The Journal of Supercomputing
https://doi.org/10.1007/s11227-024-06121-w

SPD-YOLOv8: an small-size object detection model of UAV imagery in complex scene

Rui Zhong[1] · Ende Peng[1] · Ziqiang Li[1] · Qing Ai[1] · Tao Han[1] · Yong Tang[1]

Accepted: 1 April 2024

Abstract

Traditional camera sensors rely on human observation. However, in complex scenes, people often experience fatigue when observing objects of various sizes. Moreover, human cognitive abilities have inherent limitations, leading to potential judgment errors. To overcome these challenges, object recognition technology, a pivotal means of categorizing objects captured by camera sensors is introduced. This paper presents a specialized small-size object detection algorithm designed for unique scenarios. This algorithm offers distinct advantages, including enhanced accuracy in detecting small-size objects and improved detection performance for objects of various sizes. The main innovations in this paper include the following three points. Firstly, summarizing a small-size object detection layer and reconfiguring both the feature extraction network and the feature fusion network to enhance the effectiveness of capturing small-size objects. Secondly, SPD-Conv modules are introduced to replace stride convolutions and pooling layers to improve the detection accuracy of small objects. Finally, employing the MPDIoU loss function to enhance the precision of bounding box fitting. In our experiments. We utilized authoritative official datasets. The experimental results on the Visdrone dataset demonstrate a 10.9% increase in mAP0.5 and a 9.3% increase in mAP0.5:0.95 rates compared to the original YOLOv8s. This model not only meets the accuracy requirements but also accounts for the lightweight demands when deployed on embedded devices.

Keywords Small-size objects detection · YOLOv8 · UAV · SPD-Conv

✉ Qing Ai
aiqing@hbnu.edu.cn

[1] School of Electrical Engineering and Automation, Hubei Normal University, Huangshi City 435002, Hubei, China

Published online: 20 April 2024

1 Introduction

The continuous advancement of both the economy and technology has elevated camera sensors to indispensable tools across various domains, including transportation, surveillance, healthcare, unmanned aerial vehicles, and so on [1–5]. Consequently, object recognition algorithms have emerged as one of the central components in camera sensor technology. Deep learning-based methods for object detection can be broadly classified into two main approaches: two-stage object detection algorithms and one-stage object detection algorithms.

1.1 Two-stage object detection algorithms

These algorithms, represented by FastRCNN [6], FasterRCNN [7], Mask R-CNN [8], and others, follow a two-stage detection process. Initially, they employ algorithms to generate candidate bounding boxes, extracting object regions. Subsequently, convolutional neural networks classify and recognize objects within these candidate boxes. While this approach delivers high accuracy, it tends to be relatively slow, making it unsuitable for real-time monitoring applications.

1.2 One-stage object detection algorithms

Represented by the YOLO series [9–12] and SSD [13] algorithms. They can rapidly detect object categories and determine their positions with associated probabilities.

In the field of small target detection, the YOLO series of algorithms are widely popular due to their rapid and accurate detection capabilities [14, 15]. Wang et al. [16] proposed an improved algorithm that embeds Small Target Detection Structure into the network to enhance the semantic information collection for small targets. They used a Global Attention Mechanism (GAM) to strengthen the global information features of the YOLOv8m backbone network, reducing the loss of image feature information during the sampling process. This significantly improved detection accuracy. The literature [17] used DsPAN as a replacement for PAN in small-size object detection for multi-scale defects and achieved a noticeable improvement in detection speed. Although the aforementioned algorithms have achieved some effectiveness in small-size object detection tasks, there is still room for improvement in balancing detection accuracy and speed. The literature [18] combined an improved version of the YOLO model and OSNet to capture real-time images of road damage and their precise locations. On a dataset containing 9749 road damage images, the OSNet model achieved a 99.4% average precision. It was successfully integrated into road damage detection vehicles, significantly improving road maintenance inspection efficiency. The FL-YOLOv7 algorithm optimized the detection capability of small targets in forest fire scenarios by designing lightweight modules (C3GhostV2), introducing the SimAm attention mechanism, incorporating ASFF modules, and utilizing the WIoU loss function [19]. TinyDet, as a lightweight general object detection framework, achieved improved detection results for small-size objects by using

high-resolution feature maps for dense anchoring, introducing SCConv to reduce computation, enhancing early-stage backbone network, and addressing feature misalignment issues [20].

Algorithms based on SPD-Conv(Space-to-depth convolution) techniques have rapidly developed in various domains of small target detection [21–25]. The literature [21] proposed an improved YOLOv8s algorithm, enhancing the detection accuracy and speed of tiny objects in remote sensing images by introducing the SPD-Conv module and the SPANet path aggregation network. The literature [22] introduced an improved multi-scale YOLOv5 algorithm, significantly enhancing the accuracy of traffic object detection in complex road scenes. This improvement was achieved by incorporating the CARAFE module, SPD-Conv module, a computationally efficient optimized structure, and an attention mechanism. The literature [23] proposed a YOLO-SG algorithm, based on YOLOv5, using SPD-Conv as the down-sampling structure, combining the GhostNet feature extraction architecture, optimizing the output feature structure, and focusing on detecting small traffic signs. Experiments on the GTSDB and TT100K datasets showed that YOLO-SG performed well in detection performance, improving mAP by 2.3% and 6.3%, respectively, while reducing model parameters and demonstrating superior performance in detecting small targets in complex scenes. In the field of UAV, Zhang et al. [24] proposed a new algorithm specifically for small-size target detection in UAV images. This algorithm based on the improved YOLOv5 model, enhanced detection performance for small objects in UAV images by introducing a SPD module and attention mechanism. Experiments on the VisDrone-DET2019 dataset demonstrated a 7.8% improvement over the baseline network.

The research achievements mentioned not only propel the advancement of small target detection technology but also offer innovative solutions for practical applications, particularly in the realm of UAV image processing. Building upon the existing algorithms, we have introduced the SPD-YOLOv8 algorithm, which further refines the mechanisms of feature extraction and fusion. This enhancement allows for improved detection performance, especially when it comes to identifying tiny targets set against complex backgrounds. SPD-YOLOv8 achieves this by preserving finer details of features and effectively suppressing background noise. Moreover, SPD-YOLOv8 enhances the model's real-time performance and accuracy, marking a significant step forward in the field. The primary contributions of this algorithm are as follows:

1. Enhancement of small-size object detection by introducing a dedicated small-size object detection layer, thereby increasing the detection rate for small-size objects.
2. Replacing the SPD-Conv convolution layer instead of stride convolutions and pooling layers helps resolve issues related to the loss of fine-grained information and the ineffective learning of feature representations.
3. Integration of the MPDIOU [26] loss function into the model, ensuring that the model accounts for all relevant factors, including overlapping regions, non-overlapping regions, center-point distances, and width-height deviations. Simultaneously, this simplifies the computation process.

