remote sensing

MDPI

Article

SPDC-YOLO: An Efficient Small Target Detection Network Based on Improved YOLOv8 for Drone Aerial Image

Jingxin Bi [1,2,3], Keda Li [1,2,3], Xiangyue Zheng [1,2,3], Gang Zhang [1,2,3] and Tao Lei [1,2,3,*]

[1] National Laboratory on Adaptive Optics, Chengdu 610209, China; bijingxin23@mails.ucas.ac.cn (J.B.); likeda23@mails.ucas.ac.cn (K.L.); zhengxiangyue23@mails.ucas.ac.cn (X.Z.); zhanggang@ioe.ac.cn (G.Z.)
[2] University of Chinese Academy of Sciences, Beijing 101408, China
[3] Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610209, China
* Correspondence: taoleiyan@ioe.ac.cn

Abstract: Target detection in UAV images is of great significance in fields such as traffic safety, emergency rescue, and environmental monitoring. However, images captured by UAVs usually have multi-scale features, complex backgrounds, uneven illumination, and low target resolution, which makes target detection in UAV images very challenging. To tackle these challenges, this paper introduces SPDC-YOLO, a novel model built upon YOLOv8. In the backbone, the model eliminates the last C2f module and the final downsampling module, thus avoiding the loss of small target features. In the neck, this paper proposes a novel feature pyramid, SPC-FPN, which employs the SBA (Selective Boundary Aggregation) module to fuse features from two distinct scales. In the head, the P5 detection head is eliminated, and a new detection head, Dyhead-DCNv4, is proposed, replacing DCNv2 in the original Dyhead with DCNv4 and utilizing three attention mechanisms for dynamic feature weighting. In addition, the model uses the CGB (Context Guided Block) module for downsampling, which can learn and fuse local features with surrounding contextual information, and the PPA (Parallelized Patch-Aware Attention) module replacing the original C2f module to further improve feature expression capability. Finally, SPDC-YOLO adopts EIoU as the loss function to optimize target localization accuracy. On the public dataset VisDrone2019, the experimental results show that SPDC-YOLO improves mAP_{50} by 3.4% compared to YOLOv8n while reducing the parameters count by 1.03 M. Compared with other related methods, SPDC-YOLO demonstrates better performance.

Keywords: UAV imagery; YOLOv8; SPC-FPN; DCNv4

check for updates

Academic Editor: Lei Ma

Received: 15 January 2025
Revised: 14 February 2025
Accepted: 16 February 2025
Published: 17 February 2025

Citation: Bi, J.; Li, K.; Zheng, X.; Zhang, G.; Lei, T. SPDC-YOLO: An Efficient Small Target Detection Network Based on Improved YOLOv8 for Drone Aerial Image. *Remote Sens.* **2025**, *17*, 685. https://doi.org/10.3390/rs17040685

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, the production cost of UAVs has gradually decreased, automation and control technologies have advanced, and the application of UAVs in fields such as agricultural monitoring [1], geological exploration [2], infrastructure inspection [3], and smart cities [4] has been steadily increasing. Small target detection technology plays a crucial role in these tasks, especially in real-time monitoring [5], target tracking [6], and data analysis by quickly and accurately identifying small targets. However, UAV images are typically captured from high altitudes, often feature complex backgrounds, significant illumination variations, and contain small targets, making small target detection [7] a major challenge.

The development of deep learning, especially convolutional neural networks (CNN [8]), has provided powerful tools for small target detection. Traditional methods for target detection struggle with small targets due to their reliance on hand-crafted features [9] and

insufficient adaptability to complex scenes. In contrast, CNN can effectively recognize targets in complex environments by automatically learning features, which improves both the accuracy and robustness of detection.

Despite these advances, small target detection still faces challenges such as small object size, limited image resolution, and complex backgrounds. To tackle these problems, various improvement methods have been proposed, such as optimizing CNN architectures, incorporating multi-scale detection, and applying data augmentation techniques to enhance [10] detection accuracy and speed. Multi-scale detection [11] enables the fusion of features from different scales, which effectively enhances the detection accuracy of small targets.

In the realm of object detection, YOLO (You Only Look Once [12]) stands as a highly efficient deep learning paradigm. This approach redefines object detection as a regression task, simultaneously predicting the spatial location and class of objects through a single forward pass, thereby dramatically enhancing detection speed. While YOLO has proven effective for large object detection, it has also achieved significant strides in the detection of small targets. Through the integration of multi-scale training and feature fusion techniques, YOLO has notably augmented the accuracy and robustness of small target detection, establishing itself as an indispensable tool in the area of small object detection.

In summary, with the advancements in UAV technology and deep learning, object detection, particularly small target detection, has become an indispensable component of UAV systems and has demonstrated immense potential in intelligent applications across various fields [13].

The main contributions of this paper are as follows:

1. In the backbone architecture, the final C2f module and downsampling module are removed, and the CGB [14] module is incorporated for downsampling while facilitating the fusion of contextual features. Simultaneously, the C2f module is replaced with the PPA [15] module to enhance the network's capability to capture discriminative features of small targets;
2. In the neck, a novel feature pyramid network SPC-FPN is proposed, which leverages the SBA [16] module to fuse features from different scales. For small objects, shallow features capture rich boundary details, while deep features contain valuable semantic information. Therefore, effectively combining shallow and deep features is essential for improving detection performance;
3. In the head, a new detection head, Dyhead-DCNv4 [17,18], is proposed, which utilizes scale-aware attention, spatial-aware attention, and task-aware attention to dynamically weight and adaptively adjust the features. Additionally, the P5 detection head, originally used for detecting large objects, is removed to reduce the model's parameter count.

The structure of this paper is organized as follows: Section 2 reviews the recent advancements in object detection within the remote sensing domain. Section 3 outlines the proposed improved model for small target detection in UAV imagery, providing a detailed description of the model architecture and the operational principles of the associated modules. Section 4 outlines the experimental environment and parameter configurations and examines the results of several experiments conducted on the VisDrone2019 [19] dataset, such as ablation studies, comparative evaluations, and visualization experiments aimed at evaluating the effectiveness of the proposed method. Section 5 delves into how the proposed model addresses the challenges of object detection in UAV imagery. Finally, Section 6 concludes the paper, summarizing the findings and discussing potential directions for future research.

2. Related Work

In the field of computer vision, YOLO has solidified its leadership in object detection due to its groundbreaking algorithms for small object detection, rapid processing capabilities, ease of integration and deployment, as well as its exceptional balance between speed and accuracy, multitasking potential, and high versatility. Compared to other deep learning methods, YOLO achieves an unparalleled trade-off between speed and accuracy by employing an end-to-end detection pipeline and performing a comprehensive single-pass analysis of the image. This makes it particularly suitable for applications requiring real-time responsiveness, such as autonomous driving and real-time surveillance.

With continuous technological advancements, YOLO not only maintains its leading position in object detection but also expands its capabilities to other visual tasks, particularly small object detection, by integrating multiple tasks into a unified prediction framework. This high adaptability has made YOLO the preferred choice for numerous real-world applications. Its technical innovations and strong performance provide a solid foundation for advancing real-time detection technologies and addressing various computer vision challenges. As a result, YOLO networks have become the primary choice for professionals seeking efficient and accurate detection, especially in small object detection tasks within the field of remote sensing.

2.1. The Traditional Object Detection Methods

Object detection has advanced quickly in computer vision since the early 2000s. In 2014, Ross Girshick et al. introduced R-CNN [20], a pioneering two-stage detection framework. The fundamental approach of R-CNN involves generating candidate regions via Selective Search, followed by feature extraction for each region using Convolutional Neural Networks, and concluding with classification and regression through a classifier. Although R-CNN achieves high accuracy, its processing speed is hindered by the need to individually extract features for each candidate region, resulting in slow performance.

Building upon R-CNN, a series of two-stage object detection algorithms have since been developed. In 2015, Ross Girshick presented Fast R-CNN [21], an enhanced version of R-CNN that improves efficiency by combining feature extraction and classification into one unified network. Fast R-CNN performs a single convolutional pass over the entire image and extracts features for candidate regions from shared feature maps, which significantly boosts detection speed. Additionally, it introduces bounding box regression, enabling the direct prediction of bounding boxes within the network, thus eliminating the reliance on Selective Search.

In 2015, Shaoqing Ren et al. introduced Faster R-CNN [22], an improved version of Fast R-CNN. It incorporates the Region Proposal Network (RPN), replacing Selective Search, and integrates region proposal generation with feature extraction. This allows simultaneous proposal and feature extraction, significantly boosting both detection speed and accuracy. As the first two-stage detector to achieve real-time performance, Faster R-CNN marked a major step forward in the efficiency and effectiveness of object detection.

Mask R-CNN [23], a further development of Faster R-CNN, was introduced by Kaiming He et al. in 2017. Building upon the Faster R-CNN framework, it incorporates a parallel segmentation branch designed to predict instance segmentation masks. In addition to object detection and bounding box regression, Mask R-CNN produces pixel-wise masks for every detected object, offering significant advantages for applications that require precise object delineation, such as medical image analysis and remote sensing, where accurate contouring of targets is essential.

Cascade R-CNN [24], introduced in 2018 by Zhaowei Cai and Nuno Vasconcelos, is a two-stage object detection framework that leverages a multi-stage detection architecture.

Each stage incorporates an independent detector, with progressively stricter thresholds for matching positive and negative samples as the process advances. This staged refinement allows the model to iteratively improve the accuracy of the predicted bounding boxes, significantly boosting overall detection performance. In the realm of remote sensing imagery, Cascade R-CNN's iterative approach proves particularly effective in enhancing detection precision, especially in challenging scenarios such as small object detection or in environments with complex backgrounds.

SSD (Single Shot MultiBox Detector [25]), introduced by Wei Liu et al. in 2016, is a well-known one-stage object detection algorithm. This model performs detection using a single deep neural network to analyze images. A key aspect of SSD is its discretization of the bounding box output space, where a set of default boxes with different aspect ratios and scales is assigned to each location on the feature map. During inference, the network assigns presence scores for each default box based on object class and generates adjustments to more accurately match the object's shape, thereby improving detection performance, especially in complex remote sensing environments.

DenseNet [26], introduced by Gao Huang et al. in 2017, is a convolutional neural network known for its dense connectivity, where each layer is linked to all previous layers. This design alleviates the vanishing gradient issue, enhances feature propagation, and encourages feature reuse. Additionally, it significantly reduces the number of parameters, making DenseNet a highly efficient model, especially for tasks like object detection in remote sensing, where both computational efficiency and robust feature extraction are critical.

2.2. The YOLO Series Object Detection Methods

The YOLO [12] algorithm, first introduced by Joseph Redmon in 2016, has undergone several significant iterations in the years since. As a single-stage object detection model, YOLO simplifies the traditionally complex detection pipeline into a single end-to-end regression task, leading to substantial improvements in detection speed—particularly advantageous for real-time applications. In 2017, Redmon presented YOLOv2, also referred to as YOLO9000 [27], incorporating key advancements such as anchor boxes, batch normalization, and multi-scale training, thus enhancing detection accuracy and improving small object detection capabilities. Building upon these innovations, YOLOv3 [28] was introduced in 2018, which further advanced small object detection performance by integrating a feature pyramid network (FPN) and adopting a deeper backbone architecture, Darknet-53, to bolster overall accuracy and robustness.

In 2020, Alexey Bochkovskiy introduced YOLOv4 [29], an improved version of YOLOv3, incorporating optimizations like CSPDarknet53, the Mish activation function, and the CIoU loss function. These upgrades led to significant gains in both accuracy and inference speed. That same year, Ultralytics released YOLOv5, a further refinement of YOLOv4, designed to optimize inference speed and precision, making it widely adopted in industrial applications. In 2021, the Megvii team launched YOLOX [30], an advanced variant that adopts an anchor-free approach and leverages state-of-the-art training strategies. YOLOX offers improved inference speed and greater flexibility and maintains high accuracy, which makes it especially effective for a broad spectrum of object detection tasks, including those in remote sensing applications.

In 2022, Meituan introduced YOLOv6 [31], specifically optimized for industrial applications, with significant enhancements in both inference speed and model size, making it highly suitable for large-scale deployment. That same year, Wang Chien-Yao et al. presented YOLOv7 [32], a further refinement of the YOLO series. This version improved detection accuracy by incorporating the "Bag-of-Freebies" and "Bag-of-Specials" strategies

while maintaining high inference speed, thereby setting new performance benchmarks for real-time object detection. In 2023, Ultralytics released YOLOv8, which brought a series of comprehensive advancements primarily focused on optimizing network architecture, training strategies, inference speed, and overall user-friendliness.

In 2024, Wang Chien-Yao et al. introduced YOLOv9 [33], which integrated the concept of Programmable Gradient Information (PGI) and presented a novel lightweight architecture called the Gradient-Path Planning-based General Efficient Layer Aggregation Network (GELAN). This innovation aimed to enhance both computational efficiency and detection accuracy. In the same year, researchers from Tsinghua University further optimized the model, proposing YOLOv10 [34], which incorporates a Non-Maximum Suppression (NMS)-free training strategy to effectively eliminate redundant detection boxes and improve detection precision. Moreover, in September 2024, Ultralytics released YOLOv11, marking yet another significant advancement in the YOLO series. These continuous iterations highlight the ongoing progress and innovation within the YOLO framework, reinforcing its dominance as a leading approach in the field of object detection.

