Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.065238

ARTICLE

YOLO-LE: A Lightweight and Efficient UAV Aerial Image Target Detection Model

Zhe Chen*, Yinyang Zhang and Sihao Xing

School of Computer Science, China University of Geosciences (Wuhan), Wuhan, 430074, China

*Corresponding Author: Zhe Chen. Email: chenzhe@cug.edu.cn

Received: 07 March 2025; Accepted: 21 April 2025; Published: 09 June 2025

ABSTRACT: Unmanned aerial vehicle (UAV) imagery poses significant challenges for object detection due to extreme scale variations, high-density small targets (68% in VisDrone dataset), and complex backgrounds. While YOLO-series models achieve speed-accuracy trade-offs via fixed convolution kernels and manual feature fusion, their rigid architectures struggle with multi-scale adaptability, as exemplified by YOLOv8n's 36.4% mAP and 13.9% small-object AP on VisDrone2019. This paper presents YOLO-LE, a lightweight framework addressing these limitations through three novel designs: (1) We introduce the C2f-Dy and LDown modules to enhance the backbone's sensitivity to small-object features while reducing backbone parameters, thereby improving model efficiency. (2) An adaptive feature fusion module is designed to dynamically integrate multi-scale feature maps, optimizing the neck structure, reducing neck complexity, and enhancing overall model performance. (3) We replace the original loss function with a distributed focal loss and incorporate a lightweight self-attention mechanism to improve small-object recognition and bounding box regression accuracy. Experimental results demonstrate that YOLO-LE achieves 39.9% mAP@0.5 on VisDrone2019, representing a 9.6% improvement over YOLOv8n, while maintaining 8.5 GFLOPs computational efficiency. This provides an efficient solution for UAV object detection in complex scenarios.

KEYWORDS: Deep learning; target detection; UAV image; YOLO; adaptive feature fusion

1 Introduction

With the rapid development of drone technology, aerial imagery applications have become prevalent across fields such as smart cities [1], environmental monitoring [2], precision agriculture [3], disaster warning [4], and emergency response [5]. However, as illustrated in Fig. 1, target detection in drone-acquired aerial imagery poses numerous challenges. Traditional detection algorithms frequently encounter difficulties in handling the extensive scale variations of targets caused by unique aerial perspectives. Additionally, the wide field of view, complex backgrounds, and high density of small targets in aerial images further increase the detection complexity. To address these challenges, it is necessary to enhance the model's feature fusion capabilities to retain both small-and large-target characteristics after feature fusion.

Considering the performance limitations of UAVs and the need for real-time operation, developing lightweight and efficient feature extraction and fusion algorithms has become a vital research direction for target detection in UAV aerial imagery [6].

YOLOv8 [7] constitutes a substantial progression within the You Only Look Once (YOLO) [8] series, ushering in remarkable enhancements in terms of both performance and flexibility. The architecture of YOLOv8 is composed of three principal components (as depicted in Fig. 2): the backbone, the neck, and the

Copyright © 2025 The Authors. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

head, which are respectively accountable for feature extraction, feature fusion, and the generation of final detection outcomes. YOLOv8 integrates innovative designs that significantly boost detection efficiency and accuracy. For example, it substitutes the C3 module in YOLOv5 [9] with the C2f module. This substitution enhances feature extraction efficiency by decreasing the number of convolutional layers. Moreover, in the bottleneck structure, 3 × 3 convolutions are adopted instead of 1 × 1 convolutions, further fortifying the feature extraction capabilities. Modifications to the feature fusion approach across different stages facilitate more efficient integration of feature maps at diverse scales.

Figure 1: Pictures from the perspective of drone aerial photography

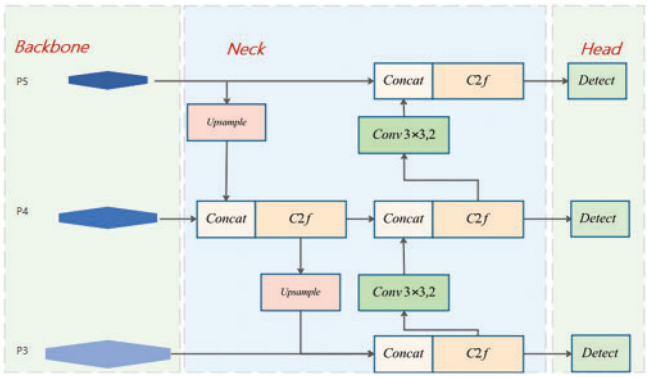


Figure 2: YOLOv8 module structure

Furthermore, YOLOv8 employs an anchor-free detection strategy, directly forecasting the center location and aspect ratio of the target. This approach streamlines the detection procedure and accelerates the detection speed. Through the separation and optimization of classification and regression tasks, YOLOv8 demonstrates enhanced adaptability to targets of diverse categories and scales. These improvements contribute to the enhancement of YOLOv8's accuracy, flexibility, generalization ability, and adaptability in object detection scenarios. Nevertheless, as YOLOv8 is developed using the COCO [10] dataset, which has a relatively small number of small objects, its performance in small-object detection tasks is still not ideal.

We present the YOLO-LE model (illustrated in Fig. 3) built upon the YOLOv8 framework. Initially, we devise two novel modules to optimize the original backbone network. These modules not only enhance its feature extraction capabilities but also reduce the computational burden. Subsequently, within the neck component, we develop an adaptive feature fusion module along with a corresponding feature pyramid

structure. This enables effective multi-scale feature fusion, thereby improving the model's robustness to the scale variations of small objects observed from diverse viewpoints. Finally, regarding the detection head, we enhance the recognition and localization accuracy for small targets by incorporating the distributed focal loss [11] and weighted local correlation computation. Experimental results validate that YOLO-LE significantly elevates the detection performance in complex scenes. The main contributions of our work can be summarized as follows:

(1) We devised a feature extraction module, namely C2f-Dy, and a lightweight downsampling module, denoted as LDown, to optimize the backbone. This was achieved by reducing the number of network parameters and enhancing the feature extraction for small targets. The C2f-Dy module promotes the gradient flow through the introduction of additional parameters and optimizes the gradient-flow branch to restrict the number of channels. The LDown module integrates pooling and shunt operations, which not only significantly cuts down the computational cost but also strengthens the feature extraction capabilities for small targets.

(2) We engineered the Adaptive Multi-Feature Fusion (AMFF) module to perform adaptive fusion of features at different scales. The AMFF module is capable of capturing the interactive and contextual relationships within features. It offers a versatile design that can be readily integrated into other models.

