An Improved Small Target Detection Algorithm Based on YOLOv8s

Guoqiang MA [1], Chuntian XU [1], Zong XU [2], Xiangyang SONG [1]*

[1]School of Mechanical Engineering and Automation, Liaoning Univ. of Science and Technology, 114002, Anshan, China
[2]Angang Steel Co. LTD, Anshan Iron & Steel, Anshan, Liaoning, China

maguo92896@gmail.com, xuchuntian@163.com*, 1105229841@qq.com, 283873951@qq.com

Submitted November 29, 2024 / Accepted January 29, 2025 / Online first April 25, 2025

Abstract. *Due to challenges such as the small size of targets, complex backgrounds, limited feature extraction capabilities, and frequent false positives and false negatives, traditional detection algorithms often perform poorly in small object detection tasks. To address these challenges, this paper proposes an enhanced small object detection algorithm, SOD-YOLO, based on YOLOv8s. First, the S_C2f_CAFM module is integrated into the feature extraction network, enabling the effective capture of fine-grained local features and broad contextual information, while simultaneously reducing model parameters and computational complexity. Second, in the feature fusion stage, the redesigned bidirectional feature pyramid network employs a spatial context awareness module to extract key features, adding a top-down path to optimize feature fusion and enhance discriminative information. In the Neck section, the D_C2f_MSPA module is introduced, which, while being lightweight, accurately models channel dependencies in feature maps, effectively reducing both false positives and false negatives for small objects. Finally, the inclusion of Normalized Wasserstein Distance (NWD) further improves detection accuracy and reduces the model's sensitivity to small positional deviations in small objects. Experimental results on the DOTAv1.0, VisDrone2019, and TT100K datasets confirm that SOD-YOLO achieves excellent performance, demonstrating the effectiveness of the modifications made to the original YOLOv8 model.*

Keywords

YOLOv8, small object detection, attention mechanism, feature fusion, loss function

1. Introduction

Object detection is a critical task in the field of computer vision, with its primary objective for the automatic identification and localization of specific objects within images or videos. With the rapid advancements in computer vision technologies and the widespread adoption of deep learning algorithms, object detection has been widely used in areas such as face recognition, identity authentication systems, and autonomous driving [1], [2]. Although deep learning-based approaches have achieved remarkable progress in detecting medium and large-sized objects, small object detection remains a challenging and exploratory area. This difficulty arises from several inherent characteristics of small objects, including their limited size, low resolution, insufficient contextual information, and disproportionate scale relative to the background and so on. These factors contribute to challenges such as inaccurate localization and an increased prevalence of false positives and false negatives.

In the domain of object detection, small object detection algorithms can be broadly categorized into traditional methods and deep learning-based approaches. However, traditional object detection algorithms [3], [4] face significant limitations, including restricted feature representation capabilities, insufficient contextual information, and high computational complexity and so on. These drawbacks make detecting small objects inadequate in complex scenarios. Deep learning-based algorithms, on the other hand, can be classified into two-stage and one-stage methods, respectively. Two-stage algorithms include R-CNN [5], Fast R-CNN [6], and Faster R-CNN [7], which are known for their high accuracy but often exhibit increased computational costs. In contrast, one-stage algorithms, such as YOLOv1 [8], YOLOv2 [9], YOLOv3 [10], YOLOv4 [11], YOLOv5 [12], YOLOv7 [13], YOLOv8 [14], SSD [15], YOLOv9 [16], YOLOv10 [17], and YOLOv11 are characterized by lower computational complexity and higher real-time performance, which are more widely adopted in practical applications.

The first-generation YOLOv1 algorithm, while faster than the SSD algorithm, performs poorly in detecting objects at close range and very small objects. YOLOv2 introduced a new direction by incorporating an anchor mechanism that is more suitable for small object detection. YOLOv3 further advanced YOLOv2 by adopting a pyramid network structure for multi-scale fusion, which significantly improved small object detection. YOLOv4 retained the header from YOLOv3 and introduced several additional improvements, maintaining YOLOv3's high accuracy in small object detection while enhancing detection performance for objects of varying sizes. YOLOv5 built upon YOLOv4, incorporating

DOI: 10.13164/re.2025.0206

the Generalized Intersection over Union (GIoU) loss function and optimizers such as Adam, which increased both accuracy and speed in detecting densely occluded objects. Following this, YOLOv6 and YOLOv7 were released, bringing improvements in network architecture and training strategies. YOLOv8, released by Ultralytics in 2013, introduced a new architecture with updated convolutional layers and detection heads, achieving significant improvements in both speed and accuracy compared to its predecessors, making it suitable for real-time object detection. The release of YOLOv9 marked a major advancement in the YOLO series. Building on YOLOv8, YOLOv9 introduced a dynamic receptive field mechanism and adaptive feature fusion modules, enabling the network to capture multi-scale information more efficiently, excelling in small object detection and complex scenarios. YOLOv10 advanced further by incorporating a Generative Adversarial Network (GAN) to generate high-quality pseudo-samples, thereby enhancing the model's generalization ability. It also used gradient sparsity regularization during optimization, improving the efficiency and stability of training. YOLOv11, released shortly afterward, elevated the YOLO framework to new heights by integrating global feature enhancement modules and large-scale mixed training strategies, enabling unified adaptation to various object shapes, sizes, and complexities, while significantly improving detection accuracy and robustness.

Despite the substantial progress made in terms of accuracy, scene adaptability, and detection performance with YOLOv9 to YOLOv11, these models still exhibit some notable drawbacks compared to YOLOv8. First, the complexity of the models has increased significantly, leading to higher hardware costs for both training and deployment, which makes them less suitable for resource-constrained environments. Secondly, due to the introduction of complex mechanisms, such as dynamic receptive fields and adaptive feature fusion, the real-time performance of the network during inference has slightly declined compared to YOLOv8, making it difficult to meet extreme real-time requirements. Additionally, these improvements have led to a higher parameter count and increased training difficulty, placing greater demands on the quality and diversity of training data. This, in turn, may negatively impact the generalization ability of the models on smaller datasets. These limitations hinder the widespread applicability of these models in specific scenarios. As a result, this paper proposes a new network architecture based on YOLOv8.

In recent years, researchers have proposed various enhancements to small object detection algorithms. For instance, Gong et al. [18] refined the feature fusion strategy in the Feature Pyramid Network (FPN), thereby improving detection accuracy and optimizing the transfer information between shallow and deep network layers. However, this approach still struggles with the detection of extremely small objects. Bai et al. [19] incorporated a multitask Generative Adversarial Network (GAN) into the detection framework, achieving a synergy between object detection and data augmentation. By leveraging GAN to enhance small object feature representation, detection performance was significantly improved. But the complexity of GAN training and the high computational resources require present notable challenges. In another study, for small object detection in drone imagery, Hong et al. [20] introduced an adaptive scale selection pyramid network, which can effectively address resolution variations in high-altitude images. Nevertheless, its applicability is limited to specific scenarios and lacks generalization. Wang et al. [21] proposed an improved method based on DIOU-NMS, which optimizes the non-maximum suppression (NMS) mechanism and significantly enhances the detection accuracy of dense small objects in aerial images. Chen et al. [22] proposed a defect detection method based on DCAM-YOLOv5, which improves feature extraction capabilities by introducing a deep channel attention module (DCAM), thereby significantly increasing the accuracy of tunnel lining defect detection. However, both methods have certain drawbacks. First, the increased model complexity leads to higher computational overhead during both training and inference, limiting their applicability in resource-constrained environments. Second, while these methods perform well in specific scenarios, their generalization and adaptability are relatively limited. When applied to different scenarios, they may require further parameter tuning or retraining. Additionally, both methods rely heavily on high-quality labeled data, and their performance may degrade when such data is insufficient. Yao et al. [23] developed HP-YOLOv8, which integrates a C2f-D-Mixer module and a Bi-Level Routing Attention-based FPN (BGFPN) to improve the fusion of local and global information. This design enhances the detection of small objects in complex backgrounds. Additionally, after that the proposed Shape Mean Perpendicular Distance Intersection over Union (SMPDIoU) loss function facilitates more accurate bounding box localization for small objects. However, the increased model complexity introduces higher computational demands. Ge Zheng et al. [24] proposed an Anchor-Free design combining a Decoupled Head architecture with efficient training techniques such as Mosaic data augmentation and SimOTA label assignment. This approach mitigates accuracy loss caused by the anchor mechanism in small object detection, simplifies the training and inference processes, and improves detection efficiency. However, the performance of the Anchor-Free mechanism in highly complex backgrounds requires further optimization. Yin et al. [25] proposed a novel framework specifically designed to address small object detection in marine environments using one-dimensional time-series signals. This approach utilizes an enhanced convolutional neural network (CNN), integrating temporal information and spatial feature fusion to improve detection accuracy for small objects. Additionally, a feature enhancement mechanism is introduced to optimize the model's capability in recognizing small objects. However, the method has several limitations. First, the model’s generalization ability is limited under complex sea conditions, with detection accuracy likely to decrease in extreme weather or noisy signal environments. Second, the network’s complexity and high computational resource demands during training can lead to performance bottlenecks in real-time detection. Moreover, the model’s performance is heavily dependent on high-quality,

accurately labeled data, and detection accuracy significantly deteriorates when data availability is insufficient. In short, although these methods above have achieved substantial progress, some limitations still persist, including trade-offs between model complexity and accuracy, increase parameter sizes and frequent leak detection.

Therefore, to address the aforementioned challenges, an improved algorithm is proposed in this paper, SOD-YOLO, based on YOLOv8s. The primary contributions of this work are as follows:

(1) Feature Extraction Stage

The S_C2f_CAFM module is designed to reduce model parameters and computational complexity while effectively extracting fine-grained local features and extensive contextual information. This can significantly enhance the detection performance for small objects. Meanwhile, to better adapt to the scales of small objects, here the Spatial Pyramid Pooling Fast (SPPF) module is improved, and the Context Aggregation method [26] is introduced to further enhance the model's capability for understanding and representing object features, respectively.

(2) Feature Fusion Stage

To enhance feature representation and effectively reduce the missed and false detection rates for small objects, a bidirectional feature pyramid network is developed, incorporating a spatial context-aware module to extract key features from shallow networks. A top-down pathway is added to optimize feature fusion, resulting in the generation of more discriminative information, a novel small-object detection layer is introduced to fully integrate shallow and deep feature information, and a D_C2f_MSPA module is designed for the Neck component, modeling the dependencies between feature map channels with high precision while maintaining a lightweight architecture, respectively. Furthermore, a lightweight upsampling module, Dysample [27], is adopted to eliminate reliance on high-resolution guiding features.

(3) Loss Function

To reduce the model's sensitivity to slight positional deviations of small objects, thereby to improve its detection accuracy, Normalized Wasserstein Distance (NWD) [28] is introduced to replace the Complete Intersection over Union (CIoU) [29] as the bounding box similarity measurement.

Compared to the methods mentioned above, the model proposed in this paper has lower complexity, reduced computational overhead during both training and inference, and does not rely on increasing the number of parameters to improve accuracy. Additionally, the model requires less stringent training data quality and diversity while demonstrating stronger generalization and adaptability. These advantages will be validated in the experimental section.

The remainder of this paper is organized as follows: Section 2 provides a detailed description of the proposed algorithm, Section 3 outlines the datasets and experimental setup, Section 4 presents an evaluation of the method's performance, Finally, in Sec. 5, our research findings and outline future important directions for studies are concluded.

2. Methodology

2.1 Fundamental YOLO v8 Model

YOLOv8, a target detection algorithm offers significant advancements in detection accuracy and speed compared to earlier YOLO models, of which network structure, depicted in Fig. 1, comprises three primary components of the backbone, neck, and the detection head.

