

Article

LUIM-YOLO: A Lightweight and Efficient Detection Model for UAV Images

Junjie Li , Yisheng Wang  and Bo Zhang * 

School of Foreign Languages, National University of Defense Technology, Nanjin 210039, China; 13260846578@163.com (J.L.); m18050965098@163.com (Y.W.)

* Correspondence: zb100403@126.com; Tel.: +86-132-6084-6578

Featured Application

This work presents a parameter-efficient object detection model optimized for UAV images. It improves small-object detection accuracy with a substantially reduced parameter count and provides a candidate architecture for edge-oriented UAV perception.

Abstract

Unmanned Aerial Vehicle (UAV)-based small object detection is a challenging computer vision task. It is constrained by two primary factors: UAV platforms have limited onboard computational resources, and high-altitude objects often have weak features that are easily overwhelmed by complex backgrounds. To address these challenges, we propose LUIM-YOLO. First, a Lightweight Multi-Scale Feature Enhancement (LMSFE) module integrates parallel multi-scale convolutions with attention to strengthen small and low-contrast object feature extraction. Second, an Adaptive Multi-Scale Bottleneck (AMSB) module enhances key semantic features of small objects and spatial correlation of medium-scale objects. Third, an Enhanced Cross-layer Compensation Feature Pyramid Network (ECC-FPN) constructs cross-level interaction pathways to improve small object position and scale perception. Experimental results on VisDrone2019 show that compared with YOLOv8n, LUIM-YOLO reduces parameters by 57% and improves mAP@50 by 12.9%. Additional full-validation-set PyTorch inference tests on NVIDIA Jetson Orin show that LUIM-YOLO achieves 88.19 ms/image in FP32, indicating a parameter-efficient accuracy-oriented design with edge deployment potential.

Keywords: Unmanned Aerial Vehicle (UAV); small object detection; YOLO; lightweight; feature pyramid network



Academic Editor: Rui Araújo

Received: 6 May 2026

Revised: 3 June 2026

Accepted: 16 June 2026

Published: 7 July 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.

1. Introduction

With the rapid development of Unmanned Aerial Vehicle (UAV) technology, UAVs have been widely applied in agricultural monitoring [1], public safety [2], and military reconnaissance [3,4]. However, the large volume of aerial imagery imposes increasingly stringent requirements for efficient and accurate visual information analysis. While deep learning has substantially advanced object detection in terms of both accuracy and efficiency [5–7], and many task-specific variants have been tailored to particular application settings [8–10], object detection from UAV imagery still poses considerable difficulties, especially when the targets are small. First, detection accuracy for small objects remains insufficient. Due to high flight altitudes and variable viewing angles of UAVs, a large proportion of objects in captured images are smaller than 32 pixels [11], resulting in extremely

weak feature representations that are easily suppressed by complex background clutter. Second, practical model deployment remains challenging. High-performance detection models generally involve substantial computational complexity and a large number of parameters, making it difficult to achieve onboard deployment and real-time inference on resource-constrained edge devices such as UAV platforms with limited computing power and energy capacity [12–14].

To address the above issues, a lightweight small object detection model, termed LUIM-YOLO, is proposed based on YOLOv8n. The model preserves essential multi-scale information while maintaining low computational cost through a lightweight contextual module. It incorporates multi-scale parallel convolutions and a dual-attention mechanism to accurately capture and enhance the feature representations of small objects. Meanwhile, a three-branch feature pyramid fusion strategy is introduced, which effectively integrates deep semantic features and improves localization accuracy. Consequently, the proposed model achieves a significant improvement in both detection performance and model efficiency.

(1) Lightweight Multi-Scale Feature Enhancement Module (LMSFE): The LMSFE module adopts a three-branch parallel architecture constructed using depthwise separable convolutions [15], which effectively captures features of small-scale objects while maintaining a lightweight design. It further embeds two complementary lightweight attention operations, one acting over channels and the other over spatial locations. The channel component re-weights the feature channels associated with the target in densely populated scenes, whereas the spatial component concentrates on the unoccluded parts of each object so as to limit interference caused by occlusion.

(2) Adaptive Multi-Scale Bottleneck (AMSB): The AMSB module dynamically reweights fused multi-scale features through a heterogeneous dual-branch architecture. It integrates an improved channel attention mechanism, and a residual connection is introduced to enhance gradient propagation. This design achieves a balance between preserving critical semantic information and expanding the receptive field at a low computational cost while also reducing background redundancy and emphasizing key target attributes. Consequently, it significantly improves semantic discriminability and detection performance for small objects in cluttered backgrounds.

(3) Enhanced Cross-layer Compensation Feature Pyramid Network (ECC-FPN): Within the neck network, the ECC-FPN not only fuses top-down semantic features from the Feature Pyramid Network (FPN) [16] and bottom-up detailed features from the Path Aggregation Network (PANet) [17] but also introduces additional feature branches directly from the backbone network. This design strengthens the model's ability to perceive positional and scale information of small objects, thereby improving localization accuracy within the neck network.

The collaborative integration of the aforementioned modules enables the proposed model to achieve an improved balance between accuracy and efficiency on the VisDrone2019 dataset. To be specific, the model decreases the parameter count to merely 1.36M, fewer than those in the base YOLOv8n model, while achieving an mAP@50 of 0.349. This clearly shows that the model possesses two great merits in both accurate detection and being a lightweight model.

The rest of this paper is structured as follows. Section 2 briefly introduces the relevant work in this area. Section 3 describes the LUIM-YOLO architecture in detail. Section 4 gives the evaluation experiments carried out on VisDrone2019 dataset. The paper finally concludes in Section 5.

2. Related Works

2.1. Advances in Object Detection

In object detection, many efforts have been made to improve accuracy while keeping model sizes and computational costs low. In general, current models for detection belong to one of the following: Transformer models, two-stage CNN models, and one-stage CNN models. All three models try to achieve an optimum combination in the sense mentioned above.

Models based on the Transformer idea include Detection Transformer (DETR) [18] and SF-DETR [19], which use the self-attention mechanism to enhance their representation learning abilities and detection accuracy. The downside of such models is their computational requirement. As a solution to this problem, Li et al. [20,21] presented an encoder–decoder framework with an approximate gradient method. Additionally, Yun et al. [22] used structure parameterization methods in order to cut down computation costs. Nonetheless, such attention-based models usually fail to detect small objects and show relatively slow convergence rates. More importantly, they usually have higher computational requirements for real-time applications, especially when running on embedded systems like UAVs.

Two-stage CNN models also face challenges in balancing detection accuracy with model complexity. This class of methods first generates candidate regions and subsequently performs classification and regression on these proposals. Representative models from the R-CNN family, such as Fast R-CNN [23] and Faster R-CNN [24], have established a strong foundation for high detection accuracy. Haghiabi et al. [25], based on VGG16, employed a global context embedding strategy and achieved a classification accuracy of 98% on SAR images. The introduction of the FPN and subsequent improvements by Kaiming He et al. [26] further strengthened this performance advantage. Although Sara Naseri Golestani et al. [27] reduced computational cost by designing a multi-scale parallel branch structure and sharing backbone feature maps, two-stage detectors are still hampered by their intricate pipelines and the costly region-proposal stage. Consequently, they tend to carry a large parameter budget and run inference comparatively slowly.

One-stage detectors, by contrast, fold localization and classification into a single forward pass and thereby reach a better compromise between compactness and accuracy. Members of the YOLO family, exemplified by YOLOv5 [28] and YOLOv7 [29], make dense predictions straight from the input image and so attain markedly higher inference efficiency. A wide range of dedicated refinements has since been proposed to push small-object accuracy higher without forfeiting this lightweight property.

