LSOD-YOLOv8s: A Lightweight Small Object Detection Model Based on YOLOv8 for UAV Aerial Images

Huikai Li, Jie Wu*

***Abstract*—Detecting small objects in aerial drone imagery is an extremely challenging research topic. This is primarily because the target size is relatively small, the background is complex, and occlusion occurs easily, which leads traditional object detection models to struggle in achieving ideal detection results. To enhance the detection performance of small objects, this paper proposes a lightweight small object detection model for aerial images captured by drones. Based on YOLOv8, this model adds a small object detection layer, introduces the FasterNet Block and dynamic upsampling method to optimize the network structure, and designs an Inner-WIoU loss to improve the localization accuracy of small objects. Evaluations on the VisDrone2019 and UAVDT datasets illustrate that the LSOD-YOLOv8s model surpasses the original YOLOv8s in average precision at an IoU of 0.5 by 6.3% and 3.3%, respectively, while achieving a 75% reduction in model parameters. Compared to other advanced models, LSOD-YOLOv8s not only possesses the fewest parameters and highest average precision, but also significantly reduces false detection and miss detection rates, meeting the demands of real-time detection for UAVs.**

***Index Terms*—YOLOv8, small object detection, lightweight network, UAV, FasterNet**

I. Introduction

In recent years, the UAV industry has experienced rapid development and has been extensively adopted in military, agricultural, and traffic management fields due to its low cost, easy operation, and good maneuverability [1][2]. As the application fields continue to expand, higher demands are being placed on the performance of aerial image target detection algorithms. Currently, deep learning-based object detection algorithms are mainly divided into two categories: two-stage detection methods and one-stage detection methods. The R-CNN series [3]-[5] is a more typical two-stage object detection method, which first generates candidate regions and then extracts and classifies the features of each candidate region. Although this method performs excellently in object detection tasks, processing speed may become a limiting factor when faced with real-time requirements. The YOLO [6] series is a typical one-stage object detection method. It completes the classification of objects and the regression of bounding boxes through a single forward propagation. The core advantage of this method lies in its exceptional real-time performance, enabling efficient object detection on embedded devices and mobile platforms. However, it may lose some detection accuracy compared to two-stage detection methods. Due to the fact that drones capture images from a certain height above the ground, most objects in the images are small targets. Additionally, these targets are susceptible to environmental and weather conditions, resulting in an increased rate of missed detections. Consequently, the prevalent detection methodologies are not directly applicable to UAV image object detection tasks. In order to resolve this issue, this paper proposes a lightweight small object detection model for UAVs based on YOLOv8. This model solves the problem of traditional models being ineffective in detecting small targets.

The principal contributions of this study are outlined below:

Based on the characteristics of small targets, a small target detection layer was constructed to effectively avoid the loss of detailed features. Additionally, the redundant deep feature extraction modules in the baseline model were removed, making the model more lightweight.

Introducing the lightweight FasterNet Block to replace the bottleneck module in the original model’s backbone, ensuring efficient feature extraction while making the model more lightweight.

Using DySample as the upsampling method for the neck network. This method avoids the computationally expensive traditional dynamic convolution and constructs a lightweight upsampling process by generating point-wise dynamic scope factors through a point-wise sampling generator. It has fewer parameters and lower latency.

Wise-IoUv3 is introduced to combine it with the idea of Inner-IoU assisted bounding box. Inner-WIoU is able to dynamically adjust the weights of the loss values based on the location and size of the bounding box as well as the contextual information associated with the target, thus enhancing the model’s detection performance for small target samples. The addition of auxiliary bounding boxes allows the model to converge faster during training, achieving higher detection accuracy with fewer training iterations.

II. Related Work

A. Small Object Detection

In the task of object detection, small objects generally refer

Manuscript received May 14, 2024; revised September 17, 2024.

Huikai Li is a postgraduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (e-mail: lihuikai11@qq.com).

Jie Wu is an associate professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (*corresponding author to provide e-mail: wujie@ustl.edu.cn).

to an object that occupies a small area in an image or video frame, has fewer pixels, and has limited detail information. Small objects often have colors and textures similar to the background, or are relatively close to adjacent objects, making it challenging for the model to distinguish them from background noise. In complex scenes, densely packed small objects are highly prone to false detections or missed detections. Since UAVs often operate at high altitudes far from the ground and are limited by camera resolution, this results in small objects having extremely low resolution in the image and lacking sufficiently detailed features for identification. Furthermore, due to the movement of the drone itself and the movement of the target during flight, the position, size, and shape of the target may rapidly change in the image, thereby increasing the difficulty of real-time detection.

In recent years, numerous researchers have conducted studies on small object detection. Deng et al. [7] proposed an extended feature pyramid network (EFPN) to address the issues of information loss and ambiguity commonly encountered by traditional feature pyramid networks (FPNs) when dealing with small objects. The method introduces a small-scale feature fusion module to extract detailed information from low-level features, along with a cross-resolution distillation mechanism to enhance the network's ability to perceive details across different scales. These mechanisms effectively enhance the network's accuracy for small object detection and are validated on multiple datasets. Hong et al. [8] proposed a scale-selective pyramid network for tiny figure detection, which optimizes the process of extracting and recognizing features of tiny figures in UAV images. Its scale-selective module is able to extract features at multiple scales, with a special focus on those scale layers that contain tiny figures, thus increasing the sensitivity of object detection for tiny objects and significantly improving the tiny character detection accuracy. YOLOv4_Drone [9], based on YOLOv4 [10], introduces hollow convolutions and a lightweight subspace attention mechanism to enhance the detection accuracy of small objects amidst complex backgrounds in drone imagery. MCS-YOLOv4 [11] combines multi-scale contextual information and the Soft-CIOU loss function to further enhance the model's performance and stability in small object detection. TPH-YOLOv5 [12] is a deep learning model used for object detection in drone aerial photography scenarios. This model replaces the detection head of YOLOv5 with a Transformer [13] detection head and integrates the convolutional block attention module (CBAM) [14], thereby effectively improving the detection capability of small objects and demonstrating good performance in drone aerial photography scenarios. MS-YOLOv7 [15] is a multi-scale object detection model for aerial images taken by drones. This model integrates Swin Transformer [16], SPPFS pyramid pooling module, and CBAM attention mechanism based on YOLOv7 [17], thereby improving the detection accuracy of small objects in complex backgrounds. The current inadequacies in small object detection research mainly lie in insufficient feature representation, unresolved issues of sample imbalance, and difficulties in balancing computational costs with detection precision. Moreover, the robustness in complex backgrounds and generalization capabilities in real-world scenarios still require further enhancement.

