Lightweight Model for Small Drone Detection in Multiple Scenarios

Qin Li, Nannan Zhao, Xinyu Ouyang

***Abstract*—To address the limitations of existing Unmanned Aerial Vehicle (UAV) target detection models in terms of computational resource constraints and poor performance on small object detection, this paper proposes a lightweight improved UAV detection algorithm named UL-Yolo (Ultralight-YOLO). A novel structure DC2f (Depthwise Convolutional C2f) is designed using lightweight depthwise separable convolution (DepthSepConv), which significantly reduces model parameters while enhancing feature extraction capabilities. In addition, the lightweight backbone PP-LCNet (PaddlePaddle Lightweight and Compact Network) is employed, leveraging lightweight convolutions and pyramid pooling to maintain high detection accuracy while reducing computational overhead, making the model more suitable for low-resource and real-time inference scenarios. Furthermore, MPDIoU (Minimum Point Distance Intersection over Union) is adopted as the bounding box regression loss function, which optimizes the matching between predicted and ground-truth boxes through minimum point distance, thereby improving detection performance and enhancing training stability. Experimental results demonstrate that UL-Yolo achieves a 2.7% improvement in mAP@0.5 on the Det-Fly dataset compared to YOLOv8, while reducing parameter count by 75.8%. Although the FPS slightly decreases from 99.7 to 98.7, the model still maintains a high inference speed, indicating a well-balanced trade-off between computational complexity and inference efficiency.**

***Index Terms*—Unmanned Aerial Vehicle, Object detection, YOLOv8, PP-LCNet, DPConv, MPDIOU.**

I. Introduction

IN recent years, unmanned aerial vehicles (UAVs) have demonstrated tremendous application potential in civil, agricultural [1], military, and scientific research fields due to their compact size, high maneuverability, and ease of operation. They play a critical role in various scenarios, including ground target detection and tracking, power line inspection in harsh environments, atmospheric monitoring, and disaster response [2]–[4]. Additionally, the detection of other airborne UAVs holds significant importance for applications such as collision avoidance [5], detection and countermeasures against enemy drones in combat scenarios, and coordinated multi-drone operations [6]. However, low-altitude, slow-speed, and small-sized UAVs pose substantial detection challenges due to their low flight altitude, slow velocity, and compact dimensions. The issue of "unauthorized flights" further exacerbates threats to public safety and national airspace security. Therefore, achieving rapid and accurate UAV identification is of paramount strategic importance for safeguarding civilian security, maintaining social stability, and protecting national interests. This makes UAV target detection technology a fundamental and essential component in building an effective defense system. Traditional vision-based drone detection [7], [8] methods are typically divided into two stages: feature extraction and classification. Several traditional feature extraction methods are widely employed in image processing, including Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT). These extracted features are subsequently fed into classification systems utilizing machine learning approaches such as Support Vector Machines (SVM) [9]. Although visual detection can effectively identify targets based on shape, color, and texture information, it is highly susceptible to weather conditions and obstructions. Overall, these five types of technologies each have their own advantages and limitations, demonstrating different strengths in specific scenarios. However, conventional drone detection and recognition methods mainly rely on handcrafted feature extraction [10], which is time-consuming, labor-intensive, and incapable of handling all complex situations effectively.

With the rapid development of deep learning [11], drone detection and identification technologies based on deep learning have achieved significant progress. A key advantage of deep learning is its ability to autonomously learn features without relying on manual extraction, greatly enhancing the accuracy and efficiency of drone detection. Liu Jiaming et al. [12] proposed a drone recognition method using deep convolutional neural networks to address low recognition accuracy. Their approach employs the SSD algorithm for drone target detection in video images, utilizes VGG16 for feature extraction and classification, and applies the BP algorithm to optimize network robustness. Zhang et al. [13] investigated air-to-air visual detection of micro-drones (UAVs) using monocular cameras. They introduced Det-Fly, a novel dataset covering diverse complex scenarios, and evaluated eight representative deep learning algorithms on this benchmark. For drone defense applications, Li et al. [14] improved the YOLOv8 algorithm by incorporating multi-detection heads and an attention mechanism (CBAM), significantly enhancing small-object detection precision in complex environments. Similarly, Cheng Q. [15] optimized YOLOv5 for lightweight detection by developing the CF2-MC feature extraction network and an MG fusion module, achieving higher accuracy with reduced complexity.Recent architectural innovations include Wang et al.'s [16] sensory wild attention module that replaces conventional convolutions, improving local-global feature integration. Feng Yunsong et al. [17]

Manuscript received April 25, 2025; revised July 25, 2025. This work is supported by the Fundamental Research Funds for the Liaoning Universities of China (Grant No. LJ212410146005).

Qin Li is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, 114051, China (e-mail: 1194951373@qq.com).

Nan-Nan Zhao is a professor of the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, 114051, China (Corresponding author, e-mail: 723306003@qq.com).