(3) We carried out optimizations on the original detection head and introduced the Local Feature Enhancement Head (LEHead). The LEHead improves the recognition of small objects by means of local feature extraction and integration. It employs distributed focal loss and a simplified self-attention mechanism to enhance bounding box regression and the processing of local features.

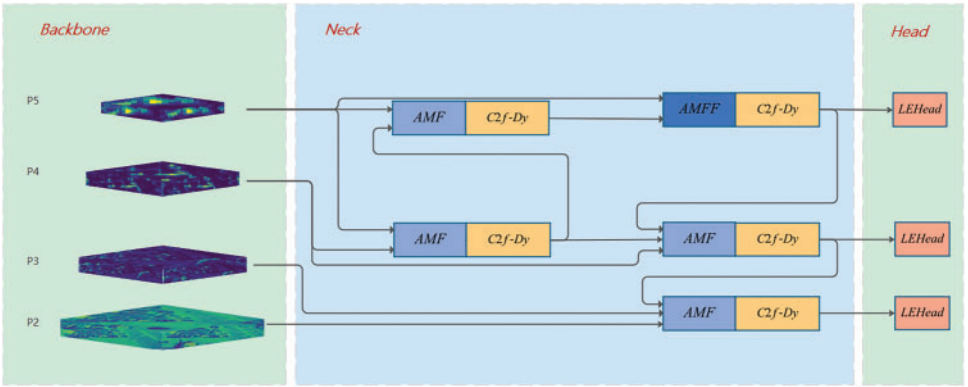


Figure 3: YOLO-LE module structure

2 Related Work

Traditional target detection methods, including Fast R-CNN [12], Faster R-CNN [13], and SPP-Net [14], have been extensively utilized in target detection for UAV-acquired images. Nevertheless, these two-stage detectors exhibit certain limitations in terms of speed, rendering them inappropriate for real-time detection applications. The YOLO series, encompassing YOLOv3 [15], YOLOv4 [16], YOLOv5, and YOLOv8, is frequently employed for target detection in UAV-acquired imagery. In contrast to traditional two-stage detectors, YOLO models substantially boost the detection speed by directly conducting target localization and classification on the input image. As shown in Table 1, the applications and existing limitations of its successive versions in the series are presented.

Table 1: Ablation experiments

Version	Main applications	Limitations
YOLOv1	Real-time object detection, such as traffic monitoring and security monitoring.	Difficulty in detecting small targets and new objects unseen during training, with low localization accuracy.
YOLOv2	Real-time detection of a large number of object categories, such as autonomous driving and security monitoring.	The model fails to effectively detect extremely small targets, and its complexity increases.
YOLOv3	Object detection in complex scenarios, such as autonomous driving and industrial defect detection.	Large model size makes it difficult to deploy on embedded devices.
YOLOv4	High-precision real-time detection, such as medical imaging and agricultural inspection	Still has limitations in detecting extremely small targets.
YOLOv5	Real-time detection on edge devices, such as drone surveillance and agricultural pest detection.	The detection ability for small targets and dense objects needs to be improved.
YOLOv6	Industrial-grade real-time detection, such as autonomous driving and security monitoring.	The robustness for complex scenarios still needs optimization.
YOLOv7	Real-time detection and complex scenarios analysis, such as medical imaging and industrial manufacturing.	High model complexity increases deployment difficulty.
YOLOv8	Real-time detection on edge devices, such as drone surveillance and agricultural detection.	The detection of extremely small targets still has room for improvement.

To tackle the problem of restricted resolution in UAV-captured aerial images, the work in presents a two-dimensional hybrid attention (DDMA) mechanism [17]. This method amalgamates channel and spatial attention to incorporate both local and non-local attention information, thereby diminishing omissions and false detections induced by dense targets. Nevertheless, this approach depends on intricate attention architectures, leading to elevated computational demands and augmented requirements for hardware.

Given the restricted computational resources on UAV platforms, lightweight models have attracted substantial attention. Models like MobileNetV2 [18], MobileNetV3 [19], and ShuffleNetV2 [20] have been incorporated into the YOLO framework to optimize the model size and inference speed. Moreover, the work in [21] suggests the utilization of depthwise separable convolution. This approach reduces the number of parameters in comparison to conventional convolution, thereby further shrinking the model size. To more effectively handle small and dense targets in aerial images, the study in [22] re-structures the feature fusion network by introducing an upsampling layer. This enhancement enables the model to pay more attention to small-target features. Additionally, SPD convolution [23] improves the model's feature extraction ability. It achieves this by downsampling the feature map while retaining crucial learning information. Finally, EIoU [24] is adopted to enhance regression accuracy by minimizing positional loss during training. Nevertheless, these methods typically face challenges in achieving a balance between model accuracy and size.