With respect to backbone construction and feature extraction, Xu et al. [30] boosted the extraction capability of the network by embedding DenseNet, and Cao et al. [31] rebuilt the shallow feature-fusion backbone using GhostConv. Tang et al. [32] introduced GhostNet to achieve a better balance between detection accuracy and computational efficiency and further incorporated a single-stage headless context module to strengthen feature representation. Regarding lightweight module design and computational optimization, Sun et al. [33] developed a partially hybrid dilated convolution (PHDC) module to construct CSPPartialStage, which improved detection efficiency while effectively reducing computational overhead. In the domain of attention mechanisms and multimodal fusion, Chen et al. [34] integrated Ghost convolution with an efficient multi-scale attention module into YOLOv8, thereby improving detection speed while maintaining a comparable parameter scale. Peng et al. [35] further advanced this approach with CAFM-YOLOv5 by extracting visible and infrared features through parallel branches and integrating channel and spatial information using a cross-attention fusion module, enabling complementary enhancement of multimodal features. The problem with UAVs is that they lack sufficient computational resources and have low memory capacity and limited energy resources.

While some lightweight versions are trying to improve efficiency through aggressive compression of the network, in most cases, the process leads to loss of important data about features of objects, which disturbs the balance between efficiency and accuracy.

2.2. UAV Object Detection

However, using the general version of the YOLO algorithm as an aerial image recognition benchmark has been inefficient because of the limitations of the UAV system hardware in terms of computational capacity, along with the characteristics of the aerial image, including the tiny and high density of target objects, complex background, various scales, and interference. In order to tackle the above issues, previous studies have mainly attempted to explore three directions: lightweight architecture design, dynamic feature extraction, and environment robustness enhancement.

For lightweight architecture design and feature extraction optimization, Qu et al. [36] eliminated the large-object branch and incorporated Space-to-Depth Convolution (SPD-Conv) and GCU, resulting in a much more accurate detection rate of tiny objects with parameters reduced by up to 65%. Similarly, Xue et al. [37] applied RFConv-Block and a CHS-FPN structure, which greatly optimized the feature extraction process for dense tiny objects through fewer floating-point operations. Further efforts were made to implement extremely lightweight models by Wang et al. [38] and Zhou et al. [39], who designed an ultra-light C2f architecture along with a VMambaNeck architecture as well as AZML and ADown structures.

Regarding the aspect of dynamic multi-scale feature fusion, Wang et al. [40] contributed to the field with the introduction of the Cross-Scale Feature Pyramid (CS-FPN) in combination with a Feature Recalibration Module (FRM). This combination helped minimize the localization errors associated with multi-scale feature fusion. Following this direction, Bi et al. [41] and Li et al. [42] boosted pyramid aggregation with the help of SPC-FPN and RepGFPN in conjunction with DyHead-DCNv4. For the purpose of bridging the gap between the semantic and spatial aspects of information at deep and shallow levels, Hou et al. [43] came up with an Efficient Path Aggregation Network (E-PAN). Meanwhile, Wang et al. [44] developed a strategy for effective vehicle detection by enhancing the feature representation in the ELA-FPN framework.

As far as the development of robustness in complex scenarios is concerned, two different approaches have been studied. First, the improvement of the attention mechanism is widely used. Hwang et al. [45], by incorporating the GAM, and Elhenidy et al. [46], by using the Ghost mechanism, boosted the model's robustness to interference by implementing GAM and Ghost mechanisms in an occluded and cluttered background. Akdoğan et al. [47] also developed a highly robust model that was able to generalize well in poor imaging scenarios through attention mechanisms and image processing. Second, robustness has also been obtained through the optimization of the loss function. In their research, Hou et al. [48] presented the α -SIoU loss function that combines the precision in localization of small targets with efficiency. Similarly, Zhang et al. [49] proposed a dynamic WIoU loss function for accurate bounding box regression.

In conclusion, despite the considerable achievements made in the aforementioned areas by previous works, certain shortcomings still exist. The feature extraction ability of some ultra-lightweight architectures is not strong enough when dealing with small objects against complex backgrounds. In contrast, the high-precision architecture fails to satisfy the real-time demands of UAVs. Therefore, effectively overcoming the bottleneck of small object detection in complex environments while maintaining low computational overhead remains a critical challenge in current research.

3. Method

3.1. Proposed UAV Object Detection Model

Because of their redundant parameters and heavy storage and memory footprint, conventional detection models are hard to deploy on UAV platforms. In addition, these models often exhibit limited capability in extracting discriminative features for small objects in UAV aerial imagery.

To address these limitations, a lightweight and efficient multi-scale enhanced detection model, termed LUIM-YOLO, is proposed. Built upon YOLOv8 and depicted in Figure 1, the overall architecture comprises three principal parts: the LMSFE module, the Adaptive Multi-Scale Bottleneck (AMSB) feature enhancement module, and the ECC-FPN.

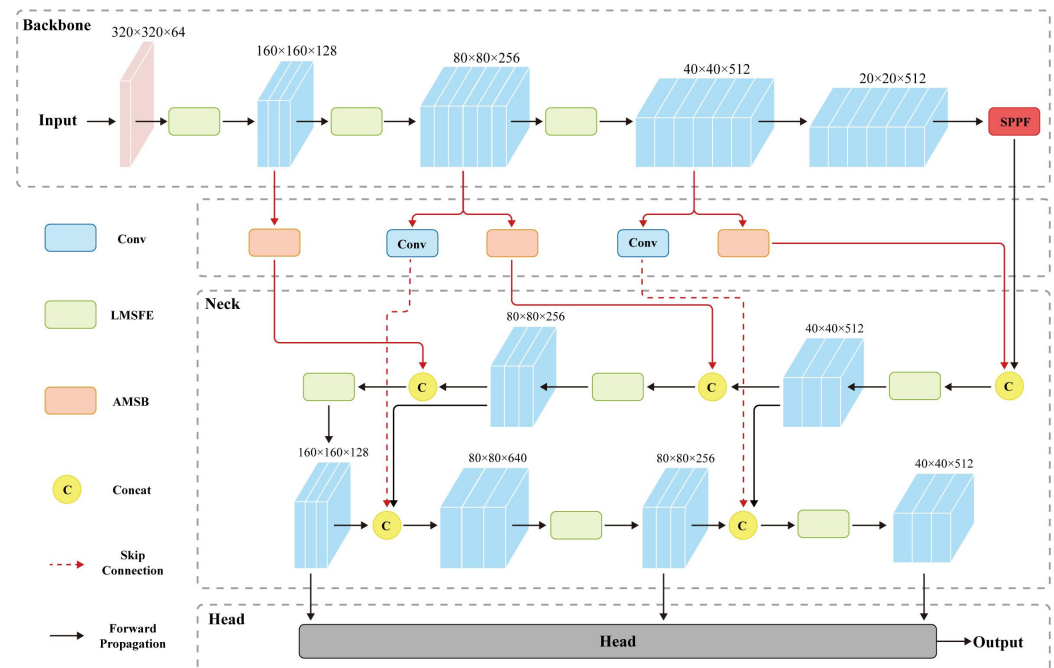


Figure 1. Architecture of the proposed LUIM-YOLO model.

3.2. LMSFE Module

Even though the current C2f module [50] provides a reasonable trade-off between the two parameters (accuracy and efficiency), there are some significant issues in the detection of small objects. First, due to the constant size of the receptive field, the same convolution operation is unable to capture both the fine-grained details of small objects and the context information of the large objects, hence affecting the module's ability to capture small objects [51]. Secondly, the process of feature fusion in the model heavily depends on channel concatenation without a specific method of handling densely packed objects to prevent feature aliasing [52,53].