The backbone utilizes a modified CSPDarknet53, where the input features undergo five down-sampling operations to generate feature layers at five different scales (P1–P5). The original Cross Stage Partial (CSP) module is replaced by the C2f (Cross Stage Partial Network Fusion) module, which improves the information flow through gradient splitting while maintaining a lightweight design. Additionally, the CBS module composed of the convolution, batch normalization, and the SiLU activation function performs these operations sequentially to produce the final output. The backbone also incorporates the SPPF module, which pools the input features into fixed-size feature maps to enhance the feature representation.

The neck employs a PAN-FPN [30] structure, which integrates the Path Aggregation Network (PAN) into the traditional FPN [31] to address FPN's limitations in capturing localization information. PAN [32] enhances the feature representation through bottom-up high-order feature fusion, effectively combining the shallow positional information with deep semantic information to improve the feature diversity and integrity.

The detection head adopts a decoupled structure with two independent branches dedicated to predicting target categories and location information. Each branch is optimized with specific loss functions: Binary Cross Entropy Loss (BCE Loss) for classification and Distribution Focal Loss (DFL) [33] combined with CIoU for bounding box regression. This architecture not only improves the detection accuracy but also accelerates the model convergence.

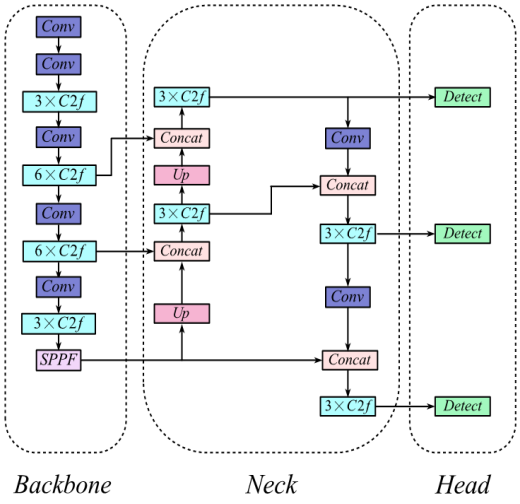


Fig. 1. Structure of YOLOv8.

2.2 Overview of SOD-YOLO

Although YOLOv8 surpasses other mainstream algorithms in small object detection accuracy, it still faces challenges such as recognition difficulties, inaccurate localization, and occurrences of false negatives and false positives. To address these limitations, an enhanced small object detection algorithm is proposed, termed SOD-YOLO (as shown in Fig. 2), in which the specific improvements are introduced compared to the original YOLOv8 and are summarized in Fig. 3.

First, the SPD-Conv [34] (Space-to-Depth Convolution) and CAFM [35] (Convolution and Attention Fusion Module) are incorporated into the backbone feature extraction network, creating the SPD_CAFM Block. And this block forms the newly designed S_C2f_CAFM module combined with the C2f module, which significantly reduces the model's parameter count and floating-point operations while enhancing the extraction of local features and global contextual information. In the original YOLOv8 backbone, the repeated application of the C2f module in the P4 and P6 layers often results in feature degradation. To deal with this, the CAFM module is employed to improve the fine-grained feature representation and reduce the repetitive use of the C2f module, achieving more efficient feature extraction. Additionally, a Context Aggregation Module is also introduced to enhance the model's ability to understand and represent target features at the base of the backbone network. Thus, the SPPF module is improved to better accommodate the scales of small objects, correspondingly.

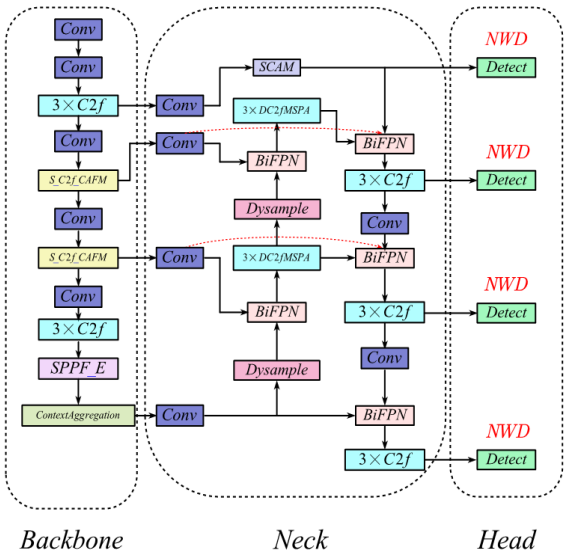


Fig. 2. Structure of SOD-YOLO.

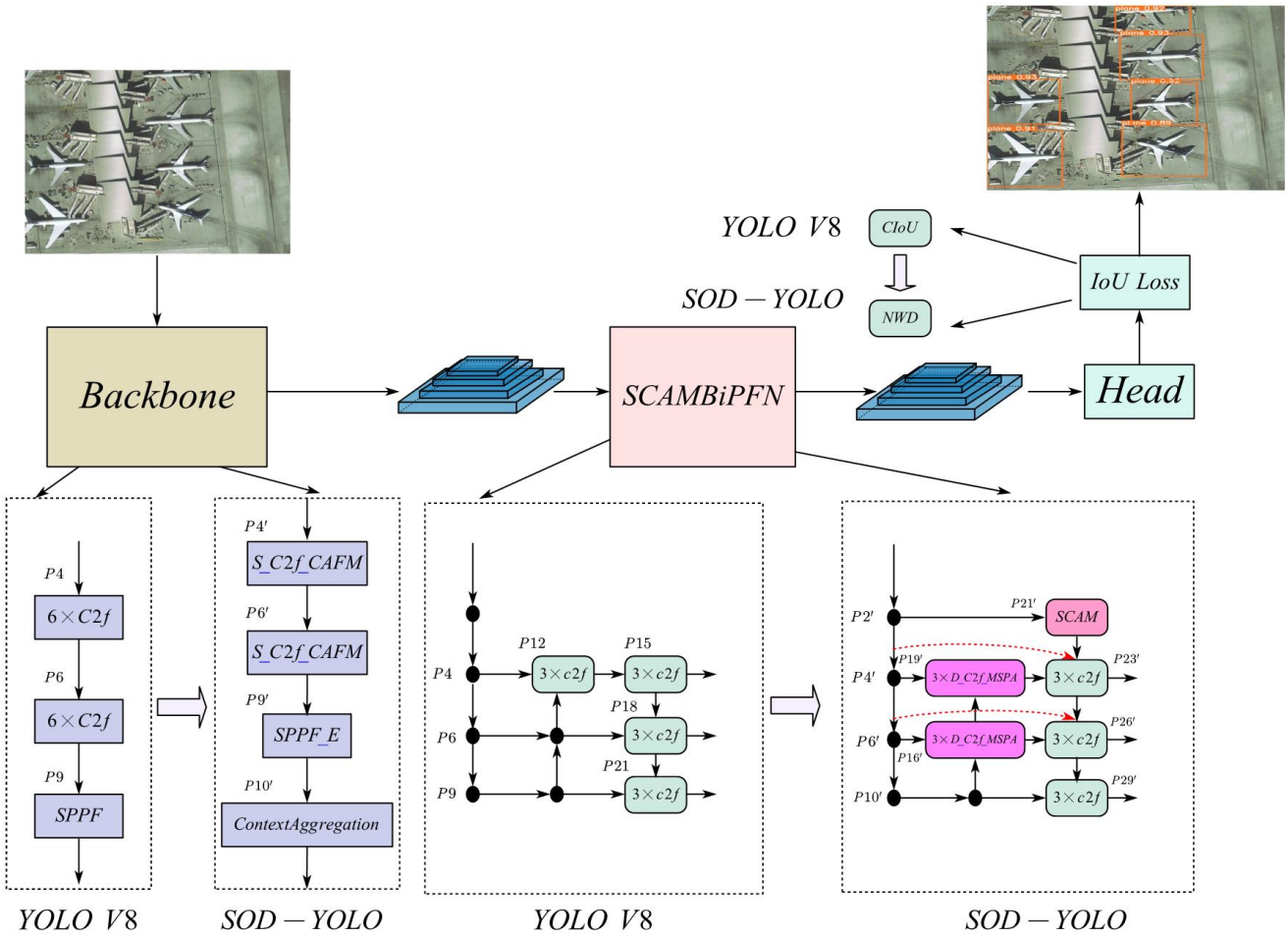


Fig. 3. Comparison of YOLOv8 and SOD-YOLO.

Next, the Bidirectional FPN (BiFPN) is redesigned. Compared to the pyramid network in the original YOLOv8, the redesigned BiFPN incorporates DCNv4 [36] (Deformable Convolution v4) and MSPA [37] (Multi-scale Spatial Pyramid Attention) mechanisms in the neck, forming the new D_C2f_MSPA module. This module not only reduces computational complexity but also accurately models dependencies between the feature map channels, thereby enhancing the feature representation and effectively mitigating false negative and false positive rates for small objects. To further improve the feature integration, fusion paths are introduced for the P4 and P6 layers in the feature pyramid network back end. Meanwhile, to optimize performance, the Spatial Context-Aware Module (SCAM) [38] is integrated, which guides the learning of pixel relationships in both spatial and channel dimensions, facilitating cross-channel and spatial context interactions. Because this redesigned feature pyramid network generates features more efficiently, the number of feature maps produced by the C2f module can be reduced, maintaining performance and controlling the model's parameter count.

Finally, an additional detection head is specifically designed for very small objects, used to cover very small, small, medium, and large objects. Thus, the SOD-YOLO's detection range is expanded. Furthermore, a new bounding box similarity metric, NWD, is proposed and a regression loss function is designed based on this metric to enhance bounding box localization accuracy. The structure and principles of each module are detailed in the following sections.

2.3 S_C2f_CAFM Module

To address the challenges of small object feature loss and insufficient scale-capturing capability, the construction of an SPD_CAFM Block is proposed that combines SPD-Conv and CAFM with the C2f module to form the novel S_C2f_CAFM module. This module effectively reduces the number of parameters and floating-point operations while extracting fine-grained local features and capturing broader contextual information, thereby enhancing the model's performance in small object detection tasks.

In object detection, the CNN serves as the foundational technology. However, when the resolution of detection images is low or the objects are small, network performance often deteriorates for the stride and pooling layers in CNN repeatedly down sampling feature maps. In deeper network structures, this leads to the loss of fine-grained information, compromising the network's ability to learn effective features. To address this issue, here the traditional stride and pooling layers are replaced with the SPD-Conv module.

The process of the SPD-Conv module is illustrated in Fig. 4. Figure 4(a) represents the input feature map with dimensions $S \times S \times C_1$. After the slicing operation in Fig. 4(b), the input is divided into four sub-maps, shown in Fig. 4(c), each with dimensions $(S/2) \times (S/2) \times C_1$. The SPD layer then concatenates all sub-maps along the channel dimension to produce a feature map with dimensions $\text{scale}^2 \times C_1$, as shown in Fig. 4(d). Finally, a convolutional

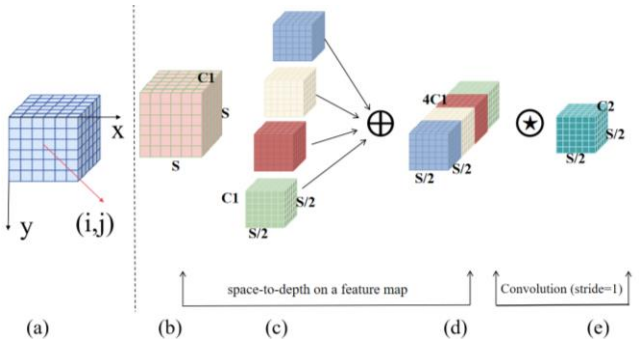


Fig. 4. Structure of SPD-Conv with scale = 2.