3 The Proposed YOLO-LE Model

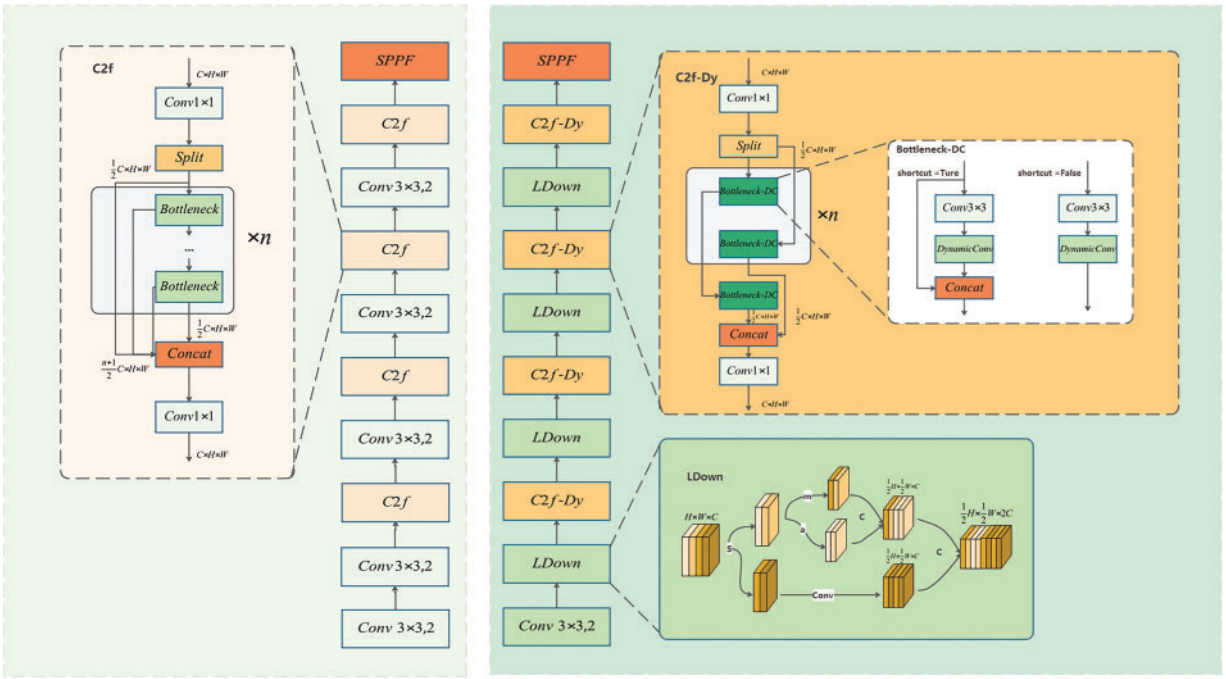
3.1 Overall Structure

The overall architecture of YOLOv8-LE is structured as follows:

1. We substitute the C2f and Conv modules in YOLOv8 with C2f-Dy and LDown. This substitution aims to augment the backbone's feature extraction capabilities for small targets, thereby attaining a more lightweight backbone design.
2. We reconfigure the neck feature pyramid by leveraging the Adaptive Multi-Feature Fusion module (AMFF) and its simplified variant, AMF. The AMFF and AMF modules operate on features spanning two to three distinct scales. This enables a more effective combination of shallow-level texture information and deep-level abstract information. In contrast to YOLOv8, we incorporate a shallower feature layer, denoted as P2, to capture a greater number of features related to small targets.
3. We employ the LFEHead for target localization and classification.

3.2 Feature Extraction Backbone

Our enhancements to the backbone are depicted in Fig. 4, in which we introduce the C2f-Dy and LDown modules.



(a) YOLOv8n Backbone (b) YOLO-LE Backbone

Figure 4: Backbone structure. Where s represents split, m and a represent average and max pooling, and c represents concat

The proposed C2f-Dy module represents a bottleneck module constructed based on the cross-stage partial (CSP) architecture [25]. As depicted in Fig. 4, it is composed of a Bottleneck-Dy component and a multi-path information fusion strategy. This design aims to augment feature extraction and target detection performance. In contrast to the original C2f module, which merely connects the initial layers

in a sequential manner, the C2f-Dy module adopts a more elaborate and efficient connection approach to enhance performance.

In the Bottleneck-DC module (illustrated in Fig. 5), both a traditional convolution and a DynamicConv [26] are incorporated. For the input F_{in}, the formula of the Bottleneck-DC is expressed as follows:

$$\begin{aligned} &\text{shortcut} = \text{True}: \\ &\quad F_{\text{out}} = \text{Concat}(\text{Conv3} \times 3(F_{\text{in}}), \text{DConv}(F_{\text{in}})) \\ &\text{shortcut} = \text{False}: \\ &\quad F_{\text{out}} = \text{DConv}(\text{Conv3} \times 3(F_{\text{in}})) \end{aligned} \tag{1}$$

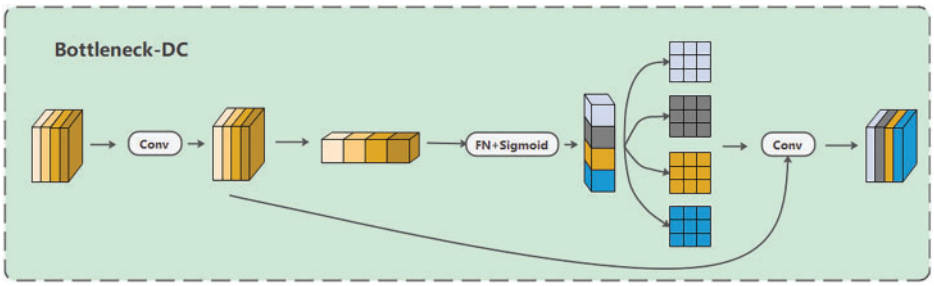


Figure 5: Bottleneck-DC module structure

In small target detection tasks, the inherent fixity of traditional convolutional kernels restricts their adaptability during the feature extraction process. This limitation diminishes their capacity to effectively capture the features of objects with diverse sizes. DynamicConv overcomes this constraint by incorporating learnable parameters into the convolution operation. This innovation empowers the network to adaptively modify the weights of the convolutional kernel in accordance with the characteristics of the input image.

The central concept of DynamicConv is the utilization of conditional convolution, in which the weights of the convolution kernel are dynamically adjusted according to the input data. Specifically, the input D_{in} undergoes global average pooling initially. Subsequently, it passes through a fully-connected layer and an activation function to generate a weight coefficient α. This generated weight coefficient is then employed in the conditional convolution operation:

$$D_{\text{out}} = \sum_{i=1}^{N} \sigma(\text{FC}(D_{\text{pool}}(D_{\text{in}})))_i \cdot \text{Conv}_i(D_{\text{in}}) \tag{2}$$

In the formula, N denotes the number of convolution kernels, α_i stands for the weight coefficient of the i-th convolution kernel, Conv_i indicates the convolution operation carried out by the i-th kernel, and D_{out} represents the final output.

YOLOv8n conducts downsampling through convolution with a kernel size of 3 and a stride of 2. Although this approach enables the learning of more intricate feature representations by tuning the parameters, it directly decreases the resolution of the feature map. Consequently, this leads to the loss of information regarding small objects. To tackle this problem, our LDown module (as illustrated in Fig. 4) executes feature segmentation. It maintains and augments feature representations by means of average pooling and max pooling, with a specific emphasis on preserving the prominent features of small objects. At the same time, it effectively reduces the computational burden of the model.

3.3 Adaptive Multi-Feature Fusion Module

The architecture of the Adaptive Multi-Scale Feature Fusion (AMFF) network proposed in this paper is depicted in Fig. 6. The AMFF network aims to optimize the feature fusion process. It does so by decreasing the quantity of fusion operations and, concurrently, increasing the number of features participating in each fusion. This approach, in turn, enhances the model's detection performance for small targets. Meanwhile, to uphold the model's lightweight characteristic, we utilize a simplified AMF module in specific feature fusion segments of the model's neck.