Xinyu Ouyang is a professor of the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, 114051, China (Corresponding author, e-mail: 13392862@qq.com).

proposed EDU-YOLO (based on YOLOv5s) with a ShuffleNetV2 backbone, Coordinate Attention, and Bidirectional FPN, while Zhao Yongjuan et al. [18] enhanced YOLOv8 via Edge-Sensitive Cross-Stage Fusion (C2f-ESCFFM) and Context-Aware Feature Pyramid (CAHS-FPN). Huang Min et al. [19] further advanced YOLOv8 with Shadow Convolution, EMA attention, and DCNv2 in their EDGS-YOLOv8 model. These innovations collectively improve UAV detection accuracy while maintaining model efficiency.

Drone-to-drone object detection faces two main challenges: First, the onboard computing resources of source drones are limited, while conventional deep learning-based detection algorithms typically require substantial computational power. Second, in aerial images captured by drones, the target drone occupies only about 0.03% of the image area. The diversity in scale caused by variations in shooting distance and angle, along with complex background interference, significantly increases detection difficulty. Although existing methods have made progress in improving object detection performance, they still suffer from high model complexity, large memory consumption, and poor performance in detecting small objects—making them difficult to deploy on edge devices. Furthermore, these models are prone to false detections or missed detections in complex environments. To address the issues of low detection accuracy for small objects under limited resources and complex conditions, this paper proposes a lightweight UL-YOLO (Ultralight-YOLO) model that enhances the feature extraction capability for small drone targets with extremely low parameter count. The model optimizes feature extraction efficiency, enhances small object detection, and maximizes computational resource utilization. By constructing a long-range pixel correlation network alongside spatial information enhancement and an edge-aware mechanism, it minimizes feature degradation, reduces missed detections, and eliminates redundant detection errors. As a result, it achieves high-precision drone target recognition even in resource-limited scenarios. The key contributions of this work include:

(1) This paper proposes a novel DC2f (Depthwise Convolutional C2f) structure that replaces the original C2f module in the Neck with DepthSepConv, a lightweight convolutional operation. This innovation not only enhances the network's feature extraction capability but also significantly reduces model parameters, resulting in a more lightweight and computationally efficient architecture.

(2) The model adopts the lightweight network PP-LCNet (PaddlePaddle Lightweight and Compact Network) as the backbone. PP-LCNet combines lightweight convolutions and pyramid pooling techniques, and utilizes the MKLDNN(Intel Math Kernel Library for Deep Neural Networks) acceleration strategy to improve feature extraction accuracy while maintaining efficient inference.

(3) MPDIoU (Minimum Point Distance Intersection over Union) is an extension of DIoU that replaces CIOU. It utilizes the minimum point distance metric to measure the similarity between predicted and ground-truth bounding boxes by directly minimizing the distance between their top-left and bottom-right corners. This approach optimizes bounding box regression, enhances object detection performance, and improves training stability.

II. Improved model

YOLOv8 [20] is an efficient object detection network based on improvements to the YOLO [21] series. Its architecture mainly consists of three parts: the backbone, the neck, and the detection head. The backbone is responsible for extracting features from the input image. YOLOv8 uses a lightweight network architecture, such as CSPDarknet (Cross Stage Partial Darknet) [22] or other efficient convolutional networks, to process low-level features of the image and pass them on to the upper layers for further processing. The feature extraction module uses deep convolutional neural networks (CNNs) to extract high-level features from the image. The neck network is responsible for feature fusion and constructing a feature pyramid, aimed at enhancing the representation of multi-scale features. The detection head generates the final detection results, including bounding box regression, class prediction, and object confidence.

To improve detection efficiency on edge devices, we propose several enhancements to YOLOv8n, aiming to increase target detection accuracy while reducing the model size. The network structure of our proposed method is illustrated in Figure 1. The architecture employs PP-LCNet as the backbone network, followed by a DC2f component designed to replace the original C2f module. Additionally, we integrate DepthSepConv into the DC2f component. By replacing traditional convolutions with DPConv, we significantly reduce computational overhead, resulting in a more compact and efficient model. These modifications collectively improve detection accuracy and robustness.

Since the target drones occupy only 0.03% of the image area, we introduced the P2 small-object detection head to mitigate accuracy loss from the lightweight design. Positioned at the shallowest level of the feature pyramid, P2 captures fine-grained features and improves sensitivity to small targets. To further optimize efficiency, we removed the P5 large-object detection head. While P5 excels at detecting large-scale objects, its lower resolution struggles to preserve small-target details. Eliminating P5 reduces computational redundancy, avoids over-optimization for large objects, and enhances both accuracy and efficiency for small-object detection.

A. Backbone

PP-LCNet (PaddlePaddle Lightweight and Compact Network) is a lightweight convolutional neural network developed by Baidu's PaddlePaddle team to address the computational constraints of deep learning models on edge devices [23]. By incorporating efficient architectural components like depthwise separable convolutions and bottleneck layers, it achieves significant reductions in both model parameters and computational complexity while preserving high accuracy and real-time performance. As illustrated in Figure 2, the network mainly consists of stacked depthwise separable convolutional (DepthSepConv) blocks.

