Electronic Research Archive

https://www.aimspress.com/journal/ERA

ERA, 32(7): 4563–4580.
DOI: 10.3934/era.2024207
Received: 01 May 2024
Revised: 15 July 2024
Accepted: 19 July 2024
Published: 24 July 2024

Research article

STD-YOLOv8: A lightweight small target detection algorithm for UAV perspectives

Dong Wu[1,2], Jiechang Li[1,2,*], and Weijiang Yang[1,2]

[1] Key Laboratory of Advanced Manufacturing and Automation Technology, Guilin University of Technology, Education Department of Guangxi Zhuang Autonomous Region, Guilin 541006, China
[2] College of Mechanical and Control Engineering, Guilin University of Technology, Guilin 541006, China

* **Correspondence:** Email: ljiechang@glut.edu.cn.

Abstract: When recognizing targets by unmanned aerial vehicles (UAVs), problems such as small size, dense dispersion, and complex background are likely to occur, resulting in low recognition rates. In order to solve the above problems, this work proposed a lightweight small target detection algorithm based on the YOLOv8n: STD-YOLOv8 algorithm. First, the regression problem of small targets in different training periods was optimized, the penalty term in the original loss was improved, and a new LIoU loss function was proposed, so that the size of the penalty term could be dynamically adjusted before and after training, thus improving the performance of the algorithm. Second, in order to better adapt to the small target scale and enhance the ability of small target feature acquisition, the SPD-Conv module was integrated in the backbone network, replacing the original stepwise convolutional layer and pooling layer, so as to solve the problems of loss of fine-grained information and low efficiency of feature representation existing in the current convolutional neural network (CNN) structure. In the neck part, nearest-neighbor upsampling was replaced by the feature reassembly assembly of features operator CARAFE (content-aware reassembly of features), which enabled the model to aggregate contextual information in a larger perceptual field and enhanced the feature representation in the neck. Finally, validation experiments were conducted by comparing different algorithms under the same VisDrone-2021 dataset. The results of the ablation experiments show that the algorithms proposed in this thesis have improved the recall (R), mAP50, and mAP95 by 4.7, 5.8 and 5.7%, respectively, compared with YOLOv8n. The results of the model generalization experiments on the TinyPerson dataset show that the algorithm in this paper has superior small target detection performance with only 1.2 M model parameters (1 M = 10^6).

Keywords: small target detection; YOLOv8; lightweight; LIoU loss; feature enhancement

1. Introduction

With the increasing popularity of drones, they are widely utilized in various fields, such as electricity, agriculture, transportation, and rescue operations. Integrating drones with object detection methods plays a crucial role in automation and development across numerous fields. Indeed, in the power industry, reference [1] employs an improved YOLOX model for transmission line anomaly detection, incorporating an alpha loss function to optimize the localization of small targets. In the field of transportation applications, reference [2] introduces a small target detection layer in YOLOv5 and incorporates the normalized Wasserstein distance (NWD) to reduce the intersection over union (IOU) sensitivity to small target positional deviations. In the livestock industry, employing drones in conjunction with object detection methods allows for the monitoring [3] of livestock to achieve intelligent management and improve efficiency. In the pursuit of model lightweighting, reference [4] reduces computational complexity and compresses model size by introducing the C2F-Ghost module. However, images captured from the perspective of a drone often feature smaller target object pixels and proportions in the image due to factors such as shooting height and angle. Additionally, there may be extensive target overlap in these photographs, which usually feature complicated backdrops. The requirements of UAV target identification are no longer addressed by traditional target detection techniques, which have issues with computing redundancy, complex artificial feature creation, and low robustness. Accurately and quickly recognizing targets from the perspective of drones has become a current focus and challenge in research.

Deep learning-based target detection algorithms can be categorized into two camps: one stage and two stage. The two-stage algorithm generates a series of target candidate regions, which are then subjected to feature extraction by a convolutional neural network for classification and localization. This category of algorithms is exemplified by the R-CNN [5–8] series, which is considered classic in this domain. Bai et al. [9] integrated the faster R-CNN model with generative adversarial nets (GAN) to address the challenge of detecting dim and small targets in automatic target recognition (ATR) systems. The results demonstrate the effectiveness of the proposed method in achieving stable performance and lower false alarm rates in various scenarios compared to traditional methods. A faster R-CNN model with the RegNet network as the backbone was proposed by Wang et al. [10] to reduce the loss of key information due to R-CNN sampling, which achieved a great boost in the detection of spacecraft components. For the problem of slow inference in two-stage networks, He et al. [11] reduced the iterative detection head and simplified the feature pyramid network to accelerate the sparse R-CNN, which not only reduced the amount of computation but also gained accuracy. However, there are still issues in implementing two-stage algorithms on small embedded devices. Conversely, the one-stage algorithms represented by the You Only Look Once (YOLO) [12–16] series, SSD [17], and RetinaNet [18] offer a notable performance gain but at the expense of reduced detection accuracy. Zhu et al. [16] integrated the convolutional block attention model (CBAM) and added detection heads to YOLOv5. However, the computational volume of the model increased dramatically to 237.7 GFLOPs, which is contrary to lightweighting. Yao et al. [19] evaluated several model architectures in the YOLO family and SSDs on a range of devices representing the capabilities of robotic and edge devices. New variants of the YOLO-LITE architecture were proposed, which exceled in lightweighting but was overwhelmed in the face of small targets. Sun et al. [20] improved the small target detection performance of the SSD algorithm by enhancing the detection function with contextual information and introducing a segmentation mask. The added complexity in the model architecture, due to the integration of segmentation and detection branches, make it more challenging to implement and fine-tune compared to simpler models. However, under the unmanned perspective, due to the fact that the small target itself occupies a relatively small area, carries little information, and has a complex image background, missed detection and misdetection occurs from time to time [21]. The accuracy of small target detection is still far less than that of medium and large targets. From the point of view of the balance between model size and detection accuracy, the YOLO series is undoubtedly one of the most popular detection algorithms.

Although YOLOv8 is the most remarkable model in the series, it performs far better at detecting conventional targets than the tiny, much like its predecessors. Therefore, an STD-YOLOv8 model based on improved YOLOv8n is proposed. The model specifically improves YOLOv8's detection layer structure, backbone, and loss functions. Overall, the contributions of this work are as follows:

1) Instead of stride-wise convolution and pooling operations, the SPD-Conv module is used in the backbone network to downsample the feature map without losing useful information.

2) In the neck network, the structure of three detection layers is maintained, and the difference is that the original P5 detection layer is replaced by a shallower P2 detection layer, forming a three-detection-layer structure of P2, P3, and P4. Greater benefits in the new detection layer structure are possible when the content-aware feature reassembly upsampling operator is substituted for nearest-neighbor upsampling simultaneously. The goal is to enable the neck to uncover and retain more small target features and reduce the model depth.