2.3. Small Target Detection Methods Based on YOLO

Jie Luo et al. introduced the ESOD-YOLO [35] model, a small target detection framework based on YOLOv8n. It improves small object feature extraction by substituting the C2f module with the RepNIBMS module and uses a Waveform Feature Pyramid Network (WFPN) for efficient multi-scale feature fusion, enhancing the integration of spatial and semantic information.

Shijie Zhang et al. proposed the RTSOD-YOLO [36] model, addressing challenges such as occlusion and complex backgrounds in UAV-based detection. By combining an adaptive spatial attention mechanism with triple feature encoding (TFE), the model optimizes small object detection performance. Additionally, it incorporates an efficient redundant feature generation module, reducing model parameters and accelerating inference speed.

Bingqi Liu et al. introduced the RE-YOLO [37] model, specifically designed for remote sensing imagery. By employing a refined, efficient module (REM) and the RE_CSP block for multi-scale feature extraction, this approach achieves an optimal balance between computational complexity and feature extraction performance. Furthermore, the model enhances semantic information capture through a spatial extraction attention module (SEAM) and promotes the blending of shallow and deep features via a three-branch path aggregation network (TBPAN), thereby improving contextual information assimilation.

Chengcheng Wang proposed the Gold-YOLO [38] model, which tackles the issue of combining features from different scales within the YOLO framework by incorporating the Gather-and-Distribute (GD) mechanism. This mechanism integrates convolutional operations with self-attention, significantly improving the model's capability to efficiently integrate features from multiple scales.

Jianqiang Wang et al. developed the EAL-YOLO [39] algorithm, a lightweight and efficient attention-based model. It optimizes the backbone network using EfficientFormerV2 and integrates large separable convolution attention (LSKA) with spatial pyramid pooling (SPPF) to improve feature extraction. Additionally, the ASF2-Neck module boosts small object detection, and the lightweight shared convolution detection head (LSCHead) strikes a balance between accuracy and computational efficiency, enabling the model to perform well even on devices with limited resources.

In recent studies, UAV-YOLO [40] has been proposed to address the challenges of small object detection from an aerial perspective (Mingjie Liu et al., 2020). This method significantly improves detection performance by optimizing the network structure and training process.

Tushar Verma et al. proposed the SOAR [41] framework, which integrates a bidirectional State Space Model (SSM) with YOLOv9 and programmable gradient information, effectively improving small object detection in aerial imagery. Inspired by their work, we introduce an enhanced feature fusion strategy to further optimize detection performance.

Recent advancements in UAV-based traffic monitoring have leveraged deep learning models for vehicle detection and multi-object tracking. Robert Fonod et al. [42] developed a robust trajectory extraction pipeline that integrates object detection, track stabilization, and georeferencing techniques to achieve high-precision vehicle tracking from aerial imagery. Inspired by their work, we adopt a similar georeferencing approach to enhance trajectory accuracy.

Moreover, a variety of YOLO-based models, such as HIC-YOLOv5 [43], FFCA-YOLO [44], TPH-YOLOv5 [45], FE-YOLOv5 [46], and CA-YOLO [47], have been developed, each offering distinctive improvements. These advancements in YOLO-based models improve small target detection accuracy while maintaining the high efficiency of the YOLO framework, making them particularly effective for detecting small objects in challenging environments, such as UAV imagery.

3. Proposed Model

3.1. Overview of YOLOv8

Developed by Ultralytics, YOLOv8 is a classic version of the YOLO series that demonstrates significant performance enhancements over its predecessors.

The model's core features a carefully designed backbone for extracting image features. By decomposing standard convolutions into depthwise and pointwise types, computational complexity is reduced. Depthwise convolutions operate within each channel, while pointwise convolutions perform 1×1 operations across channels. The use of residual connections helps alleviate the gradient vanishing issue, enhancing information flow. This design accelerates inference and improves feature extraction, especially for complex remote-sensing imagery.

The "Neck" module is tasked with multi-scale feature fusion, comprising both Feature Pyramid Networks (FPN) and Path Aggregation Networks (PAN). The FPN integrates features across different levels through a top-down pathway, enhancing the model's detection capabilities by leveraging hierarchical feature representations. PAN further refines this fusion process by introducing a bottom-up pathway, facilitating more granular feature aggregation. This dual-path approach allows the model to effectively capture and interpret objects at varying scales, which is particularly advantageous in remote sensing tasks where targets can vary widely in size and spatial resolution within aerial imagery.

The detection head is tasked with generating the final detection outcomes, encompassing both object localization and classification. Adopting an anchor-free mechanism it eliminates the complexities inherent in anchor box design. Through the use of convolutional layers, the model directly predicts the center coordinates, dimensions (height and width), and class labels of the detected objects. This design streamlines the detection process, improving the model's efficiency and flexibility, particularly in remote sensing tasks where targets can exhibit significant variations in scale, as well as experience diverse spatial and geometric distortions in aerial imagery.

Moreover, the training strategy of YOLOv8 has been comprehensively optimized, incorporating stages of pretraining, fine-tuning, and hyperparameter tuning to ensure the model achieves peak performance across diverse datasets. In the pretraining phase, the model is initialized with a large-scale dataset to facilitate the learning of generalized features. During fine-tuning, the model is further adapted to the specific target dataset, enhancing its detection capabilities. Finally, hyperparameter tuning is performed to refine

the model's configuration, optimizing its overall performance. Figure 1 illustrates the network architecture of YOLOv8.

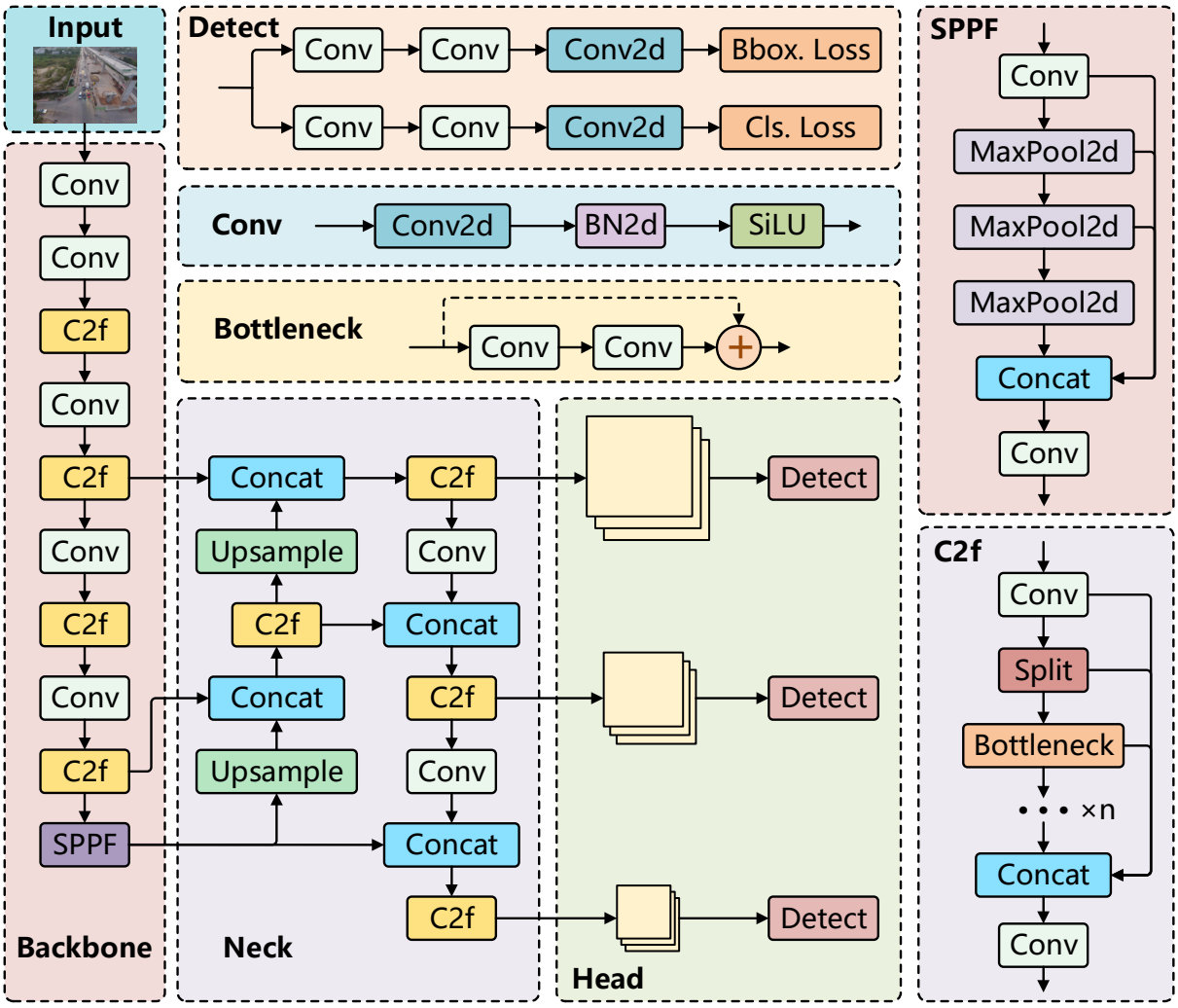


Figure 1. YOLOv8 structure diagram.

When employing YOLOv8 for small target detection, particularly in aerial drone images, several limitations and challenges emerge. As demonstrated in Figure 1, the YOLOv8 model incorporates three detection heads (P3, P4, and P5), each tailored for detecting small, medium, and large targets, respectively. The P5 head is obtained by downsampling the image by a factor of 32. This substantial downsampling leads to a significant loss of feature information for small targets, particularly those smaller than 32 pixels in size. Such a high degree of downsampling effectively reduces the target to a single point, making it incapable of capturing the intricate details of small objects.

Moreover, when the quantity of small objects within an image rises considerably, the model's detection performance tends to degrade, particularly in the context of drone imagery. The dense distribution and overlapping nature of small targets further impair detection accuracy. Additionally, YOLOv8 may lack the necessary flexibility to effectively handle targets with varying sizes and orientations, especially in drone images where target scales and angles can fluctuate widely. This variability presents an additional layer of complexity, thereby intensifying the challenges in achieving accurate detection.

Hence, for the detection of small objects, especially in scenarios where a large number of small targets are present, the inclusion of the P5 layer becomes superfluous. These challenges indicate that YOLOv8 may need further optimization, especially for small

object detection in remote sensing imagery, to improve both detection accuracy and computational efficiency.

3.2. The Proposed Method

To overcome the limitations of YOLOv8 in detecting small objects, this paper presents an innovative model, SPDC-YOLO. Figure 2 illustrates the network architecture.

First, SPDC-YOLO refines the original YOLOv8 backbone by removing the final convolutional module and the last C2f module. This modification mitigates the excessive 32× downsampling, effectively reducing the loss of small target features and retaining more fine-grained detail. Moreover, this adjustment lowers the model's parameter count, thus enhancing both the accuracy and efficiency of small target detection, which is crucial for remote sensing applications.

Additionally, in the Neck of SPDC-YOLO, upsampling is not performed. Instead, a novel feature pyramid, SPC-FPN, is introduced, incorporating a feature fusion module called SBA (Selective Boundary Aggregation) to merge shallow and deep features. This design fuses enhanced boundary features with the original feature map, resulting in a new feature map that combines both boundary and global context information. Through this approach, SPDC-YOLO is able to maintain high-resolution feature maps while integrating deep semantic information with shallow detail, thereby improving the detection of small objects.

Finally, a novel detection head, Dyhead-DCNv4, was proposed. This head innovatively replaces the DCNv2 module in the original Dyhead with the DCNv4 (Deformable Convolution v4) module. Dyhead is an adaptive detection head designed for object detection tasks, optimizing the detection process through the use of attention mechanisms. Unlike traditional fixed-structure detection heads, Dyhead dynamically adjusts its behavior based on the content of the input feature map, thereby improving detection performance, particularly in complex scenarios with multi-scale and multi-class objects.