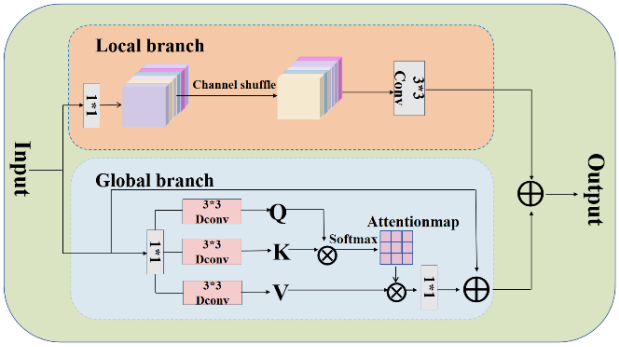


Fig. 5. CAFM structure.

layer (Conv) with a stride of 1 and an output channel count of C_2 is applied, transforming the feature map into the output map with dimensions $(S/2) \times (S/2) \times C_2$, as depicted in Fig. 4(e).

The CAFM enhances the model's capability to integrate inter-channel information and capture fine-grained features by combining local features extracted through convolution operations with global features and long-range dependencies obtained via an attention mechanism. The structure of CAFM is depicted in Fig. 5.

It is seen from Fig. 5 that the CAFM module comprises two local and global branches. A 1×1 convolution is initially applied to adjust the channel dimensions, improving the inter-channel interactions and facilitating information integration in a local branch. And a channel shuffling operation is then performed to further integrate the channel information. This operation divides the input tensor into multiple groups along the channel dimension. Each group undergoes depthwise separable convolution, and the outputs are concatenated along the channel dimension to generate the final output features. Finally, a $3 \times 3 \times 3$ convolution is applied to extract enriched local features.

But for the global branch, the branch begins with a 1×1 convolution and a 3×3 depthwise convolution to generate queries (Q), keys (K), and values (V), respectively, producing three tensors with dimensions $H \times W \times C$. Where the tensor Q is reshaped into $Q \in R^{HW \times C}$, and K is reshaped into $K \in R^{C \times HW}$. A matrix multiplication operation is then performed to compute $Q \times K$, followed by the application of the Softmax function to generate an attention map $A \in R^{C \times C}$. This design significantly reduces the computational complexity compared to traditional methods that

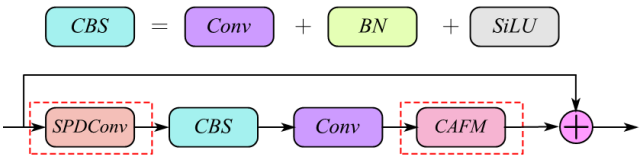


Fig. 6. Structure of SPD_CAFM block.

compute a large-scale attention map $A \in R^{HW \times HW}$, making the approach more efficient.

By integrating the outputs from both branches, the CAFM module effectively balances local and global feature extraction, enhancing the model's overall feature representation and improving its performance in small object detection tasks. Then the SPD_CAFM Block, constructed using SPDConv and CAFM, replaces the Bottleneck structure in C2f and is illustrated in Fig. 6.

It is seen from Fig. 6 that the process begins with the input passing through SPDConv for initial convolution, followed by the CBS module. This expands the number of output channels to twice the original, ensuring feature diversity. The CBS module consists of a 1×1 convolution, normalization, and activation functions. Here the 1×1 convolution is used to adjust the channel dimensions, to perform both upscaling and downscaling operations. Subsequently, the channels are reduced by another 1×1 convolution to maintain consistency with the input. Next, the CAFM attention module is incorporated to integrate inter-channel information, capture global features and long-range dependencies, obtain multi-scale feature representations, and enhance contextual information for small objects. Finally, the convolutional results are concatenated with the unprocessed input channels using a concat operation, effectively reducing redundant information.

The S_C2f_CAFM employs the SPD_CAFM Block as its Bottleneck structure. Initially, the CBS module expands the output channels to 2c. A Split operation then divides the channels into two parts, which are processed by n SPD_CAFM Blocks in sequence. This design reduces the number of parameters and computational cost while enriching the gradient flow structure. Afterward, the n concatenated SPD_CAFM Blocks are merged with the split channel feature maps, producing an output feature map with $(n + 2) \times c$ channels. Finally, the CBS module adjusts the total channel count to C_2. An example structure of S_C2f_CAFM with $n = 3$ is shown in Fig. 7.

2.4 SPPF_E Module

Inspired by YOLOv9, to enhance the fusion of shallow and deep information, mitigate the loss of fine-grained features caused by multiple convolutions, and improve the network's attention and perception capabilities for small object regions, the SPPF_E structure is introduced, which combines the efficient layer aggregation strategy of ELAN [39] with the SPPF module. Here the ELAN employs an efficient layer aggregation mechanism that not only enhances the network's feature representation capabilities but also achieves a significant balance between the network complexity and computational efficiency. The structure of SPPF_E is illustrated in Fig. 8.

It is seen from Fig. 8 that the input feature map firstly undergoes basic feature extraction and preprocessing through the CBS block, generating high-level features. Subsequently, the input feature map is processed using pooling kernels of varying sizes to capture target features across multiple spatial scales. This operation improves the model's ability to perceive small and large targets effectively. The features extracted from these different pooling paths are then fused through layer-by-layer stacking or concatenation, enabling the model to capture rich information from diverse spatial scales and further enhancing its feature representation capabilities.

In all, compared to the original SPPF structure, the SPPF_E structure provides several key advantages. Firstly, it effectively mitigates gradient vanishing, ensuring stable updates in deep networks during training. Furthermore, through parameter simplification and optimized layer design, high model performance is maintained while significantly reducing computational overhead.

2.5 D_C2f_MSPA Module

To address key challenges in small object detection, including low resolution, scale variation, positional instability, background interference, blurred boundaries, and occlusion and so on, the DCNv4 and MSPA attention mechanism

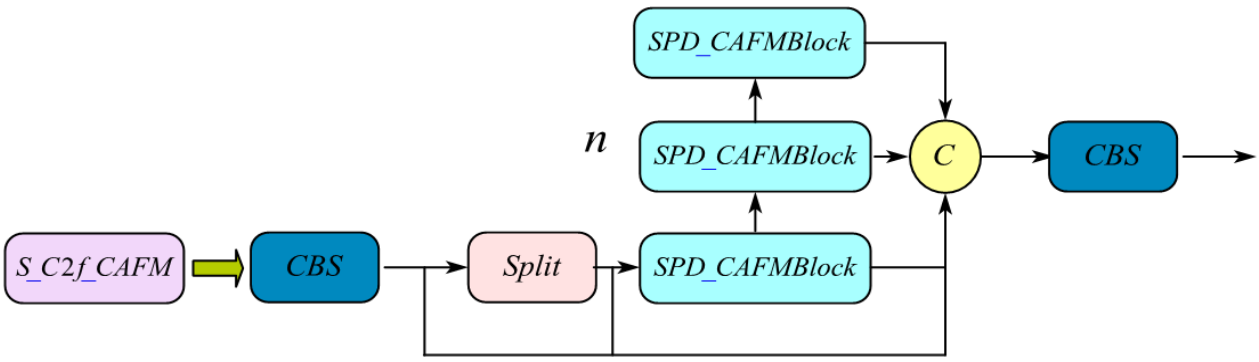


Fig. 7. Structure of S_C2f_CAFM.

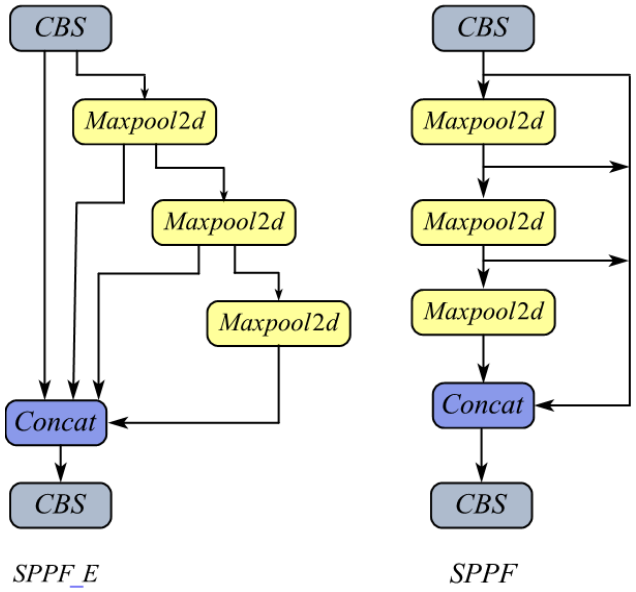


Fig. 8. Structural comparison of SPPF_E and SPPF.

is integrated into the neck of the network, and a novel D_C2f_MSPA module is proposed. This module preserves the lightweight characteristics of the network while effectively modeling dependencies between feature map channels, which enhances the feature representation, reduces the missed and false detection of small objects, and improves the overall detection performance.

Since traditional convolutional operations mainly rely on fixed-size kernels, which can only capture specific regions and are limited in their ability to model irregular object deformations, this reduces the model's capacity to effectively represent complex geometries. Deformable Convolutional Networks (DCN) address this issue by introducing adjustable kernels that allow spatial offsets during operations. These kernels can be dynamically adapted to varying scales and positions of small objects, thereby improving the model's ability to capture features and handle complex geometric deformations (see Fig. 9). Herein some traditional convolutions are replaced in the C2f structure with the latest deformable convolution, DCNv4, to improve the extraction of small object features. This substitution enhances the model's robustness and ensures better alignment with the target dimensions.

The MSPA attention mechanism (see Fig. 10) is composed of three core components: the HPC module, the SPR module, and the Softmax operation. The HPC module extracts fine-grained, multi-scale spatial information through hierarchical residual connections; the SPR module combines structural regularization with structural information via an adaptive combination mechanism enabling the learn-

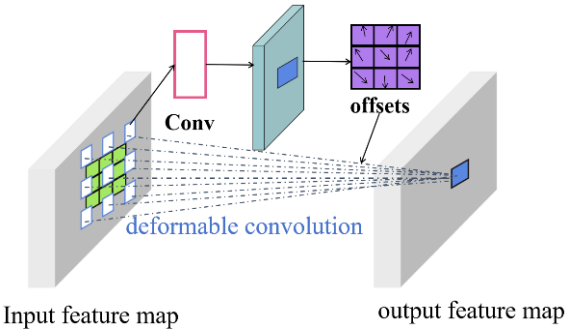


Fig. 9. Structure of DCN.

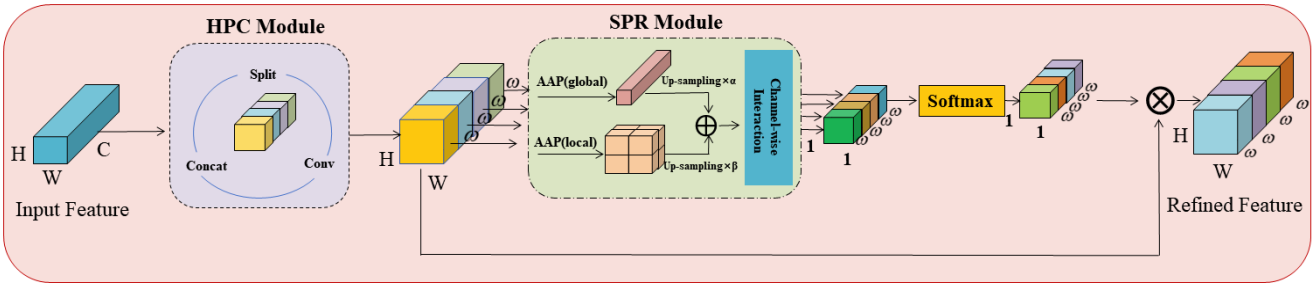


Fig. 10. MSPA attention mechanism structure diagram.