3) After considering the regression problem for tiny target bounding boxes, the penalty term of the original loss function is optimized, and the LIoU loss function is proposed, which obtains some accuracy improvement in the target dataset.

The remaining content of the article is as follows: In Section 2, we describe related work. Section 3 provides a detailed introduction to the improved method. In Section 4, relevant experiments are conducted, and results are analyzed to demonstrate the superior performance of the improved model. Section 5 concludes the entire paper.

2. Materials and methods

Small targets can be defined by both relative proportions and absolute pixels. Chen et al. [22] defined targets with a bounding box-to-image ratio between 0.08 and 0.58% as small targets. In the MS COCO [23] dataset, small targets are defined as objects with a resolution smaller than 32 × 32 pixels. Furthermore, there are other definitions based on absolute scales, e.g., in the aerial image dataset DOTA [24], targets with pixel values ranging between 10 and 50 are defined as small targets. In the pedestrian recognition dataset CityPersons [25], small targets are defined as those with a height of less than 75 pixels. Based on the above definition of small targets, most images captured from a drone's perspective would fall into the category of small targets.

2.1. Overview of the YOLOv8

In 2015, Redmon et al. [12] introduced YOLO (You Only Look Once), transforming the object detection task into a regression problem and ushering in the era of "fast" object detection. Gradually, the YOLO series has become widely recognized and extensively used as a single-stage object detection algorithm. Compared with the two-stage algorithm, YOLO has a substantial improvement in detection speed, but its detection accuracy and generalization ability are relatively poor. After several years of development, YOLO has been upgraded to version 8, which Ultralytics open-sourced on January 10, 2023.

YOLOv8 is categorized based on network width and depth into YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. YOLOv8n is the most miniature model among them, and its small target recognition performance is even better than YOLOv5s. Consequently, selecting YOLOv8n is consistent with the paper's study objectives. The network structure of YOLOv8, as shown in Figure 1, is primarily divided into the following parts: backbone, neck, and head. The backbone consists of Conv Module, C2f, and SPPF. The Conv Module is composed of a 2D convolutional layer, a 2D BatchNorm layer, and a SiLU activation function. Compared to the previous C3, C2f has a richer gradient flow and adapts different channel numbers for models of various scales. Because of its more streamlined structure, a lightweight design is ensured while richer gradient flow information may be obtained. The head section undergoes the most significant changes, as YOLOv8 adopts a decoupled head structure and replaces Anchor-Base with Anchor-Free design. It is noteworthy that the duties of classification and regression are kept apart by this decoupled head. The previous objectness branch is no longer there. Instead, it directly decouples into two separate branches. Moreover, its regression branch utilizes the integral representation. As a result, the model's speed and accuracy are improved.

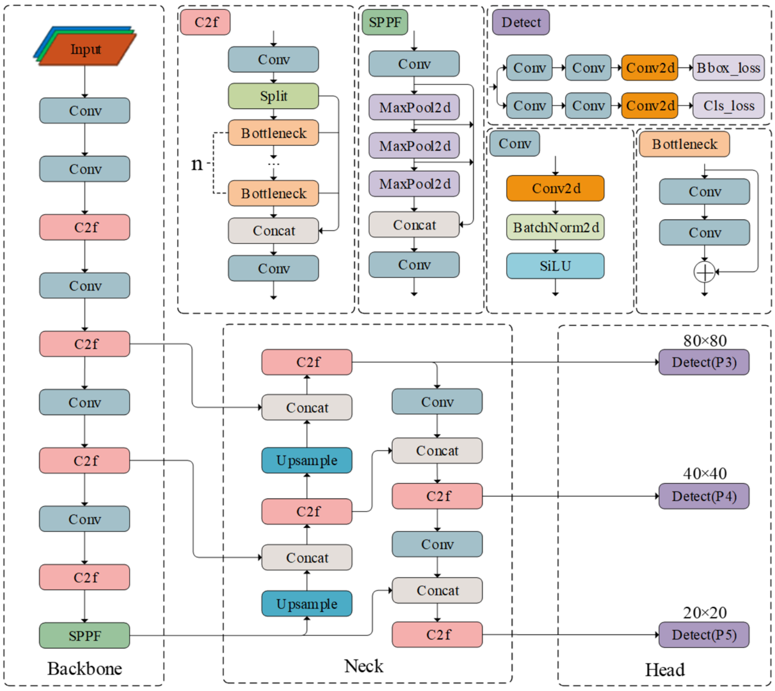

Figure 1. The structure of YOLOv8.

3. Methods

The modified model structure is illustrated in Figure 2. In the backbone network, the SPD module is integrated to replace stride-wise convolutions and pooling layers, thereby preserving feature information during the downsampling. In the head section, the P5 layer is replaced with a P2 detection layer and corresponding redundant modules are removed. This reduces network depth while simultaneously improving the detection rate for small targets. Finally, the original upsampling is replaced with content-aware reassembly of features (CARAFE) [26], enabling the network to obtain a larger receptive field on shallow feature maps to preserve features of small targets. As for the regression problem, LIoU is proposed to dynamically adjust the regression of the bounding box during the training process to improve its accuracy.

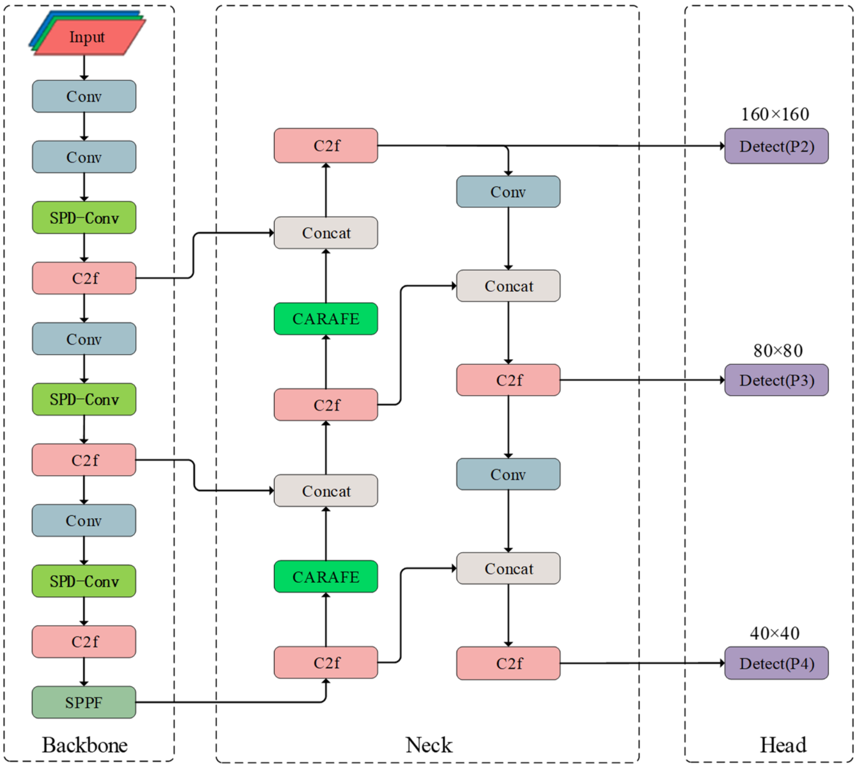

Figure 2. The structure of STD-YOLOv8.

