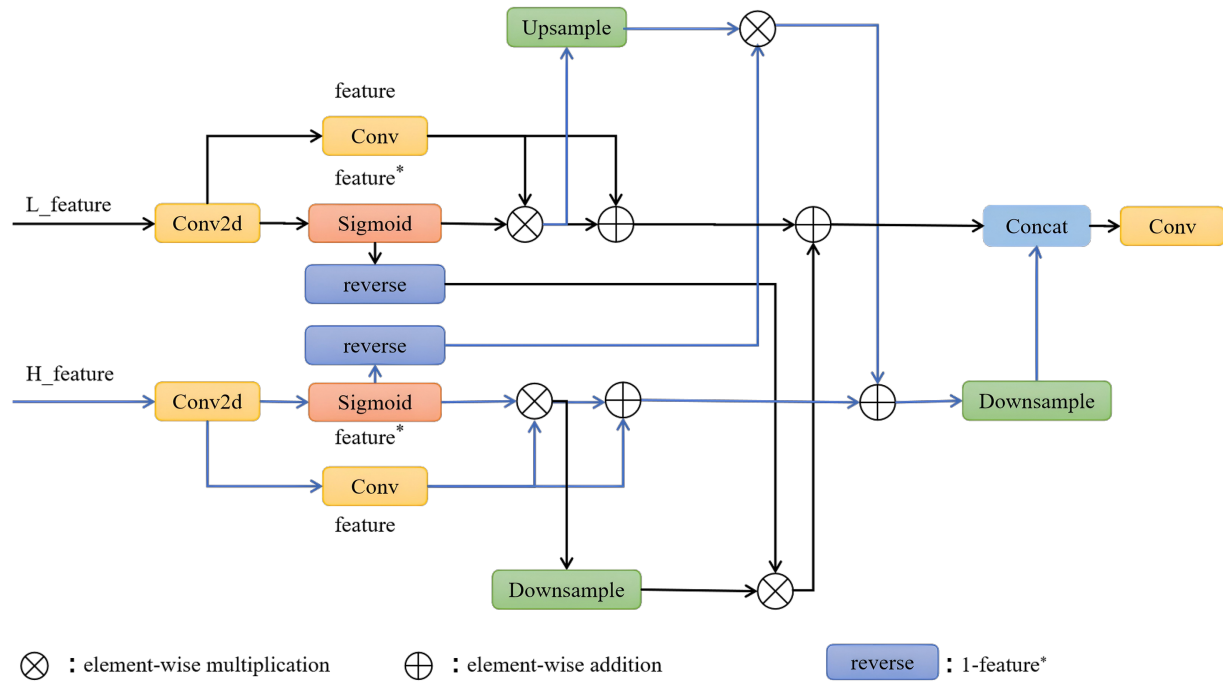


Figure 5: Structure of the proposed CSBAF module.

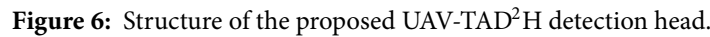
The cross-scale bi-attention feature fusion process is expressed as follows:

$$\begin{aligned}
 F_L &= \text{Conv}_{(1 \times 1)}(F_L^{\text{in}}), \quad F_H = \text{Conv}_{(1 \times 1)}(F_H^{\text{in}}) \\
 g_L &= \sigma(F_L), \quad g_H = \sigma(F_H) \\
 F_L^{\text{out}} &= F_L + F_L \odot g_L + (1 - g_L) \odot \uparrow(F_H \odot g_H) \\
 F_H^{\text{out}} &= F_H + F_H \odot g_H + (1 - g_H) \odot \downarrow(F_L \odot g_L) \\
 F_{\text{final}} &= \text{Conv}_{(3 \times 3)}([\uparrow F_H^{\text{out}}, F_L^{\text{out}}])
 \end{aligned} \tag{2}$$

where F_L^{in} and F_H^{in} denote the input low-level and high-level features, respectively. $\text{Conv}_{(1 \times 1)}(\cdot)$ is used for channel alignment. F_L and F_H denote the aligned low-level and high-level features. $\sigma(\cdot)$ denotes the sigmoid activation function. g_L and g_H are the attention gates generated for the low-level and high-level features, respectively. $\odot$ denotes element-wise multiplication. $\uparrow$ and $\downarrow$ denote upsampling and downsampling operations for spatial resolution alignment. F_L^{out} and F_H^{out} denote the enhanced low-level and high-level features after bidirectional cross-scale interaction, respectively. $[\cdot; \cdot]$ denotes channel-wise concatenation, and F_{final} is the output feature of the CSBAF module.