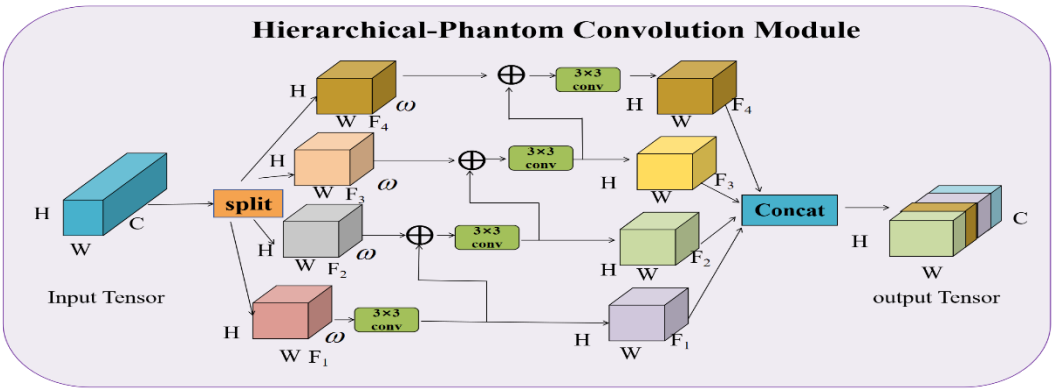


Fig. 11. HPC module structure.

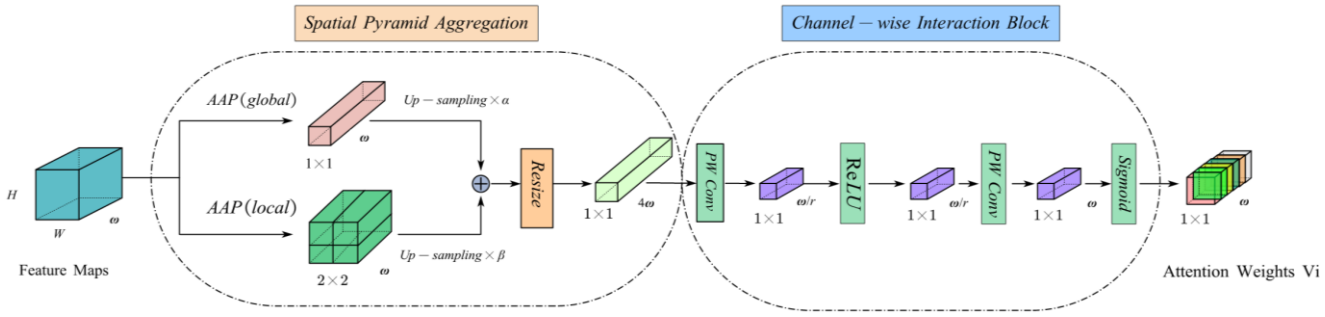


Fig. 12. SPR module structure.

ing of channel attention weights and facilitating cross-dimensional interactions; and the Softmax operation recalibrates the channel attention weights to establish long-range dependencies.

The structure of the HPC module is illustrated in Fig. 11. Here, the term ***Split*** represents uniform segmentation along the channel dimension, ***Conv*** refers to a 3×3 standard convolution layer followed by batch normalization, and ***Concat*** denotes channel-wise feature concatenation. Figure 12 depicts the schematic of the SPR module, which consists of two primary components: the spatial pyramid aggregation block and the channel interaction block. The former employs two layers of pyramid-shaped adaptive average pooling at different scales to combine structural regularization and structural information within the attention pathway. The latter generates attention maps from the outputs of the spatial pyramid aggregation block. Here, ***AAP*** refers to adaptive average pooling, ***Up-sampling*** uses nearest-neighbor interpolation, and ***PWConv*** represents pointwise convolution.

The D_M_BottleNeck structure is illustrated in Fig. 13. In this structure, the lower branch initially applies two DCNv4 convolutions. These convolutions enhance the model's flexibility in addressing object deformations, pose variations, and complex backgrounds, and thus, also improving its ability to capture the shapes and structures of small objects. Subsequently, features from the upper and lower branches are concatenated along the channel dimension to ensure channel consistency. Finally, the MSPA attention mechanism is applied to refine the model's focus on critical regions, thereby enhancing image understanding and processing performance.

The newly designed D_C2f_MSPA module is depicted in Fig. 14. The traditional BottleNeck structure in C2f is substituted with the D_M_BottleNeck structure in this module, which enriches the gradient flow and improves the diversity of network learning by integrating the feature information from different stages. By incorporating DCNv4

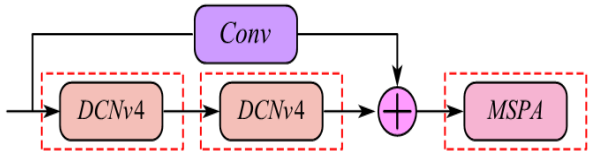


Fig. 13. Structure of D_M_BottleNeck.

and the MSPA attention mechanism, the network achieves a significant reduction in parameters while effectively preserving feature information. This design can minimize false positives and missed detection, enabling the network to adapt more effectively to diverse small-object detection scenarios.

2.6 SCAM-Bidirectional Feature Pyramid Network

Multi-scale feature fusion plays a critical role in enhancing the performance and robustness of object detection networks. Although YOLOv8 introduces secondary fusion through the PAN-FPN structure to optimize feature integration, its bidirectional fusion design is still relatively simplistic, and repeated convolutions can lead to the degradation of semantic details for small objects.

Therefore, to address these limitations and improve the model's object detection capabilities, a novel bidirectional feature pyramid network (BiFPN) is proposed that incorporates a Spatial Context-Aware Module (SCAM), as illustrated in Fig. 15. The SCAM leverages global average pooling (GAP) and global max pooling (GMP) to guide the learning of pixel relationships between spatial and channel dimensions. This mechanism supports contextual feature interaction across both spatial and channel dimensions. Furthermore, the addition of a top-down pathway allows high-level semantic feature information to flow back into the network, guiding subsequent modules in feature fusion and generating more discriminative features. In the feature fusion phase, it is more effective to model the global relationship between small targets and their backgrounds than that in the backbone phase. Using the global context information to represent the relationship between pixels across spatial dimensions cannot only suppress the irrelevant background noise, but also enhance the distinction between the objects and the background. In order to further optimize the fusion, connections are added between the first and last nodes of the intermediate feature layers to ensure more comprehensive information integration.

Inspired by GCNet [40] and SCP [41], SCAM is designed with three branches. The first branch integrates global information through GAP and GMP, enabling the capture of rich contextual information. The second branch applies a 1×1 convolution to produce the linear transfor-

mation of the feature map, referred to as "*value*" in Fig. 15. And the third branch also employs a 1×1 convolution to simplify the transformations of "*queries*" and "*keys*," collectively referred to as "QK" in Fig. 15. The outputs from the first and third branches are multiplied with the second branch to generate two representations of cross-channel contextual information and spatial contextual information, which are then fused using the broadcast Hadamard product, producing the final output of SCAM.

3. Experiments

3.1 Datasets Introduction

To ensure the reliability and validity of the experimental data, three representative public datasets are selected as DOTAv1.0, VisDrone2019, and TT100K.

The details of these datasets are as follows:

(1) DOTAv1.0 Dataset [42]: This dataset contains 2,806 images, with 188,282 labeled objects distributed across 15 categories. In this experiment, 1,414 images were used for training, 458 for validation, and 937 for testing.

(2) VisDrone2019 Dataset [43]: Collected and released by the Machine Learning and Data Mining Lab at Tianjin University, this dataset includes 8,629 images. In this study, 6,471 images were used for training, 548 for validation, and 1,610 for testing.

(3) TT100K Dataset [44]: This dataset covers road traffic signs in a variety of complex environments and weather conditions. These signs are small in size, making them challenging to detect. The TT100K dataset contains 26,349 images across 221 traffic sign categories, with annotations available for 128 categories. The training and testing sets used in this experiment consisted of 6,107 and 3,073 images, respectively. To reduce the impact of sample imbalances, we analyzed the distribution of signs across categories and selected 45 categories with more than 100 samples. These traffic signs were categorized into three types: warning signs, prohibition signs, and mandatory signs.

In the experimental design, DOTAv1.0 is chosen as the primary datasets for detailed comparative and ablation experiments. To further evaluate the model's generalization and applicability, additional experiments are conducted on the VisDrone2019 and TT100K datasets.

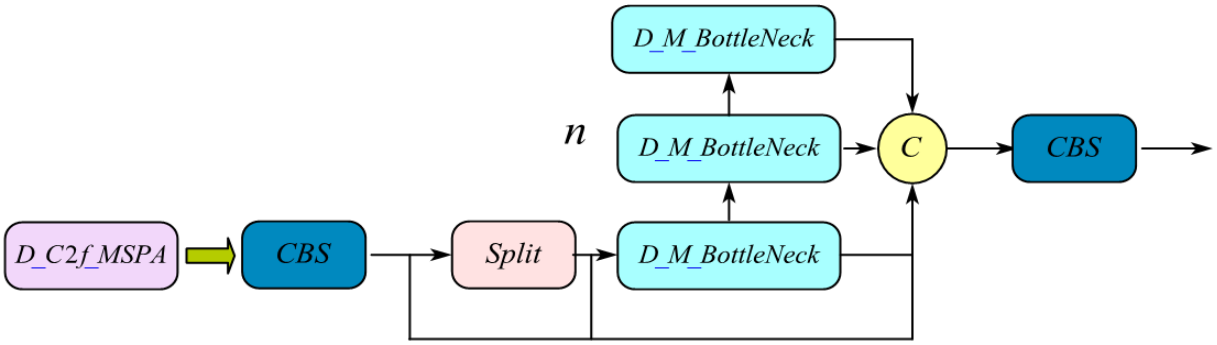


Fig. 14. Structure of D_C2f_MSPA.

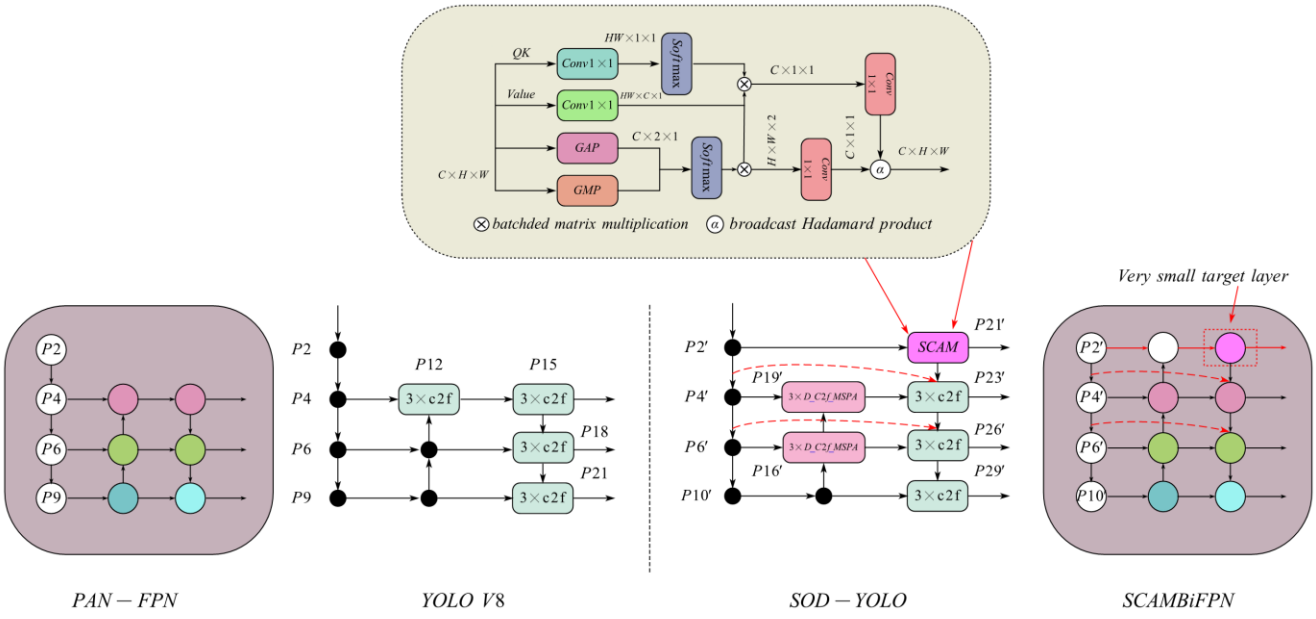


Fig. 15. SCAM- Bidirectional feature pyramid network diagram.