B. Lightweight Neural Networks

Lightweight networks aim to optimize architecture and reduce the number of parameters without significantly sacrificing performance, resulting in high efficiency and low resource requirements. Many complex neural network structures, such as ResNet [18], DenseNet [19], and Transformer, have received widespread attention. These complex structures bring excellent feature extraction capabilities but have significantly expanded in network depth and width. They usually contain millions or even billions of parameters, which not only require more storage space but also demand high computational power. Detection algorithms need to balance high efficiency and light weight for platforms with limited computational resources, such as UAVs.

At present, the more classic lightweight neural networks such as MobileNet [20]-[22], ShuffleNet [23][24], and GhostNet [25], predominantly use depthwise separable convolution (DWConv). Although DWConv is effective in reducing FLOPs, in order to achieve functionality similar to conventional convolutions, the number of feature channels is usually increased to six times prior to using DWConv, resulting in more memory access times than conventional convolutions when using DWConv, leading to non-negligible delays and reduced overall calculation speed. In order to resolve this issue, Chen et al. [26] proposed a lightweight network called FasterNet. The main module of this network, the FasterNet Block, consists of Partial Convolution (PConv) with Pointwise Convolution. The PConv module divides the channels into two groups according to a certain ratio, using either the first or the last channel consecutively to compute the entire feature map. The number of channels in the input and output feature maps remains unchanged, thereby fully utilizing the redundancy of the features. This allows FasterNet to maintain excellent feature extraction capability while reducing the number of parameters.

C. YOLO v8

YOLOv8 is the latest iteration in the YOLO series of algorithms, proposed by the original team of authors who developed YOLOv5. YOLOv8 offers five different model variants of varying sizes and complexities, based on different scaling factors. As the model size increases, the accuracy continuously improves, allowing for the selection of network models with varying depths and widths according to mission requirements. Due to the hardware limitations of drone equipment, the YOLOv8s model, which is small in size and high in precision, is more suitable for drone object detection tasks. The Backbone section of YOLOv8 adopts the new and efficient C2f feature extraction module and continues to use the spatial pyramid pooling fusion module from YOLOv5. This module can extract features at multiple scales, thereby enhancing the spatial contextual understanding of the target. The Neck section draws on the ideas of PANet, using both bottom-up and top-down feature aggregation methods to combine deep semantic features with shallow detailed features, generating richer feature representations to enhance the detection performance for targets of various scales. The Head section introduces an anchor-free mechanism to

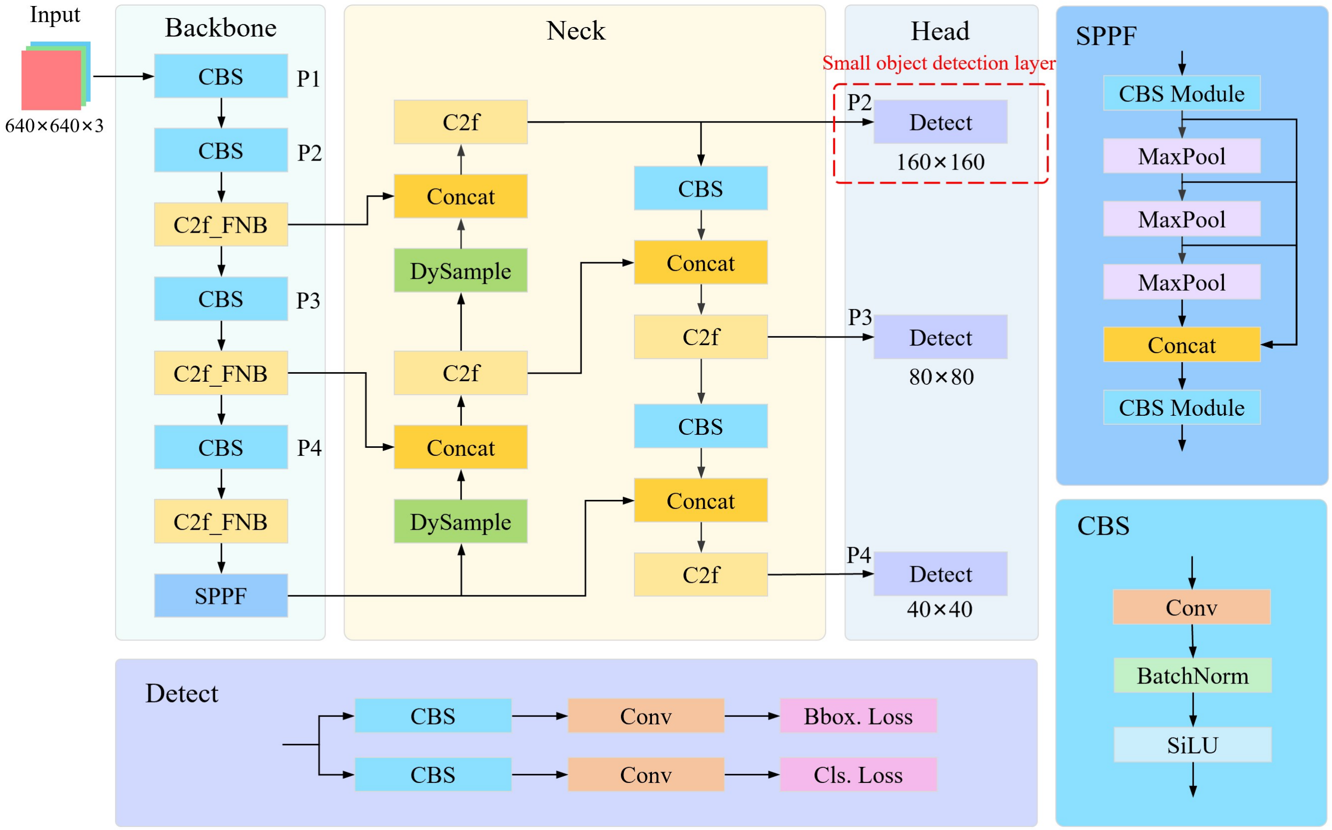


Fig. 1. Overall structure of the LSOD-YOLOv8s model.

simplify target localization and improves the traditional non-maximum suppression (NMS) to adaptive NMS, which can dynamically adjust the threshold to adapt to different target densities and categories, thus improving detection accuracy and reducing false detections and missed detections.

III. Method

A. LSOD-YOLOv8s Model Structure

LSOD-YOLOv8s model is an improved, lightweight small object detection model based on YOLOv8. Fig. 1 shows its overall structure. This model optimizes YOLOv8 in three aspects. First, an additional small object detection layer is included because the detailed information of small objects is more completely retained in the shallow feature layer. This detection layer can effectively extract the feature information of small objects, thereby improving detection accuracy. Second, the C2f module in the backbone is redesigned and a lightweight FasterNet Block is introduced, forming the C2f_FNB module. Lastly, the lightweight and efficient dynamic upsampling method DySample, which replaces the conventional upsampling layer.