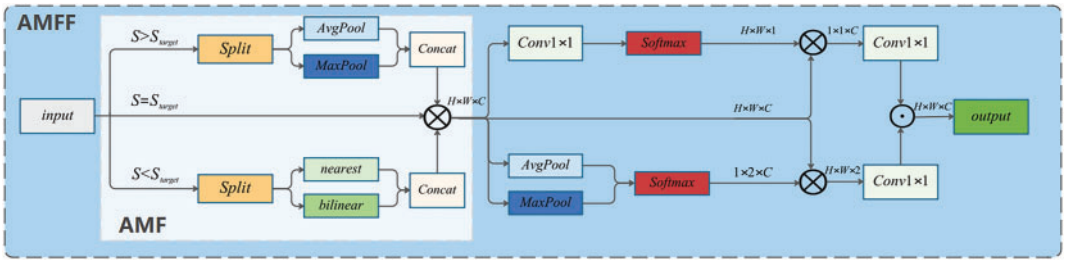


Figure 6: AMFF module structure. S represents the input feature map size ($H \times W$), and S_{target} represents the set output feature map size

The AMFF is composed of three components: adjustment of the channel number, alignment of the feature size, and weighted fusion of features. During the process of feature alignment, we implement adaptive maximum pooling and average pooling on high-resolution feature maps. The former is utilized to extract the edge features of small targets, while the latter is employed to extract their texture features, respectively. This approach enhances the network's adaptability to various types of small targets. For low-resolution feature maps, we adopt nearest neighbor interpolation. By copying the nearest pixel values, it helps to preserve the edge and texture features. In contrast, bilinear interpolation generates new pixel values through the weighted averaging of surrounding pixels, resulting in smoother transitions. This contributes to reducing noise and aliasing effects, thereby making the features of small targets more distinct. Subsequently, we carry out element-wise multiplication on the feature maps. This operation serves to amplify the influence of important features and eliminate noise, thus achieving effective feature fusion.

In the feature weighted fusion section, we integrate the spatial attention and channel attention mechanisms. By taking advantage of the weights of both, we aim to generate a more precise and elaborate attention map. Initially, we compute the spatial and channel attention weights of the features. Subsequently, we utilize the spatial attention weights to derive the channel attention weights and employ the channel attention weights to create the spatial attention map. Through the fusion of these two mechanisms, we are able to consider both global and local information, thereby enhancing the richness and comprehensiveness of the feature representation.

3.4 Local Feature Enhancement Detection Head

We have devised a Local Feature Enhancement Detection Head (LFEHead) to generate bounding boxes and class probabilities for object detection. The LFEHead is composed of several crucial components: Convolutional layers (Conv), Distributed Focus Loss (DFL), a Local Attention (CLA) module, and the final detection layer.

DFL represents a variant of Focal Loss. Through the process of assigning weights to the samples of each category, the model can more effectively learn the features of a small number of categories, thereby enhancing

the model's accuracy. The formula for DFL is presented as follows:

$$L_{\mathrm{DFL}} = -\frac{1}{N}\sum_{i=1}^{C} w_i \alpha_i (1-p_i)^{\gamma} \log(p_i) \tag{3}$$

where:

- C denotes the number of classes.
- N denotes the number of samples.
- w_i denotes the weight of the i-th class.
- α_i denotes the sample ratio of the i-th class.
- γ denotes a modulation factor to control the weight of easy samples.

The central concept of CLA is to compute the average attention score within each local range. This average attention score is then magnified (multiplied by 2). Subsequently, the amplified mean score is used to subtract the original dot product result dots. As a result, elements that initially scored above the average will have higher scores, while those that scored below the average will have lower scores. This process enhances the contrast of the attention weights, enabling the model to more clearly differentiate between important and unimportant regions.

Suppose the input feature maps are x_1 and x_2, with shapes of (B, C, W, H), respectively, and the local range is R. Then:

$$\begin{aligned} &\mathrm{Q} = \mathrm{Linear}_q(x_2), \\ &\mathrm{K} = \mathrm{Linear}_k(\mathrm{Local}(x_1)), \\ &\mathrm{V} = \mathrm{Linear}_v(\mathrm{Local}(x_1)) \end{aligned} \tag{4}$$

- Dot product calculation

$$\mathrm{dots} = \sum \frac{Q \cdot K}{R} \tag{5}$$

- Attention weight

$$\begin{aligned} &\mathrm{irr} = \mathrm{mean}(\mathrm{dotsdim} = 3).unsq(3) \times 2 - \mathrm{dots}, \\ &\mathrm{att} = \mathrm{Softmax}(\mathrm{irr}) \end{aligned} \tag{6}$$

- Output

$$\mathrm{out} = \sum (V \times \mathrm{att}.unsq(4))\mathrm{output} = \frac{\mathrm{out} + x_2}{2} \tag{7}$$

4 Equations

4.1 Dataset and Experimental Environment Configuration

The VisDrone-2019 dataset [27] is a large-scale, diverse collection of drone-captured aerial images designed for computer vision tasks. Comprising 288 video clips (261,908 frames) and 10,209 static images, it covers extensive environmental and weather conditions. The dataset includes meticulous annotations for multi-category objects (pedestrians, vehicles, buildings), providing rich scene context, object categories, and occlusion attributes. Its key characteristics—high target density, small object detection requirements, and diverse data distribution—pose significant challenges for object detection research. For this experiment,

the dataset was randomly partitioned into VisDrone-2019-train, -val, and -test subsets at a 6:2:2 ratio, with detailed experimental configurations summarized in Table 2.

Table 2: Experimental configuration

Device	
GPU	NVIDIA GeForce RTX 3080Ti
System	Ubuntu 18.04
Framework	PyTorch v1.1.0
NVIDIA CUDA	version CUDA 11.3
Python version	Python 3.8

All experiments utilized a unified setup: Ubuntu 18.04, PyTorch 1.1.0, CUDA 11.3, Python 3.8, and an NVIDIA GeForce RTX 3080Ti GPU (12 GB VRAM). Stochastic gradient descent (SGD) was applied for optimization with an initial learning rate of 0.01, 0.99 momentum, and 0.0005 weight decay. Training parameters included a 640 × 640 input resolution, batch size of 16, and 100 total epochs.

4.2 Ablation Experiments

We adopt YOLOv8n as the baseline for performing ablation experiments to validate the efficacy of our proposed method. The experimental outcomes are presented in Table 3.