3.2 Experimental Environment

The experimental environment is based on the Ubuntu 20.04 operating system. The hardware configuration includes an Intel(R) Xeon(R) Platinum 8358P CPU @ 2.60 GHz, 32 GB of RAM, and an NVIDIA GeForce RTX 3090 GPU with 24 GB of memory. Among the commonly used deep learning frameworks, the PyTorch framework is chosen for its efficiency in training and testing datasets. Detailed experimental environment parameters are provided in Tab. 1.

For training, the input image size is set to 640 × 640 pixels, with a total of 300 epochs. A batch size of 16 is used, and the Stochastic Gradient Descent (SGD) optimizer is employed to accelerate the model convergence. The rest of the parameter settings are the same as YOLOv8 by default. Detailed training parameters are listed in Tab. 2.

The model evaluation metrics includes precision (*P*), recall (*R*), mean average precision (*mAP*) parameters, and frames per second (*FPS*).

4. Experimental Results

4.1 Overall Performance of SOD-YOLO

YOLOv8 and SOD-YOLO are first trained on the DOTAv1.0 datasets. Appendix A presents the results of both methods on the DOTAv1.0 test set. Compared to the original YOLOv8, SOD-YOLO achieves average improvements of 2.5% in *P*, 7.9% in *R*, and 7.7% in *mAP*, respectively, demonstrating superior performance. Notably, for the helicopter category, the *mAP* of the original YOLOv8 is 17.9%, while SOD-YOLO is 48.3%, an increase of 30.4%.

Component	Name/Value
Operating system	Ubuntu 20.04
CPU	Intel(R) Xeon(R) Platinum 8358P CPU @ 2.60GHz
GPU	NVIDIA GeForce RTX3090
Video memory	24GB
Training acceleration	CUDA 11.8
Programming language	Python 3.8
Deep learning framework for training	PyTorch 2.0.0

Tab. 1. Experimental environmental parameters.

Component	Name/Value
Input image size	640×640 pixels
Epoch	300
Training batch size	16
Initial learning rate	0.01
Final learning rate	0.1
Momentum	0.937
Weight_decay	0.0005
Optimizer	SGD

Tab. 2. Experimental parameters of network training.

To further validate the robustness of SOD-YOLO and its performance differences compared to the original YOLOv8, additional comparative experiments are conducted on the TT100K and VisDrone2019 datasets. As shown in Appendix B, SOD-YOLO also performs exceptionally on the TT100K datasets, achieving improvements of 5.8% in *P*, 4.8% in *R*, and 6.7% in *mAP* over the original YOLOv8. Specifically, for the p12 category, YOLOv8 achieves an *mAP* of 61.2%, while SOD-YOLO reaches 84.4%, making an improvement of 23.2%.

Finally, experiments conduct on the VisDrone2019 datasets. It is demonstrated in Appendix C that SOD-YOLO surpasses YOLOv8 with gains of 10.8%, 9.6%, and 11.9% in *P*, *R*, and *mAP*, respectively. These comparisons further show that the SOD-YOLO exhibits superior performance in small object detection and significantly outperforms the original YOLOv8 in both precision and accuracy.

4.2 Ablation Experiment

To validate the effectiveness of the improved SOD-YOLO algorithm for small object detection, the YOLOv8 baseline network is defined as A. Subsequently, the modules of S_C2f_CAFM, SPPF_E, D_C2f_MSPA, SCAM-BiFPN, and NWD are incrementally added, resulting in networks labeled as B, C, D, E, and F, respectively. Ablation experiments are conducted on the DOTAv1.0 datasets to evaluate the performance contribution of each module, with results presented in Tab. 3.

It is seen that embedding the S_C2f_CAFM module into the backbone increases the *mAP* from 61.3% to 63.9%. Additionally, replacing multiple reused C2f modules with a single S_C2f_CAFM module can reduce the parameter count by 3.58%. The introduction of the SPPF_E structure slightly reduces the parameter count, surprisingly, while it is further decreased by 2.73% through adding D_C2f_MSPA due to the efficiency of DCNv4 convolution, with the *mAP* increasing by 3.41%. Replacing the original PAN-FPN with the proposed SCAM-BiFPN raises the *mAP* from 66.7% to 68.2%, while reducing the number of feature maps generated by the C2f module further lowers the parameter count. Finally, incorporating the proposed NWD improves the *mAP* to 69.3%.

Overall, despite a 31.1% increase in network depth leading to a slight decrease in FPS, SOD-YOLO improves the *mAP* by 7.7% and reduces the parameter count by 13.9%. These results demonstrate that the SOD-YOLO significantly enhances its performance compared to the original YOLOv8. Furthermore, its lightweight design makes it highly suitable for deployment on the devices with limited hardware resources.

Model	*P*/%	*R*/%	*mAP@0.5*/%	*Layers*	*Param*/×10^5	*FPS*
YOLOv8(A)	74.3	59	61.6	225	32.69	93.9
A+B	74.9	62.1	63.9	240	31.52	82.4
A+B+C	75.4	62.8	64.5	245	31.09	73.6
A+B+C+D	75.9	64.1	66.7	263	30.24	60.7
A+B+C+D+E	76.2	65.3	68.2	295	28.16	59.3
A+B+C+D+E+F	**76.8**	**66.9**	**69.3**	**295**	**28.16**	**59.3**

Tab. 3. Results of ablation experiment on DOTA-V1.0 datasets.

Model	*P*/%	*R*/%	*mAP@0.5*/%	*Param*/×10^5	*FPS*
SSD	82.8	23.8	38.1	100.2	30
Faster R-CNN	47.6	56	49	42.5	31.7
FR-O [45]	-	-	54.1	-	-
YOLOv5	76.6	53.3	56.9	20	80.1
YOLOv7	66.1	59.5	59.8	35.4	83.5
YOLOv11n	67.1	56.3	57	25.85	158.6
YOLOv8	74.3	59	61.6	32.7	93.9
YOLOv10m	73.4	59.8	61.9	164.7	101.2
DCN-YOLO [46]	-	-	63.4	-	-
ICN [47]	-	-	68.2	-	-
YOLOv9	74.9	66.3	69.1	605.3	50.8
SOD-YOLO	**76.8**	**66.9**	**69.3**	**28.16**	**59.3**

Tab. 4. Comparison of different models in experiments.

The precision-recall (P-R) curve provides an intuitive representation of model performance, as shown in Fig. 16. It is seen from the figure that the P-R curve area for SOD-YOLO is significantly larger than that of YOLOv8. This indicates that SOD-YOLO consistently achieves higher precision across various recall rates. Therefore, SOD-YOLO exhibits superior small object detection capabilities compared to YOLOv8.

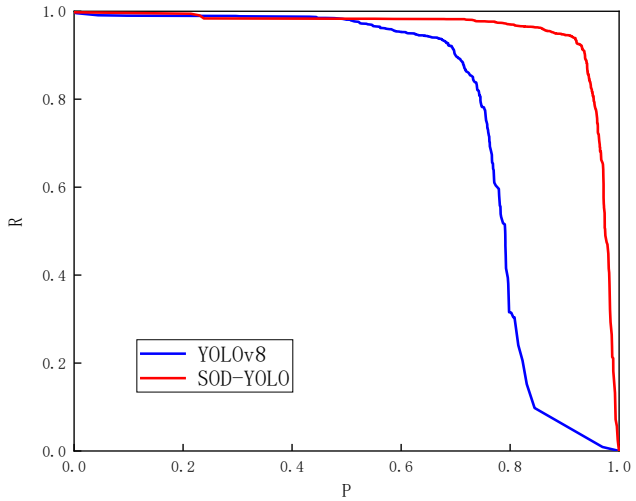


Fig. 16. Precision–recall curves on DOTA-V1.0 datasets.

4.3 Comparison Experiment

4.3.1 Comparison of Loss Functions

To evaluate the performance of NWD, it is compared against CIOU (used in YOLOv8) and other commonly employed IOU methods, including Distance Intersection over Union (DIOU) and Generalized Intersection over Union (GIOU). Using YOLOv8 as the base model, the quantitative comparison results on the DOTAv1.0 datasets are presented in Fig. 17.

As shown in Fig. 17, the loss values for all functions decrease and eventually converge as the number of epochs increases. However, the NWD demonstrates faster convergence and achieves lower loss values compared to the other functions among them. Consequently, the proposed improved network with NWD as the bounding box loss function can significantly enhance the small object detection performance.

4.3.2 Comparison with Other Detection Models

To validate the superiority of the SOD-YOLO algorithm, we conducted comparative experiments on the DOTAv1.0 dataset using identical experimental setups and training parameters, comparing SOD-YOLO with other leading small object detection models.

As shown in Tab. 4, although the *FPS* of SOD-YOLO is slightly lower than that of YOLOv8, YOLOv7, and YOLOv5, it still outperforms other algorithms in terms of detection speed. The parameter count of SOD-YOLO is 28.16M, which is significantly lower than that of other models with high *mAP*. In contrast, YOLOv9, which achieves a similar *mAP* to SOD-YOLO, has a parameter count of 605M. This demonstrates that SOD-YOLO achieves high detection accuracy with lower computational costs.

Furthermore, SOD-YOLO exhibits notable advantages in the mAP@0.5 metric. Specifically, SOD-YOLO improves mAP@0.5 by 31.2% compared to the least effective model, SSD, and surpasses YOLOv10 and YOLOv11 by 7.4% and 12.3%, respectively. These results confirm that the improved algorithm enhances the feature extraction capability for small objects of varying sizes and in complex backgrounds, leading to significantly improved detection accuracy while reducing false positives and missed detections to some extent.

4.4 Visualization Analysis

The confusion matrices for the YOLOv8 and SOD-YOLO models on the DOTAv1.0 dataset are shown in Fig. 18. Compared to YOLOv8, the SOD-YOLO model demonstrates a notable improvement in classification accuracy. Specifically, the "bridge" category saw the largest accuracy increase, by 28%, while the "ground track field" and "storage tank" categories improved by 25% and 10%, respectively. In the SOD-YOLO confusion matrix, the "plane" category achieved the highest classification accuracy, reaching 91%, while the "roundabout" category had the lowest accuracy, at 37%. This suggests that the model prioritizes different categories to varying degrees.

In YOLOv8, the confusion rate between the "basketball court" and "tennis court" categories is 0.06, indicating that YOLOv8 has a 6% chance of misidentifying a "basketball court" as a "tennis court." Similarly, the confusion rate between "basketball court" and "soccer ball field" is 0.04, meaning there is a 4% chance of misclassifying a "basketball court" as a "soccer ball field." In contrast, the SOD-YOLO model significantly reduces these issues, with confusion rates decreasing by 0.04 and 0.03, respectively. This

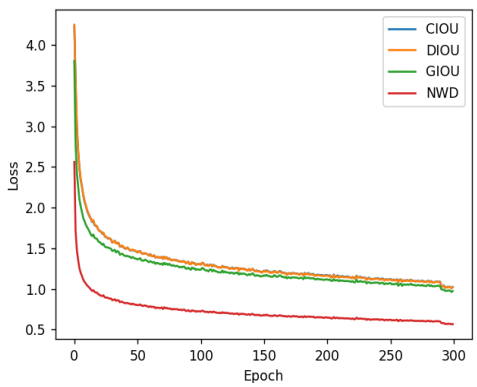


Fig. 17. Comparison of different loss functions on DOTA-V1.0 datasets.

demonstrates that the optimized model not only improves classification accuracy but also substantially mitigates the misdetection of small objects.