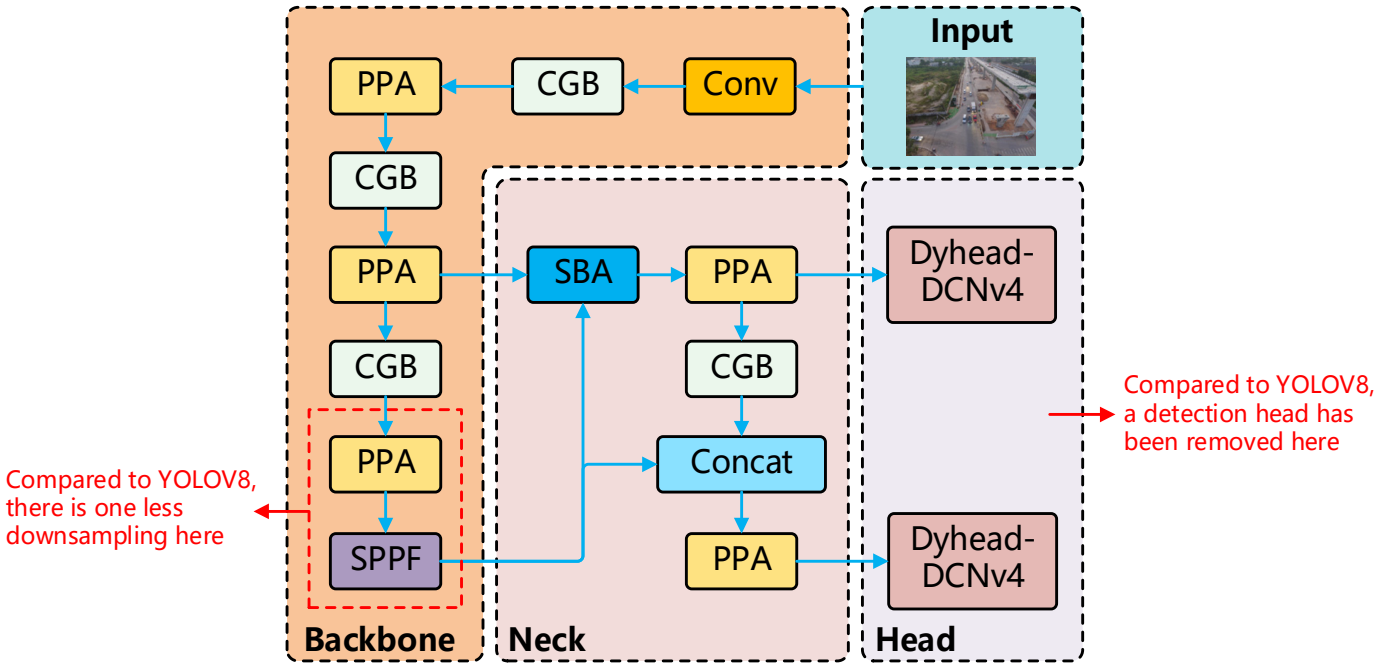


Figure 2. SPDC-YOLO structure diagram and its differences from Yolov8.

Additionally, SPDC-YOLO incorporates a novel downsampling module, the Context Guided Block (CGB), which effectively integrates both local and global contextual information to enhance the precision of small target detection. The model also replaces the C2f module with the Parallelized Patch-Aware Attention (PPA) module, enhancing the perception of local details and multi-scale features, thereby improving detection performance in complex scenarios. Moreover, the EIOU [48] loss function is employed, which integrates both shape and size information to optimize bounding box regression, yielding more precise bounding box predictions and further enhancing detection accuracy.

In conclusion, the proposed SPDC-YOLO model effectively addresses the limitations of YOLOv8 in small object detection by incorporating several key modules, thereby improving the detection performance of small objects in multi-scale aerial remote sensing images.

3.3. Proposed SPC-FPN

The SPC-FPN proposed in this paper is illustrated in Figure 3. Initially, the SBA (Selective Boundary Aggregation) module is utilized to fuse feature maps of size $40 \times 40 \times 512$ and $80 \times 80 \times 256$, simultaneously performing channel concatenation. The resulting fused feature map is then processed by the PPA (Parallelized Patch-Aware Attention) module for further feature extraction, with the output maintaining the dimensions of $80 \times 80 \times 256$. This feature map is subsequently downsampled via the CGB (Context Guided Block) module to a $40 \times 40 \times 512$ size. The downsampled feature map is then concatenated with another $40 \times 40 \times 512$ feature map from a parallel path, further enhancing the integration of features across multiple layers. Finally, the concatenated feature map undergoes another round of feature extraction through the PPA module, with the output size remaining $40 \times 40 \times 512$.

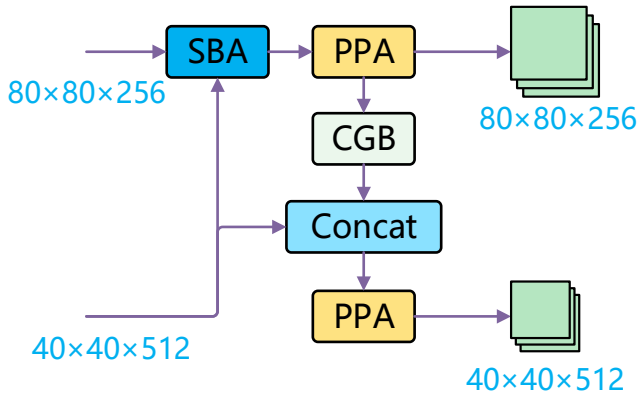


Figure 3. SPC-FPN structure diagram.

The design of this FPN enhances the multi-scale feature representation through the SBA module, while the CGB module effectively integrates contextual information to facilitate downsampling. Additionally, the PPA module strengthens the network's capacity for modeling feature relationships. The interaction of these modules works together to enhance the network's performance in detecting small objects, especially in remote sensing image analysis.

In SPC-FPN, the SBA module (Selective Boundary Aggregation [16]) is employed to selectively aggregate boundary features in the image rather than simply processing the features of the entire image. This strategy enables the model to direct greater attention to regions likely containing boundary-related information, thereby enhancing the representation of boundaries. Figure 4 shows the structure of the SBA module.

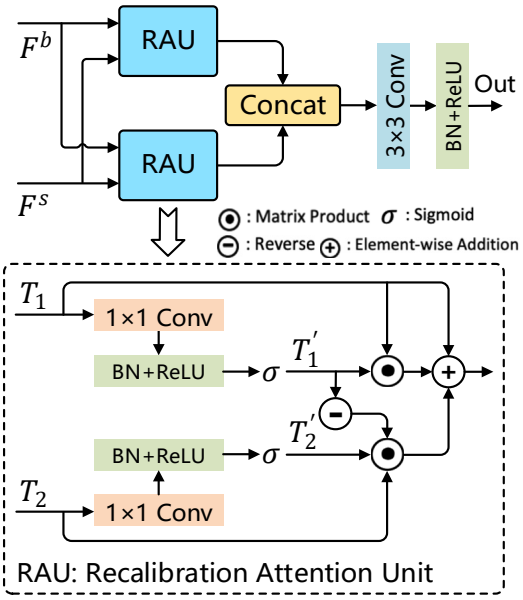


Figure 4. SBA module structure diagram.

Unlike traditional feature fusion methods, the SBA module introduces a novel RAU (Recalibration Attention Unit) designed to adaptively extract complementary representations from two input feature sets, F^b and F^s, before fusion. As shown in Figure 4, shallow and deep features are routed through distinct path ways into two RAU modules, allowing for the compensation of missing spatial boundary information in high-level semantic features and the lack of semantic details in low-level features. Outputs generated by these RAU modules are merged along the channel dimension before being fed into a 3×3 convolution layer. This aggregation strategy enables the robust fusion of multiple feature maps, further optimizing the feature representations. The operation of the RAU module, denoted as $H(\cdot, \cdot)$, can be expressed as follows:

$$T_1' = W_\theta(T_1),\ T_2' = W_\varnothing(T_2) \tag{1}$$

$$H(T_1, T_2) = T_1' \bigodot T_1 + T_2' \bigodot T_2 \bigodot (\ominus(T_1')) + T_1 \tag{2}$$

In this process, T_1 and T_2 represent the input features, which undergo linear mapping and sigmoid activation through $W_\theta(\cdot)$ and $W_\varnothing(\cdot)$, reducing the channel dimensions to 32 to yield the feature maps T_1' and T_2'. The symbol $\bigodot$ denotes element-wise multiplication, while $\ominus(\cdot)$ indicates the inverse operation, achieved by subtracting the feature map T_1'. Because high-level features do not contain specific boundary information, complementary regions and details are sequentially mined by erasing existing estimation regions from high-level output features, where existing estimates are upsampled from deeper layers. Using a convolution operation with a kernel size of 1×1 as the linear mapping process. Consequently, the SBA process can be expressed as follows:

$$Out = C_{3\times3}\left(Concat\left(H\left(F^s, F^b\right), H(F^b, F^s\right)\right)) \tag{3}$$

In this framework, $C_{3\times3}(\cdot)$ represents a 3×3 convolutional operation that includes batch normalization and a ReLU activation layer. The feature map F^s, with dimensions $40 \times 40 \times 512$, encapsulates deep semantic information, whereas F^b, with dimensions $80 \times 80 \times 256$, is rich in boundary details. The operation *Concat* indicates the concatenation of these feature maps along the channel dimension. Finally, the output of the SBA module, denoted as Out, is a feature map with dimensions $80 \times 80 \times 256$.

The innovations of the SBA module are as follows:

- Novelty of the SBA Module:
  - The SBA module integrates low-level boundary features with high-level semantic information to achieve precise modeling of object boundaries. This design effectively preserves boundary details, particularly benefiting small and boundary-ambiguous targets;
- Dynamic Feature Fusion:
  - The SBA module employs a dynamic feature fusion strategy that adaptively adjusts fusion weights based on the characteristics of input features. This mechanism enables the model to handle targets of varying scales and complexities more effectively;
- Optimization for Small Targets.

The SBA module specifically addresses the detection of small targets by aggregating low-level boundary information and high-level semantic features, allowing for more accurate localization of small target boundaries.

Traditional methods often perform simple concatenation or weighted summation of low-level and high-level features, which can lead to the loss or blurring of boundary information. The SBA module: The SBA module adaptively adjusts fusion weights through the RAU block, allowing for more precise processing of boundary features. This design not only preserves boundary details but also corrects them using semantic information, thereby improving detection accuracy.

Through the operation of the SBA module, the SPC-FPN robustly fuses features across different scales, thereby enhancing the network's feature representation. Shallow features, rich in boundary details and fine-grained information, are crucial for accurately delineating the contours of small objects, while deep features, enriched with semantic information, play a key role in distinguishing between object categories. This hierarchical feature aggregation strategy effectively captures the intricate boundaries and contours of small targets, significantly improving small object detection, particularly in drone imagery.

3.4. Replace C2f

In the standard YOLOv8 architecture, the C2f module begins by applying a convolutional layer (Conv1) to double the channel dimensions of the input feature map. This is followed by a series of Bottleneck modules that progressively extract features. Each Bottleneck module consists of multiple convolutional layers and can be configured with or without shortcut (residual) connections. Finally, the original feature map is concatenated with the output feature maps from the Bottleneck modules along the channel dimension. While this design enables the model to integrate multi-scale feature information and generate richer feature representations, it encounters limitations in small object detection. The low-resolution feature maps often fail to preserve fine details of small targets effectively, resulting in incomplete or insufficient feature representations for these objects.

Small objects cover only a limited number of pixels on feature maps, making them prone to being overlooked during feature fusion. Moreover, the C2f module's reliance on extensive contextual information can suppress the saliency of small objects, especially in complex backgrounds where noise interference further diminishes their prominence. As a result, small objects often fail to receive sufficient attention, adversely impacting detection performance. To address these limitations, the proposed SPDC-YOLO integrates a novel module, the PPA (Parallelized Patch-Aware Attention Module [15]), to replace the original C2f module. Figure 5 illustrates the structure of the PPA module.

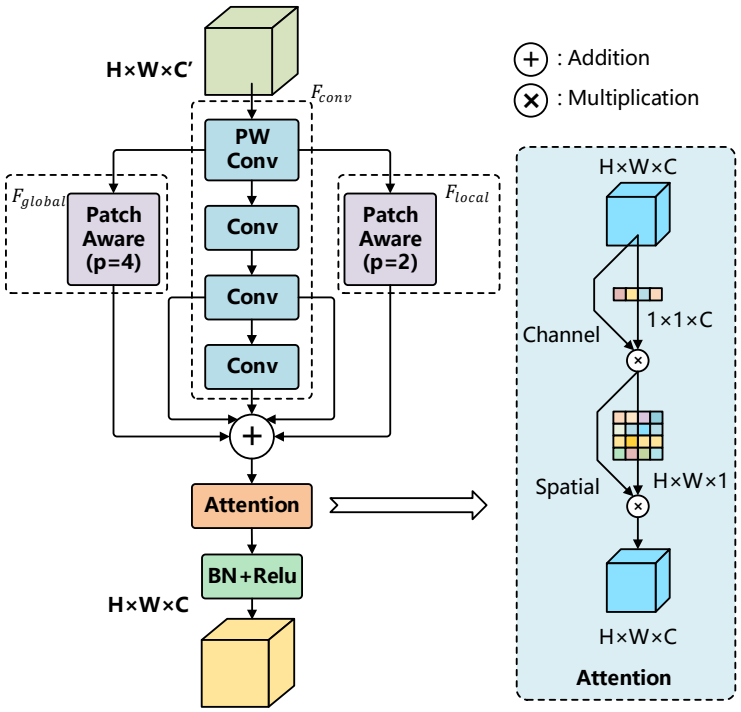


Figure 5. PPA module structure diagram.

The PPA module consists of an attention mechanism and multi-branch feature extraction as its two key components. In the feature extraction process, the input tensor F is first transformed into F' through pointwise convolution (PW Conv). It is then processed by three parallel branches: a local branch, a global branch, and a serial convolutional branch, generating F_{local}, F_{global}, and F_{conv}, respectively. The outputs of these branches are combined by summation, producing the aggregated feature tensor $\widetilde{F}$, with dimensions changing from $H \times W \times C'$ to $H \times W \times C$. The patch size parameter p distinguishes the local and global branches, facilitating spatial feature aggregation and displacement representation.