DepthSepConv [24] decomposes the standard convolution operation into two sequential steps: depthwise convolution (DW) and pointwise convolution (PW). The depthwise convolution applies independent convolutional filters to each input channel, while the pointwise convolution employs 1×1 kernels to combine channel-wise outputs and generate

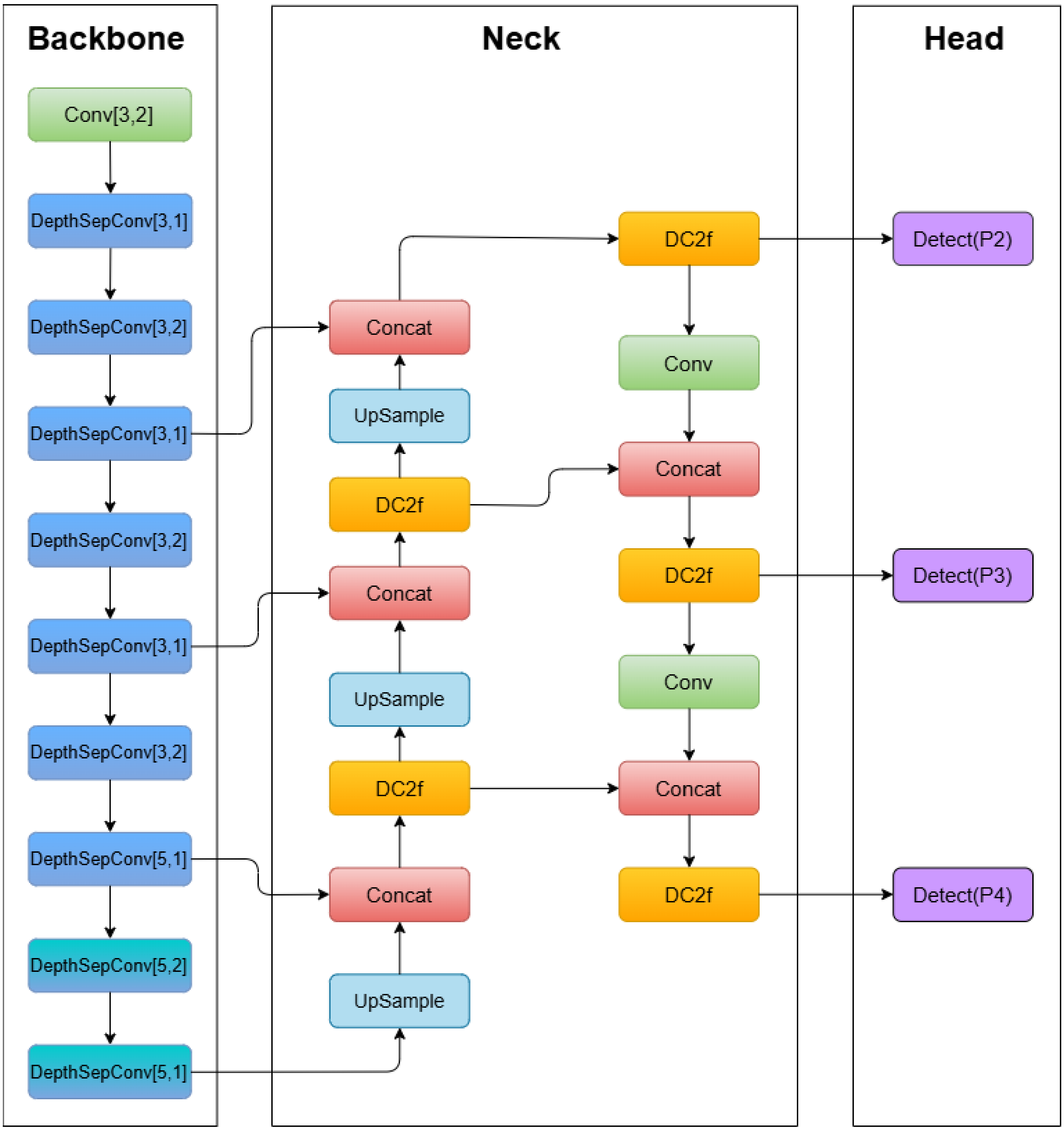


Fig. 1. UL-YOLO Structure Diagram

new feature maps. This decomposition dramatically reduces both parameter count and computational complexity, thereby enhancing overall efficiency.

The depthwise convolution operation applies separate spatial filtering to individual input channels while preserving channel independence. This process can be mathematically expressed as:

$$DW = \sum_{m=1}^{k} \sum_{n=1}^{k} W_c(m,n) \cdot X_c(i+m, j+n) \tag{1}$$

The pointwise convolution performs a linear combination between channels using 1×1 convolution kernels:

$$PW = \sum_{c=1}^{C} W_{c'}(c) \cdot X_c(i,j) \tag{2}$$

By combining depthwise and pointwise convolutions, the final output of the depthwise separable convolution is:

$$Y = PW(DW(X)) \tag{3}$$

The Squeeze-and-Excitation (SE) module [25] explicitly models inter-channel relationships, allowing the network to adaptively amplify informative features while suppressing less useful ones. In PP-LCNet, to maintain an optimal accuracy-speed tradeoff, SE modules are strategically placed only in the final network stages. Specifically, they are incorporated exclusively in the last two layers of depthwise separable convolution blocks containing 5×5 kernels. This design enhances model accuracy without compromising inference efficiency.