3.1. Structural design of the detection layer

As shown in Figure 3(a), YOLOv8 is designed with three detection layers: P3, P4, and P5, each corresponding to small, medium, and large targets, respectively. When the input image size is 640 × 640, the minimum detectable target pixel size for the P3 layer is 8 × 8, which may not meet the algorithm's requirements in the domain of small targets. In such cases, a common approach is to use a multi-scale detection method, i.e., adding a small target detection layer [16] to enhance the model's performance, as shown in Figure 3(b). The P2 layer can detect small targets with a minimum size of 4 × 4 pixels in a larger receptive field, bringing significant performance gains to the model. However, the quantity of parameters significantly increases, which is not conducive to the model being lightweight. In order to balance the size of the model with its small target performance, in this paper, the P5 layer for large targets is discarded, and the structure shown in Figure 3(c) is adopted. This detection layer structure enables the model to shift its focus from medium and large targets to small ones while reducing the network depth to drastically reduce the model parameters.

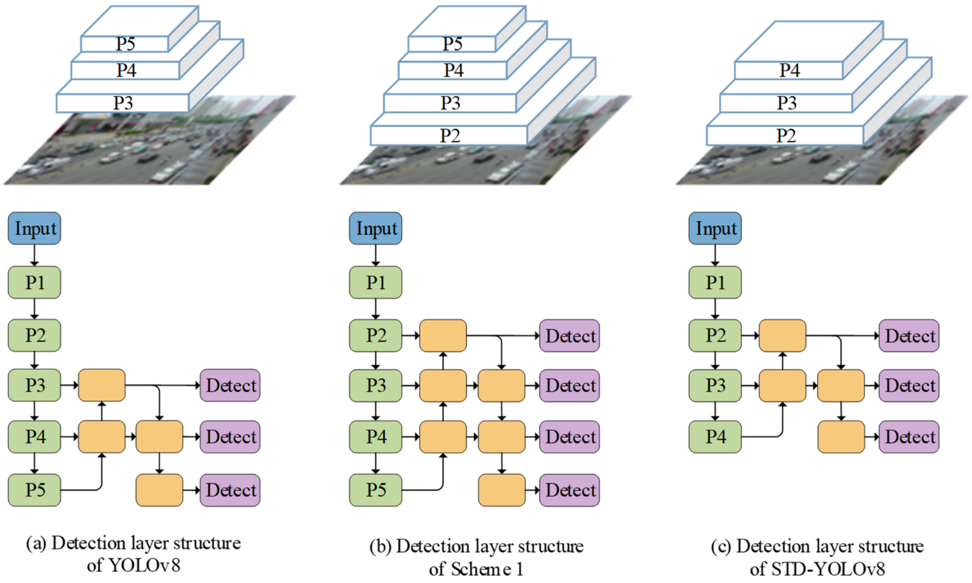
P5
P4
P3
P2
Input
P1
Detect
(a) Detection layer structure of YOLOv8
(b) Detection layer structure of Scheme 1
(c) Detection layer structure of STD-YOLOv8

Figure 3. Comparison of the detection layer structures.

3.2. CNN building block for small objects SPD-Conv

Common CNN structures in typical neural networks often adopt stride-wise convolutions and pooling layers. When confronted with images of good resolution and moderately sized objects, stride-wise convolutions and pooling can conveniently skip redundant pixel information, and the model can still learn features effectively. However, when faced with low-resolution images and small objects, the limitations of this structure begin to emerge. Fine-grained information about the target suffers from loss [27] during the process, making it challenging for the network to capture the features of small objects. Thus, the SPD-Conv module is introduced to replace stride-wise convolutions and pooling layers. The SPD-Conv module consists of a space-to-depth (SPD) layer and a non-stride convolutional layer (stride = 1). The SPD layer is the core component of SPD-Conv. It rearranges the spatial dimensions of the input feature map into the depth dimension without any information loss. When a feature map X with dimensions $S \times S \times C_1$ is obtained, the SPD layer will split it into a series of sub-feature maps according to Eq (1). Figure 4 provides an example illustration when the scale is 2. The four sub-feature maps obtained will be connected along the channel dimension, thus obtaining X' (the symbol of the plus sign in the circle in Figure 4 indicates the process). The spatial dimensions of X' are reduced by a factor of *scale*, while the channel dimensions are increased by a factor of $scale^2$. In other words, the SPD part transforms the feature map X (S, S, C_1) into an intermediate feature map X' ($S/scale$, $S/scale$, $scale^2C_1$). Subsequently, a non-stride convolutional layer with a C_2 filter performs a convolution operation on X' with the stride of 1 (the symbols of stars enclosed in circles in Figure 4 indicates the process), where $C_2 < scale^2C_1$. Generally, using a stride greater than 1 can lead to the neglect of certain pixels, resulting in the indiscriminate loss of feature information. This issue is effectively mitigated by employing non-stride convolution. Finally, the feature map X'' ($S/scale$, $S/scale$, C_2) is obtained. SPD-Conv addresses this issue by eliminating the strided convolution and pooling layers, thereby retaining all spatial information and enhancing feature learning for small objects and low-resolution inputs.

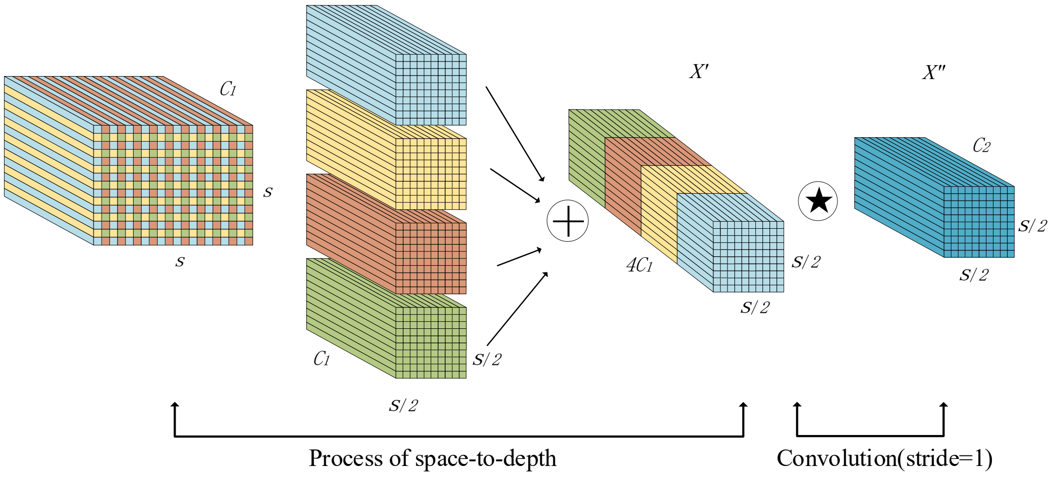

Figure 4. Illustration of SPD-Conv when *scale* = 2.