2.3 UAV-Task Aligned Dual-Dynamic Detection Head (UAV-TAD²H)

The detection head directly determines how multi-scale features are converted into category scores and bounding-box predictions. In UAV aerial images, classification and localization have different feature preferences: classification relies more on discriminative semantic cues, whereas localization requires spatially precise boundary information. When both tasks share highly coupled features, background clutter and scale variation may weaken prediction quality, especially for small and crowded targets. To address this problem within a compact detector, this study proposes a UAV-Task Aligned Dual-Dynamic Detection Head (UAV-TAD²H), as shown in Fig. 6.



After receiving the multi-scale enhanced features from the neck network, the detection head first performs feature initialization. The process can be expressed as follows:

where F_s denotes the input feature map from scale s , $s \in \{3, 4, 5\}$. $\text{Conv}_{3 \times 3}^{(C_f)}(\cdot)$ denotes a 3×3 convolution that maps the input feature to C_f channels. $\text{GN}(\cdot)$ and $\text{SiLU}(\cdot)$ denote group normalization and the SiLU activation function, respectively. This initialization improves feature stability before task-specific feature decomposition.

$$\begin{aligned} \mathbf{c}_s &= \text{GAP}(\mathbf{F}_s^{\text{init}}) \\ \mathbf{g}_s^t &= \sigma\left(\text{Conv}_{1 \times 1}^{t(\delta(\mathbf{c}_s))}\right), t \in \{\text{cls}, \text{reg}\} \\ \mathbf{F}_s^t &= \text{SiLU}\left(\text{GN}\left((\mathbf{W}^t \odot \mathbf{g}_s^t) * \mathbf{F}_s^{\text{init}}\right)\right), \mathbf{F}_s^t \in \mathbb{R}^{B \times C_f \times H_s \times W_s} \end{aligned} \quad (4)$$

where c_s denotes the global context descriptor obtained by global average pooling. g_s^t denotes the task-specific attention weight for task t , where $t = \text{cls}$ corresponds to the classification branch and $t = \text{reg}$ corresponds to the regression branch. $\delta(\cdot)$ denotes the ReLU activation function, and $\sigma(\cdot)$ denotes the sigmoid activation function. W^t represents the task-specific convolution kernel. $\odot$ denotes element-wise multiplication, and $*$ denotes convolution. F_s^t denotes the task-enhanced feature for scale s .

Through attention-guided feature modulation, the task decomposition module dynamically adjusts task-specific convolution weights, reduces feature interference between classification and regression, and improves feature alignment across different scales.

3 Experimental Results and Analysis

3.1 Dataset

In this study, the proposed method is evaluated on several public datasets to assess its effectiveness and generalization ability in complex scenarios. The main experiments are conducted on the VisDrone2019 dataset, which was collected and released by the AISKYEYE team from the Machine Learning and Data Mining Laboratory at Tianjin University [26]. The images in VisDrone2019 were captured by various UAV-mounted cameras and cover diverse urban and rural scenes across 14 cities under different weather and illumination conditions. The dataset contains 6471 training images, 548 validation images, and 1610 test images. Its large number of small objects, dense object distributions, and frequent occlusions make it a challenging benchmark for UAV aerial object detection.

Additional experiments are conducted on PASCAL VOC2012 [27] and BDD100K [28] to evaluate cross-dataset generalization. PASCAL VOC2012 contains 20 common object categories across diverse indoor and outdoor scenes, while BDD100K provides large-scale road-scene images under various real-world driving conditions.

3.2 Experimental Environment Configuration

All experiments were conducted using the PyTorch deep learning framework. The models were trained and evaluated on a workstation equipped with an NVIDIA RTX 4060 GPU. The detailed experimental environment is listed in Table 2. The input image size was set to 640×640 . Unless otherwise specified, all models were trained and evaluated under the same parameter settings to ensure a fair comparison.

Table 2: Training experiment environment configuration.

Item	Version
Operating System	Ubuntu 24.04
Programming Language	Python 3.9.21
Graphics Card	RTX 4060 (8 GB)
Processor	Intel(R) Core(TM) i7-14650HX (2.20 GHz)
Memory	32 GB
Hard Disk	1 TB
Deep Learning Framework	PyTorch 2.5.1 + cu120

3.3 UAV Platform and Configuration

To evaluate the edge deployment feasibility of the proposed model, a UAV sensing simulation platform was constructed to emulate the airborne visual perception pipeline. The platform consists of a low-power

edge computing unit, a power conversion module, and auxiliary sensing components. The edge computing unit is based on the NVIDIA Jetson Orin NX 16 GB platform, which is suitable for lightweight deep-learning inference under power-constrained UAV application scenarios.