B. DC2f module

To further enhance the model's multi-scale feature extraction capabilities while reducing computational complexity, we propose the DC2f module. This module is an optimized improvement based on the C2f structure. Specifically, the DC2f module employs a channel-wise partitioning strategy, dividing the input feature map into two parallel

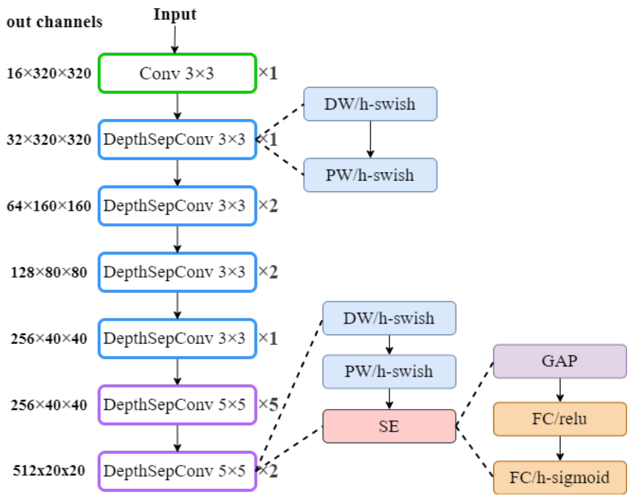


Fig. 2. PP-LCNet network backbone structure. DepthSepConv means depth-wise separable convolutions, DW means depth-wise convolution, PW means point-wise convolution, GAP means Global Average Pooling.

processing pathways: one branch retains the original information directly, while the other is processed through multiple DC2f_Bottleneck modules composed of depthwise separable convolutions. As shown in Figure 3 This design not only preserves the integrity of the original features but also reduces computational load through lightweight convolution operations. Subsequently, the feature maps from both pathways are concatenated along the channel dimension and fused using a 1×1 convolution layer, further compressing the channel count and reducing redundant information.

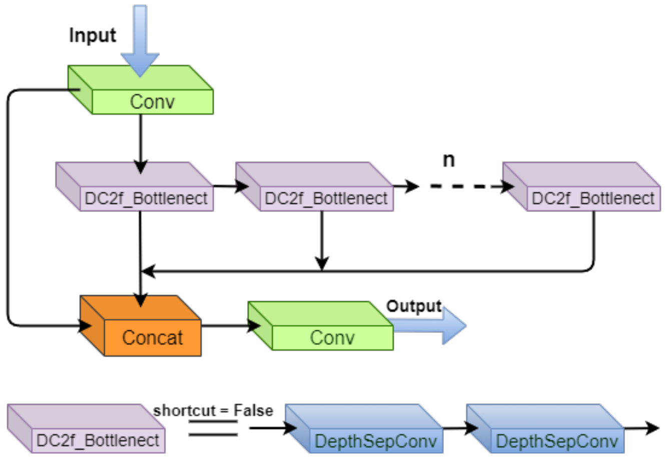


Fig. 3. Structure diagram of the DC2f module

The DC2f_Bottleneck module applies two depthwise separable convolutions (DepthSepConv) sequentially to the input feature $\mathbf{x}$:

$$y = Y_2(Y_1(x)) \tag{4}$$

The complete DC2f module can be divided into the following steps:

Input Processing:
The input feature $\mathbf{x}$ is first passed through a 1×1 convolutional layer conv$_1$, and then split into two parts: $\mathbf{y}_1$ and $\mathbf{y}_2$:

$$[y_1, y_2] = Split(Conv_1(x)) \tag{5}$$

Application of Bottleneck Modules:
The second part $\mathbf{y}_2$ is sequentially passed through n DC2f_Bottleneck modules to produce a series of features:

$$\begin{cases} y_3 = Bottleneck(y_2) \\ y_4 = Bottleneck(y_3) \\ \quad\vdots \\ y_{n+2} = Bottleneck(y_{n+1}) \end{cases} \tag{6}$$

Feature Concatenation:
All the generated features are concatenated along the channel dimension together with $\mathbf{y}_1$:

$$f_{concat} = Concat(y_1, f_1, \ldots, f_n) \tag{7}$$

Output Processing:
The concatenated feature map is passed through a final 1×1 convolutional layer conv$_2$ to generate the output:

$$y_{out} = Conv_2(f_{concat}) \tag{8}$$

C. MPDIOU loss function

To enhance detection accuracy, we propose a novel loss function named MPDIoU (Minimum Point Distance IoU) [26], which improves upon the traditional IoU metric. Unlike standard IoU that solely considers the overlap area between predicted and ground-truth boxes – potentially causing misjudgment by ignoring their positional displacement – MPDIoU incorporates the minimum point distance between their top-left and bottom-right corners for more precise similarity measurement. This approach comprehensively accounts for overlap coverage, center point distance, and aspect ratio discrepancy while maintaining computational efficiency, ultimately boosting object detection accuracy.