Table 3: Ablation experiments in VisDrone-2019-test; Backbone means using C2f-Dy and LDowm modules to improve the backbone network; Neck means using AMFF to improve the neck network; Head means using LFEHead to replace the original detection head

Model	Backbone	Neck	Head	mAP(0.5) (%)	mAP@0.5:0.95 (%)	FLOPs (G)
YOLOv8n	–	–	–	36.4	18.9	8.2
A Model	✓	–	–	37.8	20.1	**6.1**
B Model	–	✓	–	38.3	20.4	7.8
C Model	–	–	✓	38.6	21.7	11.2
D Model	✓	✓	–	39.0	21.6	6.5
E Model	✓	–	✓	39.0	21.7	9.1
F Model	–	✓	✓	39.3	22.2	10.8
G Model	✓	✓	✓	**39.9**	**22.5**	8.5

As depicted in A of Table 3, the backbone enhanced by our integration of the C2f-Dy and LDown modules achieves a 25.6% reduction in the computational load while simultaneously boosting the model's accuracy. This outcome clearly demonstrates the high efficiency of our modules in feature extraction.

In B, where solely the Neck improved by the AMFF is employed, the mean Average Precision (mAP) experiences a substantial increase (by 1.9%), accompanied by a marginal reduction in the computational load. This indicates the effectiveness of the AMFF module in enhancing feature fusion.

In C, when only the improved LFEHead is utilized, the mAP exhibits a significant improvement (2.2%); however, the computational load also rises considerably. This suggests that the new detection head adds complexity while enhancing the detection accuracy.

As can be observed from E and F, our combination of different modules leads to a further improvement in mAP while maintaining high computational efficiency. The combination of the Backbone and Head yields particularly remarkable effects.

In our experiments, we further evaluated the detection accuracy among different categories for diverse combinations, aiming to analyze the effectiveness of the improvements implemented in various components. The results are presented in Table 4. Evidently, several remarkable enhancements can be discerned from the table.

Table 4: The comparison results of ten categories evaluated on VisDrone-2019-test

Model	PP	PL	BC	CR	VN	TK	TC	AT	BS	MT	mAP (0.5) (%)
YOLOv8n	40.6	33.1	13.9	**66.2**	40.9	31.2	20.6	26.0	50.9	40.4	36.4
A Model	42.2	38.6	17.4	62.1	39.4	32.9	20.9	31.2	53.4	39.9	37.8
B Model	43.4	38.1	19.5	65.7	38.3	33.5	21.7	30.4	52.3	40.1	38.3
C Model	43.6	40.1	20.3	64.1	42.6	31.3	21.5	29.9	50.5	42.2	38.6
D Model	44.2	42.3	19.4	62.5	40.1	33.4	21.5	33.2	54.3	39.5	39.0
E Model	43.9	41.3	18.9	63.5	41.1	33.2	21.1	32.8	52.1	42.3	39.0
F Model	44.0	43.1	20.7	64.3	**42.7**	32.3	20.9	31.9	50.5	42.1	39.3
G Model	**44.2**	**42.8**	**21.5**	63.5	41.4	**33.7**	**21.8**	**33.4**	**54.6**	**42.4**	**39.9**

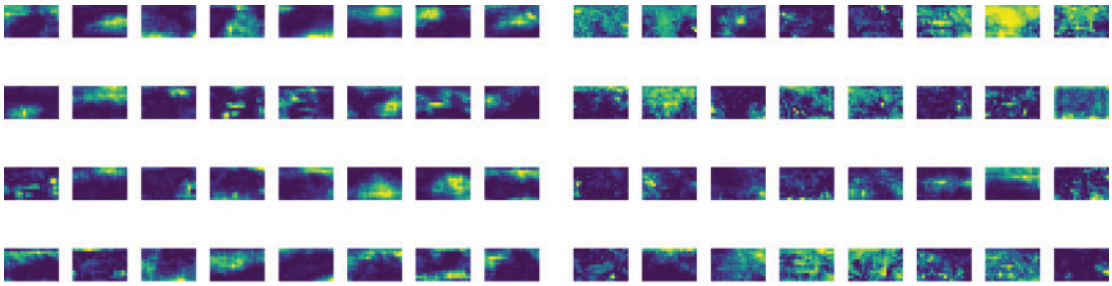
Model A exhibited enhancements in detection accuracy. Specifically, the detection accuracy for Pedestrians increased from 40.6% to 42.2%, and for Bicycle Riders, it rose from 13.9% to 17.4%. This manifestation highlights the advantage of the improved Backbone in fine-grained feature extraction, especially when dealing with small targets and intricate backgrounds, thereby showcasing the efficacy of the C2f and LDown modules.

In Model B, the upgraded Neck brought about substantial improvements across numerous categories. Notably, for Pedestrians, Bicycle Riders, and Trucks, the enhancements were remarkable, which underscores the efficiency of the AMFF model in multi-scale feature integration.

Model C further elevated the detection accuracy for Pedestrians from 40.6% to 43.6% and for Bicycle Riders from 13.9% to 20.3%, which serves as evidence of the effectiveness of the LFEHead module in small target detection.

Models F and G demonstrated even more pronounced improvements. Model F, through the combination of the enhanced Neck and Head, exhibited superior detection performance for small targets and in complex backgrounds. Model G achieved significant gains in accuracy. Specifically, the accuracy for Pedestrian detection reached 44.2% (an increase of 3.6%), for Bicycle Riders it was 21.5% (an increase of 7.6%), and for Motorcycle detection, it was 42.4% (an increase of 2%). These results indicate that the model excels in detecting small targets against complex backgrounds and performs remarkably well in multi-scale target detection.

To illustrate the merits of the C2f-Dy and LDown modules we have designed in the feature extraction of small targets, we carried out feature visualization experiments. The outcomes are presented in Fig. 7.