In summary, the input features are first processed through pointwise convolution (PW Conv) and multi-branch convolution (Conv) to extract critical representations. The Patch-Aware module then adjusts the patch size, facilitating the extraction of both local and global features. An attention mechanism further refines these extracted features, enhancing their discriminative power. Finally, the output features undergo batch normalization (BN) and ReLU activation, yielding the optimized feature representation.

After completing multi-branch feature extraction, the PPA module employs an attention mechanism to adaptively enhance the extracted features. This mechanism consists of two main components: Channel Attention and Spatial Attention, both of which contribute to refining the feature representations for improved performance.

The PPA module integrates multi-scale feature extraction with channel and spatial attention mechanisms, offering a more precise and enhanced representation for small object detection. This design is especially effective for detecting small objects in drone aerial imagery, as it not only captures the fine-grained features of small targets but also effectively suppresses background noise, thereby improving detection performance and accuracy.

3.5. Optimize Downsampling

In YOLOv8, the Conv Module faces several key challenges in detecting small objects. First, the limited receptive field of a single convolutional layer restricts its ability to capture the global contextual information needed for small object detection, as these objects occupy only a few pixels in an image. Second, the progressive reduction in feature map resolution

with increasing network depth leads to a loss of spatial information, making it difficult to recognize and localize small objects. Additionally, the ineffective fusion of multi-scale features in deeper layers often results in small object features being diluted or overlooked in higher-level representations, further compromising detection accuracy. Finally, convolutional layers are inherently more effective at extracting features from large objects, which occupy more pixels and exhibit clearer characteristics, while small objects lack prominent features, reducing the network's sensitivity to them.

To overcome these challenges, this paper introduces a new downsampling method, the Context Guided Block (CGB [14]), inspired by the human visual system's use of contextual information for scene interpretation. The CGB captures local features, surrounding context, and global context, seamlessly integrating them to enhance detection accuracy. By effectively fusing multi-scale feature information, this approach strengthens the model's capability to extract small object features, reducing resolution loss and preserving critical details.

The workflow of the CGB is depicted in Figure 6. The CGB comprises two key components: a local feature extractor, $f_{loc}(*)$, and a surrounding context feature extractor, $f_{sur}(*)$, designed to capture fine-grained features from the target region and contextual information from its surroundings, respectively. The local feature extractor $f_{loc}(*)$ utilizes a standard 3×3 convolution to extract localized details while preserving the input and output channel dimensions. Meanwhile, the surrounding context feature extractor $f_{sur}(*)$ employs a 3×3 dilated convolution with a specified dilation rate, expanding the receptive field to capture broader contextual information. This method successfully combines contextual information from the surrounding area, boosting the model's capacity to fully comprehend the target zone.

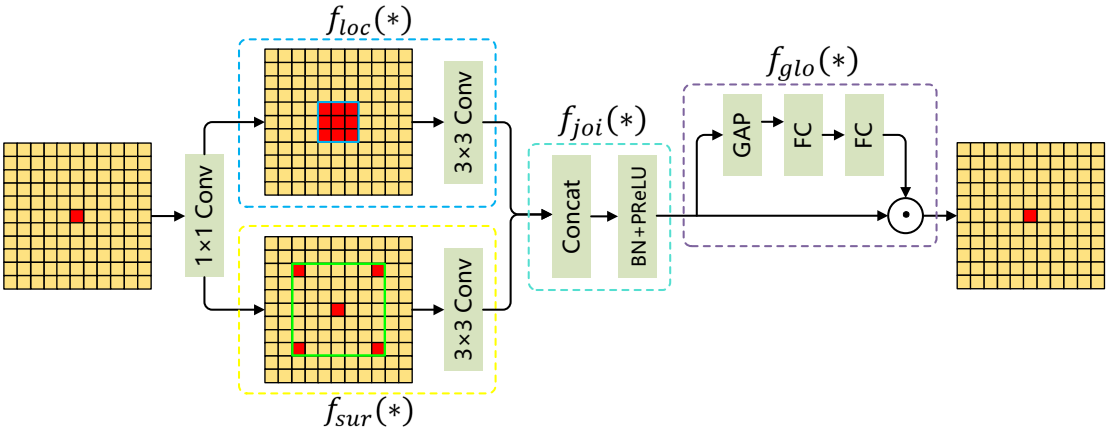


Figure 6. Context Guided Block flowchart.

The CGB merges local and surrounding context features along the channel dimension. The concatenated features undergo batch normalization (BN) and a parametric activation function (ReLU) to produce fused features, $f_{joi}(*)$, thereby boosting the model's representational power. To further optimize these fused features, the CGB employs a global context extractor, $f_{glo}(*)$ which utilizes global average pooling (GAP) to capture global contextual information. A fully connected layer is applied to reweight each channel, enhancing the representation of important features while reducing irrelevant ones. This approach enables the model to focus more effectively on semantically significant regions, improving overall detection performance.

The Context Guided Block provides substantial advantages in small object detection by integrating multi-scale features that combine local, contextual, and global information. This fusion enables the model to capture fine details of small objects while simultaneously understanding their surrounding context, thereby enhancing detection accuracy. The global

context module refines key features while mitigating irrelevant noise, allowing the model to focus more effectively on small targets. Additionally, the adaptive feature enhancement mechanism adjusts feature weights based on the input content, enabling the model to handle complex scenarios while preserving the details of small objects. These combined capabilities improve both the accuracy and robustness of small target detection.

3.6. Enhancing DyHead with DCNv4

Compared to the basic YOLOv8, the proposed SPDC-YOLO introduces a new detection head, Dyhead-DCNv4, in the head section. The core innovation lies in replacing the DCNv2 module in the Dyhead (Dynamic Head) with the DCNv4 (Deformable Convolution v4 [17]) module. Additionally, the P5 detection head has been removed in SPDC-YOLO, as it relies on a 32× downsampled feature map, which negatively impacts the detection of small targets smaller than 32 pixels.

DCNv4 removes the softmax normalization during spatial aggregation, enhancing the network's dynamic adaptability and representational capacity. The removal of softmax overcomes its inherent limitations, including restricted convergence speed and reduced operator expressiveness. By adopting adaptive windows and dynamically unconstrained weights, DCNv4 provides greater flexibility in feature processing. Moreover, DCNv4 optimizes memory access patterns through instruction-level kernel analysis, reducing redundant operations and significantly improving computational efficiency. The optimized memory access reduces overhead by a factor of two, substantially enhancing the operator's overall performance.

Figure 7 illustrates the structure of DyHead, comprising three core attention mechanisms: scale-aware attention (π_L), spatial-aware attention (π_S), and task-aware attention (π_C). These modules are applied sequentially to the input features, enabling dynamic weighting and adaptive refinement of feature representations at each stage.

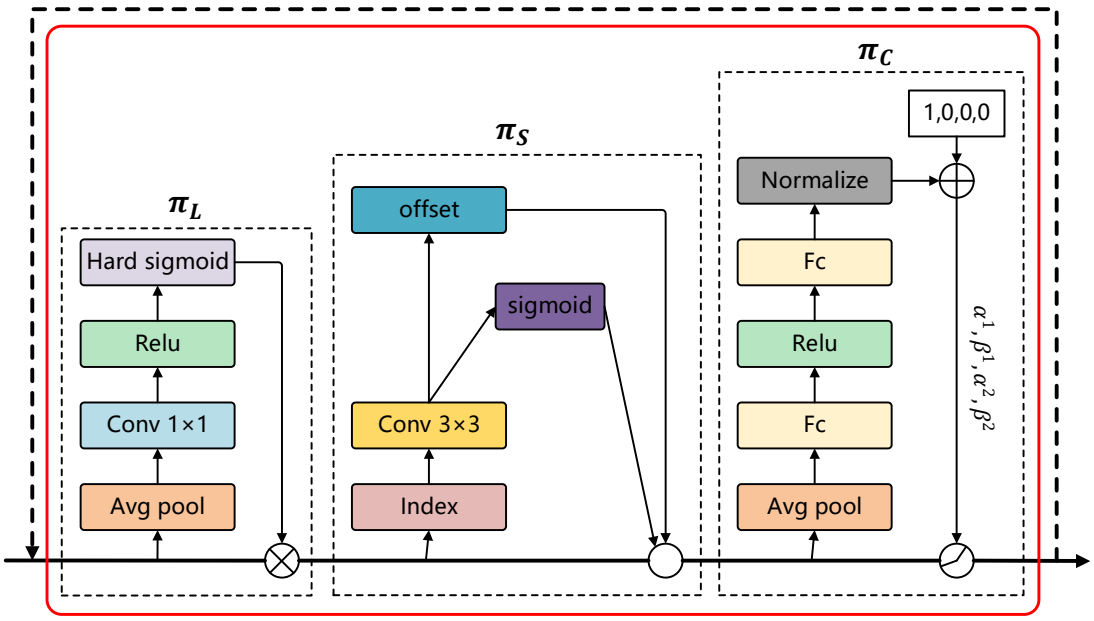


Figure 7. Dyhead Structure Diagram.

Given a feature tensor $F \in \mathbb{R}^{L \times S \times C}$, its self-attention mechanism can be generally expressed in the following form:

$$W(F) = \pi(F) \cdot F \tag{4}$$

In this context, $\pi(\cdot)$ denotes the attention function. Due to the prohibitive computational cost and impracticality of directly learning an attention function across all dimensions, DyHead decomposes it into three sequential attention modules, each focusing on a distinct perspective:

$$W(F) = \pi_C(\pi_S(\pi_L(F)\cdot F)\cdot F)\cdot F \tag{5}$$

The computation methods for π_L, π_S, and π_C in the above equations are defined as follows:

$$\pi_L(F)\cdot F = \sigma(f(\frac{1}{SC}\sum_{S,C} F))\cdot F \tag{6}$$

$$\pi_S(F)\cdot F = \frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K} w_{l,k}\cdot F(l; p_k + \Delta p_k; c)\cdot \Delta m_k \tag{7}$$

$$\pi_C(F)\cdot F = \max\left(\alpha^1(F)\cdot F_c + \beta^1(F), \alpha^2(F)\cdot F_C + \beta^2(F)\right) \tag{8}$$

In Equation (6), $f(\cdot)$ represents a linear function, while $\sigma(x) = \max\left(0, \min\left(1, \frac{x+1}{2}\right)\right)$ denotes a hard-sigmoid function. In Equation (7), K represents the number of sparse sampling positions and $p_k + \Delta p_k$ indicates the sampling positions adjusted by the learnable spatial offsets Δp_k. The term Δm_k represents a learnable importance scalar at position p_k.

In Equation (8), F_c refers to the feature slice of the c-th channel and $\left[\alpha^1, \alpha^2, \beta^1, \beta^2\right]^T = \theta(\cdot)$ is a meta-function designed to learn and control activation thresholds. This process involves global average pooling along the L and S dimensions to reduce spatial resolution, followed by two fully connected layers and a normalization layer. The output is then normalized to the range $[-1, 1]$ using a shifted sigmoid function.

Through this multi-level attention mechanism design, Dyhead-DCNv4 is capable of adapting to complex features across multiple scales, spatial dimensions, and tasks. It also enhances processing speed, significantly improving the detection performance of multi-scale small objects in aerial remote-sensing images.

3.7. Optimize Loss Function

In small object detection, the CIoU loss function used in YOLOv8 has several shortcomings, including low sensitivity to small positional shifts, excessive penalties for minor center offsets, and a significant impact due to aspect ratio mismatches. Additionally, since small objects typically yield lower IoU values, CIoU is less effective at optimizing such low-IoU bounding boxes, which hampers the reduction of errors and ultimately reduces detection accuracy for small objects.

To overcome these limitations, the proposed SPDC-YOLO utilizes the EIoU [48] loss function, which considers both the distance between centers and differences in aspect ratios between the predicted and true bounding boxes. This allows for more accurate optimization of box regression, leading to predictions that align more closely with the actual target and enhancing localization accuracy in object detection.

The EIoU loss comprises three components: IoU loss (L_{IOU}), distance loss (L_{dis}), and aspect ratio loss (L_{asp}). The IoU loss (L_{IOU}) quantifies the intersection between the predicted and ground-truth bounding boxes, quantifying their intersection. The distance loss (L_{dis}) captures the difference between the center points of the predicted and target boxes using the Euclidean distance, which is normalized in the formulation to emphasize larger discrepancies. The aspect ratio loss (L_{asp}) considers the variations in width and height between the predicted and target boxes, reducing shape discrepancies by normalizing the squared differences of width and height.

In Figure 8, the blue rectangle represents the Anchor Box with an area A, while the red rectangle represents the Target Box with an area B. The yellow rectangle denotes the smallest enclosing box that encompasses both the Anchor Box and the Target Box. The Euclidean distance between the centers of the Anchor Box and the Target Box is represented by $\rho(b, b^{gt})$. The EIoU calculation is given by the following:

$$L_{EIOU} = L_{IOU} + L_{dis} + L_{asp} \tag{9}$$

$$L_{IOU} = 1 - \frac{|A \cap B|}{|A \cup B|} \tag{10}$$

$$L_{dis} = \frac{\rho^2(b, b^{gt})}{(w^c)^2 + (h^c)^2} \tag{11}$$

$$L_{asp} = \frac{\rho^2(w, w^{gt})}{(w^c)^2} + \frac{\rho^2(h, h^{gt})}{(h^c)^2} \tag{12}$$

The enhanced distance and aspect ratio losses in EIoU not only retain the advantages of CIoU but also more effectively reduce discrepancies in bounding box dimensions. As a result, EIoU ensures better alignment of both position and shape. This precise loss function accelerates convergence and enhances localization accuracy, making it particularly well-suited for small object detection tasks in drone imagery.