B. Small Object Detection Layer

The backbone network of YOLOv8 has a deep hierarchical structure, which helps in extracting high-level semantic features. However, small objects often contain more detailed features, which tend to become more blurred or even lost after multiple downsamplings, especially when the network has a deep hierarchy. In such cases, the spatial resolution of the features is too low to effectively capture the detailed information of small objects, leading to a decline in the performance of small object detection. To address the aforementioned issues, the P5 detection layer, which retains less feature information for small objects in the original model, and the deep feature extraction module of the backbone network, were removed to reduce redundant computation, as shown in Fig. 2(a). Furthermore, a P2 detection layer, suitable for small targets, was added in the shallow feature extraction stage where the feature information is richer, as shown in Fig. 2(b). The shallow feature maps have larger dimensions, so the detailed information of small objects is retained more completely. By the lateral connections of the FPN, shallow feature maps in the backbone network that have high resolution but less semantic information are fused with upsampled deep feature maps. This fusion effectively combines deep semantic information with shallow detail information, enabling the generated feature maps to have both rich contextual semantics and local details. As a result, the newly added small object detection layer can more accurately locate and identify small objects, avoiding the issue of feature blurriness caused by relying solely on deep features.

C. C2f_FNB Module

To minimize redundant computation and memory access, while also decreasing the number of model parameters, the FasterNet Block from the lightweight network FasterNet is introduced to form the C2f_FNB module, replacing the feature extraction module of the original model backbone. The structure of this modification is illustrated in Fig. 3. The C2f_FNB module adopts the idea of feature extraction and

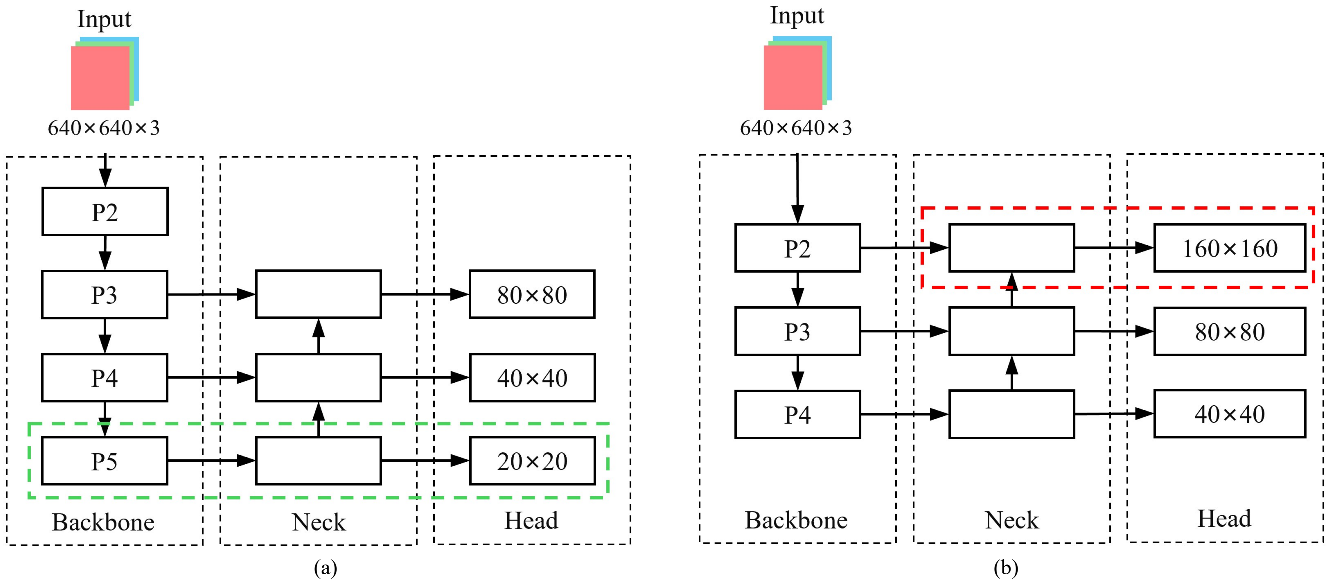


Fig. 2. (a) YOLOv8s detection layer structure; (b) LSOD-YOLOv8s detection layer structure. The green dashed box indicates the removed deep feature extraction module as well as the P5 detection layer, and the red dashed box indicates the added P2 small object detection layer.

diversion from the cross stage partial network, as well as the concept of residual structure. It uses the FasterNet Block to replace the Bottleneck module as the primary gradient flow branch. The stacking number of FasterNet Blocks is controlled by the parameter 'n', which varies with different scaling coefficients for models of different scales.

The FasterNet Block combines two efficient operators, PConv and pointwise convolution (PWConv). PConv avoids unnecessary calculations on invalid or padding regions, allowing the model to focus more on processing useful feature information, thereby extracting spatial features more effectively. PWConv is used for feature transformation and channel adjustment of the intermediate layers, effectively reducing computational complexity and the number of parameters. The FLOPs calculation formula for PConv is as follows:

$$FLOPs = c_p^2 \times h \times w \times k^2, \tag{1}$$

where c_p represents either the first or the last consecutive channel. When it accounts for one-quarter of the total channels, the FLOPs for PConv are one-sixteenth of those for normal convolution. This means smaller memory access and fewer model parameters, making the model more lightweight while maintaining its ability to effectively extract target feature information.

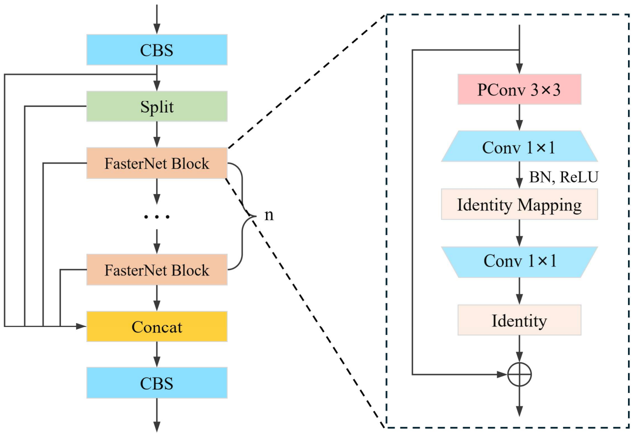


Fig. 3. Structure of C2f_FNB Module.