During deployment, TensorRT FP16 optimization was used to accelerate inference through graph optimization, layer fusion, kernel auto-tuning, and reduced-precision computation. This reduces computational and memory access overhead while maintaining comparable detection accuracy, making it suitable for UAV edge devices with limited computing resources and power budgets.

Table 3 lists the configuration of the edge computing device.

Table 3: Edge computing environment configuration.

Item	Version
System	Ubuntu 22.04 LTS
Language	Python 3.10
Edge Computing Platform	NVIDIA Jetson Orin NX 16 GB
Processor	Arm Cortex-A78AE v8.2 CPU
Memory	16 GB LPDDR5
Storage	128 GB NVMe SSD
Inference Framework	PyTorch 2.0.1
TensorRT	8.6.1
Power Mode	10 W low-power mode

The edge deployment performance of YOLOv11n and the proposed model on Jetson Orin NX is summarized in Table 4.

Table 4: Edge deployment performance on Jetson Orin NX.

Model	Input Size	Batch Size	Precision	FPS	Latency (ms/image)	Power Mode	Preprocessing/NMS
YOLOv11n	640×640	1	TensorRT FP16	46	21.74	10 W low-power mode	Excluded
Ours	640×640	1	TensorRT FP16	33	30.30	10 W low-power mode	Excluded

Based on the TensorRT FP16 optimization, the proposed model achieves 33 FPS at an input resolution of 640×640 on the NVIDIA Jetson Orin NX 16GB under the 10 W low-power mode. The corresponding latency is 30.30 ms per image. Under the same setting, the baseline YOLOv11n achieves 46 FPS, corresponding to 21.74 ms per image. Although the proposed model introduces additional computational cost, its inference speed remains suitable for real-time inference in the tested UAV edge-deployment setting. Therefore, the proposed method achieves a practical trade-off between detection accuracy and edge inference efficiency.

The reported FPS only measures TensorRT engine inference time. Image preprocessing, bounding-box decoding, and NMS post-processing are excluded from the FPS calculation. It should be noted that the 10 W value denotes the configured power mode of Jetson Orin NX rather than the measured runtime power consumption.

3.4 Model Evaluation Metrics

In UAV aerial small-object detection, false detections and missed detections are common due to dense object distributions, scale variations, and complex backgrounds. Therefore, mAP@0.5 and mAP@0.5:0.95 are used as the main evaluation metrics. Precision and recall are also reported to evaluate the model's detection reliability and completeness. In addition, the number of parameters and GFLOPs are used to measure model size and computational complexity.

- (1) The average precision (AP) and mean average precision (mAP) are defined as follows:

$$AP = \int_0^1 p(r) dr, \quad mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (5)$$

where $p(r)$ denotes the precision at recall level r , and N denotes the number of object categories.

- (2) Precision is defined as follows:

$$P = \frac{TP}{TP + FP} \quad (6)$$

where TP denotes true positives, and FP denotes false positives.

- (3) Recall is defined as follows:

$$R = \frac{TP}{TP + FN} \quad (7)$$

where FN denotes false negatives, namely objects that actually exist but are not detected.

- (4) Number of parameters. The number of parameters is used to evaluate the model size and storage requirement. A smaller number of parameters usually indicates a more compact model, while a larger number of parameters may improve feature representation but requires more storage and computational resources.
- (5) GFLOPs, namely giga floating-point operations, represent the number of floating-point operations required for one forward inference process and are commonly used to evaluate model computational complexity.

3.5 Comparative Experiments

[Table 5](#) compares the proposed model with representative general detectors, recent UAV-oriented YOLO-based methods, and transformer-based UAV detectors on the VisDrone2019 validation set. For UAV aerial small-object detection, both accuracy and model complexity should be considered because practical deployment is often limited by onboard computing resources.

As shown in [Table 5](#), the proposed model substantially improves the YOLOv11n baseline while keeping the model size relatively small. Compared with YOLOv11n, mAP@0.5 increases from 0.324 to 0.375, and mAP@0.5:0.95 increases from 0.187 to 0.230. These gains indicate that the redesigned backbone, neck, and detection head are beneficial for dense UAV aerial scenes, where small objects require both local detail preservation and cross-scale semantic guidance.