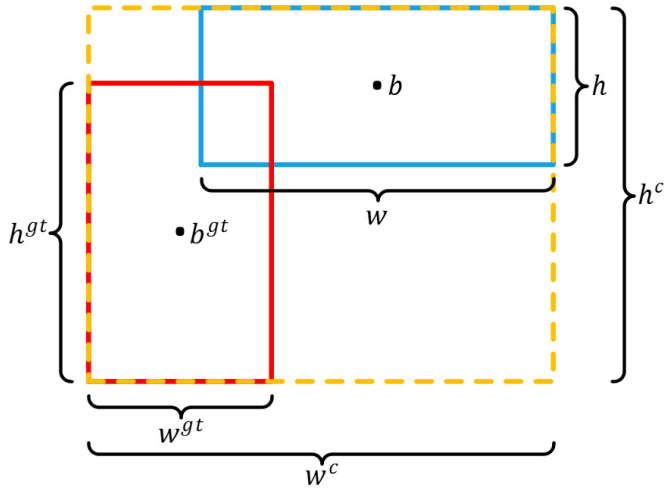


Figure 8. EIoU Calculation Diagram.

4. Experiment

4.1. Experimental Dataset

This study employs the VisDrone2019 dataset for validation, which was meticulously collected and annotated by the AISKYEYE team at Tianjin University's Machine Learning and Data Mining Laboratory. The dataset is partitioned into 6471 training images, 548 validation images, and 1610 test images, encompassing 10 distinct target categories. And the spatial resolution of the dataset ranges from 0.02 m per pixel to 0.1 m per pixel. The distribution of data in the training set is shown in Figure 9.

In Figure 9, the top-left plot illustrates the distribution of target counts across different categories, with the x-axis representing the various categories in the dataset and the y-axis indicating the number of targets in each category. The top-right plot illustrates the styles and sizes of the target bounding boxes within the dataset. The bottom-left plot illustrates the distribution of the center points of the target bounding boxes within the image. The x-axis and y-axis represent the normalized coordinates of these center points along the

image's width and height, respectively, and the color intensity indicates the number of targets at each location. The bottom-right plot illustrates the distribution of target aspect ratios relative to the entire image. The x-axis and y-axis represent the normalized width and height of the targets within the image, respectively, scaled between 0 and 1, and the color intensity indicates the number of targets at each aspect ratio.

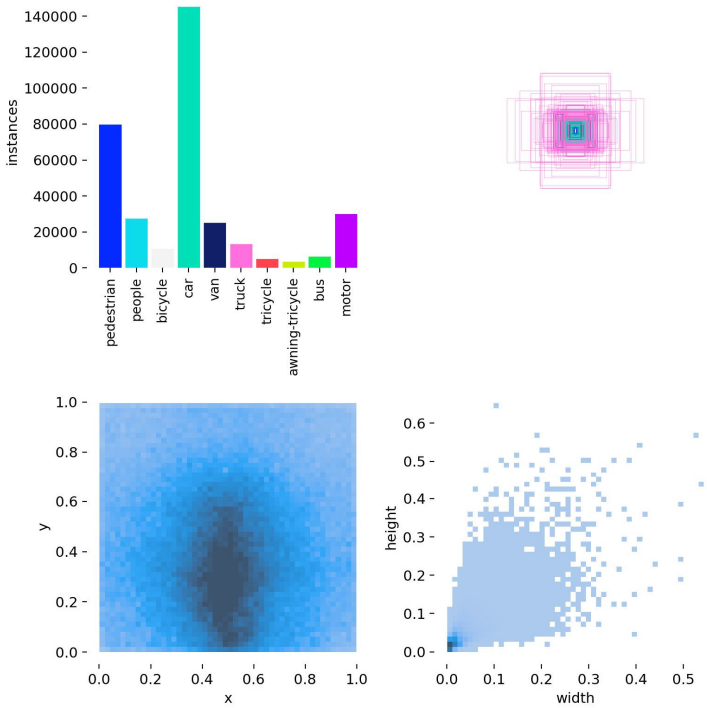


Figure 9. VisDrone2019 data distribution.

Furthermore, the dataset comprises high-resolution images acquired from a wide range of urban and rural settings, including traffic-heavy areas and bustling street environments. The images are captured from multiple viewpoints and feature diverse backgrounds, thereby contributing to the dataset's high degree of realism. Example samples from the dataset are presented in Figure 10.



Figure 10. Typical images from the VisDrone2019 dataset.

4.2. Experimental Setup and Evaluation Metrics

4.2.1. Experimental Setup

In the experiment of this article, the Ubuntu 18.04 operating system was employed as the experimental platform, utilizing Python version 3.12, PyTorch version 2.5.0, and CUDA version 11.8. The proposed models were trained, tested, and validated under consistent hyperparameter settings. The computational hardware configuration included an Intel(R) Xeon(R) Gold 5220R processor and an NVIDIA A100-PCIE GPU. Additional essential training parameters are detailed in Table 1.

Table 1. Experimental parameter configurations.

Training Parameters	Batch Size	32
	Epochs	300
	Optimizer	SGD
	Momentum	0.9
	image size	1024 × 640
	Initial learning rate	0.01
Computer configurations	Operating System	Ubuntu 18.04
	CPU	1 GHz
	Graphics memory	40 GB

4.2.2. Evaluation Metrics

To comprehensively and effectively evaluate the proposed model, four metrics were employed: Precision (P), Recall (R), Average Precision (AP), mean Average Precision (mAP), the number of neural network parameters (M), and Giga Floating-point Operations Per Second (GFLOPs). The calculation method for some evaluation indicators is as follows:

$$R = \frac{TP}{TP + FN} \tag{13}$$

$$P = \frac{TP}{TP + FP} \tag{14}$$

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i \tag{15}$$

In the formulas presented above, C represents the number of classes, while Average Precision (AP) is derived by calculating the area under the Precision–Recall (P-R) curve. False Positives (FP) are negative samples that are mistakenly anticipated to be positive, True Negatives (TN) are correctly categorized negative samples, True Positives (TP) are correctly identified positive samples, and False Negatives (FN) are positive samples that are misclassified as negative.

4.3. Ablation Experiment

A number of ablation studies were conducted on the validation set of the VisDrone2019 dataset in order to fully assess the efficacy of the enhancements made to SPDC-YOLO. These experiments aimed to isolate and quantify the contribution of each individual enhancement to model performance, as well as to assess the synergistic effects when these techniques are integrated. All experiments were carried out using consistent hyperparameter settings to guarantee the comparability of the results. This systematic approach enables a more precise analysis of the specific impact of each improvement and reveals how these modifications work in concert to optimize the overall performance of SPDC-YOLO.

The ablation experiment results are shown in Table 2. After removing the P5 detection head and introducing SPC-FPN, the model achieves a 66.7% reduction in parameters, with only a 1% decrease in mAP_{50}. This indicates that the P5 detection head is not critical for small object detection. Additionally, replacing the original C2f module with the PPA module on top of SPC-FPN outperforms the original YOLOv8n while reducing the parameter count by more than 50%. The integration of Dyhead-DCNv4 resulted in a 2.2% increase in mAP_{50}, underscoring the advantages of DCNv4-enhanced Dyhead for small object detection. Lastly, incorporating the EIoU loss function delivered a significant improvement: compared to YOLOv8n, mAP_{50} increased by 3.4 percentage points, mAP_{50-95} improved by 2.4 percentage points, and the parameter count was reduced by 103 million.

Table 2. Ablation experiment result in VisDrone2019-val.

	SPC-FPN	CGB	PPA	DyHead-DCNv4	EIoU	mAP_{50} (%)	mAP_{50-95} (%)	Parameters (M)
YOLOv8n						43.1	26.0	3.00
	√					42.1	25.3	1.00
	√	√				42.8	26.0	1.14
	√		√			43.2	26.2	1.24
	√			√		44.8	27.3	1.59
	√	√	√			43.7	26.7	1.38
	√		√	√		45.4	27.7	1.84
	√	√	√	√		45.9	28.1	1.97
SPDC-YOLO-n	√	√	√	√	√	46.5	28.4	1.97

4.4. Comparative Experiment on Improving Modules

4.4.1. Comparison of the Effectiveness of Different Feature Fusion Methods

To further evaluate the effectiveness of the SBA module in small object detection tasks, we introduced the SBA module into the YOLOv8 framework and compared its performance with several common feature fusion methods. The VisDrone2019 validation set was chosen as the experimental dataset to assess the performance differences between the SBA module and other feature fusion techniques in complex real-world scenarios. The experimental results, presented in Table 3, show a comparison of various methods across different metrics for small object detection. Through these comparative experiments, we are able to clearly analyze the advantages of the SBA module in enhancing small object detection performance.

Table 3. Comparison of different feature fusion methods.

Methods	P (%)	R (%)	mAP_{50} (%)	mAP_{50-95} (%)	Test Speed (ms)
biFPN	52.1	42.8	43.9	26.5	2.4
CGRFPN	51.4	39.7	41.3	24.6	2.8
Yolov8	51.5	42.3	43.1	26.0	2.5
Yolov8-SBA	52.3	42.6	44.0	26.6	2.4

The experimental results demonstrate that the SBA module exhibits a clear advantage in small object detection tasks compared to other feature fusion methods. YOLOv8-SBA performs exceptionally well in terms of accuracy, particularly with improvements in mAP_{50} (44.0% vs. 43.1%) and mAP_{50-95} (26.6% vs. 26.0%), highlighting the effectiveness of the SBA module in small object detection. Additionally, YOLOv8-SBA outperforms the baseline YOLOv8 in Precision (52.3%) and Recall (42.6%), indicating an enhancement in small object detection capability. While the inference speed of YOLOv8-SBA (2.4 ms) is comparable to YOLOv8 (2.5 ms) and faster than CGRFPN (2.8 ms), the performance improvement does not

significantly impact computational efficiency. In contrast, biFPN shows better performance in mAP_{50} but lags behind YOLOv8-SBA in mAP_{50-95} and small object detection. Therefore, by incorporating the SBA module, YOLOv8-SBA achieves superior overall performance, enhancing small object detection accuracy while maintaining a high inference speed.

4.4.2. Comparison Experiment of Different Improved Detection Heads

To further validate the effectiveness of using Dyhead-DCNv4 in improving the model's small object detection accuracy, we conducted a detailed comparative experiment on the VisDrone2019 validation set. In the experiment, all other modules were kept constant, and the performance of different detection heads, including traditional and several advanced detection head structures, was compared. Through this comparison, we aim to thoroughly analyze the advantages of Dyhead-DCNv4 in small object detection tasks. The experimental results are presented in Table 4.

Table 4. Comparison between Dyhead-DCNv4 and other detection heads.

Head	P (%)	R (%)	mAP_{50} (%)	mAP_{50-95} (%)	Test Speed (ms)
SEAMhead	51.1	41.9	42.4	25.8	3.7
LSCD	51.9	42.1	42.6	25.9	3.6
EfficientHead	51.8	41.4	42.7	25.8	**3.3**
LSDECD	51.4	41.9	42.4	25.6	3.8
Dyhead	**55.5**	42.6	45.1	27.7	3.9
Dyhead-DCNv4	55.4	**43.6**	**45.9**	**28.1**	3.6

Note: Bold data indicate the best performance.

The experimental results show that Dyhead-DCNv4 performs excellently in small object detection tasks, particularly in terms of accuracy and detection capability. It significantly outperforms other methods in Precision (55.4%) and Recall (43.6%) while also excelling in mAP_{50} (45.9%) and mAP_{50-95} (28.1%). Compared to Dyhead, Dyhead-DCNv4 shows improvements in both mAP_{50} and mAP_{50-95}, demonstrating that integrating DCNv4 into Dyhead effectively enhances small object detection. The inference speed of Dyhead-DCNv4 is 3.6 ms, which is slightly slower than EfficientHead (3.3 ms), but it achieves a better balance between performance and speed by improving accuracy. Overall, Dyhead-DCNv4 improves detection accuracy while maintaining high efficiency, outperforming other methods. This indicates that the enhanced DCNv4 structure significantly contributes to performance improvement.

4.4.3. Comparative Experiment of Different Loss Functions

In the proposed SPDC-YOLO model, EIoU was employed as the loss function. To demonstrate its effectiveness, a series of comparative experiments were conducted after integrating Dyhead-DCNv4 while maintaining all other components unchanged. EIoU was compared against DIoU, GIoU, ShapeIoU, Inner_CIoU, and Inner_DIoU on the VisDrone2019 validation set, with the outcomes shown in Table 5. These experiments aim to highlight the advantages of EIoU in enhancing detection performance, particularly for small targets in remote sensing imagery.

According to the experimental findings, the EIoU loss function performs exceptionally well across a number of important criteria, including Recall (R), mAP_{50}, and mAP_{50-95}. While the precision achieved by EIoU is slightly lower than that of Inner_DIoU, it shows a notable improvement in recall, highlighting its enhanced sensitivity to small object detection. The comparative analysis of these results provides compelling evidence of the effectiveness of the EIoU loss function for small target detection tasks, particularly within remote sensing imagery.