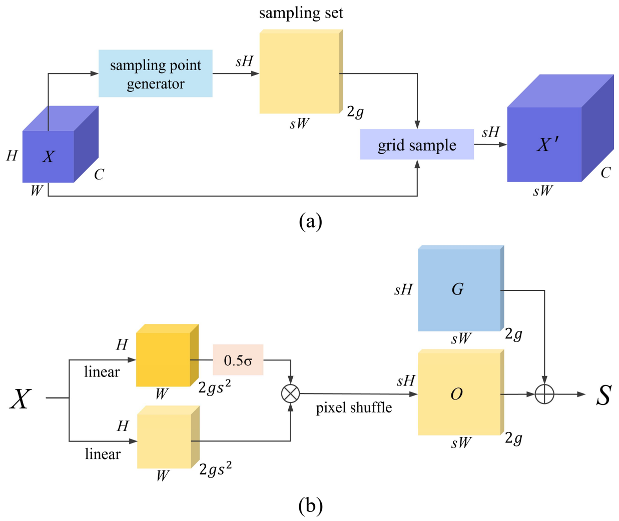


Fig. 4. (a) Structure of dynamic upsampling; (b) Sampling point generator. 'σ' denotes the sigmoid function.

D. DySample

YOLOv8 uses nearest neighbor interpolation as the upsampling method, enhancing the resolution of the feature map by directly copying the values of neighboring pixels. Small objects usually contain fewer salient features, and if nearest-neighbor interpolation is used for image scaling, vital detail information may be lost, making the already insignificant features difficult to identify and further increasing the difficulty of small object detection. To this end, LSOD-YOLOv8s introduces an efficient dynamic upsampling method called DySample [27], as shown in Fig. 4(a). This method distinguishes itself from the kernel-based dynamic upsampling method [28] by reconstructing the upsampling process from the perspective of point sampling. It features fewer parameters, FLOPs, and delays, thus facilitating an increase in image resolution without imposing an additional burden. The sampling point generator in DySample, depicted in Fig. 4(b), processes the input feature X through two linear layers to generate the initial upsampling position. To prevent the overlapping of point sampling positions from affecting the prediction near the boundary between the object and the background, one of the initial offsets is multiplied by the learnable parameter dynamic scope factor. The dynamic scope factor takes values in the range of [0, 0.5], which ensures that the sampled points do not move out of a reasonable range. Finally, the offset grid *O* is generated by pixel shuffling and added to the original grid *G* to get the set of sampling points *S*, as shown in

$$O = 0.5 \cdot \text{sigmoid}(W_1 X + b_1) \cdot (W_2 X + b_2)) \,, \tag{2}$$

$$S = G + O \,. \tag{3}$$

E. Inner-WIoU Loss Function

During training, YOLOv8 employs various loss functions to optimize model performance. Among them, the regression loss functions are the distribution focal loss and CIoU loss. The CIoU is defined as

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b^{pred}, b^{gt})}{c^2} + \alpha V \,, \tag{4}$$

$$\alpha = \frac{V}{(1 - IoU) + V} \,, \tag{5}$$

$$V = 4 \times \frac{1}{\pi^2} \times \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \,, \tag{6}$$

where b^{pred} and b^{gt} represent the center points of the predicted bounding box and the ground truth box, respectively. α is a trade-off parameter, and V is used to measure whether the aspect ratios of the two boxes match.

Although CIoU enhances the convergence speed and positioning accuracy by considering the center distance and aspect ratio of the bounding boxes, it becomes less sensitive to adjustments in the position of the bounding boxes when the aspect ratio is the same. This is especially true in cases of high IoU, where the gradient approaches zero. This situation weakens the model's capability in fine-tuning and further optimizing the predicted boxes. For targets with significant size differences, such as very large and very small targets, CIoU appears to be inadequate. Unlike traditional IoU loss, Wise-IoU [29] does not lead to gradients approaching zero. It can adopt different strategies to balance the gradient updates in situations of low IoU and high IoU, allowing the model to obtain effective gradients throughout the training process. The formula for WIoU v1 loss is shown in

$$L_{WIoUv1} = R_{WIoU} L_{IoU} \,, \tag{7}$$

$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)^*}\right) , \tag{8}$$

where W_g and H_g represent the dimensions of the smallest enclosing box. To prevent R_{WIoU} from producing gradients that could impede convergence, W_g and H_g are separated from the computation graph (i.e., they do not participate in backpropagation, denoted by *).

WIoU v3 introduces a dynamic non-monotonic focusing mechanism that adjusts gradient allocation based on the quality of anchor boxes, as shown in

$$\beta = \frac{L_{IoU}^*}{\overline{L_{IoU}}} \in [0, +\infty) \,, \tag{9}$$

$$L_{WIoUv3} = r L_{WIoUv1}, r = \frac{\beta}{\delta \alpha^{\beta - \delta}} \,, \tag{10}$$

where β represents the outlier degree. r denotes the gradient gain factor. α and δ are hyperparameters, with α controlling the shape of the gradient gain curve, and δ specifying the peak position of the gradient gain curve.

When β is relatively small, it indicates that this is a high-quality anchor box; conversely, it might be a low-quality anchor box or even an outlier. Low-quality anchor boxes are assigned smaller gradient gains to avoid negatively impacting the model, while high-quality anchor boxes also reduce gradient competition. When β equals δ, the gradient gain r reaches its maximum value, making the model focus the most and perform the greatest gradient updates. Thus, WIoUv3 better balances attention to both high-quality and regular-quality anchor boxes at different stages of training, thereby enhancing the stability and generalization ability of the model.

Inner-IoU [30] is an enhanced IoU loss function based on auxiliary bounding boxes, designed to accelerate bounding box regression and enhance localization accuracy. This method controls the generation of auxiliary bounding boxes of different sizes for calculating IoU loss by adjusting the scaling ratio, as shown in

$$b_l^{gt} = x_c^{gt} - \frac{w^{gt} \times ratio}{2}, b_r^{gt} = x_c^{gt} + \frac{w^{gt} \times ratio}{2} \,, \tag{11}$$

$$b_t^{gt} = y_c^{gt} - \frac{h^{gt} \times ratio}{2}, b_b^{gt} = y_c^{gt} + \frac{h^{gt} \times ratio}{2} \,, \tag{12}$$

$$b_l^{pred} = x_c - \frac{w \times ratio}{2}, b_r^{pred} = x_c + \frac{w \times ratio}{2} \,, \tag{13}$$

$$b_t^{pred} = y_c - \frac{h \times ratio}{2}, b_b^{pred} = y_c + \frac{h \times ratio}{2} \,, \tag{14}$$

$$\begin{gathered} inter = (\min(b_r^{gt}, b_r^{pred}) - \max(b_l^{gt}, b_l^{pred})) \times \\ (\min(b_b^{gt}, b_b^{pred}) - \max(b_t^{gt}, b_t^{pred})) \end{gathered} \,, \tag{15}$$

$$union = w^{gt} \times h^{gt} \times ratio^2 + w \times h \times ratio^2 - inter \,, \tag{16}$$

$$IoU^{inner} = \frac{inter}{union} \,, \tag{17}$$

$$L_{Inner-IoU} = 1 - IoU^{inner} \,, \tag{18}$$

where $\text{ratio} \in [0.5, 1.5]$. In different scenarios, appropriately choosing the ratio can provide the model with better generalization capabilities. For example, in detection tasks with high precision requirements, a smaller ratio can be used to optimize the fine bounding box positioning; in scenarios requiring rapid convergence, using a larger ratio helps handle difficult samples.