To compensate for these limitations, we propose the Lightweight Multi-Scale Feature Enhancement (LMSFE) module. The essence of the approach is a Multi-Scale Convolutional Structure (MSCS) bottleneck, which allows expanding the receptive field during the feature modeling process and enhancing the detailed representation of tiny objects. In the rest of the paper, LMSFE is understood as an overall enhancement block, while MSCS bottleneck refers to the bottleneck block of the LMSFE module.

As can be seen from Figure 2, the MSCS bottleneck comprises three basic blocks, namely multi-scale parallel feature extraction, channel-spatial attention calibration and gradient stable propagation path. The feature extraction block is designed using a heterogeneous structure of three parallel branches, namely a 1×1 pointwise convolution

branch, a 3×3 depthwise separable convolution branch and a 5×5 depthwise separable convolution branch. By using different kernel sizes in each parallel branch, the model obtains multi-level receptive fields, enabling it to focus on tiny objects with small kernels while capturing the contextual and semantic information of larger objects with large kernels. Simultaneously, applying depthwise separable convolution decreases the computation cost and number of parameters of the network.

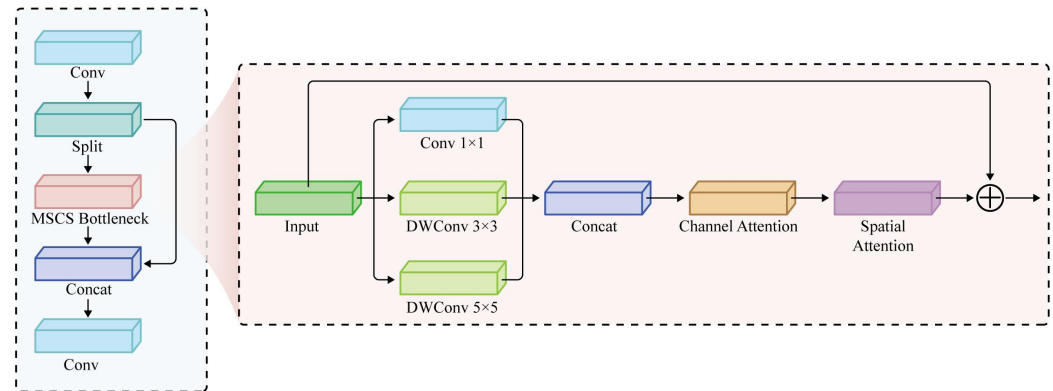


Figure 2. Structure of the LMSFE module.

In order to further increase the discrimination and robustness of the feature representation, a channel–spatial-attention method is proposed in the module. Specifically, the channel attention part automatically generates weights using the global spatial context to highlight important features and filter out irrelevant ones. On the other hand, the spatial attention part uses the average pooling descriptor and max-pooling descriptor to remove interference caused by occlusions and backgrounds.

Furthermore, the module uses conditional residual connections, meaning that the fusion of residual connections happens only if there is alignment between input and output feature dimensions. This prevents the issue of having redundant features due to dimensionality differences while at the same time ensuring the smooth gradient propagation in the deep model.

The calculation steps for the proposed bottleneck module named MSCS are as follows. The input feature tensor is set as $\mathbf{X} \in \mathbb{R}^{N \times C_{in} \times H \times W}$, where N is the batch size, C_{in} is the number of input channels, and H and W denote the spatial resolution. In order not to reduce the representational capacity because of the small channel width, the number of output channels in the case of three-branch concatenation is set as

$$C_{mid} = \max(\lfloor C_{out} \cdot e \rfloor, 8). \quad (1)$$

The three parallel branches are then computed as

$$S(\mathbf{X}) = \text{Conv}_{1 \times 1} \left(\mathbf{X}; C_o = \left\lfloor \frac{C_{mid}}{3} \right\rfloor \right), \quad (2)$$

$$G(\mathbf{X}) = \text{Conv}_{1 \times 1} \left(\text{DWConv}_{3 \times 3}(\mathbf{X}); C_o = \left\lfloor \frac{C_{mid}}{3} \right\rfloor \right), \quad (3)$$

$$T(\mathbf{X}) = \text{Conv}_{1 \times 1} \left(\text{DWConv}_{5 \times 5}(\mathbf{X}); C_o = C_{mid} - 2 \left\lfloor \frac{C_{mid}}{3} \right\rfloor \right). \quad (4)$$

The channel dimensions of $S(\mathbf{X})$, $G(\mathbf{X})$, and $T(\mathbf{X})$ therefore sum exactly to C_{mid} . The branch outputs are concatenated along the channel dimension:

$$\mathbf{Y}_{multi} = \text{Cat}(S(\mathbf{X}), G(\mathbf{X}), T(\mathbf{X})), \quad \mathbf{Y}_{multi} \in \mathbb{R}^{N \times C_{mid} \times H \times W}. \quad (5)$$

As for the problem of the aliasing effect due to the high density of objects, the network implements a channel attention technique, in which the weights of different feature channels are obtained adaptively by utilizing the spatial information, leading to the improvement and downweighting of relevant channels. First, global adaptive average pooling is applied to the multi-scale feature map:

$$\mathbf{Z} = \text{AdaptiveAvgPool2d}(\mathbf{Y}_{\text{multi}}, 1), \quad \mathbf{Z} \in \mathbb{R}^{N \times C_{\text{mid}} \times 1 \times 1}. \quad (6)$$

Channel attention weights are subsequently obtained by feeding this descriptor through two stacked 1×1 convolutions together with a nonlinear activation:

$$M_c = \text{Sigmoid}(\text{Conv}_{1 \times 1}(\text{ReLU}(\text{Conv}_{1 \times 1}(\mathbf{Z})))), \quad M_c \in \mathbb{R}^{N \times C_{\text{mid}} \times 1 \times 1}. \quad (7)$$

The computed channel weights are applied to the multi-scale feature map by broadcast element-wise multiplication:

$$\mathbf{Y}_{\text{channel}} = \mathbf{Y}_{\text{multi}} \odot M_c. \quad (8)$$

To reinforce the visible portions of a target and damp distracting responses from occlusion or background clutter, a spatial attention module is applied. It first computes channel-wise average pooling and max pooling and concatenates the two descriptors:

$$\mathbf{Y}_{\text{spatial_in}} = \text{Cat}(\text{Avg}_c(\mathbf{Y}_{\text{channel}}), \text{Max}_c(\mathbf{Y}_{\text{channel}})), \quad \mathbf{Y}_{\text{spatial_in}} \in \mathbb{R}^{N \times 2 \times H \times W}. \quad (9)$$

Spatial attention weights are then obtained by a 7×7 convolution followed by a Sigmoid function:

$$M_s = \text{Sigmoid}(\text{Conv}_{7 \times 7}(\mathbf{Y}_{\text{spatial_in}}; C_o = 1)), \quad M_s \in \mathbb{R}^{N \times 1 \times H \times W}. \quad (10)$$

The spatially refined feature map and final projection are defined as

$$\mathbf{Y}_{\text{spatial}} = \mathbf{Y}_{\text{channel}} \odot M_s, \quad (11)$$

$$H(\mathbf{Y}_{\text{spatial}}) = \text{Conv}_{3 \times 3}(\mathbf{Y}_{\text{spatial}}; C_o = C_{\text{out}}, g = g). \quad (12)$$

Finally, a conditional residual connection is used when the input and output dimensions are consistent:

$$\mathbf{Y}_{\text{out}} = \begin{cases} \mathbf{X} + H(\mathbf{Y}_{\text{spatial}}), & \text{if } C_{\text{in}} = C_{\text{out}} \text{ and residual connection is enabled,} \\ H(\mathbf{Y}_{\text{spatial}}), & \text{otherwise.} \end{cases} \quad (13)$$

The key symbols and parameter definitions used in the MSCS bottleneck are summarized in Table 1.

Table 1. Core symbols and parameter definitions of the MSCS bottleneck.