(a) YOLO-LE Backbone Feature Visualization (b) YOLOv8n Backbone Feature Visualization

Figure 7: Feature visualization of YOLO-LE backbone and YOLOv8n's backbone

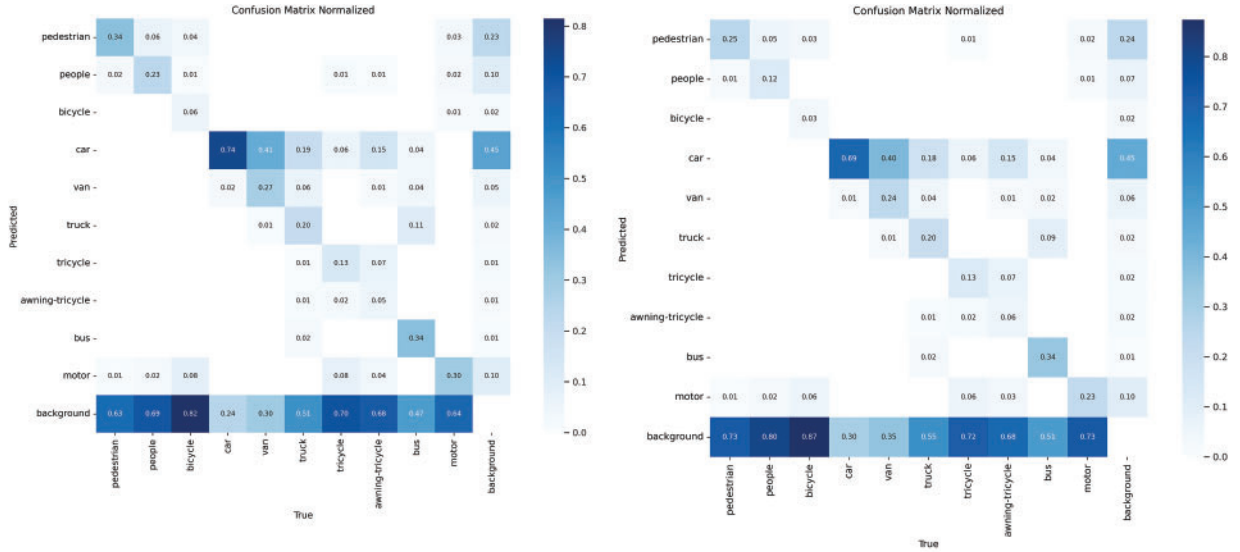
From the feature map on the left, it is clearly discernible that our backbone network demonstrates robust small object feature extraction capabilities across multiple feature maps. The activation regions are concentrated, which implies that our network efficiently captures the intricate details of small targets. The bright spots and high-contrast areas in the feature map accentuate the network's sensitivity and precision in handling small objects. Conversely, although the feature map on the right also shows evidence of small target feature extraction, the activation area is more dispersed, and some feature maps exhibit low contrast. This indicates that the network might fail to fully capture the nuances of small targets under certain circumstances. These experiments showcase that our designed module outperforms others in the task of small target feature extraction.

To validate the small-object detection performance of our proposed detection head, we conducted comparative experiments on classification accuracy between Model C and YOLOv8. The heatmaps reveal significant differences in predictions across different categories between the two models.

For YOLOv8n (Fig. 8b), in the prediction confidence matrices of some categories, there are regions with lighter colors (lower confidence). This indicates that the classification accuracy for these categories is poor, especially in terms of the ability to distinguish between some similar categories. For example, in the case of categories with certain visual similarities such as "apple" and "orange", the confidence in the cross-region is not high enough, which shows that the model has difficulty in accurately distinguishing between them. In contrast, for our LFEHead (Fig. 8a), the colors in the overall confidence matrix are generally darker, indicating more accurate classification for all categories. When it comes to distinguishing between similar categories, the improved model performs much better. Taking "apple" and "orange" as an example, the confidence in their cross-region is significantly higher than that of YOLOv8n, reducing the probability of misclassification. In other categories, such as "book" and "bottle", the improved model also shows higher confidence. This demonstrates that the improvement measures applied to the detection head effectively enhance the model's recognition accuracy for different categories, enabling more reliable classification of targets and leading to more accurate detection results in practical applications.

4.3 Comparison Experiments

To comprehensively evaluate the performance and generalization ability of YOLO-LE, we conducted comparative experiments on two distinct datasets: VisDrone-2019 test set and DOTAv2. The former validates the model's detection capabilities in scenarios with dense small targets and complex backgrounds, while the latter evaluates its transferability to cross-domain aerial imagery tasks.



(a) Classification Accuracy of LFEHead (b) Classification Accuracy of YOLOv8n

Figure 8: Classification accuracy comparison of LFEHead and YOLOv8n's head

We compared YOLO-LE with mainstream detectors on the VisDrone-2019 test set, including two-stage detectors (Faster R-CNN), single-stage lightweight models (YOLOv8n, YOLOv6n, CDNet), and emerging architectures (RT-DETR-r50). As shown in Table 5, the results show that YOLO-LE achieves 39.9% and 22.5% in mAP@0.5 and mAP@0.5:0.95, respectively, significantly outperforming the baseline YOLOv8n (36.4%, 20.9%). It surpasses all comparative models in mAP@0.5, demonstrating stronger detection capabilities for small and dense targets. With a parameter count of 4.0×10^6 and FLOPS of 8.5×10^9, YOLO-LE maintains comparable computational cost to YOLOv8n despite a slight increase in parameters, far lower than RT-DETR-r50 and Faster R-CNN, achieving a balance between accuracy and efficiency. In scenarios with a high proportion of small targets and complex backgrounds, YOLO-LE improves mAP@0.5 by 6.3% compared to YOLOv6n and by 2.5% compared to RT-DETR-r50, demonstrating that the C2f-Dy and LDown modules, along with AMFF neck optimization, effectively alleviate the limitations of traditional models in small-target detection and multi-scale adaptability. The comparative experiments indicate that YOLO-LE outperforms mainstream models in accuracy, computational efficiency, and complex-scene adaptability, validating the effectiveness of backbone network optimization, adaptive feature fusion, and detection head improvements, and providing an efficient solution for real-time target detection in UAV aerial photography scenarios.

Table 5: Performance evaluation experiments conducted on VisDrone-2019-test

Model	mAP (0.5) (%)	mAP (0.95) (%)	Param ($\times 10^6$)	FLOPS ($\times 10^9$)
SSD [28]	17.2	9.1	58.0	–
CDNet [29]	32.2	17.2	1.8	–
Faster RCNN	21.8	15.8	165.6	–
YOLOv6n [30]	33.6	17.4	4.2	11.9
DETR [31]	37.4	21.1	42.9	136
YOLOv8n	36.4	20.9	3.0	**8.2**
Ours	**39.9**	**22.5**	**4.0**	8.5

To validate the model's transferability, we directly applied the YOLO-LE model trained on VisDrone-2019-train to the DOTAv2 dataset, which contains large-scale satellite imagery with diverse object orientations and sparse target distributions. As shown in Table 6, YOLO-LE achieves 40.9% mAP@0.5 and 22.8% mAP@0.5:0.95, outperforming YOLOv8n by 3.5% and 1.3%, respectively. This indicates that the lightweight architecture and adaptive feature fusion mechanisms generalize well to unseen domains.