Table 5. Comparison between EIoU and other loss functions.

Loss Function	P (%)	R (%)	mAP_{50} (%)	mAP_{50-95} (%)
CIoU	55.4	43.6	45.9	28.1
DIoU	54.8	44	46.2	28.3
GIoU	55.6	43.9	46.1	28.2
ShapeIoU	55.1	44.1	46.1	28.2
Inner_CIoU	55.3	43	45.7	27.8
Inner_DIoU	**56.3**	43.9	46.3	28.3
EIoU	55.3	**44.8**	**46.5**	**28.4**

Note: Bold data indicate the best performance.

4.5. Comparison Between SPDC-YOLO and Other Models

4.5.1. Comparison with Traditional Models

To further evaluate the performance of SPDC-YOLO in small object detection tasks, we conducted a detailed comparative experiment on the VisDrone2019 validation set. In the experiment, we compared the detection results of SPDC-YOLO with several classic models, which represent various architectures and complexities. Through these comparisons, we were able to comprehensively assess the performance of SPDC-YOLO in real-world applications. The experimental results, presented in Table 6, show the differences between the models across various metrics.

Table 6. Comparison of SPDC-YOLO and classical models.

Model	mAP_{50} (%)	mAP_{50-95} (%)	Parameters (M)	GFLOPs	Test Speed (ms)
Faster R-CNN [49]	33.2	19.9	-	208	72.1
RetinaNet [49]	29.0	17.2	-	210	74.3
Cascade R-CNN [50]	39.3	25.6	68.9	-	-
EfficientDet [50]	38.5	54.6	34.7	-	-
DTSSNet [51]	39.9	24.2	10.1	50.4	12.8
ATSS [51]	31.7	18.6	10.3	57.0	13.2
SPDC-YOLO-n	**46.5**	**28.4**	**1.97**	**10.0**	**3.5**
SPDC-YOLO-s	**51.1**	**31.8**	**5.34**	**30.0**	**4.5**

Note: All of the experimental data presented above are derived from published papers. Bold data indicate the model proposed in this paper.

The experimental results demonstrate that SPDC-YOLO-s and SPDC-YOLO-n significantly outperform other models across key metrics, particularly in mAP_{50} and mAP_{50-95}. SPDC-YOLO-s achieved an mAP_{50} of 51.1% and an mAP_{50-95} of 31.8%, showing excellent detection accuracy. SPDC-YOLO-n also demonstrated strong performance with an mAP_{50} of 46.5% and an mAP_{50-95} of 28.4% while maintaining a low parameter count (1.97 M) and reduced computational cost (10.0 GFLOPs), ensuring high efficiency.

4.5.2. Comparison with YOLO Series Models

Similarly, to demonstrate the competitiveness of the proposed SPDC-YOLO, a comprehensive comparison was conducted between SPDC-YOLO and various versions of the YOLO series models on the VisDrone2019 validation set. The performance of each model across a range of key parameters was thoroughly evaluated, emphasizing the advantages of SPDC-YOLO in small object detection. The experimental results are shown in Table 7.

Table 7. Comparison of SPDC-YOLO and YOLO Series Models.

Network	P (%)	R (%)	mAP_{50} (%)	mAP_{50-95} (%)	Parameters (M)	GFLOPs	Test Speed (ms)
Yolov3-tiny	39.5	30.5	29.0	14.8	8.68	12.9	4.4
Yolov5-n	47.1	37.7	37.7	21.3	**1.77**	**4.2**	**2.1**
Yolov6	46.5	40.0	36.1	21.0	4.63	29.04	12.1
Yolov8-n	51.5	42.3	43.1	26.0	3.0	8.1	2.5
Yolov9-t	54.3	43.3	44.9	22.7	2.62	10.7	3.8
Yolov10-n	51.4	40.5	42.2	25.6	2.69	8.2	3.0
Yolov11-n	50.5	42.2	42.5	25.8	2.58	6.3	2.4
SPDC-YOLO-n	**55.3**	**44.8**	**46.5**	**28.4**	1.97	10.0	3.5

Note: Bold data indicate the best performance.

The experimental results show that SPDC-YOLO-n outperforms other networks in both mAP_{50} and mAP_{50-95}. For example, YOLOv3-tiny achieves only 29.0% in mAP_{50}, while SPDC-YOLO-n improves by 17.5 percentage points. Compared to YOLOv5-n and YOLOv8-n, SPDC-YOLO-n also achieves higher detection accuracy. SPDC-YOLO-n has a parameter count of just 1.97 M, GFLOPs of 10.0, and an inference speed of 3.5 ms, offering a low computational cost. In contrast, YOLOv6 has a slower inference speed, while SPDC-YOLO-n strikes a better balance between accuracy and efficiency. Overall, SPDC-YOLO-n demonstrates excellent performance in small object detection, combining high precision with high efficiency.

4.5.3. Comparison with Other Improved YOLO Methods

To further evaluate the performance of the SPDC-YOLO model, we conduct a comparative analysis against various YOLO-based improved models on the VisDrone2019 validation set. These models are specifically designed for small object detection. The experimental results are summarized in Table 8.

Table 8. Comparison with an improved model based on YOLO.

Model	mAP_{50} (%)	mAP_{50-95} (%)	Parameters (M)	GFLOPs	Test Speed (ms)
GCGE-YOLO [52]	34.1	19.2	4.5	10.8	4.4
C3TB-YOLOv5 [53]	38.3	22.0	8.0	19.7	5.3
TPH-YOLO [45]	39.3	23.6	51.5	138.1	22.1
LE-YOLO [54]	39.3	22.7	2.1	13.1	4.1
Modified YOLOv8 [55]	42.2	-	9.66	-	-
UAV-YOLOv8 [56]	47.0	29.2	10.3	-	-
Drone-YOLO-s [57]	44.3	27.0	10.9	-	-
SOD-YOLO-s [58]	42.0	25.1	1.75	20.1	3.9
SPDC-YOLO-n	**46.5**	**28.4**	**1.97**	**10.0**	**3.5**
SPDC-YOLO-s	**51.1**	**31.8**	**5.34**	**30.0**	**4.5**

Note: Bold data indicate the model proposed in this paper.

The experimental results in Table 8 show that UAV-YOLOv8 slightly outperforms SPDC-YOLO-n but at the cost of a significant increase in parameter count. Notably, compared to UAV-YOLOv8, SPDC-YOLO-s achieves a 4.1 percentage point improvement in mAP_{50} while maintaining a lower parameter count, and SPDC-YOLO demonstrates faster inference speed than other methods. Overall, the proposed SPDC-YOLO model exhibits exceptional efficiency and accuracy, achieving higher mAP_{50} and mAP_{50-95} scores without compromising inference speed.

4.6. Experimental Results on Each Subcategory

To further assess the performance improvements of SPDC-YOLO over YOLOv8 in multi-class object detection tasks, experiments were conducted using the VisDrone2019 dataset, evaluating each of the ten classes individually. The models compared include YOLOv8n, YOLOv8s, SPDC-YOLO-n, and SPDC-YOLO-s, with their performance evaluated on both the validation and test sets. Detection accuracy for each class was quantified using mAP_{50} (mean Average Precision), and bar charts were utilized to visually present the performance differences across categories. Figure 11 shows the findings for the validation set, and Figure 12 shows the results for the test set.

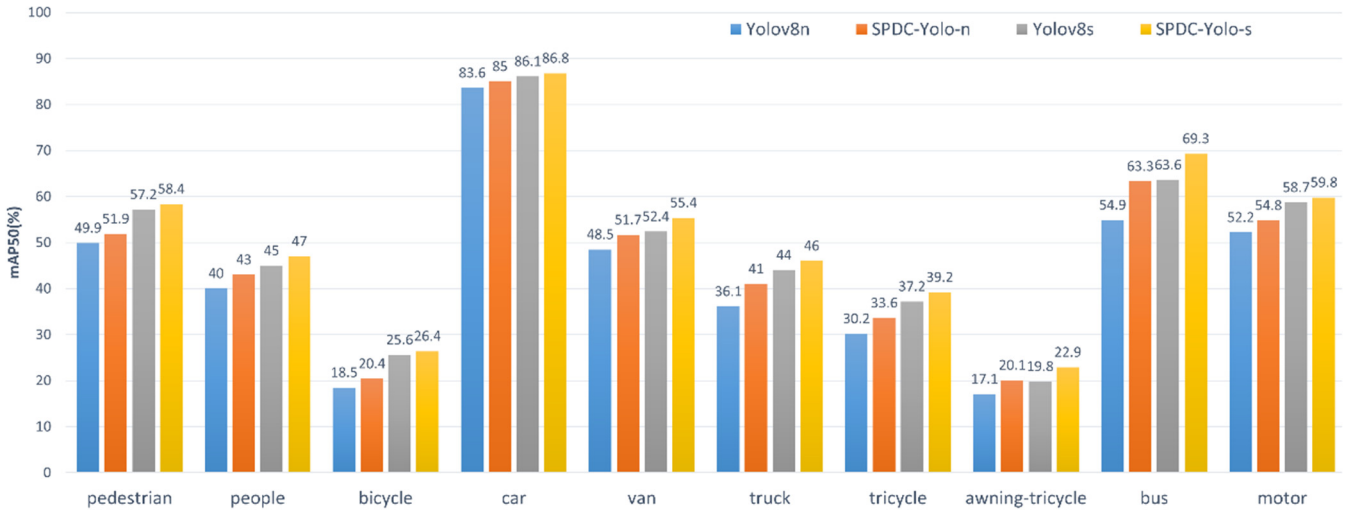


Figure 11. Comparison of different models in the validation set for each category mAP_{50}.

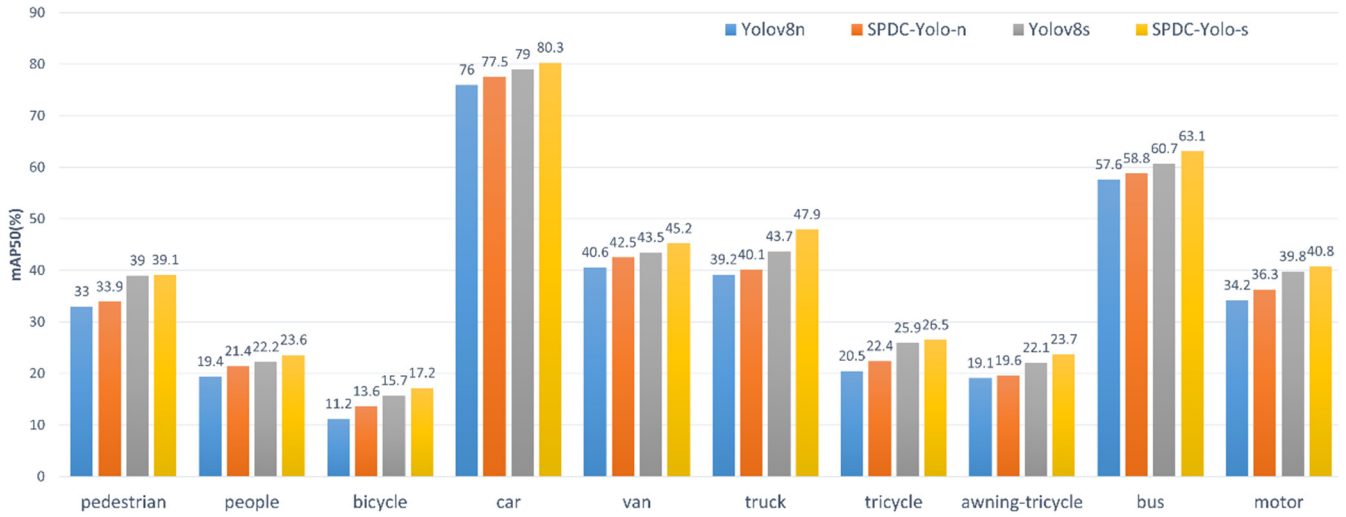


Figure 12. Comparison of different models in the test set for each category mAP_{50}.

According to the experimental findings, SPDC-YOLO performs better than the original YOLOv8 model in every category on both the test and validation sets. On the validation set, SPDC-YOLO-n achieved an mAP_{50} of 63.3% for the "bus" category, reflecting an 8.4 percentage point improvement over YOLOv8n, while SPDC-YOLO-s reached an mAP_{50} of 69.3%, marking a 5.7 percentage point gain over YOLOv8s. These results emphasize the effectiveness of the enhancements incorporated into SPDC-YOLO. On the test set, while the improvements were less pronounced for challenging categories such as "people", "bicycle", and "awning-tricycle", SPDC-YOLO still demonstrated noticeable performance

gains, underscoring the robustness of the enhanced model across various categories. SPDC-YOLO outperformed in detection accuracy across all categories, showcasing improved generalization for small object detection in remote sensing imagery.

4.7. Visualization Experiment

To evaluate SPDC-YOLO-n's detection capabilities against the reference YOLOv8n, three images were selected from the VisDrone2019 test dataset. These images were chosen to represent diverse scenes, varying lighting conditions, and different viewing angles. Key detection areas were highlighted with red circles to provide a clearer visualization of the results. This approach enables an intuitive comparison of the performance differences between SPDC-YOLO-n and YOLOv8n under real-world conditions.