Compared with YOLOv11s, the proposed model achieves nearly the same mAP@0.5:0.95 with fewer parameters and lower computational cost. Specifically, the proposed model contains 3.12M parameters and requires 12.3 GFLOPs, whereas YOLOv11s contains 9.416M parameters and requires 21.3 GFLOPs. This comparison suggests that the improvement is not obtained by simply enlarging the model, but by reallocating computation to modules that are more relevant to UAV small-object detection.

Table 5: Algorithm comparative experiments.

Model	Param (M)	GFLOPs	P	R	mAP@0.5	mAP@0.5:0.95
Faster-R-CNN	63.20	370.0	0.346	0.368	0.309	0.131
SSD	12.30	63.2	0.211	0.358	0.240	0.119
YOLOv5s	9.10	23.8	0.504	0.380	0.388	0.232
YOLOv8s	11.20	28.5	0.506	0.381	0.390	0.192
YOLOv10n	2.26	6.5	0.442	0.342	0.342	0.198
YOLOv10s	7.22	21.4	0.506	0.381	0.390	0.238
YOLOv11n	2.58	6.3	0.43	0.332	0.324	0.187
YOLOv11s	9.416	21.3	0.505	0.375	0.386	0.231
AEM-YOLOv8s [11]	7.55	—	0.595	0.475	0.5	0.31
LPS-YOLO [17]	3.55	10.9	0.505	—	0.408	0.24
YOLOv8n-ACW [23]	2.1	6.1	0.563	0.367	0.37	0.212
UAV-DETR-R18 [29]	20.0	77.0	—	—	0.488	0.298
UAV-DETR-EV2 [29]	13.0	43.0	—	—	0.475	0.287
Ours	3.12	12.3	0.482	0.37	0.375	0.23

Note: UAV-DETR results are taken from the original paper. AP and AP50 are converted to mAP@0.5:0.95 and mAP@0.5, respectively. P and R are not reported and are marked as “—”.

Some UAV-oriented methods, such as AEM-YOLOv8s and LPS-YOLO, achieve higher absolute accuracy than the proposed model. Transformer-based UAV-DETR variants also obtain higher mAP values; for example, UAV-DETR-R18 reaches 0.488 mAP@0.5 and 0.298 mAP@0.5:0.95, but requires 20.0M parameters and 77.0 GFLOPs. In contrast, the proposed model uses only 3.12M parameters and 12.3 GFLOPs. Therefore, the proposed method does not aim to outperform all existing UAV-oriented detectors in absolute accuracy. Instead, it focuses on improving the YOLOv11n baseline and achieving a compact accuracy–complexity trade-off for edge-oriented UAV detection.

3.6 Ablation Experiment

To evaluate the contribution of each proposed module, ablation experiments were conducted on the VisDrone2019 dataset. YOLOv11n was used as the baseline model, and the proposed modules were gradually added to the baseline. Specifically, the C3k2 module was first replaced with the proposed C3k2-DC structure. Then, the upsampling-and-concatenation operation in the neck network was replaced with the CSBAF module. Finally, the original detection head was replaced with the UAV-TAD²H detection head. The detailed ablation results are shown in Table 6.

Table 6: Ablation results of the proposed modules.

Baseline	C3k2-DC	CSBAF	UAV-TAD ² H	FPS	Param (M)	GFLOPs	P	R	mAP@0.5	mAP@0.5:0.95
✓				46	2.58	6.3	0.43	0.332	0.324	0.187
✓	✓			42	2.60	6.5	0.45	0.342	0.341	0.199
✓	✓	✓		36	3.52	11.7	0.468	0.357	0.364	0.216
✓	✓	✓	✓	33	3.12	12.3	0.482	0.37	0.375	0.23

As shown in Table 6, each proposed module contributes to the detection performance of the model. After replacing C3k2 with the C3k2-DC structure, mAP@0.5 increases from 0.324 to 0.341, and mAP@0.5:0.95 increases from 0.187 to 0.199. Meanwhile, the number of parameters only slightly increases from 2.58M to 2.60M, indicating that DCLSCConv improves feature extraction with limited additional parameters. This result suggests that the combination of large-kernel contextual perception and small-kernel dynamic aggregation is beneficial for multi-scale small-object representation.

After introducing the CSBAF module, mAP@0.5 further increases from 0.341 to 0.364, and mAP@0.5:0.95 increases from 0.199 to 0.216. This improvement shows that CSBAF can enhance cross-scale feature fusion and reduce the loss of small-object information during feature transmission. However, CSBAF also increases the computational cost, with GFLOPs increasing from 6.5 to 11.7. This indicates that the improved feature fusion capability is obtained at the cost of additional computation.