The structure of this paper is divided as follows: In Sect. 2, we introduce YOLOv8, along with the reasons for selecting it as the baseline and the significance of this choice. Section 3 primarily focuses on detailing the improvements introduced in this paper, outlining the enhanced methods and techniques employed. Section 4 places emphasis on presenting the experimental results and comparative experiments. Section 5 concludes the paper, and provides further research and improvement direction of this subject.

2 Related work

Currently, the most common method for acquiring small-size object data involves the use of ground-based sensors such as fixed cameras, piezoelectric sensors, induction loops, and more. For example, Lou [1] proposed a detection method based on a novel down-sampling method. Furthermore, Zou et al. [2] introduced a novel method for day and night obstacle detection based on a camera module. In practical applications, achieving the highest level of real-time detection accuracy necessitates the selection of the currently most popular one-stage algorithm, namely the YOLO algorithm family.

2.1 The reason for choosing YOLOv8 as the baseline

In this section, we introduce the most popular algorithms in recent years, and the article provides a detailed description of some of the primary enhancements made to YOLOv8 in this paper.

YOLO currently stands as one of the most popular real-time object detectors, primarily due to the following reasons: (a) Its lightweight network architecture. (b) Its efficient feature fusion techniques. (c) The ability to deliver more accurate detection results.

The most widely employed versions within the YOLO series remain YOLOv5 and YOLOv7 [27]. YOLOv5 is renowned for its accelerated training pace and enhanced accuracy in contrast to YOLOv4. YOLOv5 incorporates the Mosaic data augmentation method, which not only enriches the dataset but also notably boosts the network training speed and reduces the model's memory demands. Nonetheless, YOLOv5 still exhibits certain limitations in the domain of small-size object detection.

On the other hand, YOLOv7 attains higher accuracy and faster detection speed compared to YOLOv5 while maintaining an equivalent model size. It has been optimized during training and leverages TBoF (Trainable Bag of Freebies) to enhance detector accuracy. However, it faces constraints in terms of detection performance, network structure, feature extraction, and training efficiency, which can lead to noticeable performance degradation in certain scenarios.

Released in 2023, YOLOv8 aspires to amalgamate the finest attributes of real-time object detection. It still embraced the concept of CSP in YOLOv5 [28] and its enhancements encompass: (a) The C3 module has been replaced with the C2f

[27] module, resulting in further lightweighting. This change enhances the model's efficiency and reduces its resource requirements. (b) The head section has been transformed into the prevailing decoupled head structure. This separation of classification and detection heads enhances the model's flexibility. Additionally, it has transitioned from an Anchor-Based [29] to an Anchor-Free approach, offering advantages in terms of object localization. (c) During training, data augmentation techniques from YOLOX have been adopted. In the final 10 epochs of training, the Mosaic augmentation operation [12] is disabled, effectively enhancing the model's accuracy. (d) A regression loss formulation that combines DFL and CIOU [30] has been implemented, contributing to improved performance in object detection. DFL optimizes the probabilities of the two positions closest to the label y through cross-entropy. This enables the network to promptly focus on the distribution near the target's immediate vicinity, as demonstrated in Eq. (1). In essence, the output distribution theoretically approaches the true floating-point coordinates and acquires weights for distances from integer coordinates using linear interpolation.

$$\mathrm{DFL}_{(s_i,s_{i+1})} = -((y_{i+1} - y)\log(s_i) + (y - y_i)\log(s_{i+1})) \tag{1}$$

where S_i is the output of the network's Sigmoid function, and y_i is uniformly sampled from the possible interval $[y_0, y_n]$ of y.

YOLOv8 uses Anchor Base instead of Anchor Free. It designs an Anchor alignment metric to evaluate classification scores and IoU. Equation (2) is employed to compute the alignment degree of Anchor-Level for each instance.

$$t = s^a \times u^b \tag{2}$$

where s represents the classification score, u signifies the IoU value, and a and b denote weight hyperparameters. From the formula above, it becomes evident that t concurrently governs the optimization of both the classification score and IoU to attain Task-Alignment, guiding the network to dynamically emphasize high-quality Anchors. With these enhancements, YOLOv8 attains a 1% higher accuracy than YOLOv5, marking it as the most accurate detector to date.

One of YOLOv8's distinguishing features is its comprehensive framework design, capable of supporting all prior versions of YOLO. This characteristic simplifies the process of switching between different versions and facilitates performance comparisons. Therefore, YOLOv8 is chosen as the baseline.

2.2 Network structure of YOLOv8

The YOLOv8 network comprises three key components: a Backbone network, a Neck network, and a Head. The network's architecture is visualized in Fig. 1.

The backbone network of YOLOv8 draws inspiration from the ELAN design concept of YOLOv7. It replaces the C3 module from YOLOv5 with the C2f module and replaces the channels for different scale models. This adaptation allows it to maintain a lightweight profile while enriching gradient flow information. Towards the end of the backbone, it continues to use the widely-used SPPF module, passing three consecutive Maxpools of size 5×5, followed by the concatenation of each

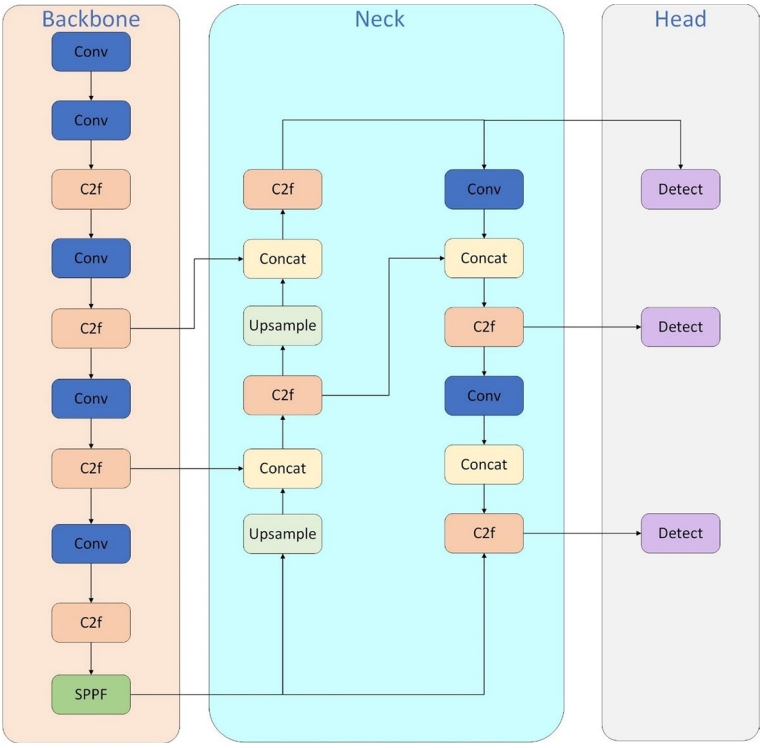

Fig. 1 YOLOv8 network structure

layer. This design ensures precision in object detection across various scales while preserving the network's lightweight.

In the Neck network, the chosen feature fusion method remains PAN-FPN [31, 32], where multiple C2f modules are used to merge feature maps of different scales from the three stages of the Backbone. This integration facilitates the amalgamation of shallow information into the deeper layers.