Inner-IoU can refine the regression process of high-IoU samples through adjusting the size of the auxiliary bounding box, thereby enhancing localization accuracy. Meanwhile, WIoUv3's dynamic focusing mechanism assigns greater weight to small objects, further strengthening the attention on small objects. Therefore, combining the two can significantly improve the localization accuracy of small targets, reduce the incidence of missed detections, and thus enhance the detection precision of small targets. The expression for Inner-WIoU is given by equation (19).

$$L_{Inner-WIoU} = rR_{WIoU} L_{Inner-IoU} \tag{19}$$

IV. Experiments

A. Experiment Overview

(1) Dataset

This study utilizes the VisDrone2019 [31] and the UAVDT [32] as the experimental datasets. Both are typical drone aerial image datasets used for object detection tasks. The VisDrone2019 dataset serves as the primary experimental dataset, while the UAVDT dataset facilitates further validation of the model's applicability.

The VisDrone2019 dataset consists of 10,209 images, with 6,471 used for training, 548 for validation, and 3,190 for testing. These images were captured by drones of different models under various environmental conditions. They encompass a broad range of real-world scenes, including various terrains such as cities and countryside, and various target categories like pedestrians, vehicles, bicycles, motorcycles, etc. The dataset also includes challenging factors such as weather changes, target occlusions, and scale variations.

The UAVDT dataset contains 40,735 images, all of which are extracted frames from videos captured by drones. Of these, 24,206 images are designated for the training set, while 16,529 images are allocated to the validation set. Due to the dataset covering diverse scenarios, such as different weather conditions, flying altitudes, camera perspectives, and vehicle categories, the targets in this dataset may face challenges such as scale variations, occlusions, cluttered backgrounds, etc., posing high demands on the algorithm's robustness and accuracy.

(2) Experimental parameters and evaluation indicators

All experiments were conducted using an NVIDIA 4070 (12GB) graphic processor for both training and inference. We chose Pytorch 2.2 as the deep learning framework and used Python 3.8 and CUDA 11.8. During the training phase, the size of all input images was set to 640×640. The initial learning rate was set to 0.01, the momentum parameter to 0.937, the weight decay coefficient to 5×10^{-4}, and the batch size to 8, with the stochastic gradient descent (SGD) optimizer. All training was done from scratch without the use of pre-trained weights, with a total of 200 training iterations.

To evaluate the detection performance of the proposed model, we use Precision, Recall, mAP (mean Average Precision), Parameters, FLOPs, and FPS as evaluation metrics.

B. Experimental Results

(1) Comparison with YOLOv8s

To verify the effectiveness of the proposed model for small target detection, we compared the detection accuracy of the proposed model with the YOLOv8s model on each category in the VisDrone2019 dataset. The experimental results, as shown in Table I, indicate that the detection accuracy of the proposed model is higher than that of the YOLOv8s model across 10 categories. Notably, there is a significant increase in average precision for extremely small objects, including Pedestrian, People, and Motor, with an improvement of 6.3% in mAP at IoU of 0.5, reaching 45.2%. These results fully validate the effectiveness of LSOD-YOLOv8s for small object detection tasks.

TABLE I
DETECTION RESULTS FOR EACH CATEGORY ON THE VISDRONE2019 DATASET

Category	YOLOv8s	LSOD-YOLOv8s
People	31.6	42.6
Pedestrian	42.1	52.4
Bicycle	13.0	17.9
Van	44.9	49.2
Car	79.5	84.2
Truck	36.4	39.1
Bus	55.8	63.0
Motor	43.1	52.9
Tricycle	28.0	32.3
Awning-Tricycle	15.5	18.4
All	38.9	45.2

(2) Comparison with other versions of YOLO

To further verify the superiority and applicability of the model, we used the same experimental parameters to compare the LSOD-YOLOv8s model with other versions of the YOLO model on the VisDrone2019 dataset and the UAVDT dataset. The other versions of YOLO network models used in the experiment include: YOLOv3 [33], which is based on the Darknet-53 architecture; YOLOv5, which adopts CSPNet and Focus structures for a more concise and efficient network design; YOLOv6 [34], which is designed with a more concise and efficient decoupled head; YOLOv7, which utilizes the efficient ELAN structure, as well as different scales of YOLOv8.

The comparison results on the VisDrone2019 dataset are shown in Table II. The experimental results indicate that, compared to previous versions of the YOLO model, the YOLOv8n model, which has fewer parameters, performs well in terms of detection accuracy and has the lowest FLOPs. The YOLOv3-tiny has the fastest detection speed, but the accuracy is too low to perform the task of UAV object detection. The LSOD-YOLOv8s model not only has the fewest parameters but also the highest detection accuracy among all models. The FLOPs have decreased compared to the YOLOv8s model, which has improved the model's operational efficiency to some extent. Although the FPS metric is relatively lower compared to other YOLO models, it still satisfies the performance requirements for real-time drone detection.

The comparison results on the UAVDT dataset are shown in Table III. The experimental results indicate that the LSOD-YOLOv8s model performs the best in terms of detection accuracy. Compared to the YOLOv8s model, the mAP50 value has increased by 3.3%, while the FPS is second only to the best-performing YOLOv3. This further demonstrates that the network model proposed in this paper has good applicability.

TABLE II
COMPARISON WITH OTHER VERSIONS OF THE YOLO MODEL ON THE VISDRONE2019 DATASET

Model	Precision (%)	Recall (%)	$mAP_{0.5}$(%)	$mAP_{0.5:0.95}$(%)	Params(M)	FLOPs(G)	FPS
YOLOv3-tiny	39.5	24.1	23.6	12.9	12.1	18.9	245.2
YOLOv5n	44.1	32.1	32.2	18.5	2.5	7.1	105.6
YOLOv5s	48.6	38.0	38.5	22.8	9.1	23.8	159.2
YOLOv6s	39.8	29.4	29.1	17.0	16.3	44.2	151.7
YOLOv7-tiny	42.3	37.8	33.9	17.4	6.0	13.3	85.2
YOLOv8n	44.5	33.1	33.2	19.1	3.0	8.1	164.1
YOLOv8s	48.0	39.2	38.9	23.3	11.1	28.5	152.7
LSOD-YOLOv8s	54.9	43.4	45.2	27.4	2.7	23.5	126.9