After replacing the original detection head with UAV-TAD²H, mAP@0.5 increases from 0.364 to 0.375, and mAP@0.5:0.95 increases from 0.216 to 0.230. At the same time, the number of parameters decreases from 3.52M to 3.12M. This result indicates that the task-aligned detection head improves the coordination between classification and regression branches while maintaining a relatively compact model size.

When all proposed modules are integrated, the final model achieves mAP@0.5 and mAP@0.5:0.95 values of 0.375 and 0.230, respectively, which are 5.1 and 4.3 percentage points higher than those of the baseline YOLOv11n. Although the GFLOPs increase from 6.3 to 12.3, the proposed model still uses fewer parameters and lower computational cost than YOLOv11s while achieving comparable detection accuracy. Therefore, the proposed method aims to achieve a favorable accuracy–complexity trade-off rather than simply minimizing computational cost.

To further illustrate the effectiveness of the proposed modules, heatmap visualizations are shown in Fig. 7. In the upper row, after introducing DCLSCConv, the model produces stronger responses around small and distant objects, indicating improved fine-grained small-object feature representation. In the lower row, after introducing CSBAF, the feature responses become more concentrated on dense object regions, while responses in irrelevant background areas are relatively weakened. These visualization results indicate that DCLSCConv and CSBAF help enhance small-object representation and cross-scale feature fusion in complex UAV aerial scenes.

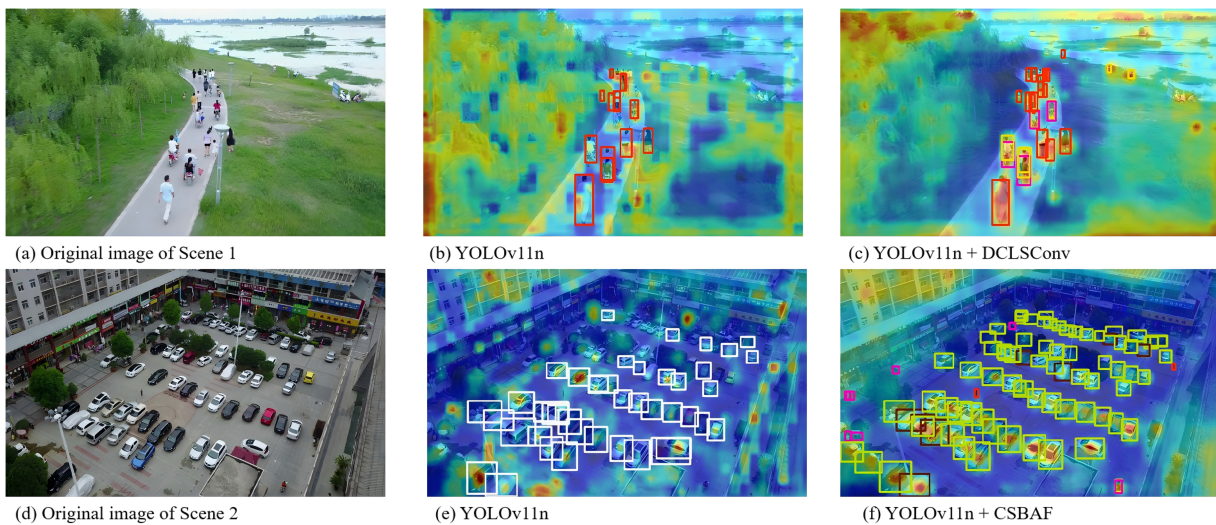


Figure 7: Heatmap visualization of feature responses after introducing DCLSCConv and CSBAF.

As shown in Table 7, the proposed model consistently outperforms the YOLOv11n baseline over three independent runs. The relatively small standard deviations indicate that the performance improvement is stable and not caused by random initialization. These results further verify the statistical robustness of the proposed method.

Table 7: Statistical robustness analysis over three runs on VisDrone2019.

Model	Seed	mAP@0.5	mAP@0.5:0.95
YOLOv11n	0	0.322	0.186
YOLOv11n	1	0.325	0.188
YOLOv11n	2	0.324	0.187
YOLOv11n	Mean $\pm$ Std	0.324 $\pm$ 0.002	0.187 $\pm$ 0.001
Ours	0	0.373	0.229
Ours	1	0.376	0.231
Ours	2	0.375	0.230
Ours	Mean $\pm$ Std	0.375 $\pm$ 0.002	0.230 $\pm$ 0.001

3.7 Visual Comparison of Detection Results

The qualitative detection results are shown in Fig. 8.