In the Head section, a decoupled head structure is employed, segregating the classification and localization prediction components. This separation mitigates conflicts between the classification and localization tasks.

3 Network structure enhancement

The YOLOv8 algorithm excels in various aspects; however, it still faces challenges in detecting small-size objects. These challenges can be attributed to the following reasons: (a) Compared with regular-size objects, small-size objects have difficulty having their characteristic features effectively learned in deeper feature maps. (b) Small-size objects are more prone to overlap with other objects, making it challenging to distinguish and precisely locate them within an image.

In order to address these issues, we proposed a detection algorithm that enhances the detection performance of small-size objects while the basis of ensuring the detection accuracy of normal-size objects. First, we introduced a small-object detection layer at the head of the network. This method increased the depth of the entire network structure and improved the detection rate of small-size objects without significantly increasing computational costs. Secondly, improvements have been made to the convolution layers by utilizing the enhanced SPD-Conv convolution layer. This enhancement results in a more comprehensive extraction of information during the network's feature extraction process, addressing the challenge of learning feature information in deeper feature maps. Finally, an MPDIOU loss function minimizes the distances between the predicted bounding boxes' top-left and bottom-right corners and the actual annotated bounding boxes. This approach enabled us to capture more contextual information and effectively address the issue of low detection accuracy caused by object overlap.

3.1 Small size object detection layer

The small-size object detection layer introduced in this paper corresponds to a detection feature map size of 160×160, which is utilized for the detection of objects with sizes of 4×4 or larger.

Typically, object detection layers entail the use of three detection feature maps: 80×80 for detecting objects larger than 8×8, 40×40 for objects larger than 16×16, and 20×20 for objects larger than 32×32. However, small-size objects can sometimes be smaller than 8×8, and their feature information is susceptible to being obscured or even overwhelmed by larger objects. Consequently, using these three detection feature maps might inadvertently filter out certain target information, so the detection effect is poor.

Google introduced MobileNet [33–37], which has less calculation, requiring only one-third of the calculation compared to regular convolutions during the training. However, as the calculational load decreases, several valuable details are overlooked, resulting in a reduction in detection accuracy. Consequently, in most research endeavors, regular convolution down-sampling operations are utilized. Nevertheless, these down-sampling operations across the network contribute significantly to computational demands, which cannot be overlooked.

However, in this paper, while incorporating a small-size object detection layer, we have retained the original detection layer designed for normal-size objects. Although this addition may slightly increase calculational requirements, it could effectively compensate for the information lost during down-sampling in each stage, ensuring the preservation of contextual information. Multiple experiments have demonstrated the superior effectiveness of incorporating the small-size object detection layer compared to the original approach. The specific structure of this layer is illustrated in Fig. 2.

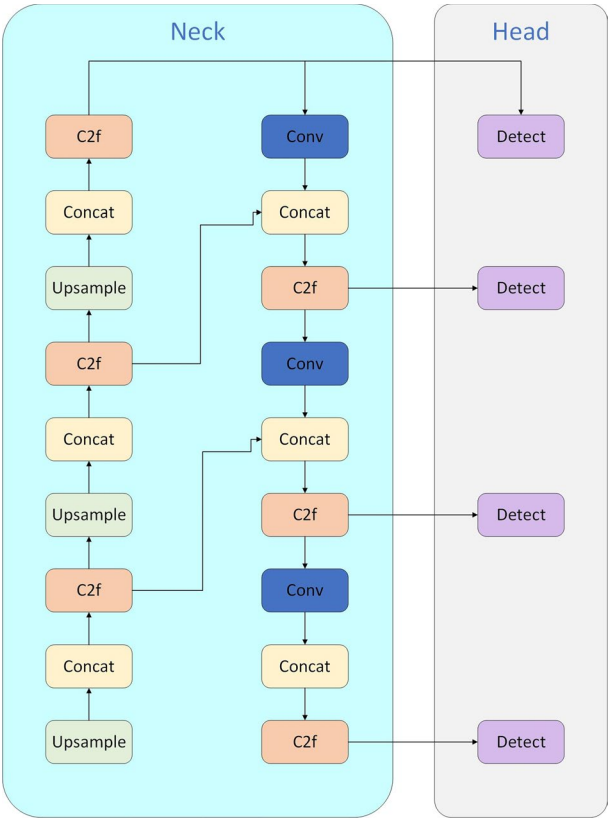

Fig. 2 Diagram of the Object Detection Layer Structure

3.2 SPD-Conv

In the object detection process of YOLOv8, the detection accuracy for small objects is often lower than that for normal-size objects. Images used for detection typically contain a smaller proportion of pixels dedicated to small-size objects, which means there is limited information available for the model to learn from. Furthermore, small-size objects are frequently found alongside other large ones, making their detection more challenging. Which makes the reliability of convolutional neural networks decrease.

In the early layers of the convolutional neural network structure, Step length volume is well-suited for a large amount of redundant information, leading to better feature learning. However, when the target sizes are small, there is less redundant data. In these cases, strided convolutions and pooling can result in the loss of fine-grained information, which is the key factor for the low efficiency of small-size object detection.

To address this issue, this paper introduces a new module known as SPD-Conv to replace stride convolutions step size and pooling layers in this convolutional network. The SPD module comprises two layers: a spatial-to-depth layer and a volume layer. This module down-samples image features to retain necessary information in the channels and subsequently conducts non-stride convolution to simplify the number of channels used in the added convolutional layers. The structure of the SPD module is depicted in Fig. 3.

We assume a feature map X with dimensions $S \times S \times C_1$. It is divided into a series of sub-feature sequences at each stride as follows:

$$\begin{aligned}
&f_{0,0} = X[0:S:\text{scale}, 0:S:\text{scale}], f_{1,0} = X[1:S:\text{scale}, 0:S:\text{scale}], ..., \\
&f_{\text{scale}-1,0} = X[\text{scale}-1:S:\text{scale}, 0:S:\text{scale}]; \\
&f_{0,1} = X[0:S:\text{scale}, 1:S:\text{scale}], f_{1,1}, ..., f_{\text{scale}-1,1} = X[\text{scale}-1:S:\text{scale}, 1:S:\text{scale}]; \\
&... \\
&f_{0,\text{scale}-1} = X[0:S:\text{scale}, \text{scale}-1:S:\text{scale}], f_{1,\text{scale}-1}, ..., \\
&f_{\text{scale}-1,\text{scale}-1} = X[\text{scale}-1:S:\text{scale}, \text{scale}-1:S:\text{scale}];
\end{aligned} \tag{3}$$

The SPD module in this context aims to maintain the discriminant feature information to the maximum possibility by using non-stride convolutions. When using filters with an odd stride, like a step size of 3, the feature map is reduced in size, with each pixel sampled only once. On the other hand, if the filter stride is even, such as a step size of 2, it results in unbalanced sampling, with inconsistent sampling of even and odd rows (columns).