Table 6: Performance evaluation experiments conducted on DOTAv2

Model	mAP (0.5) (%)	mAP (0.95) (%)
SSD [28]	16.9	9.8
CDNet [29]	33.2	17.8
Faster RCNN	22.7	15.9
YOLOv6n [30]	33.9	17.3
DETR [31]	38.6	22.1
YOLOv8n	37.4	21.5
Ours	**40.9**	**22.8**

The performance gap between YOLO-LE and other models (e.g., Faster R-CNN and DETR) is even more pronounced on DOTAv2, particularly for elongated or rotated targets. For example, YOLO-LE improves truck detection accuracy by 12.1% over YOLOv8n, demonstrating its robustness to geometric variations. This cross-domain success can be attributed to the C2f-Dy module's dynamic convolution, which adapts to varying target scales, and the LFEHead's local attention mechanism, which enhances spatial sensitivity without overfitting to dataset-specific features.

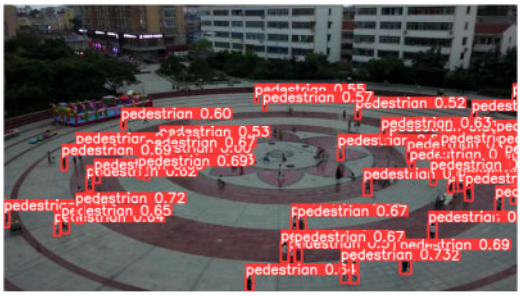
The experiments confirm YOLO-LE's superiority in both native and cross-domain scenarios. Its lightweight design ensures computational efficiency, while the adaptive modules enable robust feature representation across diverse aerial imaging conditions. The significant improvement on DOTAv2 further underscores the model's potential for real-world deployment, where training data may be limited or domain shifts exist. Future work will focus on optimizing domain adaptation strategies to further enhance generalization.

4.4 Visualization Analysis

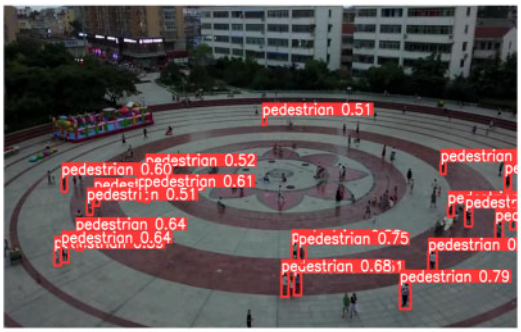
To validate the detection performance of the model, we chose two representative scenes for the experiments: a bustling square at night and a road viewed from an oblique perspective. The experiment compares and presents the detection visualization results of our model and the YOLOv8n model.

As is evident from Fig. 9, our model is capable of detecting a greater number of small targets that are overlooked by the YOLOv8n model, which suggests a superior ability in recognizing small targets.

As shown in Fig. 10, our model can accurately identify distant small targets without being influenced by nearby large targets, thereby demonstrating its remarkable feature fusion capability. In general, our model brings about a substantial improvement in detection performance.

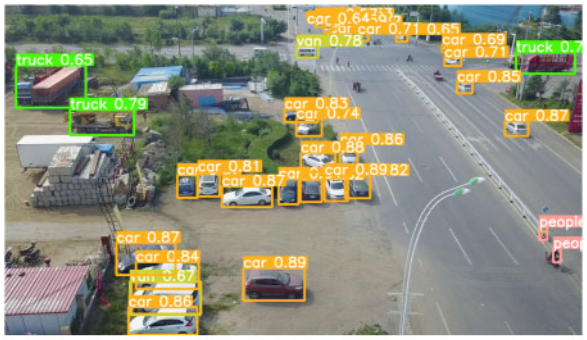


(a) YOLO-LE Detection in Crowded Square

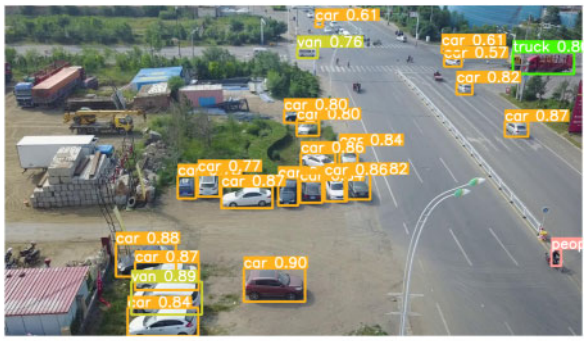


(b) YOLOv8n Detection in Crowded Square

Figure 9: Crowded squares at night



(a) YOLO-LE Detection on Roads



(b) YOLOv8n Detection on Roads

Figure 10: Roads with inclined angles

5 Conclusions

YOLO-LE enhances the fundamental feature extraction and downsampling modules within the backbone network. Consequently, there is a notable reduction in the number of model parameters, and simultaneously, the feature extraction ability, particularly for small targets, is significantly bolstered. The neck component is integrated with our proposed AMFF module. This module adaptively fuses multi-scale features and optimizes the neck architecture, leading to a decrease in model parameters and an increase in the processing speed. In the detection head, a local self-attention module is devised to substitute the distributed focal loss, which effectively improves both the model's convergence rate and detection precision.

Experimental results indicate that our model exhibits substantial performance enhancements. Nevertheless, the algorithm proposed in this paper still has certain limitations. Future research efforts will be concentrated on further minimizing the model complexity and boosting the recognition speed.

While YOLO-LE demonstrates significant advancements, several directions for future research can address its remaining limitations and expand its applicability:

1. Explore lightweight neural architecture search (NAS) techniques to automate the design of more efficient backbone and neck structures, balancing parameter reduction with feature representation capability.

2. Investigate sparse or dynamic attention strategies to further reduce the computational cost of the LFEHead, ensuring real-time inference on edge devices with limited resources.

3. Integrate complementary data sources (e.g., infrared imagery, LiDAR point clouds) to enhance detection performance in low-light or occluded scenarios, leveraging cross-modal feature fusion to improve small-target visibility.