$$\begin{gathered}
f_{0,0} = X[0:S:scale, 0:S:scale], f_{1,0} = X[1:S:scale, 0:S:scale], \cdots, \\
f_{scale-1,0} = X[scale-1:S:scale, 0:S:scale]; \\
\vdots \\
f_{0,1} = X[0:S:scale, 1:S:scale], f_{1,1} = X[1:S:scale, 1:S:scale], \cdots, \\
f_{scale-1,scale-1} = X[scale-1:S:scale, scale-1:S:scale]
\end{gathered} \tag{1}$$

3.3. The feature reassembly upsampling operator

The nearest-neighbor method is used by YOLOv8 to determine the upsampling kernel, which is based only on the spatial position of pixel points. Its receptive field is typically small, resulting in suboptimal utilization of semantic information in the feature map. This method is prone to generating noise in low-resolution images, leading to feature loss. Accordingly, this paper adopts content-aware reassembly of features (CARAFE) with a larger receptive field. In order to achieve greater performance than mainstream upsampling operators, this study employs the CARAFE upsampling operator, which has a bigger receptive field and can use adaptive and optimized recombination kernels at different places. As shown in Figure 5, the CARAFE operator consists of a kernel prediction module and a content-aware reorganization module. The kernel prediction module is responsible for generating reorganization kernels in a content-aware manner. Each original position on Y corresponds to σ^2 target positions on Y'. Assuming each target position requires a recombination kernel of size $K \times K$ when the input feature map is of size $H \times W \times C$, this module outputs recombination kernels of size $H \times W \times \sigma^2 \times K^2$. For each recombination kernel M_ι, the content-aware module reassembles local region features with a simple weighted sum operation. Under the influence of the recombination kernel, each pixel in region N_ι contributes differently to the upsampled pixel. The process is based on the content of the features rather than positional information. Information from relevant points in the feature region can receive more attention. The visualized feature maps before and after the improvement are shown in Figure 6 (nearest neighbors on the left side of the dashed line and CARAFE on the right side). Obviously, at the same depth, the feature map on the right side has a larger receptive field and more pronounced features after reassembly. Additionally, the "gain" on this feature increases with the number of upsampling.

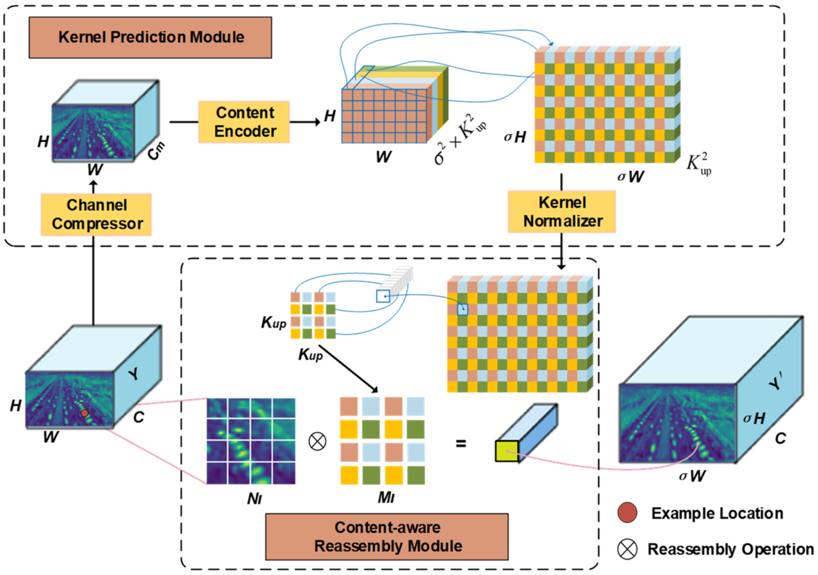

Figure 5. Structure of the CARAFE.

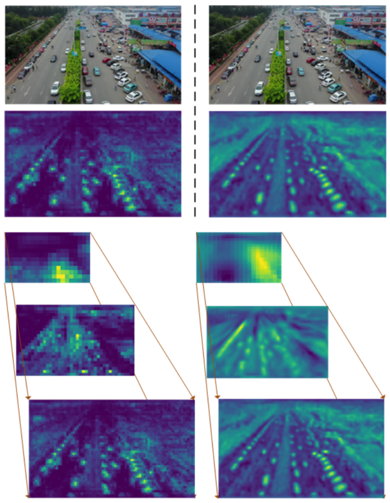

Figure 6. Feature maps before and after improvement.

3.4. The optimized loss function

The sea of deep learning detection algorithms have differences in their frameworks. However, the regression of the bounding box is crucial for all of them, which involves prediction as well as localization of the target box. In regression problems, intersection over union (IoU) is a widely used metric which is defined as:

$$L_{IoU} = 1 - IoU = 1 - \frac{\left|B \cap B_{gt}\right|}{\left|B \cup B_{gt}\right|} \tag{2}$$

where B is the predicted box, and B_{gt} is the ground truth box. Unfortunately, when there is no overlap between the predicted box and the ground truth box, the IoU loss loses its gradient. The generalized intersection over union [28] (GIoU) loss was introduced to address this issue, incorporating a penalty term based on the minimum enclosing rectangle. The GIoU loss is defined as follows:

$$L_{GIoU} = 1 - IoU + \frac{\left|C - B \cup B_{gt}\right|}{\left|C\right|} \tag{3}$$

where C is the minimum enclosing rectangle of the two boxes, which ensures gradient updates even in cases of no overlap. Nevertheless, when there is a containment relationship between the two boxes, GIoU degenerates into IoU. In this case, DIoU and CIoU are proposed simultaneously in [29]. DIoU defines the penalty term as the normalized length of the line joining the centroids of the two bounding boxes, which is given by:

$$L_{DIoU} = 1 - IoU + \frac{\rho^2(b, b_{gt})}{c^2} \tag{4}$$

where R is the penalty term, and b and b_{gt} are the centers of the predicted and ground truth boxes, respectively. $\rho^2(b, b_{gt})$ is the Euclidean distance between the two, and c is the length of the diagonal of the minimum outer rectangle, as shown in Figure 7.

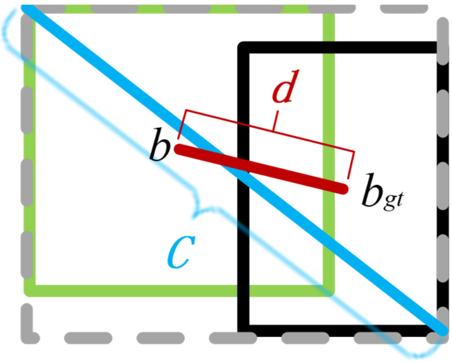

Figure 7. Principle of DIoU regression.