Within the SPD module, the feature map X is proportionally divided into $i+x$ and $j+y$, which donate as $f_{x,y}$, with each sub-map being proportionally down-sampled from X. The scale is 2, and the size of each sub-map is $s/2 \times s/2 \times C_1$. These four sub-maps are then concatenated along the channel dimension to create the feature map X‘, which has a size of $s/2 \times s/2 \times 2^2 C_1$. In the Convolution part of the SPD-Conv module, a convolution layer with a stride of 1 is utilized to resize the feature map X‘ to $s/2 \times s/2 \times C_2$, where $C2 < 2^2 C_1$, with the intention of preserving crucial information as much as possible.

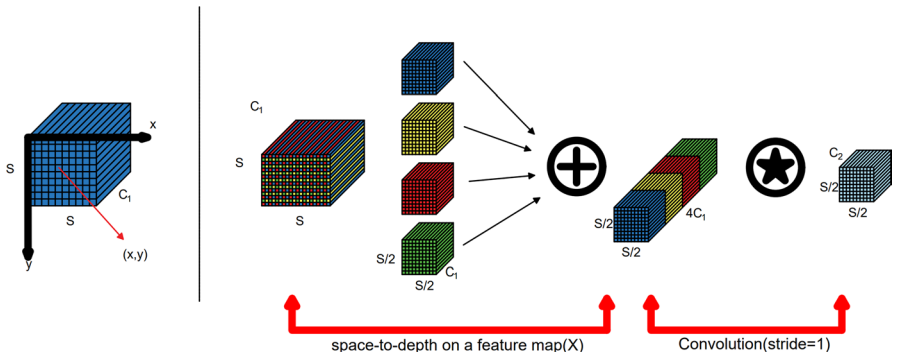

Fig. 3 SPD-Conv down-sampling process

3.3 MPDIOU loss function

The bounding box loss function is a crucial part of the YOLOv8 algorithm, and the efficacy of its detection performance is profoundly shaped by the design of the loss function. The original bounding box loss function of the YOLOv8 algorithm employs the CIOU loss, its loss function is shown in Eqs. (4)–(6).

$$\mathrm{CIOU}_{\mathrm{loss}} = 1 - \mathrm{IOU} + \frac{\lambda^2\left(a, a^{\mathrm{gt}}\right)}{c^2} + \alpha\mu \tag{4}$$

$$\alpha = \frac{\mu}{(1 - \mathrm{IOU}) + \mu} \tag{5}$$

$$\mu = \frac{4}{\pi}\left[\left(\arctan\frac{w^{\mathrm{gt}}}{h^{\mathrm{gt}}}\right) - \arctan\frac{w}{h}\right]^2 \tag{6}$$

In Eqs. (4)–(6), the following variables are defined: IoU is the intersection ratio between the predicted box and the ground truth box, $\lambda^2(a, a^{\mathrm{gt}})$ denotes the Euclidean distance between the centers of the predicted box and the ground truth box. c represents the diagonal distance of the smallest closed bounding box region that can simultaneously contain both the predicted box and the ground truth box. α is a weight parameter for the function. μ is a parameter used to gauge aspect ratio consistency. w and h represent the width and height of the ground truth box, respectively. w^{gt} and h^{gt} represent the width and height of the predicted box, respectively.

However, most of the existing approaches, with CIoU as a representative example, do not involve image dimensions in their consideration. This limitation makes them unable to optimize cases in which prediction boxes and ground truth boxes share the same aspect ratio while possessing significantly different width and height values.

In order to address this issue, the algorithm introduced a loss function called MPDIOU, which is based on minimizing the distance between the top left and bottom right corners of the predicted bounding box and the annotated bounding box. This loss function not only achieves accurate and efficient bounding box regression but also simplifies the calculation process. As shown in Fig. 4, The computation method for MPDIoU is as follows:

$$d_1^2 = \left(x_1^B - x_1^A\right)^2 + \left(y_1^B - y_1^A\right)^2 \tag{7}$$

$$d_2^2 = \left(x_2^B - x_2^A\right)^2 + \left(y_2^B - y_2^A\right)^2 \tag{8}$$

$$\mathrm{MPDIOU} = \frac{A \cap B}{A \cup B} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{9}$$

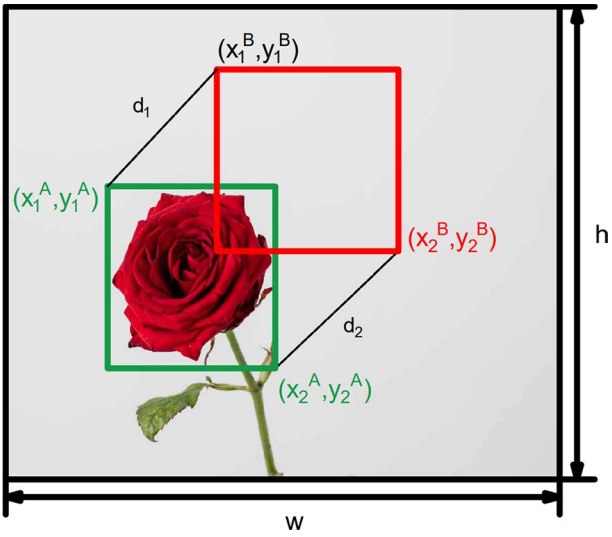

Fig. 4 MPDIOU's factors

In the equations, it can be seen that the loss function of this model is the same as the loss function in YOLOv8, consisting of two parts, regression and classification: A and B represent two arbitrary convex shapes, the width and height of the input image are represented as w and h. (x_1^A, y_1^A) and (x_2^A, y_2^A) represent the top left and bottom right point coordinates of A. Similarly, (x_1^B, y_1^B) and (x_2^B, y_2^B) represent the top left and bottom right point coordinates of B.

Through the implementation of the three aforementioned enhancement methods, we have effectively bolstered the learning capacity of the new network. Consequently, we have developed an SPD-YOLOv8 network algorithm. For detailed insights into the structure, please refer to Fig. 5.

4 Experiment

We conducted both the training and testing phases of our algorithm using Visdrone [38–41] datasets. Our objective was to enhance each stage and make a comprehensive comparison with YOLOv8. In order to demonstrate that the algorithm's improvements enhance the detection accuracy of small-sized objects without compromising the detection accuracy of objects of other sizes, we conducted comparative experiments on the UA-DETRAC [42, 43] dataset as well. Lastly, we used complex scene pictures from diverse complex scenarios to compare the detection effects of the algorithm introduced in this paper with that of the YOLOv8 algorithm.

Following many experiments, it was observed that the algorithm begins to converge at approximately 110 iterations. Through multiple experimental attempts, the following parameters were set with a batch size of 8 and a total of 200 training epochs.