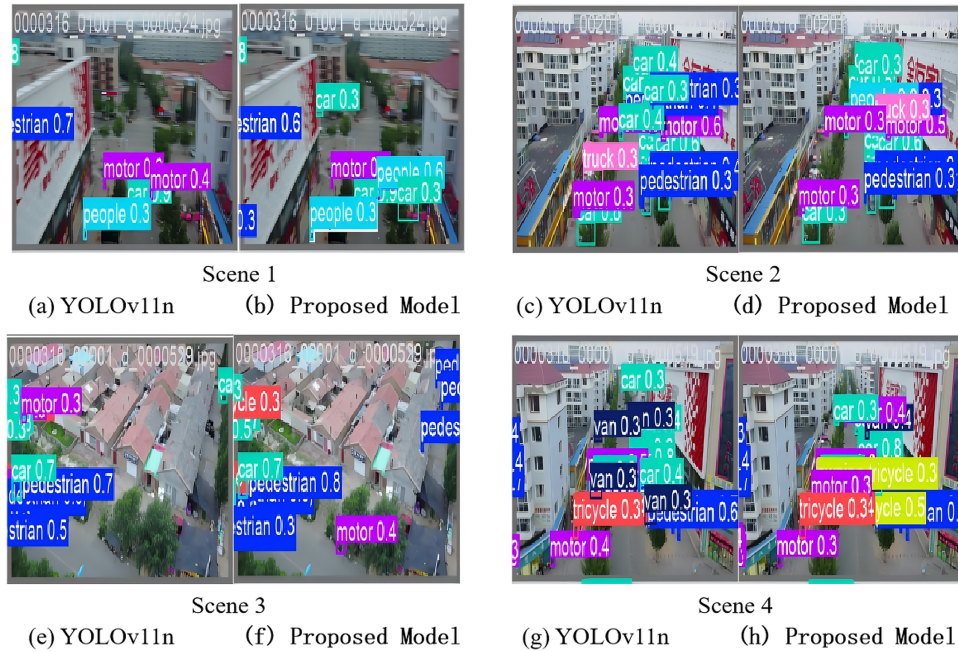


Figure 8: Qualitative comparison between YOLOv11n and the proposed model on representative UAV aerial scenes. Scenes 1–4 correspond to distant small-object detection, dense street-object detection, occluded rural-background detection, and multi-category small-object detection, respectively.

Across the four representative UAV aerial scenes, the proposed model shows clearer advantages over YOLOv11n. In distant-object scenes, the proposed model recovers more low-resolution targets that are missed by the baseline. In dense street scenes, it produces more complete predictions for clustered vehicles and pedestrians. In occluded rural-background scenes, the model maintains better localization continuity

for partially visible objects. In multi-category traffic scenes, the predictions are more stable across targets with different sizes and appearances. These observations suggest that the proposed modules improve the use of fine spatial details and multi-scale semantic cues, thereby reducing missed detections in crowded and complex UAV images.

3.8 Cross-Dataset Generalization Evaluation

To examine whether the proposed modules remain effective beyond UAV-specific VisDrone scenes, the trained framework was further evaluated on PASCAL VOC2012 [27] and BDD100K [28]. The corresponding results are reported in Table 8.

Table 8: Cross-dataset generalization results on PASCAL VOC2012 and BDD100K.

Datasets	Methods	P	R	mAP@0.5	mAP@0.5:0.95
PASCAL VOC2012	YOLOv11n	0.781	0.659	0.74	0.539
PASCAL VOC2012	Ours	0.791	0.681	0.755	0.555
BDD100K	YOLOv11n	0.58	0.412	0.441	0.292
BDD100K	Ours	0.592	0.43	0.453	0.304

As shown in Table 8, the proposed model achieves consistent improvements over YOLOv11n on both datasets. On PASCAL VOC2012, the proposed model improves P, R, mAP@0.5, and mAP@0.5:0.95 by 0.010, 0.022, 0.015, and 0.016, respectively. On BDD100K, the corresponding improvements are 0.012, 0.018, 0.012, and 0.012, respectively. These results indicate that the proposed method can maintain stable performance gains across different detection scenarios. Although the improvements on PASCAL VOC2012 and BDD100K are moderate, they suggest that the proposed modules have a certain degree of generalization ability beyond the VisDrone2019 dataset.