Grad-CAM [48] is a visualization technique used to identify the regions of feature maps in deep neural networks that contribute the most to prediction outcomes. By localizing specific image regions, it enhances the interpretability and visual comprehensibility of the prediction process for YOLOv8 (a) and SOD-YOLO (b). As shown in Fig. 19, the Grad-CAM visualization results reveal that the improved model (SOD-YOLO) focuses more accurately on target regions compared to the original YOLOv8, which often emphasizes background regions. This demonstrates that the improved model more effectively captures target features in images, and thus enhances the accuracy and overall performance.

To visually illustrate detection performance, Figure 20 showcases the detection results of SOD-YOLO and YOLOv8 on the DOTAv1.0, TT100K, and VisDrone2019 datasets, respectively. The results show that YOLOv8 exhibits varying degrees of missed detection and false positives, while SOD-YOLO effectively mitigates these issues. Notably, in scenarios with significant overlap among multiple target objects, SOD-YOLO provides more accurate bounding box predictions. These findings further validate the proposed improvements, demonstrating their ability to enhance the original model's performance and thus, significantly increase the accuracy of small object detection.

5. Conclusions and Future Work

A novel small object detection algorithm, SOD-YOLO, is proposed based on YOLOv8s framework. First, the C2f module in the feature extraction and fusion networks is redesigned, which effectively reduce the model's parameter count and floating-point computations and thus enhance its performance in small object detection tasks. Second, a new bidirectional feature pyramid network is developed to generate more distinctive features. Finally, the NWD bounding box loss function is introduced to further improve detection accuracy.

Extensive tests are conducted on DOTAv1.0, TT100K, and VisDrone2019 datasets. It demonstrated that SOD-YOLO achieved *mAP* improvements of 7.7%, 6.7%, and 11.9% over original YOLOv8, while reducing the parameter count by 13.9%. Additionally, SOD-YOLO exhibited superior performance in *mAP*, parameter efficiency, and *FPS* compared to other classical detection networks.

Overall, the improved SOD-YOLO significantly addresses challenges in small object detection, such as small target sizes, insufficient feature extraction capabilities, and complex backgrounds, resulting in more accurate and robust detection outcomes.

Experimental results demonstrate that small objects are more susceptible to aggregation and occlusion. Consequently, enhancing the algorithm's detection performance

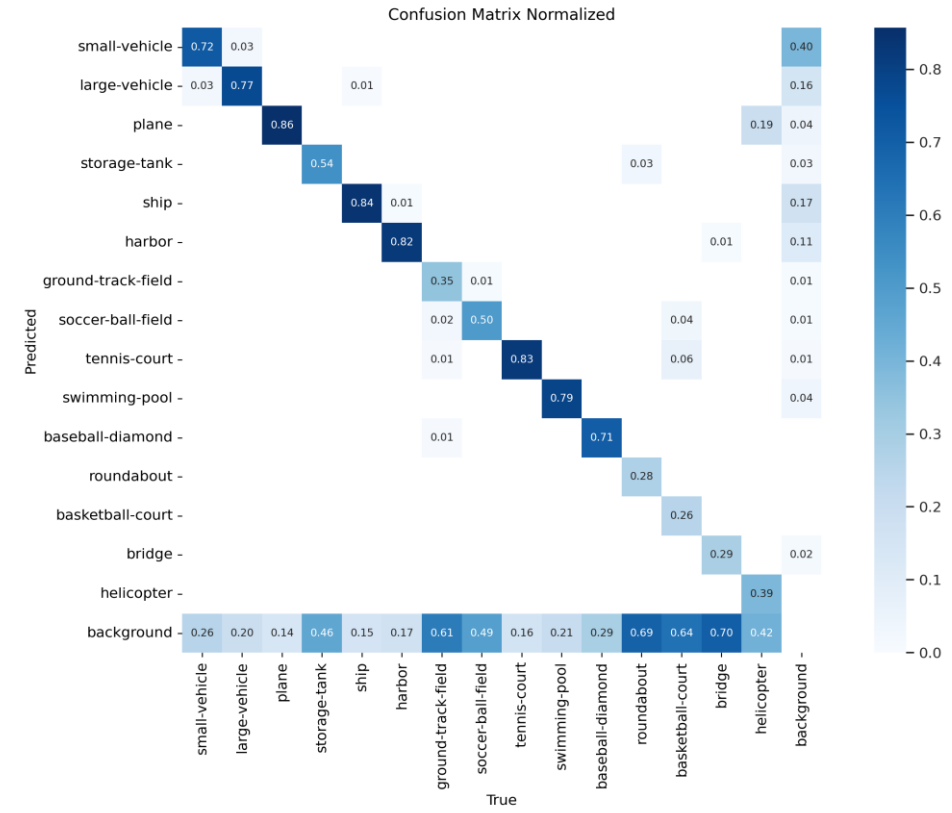


Fig. 18. (a) Confusion matrix diagram of YOLOv8.

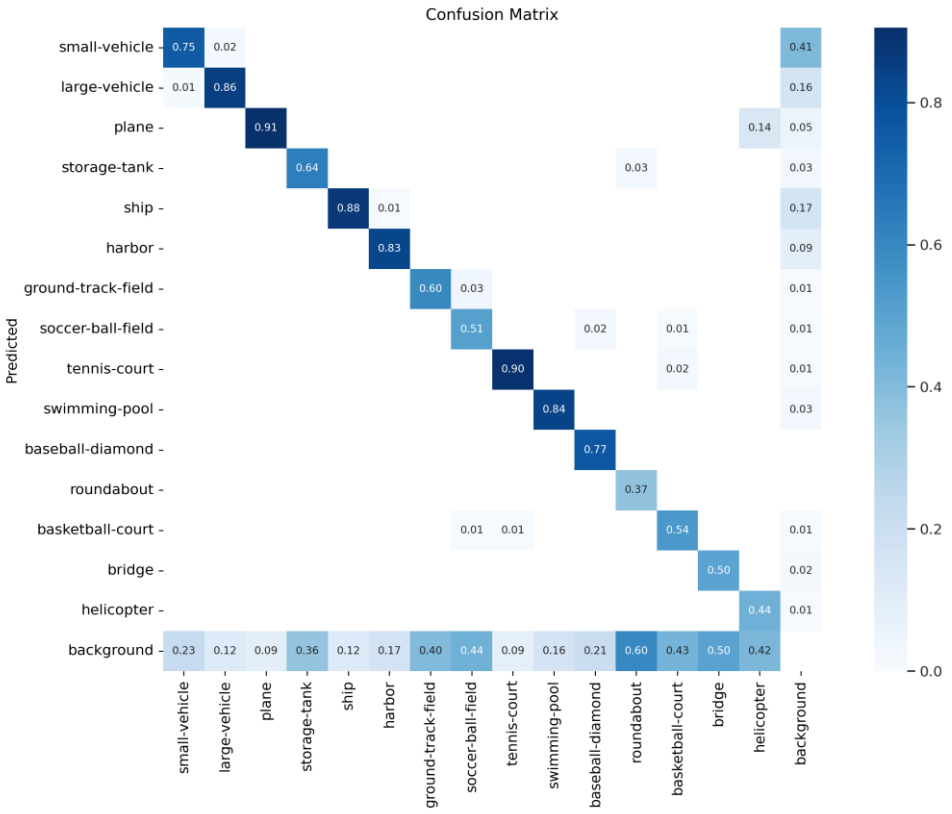
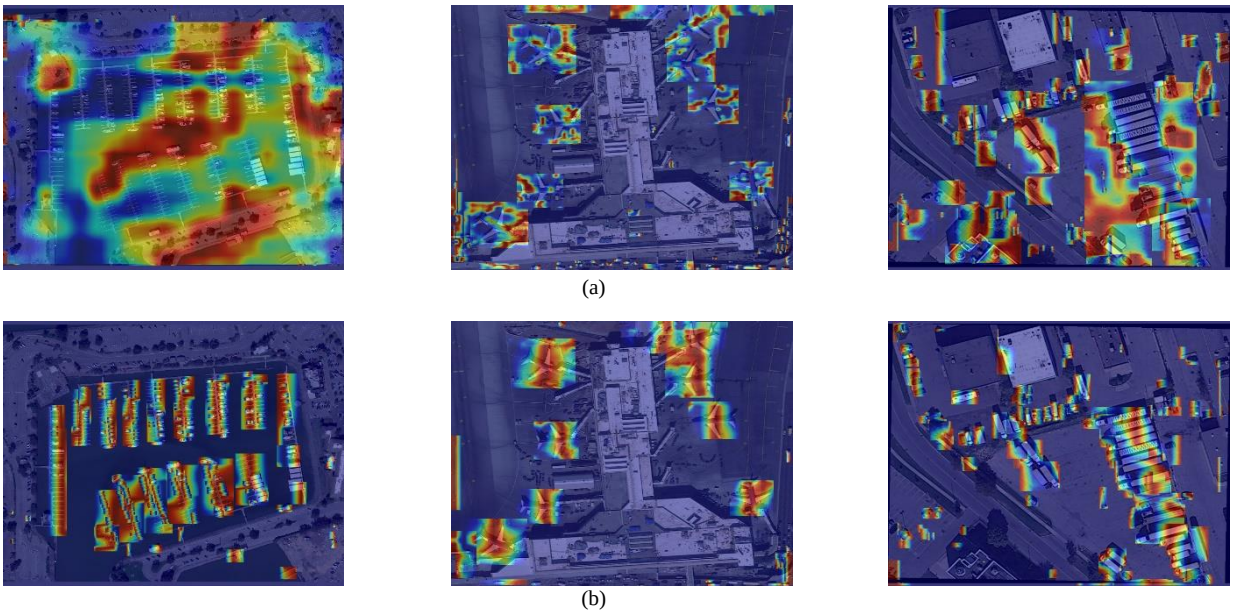
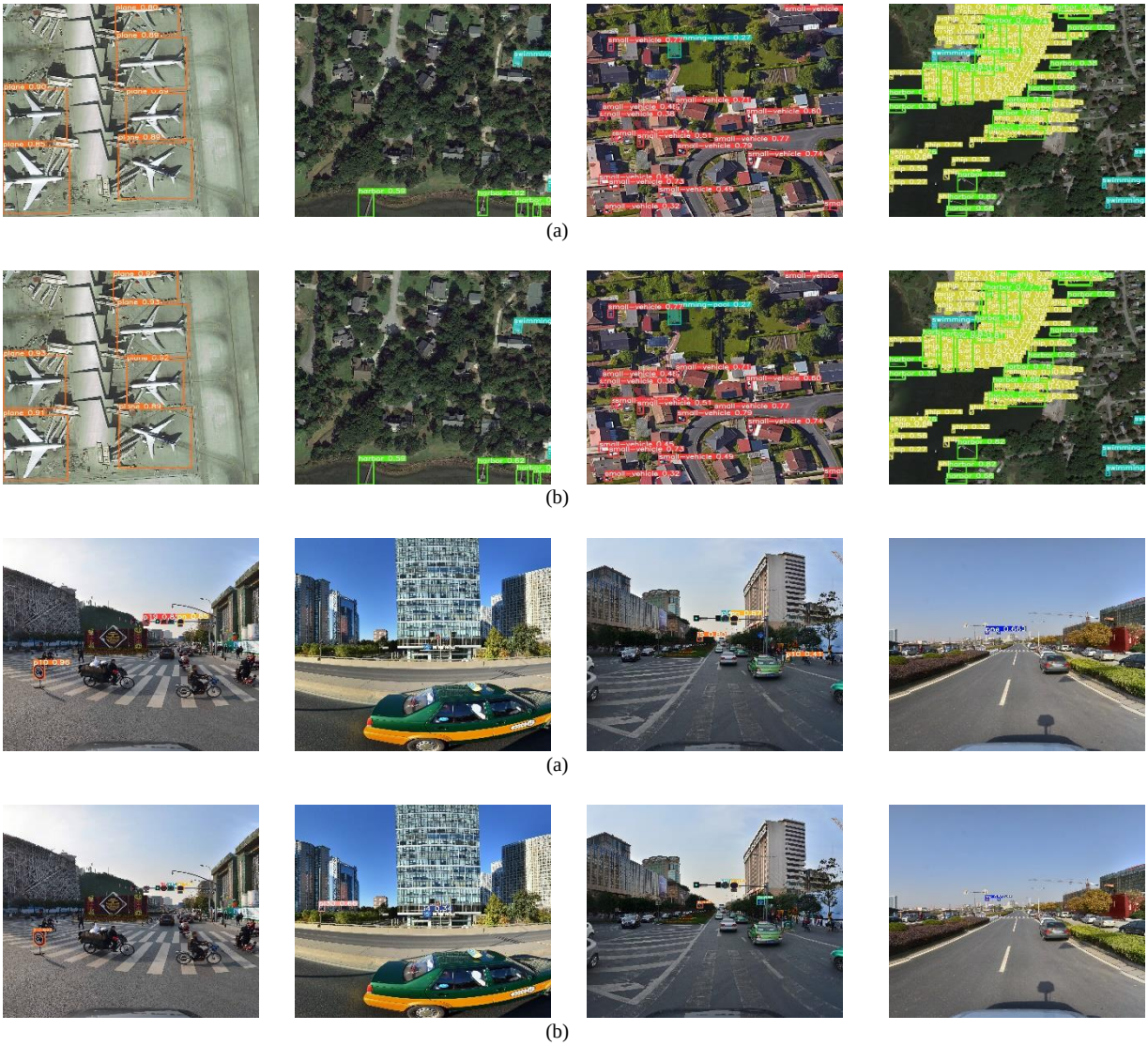