Figure 13 shows an image captured by a drone at dusk, depicting an intersection that poses significant challenges for object detection due to low-light conditions. The image includes three categories: car, motor, and pedestrian. The baseline YOLOv8 model struggled to accurately detect pedestrians on the crosswalk and motors parked along the roadside. In contrast, the proposed SPDC-YOLO model successfully detected these challenging targets. Red circles have been used to highlight these difficult-to-detect objects for easier observation and analysis. These results demonstrate the enhanced capability of SPDC-YOLO to handle complex detection scenarios, particularly under low-light conditions.

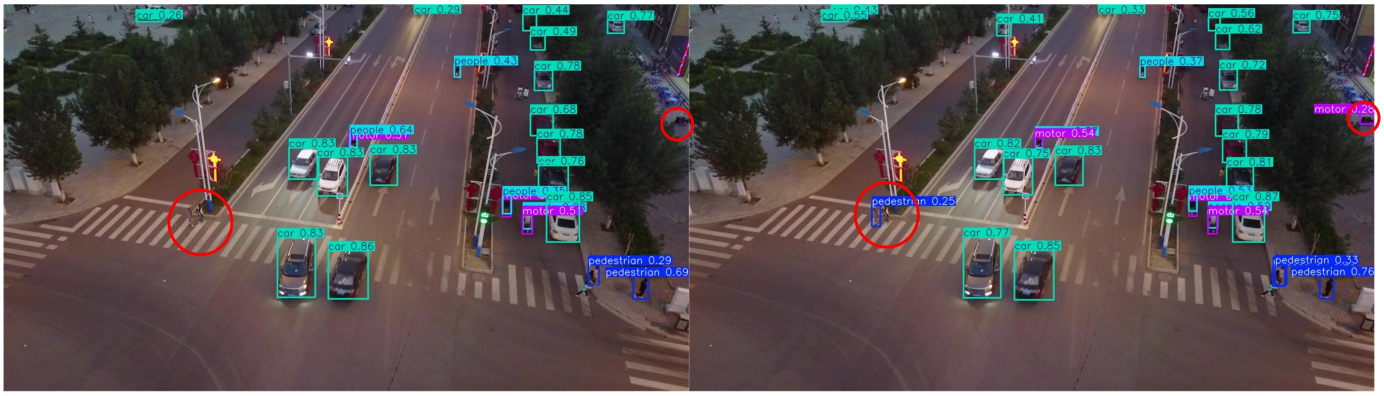
Figure 14 depicts an overhead view of an urban streetscape captured during daylight. The ample lighting provides clearer visibility of vehicle contours and colors in well-lit areas, facilitating object detection. However, intense illumination can also create glare on some vehicle surfaces, potentially interfering with detection algorithms. Conversely, in shadowed areas with limited lighting, vehicle features may be less distinct, making it challenging for the algorithm to accurately capture details such as color and shape. In Figure 14, the YOLOv8n model failed to detect a vehicle marked by a red circle, whereas the proposed SPDC-YOLO model successfully identified it. This result demonstrates the superior adaptability of SPDC-YOLO to varying lighting conditions, enabling it to maintain high detection accuracy and precise localization across diverse illumination scenarios.

Figure 15 presents an overhead view of an urban street captured by a drone at night. The image primarily features cars, trucks, and buses, with the movement of vehicles captured as light trails, illustrating their trajectories. Notably, several vehicles are also present in areas along the street that are not directly illuminated by streetlights. The uneven lighting throughout the scene significantly increases the complexity of the detection task. Comparing panels (a) and (b) in Figure 15 reveals that the baseline YOLOv8n model failed to detect the vehicles marked by red circles, whereas the proposed SPDC-YOLO successfully identified these challenging targets. This result further demonstrates the superior performance of SPDC-YOLO under uneven lighting conditions, effectively enhancing detection accuracy in complex nighttime environments.

To validate the effectiveness of the PPA module in small object detection tasks, this study compares the performance of the original YOLOv8 model and the YOLOv8 model enhanced solely with the PPA module in an urban intersection scenario. Specifically, both models were applied to the same input image to generate the corresponding heatmaps, as illustrated in Figure 16.

As shown in Figure 16, the heatmap generated by the PPA-enhanced model exhibits a broader coverage area, a more uniform overall heat intensity distribution, and a higher number of detected vehicles. Additionally, to provide a more intuitive comparison of detection differences in the same regions, red rectangles were used to highlight corresponding areas in both models. The comparison reveals that in these regions, the PPA-enhanced

model demonstrates higher heat intensity and detects more small objects, further validating the effectiveness of the PPA module in improving small object detection performance.



(a) YOLOv8-n detection effect (b) SPDC-YOLO-n detection effect

Figure 13. Aerial images taken at a tilted angle at the intersection in the evening.



(a) YOLOv8-n detection effect (b) SPDC-YOLO-n detection effect

Figure 14. Road images taken under strong light.



(a) YOLOv8-n detection effect (b) SPDC-YOLO-n detection effect

Figure 15. Images captured by drones over the road at night.

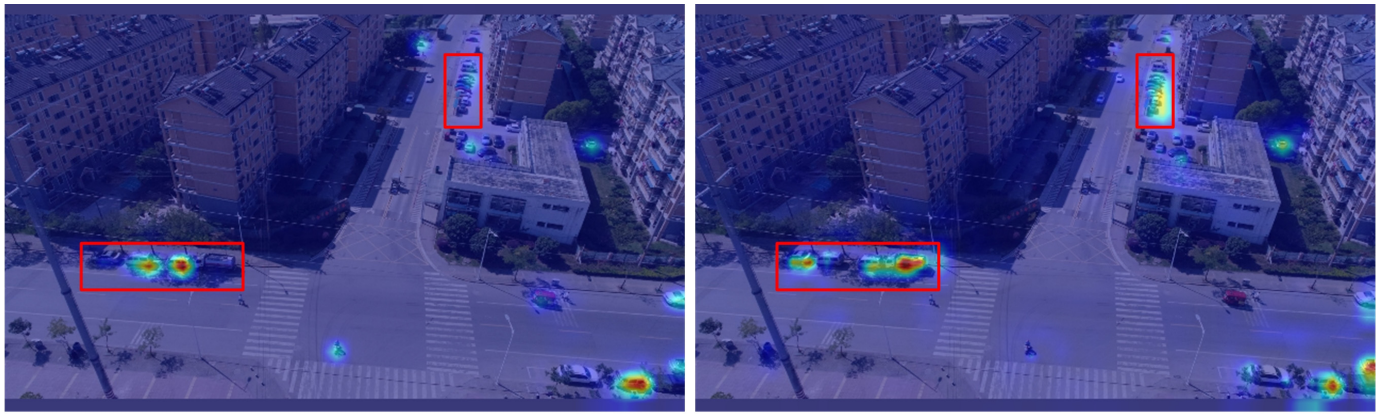


Figure 16. Comparison of heat maps before and after adding PPA modules.

To provide a more intuitive representation of SPDC-YOLO's performance improvements in complex environments, heatmap visualization analysis was performed on the nighttime urban street scene depicted in Figure 15, with results shown in Figure 17. Panel (a) displays the original image, while panels (b) and (c) illustrate heatmaps generated by YOLOv8n and the proposed SPDC-YOLO, respectively. Comparative analysis reveals that the YOLOv8n model primarily focuses on moving vehicles, neglecting stationary vehicles marked by red circles along the roadside and exhibiting relatively low heat intensity. In contrast, SPDC-YOLO's heatmap demonstrates broader coverage, detecting more vehicles with a more uniform and higher heat intensity. This suggests that SPDC-YOLO has enhanced sensitivity in detecting a greater number of targets, particularly in challenging conditions.

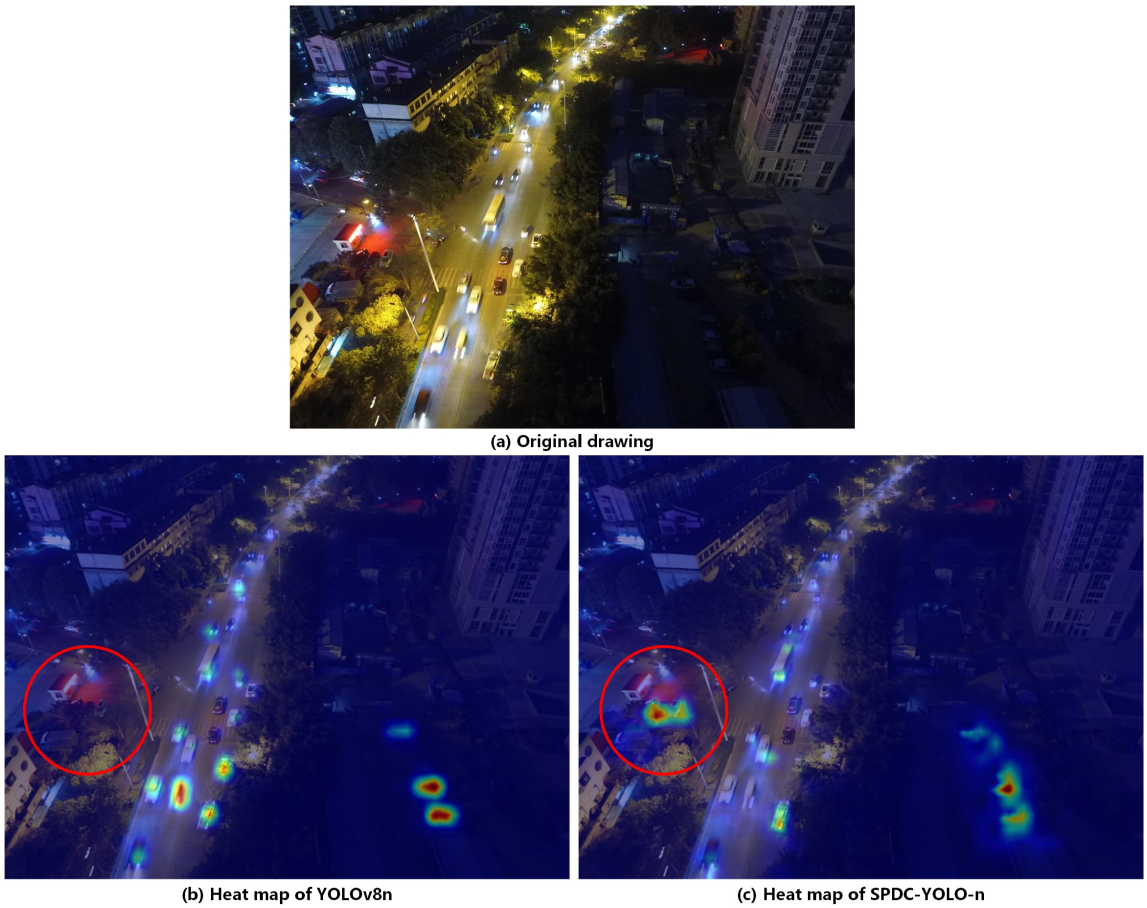


Figure 17. Comparison of heat maps.

4.8. Comparative Experiment of DIOR Dataset

This study also conducts comparative experiments using SPDC-YOLO on the DIOR [59] dataset to further validate its effectiveness and comprehensively assess its performance across various scenarios. The DIOR dataset, a large-scale benchmark for small object detection in optical remote sensing images, consists of 23,463 remote sensing images and 190,288 annotated target instances, covering 20 object categories. These 20 object classes are airplane, airport, baseball field, basketball court, bridge, chimney, dam, expressway service area, expressway toll station, harbor, golf course, ground track field, overpass, ship, stadium, storage tank, tennis court, train station, vehicle, and wind mill. The experimental results are shown in Table 9.

Table 9. Results of comparison experiments on DIOR.

Model	P (%)	R (%)	mAP_{50} (%)	mAP_{50-95} (%)	GFLOPs
GhostNet	87.0	76.9	83.8	59.7	5.2
MobileNetv3	84.0	73.4	79.9	54.9	5.7
YOLOv3	77.6	74.5	76.2	50	18.9
YOLOv5	78.4	79.7	82.5	53.5	7.1
YOLOv8	91.1	80.4	87.9	65.1	8.1
SPDC-YOLO-n	**90.0**	**83.0**	**89.4**	**67.4**	**10.0**

Note: Bold data indicate the model proposed in this paper.

The experimental results in Table 9 demonstrate that SPDC-YOLO excels across all metrics. Its Precision (P) is 90.0%, and Recall (R) is 83.0%, significantly outperforming YOLOv3 and YOLOv5, especially in Recall. While YOLOv8 slightly surpasses SPDC-YOLO in Precision, SPDC-YOLO still outperforms other models in Recall. In terms of mAP, SPDC-YOLO achieves an mAP_{50} of 89.4%, significantly higher than YOLOv3 and MobileNetv3, with an mAP_{50-95} of 67.4%, showcasing its strong performance under more stringent evaluation criteria. Despite having a GFLOPs value of 10.0, higher than YOLOv5, it remains lower than YOLOv3, indicating that SPDC-YOLO effectively balances enhanced performance with computational efficiency, making it suitable for resource-constrained environments. Overall, experimental results on the DIOR dataset suggest that SPDC-YOLO improves performance across various scenarios and tasks.