CIoU further takes into account the aspect ratio of the bounding boxes based on the former. The specific formula is as follows:

$$R_{CIoU} = \frac{\rho^2(b, b_{gt})}{c^2} + \alpha v, \alpha = \frac{v}{L_{IoU} + v}, \quad v = \frac{4}{\pi}(\arctan\frac{w_{gt}}{h_{gt}} - \arctan\frac{w}{h})^2 \tag{5}$$

where α is the trade-off parameter and v describes the aspect ratio consistency. Considering that $w^2 + h^2$ is relatively small within the range (0,1), which may lead to gradient explosion, by simply replacing ($1 / w^2 + h^2$) with 1, the gradient direction will remain the same as in Eq (6). When the detected objects are small targets, the smaller size of the bounding boxes makes them more sensitive to changes in distance. To address this, a dynamic adjustment factor, denoted as "*l*", is designed to optimize the penalty term. The new penalty term, R_{LIoU}, is defined as in Eq (7).

$$\frac{\partial v}{\partial w}=\frac{8}{\pi^2}(\arctan\frac{w_{gt}}{h_{gt}}-\arctan\frac{w}{h})\times\frac{h}{w^2+h^2},\ \frac{\partial v}{\partial h}=-\frac{8}{\pi^2}(\arctan\frac{w_{gt}}{h_{gt}}-\arctan\frac{w}{h})\times\frac{w}{w^2+h^2} \quad (6)$$

$$R_{LIoU}=\frac{\rho^2(b,b_{gt})}{c^2}+l\times\alpha v, l=(e^{\frac{\rho^2(b,b_{gt})}{c^2}}-1) \quad (7)$$

$$L_{LIoU}=1-IoU+\frac{\rho^2(b,b_{gt})}{c^2}+l\times\alpha v \quad (8)$$

The graph of $y = e^x - 1$ is illustrated in Figure 8, with the range of x being (0,1). During the initial stages of training, when the predicted bounding boxes are relatively distant from the ground truth boxes corresponding to region A in the diagram, the penalty intensity is increased. As training progresses and the overlap between the predicted and ground truth bounding boxes increases, entering region B in the function, this reduces the penalty intensity. The model's overall performance is improved by making such dynamic adjustments in both the pre-training and post-training stages to enhance regression accuracy. The LIoU loss function is defined by Eq (8).

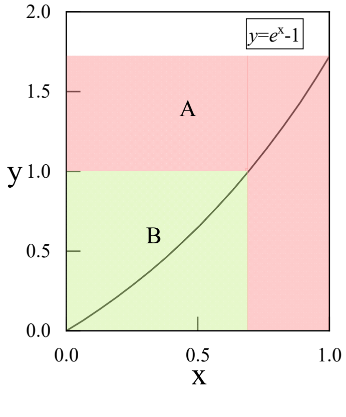

Figure 8. The function $y = e^x - 1$ (when x ϵ(0, 1)).

4. Experiments and analysis

4.1. Dataset

The dataset utilized in the experiments of this study is the VisDrone-2021 [30] dataset, publicly released by the AISKYEYE team from Tianjin University. The dataset is captured by various cameras mounted on drones, totaling 8629 images, which include 6471 for training, 548 for validation, and 3190 for testing. The image annotations include ten categories: awning-tricycle, bicycle, bus, car, people, pedestrian, motor, truck, tricycle, and van. Figure 9 illustrates the distribution of label sizes in the training set. It is evident from the figure that labels are predominantly concentrated in the bottom-left corner, indicating a significant proportion of small targets in the VisDrone-2021 dataset. This poses a considerable challenge for detection.

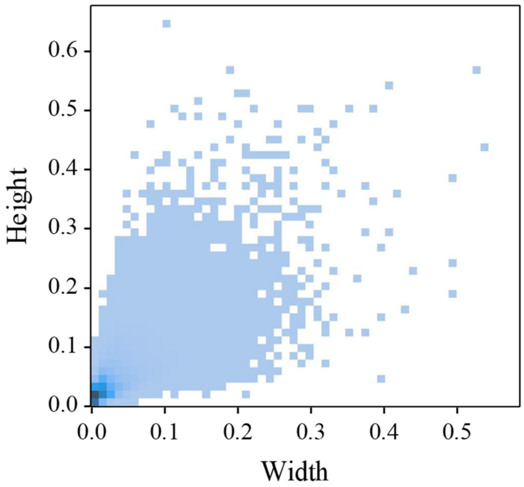

Figure 9. Label size distribution of the training dataset.

4.2. Experimental details and evaluation criteria

Experimental Details: The experiments in this study were conducted using hardware primarily equipped with an NVIDIA RTX 3080 GPU and an i5-12600KF CPU. The software configuration involved the PyTorch 2.0.1 deep learning framework on a Windows 10 operating system. The input image size was 640 × 640, with a total of 300 iterations. The batch size was 16, the initial learning rate was 0.01, the learning rate momentum was 0.937, and the weight decay coefficient was 0.0005. All experiments were trained from scratch and no pre-trained models were utilized.

Evaluation Metrics: To assess the effectiveness of the proposed model in this study, model complexity was evaluated based on the number of parameters and the memory footprint. In terms of performance, precision, recall, mAP_{50}, and mAP were employed as evaluation metrics. The mAP refers to the mean average precision, calculated by taking the mean of all average precisions (AP) computed at IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05, which is the same as mAP_{95}.

4.3. Comparison of detection layer structures

In order to verify the validity of the detection layer structure in this paper, we used Scheme 1 and this paper's scheme for comparison experiments. The two experimental models differ only in the detection layer structure. The experimental results are shown in Table 1. It can be observed that both schemes yield the same mAP_{50}. The proposed scheme in this study outperformed Scheme 1 in precision by 0.9% while lagging by 0.3% in recall. However, our scheme significantly led in terms of model parameters, achieving a reduction of 57.1%. To put it simply, the components of the original model designed for medium and large targets became redundant when dealing with small targets. Consequently, our scheme achieved comparable performance with less than half the number of parameters. The experimental results provide the best evidence.

Table 1. Performance comparison of the two schemes.