Fig. 18. (b) Confusion matrix diagram of SOD-YOLO.



(a) (b)

Fig. 19. Grad-CAM visualization results: (a) YOLOv8, (b) SOD-YOLO.



(a) (b) (a) (b)

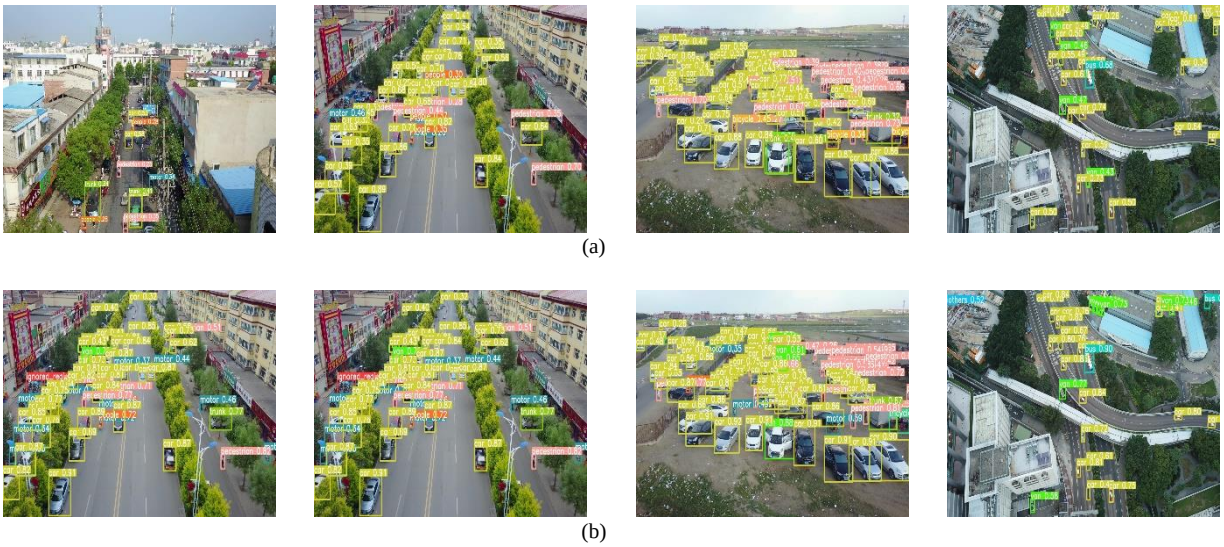


Fig. 20. Detection results: (a) YOLOv8, (b) SOD-YOLO.

for occluded objects is of critical practical significance and will be a key area of focus in future research. The proposed model has the potential to be applied to other detection tasks through approaches such as transfer learning, thereby improving its generalization capability. While the algorithm developed in this study enhances detection accuracy for small objects, it is associated with relatively high parameter and computational overhead. To address this, future work can explore optimization techniques, such as knowledge distillation and model pruning, to make the model more lightweight and computationally efficient.

Acknowledgments

The authors would like to acknowledge the support by key research and development program for the Natural Science Foundation of Liaoning Province (No. 2023JH2/101300218).

More information regarding the mathematical model and the source code used in this study is available upon request. Please contact the corresponding author for further details.

References

[1] LI, L., MU, X., LI, S., et al. A review of face recognition technology. *IEEE Access*, 2020, vol. 8, p. 139110–139120. DOI: 10.1109/ACCESS.2020.3011028

[2] ISLAM, S. M. M., BORIĆ-LUBECKE, O., ZHENG, Y., et al. Radar-based non-contact continuous identity authentication. *Remote Sensing*, 2020, vol. 12, no. 14, p. 1–22. DOI: 10.3390/rs12142279

[3] CORTES, C., VAPNIK, V. Support-vector networks. *Machine Learning*, 1995, vol. 20, no. 3, p. 273–297. DOI: 10.1007/BF00994018

[4] FREUND, Y., SCHAPIRE, R. E. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 1997, vol. 55, no. 1, p. 119–139. DOI: 10.1006/jcss.1997.1504

[5] GIRSHICK, R., DONAHUE, J., DARELL, T., et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus (USA), 2014, p. 580–587. DOI: 10.1109/CVPR.2014.81

[6] GIRSHICK, R. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*. Santiago (Chile), 2015, p. 1440–1448. DOI: 10.1109/ICCV.2015.169

[7] REN, S., HE, K., GIRSHICK, R., et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, vol. 39, no. 6, p. 1137–1149. DOI: 10.1109/TPAMI.2016.2577031

[8] REDMON, J., DIVVALA, S., GIRSHICK, R., et al. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas (USA), 2016, p. 779–788. DOI: 10.1109/CVPR.2016.91

[9] REDMON, J., FARHADI, A. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu (USA), 2017, p. 6517–6525. DOI: 10.1109/CVPR.2017.690

[10] REDMON, J., FARHADI, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018, p. 1–6. DOI: 10.48550/arXiv.1804.02767

[11] BOCHKOVSKIY, A., WANG, C. Y., LIAO, H. Y. M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. DOI: 10.48550/arXiv.2004.10934

[12] ZHU, X., LYU, S., WANG, X., et al. 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)*. Montreal (Canada), 2021, p. 2778–2788. DOI: 10.1109/ICCVW54120.2021.00312

[13] WANG, C. Y., BOCHKOVSKIY, A., LIAO, H. Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv e-prints*, 2022, p. 1–15. DOI: 10.48550/arXiv.2207.02696

[14] TERVEN, J., CORDOVA-ESPARZA, D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv preprint arXiv:2304.00501*, 2023, p. 1–27. DOI: 10.48550/arXiv.2304.00501

[15] LIU, W., ANGUELOV, D., ERHAN, D., et al. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*. Amsterdam (Netherlands), 2016, p. 21–37. DOI: 10.1007/978-3-319-46448-0_2

[16] WANG, C. Y., YEH, I. H., LIAO, H. Y. YOLOv9: Learning what you want to learn using programmable gradient information. In *European Conference on Computer Vision*. Cham (Switzerland), 2025, p. 1–21. DOI: 10.1007/978-3-031-72751-1_1

[17] WANG, A., CHEN, H., LIU, L., et al. YOLOv10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024, p. 1–21. DOI: 10.48550/arXiv.2405.14458

[18] GONG, Y., YU, X., DING, Y., et al. Effective fusion factor in FPN for tiny object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. Waikoloa (HI, USA), 2021, p. 1159–1167. DOI: 10.1109/WACV48630.2021.00120

[19] BAI, Y., ZHANG, Y., DING, M., et al. SOD-MTGAN: Small object detection via multi-task generative adversarial network. In *Proceedings of the European Conference on Computer Vision* (ECCV). Munich (Germany), 2018, p. 206–221. DOI: 10.1007/978-3-030-01261-8_13

[20] HONG, M., LI, S., YANG, Y., et al. SSPNet: Scale selection pyramid network for tiny person detection from UAV images. *IEEE Geoscience and Remote Sensing Letters*, 2021, vol. 19, p. 1 to 5. DOI: 10.1109/LGRS.2021.3103069

[21] WANG, Y., ZOU, X., SHI, J., et al. YOLOv5-based dense small target detection algorithm for aerial images using DIOU-NMS. *Radioengineering*, 2024, vol. 33, no. 1, p. 12–23. DOI: 10.13164/re.2024.0012

[22] CHEN, D., XIONG, S., GUO, L. Research on detection method for tunnel lining defects based on DCAM-YOLOv5 in GPR B-scan. *Radioengineering*, 2023, vol. 32, no. 3, p. 299–311. DOI: 10.13164/re.2023.0299

[23] YAO, G., ZHU, S., ZHANG, L., et al. HP-YOLOv8: High-precision small object detection algorithm for remote sensing images. *Sensors*, 2024, vol. 24, no. 15, p. 1–23. DOI: 10.3390/s24154858

[24] GE, Z., LIU, S., WANG, F., et al. YOLOX: Exceeding YOLO series in 2021. *arXiv preprint arXiv:2107.08430*, 2021, p. 1–7. DOI: 10.48550/arXiv.2107.08430

[25] YIN, X., LI, W., WANG, L., et al. Sea surface small target detection on one-dimensional sequential signals. *Radioengineering*, 2024, vol. 33, no. 3, p. 463–476. DOI: 10.13164/re.2024.0463

[26] GAO, P., LU, J., LI, H., et al. Container: Context aggregation network. *arXiv preprint arXiv:2106.01401*, 2021, p. 1–12. DOI: 10.48550/arXiv.2106.01401

[27] LIU, W., LU, H., FU, H., et al. Learning to upsample by learning to sample. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Paris (France), 2023, p. 6004 to 6014. DOI: 10.1109/ICCV51070.2023.00554

[28] WANG, J., XU, C., YANG, W., et al. A normalized Gaussian Wasserstein distance for tiny object detection. *arXiv preprint arXiv:2110.13389*, 2021, p. 1–12. DOI: 10.48550/arXiv.2110.13389

[29] ZHENG, Z., WANG, P., REN, D., et al. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybernetics*, 2022, vol. 52, no. 8, p. 8574–8586. DOI: 10.1109/TCYB.2021.3095305

[30] WANG, K., LIEW, J. H., ZOU, Y., et al. PANet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Seoul (South Korea), 2019, p. 9196–9205. DOI: 10.1109/ICCV.2019.00929

[31] LIN, T. Y., DOLLÁR, P., GIRSHICK, R., et al. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. *Honolulu* (HI, USA), 2017, p. 2117–2125. DOI: 10.1109/CVPR.2017.106