These directions aim to further enhance YOLO-LE's efficiency, robustness, and practical utility, making it a more versatile solution for real-world UAV-based target detection tasks.

Acknowledgement: We sincerely thank all participants for their dedicated input. Their engagement was vital to this research. Also, gratitude goes to our friends for offering emotional support and helpful ideas during the study.

Funding Statement: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Author Contributions: Zhe Chen and Yinyang Zhang contributed to the conception and design of the study. The first draft of the manuscript was written by Yinyang Zhang, and all authors commented on previous versions of the manuscript. Sihao Xing provided assistance in the drawing of some model structure diagrams of this paper. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data were obtained from the authors' experiments and from the original papers. Supporting data are available from the corresponding author.

Ethics Approval: This study did not involve human subjects, human data, or animal experiments, and therefore no ethical approval was required.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Mohamed N, Al-Jaroodi J, Jawhar I, Idries A, Mohammed F. Unmanned aerial vehicles applications in future smart cities. Technol Forecast Soc Change. 2020;153(4):119293. doi:10.1016/j.techfore.2018.05.004.
2. Pan M, Chen C, Yin X, Huang Z. UAV-aided emergency environmental monitoring in infrastructure-less areas: lora mesh networking approach. IEEE Internet Things J. 2022;9(4):2918–32. doi:10.1109/JIOT.2021.3095494.

3. Tokekar P, Hook JV, Mulla D, Isler V. Sensor planning for a symbiotic uav and ugv system for precision agriculture. IEEE Trans Robot. 2016;32(6):1498–1511. doi:10.1109/TRO.2016.2603528.
4. Yang L, Sun Q, Ye Z-S. Designing mission abort strategies based on early-warning information: application to uav. IEEE Trans Ind Inform. 2020;16(1):277–87. doi:10.1109/TII.2019.2912427.
5. R.W. L, Boukerche A. Uav-mounted cloudlet systems for emergency response in industrial areas. IEEE Trans Ind Inform. 2022;18(11):8007–16. doi:10.1109/TII.2022.3174113.
6. Al-lQubaydhi N, Alenezi A, Alanazi T, Senyor A, Alanezi N, Alotaibi B, et al. Deep learning for unmanned aerial vehicles detection: a review. Comput Sci Rev. 2024;51(1):100614. doi:10.1016/j.cosrev.2023.100614.
7. Jocher G, Chaurasia A, Qiu J. Ultralytics YOLO [Internet]. [cited 2025 Apr 20]. Available from: https://github.com/ultralytics/ultralytics.
8. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: 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.
9. Jocher G. YOLOv5 by ultralytics [Internet]. [cited 2025 Apr 20]. Available from: https://github.com/ultralytics/yolov5.
10. Lin T-Y, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, et al. Microsoft COCO: common objects in context; 2015. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/1405.0312.
11. Li X, Wang W, Wu L, Chen S, Hu X, Li J, et al. Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/2006.04388.
12. Girshick R. Fast R-CNN; 2015. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/1504.08083.
13. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks; 2016. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/1506.01497.
14. He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Computer Vision—ECCV 2014; 2014 Sep 6–12; Zurich, Switzerland. p. 346–61. doi:10.1007/978-3-319-10578-9_23.
15. Redmon J, Farhadi A. YOLOv3: an incremental improvement; 2018. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/1804.02767.
16. Bochkovskiy A, Wang C-Y, Liao H-YM. YOLOv4: optimal speed and accuracy of object detection; 2020. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/2004.10934.
17. Bao W, Zhu Z, Hu G, Zhou X, Zhang D, Yang X. Uav remote sensing detection of tea leaf blight based on DDMA-YOLO. Comput Electron Agric. 2023;205:107637. doi:10.1016/j.compag.2023.107637.
18. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. MobileNetV2: inverted residuals and linear bottlenecks; 2019. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/1801.04381.
19. Howard A, Sandler M, Chu G, Chen L-C, Chen B, Tan M, et al. Searching for mobileNetV3; 2019. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/1905.02244.
20. Ma N, Zhang X, Zheng H-T, Sun J. Shufflenet v2: practical guidelines for efficient CNN architecture design. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. Computer Vision—ECCV 2018; 2018 Sep 8–14. Munich, Germany. p. 122–38.
21. Wang X, Zhao Q, Jiang P, Zheng Y, Yuan L, Yuan P. LDS-YOLO: a lightweight small object detection method for dead trees from shelter forest. Comput Electron Agric. 2022;198(8):107035. doi:10.1016/j.compag.2022.107035.
22. Li S, Liu C, Tang K, Meng F, Zhu Z, Zhou L, et al. Improved yolov5s algorithm for small target detection in uav aerial photography. IEEE Access. 2024;12:9784–91. doi:10.1109/ACCESS.2024.3353308.
23. Sunkara R, Luo T. No more strided convolutions or pooling: a new CNN building block for low-resolution images and small objects; 2022. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/2208.03641.
24. Zhang Y-F, Ren W, Zhang Z, Jia Z, Wang L, Tan T. Focal and Efficient IOU loss for accurate bounding box regression; 2022. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/2101.08158.
25. Wang C-Y, Liao H-YM, Yeh I-H, Wu Y-H, Chen P-Y, Hsieh J-W. CSPNet: a new backbone that can enhance learning capability of CNN; 2019. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/1911.11929.
26. Chen Y, Dai X, Liu M, Chen D, Yuan L, Liu Z. Dynamic convolution: attention over convolution kernels; 2020. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/1912.03458.

27. Zhu P, Wen L, Du D, Bian X, Fan H, Hu Q, et al. Detection and tracking meet drones challenge. IEEE Trans Pattern Anal Mach Intell. 2021;44(11):7380–99. doi:10.1109/TPAMI.2021.3119563.
28. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y. SSD: single shot MultiBox detector. In: Computer Vision—ECCV 2016; 2016 Oct 11–14; Amsterdam, The Netherlands. p. 21–37. doi:10.1007/978-3-319-46448-0_2.
29. He H, Huang Z, Ding Y, Song G, Wang L, Ren Q, et al. Cdnet: centripetal direction network for nuclear instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021 Oct. 11–17; Montreal, BC, Canada. p. 4026–35.
30. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, et al. YOLOv6: a single-stage object detection framework for industrial applications; 2022. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/2209.02976.
31. Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang Q, et al. DETRs beat YOLOs on real-time object detection; 2024. [cited 2025 Apr 20]. Available from: https://arxiv.org/abs/2304.08069.