Methods	P (%)	R (%)	mAP$_{50}$ (%)	Params (M)	GFLOP	Size (mb)
Scheme 1	50.0	**39.6**	40.7	3.0	16.4	6.70
Our Scheme	**50.9**	39.3	40.7	**1.2**	**14.1**	**2.70**

4.4. Ablation experiment

To assess the effectiveness of the SPD_Conv module, CARAFE upsampling operator, and LIoU loss function, this study designed six ablation experiments on the VisDrone-2021 dataset. The results of the ablation experiments are shown in Table 2. Considering the focus on model lightweight design, except for the baseline, the rest of the models in the table adopt the new detection layer structure (P2 + P3 + P4). Experiments A, B, and C show that the three improvement methods have each contributed to a certain extent in enhancing the model's performance. Among them, the improvement brought by the SPD-Conv module is the most significant, with a mAP$_{50}$ of 39.9%. It is evident that the performance of the model gradually improves with the increase of improvement modules. Ultimately, compared to the baseline, the proposed model demonstrated improvements in precision, recall, and mAP$_{50}$ by 5.1, 5.1 and 6.2%, respectively, with a 58.3% reduction of parameters. Experimental results show that the improved methods in this paper are feasible in terms of reducing model complexity and improving the performance of small targets.

Table 2. Results of the ablation experiments.

Models	LIoU	SPD	CARAFE	P (%)	R (%)	mAP$_{50}$ (%)	mAP$_{95}$ (%)	Params (M)	Size (mb)
Baseline				45.8	34.2	34.5	19.7	3.0	5.96
A	√			47.8	37.8	38.3	22.6	1.1	2.40
B		√		49.6	38.8	39.9	23.5	1.2	2.61
C			√	48.2	38.0	38.3	22.5	1.1	2.48
D	√		√	48.4	38.0	38.6	22.8	1.1	2.48
E		√	√	49.8	39.2	40.1	23.4	1.2	2.70
F	√	√		48.8	40.0	40.2	23.7	1.2	2.61
Ours	√	√	√	50.9	39.3	40.7	24.0	1.2	2.70

4.5. Contrast experiment

Existing classical algorithms are compared on the VisDrone dataset to validate the performance of STD-YOLOv8 for small target detection. Table 3 displays the performance metrics of classical algorithms. Compared to other algorithms in the table, STD-YOLOv8 gets the lead in both complexity and mAP metrics. While the FPS may not be optimal, it still reaches 96, which fully satisfies the real-time requirements. In comparison, specifically with YOLO-UAVlite [31] and [32], which also focus on lightweight design, STD-YOLOv8 achieves superior performance with even fewer parameters. Similarly, compared to the lightweight version of TPH-YOLOv5 [16], which performs well in the VisDrone dataset challenges, STD-YOLOv8 maintains an advantageous position. The boost obtained by our method is even more significant (green part of the table) when reducing the resolution of the input image to 480 × 480. This implies that the model has better adaptability, performing well even when the camera imaging quality is low. In summary, STD-YOLOv8 performs excellently in small-target and low-resolution image detection. Figure 10 shows the relationship between the model's parameter size and its mAP95. STD-YOLOv8 achieved 24.0% mAP$_{95}$ with a parameter of 1.2M, compared to YOLOv8-n. The parameters of STD were reduced by 58.3%, while mAP$_{95}$ improved by 4.3%. Compared to the recent study [32], the parameters decreased by 37.5% and map95 increased by 1.4%. This demonstrates that the improvement strategy proposed in this paper is effective.

Table 3. Results of the contrast experiments.

Models	ImgSize	mAP$_{50}$ (%)	mAP$_{95}$ (%)	Params (M)	Size (mb)	GFLOPs	Speed (ms)	FPS
anchors-base								
YOLOv5s	640 × 640	32.1	17.3	7.0	14.40	15.8	15.6	64
YOLOv5m	640 × 640	36.9	20.7	20.9	42.20	48.3	12.8	78
YOLO-UAVlite [31]	640 × 640	36.6	20.6	1.4	-	-	15.9	63
TPH-YOLOv5 [16]	640 × 640	36.5	19.7	6.5	-	28.0	23.3	43
anchors-free								
YOLOv6	640 × 640	38.1	22.7	16.3	32.80	44.2	8.0	125
YOLOv7-tiny	640 × 640	34.8	18.1	6.0	11.70	13.1	20.0	50
Reference [32]	640 × 640	38.3	22.6	2.0	4.1	13.6	8.0	125
YOLOv8n	640 × 640	34.5	19.7	3.0	5.96	8.1	8.4	119
	480 × 480	27.9	15.5	3.0	5.96		7.8	129
STD-YOLOv8n	640 × 640	40.7 (+6.2)	24.0 (+4.3)	1.2	2.70	13.9	10.4	96
	480 × 480	35.1 (+7.2)	20.3 (+4.8)	1.2	2.70		9.4	106

**Note:* The "-" indicates that the cited study did not provide this metric.

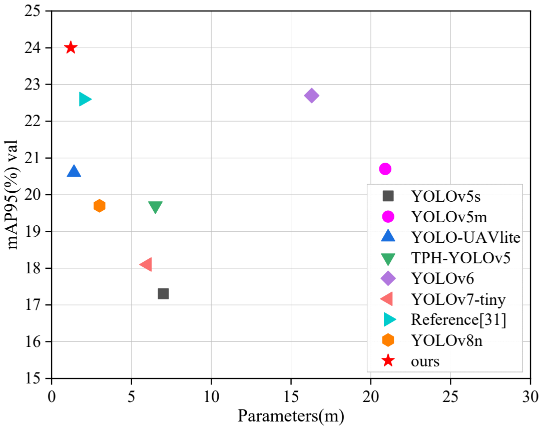

Figure 10. Comparison of parameters and mAP$_{95}$ for different models.

4.6. Comparison of loss functions

This paper additionally designs experiments to compare common loss functions on the VisDrone dataset. The results are presented in Table 4. Compared to the baseline, LIoU outperforms in terms of recall (R), mAP50, and mAP95 by 0.8, 0.9, and 0.7%, respectively. Further, compared to GIoU, EIoU, and WIoU, while not every metric is the highest, LioU's overall performance is the best of the group. Experimental results demonstrate that the proposed improvement to the loss function in this paper is feasible on the small target dataset.

Table 4. Comparison of loss functions.

Datasets	Loss function	P (%)	R (%)	mAP_{50} (%)	mAP_{95} (%)	Speed (ms)	FPS
VisDrone	Baseline (CIoU)	47.2	37.0	37.4	21.9	11.4	88
	GIoU	47.5	36.9	37.6	21.5	11.6	86
	WIoU	48.0	37.6	38.1	21.5	10.6	94
	EIoU	47.6	37.2	37.9	21.6	11.1	90
	LIoU	47.8	37.8	38.3	22.6	11.1	90

4.7. Generalization experiments

By conducting experiments on the TinyPerson and AI-TOD datasets, this study validates the generalization capability of our model. Similar to the VisDrone dataset, the TinyPerson dataset is also based on the perspective of uncrewed aerial vehicles. The dataset has a total of 1610 images and 72,651 manually labeled instances. However, the average size of the targets is only 18 pixels. The AI-TOD dataset, on the other hand, is composed of remote sensing images, with the average size of the targets being only 12.8 pixels. Detection algorithms face greater challenges in the above datasets. The performance of this paper's model on the above datasets is given in Table 5. It is evident that the raises brought by the proposed algorithm are substantial. In the TinyPerson dataset, recall significantly increased by 10.2%, precision by 5.7%, mAP_{50} by 9.8%, and mAP_{95} by 3.4%. The corresponding data in the AI-TOD dataset are 16.7, 0.3, 2.8, and 1.3%, respectively. This indicates that the model proposed in this paper achieves significant improvements on datasets that are much more challenging to detect, demonstrating its excellent robustness.