MPDIoU addresses this limitation by introducing Partial Distance (PD), which quantifies the displacement between bounding boxes to measure their dissimilarity. By replacing CIOU with MPDIoU, we enhance small drone detection accuracy and stabilize the training process. MPDIoU significantly boosts model performance in complex object detection scenarios by incorporating relative geometric relationships between boxes. The computation of MPDIoU involves the following steps:

1. Compute the IoU between the two bounding boxes to obtain the IoU value.

2. Calculate the Partial Distance (PD) between the two boxes.

3. Divide the PD value by the IoU value to obtain the MPDIoU score.

MPDIoU provides a more accurate assessment of object detection algorithm performance, especially for detecting distant objects. It has been widely used in object detection competitions and has become an important evaluation metric for object detection algorithms.

The inference formula for MPDIoU is defined as follows:

$$d_1^2 = (x_1^{\text{prd}} - x_1^{\text{gt}})^2 + (y_1^{\text{prd}} - y_1^{\text{gt}})^2 \tag{9}$$

$$d_2^2 = (x_2^{\text{prd}} - x_2^{\text{gt}})^2 + (y_2^{\text{prd}} - y_2^{\text{gt}})^2 \tag{10}$$

Let (x_1^{gt}, y_1^{gt}) and (x_2^{gt}, y_2^{gt}) define the top-left and bottom-right vertices of the ground truth box, while (x_1^{prd}, y_1^{prd})

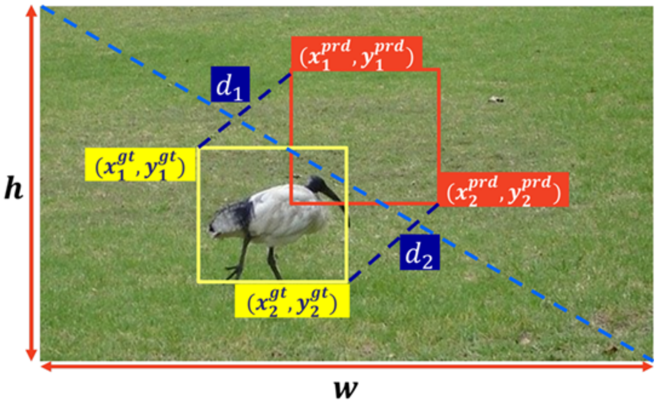


Fig. 4. Schematic diagram of the MPDIoU parameter structure

and (x_2^{prd}, y_2^{prd}) specify the corresponding corners of the predicted bounding box.

$$\text{MPDIoU} = \frac{A \cap B}{A \cup B} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{11}$$

$$\mathcal{L}_{\text{MPDIoU}} = 1 - \text{MPDIoU} \tag{12}$$

III. Experiments

A. Experimental Condition Setting

The hardware setup consists of an AMD Ryzen 7 7735H CPU with 8 cores and 16 threads, running at a base clock speed of 3.20 GHz, 16GB of RAM, and an NVIDIA GeForce RTX 4060 GPU with 8GB of VRAM. The experimental setup utilizes Python 3.10.15 along with PyTorch 2.0.1 and Torchvision 0.15.2 as the deep learning platform. Model training is implemented based on officially released pre-trained weights.

TABLE I
Training Parameter Configurations

Parameters	**Setup**
Epochs	300
Batch	8
Optimizer	SGD
NMS IoU	0.7
Base Learning Rate	0.01
End Learning Rate	0.01
Momentum	0.937
Weight-Decay	0.0005
Image Translation	0.1
Image Scale	0.5
Mosaic	1
Close Mosaic	Last 10 epochs

B. Experiment Dataset

This paper employs the Det-Fly dataset, comprising over 13,000 in-flight drone images captured by a chase drone. Compared to existing datasets, Det-Fly offers superior comprehensiveness by encompassing diverse background scenes, viewing angles, relative distances, flight altitudes, and lighting conditions – resulting in both high complexity and diversity while closely mimicking real-world operational scenarios. The dataset is split into 80% for training, 10% for validation, and 10% for testing. All images are normalized to 640×640 resolution, striking an optimal balance between real-time processing and detection accuracy. This standardized resolution enables efficient edge-device deployment while maintaining critical visual features for reliable detection.

C. Ablation Studies

To verify the efficacy of our method, we conduct systematic ablation experiments to analyze each component's contribution. The evaluation results, based on different metrics (mAP50, mAP95, GFLOPS, Parameters, and FPS), are shown in Table II. The baseline model (A), designed backbone (N), addition of small object detection head P2 and removal of P5 (P), replacement of loss function (I), and construction of DC2f module replacing C2f (C) are evaluated.

The table systematically compares five model configurations: the baseline (A), and its progressive enhancements (A+N, A+N+P, A+N+P+I, and A+N+P+I+C), quantitatively analyzing the metric variations across these versions. This approach ensures a thorough assessment of model effectiveness. Furthermore, the mAP@0.5 curve provides intuitive visualization of algorithmic improvements.