Category	Symbol	Definition
Tensor and dimension	$\mathbf{X}$	Input tensor in $\mathbb{R}^{N \times C_{\text{in}} \times H \times W}$
	$\mathbf{Y}_{\text{multi}}$	Concatenated multi-scale feature map in $\mathbb{R}^{N \times C_{\text{mid}} \times H \times W}$
	$\mathbf{Y}_{\text{out}}$	Final output feature tensor
	N	Batch size
	$C_{\text{in}}, C_{\text{out}}$	Input and output channel numbers
	C_{mid}	Intermediate channel number after three-branch concatenation
	H, W	Height and width of the feature map

Table 1. Cont.

Category	Symbol	Definition
Branch features	$S(\mathbf{X})$	1×1 pointwise convolution branch
	$G(\mathbf{X})$	3×3 depthwise separable convolution branch
	$T(\mathbf{X})$	5×5 depthwise separable convolution branch
Attention and operators	M_c	Channel attention weight
	M_s	Spatial attention weight
	$\text{Conv}_{k \times k}(\cdot)$	Convolution with kernel size k
	$\text{DWConv}_{k \times k}(\cdot)$	Depthwise convolution with kernel size k
	$\text{Cat}(\cdot)$	Concatenation along the channel dimension
	$\text{Avg}_c(\cdot), \text{Max}_c(\cdot)$	Average pooling and max pooling along the channel dimension
	$\odot$	Broadcast element-wise multiplication
	$\lfloor \cdot \rfloor$	Floor operation, which rounds a real value down to the nearest integer
	e	Channel expansion factor
	g	Number of groups in the final convolution

3.3. AMSB Feature Enhancement Module

In the shallow maps emitted by the YOLOv8n backbone, the fine features of small objects are commonly intertwined with background noise, while the contextual cues of medium-sized objects stay poorly defined. The essential semantics of small objects may therefore be diluted during fusion, which holds back detection accuracy in cluttered aerial scenes.

To overcome these issues, we introduce an Adaptive Multi-Scale Bottleneck (AMSB) feature enhancement module that reorganizes the semantic structure of the features and improves discriminability across classes. Figure 3 depicts the overall layout of the proposed AMSB module.

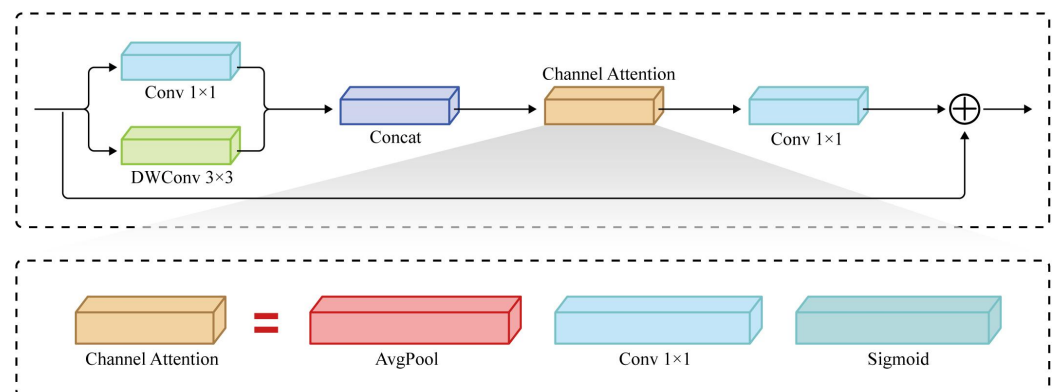


Figure 3. Structure of the AMSB module.

AMSB is organized as a lightweight, heterogeneous pair of branches. The first applies a 1×1 pointwise convolution that carries out an adaptive channel-wise mapping and trims feature redundancy, while the second uses a 3×3 depthwise separable convolution to widen the local receptive field at low cost. Concatenating the two branch outputs along the channel axis yields a fused descriptor that blends fine detail with local context.

In order to tackle the problem of feature redundancy and semantic ambiguity in the channel dimension further, AMSB proposes a light non-compressing channel attention mechanism. The light non-compressing channel attention mechanism adopts global adaptive average pooling to aggregate global spatial information and utilizes a single 1×1 convolution layer to infer the importance of channels. The output channel weights of the 1×1 convolution layer are normalized with a Sigmoid activation function and multiplied

channel-wise into the fused features. The structure of the lightweight channel attention unit is illustrated in Figure 4.

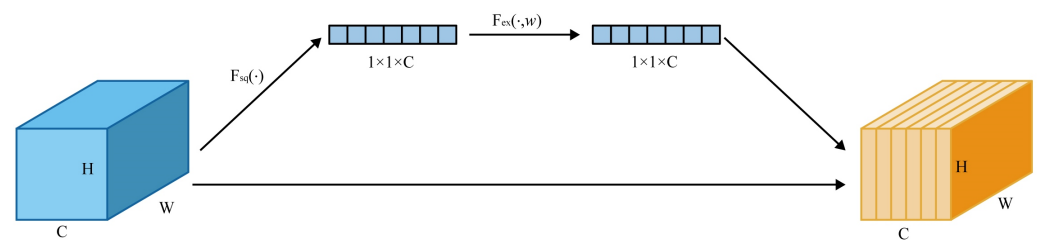


Figure 4. Structure of the lightweight non-compressing channel attention unit.

In the attention-based refinement module, the attention-weighted features are further processed with a 1×1 pointwise convolution layer. It helps in conducting structured channel fusion and projecting aggregated features into the predefined output channel space. In the case the dimension of the input matches that of the output, a residual connection is added to the structure of the refinement module. Such a strategy helps stabilize the gradient backpropagation process and avoid feature degradation during propagation.

3.4. ECC-FPN Feature Pyramid Network

Feature semantic degeneration and spatial detail degradation have become unavoidable factors due to the deepening of the model in neural networks, especially in complex UAV aerial imaging.

Shallow network layers and deep network layers differ greatly in feature extraction. The shallow layers have high-resolution images that contain small details like the shape of objects and location information but are affected by high levels of noise [16]. However, the deep layers can detect high-level semantics and global contextual information, though they lack spatial details.

The neck network in YOLOv8n uses a PAN-FPN architecture, which involves bottom-up augmentation in addition to the regular feature pyramid architecture of the neck network of YOLOv8n. Though this helps improve feature extraction somewhat, it might increase feature degeneration and misalignment across different layers. Therefore, it cannot meet the requirements of detecting small objects in aerial imaging precisely.

To overcome the aforementioned drawback, we present ECC-FPN, which facilitates cross-layer feature compensation between the backbone and feature pyramid networks. Essentially, the key point is to build two-way feature communication between the backbone network and feature pyramid. For instance, features extracted via the proposed AMSB are injected into the corresponding layers of the feature pyramid, thus realizing effective feature interaction and fusion between spatial details at a finer scale and semantics from a higher perspective. When performing feature extraction, the enhanced features are first forwarded into the bottom-up pathway, during which more efficient interaction between spatial and channel-wise semantic information is achieved based on the LMSFE operation. Finally, these features are sent across dimensions to be processed in the top-down pathway of the feature pyramid, guaranteeing that low-level and high-level feature representations are adequately complementary in all layers. This greatly helps mitigate issues associated with information leakage when forwarding features between layers, in addition to reducing the gap between semantics and spatial features within a single layer structure.

The ECC-FPN structure is illustrated in Figure 5. It establishes precise connections between the AMSB refined features and the corresponding levels of the bottom-up pathway. The LMSFE module is then employed to perform fine-grained multi-scale feature extraction and inter-feature relationship modeling. Subsequently, the enhanced features

are propagated across layers into the top-down pyramid pathway, enabling bidirectional interaction among features at all hierarchical levels.