4 Conclusions

This study presents a YOLOv11n-based detector for UAV aerial small-object detection under edge-computing constraints. The method focuses on three stages of the detection pipeline. First, DCLSCov is embedded into the C3k2 structure to form C3k2-DC, which combines dual large-kernel contextual perception with small-kernel dynamic aggregation. This design helps the backbone retain local small-object details while expanding contextual perception. Second, CSBAF is introduced into the neck network to perform gated bidirectional interaction between low-level spatial features and high-level semantic features. Compared with direct upsampling and concatenation, this fusion strategy is more selective and better preserves useful small-object information. Third, UAV-TAD²H is designed for task-specific prediction by combining shared convolution, task decomposition, dynamic regression adjustment, and spatial attention enhancement.

Experiments on VisDrone2019 verify the effectiveness of the proposed method. Compared with YOLOv11n, the proposed model improves mAP@0.5 from 0.324 to 0.375 and mAP@0.5:0.95 from 0.187 to 0.230. Although the computation increases to 12.3 GFLOPs, the model remains compact with 3.12M parameters and achieves 33 FPS on Jetson Orin NX with TensorRT FP16 inference. The ablation results further show that each module contributes to the final performance gain, while the heatmap and qualitative detection results indicate improved attention to dense and small-object regions. Additional experiments on PASCAL VOC2012 and BDD100K show moderate but consistent improvements, suggesting that the proposed modules are not limited to a single dataset.

The current work mainly evaluates the model under fixed input resolution and limited edge-device settings. Future work will extend the evaluation to more UAV platforms, different power modes, and more diverse aerial scenes, and will further explore model compression strategies for practical onboard deployment.

Acknowledgement: Not applicable.

Funding Statement: This research was funded by Natural Science Foundation of Inner Mongolia Autonomous Region of China, grant number 2023LHMS06016.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Chanchan Zhao and Xiaoyu Gao; methodology, Xiaoyu Gao; software, Xiaoyu Gao; validation, Xiaoyu Gao and Ziyang Zhang; formal analysis, Xiaoyu Gao; investigation, Xiaoyu Gao; resources, Xiaoyu Gao; data curation, Xiaoyu Gao; writing—original draft preparation, Xiaoyu Gao; writing—review and editing, Xiaoyu Gao and Chanchan Zhao; visualization, Xiaoyu Gao; supervision, Chanchan Zhao and Bao Shi; project administration, Chanchan Zhao and Bao Shi; funding acquisition, Chanchan Zhao. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used in this study are publicly available. The VisDrone2019 dataset is available at <https://opendatalab.org.cn/OpenDataLab/VisDrone>. PASCAL VOC2012 and BDD100K are also publicly available from their official dataset websites.

Ethics Approval: Not applicable.

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

References

1. Li Q, Kao YY, Zhang Y, Xu P. Review on object detection in UAV aerial images. *J Graph*. 2024;45(6):1145–64. doi:10.11996/JG.j.2095-302X.2024061145.
2. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 2016 Jun 27–30; Las Vegas, NV, USA. p. 779–88. doi:10.1109/CVPR.2016.91.
3. Redmon J, Farhadi A. YOLOv3: an incremental improvement. *arXiv:1804.02767*. 2018.
4. Wang A, Chen H, Liu L, Chen K, Lin Z, Han J, et al. YOLOv10: real-time end-to-end object detection. In: *Advances in Neural Information Processing Systems 37*; 2024 Dec 10–15; Vancouver, BC, Canada. p. 107984–8011. doi:10.52202/079017-3429.
5. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot MultiBox detector. In: *Computer vision—ECCV 2016*. Cham, Switzerland: Springer International Publishing; 2016. p. 21–37. doi:10.1007/978-3-319-46448-0_2.
6. Tan M, Pang R, Le QV. EfficientDet: scalable and efficient object detection. In: *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; 2020 Jun 13–19; Seattle, WA, USA. p. 10778–87. doi:10.1109/cvpr42600.2020.01079.
7. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: *Computer vision—ECCV 2020*. Cham, Switzerland: Springer International Publishing; 2020. p. 213–29. doi:10.1007/978-3-030-58452-8_13.
8. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*; 2014 Jun 23–28; Columbus, OH, USA. p. 580–7. doi:10.1109/CVPR.2014.81.
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(6):1137–49. doi:10.1109/TPAMI.2016.2577031.