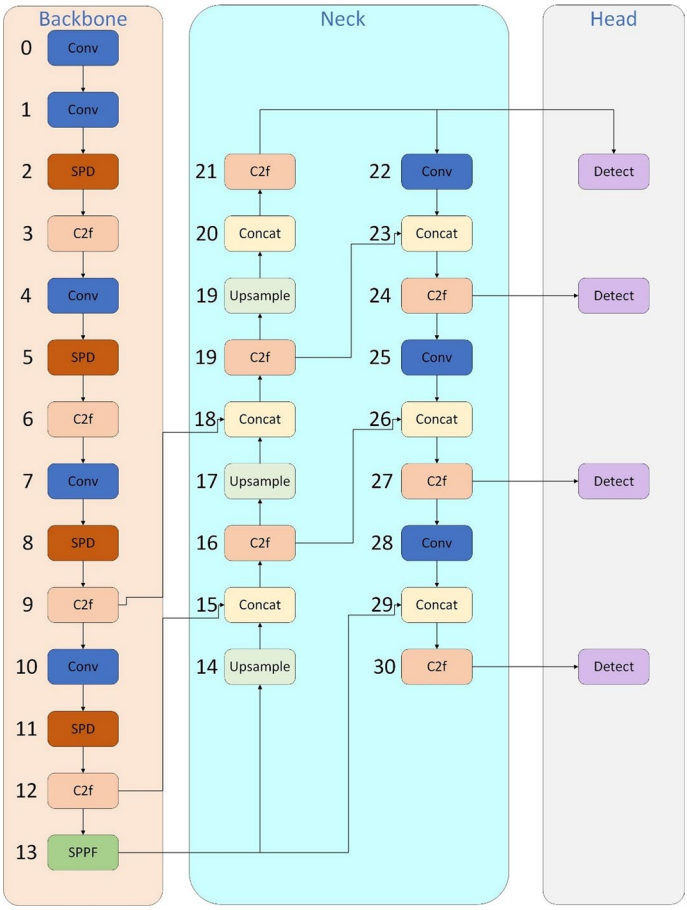

Fig. 5 SPD-YOLOv8 network structure

4.1 Experimental platform

The experiments detailed in this paper were trained and tested by using an NVIDIA RTX 4060 GPU, and an Intel i7-12th CPU. The software platform employed was PyTorch 2.0.1 with CUDA 11.8. Other computer settings are described in Table 1.

4.2 Valuation index

The evaluation metrics included mean average precision (mAP), average precision (AP), precision (P), and recall (R). The formulas for P and R are as follows:

$$P = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FP})} \tag{10}$$

Table 1 Training/testing enviroment

Device	Configuration
Operating system	Windows 11
Processor	12th Gen Intel(R) Core(TM) i7-12650H
GPU	NVIDIA GeForce RTX 4060 Laptop GPU
GPU accelerator	CUDA 11.8
Framework	PyTorch 2.0
Complier IDE	VSCode
Scripting language	Python 3.8

$$R = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FN})} \tag{11}$$

In these equations, TP is the number of samples correctly predicted as positive. FP is the number of samples incorrectly predicted as positive. FN is the number of samples incorrectly predicted as negative.

The formula for average precision and average precision mean are as follows:

$$\mathrm{AP} = \int_0^1 p(r)\mathrm{d}r \tag{12}$$

$$\mathrm{mAP} = \frac{1}{k}\sum_{i=1}^{k} \mathrm{AP}_i \tag{13}$$

In the equations: The parameter k represents the number of categories and AP is the average accuracy of each category.

4.3 Ablation experiments

In order to validate the effect of the improved methods proposed in this paper at each stage for small-size object detection, a series of ablation experiments was conducted at each stage using the Visdrone dataset. The results of these experiments were then compared with those of YOLOv8.

The VisDrone2019 dataset was collected by the AISKYEYE team at Tianjin University. The baseline dataset comprises 288 video clips, totaling 261,908 frames, and 10,209 static images, all captured by UAV. It covers a diverse array of subjects, including pedestrians, vehicles, bicycles, and more, in 14 different urban and rural areas spread across thousands of kilometers in China. These areas encompass both

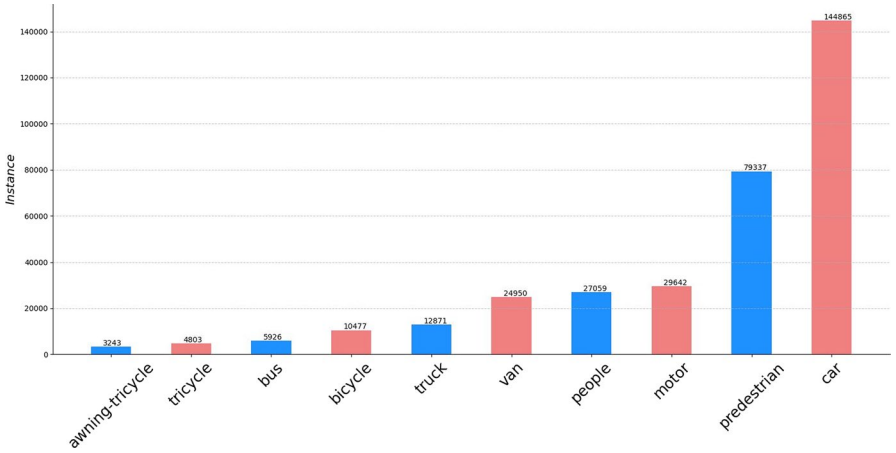

Fig. 6 Number of instances of various categories in the Visdrone dataset

sparse and crowded scenes, as shown in Fig. 6. In this study, the data set was used for this experiment.

In order to clearly demonstrate the authenticity of these experiments, two evaluation metrics, mAP0.5 and mAP0.5:0.95, were used. The results are presented in Table 2. Figure 7 shows the comparison of the training curves between YOLOv8s and SPD-YOLOv8.

In Table 2, Model 1 is the YOLOv8s model. Model 2 adds a small-size target detection layer into the YOLOv8s model. Model 3 replaces the loss function with MPDIOU based on model 2. Building upon Model 3, Model 4 replaces the convolution layer with SPD-Conv.

Based on the data presented in Table 2 and Fig. 7, we can observe the following performance improvements in training on the Visdrone dataset: (1) After adding the small size target detection layer, mAP0.5 increased by 5.22%, and mAP0.5:0.95 increased by 3.75%. (2) After replacing the MPDIOU loss function, mAP0.5 increased by 2.39%, and mAP0.5:0.95 increased by 1.25%. (3) After incorporating the convolutional layer SPD-Conv, mAP0.5 increased by 3.29%, and mAP0.5:0.95 increased by 4.3%. The figure shows that the precision, mAP0.5, and mAP0.5:0.95 of SPD-YOLOv8 have seen significant improvements.