Due to their relatively small receptive fields, low-level feature maps preserve more precise spatial information, including object localization cues and edge details. When reintroduced into the feature pyramid fusion framework, these representations effectively compensate for the limited spatial precision of high-level features. Consequently, high-level representations with strong semantic abstraction capability are further enhanced with improved spatial localization accuracy.

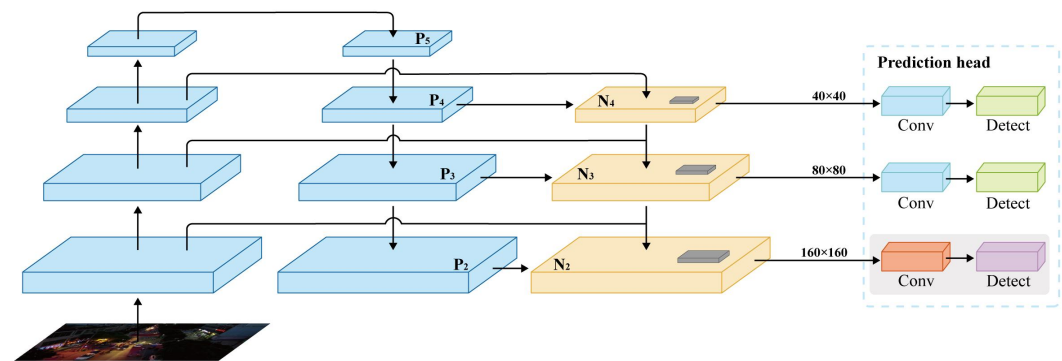


Figure 5. Structure of ECC-FPN.

To make the detector more appropriate for UAV imagery, its head is further tuned on the basis of ECC-FPN. Standard YOLOv8n predicts at three scales (20×20 , 40×40 , and 80×80), which easily miss densely packed small objects; we therefore add a 160×160 high-resolution head, produced through convolution and upsampling, to sharpen sensitivity to ultra-small objects, and we remove the 20×20 low-resolution head. This adjustment cuts model complexity, raises efficiency, and further improves small-object detection in aerial environments.

4. Experimental Results

4.1. Datasets and Metrics

For validating the efficiency of LUIM-YOLO in detecting small objects using UAVs, the VisDrone2019 [54] will be employed as the main performance metric.

VisDrone2019 dataset is a globally renowned object detection benchmark developed for UAV aerial images by the researchers at Tianjin University and its partnering organizations. The dataset consists of over 10 diverse scene types gathered from 28 different Chinese cities, namely urban roads, rural scenes, industrial scenes, and commercial scenes. Overall, the VisDrone2019 dataset is made up of 10,209 images, out of which there are 6471 images for training, 1610 images for validation, and 2128 images for testing. Furthermore, the annotations consist of 10 common objects, including pedestrians, vehicles, bicycles, and drones. It further features typical UAV scenes involving multi-scale objects, high object densities, and heavy occlusions.

The suggested approach is then assessed on the grounds of accuracy and complexity. As is customary for object detection tasks, Precision, Recall, Average Precision (AP), and Mean Average Precision (mAP) are chosen to be the primary accuracy criteria:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (14)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (15)$$

$$AP_{\text{integral}} = \int_0^1 \text{Precision}(R) dR \quad (16)$$

$$mAP = \frac{1}{C} \sum_{c=1}^C AP_c \quad (17)$$

where TP , FP , and FN represent the numbers of true positives, false positives, and false negatives, respectively, and C denotes the number of object categories.

4.2. Experimental Settings and Implementation Details

All experiments were conducted on the VisDrone2019-DET dataset using the official training and validation splits. To keep the comparison fair, every YOLOv8-based ablation model used the same training configuration. Except where noted, each model was trained a single time with deterministic training enabled and the random seed pinned to 0. The detailed software environment, optimization strategy, data augmentation settings, and validation configuration are summarized in Table 2.

Table 2. Experimental environment and reproducibility settings.

Category	Configuration
Hardware and software environment	
Operating system	Windows 10.0
GPU	NVIDIA RTX A6000, 48 GB memory, CUDA device 0
Python/PyTorch/CUDA	Python 3.12.7, PyTorch 2.7.1 + cu118, CUDA runtime 11.8
Detection framework	Ultralytics 8.3.227
Training configuration	
Dataset split	VisDrone2019-DET train set for training and validation set for evaluation
Epochs/batch size/image size	200/24/640 × 640
Optimizer	AdamW
Initial learning rate/final LR factor	0.001/0.01
Learning-rate schedule	Cosine annealing enabled
Momentum/weight decay	0.95/0.0005
Warm-up settings	10 epochs, warm-up momentum 0.8, warm-up bias learning rate 0.1
Loss weights	box 7.5, cls 0.5, DFL 1.5
Pretrained weights	Not used; models were trained from scratch
Mixed precision/cache/workers	AMP enabled, cache disabled, 8 dataloader workers
Seed and run count	Seed 0, deterministic training enabled, one run per model unless otherwise stated
Data augmentation	
Mosaic/close mosaic	0.5/disabled during the last 15 epochs
MixUp/CutMix/Copy-Paste	0.15/0.0/0.0
HSV augmentation	$h = 0.02$, $s = 0.8$, $v = 0.5$
Geometric augmentation	translate 0.2, scale 0.5, horizontal flip 0.5, vertical flip 0.0
Rotation/shear/perspective	0.0/0.0/0.0
Random erasing	0.4
Validation and post-processing	
Validation split	VisDrone2019-DET validation set
Confidence threshold	Ultralytics default validation setting; not manually overridden
NMS IoU threshold/maximum detections	0.7/300
Class-agnostic NMS	Disabled
Evaluation metrics	Precision, recall, mAP@50, and mAP@50–95

4.3. Ablation Experiments

To evaluate the contribution of the three proposed modules to UAV-based object detection performance under consistent experimental settings, ablation studies were conducted on the VisDrone2019 dataset. YOLOv8n was selected as the baseline model, and the comparative results of different module configurations are summarized in Table 3.

Table 3. Ablation experiments on VisDrone2019.

Model	LMSFE	AMSB	ECC-FPN	mAP@50	mAP	Parameters (M)	FLOPs (G)
Baseline				0.309	0.179	3.16	8.9
Baseline + LMSFE	✓			0.295	0.167	2.88	7.5
Baseline + AMSB		✓		0.313	0.179	3.08	8.5
Baseline + ECC-FPN			✓	0.343	0.201	3.26	8.6
Baseline + LMSFE + ECC-FPN	✓		✓	0.330	0.191	1.28	10.5
Baseline + LMSFE + AMSB	✓	✓		0.339	0.199	1.32	10.8
Baseline + AMSB + ECC-FPN		✓	✓	0.348	0.206	1.50	11.8
Baseline + All (Ours)	✓	✓	✓	0.349	0.202	1.36	11.0

As shown in Table 3, the contribution of each component is not strictly additive, because the proposed modules are implemented as architectural replacements and channel reallocation strategies rather than simple plug-in layers inserted into an unchanged baseline. ECC-FPN provides the largest individual accuracy gain, increasing mAP@50 from 30.9% to 34.3%. This indicates that high-resolution feature compensation and the use of a small-object-oriented pyramid structure are particularly important for VisDrone2019.

The LMSFE-only variant obtains a lower mAP@50 than the baseline. This result suggests that local multi-scale enhancement alone is insufficient when it is not supported by an appropriate feature pyramid and semantic compensation pathway. In particular, replacing the original feature extraction blocks with a lightweight multi-branch structure reduces parameter count and FLOPs but may also weaken deep semantic aggregation and disturb the original balance between localization and classification features. Therefore, LMSFE is interpreted primarily as a lightweight local feature enhancement unit whose benefit becomes evident when combined with ECC-FPN or AMSB.