10. He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22–29; Venice, Italy. p. 2980–8. doi:10.1109/iccv.2017.322.
11. Jiang W, Wang WH, Yang JJ. AEM-YOLOv8s: small target detection algorithm for UAV aerial images. *Comput Eng Appl.* 2024;60(17):191–202. (In Chinese). doi:10.3778/j.issn.1002-8331.2403-0256.
12. Wang H, Zhang Y, Zhu C. DAFPN-YOLO: an improved UAV-based object detection algorithm based on YOLOv8s. *Comput Mater Contin.* 2025;83(2):1929–49. doi:10.32604/cmc.2025.061363.
13. Chen D, Qin Z, Zhang J, Wang H. MFF-YOLO: a target detection algorithm for UAV aerial photography. *Comput Mater Contin.* 2026;86(2):1–17. doi:10.32604/cmc.2025.072494.
14. Liu H, Tan F, Jin Y. Dual-YOLO: dual-path UAV aerial image target detection algorithm. *J Supercomput.* 2025;81(7):809. doi:10.1007/s11227-025-07290-y.
15. Xu S, Cui K, Chen G, Zhang M. SGA-YOLOv8: sobel-GELAN and AIFI-dilated-attention fusion network for UAV remote-sensing small target detection. *J Supercomput.* 2025;81(14):1291. doi:10.1007/s11227-025-07790-x.
16. Li B, Li SL. Improved YOLOv11n small object detection algorithm in UAV view. *Comput Eng Appl.* 2025;61(7):96–104. (In Chinese). doi:10.3778/j.issn.1002-8331.2411-0072.
17. Lu Y, Sun M. Lightweight multidimensional feature enhancement algorithm LPS-YOLO for UAV remote sensing target detection. *Sci Rep.* 2025;15(1):1340. doi:10.1038/s41598-025-85488-z.
18. Li S, Chen C. MFA-YOLO: a multi-feature aggregation approach for small-object detection method in drone imagery. *Sci Rep.* 2026;16(1):2484. doi:10.1038/s41598-025-32247-9.
19. Chen Z, Zhang Y, Xing S. YOLO-LE: a lightweight and efficient UAV aerial image target detection model. *Comput Mater Contin.* 2025;84(1):1787–803. doi:10.32604/cmc.2025.065238.
20. Gao P, Li Z. YOLO-S3DT: a small target detection model for UAV images based on YOLOv8. *Comput Mater Contin.* 2025;82(3):4555–72. doi:10.32604/cmc.2025.060873.
21. Tu YZ, Wang FX, Wu CL. Lightweight UAV aerial small object detection model integrating multi-attention mechanisms. *Comput Eng Appl.* 2025;61(11):93–104. (In Chinese). doi:10.3778/j.issn.1002-8331.2412-0324.
22. Ma F, Zhang R, Zhu B, Yang X. A lightweight UAV target detection algorithm based on improved YOLOv8s model. *Sci Rep.* 2025;15(1):15352. doi:10.1038/s41598-025-00341-7.
23. Xue B, Zhang B, Cheng Q. Experiment study on UAV target detection algorithm based on YOLOv8n-ACW. *Sci Rep.* 2025;15(1):11352. doi:10.1038/s41598-025-91394-1.
24. Zheng B, Angkawisittpan N, Huang L, Sonasang S. RSP-YOLOv11n multi-module optimized algorithm for insulator defect detection in UAV images. *Sci Rep.* 2025;15(1):35426. doi:10.1038/s41598-025-19059-7.
25. Wang A, Chen H, Lin Z, Han J, Ding G. LSNet: see large, focus small. *arXiv:2503.23135.* 2025.
26. Cao Y, He Z, Wang L, Wang W, Yuan Y, Zhang D, et al. VisDrone-DET2021: the vision meets drone object detection challenge results. In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW); 2021 Oct 11–17; Montreal, BC, Canada. p. 2847–54. doi:10.1109/iccvw54120.2021.00319.
27. Everingham M, Ali Eslami SM, Van Gool L, Williams CKI, Winn J, Zisserman A. The pascal visual object classes challenge: a retrospective. *Int J Comput Vis.* 2015;111(1):98–136. doi:10.1007/s11263-014-0733-5.
28. Yu F, Chen H, Wang X, Xian W, Chen Y, Liu F, et al. BDD100K: a diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 2633–42. doi:10.1109/cvpr42600.2020.00271.
29. Zhang H, Zhang H, Liu K, Gan Z, Zhu GN. UAV-DETR: efficient end-to-end object detection for unmanned aerial vehicle imagery. In: Proceedings of the 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2025 Oct 19–25; Hangzhou, China. p. 15143–9. doi:10.1109/IROS60139.2025.11246176.