[32] LIU, S., QI, L., QIN, H., et al. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Salt Lake City (UT, USA), 2018, p. 8759–8768. DOI: 10.1109/CVPR.2018.00913

[33] LI, X., WANG, W., WU, L., et al. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *arXiv preprint arXiv:2006.04388*, 2020, p. 1–14. DOI: 10.48550/arXiv.2006.04388

[34] SUNKARA, R., LUO, T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Grenoble (France), 2022, part III, p. 443–459. DOI: 10.1007/978-3-031-26409-2_27

[35] HU, S., GAO, F., ZHOU, X., et al. Hybrid convolutional and attention network for hyperspectral image denoising. *IEEE Geoscience and Remote Sensing Letters*, 2024, vol. 21, p. 1–5. DOI: 10.1109/LGRS.2024.3370299

[36] XIONG, Y., LI, Z., CHEN, Y., et al. Efficient deformable convnets: Rethinking dynamic and sparse operator for vision applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* Seattle (USA), 2024, p. 5652–5661. DOI: 10.1109/CVPR52733.2024.00540

[37] YU, Y., ZHANG, Y., CHENG, Z., et al. Multi-scale spatial pyramid attention mechanism for image recognition: An effective approach. *Engineering Applications of Artificial Intelligence*, 2024, vol. 133, p. 1–15. DOI: 10.1016/j.engappai.2024.108261

[38] ZHANG, Y., YE, M., ZHU, G., et al. FFCA-YOLO for small object detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 2024, vol. 62, p. 1–15. DOI: 10.1109/TGRS.2024.3363057

[39] ZHANG, X., ZENG, H., GUO, S., et al. Efficient long-range attention network for image super-resolution. In *European Conference on Computer Vision*. Tel Aviv (Israel), 2022, part XVII, p. 649–667. DOI: 10.1007/978-3-031-19790-1_39

[40] CAO, Y., XU, J., LIN, S., et al. GCNet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.* Seoul (South Korea), 2019, p. 1971–1980. DOI: 10.1109/ICCVW.2019.00246

[41] LIU, Y., LI, H., HU, C., et al. LUO, S., LUO, Y., & WEN CHEN, C. Learning to aggregate multi-scale context for instance segmentation in remote sensing images. *IEEE Transactions on Neural Networks and Learning Systems*, 2025, vol. 36, no. 1, p. 595–609. DOI: 10.1109/TNNLS.2023.3336563

[42] DOTA Dataset. *DOTA: A Large-scale Dataset for Object Detection in Aerial Images*. [Online] Cited 2024-11-29. Available at: https://captain-whu.github.io/DOTA/dataset.html

[43] VisDrone2019 Dataset. *Visdrone-vid2019: The Vision Meets Drone Object Detection in Video Challenge Results*. [Online] Cited 2024-11-29. Available at: https://github.com/VisDrone/VisDrone-Dataset?tab=readme-ov-file

[44] TT100K Dataset. *Traffic-sign Detection and Classification in the Wild.* [Online] Cited 2024-11-29. Available at: https://cg.cs.tsinghua.edu.cn/traffic-sign/

[45] XIA, G. S., BAI, X., DING, J., et al. DOTA: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* Salt

Lake City (UT, USA), 2018, p. 3974–3983. DOI: 10.1109/CVPR.2018.00418

[46] TIAN, B., CHEN, H. Remote sensing image target detection method based on refined feature extraction. *Applied Sciences*, 2023, vol. 13, no. 15, p. 1–13. DOI: 10.3390/app13158694

[47] AZIMI, S. M., VIG, E., BAHMANYAR, R., et al. Towards multi-class object detection in unconstrained remote sensing imagery. In *14th Asian Conference on Computer Vision.* Perth (Australia), 2018, part III, p. 150–165. DOI: 10.1007/978-3-030-20893-6_10

[48] SELVARAJU, R. R., COGSWELL, M., DAS, A., et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision.* Venice (Italy), 2017, p. 618–626. DOI: 10.1109/ICCV.2017.74

About the Authors ...

Guoqiang MA was born in 1998. He is currently pursuing a master's degree in Mechanical Engineering and Automation at Liaoning University of Science and Technology. His research interests include pattern recognition, image processing, and computer vision.

Chuntian XU was born in 1972. He received his Ph.D. degree from Harbin Institute of Technology in 2016. Since 2020, he has been teaching in the School of Mechanical Engineering and Automation, Liaoning University of Science and Technology, and serving as a tutor for master's students. His main courses include CNC technology, microcontroller application basics, etc. His research interests include intelligent equipment manufacturing and digital technology. His research interests include intelligent equipment manufacturing and digital technology, specifically: synchronous motion control of spacecraft docking system; machine vision recognition; transport and palletising design and control.

Zong XU was born in 1981. Engineer, research direction: intelligent control and digital technology of equipment.

Xiangyang SONG, born in 1999, is currently studying for a master's degree in Mechanical Engineering and Automation at Liaoning University of Science and Technology.

Appendix A: Experimental Results on the DOTAv1.0 Datasets

Model	class	SV	LV	PL	ST	SH	HA	GTF	SBF	TC	SP	BD	RA	BC	BR	HC	Avg
YOLOv8	P/%	61.3	79.9	90.3	91.7	90.9	80.9	62.2	70.2	92.4	61.6	89.2	73	62.5	72.9	35.2	74.3
YOLOv8	R/%	68.1	81.6	85.1	53	83.2	77.3	41.7	45.3	87.8	74.7	68.7	36.9	31.2	33.1	16.6	59
YOLOv8	mAP/%	66.5	85.2	88.6	62.2	87.8	78	38.2	50.9	91.4	66.6	75.7	38.1	36.8	40.5	**17.9**	61.6
SOD-YOLO	P/%	63.1	83.3	91.3	93.3	92	80	65.9	69.2	94.8	58.6	85.9	79.5	71.6	66.6	57.4	**76.8**
SOD-YOLO	R/%	73	84.1	89.5	61.1	86.9	79.6	57.1	53.8	88.9	79.1	75.8	32.3	52.3	45.4	44.4	**66.9**
SOD-YOLO	mAP/%	69.2	86.9	92.6	73.2	90.4	82.6	62.5	55.4	93.1	65.6	76.4	42.9	52	48	**48.3**	**69.3**

Performance of YOLOv8 and SOD-YOLO on DOTAv1.0: small vehicle (SV), large vehicle (LV), plane (PL), storage tank (ST), ship (SH), harbor (HA), ground track field (GTF), soccer ball field (SBF), tennis court (TC),swimming pool (SP), baseball diamond (BD), roundabout (RA), basketball court (BC), bridge (BR), and helicopter (HC).

Appendix B: Experimental Results on the TT100K Datasets

Model	class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
YOLOv8	P/%	74.2	68.9	87.7	83.7	69	86.7	86.3	76.7	85.5	95.1	80.8	77	83.1	90	96.8	73.4
YOLOv8	R/%	79.2	57.9	78.5	81.2	69.8	80.3	77.3	79.4	76.2	72.4	84.1	66.7	76.8	70	77.1	74.3
YOLOv8	mAP/%	81.2	60.3	87.6	85.2	72.4	89.8	85.2	84.6	84.8	84.6	90.6	72.4	86.4	77.7	90.5	77.2
SOD-YOLO	P/%	88.7	79	98	96.3	75.3	95.7	92.1	80.4	97.8	93.7	88.8	79.8	95.5	100	98.3	85.2
SOD-YOLO	R/%	81.9	56.2	89.8	94.1	76	81.4	77.8	84.9	82.4	100	75.3	86.7	78.8	72.2	79.6	68.4
SOD-YOLO	mAP/%	88.8	67.1	91.8	96.5	77.8	96.8	89.2	88.7	88.6	99.5	85.4	87.3	89.4	94.3	93.9	75.5
Model	**class**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
YOLOv8	P/%	71.5	98.1	81	52.4	88.2	78.1	61.3	92.8	93.1	83.3	85.3	85.3	61.1	65.4	77.8	88.4
YOLOv8	R/%	52.8	76.7	68.2	48.9	80.7	62.4	54.9	78.3	62.8	78	72	81.1	42.4	66.7	65.9	78.4

YOLOv8	mAP/%	**61.2**	92	79.4	53.5	90.9	72.5	59.5	89.5	77.2	86.6	80.5	91.7	50.2	67	76.5	86.7
SOD-YOLO	P/%	80.9	85.7	85.6	68.9	93	72.9	92.7	94.9	77.7	86.3	95.1	94.4	79.5	62	88.1	92.7
SOD-YOLO	R/%	77.8	92.3	71.3	75	84.1	71.6	65.2	78.5	70.6	80.6	78	83.8	50	60	73.3	90.2
SOD-YOLO	mAP/%	**84.4**	90.6	84.3	69.4	95.1	78.9	76.9	92.7	83.1	92.1	88.7	94.6	60.3	75.7	82.9	94.4
Model	**class**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**Avg**		
YOLOv8	P/%	91.9	78	73.7	95.5	86.5	88.2	77	91.6	75.5	75	92.4	100	74.4	81.7		
YOLOv8	R/%	79.7	58.3	58.5	91.3	79.8	85.1	81.4	82.1	61.8	69.2	80.6	81.9	60.3	72		
YOLOv8	mAP/%	87.1	69.4	69.1	96.8	89.1	92.1	81.5	93.5	70.8	72.4	89.7	91	71.6	80		
SOD-YOLO	P/%	83.2	88.4	84.9	91.2	90.8	95.8	83.7	93.7	88.6	60.8	95	100	85.5	**87.5**		
SOD-YOLO	R/%	84.4	58.6	57.1	87.5	81.1	79.9	82.9	64.3	71.4	72.7	84.8	81	63.2	**76.8**		
SOD-YOLO	mAP/%	91	79.8	75	95.7	94.4	93.1	93	91.2	82.8	79.2	93.2	98.5	78.6	**86.7**		

Performance of YOLOv8 and SOD-YOLO on TT100K (1:pl80, 2:p6, 3:p5, 4:pm55, 5:pl60, 6:ip, 7:p11, 8:i2r, 9:p23, 10:pg, 11:il80, 12:ph4, 13:i4, 14:pl70, 15:pne:, 16:ph4.5, 17:p12, 18:p3, 19:pl5, 20:w13, 21:i4l, 22:pl30, 23:p10, 24:pn, 25:w55, 26:p26, 27:p13, 28:pr40, 29:pl20, 30:pm30, 31:pl40, 32:i2, 33:pl120, 34:w32, 35:ph5, 36:il60, 37:w57, 38:pl100, 39:w59, 40:il100, 41:p19, 42:pm20, 43:i5, 44:p27, 45:pl50).

Appendix C: Experimental Results on the VisDrone2019 Datasets

Model	**class**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**Avg**
YOLOv8	P/%	47.7	38.7	29.4	65.5	41	41.6	23.3	24.6	49.7	41.9	40.3
YOLOv8	R/%	29.9	16.9	11.6	59	28.6	29.7	27.7	23.7	41.8	27.8	29.7
YOLOv8	mAP/%	30.9	17.5	10.4	61.3	27.8	27.6	17	15.5	40	25	27.3
SOD-YOLO	P/%	58.4	49	43	74.5	48.9	58.2	31.8	27.8	67.9	51.9	**51.1**
SOD-YOLO	R/%	39.1	22	19.5	67	41.4	40.9	36.4	36.9	53.1	36.6	**39.3**
SOD-YOLO	mAP/%	42.7	23.4	21.8	71.5	41.8	44.7	26.3	24.9	57.7	37.6	**39.2**

Performance of YOLOv8 and SOD-YOLO onVisDrone2019 (C1: pedestrian, C2: people, C3: bicycle, C4: car, C5: van, C6: trunk, C7: tricycle, C8: awning-tricycle, C9: bus, C10: motor).