Baseline + ECC-FPN retains wider baseline feature extraction paths while modifying the neck structure, resulting in 3.26 M parameters. In contrast, configurations involving AMSB and LMSFE adopt more compact backbone and neck channel allocations and replace several standard convolutional bottlenecks with lightweight depthwise-separable or dual-branch structures. Consequently, the Baseline + AMSB + ECC-FPN variant reduces the parameter count to 1.50 M while increasing FLOPs, because the architecture preserves higher-resolution feature maps for small-object detection. By bringing LMSFE, AMSB, and ECC-FPN together, the complete model reaches an mAP@50 of 34.9% using just 1.36 M parameters. These figures indicate that the design favors a strong accuracy-to-parameter balance for UAV small-object detection.

4.4. Comparison Experiments

We carried out a set of comparative experiments on VisDrone2019 in order to assess the effectiveness of the proposed model for UAV-oriented object detection.

To ensure experimental fairness and result reliability, several representative models with publicly available code and trained on the same dataset were selected for comparison.

The comparison includes single-stage detectors such as [55], YOLOv5n, YOLOv9n [56], YOLOv10n [57], as well as several recently proposed state-of-the-art models. All methods are evaluated under consistent training and testing settings to ensure a fair comparison.

The evaluation results of LUIM-YOLO and other object detection models on the VisDrone2019 dataset are presented in Table 4.

LUIM-YOLO significantly reduces the number of parameters to 1.36 M, which is approximately 57% lower than the baseline YOLOv8n (3.16 M). Meanwhile, its parameter size is also substantially smaller than that of YOLOv9t (2.13 M), YOLOv11n (2.62 M), and YOLOv12n (2.60 M), making it the most compact model among all compared methods. This greatly reduces storage and deployment requirements for edge devices.

Table 4. Detection performance of different methods on the VisDrone2019 validation dataset.

Model	mAP@50	mAP	Parameters (M)	FLOPs (G)
SSD	0.242	0.127	24.95	138.78
YOLOv5n	0.299	0.172	2.65	7.84
YOLOv6n	0.273	0.157	4.50	13.03
YOLOv8n (Baseline)	0.309	0.179	3.16	8.9
YOLOv9t	0.297	0.170	2.13	8.48
YOLOv10n	0.277	0.160	2.78	8.74
YOLOv11n	0.287	0.164	2.62	6.61
YOLOv12n	0.276	0.156	2.60	6.65
LUIM-YOLO	0.349	0.202	1.36	11.0

Where accuracy is concerned, LUIM-YOLO attains the top scores among all compared models, with an mAP@50 of 34.9% and an mAP of 20.2%. Concretely, it lifts mAP@50 by about 12.9% over the YOLOv8n baseline and is also more accurate than YOLOv9t and YOLOv11n. Compared with the latest lightweight model YOLOv12n, LUIM-YOLO achieves both a smaller parameter size and higher detection accuracy.

Overall, comprehensive experimental results on the VisDrone2019 dataset demonstrate that the proposed LUIM-YOLO achieves a favorable trade-off between model size and detection accuracy. It shows clear advantages in parameter efficiency and small-object detection accuracy over mainstream lightweight object detection models, while its practical runtime behavior on edge hardware is further evaluated below.

4.5. Scale-Specific Evaluation

To further examine whether the accuracy gain is related to the small-object characteristics of UAV imagery, a COCO-style scale-specific evaluation was conducted on the VisDrone2019 validation set. The object area thresholds follow the COCO protocol, and the same confidence threshold, NMS IoU threshold, and maximum detection setting were used for both models. The results are summarized in Table 5.

Table 5. Scale-specific AP and AR on the VisDrone2019 validation set.

Model	AP _s	AP _m	AP _l	AR _s	AR _m	AR _l	mAP@50	mAP
YOLOv8n (Baseline)	0.0704	0.2410	0.3798	0.1255	0.3758	0.5264	0.2618	0.1525
LUIM-YOLO	0.1002	0.2602	0.3586	0.1848	0.4089	0.4643	0.3065	0.1768

As shown in Table 5, LUIM-YOLO improves AP_s from 0.0704 to 0.1002 and AR_s from 0.1255 to 0.1848. The medium-object AP and AR also increase from 0.2410 to 0.2602 and from 0.3758 to 0.4089, respectively. These results support the intended design goal of strengthening small- and medium-scale object representation in aerial scenes.

4.6. Edge-Device Latency Evaluation

To further evaluate the deployment behavior of the proposed model on edge hardware, an additional latency experiment was conducted on an NVIDIA Jetson Orin platform. The models were tested using PyTorch inference with an input size of 640 × 640 and a confidence threshold of 0.25. The complete VisDrone2019 validation set with 548 images was used for timing after warm-up. The complete VisDrone2019 validation set with 548 images was used for timing after warm-up. The reported latency includes end-to-end prediction overhead in the Ultralytics inference pipeline, and mean, median, P95 latency, and FPS are reported. The results are summarized in Table 6.

Table 6. PyTorch inference latency on NVIDIA Jetson Orin.

Model	Precision	Images	Parameters (M)	Mean (ms)	Median (ms)	P95 (ms)	FPS
YOLOv8n (Baseline)	FP32	548	3.01	62.11	60.36	104.14	16.10
YOLOv8n (Baseline)	FP16	548	3.01	62.32	60.48	104.23	16.05
LUIM-YOLO	FP32	548	1.36	88.19	85.77	136.60	11.34
LUIM-YOLO	FP16	548	1.36	89.65	86.62	138.69	11.16

The Jetson results show that LUIM-YOLO substantially reduces the number of parameters compared with YOLOv8n, but its PyTorch inference latency is higher. This behavior is consistent with the FLOP analysis in Tables 3 and 4: the use of high-resolution small-object features and additional feature interaction paths improves detection accuracy and parameter efficiency but increases runtime cost on edge hardware. Therefore, LUIM-YOLO is a parameter-efficient and accuracy-oriented UAV detector with edge deployment potential.

4.7. Qualitative Result Analysis

To further substantiate the model's behavior in realistic and complex settings, we performed experiments on VisDrone2019, whose objects vary widely in scale and a large fraction of which are extremely small.

A qualitative comparison between LUIM-YOLO and the baseline model (YOLOv8n) is presented in Figure 6.

In the figure, column (a) gives the YOLOv8n results and column (b) gives those of the proposed LUIM-YOLO. The juxtaposition shows that YOLOv8n overlooks a number of distant vehicle targets. In contrast, the proposed model achieves more accurate detection of such small and distant objects. Moreover, YOLOv8n shows clear limitations in densely populated scenes, where multiple detection omissions occur. In comparison, LUIM-YOLO is able to reliably identify individual pedestrian targets in crowded environments, demonstrating improved robustness and detection accuracy in complex aerial scenarios.

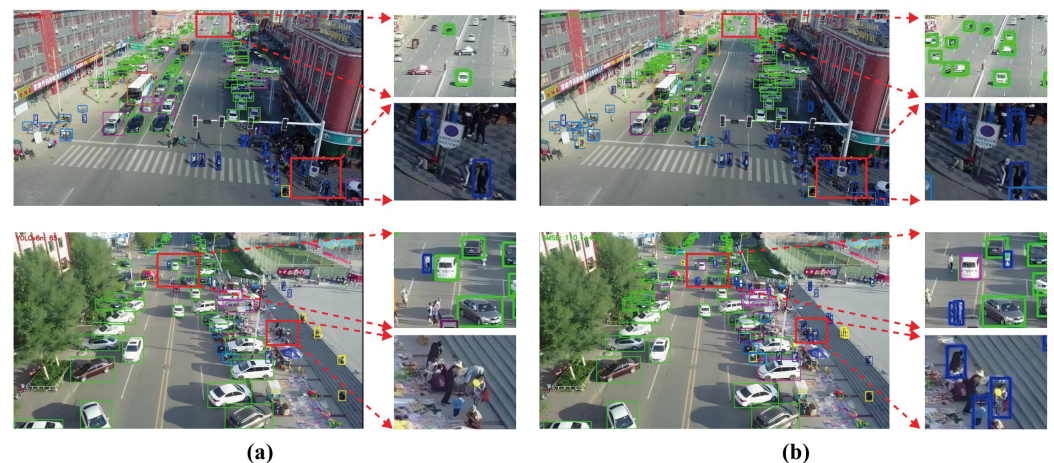


Figure 6. Comparison of detection results on the VisDrone2019 dataset. (a) YOLOv8n baseline. (b) Proposed LUIM-YOLO.