Table 2 Algorithm comparison at each stage

Detection algorithm	*P* (%)	*R* (%)	mAP0.5 (%)	mAP0.5:0.95 (%)
Model 1	42.12	32.96	30.10	17.20
Model 2	44.94	35.06	35.32	20.95
Model 3	50.36	37.72	37.71	22.20
Model 4	**55.24**	**41.42**	**41.00**	**26.50**

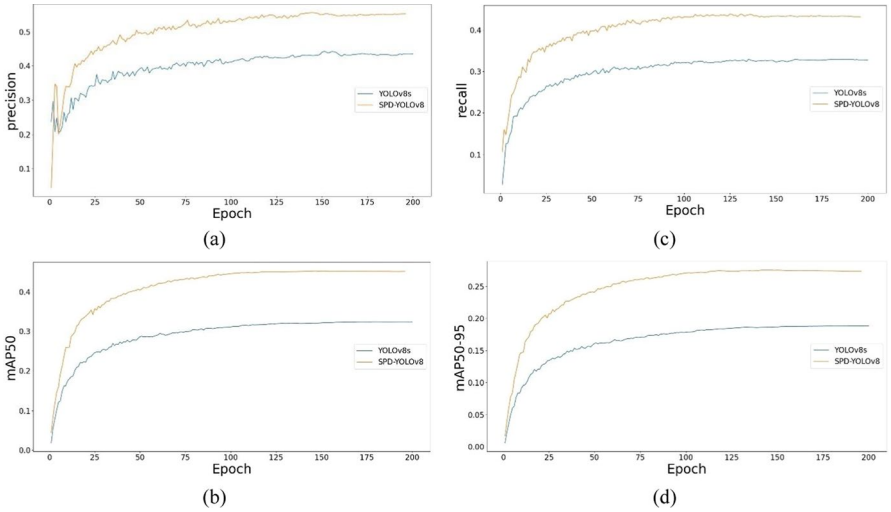

Fig.7 Comparison of training curves. **a** is the precision comparison between YOLOv8s and SPD-YOLOv8. **b** is the mAP0.5 comparison between YOLOv8s and SPD-YOLOv8. **c** is the recall comparison between YOLOv8s and SPD-YOLOv8. **d** is the mAP0.5:0.95 comparison between YOLOv8s and SPD-YOLOv8

However, its recall rate has also increased, indicating that there is still room for further improvement. The effectiveness of the three improvement methods in this experiment can be improved for the following reasons: (a) The addition of the small-size object detection layer facilitates the comprehensive detection of information that might otherwise be lost during down-sampling. (b) Adding the SPD-Conv convolution layer effectively increases the network's attention to small-size objects. (c) The replacement of the MPDIOU loss function effectively resolves the problem of small-size object information being influenced by large-object information. The experiments demonstrated that enhancements at each stage have significantly bolstered the model's learning capabilities.

To compare the improved algorithm to the original algorithm across various object types, we recorded the mAP for ten types of objects in the Visdrone dataset. The specific results are presented in Fig. 8 From these results, it is evident that certain objects exhibit higher recognition accuracy than the overall dataset's average level. The SPD-YOLOv8 consistently exhibits slight improvements for larger objects such as cars and notable improvements for smaller objects like pedestrians, people, tricycles, bicycles, and awning tricycles.

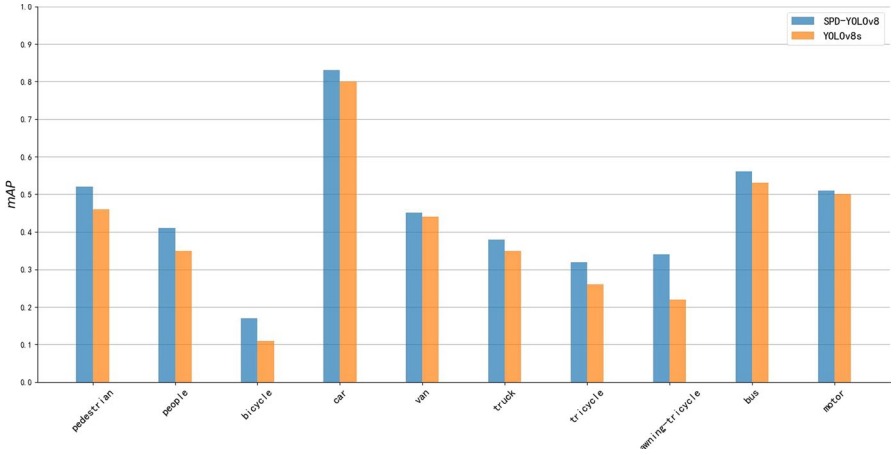

Fig. 8 Comparison between the SPD-YOLOv8 model and the original model, blue is the result of SPD-YOLOv8, and orange is the result of YOLOv8s

4.4 The experimental comparison for objects of different sizes

The second set of experiments was a comparison of varying sizes. The datasets utilized were the UA-DETRAC datasets, UA-DETRAC are divided into four categories: cars, busses, vans, and others, which are taken at 24 different locations in Beijing and Tianjin in China. We tested and compared the performance of YOLOv3, YOLOv5, YOLOv8, and the SPD-YOLOv8 algorithm on the UA-DETRA dataset, as shown in Table 3. During testing on the UA-DETRAC dataset, we recorded the recall rate (R) and mAP0.5. Figure 9 shows the training curves by SPD-YOLOv8.

Based on the data presented in Table 3 and Fig. 9, it's evident that the SPD-YOLOv8 exhibits slightly better experimental results for targets of normal-size when compared to other algorithms. Specifically, on the UA-DETRAC dataset, there was a 1% improvement in recall (R), with negligible differences in mAP0.5. These results have been documented for further validation.

4.5 Comparison of experiments with real-world

To compare the detection performance of YOLOv8s and the SPD-YOLOv8, we selected real-time photos of complex scenes, including well-illuminated roads,

Table 3 Comparison of different algorithms on UA-DETRA datasets

Datasets	Result (%)	YOLOv3	YOLOv5	YOLOv8	SPD-YOLOv8
UA-DETRA	R	74.31	77.62	77.22	**78.22**
	mAP0.5	76.31	78.53	**79.71**	79.11

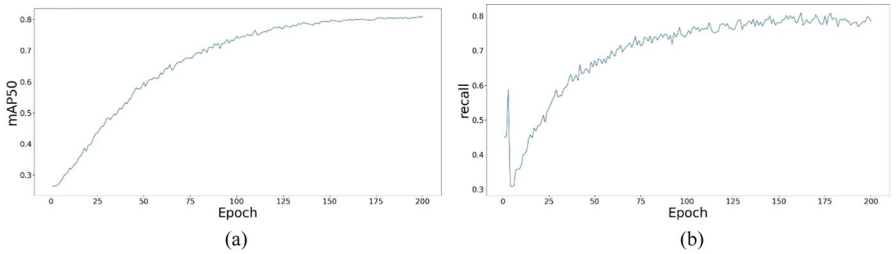

Fig. 9 Training curves trained on the UA-DETRA dataset. **a** is mAP0.5 and **b** is recall of UA-DETRA

and crowded areas. We used models trained by the Visdrone dataset and UA-DETRAC dataset and compared them with the original YOLOv8s to showcase the distinctions between these models.

Figure 10 illustrates the performance comparison between the YOLOv8s model and the SPD-YOLOv8 model on crowded areas captured by UAV after training with the Visdrone dataset, respectively. The left image corresponds to the SPD-YOLOv8 model, while the right image corresponds to the YOLOv8s model.