Table 5. Generalization experiment results of STD-YOLOv8.

Datasets	model	P (%)	R (%)	mAP_{50} (%)	mAP_{95} (%)	Speed (ms)	FPS
TinyPerson	YOLOv8	40.6	21.9	19.8	7.3	9.7	103
	STD-YOLOv8	46.3	32.1	29.7	10.7	10.8	92
AI-TOD	YOLOv8	47.2	38.3	37.2	15.6	11.1	90
	STD-YOLOv8	63.9	38.6	40.0	16.9	10.4	96

4.8. Actual detection results and analysis

In order to verify the model of this paper in real scenarios, some images with high detection challenges in VisDrone-2021 and TinyPerson datasets are selected for testing. The results are presented in Figures 11 and 12, respectively. By comparing the highlighted regions in Figure 11(a),(c) and 11(b),(d), it can be observed that STD-YOLOv8 exhibits a higher detection rate for small objects near the image edges, demonstrating excellent detection performance. The advantage is equally evident in the TinyPerson dataset. Figure 12(c),(d) demonstrate the model's performance under extremely challenging conditions. STD-YOLOv8 detected 342 targets in this image (219 for earth_person and 123 for sea_person), while the baseline model detected only 197 (115 for earth_person and 82 for sea_person). In a nutshell, benefiting from the targeted improvements, STD-YOLOv8 exhibits better detection capabilities for drone images with complex backgrounds and dense distribution, which can effectively suppress the interference of image background noise information and retain small target feature information from it.

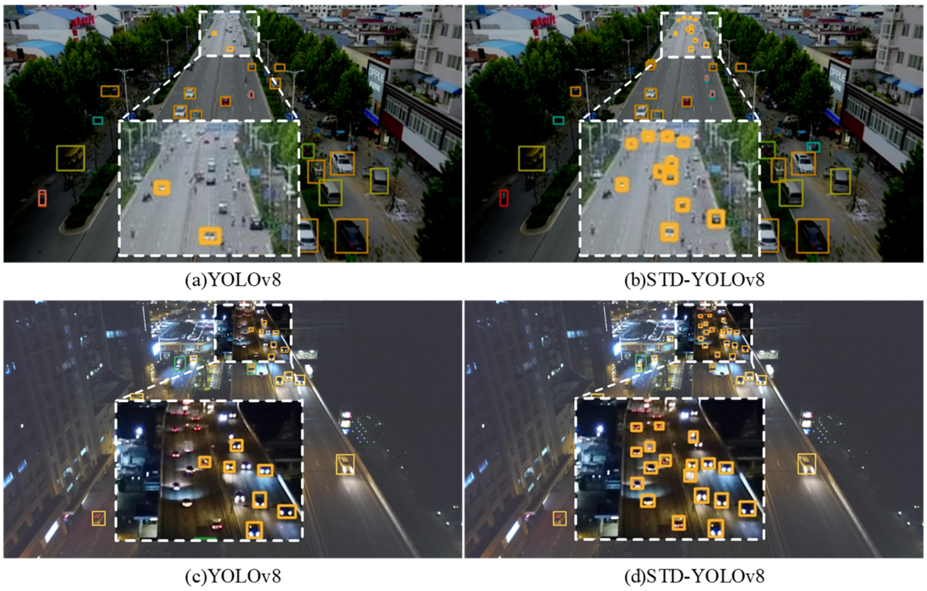
(a)YOLOv8 (b)STD-YOLOv8 (c)YOLOv8 (d)STD-YOLOv8

Figure 11. Detection results on the VisDrone-2021 dataset.

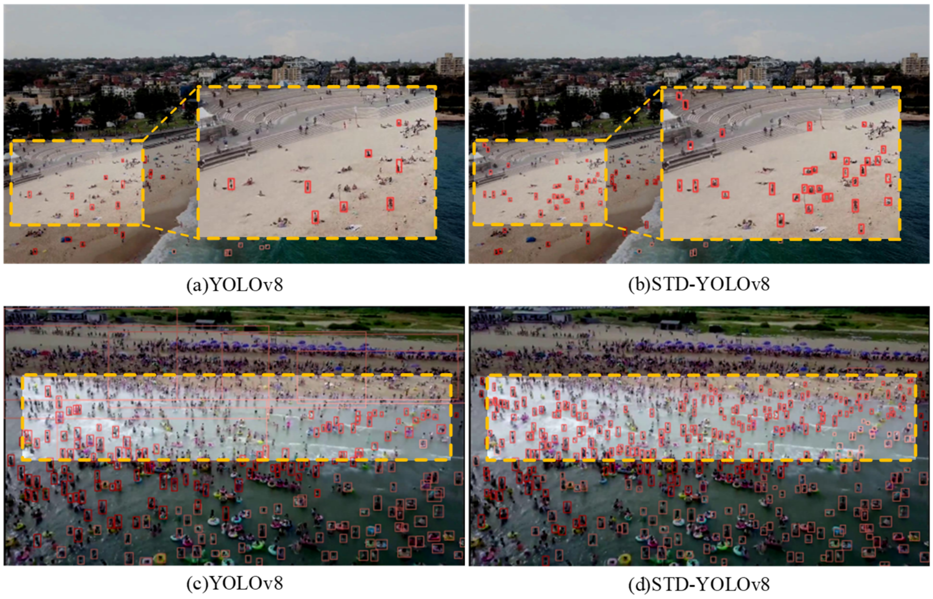
(a)YOLOv8 (b)STD-YOLOv8 (c)YOLOv8 (d)STD-YOLOv8

Figure 12. Detection results on the Tiny-Person dataset.

5. Conclusions

To address challenges arising from the complexity of backgrounds and small target scales in UAV aerial images, a lightweight small object detection algorithm, named STD-YOLOv8, is proposed. This algorithm builds upon improvements based on YOLOv8n. First, a new detection layer combination (P2 + P3 + P4) is employed to explore small object features in the shallow network layers. In the backbone, the SPD-Conv module is incorporated to reduce the feature loss caused by strided convolution in the downsampling process, which improves the feature extraction capability of the backbone network in low-resolution and small-target images. For the neck network, the CARAFE operator is utilized instead of nearest-neighbor upsampling, resulting in more salient feature information over a larger receptive field. The LIoU loss function, on the other hand, adopts a bolder regression strategy, which is designed according to the actual situation of the small target and boosts the model performance to some extent. The results on small target datasets like VisDrone and TinyPerson suggest that STD-YOLOv8 balances accuracy and complexity and has excellent small target detection properties. In our future research, we will aim to further eliminate redundant components of the model and employ techniques such as knowledge distillation to achieve higher detection accuracy from the larger model. This will help in making our model both lighter and more accurate.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was supported by Guangxi Natural Science Foundation(2022GXNSFBA035613).