TABLE II
Ablation Experiment

Model	Map@0.5	Map@0.5:0.95	Params/M	GFLOPs	FPS(Tasks/s)
A	0.883	0.543	2.385	6.1	99.7
A+N	0.853	0.489	1.260	3.5	104.2
A+N+P	0.888	0.536	0.681	7.1	102.6
A+N+P+I	0.900	0.553	0.681	7.1	102.1
A+N+P+I +C	0.907	0.545	0.578	6.5	98.7

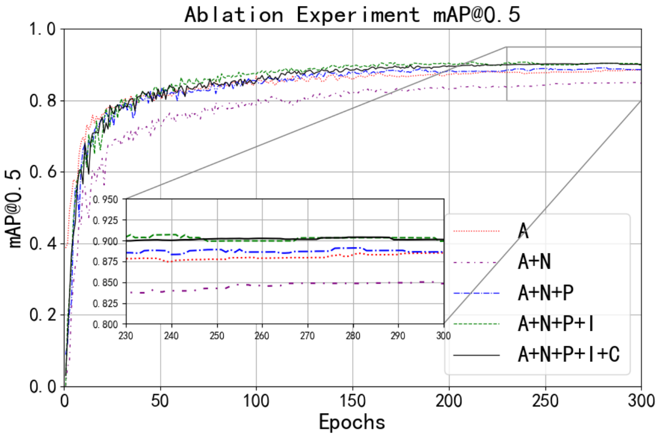


Fig. 5. mAP@0.5 curve

Through a series of ablation experiments, we systematically evaluated the impact of different improvements on the model's performance. The baseline model was optimized through several modifications: adjusting the backbone, adding the small object detection head P2 while removing P5, replacing the loss function, and substituting the C2f module with DC2f, resulting in the final improved model. The experimental results indicate that different modules have varying effects on model performance and resource consumption. The baseline model A performs well but has a high parameter count and computational complexity. After adjusting the backbone, the model's parameter count and computational complexity were significantly reduced, and inference speed improved, although with a slight performance decrease. The addition of the small object detection head module led to improved performance and parameter efficiency, but increased computational complexity. Replacing C2f with the DC2f module further compressed the model's parameters and computational complexity, making the model more lightweight. The improved model achieved an mAP@0.5 of 0.907, representing a 2.7% improvement over the baseline model, while reducing the parameter count by 75.8% and decreasing the model size by 3.3 times, effectively lowering storage and computational costs. In terms of inference speed, although FPS dropped from 99.7 to 98.7, it remained at a high level, indicating that the improved model achieves a good balance between computational complexity and speed. Additionally, replacing the original CIoU with the MPDIoU loss function further enhanced detection accuracy. Overall, the final improved model strikes a good balance between small object detection, model lightweighting, and computational efficiency, making it suitable for resource-constrained applications demanding high detection accuracy, such as drone detection and embedded device deployments.

To validate the performance of the UL-YOLO algorithm in achieving model lightweighting and small object detection, we conducted comparative experiments with the Faster R-CNN (a general object detection algorithm) and seven advanced YOLO-series algorithms (including YOLOv5, YOLOv8, Shufflenetv2-YOLOv8n, GhostNet-YOLOv8n, MobileNetV3-YOLOv8n, YOLOv10, and YOLOv11). These state-of-the-art algorithms were compared against the proposed UL-YOLO algorithm. All models were trained and tested on the Det-Fly dataset, and the comparison results in terms of parameter count, FPS, and GFLOPs are shown in Table III.

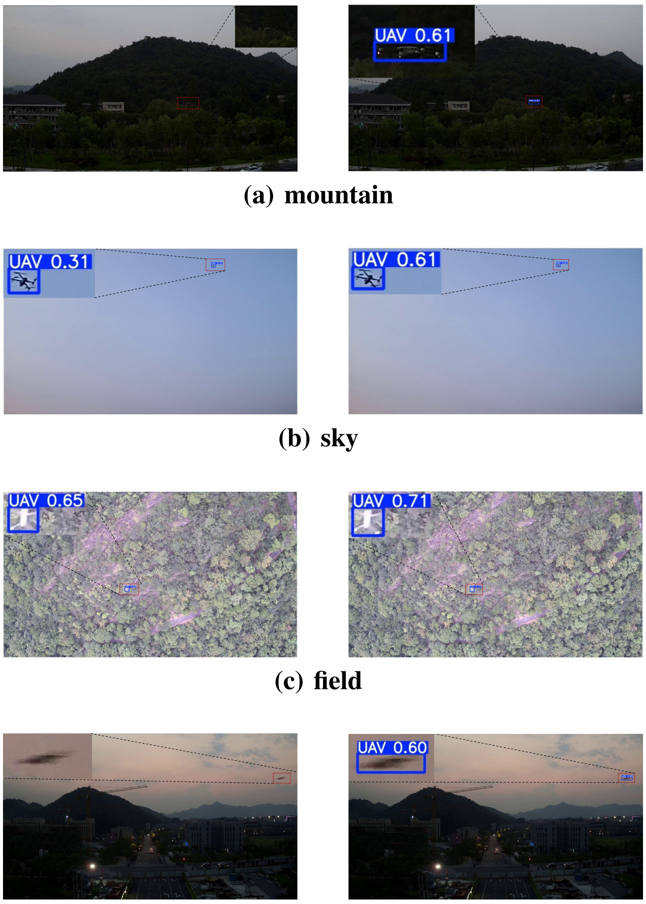
TABLE III
COMPARATIVE EXPERIMENT

Model	Map@0.5	Map@0.5:0.95	Params/M	GFLOPs	FPS (Tasks/s)
Faster R-CNN	0.656	0.356	54.534	129.7	8.1
YOLOV5n	0.921	0.566	1.761	4.1	101.2
YOLOV8n	0.883	0.543	2.385	6.1	99.7
YOLOV8n*	0.831	0.450	1.243	3.4	89.3
YOLOV8n*	0.847	0.507	2.883	5.4	49.8
YOLOV8n*	0.821	0.463	1.884	4.2	57.3
YOLOV10n	0.876	0.515	2.707	8.4	92.4
YOLOV11n	0.864	0.535	2.590	6.3	91.6
UL-YOLO	0.907	0.545	0.578	6.5	98.7