5. Conclusions

With the goals of greater suitability for UAV platforms and parameter-efficient detection, this paper proposes an improved lightweight model, LUIM-YOLO, that aims to raise small-object accuracy while keeping the network compact.

At the core of the model is the LMSFE module, which couples multi-scale parallel convolutions with channel- and spatial-attention to extract and reinforce the faint signals of small objects. Complementing it, the AMSB module is built from depthwise separable

convolutions so that multi-scale context and salient semantics are retained without inflating the parameter count. A further component, the ECC-FPN, employs a multi-path fusion scheme that lets shallow spatial detail and deep semantics complement one another, and that improves the integration of localization information. Finally, the detection head is re-engineered to make the model more responsive to the position and scale of ultra-small objects.

Experiments on VisDrone2019 confirm that the proposed approach raises detection accuracy markedly while cutting the parameter count substantially, which points to its promise for edge-side UAV perception. This gain, nonetheless, comes with a moderate rise in floating-point operations (FLOPs), and the latency study on the Jetson Orin indicates that its PyTorch inference is still slower than the YOLOv8n baseline. LUIM-YOLO should therefore be regarded as a parameter-efficient, accuracy-oriented detector rather than the fastest low-latency option. Subsequent work will pursue lighter feature-interaction mechanisms, low-latency attention modules, TensorRT-based deployment, pruning, and hardware–algorithm co-design in order to further boost inference efficiency on UAV platforms.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The VisDrone2019 dataset used in this study is available at <https://github.com/VisDrone/VisDrone-Dataset> (accessed on 29 April 2026).

Acknowledgments: The authors would like to thank the National University of Defense Technology for their support during this research. During the preparation of this manuscript, the authors used ChatGPT 4o (OpenAI, San Francisco, CA, USA) for grammar checking and language polishing. The authors have reviewed and edited the output extensively and take full responsibility for the content of this publication.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
YOLO	You Only Look Once
FPN	Feature Pyramid Network
PANet	Path Aggregation Network
LMSFE	Lightweight Multi-Scale Feature Enhancement
AMSB	Adaptive Multi-Scale Bottleneck
ECC-FPN	Enhanced Cross-layer Compensation Feature Pyramid Network

References

1. Wu, Z.; Jiang, W.; Liu, A. Sensing and Communication Coverage in Internet of Drone Things: Challenges and Opportunities. *IEEE Internet Things Mag.* **2025**, *8*, 86–92. [CrossRef]
2. Cai, H.; Zhang, J.; Xu, J. ALDNet: A Lightweight and Efficient Drone Detection Network. *Meas. Sci. Technol.* **2025**, *36*, 025402. [CrossRef]

3. Van Oers, A.M.; Rammers, T. Adversarial Patch Size and Positioning. In *Proceedings of the Target and Background Signatures XI: Traditional Methods and Artificial Intelligence*; SPIE: Bellingham, WA, USA, 2025; p. 4. [\[CrossRef\]](#)
4. Shi, K.; Li, Q.; Li, H. LMFAN: Lightweight Multi-Scale Feature Aggregation Network with Channel Pruning and Knowledge Distillation for Ship Detection in Remote Sensing Images. *Remote Sens. Appl. Soc. Environ.* **2025**, *39*, 101692. [\[CrossRef\]](#)
5. Shehzadi, T.; Hashmi, K.A.; Liwicki, M.; Stricker, D.; Afzal, M.Z. Object Detection with Transformers: A Review. *Sensors* **2025**, *25*, 6025. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Zhai, H.; Du, J.; Ai, Y.; Hu, T. Edge Deployment of Deep Networks for Visual Detection: A Review. *IEEE Sens. J.* **2025**, *25*, 18662–18683. [\[CrossRef\]](#)
7. Sharif, M.H.; Jiao, L.; Omlin, C.W. Deep Crowd Anomaly Detection: State-of-the-Art, Challenges, and Future Research Directions. *Artif. Intell. Rev.* **2025**, *58*, 139. [\[CrossRef\]](#)
8. Qin, H.; Xu, T.; Tang, Y.; Xu, F.; Li, J. OSFormer: One-Step Transformer for Infrared Video Small Object Detection. *IEEE Trans. Image Process.* **2025**, *34*, 5725–5736. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Liu, J.; Lu, Z.; Cen, Y. Enhancing Object Detection with Fourier Series. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 2581–2596. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Asad, M.; Ullah, I.; Sistu, G.; Madden, M.G. Towards Robust Autonomous Driving: Out-of-Distribution Object Detection in Bird's Eye View Space. *IEEE Open J. Veh. Technol.* **2025**, *6*, 1673–1685. [\[CrossRef\]](#)
11. Nikouei, M.; Baroutian, B.; Nabavi, S. Small Object Detection: A Comprehensive Survey on Challenges, Techniques and Real-World Applications. *Intell. Syst. Appl.* **2025**, *27*, 200561. [\[CrossRef\]](#)
12. Sun, Y.; Wang, J.; You, Y. YOLO-E: A Lightweight Object Detection Algorithm for Military Targets. *Signal Image Video Process.* **2025**, *19*, 241. [\[CrossRef\]](#)
13. Chen, J.; Huang, J.; Zhang, Z.; Yang, J.; Wu, Z.; Luo, R. ATM-Net: A Lightweight Multimodal Fusion Network for Real-Time UAV-Based Object Detection. *Drones* **2026**, *10*, 67. [\[CrossRef\]](#)
14. John, M.; Santhanalakshmi, S.; Amudha, J.; Zhou, J. Comparative Analysis of Image Mosaicing Techniques for Aerial Agriculture Field Imaging. *Eur. J. Remote Sens.* **2025**, *58*, 2507744. [\[CrossRef\]](#)
15. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2017; pp. 1800–1807. [\[CrossRef\]](#)
16. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2017; pp. 936–944. [\[CrossRef\]](#)
17. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2018; pp. 8759–8768. [\[CrossRef\]](#)
18. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *Proceedings of the Computer Vision—ECCV 2020; Lecture Notes in Computer Science*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; Volume 12346, pp. 213–229. [\[CrossRef\]](#)
19. Wang, H.; Gao, J. SF-DETR: A Scale-Frequency Detection Transformer for Drone-View Object Detection. *Sensors* **2025**, *25*, 2190. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Zhuang, J.; Zheng, Y.; Guo, B.; Yan, Y. Globally Deformable Information Selection Transformer for Underwater Image Enhancement. *IEEE Trans. Circuits Syst. Video Technol.* **2025**, *35*, 19–32. [\[CrossRef\]](#)
21. Li, Z.; Song, J.; Yuan, G.; Li, J. CDSS-Pan: Convolutional Dictionary Learning with State Space Model for Pansharpening. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2025**, *18*, 1734–1751. [\[CrossRef\]](#)
22. Li, Z.; Yuan, G.; Li, J. DUCD: Deep Unfolding Convolutional-Dictionary Network for Pansharpening Remote Sensing Image. *Expert Syst. Appl.* **2024**, *249*, 123589. [\[CrossRef\]](#)
23. Girshick, R. Fast R-CNN. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2015; pp. 1440–1448. [\[CrossRef\]](#)
24. 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. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Haghiabi, Z.; Mokari, N.; Abbasi Arand, B.; Imani, M. Building Detection in SAR Images Based on Fusion of Classic and Deep Learning Features. *Int. J. Remote Sens.* **2024**, *45*, 3577–3596. [\[CrossRef\]](#)
26. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Golestani, S.N.; SadeghiBakhi, M.; Ma, K.F.; Leung, H. Hydra-Mask-RCNN: An Adaptive HydraNet Architecture for Autonomous Aerial Vehicle Object Detection. *IEEE Open J. Instrum. Meas.* **2025**, *4*, 1–12. [\[CrossRef\]](#)
28. Liu, W.; Quijano, K.; Crawford, M.M. YOLOv5-Tassel: Detecting Tassels in RGB UAV Imagery with Improved YOLOv5 Based on Transfer Learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 8085–8094. [\[CrossRef\]](#)

29. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* **2022**, arXiv:2207.02696. [[CrossRef](#)]
30. Liu, C.; Wu, Y.; Liu, J.; Sun, Z. Improved YOLOv3 Network for Insulator Detection in Aerial Images with Diverse Background Interference. *Electronics* **2021**, *10*, 771. [[CrossRef](#)]
31. Yang, J.; Yang, B.; Li, Z.; Yamaguchi, Y.; Xiao, W. Parameter-Reduced YOLOv8n with GhostConv and C3Ghost for Automated Blood Cell Detection. *Bioengineering* **2026**, *13*, 321. [[CrossRef](#)] [[PubMed](#)]
32. Tang, J.; Xu, B.; Li, J.; Zhang, M.; Huang, C.; Li, F. Ghost-YOLO-GBH: A Lightweight Framework for Robust Small Traffic Sign Detection via GhostNet and Bidirectional Multi-Scale Feature Fusion. *Eng* **2025**, *6*, 196. [[CrossRef](#)]
33. Sun, Y.; Zhang, C.; Li, X. MDSF-YOLO: Advancing Object Detection with a Multiscale Dilated Sequence Fusion Network. *IEEE Trans. Neural Netw. Learn. Syst.* **2026**, *37*, 1236–1247. [[CrossRef](#)] [[PubMed](#)]
34. Chen, Y.; Yuan, X.; Wang, J. YOLO-MS: Rethinking Multi-Scale Representation Learning for Real-Time Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 4240–4252. [[CrossRef](#)] [[PubMed](#)]
35. Peng, R. A Camouflaged Target Detection Method with Improved YOLOv5 Algorithm. *J. Electron. Inf. Technol.* **2024**, *46*, 3324. [[CrossRef](#)]
36. Qu, J.; Li, Q.; Pan, J. SS-YOLOv8: Small-Size Object Detection Algorithm Based on Improved YOLOv8 for UAV Imagery. *Multimed. Syst.* **2025**, *31*, 42. [[CrossRef](#)]
37. Xue, R.; Hua, S.; Xu, H. FECI-RTDETR a Lightweight Unmanned Aerial Vehicle Infrared Small Target Detector Algorithm Based on RT-DETR. *IEEE Access* **2025**, *13*, 9578–9591. [[CrossRef](#)]
38. Wang, Y.; Lin, X.; Xiang, Z.; Su, W.H. VM-YOLO: YOLO with VMamba for Strawberry Flowers Detection. *Plants* **2025**, *14*, 468. [[CrossRef](#)] [[PubMed](#)]
39. Zhou, M.; Wan, X.; Yang, Y. EBR-YOLO: A Lightweight Detection Method for Non-Motorized Vehicles Based on Drone Aerial Images. *Sensors* **2025**, *25*, 196. [[CrossRef](#)] [[PubMed](#)]
40. Wang, C.; Han, Y.; Yang, C. CF-YOLO for Small Target Detection in Drone Imagery Based on YOLOv11 Algorithm. *Sci. Rep.* **2025**, *15*, 16741. [[CrossRef](#)] [[PubMed](#)]
41. Bi, J.; Li, K.; Zheng, X.; Zhang, G.; Lei, T. SPDC-YOLO: An Efficient Small Target Detection Network Based on Improved YOLOv8 for Drone Aerial Image. *Remote Sens.* **2025**, *17*, 685. [[CrossRef](#)]
42. Li, H.; Li, Y.; Xiao, L.; Zhang, Y.; Cao, L.; Wu, D. RLRD-YOLO: An Improved YOLOv8 Algorithm for Small Object Detection from an Unmanned Aerial Vehicle (UAV) Perspective. *Drones* **2025**, *9*, 293. [[CrossRef](#)]
43. Hou, T.; Leng, C.; Wang, J. MFEL-YOLO for Small Object Detection in UAV Aerial Images. *Expert Syst. Appl.* **2025**, *291*, 128459. [[CrossRef](#)]
44. Wang, Z.; Zhang, K.; Wu, F.; Lv, H. YOLO-PEL: The Efficient and Lightweight Vehicle Detection Method Based on YOLO Algorithm. *Sensors* **2025**, *25*, 1959. [[CrossRef](#)] [[PubMed](#)]
45. Hwang, H.; Kim, D.; Kim, H. FD-YOLO: A YOLO Network Optimized for Fall Detection. *Appl. Sci.* **2025**, *15*, 453. [[CrossRef](#)]
46. Elhenidy, A.M.; Labib, L.M.; Haikal, A.Y.; Saafan, M.M. GY-YOLO: Ghost Separable YOLO for Pedestrian Detection. *Neural Comput. Appl.* **2025**, *37*, 14907–14933. [[CrossRef](#)]
47. Akdoğan, C.; Özer, T.; Oğuz, Y. PP-YOLO: Deep Learning Based Detection Model to Detect Apple and Cherry Trees in Orchard Based on Histogram and Wavelet Preprocessing Techniques. *Comput. Electron. Agric.* **2025**, *232*, 110052. [[CrossRef](#)]
48. Hou, W.; Wu, H.; Wu, D. Small Object Detection Method for UAV Remote Sensing Images Based on α S-YOLO. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2025**, *18*, 8984–8994. [[CrossRef](#)]
49. Zhang, G.; Peng, Y.; Li, J. YOLO-MARS: An Enhanced YOLOv8n for Small Object Detection in UAV Aerial Imagery. *Sensors* **2025**, *25*, 2534. [[CrossRef](#)] [[PubMed](#)]
50. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO. 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 29 April 2026).
51. Wang, H.; Wang, C.; Fu, Q. YOLOFIV: Object Detection Algorithm for Around-the-Clock Aerial Remote Sensing Images by Fusing Infrared and Visible Features. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 15269–15287. [[CrossRef](#)]
52. Li, X.; Diao, W.; Mao, Y. OGMN: Occlusion-Guided Multi-Task Network for Object Detection in UAV Images. *Isprs J. Photogramm. Remote Sens.* **2023**, *199*, 242–257. [[CrossRef](#)]
53. Li, M.; Li, S.; Xiang, W.; Zhang, L. MDQE: Mining Discriminative Query Embeddings to Segment Occluded Instances on Challenging Videos. *arXiv* **2023**, arXiv:2303.14395. [[CrossRef](#)]
54. Du, D.; Zhu, P.; Wen, L. VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*; IEEE: New York, NY, USA, 2019; pp. 213–226. [[CrossRef](#)]
55. Liu, W.; Anguelov, D.; Erhan, D. SSD: Single Shot MultiBox Detector. In *Proceedings of the Computer Vision—ECCV 2016*; Lecture Notes in Computer Science; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; Volume 9905, pp. 21–37. [[CrossRef](#)]

56. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616. [[CrossRef](#)]
57. Wang, A.; Chen, H.; Liu, L. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458. [[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.