The detection results, illustrated in Fig. 10, demonstrate that SPD-YOLOv8 performs well in complex scenes. Particularly, in images with higher complexity, YOLOv8's detection method shows errors, while SPD-YOLOv8 accurately locates and classifies targets. These results emphasize that SPD-YOLOv8 enhances detection accuracy and mitigates false positives. Compared to

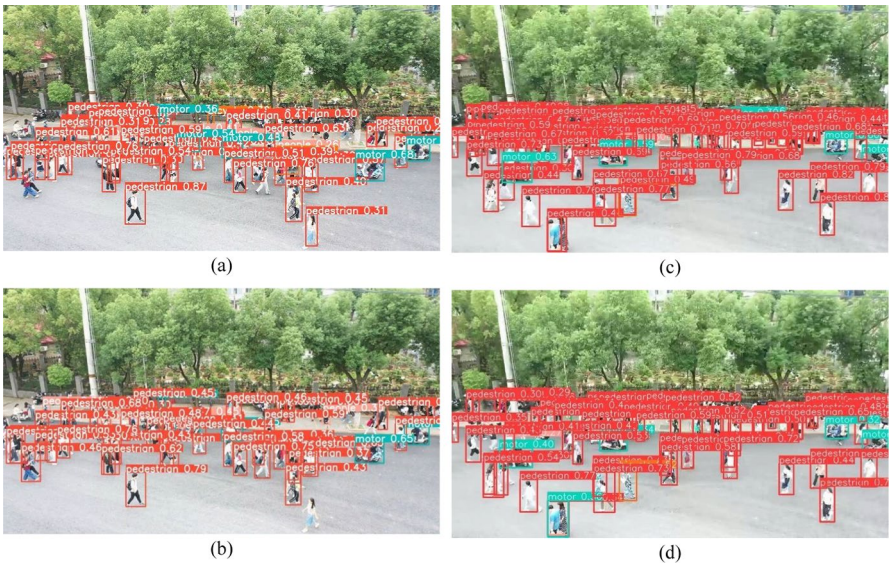

Fig. 10 Comparison between SPD-YOLOv8 and YOLOv8s in a dense crowd captured by the UAV. **a** and **c** correspond to the SPD-YOLOv8 model. **b** and **d** correspond to the YOLOv8s model

YOLOv8s, SPD-YOLOv8 exhibits higher detection accuracy in complex scenes, confirming the effectiveness of the model in detecting complex scenarios.

Figure 11 represents the performance comparison between the YOLOv8s model and the SPD-YOLOv8 model in a well-illuminated road captured by UAV after training on the UA-DETRAC dataset, respectively.

For normal-size objects like busses, the accuracy of detection has been significantly improved. For small-size objects, the recall rate has also been significantly improved. The experimental results indicate that SPD-YOLOv8 performs better than YOLOv8s in detecting objects of different scales, especially normal-size and small-size objects. Furthermore, SPD-YOLOv8 effectively addresses the issues of false positives and false negatives encountered by YOLOv8s in complex scenes. Experimental results in various complex scenarios demonstrate that the proposed SPD-YOLOv8 significantly improves the overall detection rate of the model.

5 Conclusions

In this study, we have proposed a detection model called SPD-YOLOv8 for small-size object detection. The SPD-Conv is used for down-sampling, which is aimed at retaining the original features of the target adequately. The MPDIOU loss function is applied to consider all relevant factors, effectively extracting over-lapping targets. Furthermore, the addition of a small object detection layer improves the alignment of the network's detection layer size with the image.

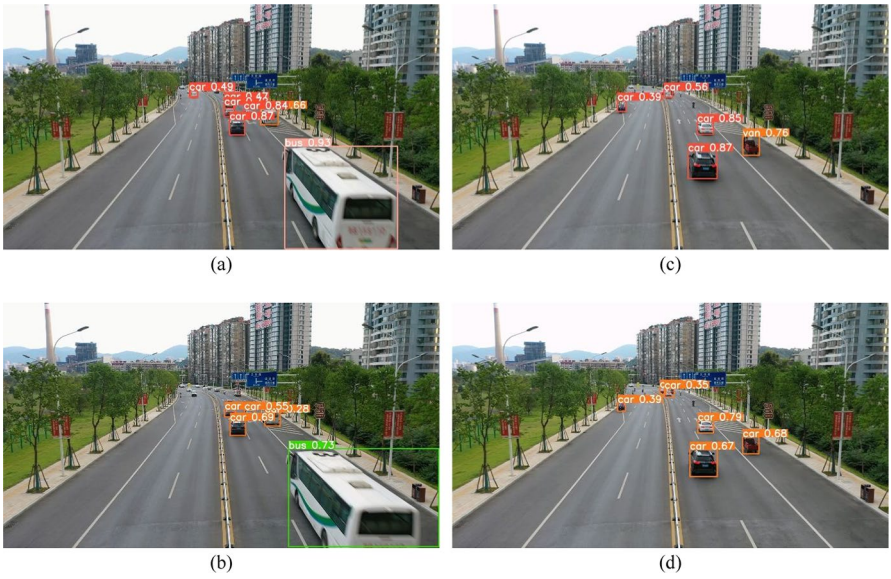

Fig. 11 The comparison of experiments in road scenes captured by UAV. **a** and **c** correspond to the SPD-YOLOv8 model, **b** and **d** correspond to the YOLOv8s model

The proposed SPD-YOLOv8 is experimentally evaluated on the Visdrone and UA-DETRAC datasets. The results indicate that, compared to YOLOv8s, SPD-YOLOv8 demonstrates a 10.9% improvement in mAP0.5 and a 9.3% improvement in mAP0.5:0.95 on the Visdrone dataset, with minimal differences on the UA-DETRAC dataset. In the future, we aim to continue researching target detection algorithms for UAV camera sensors. Improving the inference speed of the model will also be a crucial objective for future research, facilitating deployment on edge devices.

Author contributions R.Z. contributed to conceptualization and the draft writing; E.P. helped with the methodology; Z.L. assisted with software; Q.A. carried out validation, review and editing; T.H. conducted investigation; Y.T. were involved in writing. All authors have read and agreed to the published version of the manuscript.

Funding This work was supported by Natural Science Foundation of Hubei Province (Grant no 2022CFB488) and National Natural Science Foundation of China (Grant no 62071173).

Data availability Not applicable.

Declarations

Conflict of interests The authors declare no conflict of interest.