YOLOv8n* ranks as follows: Shufflenetv2-YOLOv8n, GhostNet-YOLOv8n, MobileNetV3-YOLOv8n.

As shown in the table, the UL-YOLO model achieves an excellent balance between lightweight design and detection performance, demonstrating significant practical value. It attains an mAP@0.5 of 0.907, ranking second only to YOLOv5n. In terms of lightweight design, UL-YOLO contains merely 577,887 parameters, accounting for 32.8% of YOLOv5n's and 24.2% of YOLOv8n's. Compared to YOLOv8n and YOLOv5n, it reduces the parameter count by 69.9% and 62.1%, respectively, substantially decreasing storage requirements and computational complexity. This lightweight architecture makes UL-YOLO an ideal solution for resource-constrained environments, particularly suited for embedded devices and edge computing. Additionally, UL-YOLO exhibits outstanding computational efficiency, achieving 98.7 FPS. Although marginally lower than YOLOv5n and YOLOv8n, it still satisfies real-time processing demands. In applications such as drone-based air-to-air target detection and tracking, it can rapidly process targets amid complex backgrounds. Overall, UL-YOLO achieves an optimal balance among high detection accuracy, model lightweighting, and computational efficiency.

D. Visualization Analysis



(a) mountain

(b) sky

(c) field

(d) urban

Fig. 6. Test results in different scenarios

The test results demonstrate that under different background conditions, our proposed UL-YOLO model surpasses the baseline model in partial performance metrics. In certain test images, the baseline model exhibits significant issues with both false negatives and false positives, including misclassifying highlighted background regions as targets or failing to detect small distant targets. By incorporating a lightweight feature extraction network and an enhanced loss function, our model effectively addresses these limitations, exhibiting remarkable robustness and detection accuracy. These experimental results and visual analyses collectively validate the superior capability of our proposed model in handling small target detection and complex background interference scenarios.

IV. Conclusions and prospects

Vision-based air-to-air target detection for drones shows significant potential in military, security, and logistics applications. This paper presents a lightweight air-to-air drone detection method specifically optimized for the constrained computational resources of drone platforms. Experimental results demonstrate that the proposed UltraL-YOLO algorithm achieves excellent performance across multiple key metrics: compared with conventional methods, it reduces model parameters by 75.8% and compresses model size to 69.9% of the original, substantially decreasing computational and storage requirements. Remarkably, the model maintains superior detection accuracy, particularly in processing complex backgrounds and identifying small targets, rendering it highly suitable for edge computing devices including handheld terminals and low-power drones.

Future research will focus on assessing the algorithm's performance on video datasets and further optimizing its integration with target tracking algorithms [27]. The proposed framework will employ UL-YOLO for initial drone target detection and location prediction, subsequently feeding these results into a tracking algorithm. By incorporating advanced tracking methodologies, the system is designed to achieve efficient and precise air-to-air target tracking. In cases where targets are lost due to occlusion or illumination variations, the detection module will automatically reactivate when targets reappear within the field of view, enabling smooth detection-to-tracking transitions. This integrated approach not only improves tracking stability and robustness but also establishes a foundation for drone interaction and cooperative operations in complex dynamic environments, thereby extending the practical applications and value of the proposed algorithm.

References

[1] Manuel Chad Agurob, Amiel Jhon Bano, Immanuel Paradela, Steve Clar, Earl Ryan Aleluya, and Carl John Salaan, "Autonomous Vision-based Unmanned Aerial Spray System with Variable Flow for Agricultural Application," *IAENG International Journal of Computer Science*, vol. 50, no. 3, pp. 1058-1073, 2023.

[2] Junhao Zhao, Gang Xiao, Xingchen Zhang, and Durga Prasad Bavirisetti, "A Survey on Object Tracking in Aerial Surveillance," In *Proceedings of the International Conference on Aerospace System Science and Engineering 2018*, Springer, 2019, pp. 53-68.

[3] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera, "Deep Learning in Video Multi-Object Tracking: A Survey," *Neurocomputing*, vol. 381, pp. 61-88, 2020.

[4] Zhenyu Na, Bowen Li, Xin Liu, Jun Wang, Mengshu Zhang, Yue Liu, and Beihang Mao, "UAV-based Wide-area Internet of Things: An Integrated Deployment Architecture," *IEEE Network*, vol. 35, no. 5, pp. 122-128, 2021.