TABLE III
COMPARISON WITH OTHER VERSIONS OF THE YOLO MODEL ON THE UAVDT DATASET

Model	Precision (%)	Recall (%)	$mAP_{0.5}$(%)	$mAP_{0.5:0.95}$(%)	Params(M)	FLOPs(G)	FPS
YOLOv3-tiny	49.3	39.7	38.3	23.2	12.1	18.9	328.3
YOLOv5n	47.0	37.7	40.2	25.5	2.5	7.1	238.2
YOLOv5s	50.9	40.7	45.5	30.6	9.1	23.8	242.9
YOLOv6s	37.8	54.3	44.3	30.0	16.3	44.2	243.2
YOLOv7-tiny	42.5	46.8	43.0	24.3	6.0	13.3	164.4
YOLOv8n	44.8	42.7	39.5	24.8	3.0	8.1	245.2
YOLOv8s	49.1	40.8	45.0	30.4	11.1	28.5	251.3
LSOD-YOLOv8s	55.9	49.6	48.3	30.7	2.7	23.5	259.3

TABLE IV
COMPARISON WITH OTHER SMALL OBJECT DETECTION MODELS

Model	$mAP_{0.5}$(%)	$mAP_{0.5:0.95}$(%)	Params(M)	FLOPs(G)	FPS
UN-YOLOv5s [35]	40.5	22.5	/	37.4	/
LAI-YOLOv5s [36]	40.4	/	6.3	29.0	51.3
CPAM-YOLO [37]	43.0	25.0	27.5	106.4	/
DM-YOLOX [38]	41.9	/	9.6	27.9	78.3
PDWT-YOLO [39]	41.2	22.5	6.4	24.2	/
Li et al. [40]	42.2	/	9.6	/	167.0
MPE-YOLO [41]	37.0	21.4	4.4	/	/
LSOD-YOLOv8s	45.2	27.4	2.7	23.5	126.9

(3) Comparison with other small object detection models

Table IV compares the experimental results of the LSOD-YOLOv8s model with seven other small object detection models on the VisDrone2019 dataset. Due to missing Precision and Recall data for some models, these evaluation metrics were excluded in this experiment. The data in the table indicates that the CPAM-YOLO model shows considerable improvement in detecting small objects; however, its extensive computational demand makes it unsuitable for target detection tasks on drone platforms. The MPE-YOLO model performs well in terms of lightweight characteristics, but the overall detection accuracy is unsatisfactory. The LSOD-YOLOv8s model proposed in this paper not only features a lightweight design but also exhibits excellent detection accuracy.

(4) Comparison of different loss functions

To validate the convergence of the proposed Inner-WIoU and its improvement on model localization accuracy, we conducted comparative experiments using the YOLOv8s model with multiple loss functions on the VisDrone2019 dataset. The loss functions involved in the comparison include CIoU, EIoU, SIoU, and WIoUv3. The results are shown in Table V. Both EIoU and SIoU improved the average precision by 0.2% compared to the default CIoU. Among the various classical loss functions, WIoUv3 performed the best, with its average precision improving by 0.7% compared to CIoU. The proposed Inner-WIoU achieves the highest mean precision, with a mean precision reaching 40.1% when the ratio value is 1.2. This demonstrates that incorporating the auxiliary bounding box concept into Inner-WIoU can significantly improve the localization accuracy of small targets, thereby enhancing the detection precision of small targets.

TABLE V
COMPARISON OF DETECTION RESULTS OF DIFFERENT LOSS FUNCTIONS ON THE YOLOV8S MODEL

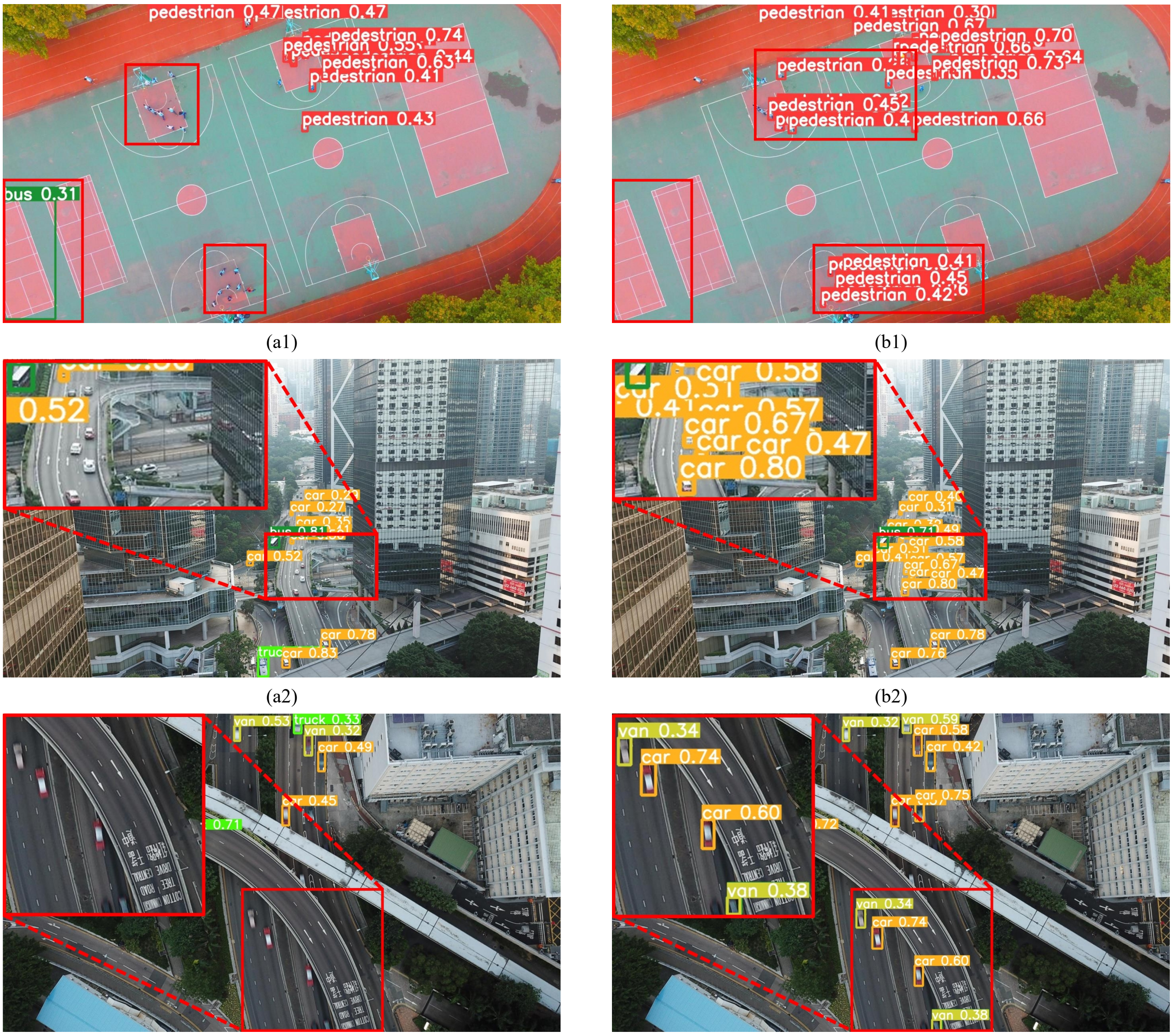
Loss Function	$mAP_{0.5}$(%)	$mAP_{0.5:0.95}$(%)
CIoU	38.9	23.3
EIoU	39.1	23.2
SIoU	39.1	23.3
WIoUv3	39.6	23.4
Inner-WIoU(ratio=1.15)	39.7	23.3
Inner-WIoU(ratio=1.20)	40.1	23.7
Inner-WIoU(ratio=1.25)	39.7	23.6