References

1. Lou H, Duan X, Guo J, Liu H, Guo J, Liu H et al (2023) DC-YOLOv8: small-size object detection algorithm based on camera sensor. Electronics 12(10):2323. https://doi.org/10.3390/electronics12102323
2. Zou M, Yu J, Lv Y, Lu B, Chi W, Sun L (2023) A novel day-to-night obstacle detection method for excavators based on image enhancement and multisensor fusion. IEEE Sens J 23(10):10825–10835. https://doi.org/10.1109/JSEN.2023.3254588
3. Liu H, Li L (2023) Anomaly detection of high-frequency sensing data in transportation infrastructure monitoring system based on fine-tuned model. IEEE Sens J 23(8):8630–8638. https://doi.org/10.1109/JSEN.2023.3254506
4. Guo J, Liu X, Bi L, Liu H, Lou H (2023) UN-YOLOv5s: a UAV-based aerial photography detection algorithm. Sensors 23(13):5907. https://doi.org/10.3390/s23135907
5. Liu H, Yu Y, Liu S, Wang W (2022) A military object detection model of UAV reconnaissance image and feature visualization. Appl Sci 12(23):12236. https://doi.org/10.3390/app122312236
6. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp 1440–1448
7. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: toward real-time object detection with region proposal networks. IEEE T Pattern Anal 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
8. He K, Gkioxari G, Dollar P, Girshick R (2017) Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp 2961–2969
9. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 779–788 https://doi.org/10.1109/CVPR.2016.91
10. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 6517–6525. https://doi.org/10.1109/CVPR.2017.690

11. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 https://doi.org/10.48550/arXiv.1804.02767
12. Bochkovskiy A, Wang C-Y, Liao H-YM (2020) YOLOv4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 https://doi.org/10.48550/arXiv.2004.10934
13. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: single shot multibox detector. In: Computer Vision-ECCV 2016 (ECCV), pp 21–37
14. Liu H, Duan X, Chen H, Lou H, Deng L (2023) DBF-YOLO: UAV small targets detection based on shallow feature fusion. IEEJ T Electr Electr 18(4):605–612. https://doi.org/10.1002/tee.23758
15. Liu H, Sun F, Gu J, Deng L (2022) SF-YOLOv5: a lightweight small object detection algorithm based on improved feature fusion mode. Sensors 22(15):5817. https://doi.org/10.3390/s22155817
16. Wang F, Wang H, Qin Z, Tang J (2023) UAV target detection algorithm based on improved YOLOv8. IEEE Access 11:116534–116544. https://doi.org/10.1109/ACCESS.2023.3325677
17. Zhang Y, Zhang H, Huang Q, Han Y, Zhao M (2024) DsP-YOLO: an anchor-free network with DsPAN for small object detection of multiscale defects. Expert Syst Appl 241:122669. https://doi.org/10.1016/j.eswa.2023.122669
18. Li J, Yuan C, Wang X (2023) Real-time instance-level detection of asphalt pavement distress combining space-to-depth (SPD) YOLO and omni-scale network (OSNet). Automat Constr 155:105062. https://doi.org/10.1016/j.autcon.2023.105062
19. Ao Z, Wan F, Lei G, Ong Y, Xu L, Ye Z et al (2023) FL-YOLOv7: a lightweight small object detection algorithm in forest fire detection. Forests 14(9):1812. https://doi.org/10.3390/f14091812
20. Chen S, Cheng T, Fang J, Zhang Q, Li Y, Liu W, Wang X. (2023) TinyDet: accurate small object detection in lightweight generic detectors. arXiv preprint arXiv: 2304.03428 https://doi.org/10.48550/arXiv.2304.03428
21. Ma M, Pang H (2023) SP-YOLOv8s: an improved YOLOv8s model for remote sensing image tiny object detection. Appl Sci 13(14):8161. https://doi.org/10.3390/app13148161
22. Li A, Sun S, Zhang Z, Feng M, Wu C, Li W (2023) A multi-scale traffic object detection algorithm for road scenes based on improved YOLOv5. Electronics 12(4):878. https://doi.org/10.3390/electronics12040878
23. Han Y, Wang F, Wang W, Li A, Zhang J (2024) YOLO-SG: small traffic signs detection method in complex scene. J Supercomput 80:2025–2046. https://doi.org/10.1007/s11227-023-05547-y
24. Zhang J, Wan G, Jiang M, Lu G, Tao X, Huang Z (2023) Small object detection in UAV image based on improved YOLOv5. Syst Sci Control Eng 11(1):2247082. https://doi.org/10.1080/21642583.2023.2247082
25. Zhai X, Huang Z, Li T, Liu H, Wang S (2023) YOLO-Drone: an optimized YOLOv8 network for tiny UAV object detection. Electronics 12(17):3664. https://doi.org/10.3390/electronics12173664
26. Siliang M, Yong X (2023) MPDIoU: a loss for efficient and accurate bounding box regression. arXiv preprint arXiv:2307.07662 https://doi.org/10.48550/arXiv.2307.07662
27. Wang C-Y, Bochkovskiy A, Liao H-YM (2023) YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 7464–7475
28. Wang C-Y, Liao H-YM, Wu Y-H, Chen P-Y, Hsieh J-W, Yeh I-H (2020) CSPNet: a new backbone that can enhance learning capability of CNN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 390–391
29. Zhang S, Chi C, Yao Y, Lei Z, Li SZ (2020) Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 9759–9768
30. Zheng Z, Wang P, Ren D, Liu W, Ye R, Hu Q, Zuo W (2022) Enhancing geometric factors in model learning and inference for object detection and instance segmentation. IEEE T Cybern 52(8):8574–8586. https://doi.org/10.1109/TCYB.2021.3095305
31. Li H, Xiong P, An J, Wang L (2018) Pyramid attention network for semantic segmentation. arXiv preprint arXiv:1805.10180 https://doi.org/10.48550/arXiv.1805.10180
32. Lin T-Y, Dollar P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2117–2125
33. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T et al (2017) MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 https://doi.org/10.48550/arXiv.1704.04861

34. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 4510–4520
35. Howard A, Sandler M, Chu G, Chen L-C, Chen B, Tan M et al (2019) Searching for MobileNetV3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp 1314–1324
36. Wang L, Guo S, Huang W, Qiao Y (2015) Places205-VGGNet models for scene recognition. arXiv preprint arXiv:1508.01667 https://doi.org/10.48550/arXiv.1508.01667
37. Xu Y, Xie L, Xie C, Dai W, Mei J, Qiao S et al (2023) BNET: batch normalization with enhanced linear transformation. IEEE T Pattern Anal 45(7):9225–9232. https://doi.org/10.1109/TPAMI.2023.3235369
38. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90. https://doi.org/10.1145/3065386
39. Du D, Zhu P, Wen L, Bian X, Lin H, Hu Q et al (2019) VisDrone-DET2019: the vision meets drone object detection in image challenge results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp 0–0
40. Zhu P, Wen L, Du D, Bian X, Ling H, Hu Q et al (2018) VisDrone-DET2018: the vision meets drone object detection in image challenge results. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 0–0
41. Cao Y, He Z, Wang L, Wang W, Yuan Y, Zhang D et al (2021) VisDrone-DET2021: the vision meets drone object detection challenge results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp 2847–2854
42. Wen L, Du D, Cai Z, Lei Z, Chang M-C, Qi H et al (2020) UA-DETRAC: a new benchmark and protocol for multi-object detection and tracking. Comput Vis Image Und 193:102907. https://doi.org/10.1016/j.cviu.2020.102907
43. Lyu S, Chang M-C, Du D, Wen L, Qi H, Li Y et al (2017) UA-DETRAC 2017: report of AVSS2017 & IWT4S challenge on advanced traffic monitoring. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp 1–7. https://doi.org/10.1109/AVSS.2017.8078560