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. Z. Bi, L. Jing, C. Sun, M. Shan, YOLOX++ for transmission line abnormal target detection, *IEEE Access*, **11** (2023), 38157–38167. https://doi.org/10.1109/ACCESS.2023.3268106
2. R. Li, Y. Chen, C. Sun, W. Fu, Improved algorithm for small target detection of traffic signs on YOLOv5s, in *2023 4th International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, (2023), 339–344. https://doi.org/10.1109/ICHCI58871.2023.10278065
3. F. A. Kurniadi, C. Setianingsih, R. E. Syaputra, Innovation in livestock surveillance: Applying the YOLO algorithm to UAV imagery and videography, in *2023 IEEE 9th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, (2023), 246–251. https://doi.org/10.1109/ICSIMA59853.2023.10373473
4. H. Y. Jiang, F. Hu, X. Q. Fu, C. R. Chen, C. Wang, L. X. Tian, et al., YOLOv8-Peas: a lightweight drought tolerance method for peas based on seed germination vigor, *Front. Plant Sci.*, **14** (2023). https://doi.org/10.3389/fpls.2023.1257947

5. R. Girshick, Fast R-CNN, in *2015 IEEE International Conference on Computer Vision (ICCV)*, (2015), 1440–1448. https://doi.org/10.1109/ICCV.2015.169
6. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, (2014), 580–587. https://doi.org/10.1109/CVPR.2014.81
7. K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in *2017 IEEE International Conference on Computer Vision (ICCV)*, (2017), 2980–2988. https://doi.org/10.1109/ICCV.2017.322
8. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39** (2017), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
9. J. Bai, H. Zhang, Z. Li, The generalized detection method for the dim small targets by faster R-CNN integrated with GAN, in *2018 IEEE 3rd International Conference on Communication and Information Systems (ICCIS)*, (2018), 1–5. https://doi.org/10.1109/ICOMIS.2018.8644960
10. Z. Wang, Y. Cao, J. Li, A detection algorithm based on improved faster R-CNN for spacecraft components, in *2023 IEEE International Conference on Image Processing and Computer Applications (ICIPCA)*, (2023), 1–5. https://doi.org/10.1109/ICIPCA59209.2023.10257992
11. Z. H. He, X. Ye, Y. Li, Compact sparse R-CNN: Speeding up sparse R-CNN by reducing iterative detection heads and simplifying feature pyramid network, *AIP Adv.*, **13** (2023). https://doi.org/10.1063/5.0146453
12. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only look once: Unified, real-time object detection, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 779–788. https://doi.org/10.1109/CVPR.2016.91
13. J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 6517–6525. https://doi.org/10.1109/CVPR.2017.690
14. J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, preprint, arXiv:1804.02767. https://doi.org/10.48550/arXiv.1804.02767
15. A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, YOLOv4: Optimal speed and accuracy of object detection, preprint, arXiv:2004.10934. https://doi.org/10.48550/arXiv.2004.10934
16. X. Zhu, S. Lyu, X. Wang, Q. Zhao, TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios, in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, (2021), 2778–2788. https://doi.org/10.1109/ICCVW54120.2021.00312
17. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Y. Fu, et al., SSD: Single shot multibox detector, in *Computer Vision–ECCV 2016. ECCV 2016. Lecture Notes in Computer Science()*, Springer, **9905** (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
18. T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, **42** (2020), 318–327. https://doi.org/10.1109/TPAMI.2018.2858826
19. Z. Yao, W. Douglas, S. O'Keeffe, R. Villing, Faster YOLO-LITE: Faster object detection on robot and edge devices, in *RoboCup 2021: Robot World Cup XXIV*, Springer, **13132** (2021), 226–237. https://doi.org/10.1007/978-3-030-98682-7_19
20. C. Sun, Y. B. Ai, S. Wang, W. D. Zhang, Mask-guided SSD for small-object detection, *Appl. Intell.*, **51** (2021), 3311–3322. https://doi.org/10.1007/s10489-020-01949-0

21. H. Wang, H. Qian, S. Feng, GAN-STD: small target detection based on generative adversarial network, *J. Real-Time Image Process.*, **21** (2024), 65. https://doi.org/10.1007/s11554-024-01446-4
22. C. Chen, M. Y. Liu, O. Tuzel, J. Xiao, R-CNN for small object detection, in *Computer Vision—ACCV 2016*, Springer, **10115** (2017), 214–230. https://doi.org/10.1007/978-3-319-54193-8_14
23. X. Chen, H. Fang, T. Y. Lin, R. Vedantam, S. Gupta, P. Dollár, et al., Microsoft COCO captions: Data collection and evaluation server, preprint, arXiv:1504.00325. https://doi.org/10.48550/arXiv.1504.00325
24. G. S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, et al., DOTA: A large-scale dataset for object detection in aerial images, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 3974–3983. https://doi.org/10.1109/CVPR.2018.00418
25. S. Zhang, R. Benenson, B. Schiele, CityPersons: A diverse dataset for pedestrian detection, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 4457–4465. https://doi.org/10.1109/CVPR.2017.474
26. J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, D. Lin, CARAFE: Content-aware reassembly of features, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2019), 3007–3016. https://doi.org/10.1109/ICCV.2019.00310
27. R. Sunkara, T. Luo, No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects, in *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2022*, Cham: Springer Nature Switzerland, **13715** (2022), 443–459. https://doi.org/10.1007/978-3-031-26409-2_27
28. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 658–666. https://doi.org/10.1109/CVPR.2019.00075
29. Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU loss: Faster and better learning for bounding box regression, in *AAAI Conference on Artificial Intelligence*, (2019). https://doi.org/10.48550/arXiv.1911.08287
30. Y. Cao, Z. He, L. Wang, W. Wang, Y. Yuan, D. Zhang, et al., VisDrone-DET2021: The vision meets drone object detection challenge results, in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, (2021), 2847–2854. https://doi.org/10.1109/ICCVW54120.2021.00319
31. C. Liu, D. G. Yang, L. Tang, X. Zhou, Y. Deng, A lightweight object detector based on spatial-coordinate self-attention for UAV aerial images, *Remote Sens.*, **15** (2022), 83. https://doi.org/10.3390/rs15010083
32. H. J. Nie, H. L. Pang, M. Y. Ma, R. K. Zheng, A lightweight remote sensing small target image detection algorithm based on improved YOLOv8, *Sensors*, **24** (2024), 2952. https://doi.org/10.3390/s24092952