C. Ablation Experiment

To evaluate the performance of each module, this section conducts ablation experiments on the proposed model using the VisDrone2019 dataset. This experiment enumerates in detail the impact of the added P2 detection layer, C2f_FNB module, DySample dynamic downsampling layer, and the

TABLE VI
ABLATION EXPERIMENTS ON THE VISDRONE2019 DATASET

Baseline	P2	Inner-WIoU	DySample	C2f_FNB	$mAP_{0.5}$(%)	$mAP_{0.5:0.95}$(%)	Params(M)
YOLOv8s					38.9	23.3	11.1
	√				43.6	26.5	3.3
	√	√			44.5	26.8	3.3
	√	√	√		44.7	26.9	3.3
	√	√	√	√	45.2	27.4	2.7



(a1)

(b1)

(a2)

(b2)

(a3)

(b3)

Fig. 5. Visual comparison of detection results on the VisDrone2019 test set: (a1-a3) YOLOv8s, (b1-b3) LSOD-YOLOv8s

Inner-WIoU loss function on the model performance. Table VI shows the performance evaluation results of the baseline model after combining different modules. Experimental results indicate that after adding the P2 small target detection layer, the model's feature extraction capability for small targets is significantly improved, and the mAP value is also noticeably enhanced. In addition, removing the deep feature extraction module and the P5 detection layer from the baseline model substantially reduces the number of parameters in the model. Then, replacing the loss function of the baseline model with Inner-WIoU resulted in a 0.9% increase in mAP0.5, which is attributable to the Inner-WIoU loss function's ability to dynamically adjust gradient gains. Although the addition of the DySample upsampling method results slightly increases the number of parameters, it effectively avoids the loss of detail information compared to the traditional feature downsampling modules. The FasterNet Block seeks to minimize feature redundancy and reduce the number of model parameters. This effect is verified in the last row of the table, showing an improvement in detection accuracy while the number of parameters is reduced.

D. *Visualization Analysis*

In addition to the aforementioned comparison of experimental data, this section will also visualize the detection results to intuitively demonstrate the small target detection performance of the LSOD-YOLOv8s model. We selected three representative sets of comparative images from the experimental results of the VisDrone2019 test dataset, which include a crowded basketball court, dense traffic, and scenes where detection targets are easily obscured.

In Figure 5, the areas of comparison are marked with red rectangular boxes. From the first set of basketball court images, it is evident that the YOLOv8 model incorrectly detects the red block on the basketball court as a bus and misses detecting two crowd areas. However, the improved model proposed in this paper accurately detects the crowd in the basketball court and avoids any false detections. The same is true for the second and third sets of images; in dense traffic, every vehicle and its type were accurately identified. Overall, the LSOD-YOLOv8s model proposed in this paper enhances the detection performance of small objects in complex backgrounds, improving the issues of false detections and missed detections, thereby proving the effectiveness of the model.

In addition, the confusion matrix of the LSOD-YOLOv8s model on the VisDrone2019 dataset is shown in Fig. 6. In this matrix, the rows represent the predicted classes by the model, the columns represent the actual classes, and the values on the diagonal indicate correct predictions by the model, i.e., the predicted classes are consistent with the actual classes. Through observation, it can be found that the model performs best in detecting cars, while categories such as bicycles, tricycles, and people are more likely to be misjudged as background. The primary reason for this phenomenon is that the model might be influenced by factors such as resolution, occlusion, or target size when detecting these extremely small targets, leading to an inability to effectively distinguish small targets from the background. Additionally, the dataset used for training has an uneven distribution in terms of the number and position of each category, with fewer samples of extremely small targets and significant positional deviations in the images. Consequently, the model encounters difficulties in adequately learning the features of these extremely small targets during training.

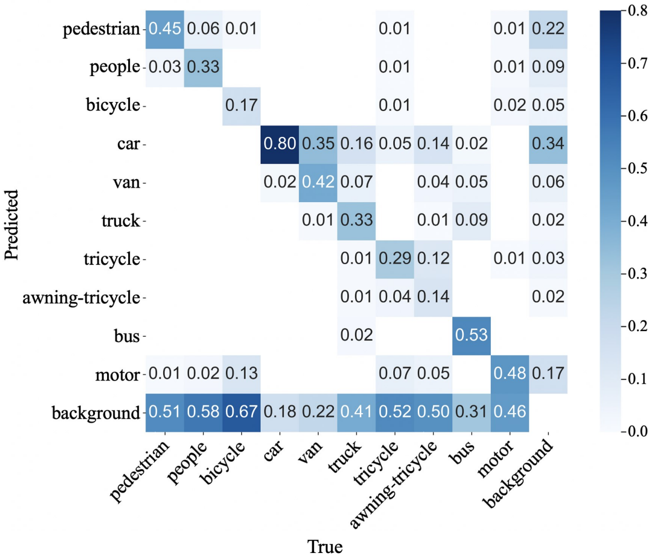


Fig. 6. The confusion matrix obtained by LSOD-YOLOv8s on the VisDrone2019 dataset.

V. Conclusions

In the task of small object detection, the detection efficiency of traditional models is greatly reduced due to factors such as target size, occlusion, and complex environmental backgrounds. Therefore, this paper proposes a lightweight small object detection model for aerial images captured by drones. Firstly, considering the characteristics of small targets, a small object detection layer was constructed, and the redundant deep feature extraction module in the baseline model was removed, making the model more lightweight. Secondly, the feature extraction module of the YOLOv8 backbone was improved, and a lightweight Faster-Net Block was introduced to construct a new C2f_FNB module. Meanwhile, in the neck part of the model, the traditional upsampling method was replaced with the DySample dynamic upsampling method. Lastly, Inner-WIoU was used as the bounding box regression loss function, dynamically adjusting the loss weights for targets of different sizes by evaluating the quality of anchor boxes through outlier assessment, thereby enhancing the model's generalization and localization accuracy. The comparative and ablation experiments conducted on the VisDrone2019 dataset substantiate the effectiveness of LSOD-YOLOv8s. This model strikes an optimal balance between detection accuracy and the number of parameters, significantly reducing the missed detection rate of small targets. Nevertheless, there remains potential for improvement in the detection accuracy of extremely small targets. Future research will continue to optimize the detection accuracy and efficiency of the model, further balancing detection speed and accuracy to address more complex small object detection scenarios in drones.