[5] Yaru Cao, Zhijian He, Lujia Wang, Wenguan Wang, Yixuan Yuan, Dingwen Zhang, Jinglin Zhang, Pengfei Zhu, Luc Van Gool, Junwei Han, et al., "VisDrone-DET2021: The Vision Meets Drone Object Detection Challenge Results," In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2847-2854.

[6] Vikas Hassija, Vinay Chamola, Adhar Agrawal, Adit Goyal, Nguyen Cong Luong, Dusit Niyato, Fei Richard Yu, and Mohsen Guizani, "Fast, Reliable, and Secure Drone Communication: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2802-2832, 2021.

[7] Dong-Hyun Lee, "CNN-based Single Object Detection and Tracking in Videos and its Application to Drone Detection," *Multimedia Tools and Applications*, vol. 80, no. 26, pp. 34237-34248, 2021.

[8] Bilal Taha and Abdulhadi Shoufan, "Machine Learning-based Drone Detection and Classification: State-of-the-art in Research," *IEEE Access*, vol. 7, pp. 138669-138682, 2019.

[9] F. Gökçe, G. Üzölük, Ş. Erol, and S. Kalkan, "Vision-Based Detection and Distance Estimation of Micro Unmanned Aerial Vehicles," *Sensors*, vol. 15, no. 9, pp. 23805-23846, 2015.

[10] Yun Lin, Jicheng Jia, Sen Wang, Bin Ge, and Shiwen Mao, "Wireless Device Identification based on Radio Frequency Fingerprint Features," In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, IEEE, 2020, pp. 1-6.

[11] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I. Sánchez, "A Survey on Deep Learning in Medical Image Analysis," *Medical Image Analysis*, vol. 42, pp. 60-88, 2017.

[12] J. Liu, "Research on UAV Recognition Method Based on Deep Convolutional Neural Network," *Ship Electronic Engineering*, vol. 39, no. 2, pp. 5, 2019. [in Chinese]

[13] Y. Zheng, Z. Chen, D. Lv, Z. Li, and S. Zhao, "Air-to-Air Visual Detection of Micro-UAVs: An Experimental Evaluation of Deep Learning," *IEEE Robotics and Automation Letters*, vol. PP, no. 99, pp. 1-1, 2021.

[14] Can Li, Zhen Zuo, Bei Sun, Shudong Yuan, Zhaoyang Dang, and Jianfei Di, "UAV Target Detection Method Based on Multi-detection Head and Attention in Urban Background," In *International Conference on Autonomous Unmanned Systems*, Springer, 2023, pp. 168-177.

[15] Chuanyun Wang, Zhenfei Li, Qian Gao, Tong Cui, Dongdong Sun, and Wang Jiang, "Lightweight and Efficient Air-to-Air Unmanned Aerial Vehicle Detection Neural Networks," In *2023 IEEE International Conference on Unmanned Systems (ICUS)*, IEEE, 2023, pp. 1575-1580.

[16] Changyou Wang, Qing Zhang, and Jie Huang, "An Improved Multi-target Detection Algorithm in UAV Aerial Images Based on YOLOv8s Framework," *Engineering Letters*, vol. 33, no. 4, pp. 998-1007, 2025.

[17] Yunsong Feng, Tong Wang, Qiangfu Jiang, Chi Zhang, Shaohang Sun, and Wangjiahe Qian, "A Efficient and Accurate UAV Detection Method Based on YOLOv5s," *Applied Sciences*, vol. 14, no. 15, pp. 6398, 2024.

[18] Yongjuan Zhao, Lijin Wang, Guannan Lei, Chaozhe Guo, and Qiang Ma, "Lightweight UAV Small Target Detection and Perception Based on Improved YOLOv8-E," *Drones*, vol. 8, no. 11, pp. 681, 2024.

[19] Min Huang, Wenkai Mi, and Yuming Wang, "EDGS-YOLOv8: An Improved YOLOv8 Lightweight UAV Detection Model," *Drones*, vol. 8, no. 7, pp. 337, 2024.

[20] Rejin Varghese and M. Sambath, "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, IEEE, 2024, pp. 1-6.

[21] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You Only Look Once: Unified, Real-time Object Detection," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779-788.

[22] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh, "CSPNet: A New Backbone That Can Enhance Learning Capability of CNN," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 390-391.

[23] C. Cui, T. Gao, S. Wei, Y. Du, R. Guo, S. Dong, B. Lu, Y. Zhou, X. Lv, and Q. Liu, "PP-LCNet: A Lightweight CPU Convolutional Neural Network," *ArXiv Preprint ArXiv:2109.15099*, 2021.

[24] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *ArXiv Preprint ArXiv:1704.04861*, 2017.

[25] J. Hu, L. Shen, G. Sun, and S. Albanie, "Squeeze-and-Excitation Networks," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132-7141.

[26] Siliang Ma and Yong Xu, "MPDIoU: A Loss for Efficient and Accurate Bounding Box Regression," *ArXiv Preprint ArXiv:2307.07662*, 2023.

[27] Zhengpeng Li, Yuhang Bai, Jun Hu, Bin Yang, and Xuange Liu, "Multiple Object Tracking for Complex Motion Patterns," *Engineering Letters*, vol. 33, no. 4, pp. 1173-1184, 2025.