References

[1] Yuqi Han, Huaping Liu, Yufeng Wang, and Chunlei Liu, "A comprehensive review for typical applications based upon unmanned aerial vehicle platform," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, 2022, pp. 9654–9666.

[2] 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, pp1058-1073, 2023.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, 23-28 June, 2014, pp. 580–587.

[4] Ross Girshick, "Fast R-CNN," Proceedings of the IEEE International Conference on Computer Vision 2015, 7-13 December, 2015, pp. 1440–1448.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no.6, 2016, pp. 1137–1149.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, 27-30 June, 2016, pp. 779–788.

[7] Chunfang Deng, Mengmeng Wang, Liang Liu, Yong Liu, and Yunliang Jiang, "Extended feature pyramid network for small object detection," IEEE Transactions on Multimedia, vol. 24, 2021, pp. 1968–1979.

[8] Mingbo Hong et al., "SSPNet: Scale selection pyramid network for tiny person detection from UAV images," IEEE Geoscience and Remote Sensing Letters, vol. 19, 2021, pp. 1–5.

[9] Li Tan, Xinyue Lv, Xiaofeng Lian, and Ge Wang, “YOLOv4_Drone: UAV image target detection based on an improved YOLOv4 algorithm,” Computers & Electrical Engineering, vol. 93, 2021, p. 107261.

[10] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” arXiv:2004.10934, 2020.

[11] Shu-Jun Ji, Qing-Hua Ling, and Fei Han, “An improved algorithm for small object detection based on YOLO v4 and multi-scale contextual information,” Computers and Electrical Engineering, vol. 105, 2023, p. 108490.

[12] Xingkui Zhu, Shuchang Lyu, Xu Wang, and Qi Zhao, “TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios,” Proceedings of the IEEE/CVF International Conference on Computer Vision 2021, 10-17 October, 2021, pp. 2778–2788.

[13] Ashish Vaswani et al., “Attention is all you need,” Advances in Neural Information Processing Systems, vol. 30, 2017.

[14] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, “CBAM: Convolutional Block Attention Module,” Proceedings of the European Conference on Computer Vision 2018, 8-14 September, 2018, pp. 3–19.

[15] LiangLiang Zhao, and MinLing Zhu, “MS-YOLOv7: YOLOv7 based on multi-scale for object detection on UAV aerial photography,” Drones, vol. 7, no.3, 2023, p. 188.

[16] Ze Liu et al., “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows,” Proceedings of the IEEE/CVF International Conference on Computer Vision 2021, 10-17 October, 2021, pp. 10012–10022.

[17] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023, 18-22 June, 2023, pp. 7464–7475.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep Residual Learning for Image Recognition,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, 27-30 June, 2016, pp. 770–778.

[19] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger, “Densely Connected Convolutional Networks,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, 21-26 July, 2017, pp. 4700–4708.

[20] Andrew G. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv:1704.04861, 2017.

[21] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, 18-22 June, 2018, pp. 4510–4520.

[22] Andrew Howard et al., “Searching for MobileNetV3,” Proceedings of the IEEE/CVF International Conference on Computer Vision 2019, 27 October - 2 November, 2019, pp. 1314–1324.

[23] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun, “ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, 18-22 June, 2018, pp. 6848–6856.

[24] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun, “ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design,” Proceedings of the European Conference on Computer Vision 2018, 8-14 September, 2018, pp. 116–131.

[25] Kai Han et al., “GhostNet: More Features from Cheap Operations,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, 14-19 June, 2020, pp. 1580–1589.

[26] Jierun Chen et al., “Run, Don’t Walk: Chasing Higher FLOPS for Faster Neural Networks,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023, 18-22 June, 2023, pp. 12021–12031.

[27] Wenze Liu, Hao Lu, Hongtao Fu, and Zhiguo Cao, “Learning to Upsample by Learning to Sample,” Proceedings of the IEEE/CVF International Conference on Computer Vision 2023, 2-6 October, 2023, pp. 6027–6037.

[28] Jiaqi Wang et al., “CARAFE: Content-Aware ReAssembly of FEatures,” Proceedings of the IEEE/CVF International Conference on Computer Vision 2019, 27 October - 2 November, 2019, pp. 3007–3016.

[29] Zanjia Tong, Yuhang Chen, Zewei Xu, and Rong Yu, “Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism,” arXiv:2301.10051, 2023.

[30] Hao Zhang, Cong Xu, and Shuaijie Zhang, “Inner-IoU: More Effective Intersection over Union Loss with Auxiliary Bounding Box,” arXiv:2311.02877, 2023.

[31] Dawei Du et al., “VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results,” Proceedings of the IEEE/CVF International Conference on Computer Vision 2019, 27 October - 2 November, 2019.

[32] Dawei Du et al., “The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking,” Proceedings of the European Conference on Computer Vision 2018, 8-14 September, 2018, pp. 370–386.

[33] Joseph Redmon, and Ali Farhadi, “YOLOv3: An Incremental Improvement,” arXiv:1804.02767, 2018.

[34] Chuyi Li et al., “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications,” arXiv:2209.02976, 2022.

[35] Junmei Guo, Xingchen Liu, Lingyun Bi, Haiying Liu, and Haitong Lou, “UN-YOLOv5s: A UAV-Based Aerial Photography Detection Algorithm,” Sensors, vol. 23, no.13, 2023, p. 5907.

[36] Lixia Deng et al., “Lightweight aerial image object detection algorithm based on improved YOLOv5s,” Scientific Reports, vol. 13, no.1, 2023, p. 7817.

[37] Huixin Wu, Yang Zhu, and Mengdi Cao, “An algorithm for detecting dense small objects in aerial photography based on coordinate position attention module,” IET Image Processing, vol. 18, no.7, 2024, pp. 1759-1767.

[38] Xiangyu Li, Fengping Wang, Wei Wang, Yanjiang Han, and Jianyang Zhang, “DM-YOLOX aerial object detection method with intensive attention mechanism,” The Journal of Supercomputing, vol. 80, 2024, pp. 12790-12812.

[39] Linhua Zhang et al., “Improved object detection method utilizing yolov7-tiny for unmanned aerial vehicle photographic imagery,” Algorithms, vol. 16, no.11, 2023, p. 520.

[40] Yiting Li, Qingsong Fan, Haisong Huang, Zhenggong Han, and Qiang Gu, “A modified YOLOv8 detection network for UAV aerial image recognition,” Drones, vol. 7, no.5, 2023, p. 304.

[41] Jia Su, Yichang Qin, Ze Jia, and Ben Liang, “MPE-YOLO: Enhanced Small Target Detection in Aerial Imaging,” Scientific Reports, vol. 14, no.1, 2024, p. 17799.