5. Discussion

In multi-scale drone imagery, object detection poses numerous challenges, including small target sizes, complex backgrounds, and uneven illumination. To address these challenges, SPDC-YOLO employs the novel SPC-FPN for feature fusion and removes the P5 detection head, which is designed for large objects, retaining only the 40×40 and 80×80 detection heads. This method successfully addresses the difficulties involved in small object detection in aerial photography by increasing sensitivity to small targets while lowering the model's parameter count. Furthermore, during the downsampling stage, the CGB module learns joint features of local objects and their surrounding context, improving feature representation by incorporating global contextual information, thus mitigating the issue of sparse target features. In the detection head stage, SPDC-YOLO uses the new Dyhead-DCNv4. Dyhead consists of three core attention mechanisms, enabling dynamic feature weighting and adaptive adjustment. DCNv4 optimizes memory access patterns, reduces redundant operations, and significantly improves processing speed. As illustrated in Figure 13, YOLOv8 failed to detect some small, slightly blurred targets, whereas SPDC-YOLO successfully identified these missed detections, effectively overcoming the challenge of missed detections caused by complex backgrounds in drone imagery.

The statistics shown in Tables 6 and 7 demonstrate that SPDC-YOLO outperforms other conventional models and variations of the YOLO series in terms of accuracy and speed, demonstrating notable advantages in the detection of tiny and medium-sized items. These advantages are reflected not only in higher detection accuracy and lower false positive rates but also in a reduced parameter count and lower computational complexity. Furthermore, in Figure 14, despite the high shooting angle and excessive illumination, which degrade the quality of drone imagery and introduce significant noise and interference, SPDC-YOLO successfully localizes targets. This demonstrates its remarkable robustness and resistance to interference, underscoring its potential for application in complex and challenging environments.

6. Conclusions

This paper proposes a new model, SPDC-YOLO, based on YOLOv8, designed for detecting small objects in multi-scale UAV images. The model achieves fewer parameters while maintaining a high mAP_{50}. It addresses common challenges in small object detection tasks by incorporating several innovations. Specifically, SPDC-YOLO introduces the newly proposed SPC-FPN in the Neck section to fuse multi-scale features, employs the CGB module for downsampling to integrate contextual information, and replaces the original C2f module with the PPA module to enhance feature representation. Additionally, we utilize Dyhead-DCNv4 for both localization and classification. To further improve the bounding box regression process, the EIoU loss function is employed, which considers both the center distance and aspect-ratio difference between the predicted bounding box and the ground truth, thereby improving localization accuracy.

Experimental results on the VisDrone2019 dataset demonstrate that SPDC-YOLO significantly outperforms YOLOv8 and other YOLO variations. Compared to other enhanced YOLO-based models, SPDC-YOLO achieves superior detection accuracy while maintaining a balance between performance and computational efficiency. Furthermore, it consistently improves detection accuracy across all 10 categories of the VisDrone2019 dataset on both the validation and test sets.

Despite its advantages, SPDC-YOLO incurs a slightly higher computational cost than YOLOv8, and no model pruning or knowledge distillation techniques have been applied. Future work could focus on further optimizing the model for real-time deployment, including lightweight network compression techniques such as pruning and knowledge distillation. Additionally, integrating SPDC-YOLO with edge computing frameworks would enhance its applicability in UAV-based real-time detection tasks. Another promising direction is adapting SPDC-YOLO for other remote sensing applications, such as synthetic aperture radar (SAR) imagery, to evaluate its robustness and generalization capabilities in different sensing modalities.

Author Contributions: Conceptualization, J.B.; methodology, J.B. and K.L.; validation, J.B. and X.Z.; investigation, J.B. and G.Z.; writing—original draft preparation, J.B.; writing—review and editing, J.B., K.L., X.Z., G.Z. and T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this article are included. If you have further questions, please contact the first author.

Acknowledgments: Sincere thanks to the editors, reviewers, and all staff of the journal. It is your professional spirit and unremitting efforts that make every work shine brightly.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zheng, Z.; Yuan, J.; Yao, W.; Kwan, P.; Yao, H.; Liu, Q.; Guo, L. Fusion of UAV-Acquired Visible Images and Multispectral Data by Applying Machine-Learning Methods in Crop Classification. *Agronomy* **2024**, *14*, 2670. [CrossRef]
2. Zhang, Z.; Yao, F.; Li, J. Dynamic penetration test based on YOLOv5. In Proceedings of the 2022 3rd International Conference on Geology, Mapping and Remote Sensing (ICGMRS), Zhoushan, China, 22–24 April 2022; pp. 102–105.
3. Sato, S.; Anezaki, T. Autonomous flight drone for infrastructure (transmission line) inspection (2). In Proceedings of the 2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Okinawa, Japan, 24–26 November 2017; pp. 294–296.
4. Menkhoff, T.; Tan, E.K.B.; Ning, K.S.; Hup, T.G.; Pan, G. Tapping drone technology to acquire 21st century skills: A smart city approach. In Proceedings of the 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), San Francisco, CA, USA, 4–8 August 2017; pp. 1–4.
5. Ryoo, D.W.; Lee, M.S.; Lim, C.D. Design of a Drone-Based Real-Time Service System for Facility Inspection. In Proceedings of the 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 11–13 October 2023; pp. 1566–1568.
6. Sun, X.; Zhang, W. Implementation of Target Tracking System Based on Small Drone. In Proceedings of the 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 20–22 December 2019; pp. 1863–1866.
7. Wang, Z.; Xia, F.; Zhang, C. FD_YOLOX: An improved YOLOX object detection algorithm based on dilated convolution. In Proceedings of the 2023 IEEE 18th Conference on Industrial Electronics and Applications (ICIEA), Ningbo, China, 18–22 August 2023; pp. 1263–1268.
8. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
9. Tianyu, Z.; Zhenjiang, M.; Jianhu, Z. Combining CNN with Hand-Crafted Features for Image Classification. In Proceedings of the 2018 14th IEEE International Conference on Signal Processing (ICSP), Beijing, China, 12–16 August 2018; pp. 554–557.
10. Long, P.; Hu, Y. Multiple Image Augmentations for Enhanced YOLO-based Traffic Sign Detection. In Proceedings of the 2023 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), Chengdu, China, 3–5 November 2023; pp. 132–137.
11. Pan, W.; Huan, W.; Xu, L. Improving High-Voltage Line Obstacle Detection with Multi-Scale Feature Fusion in YOLO Algorithm. In Proceedings of the 2024 6th International Conference on Electronics and Communication, Network and Computer Technology (ECNCT), Guangzhou, China, 19–21 July 2024; pp. 270–273.
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
13. Zong, H.; Pu, H.; Zhang, H.; Wang, X.; Zhong, Z.; Jiao, Z. Small object detection in UAV image based on slicing aided module. In Proceedings of the 2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 29–31 July 2022; pp. 366–370.
14. Wu, T.; Tang, S.; Zhang, R.; Cao, J.; Zhang, Y. CGNet: A Light-Weight Context Guided Network for Semantic Segmentation. *IEEE Trans. Image Process.* **2021**, *30*, 1169–1179. [CrossRef] [PubMed]
15. Xu, S.; Zheng, S.; Xu, W.; Xu, R.; Wang, C.; Zhang, J.; Teng, X.; Li, A.; Guo, L. HCF-Net: Hierarchical Context Fusion Network for Infrared Small Object Detection. *arXiv* **2024**, arXiv:2403.10778.
16. Tang, F.; Huang, Q.; Wang, J.; Hou, X.; Su, J.; Liu, J. DuAT: Dual-Aggregation Transformer Network for Medical Image Segmentation. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision, Shenzhen, China, 14–17 October 2022.
17. Xiong, Y.; Li, Z.; Chen, Y.; Wang, F.; Zhu, X.; Luo, J.; Wang, W.; Lu, T.; Li, H.; Qiao, Y.; et al. Efficient Deformable ConvNets: Rethinking Dynamic and Sparse Operator for Vision Applications. *arXiv* **2024**, arXiv:2401.06197.
18. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; Zhang, L. Dynamic Head: Unifying Object Detection Heads with Attentions. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 20–25 June 2021; pp. 7369–7378.
19. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y.; et al. VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 213–226.
20. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
21. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083.

22. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
23. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *arXiv* **2017**, arXiv:1703.06870.
24. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into High Quality Object Detection. *arXiv* **2017**, arXiv:1712.00726.
25. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
26. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv* **2016**, arXiv:1608.06993.
27. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
28. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
29. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
30. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
31. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
32. 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* **2022**, arXiv:2207.02696.
33. Wang, C.-Y.; Yeh, I.H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
34. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
35. Luo, J.; Liu, Z.; Wang, Y.; Tang, A.; Zuo, H.; Han, P. Efficient Small Object Detection You Only Look Once: A Small Object Detection Algorithm for Aerial Images. *Sensors* **2024**, *24*, 7067. [CrossRef]
36. Zhang, S.; Yang, X.; Geng, C.; Li, X. A Reparameterization Feature Redundancy Extract Network for Unmanned Aerial Vehicles Detection. *Remote Sens.* **2024**, *16*, 4226. [CrossRef]
37. Liu, B.; Mo, P.; Wang, S.; Cui, Y.; Wu, Z. A Refined and Efficient CNN Algorithm for Remote Sensing Object Detection. *Sensors* **2024**, *24*, 7166. [CrossRef]
38. Wang, C.; He, W.; Nie, Y.; Guo, J.; Liu, C.; Han, K.; Wang, Y. Gold-YOLO: Efficient Object Detector via Gather-and-Distribute Mechanism. *arXiv* **2023**, arXiv:2309.11331.
39. Wang, J.; Sun, Y.; Lin, Y.; Zhang, K. Lightweight Substation Equipment Defect Detection Algorithm for Small Targets. *Sensors* **2024**, *24*, 5914. [CrossRef]
40. Liu, M.; Wang, X.; Zhou, A.; Fu, X.; Ma, Y.; Piao, C. UAV-YOLO: Small Object Detection on Unmanned Aerial Vehicle Perspective. *Sensors* **2020**, *20*, 2238. [CrossRef] [PubMed]
41. Verma, T.; Singh, J.; Bhartari, Y.; Jarwal, R.; Singh, S.; Singh, S. SOAR: Advancements in Small Body Object Detection for Aerial Imagery Using State Space Models and Programmable Gradients. *arXiv* **2024**, arXiv:2405.01699.
42. Fonod, R.; Cho, H.; Yeo, H.; Geroliminis, N. Advanced computer vision for extracting georeferenced vehicle trajectories from drone imagery. *arXiv* **2024**, arXiv:2411.02136.
43. Tang, S.; Zhang, S.; Fang, Y. HIC-YOLOv5: Improved YOLOv5 For Small Object Detection. *arXiv* **2023**, arXiv:2309.16393.
44. Zhang, Y.; Ye, M.; Zhu, G.; Liu, Y.; Guo, P.; Yan, J. FFCA-YOLO for Small Object Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5611215. [CrossRef]
45. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Virtual, 11–17 October 2021; pp. 2778–2788.
46. Wang, M.; Yang, W.; Wang, L.; Chen, D.; Wei, F.; KeZiErBieKe, H.; Liao, Y. FE-YOLOv5: Feature enhancement network based on YOLOv5 for small object detection. *J. Vis. Commun. Image Represent.* **2023**, *90*, 103752. [CrossRef]
47. Shen, L.; Lang, B.; Song, Z. CA-YOLO: Model Optimization for Remote Sensing Image Object Detection. *IEEE Access* **2023**, *11*, 64769–64781. [CrossRef]
48. Zhang, Y.-F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]
49. Ping, H.; Jie, L.; Huahong, Z. DAID-YOLO: Small Object Detection Algorithm for Drone Aerial Images. In Proceedings of the 2024 9th International Conference on Signal and Image Processing (ICSIP), Nanjing, China, 12–14 July 2024; pp. 591–595.
50. Chen, H.; Wang, D.; Fang, J.; Li, Y.; Xu, S.; Xu, Z. LW-YOLOv8: An Lightweight Object Detection Algorithm for UAV Aerial Imagery. In Proceedings of the 2024 6th International Conference on Natural Language Processing (ICNLP), Xi'an, China, 22–24 March 2024; pp. 446–450.

51. Chen, L.; Liu, C.; Li, W.; Xu, Q.; Deng, H. DTSSNet: Dynamic Training Sample Selection Network for UAV Object Detection. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5902516. [CrossRef]
52. Xiong, G.; Qi, J.; Wang, M.; Wu, C.; Sun, H. GCGE-YOLO: Improved YOLOv5s Algorithm for Object Detection in UAV Images. In Proceedings of the 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 24–26 July 2023; pp. 7723–7728.
53. Wu, Q.; Li, Y.; Huang, W.; Chen, Q.; Wu, Y. C3TB-YOLOv5: Integrated YOLOv5 with transformer for object detection in high-resolution remote sensing images. *Int. J. Remote Sens.* **2024**, *45*, 2622–2650. [CrossRef]
54. Yue, M.; Zhang, L.; Huang, J.; Zhang, H. Lightweight and Efficient Tiny-Object Detection Based on Improved YOLOv8n for UAV Aerial Images. *Drones* **2024**, *8*, 276. [CrossRef]
55. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones* **2023**, *7*, 304. [CrossRef]
56. Wang, G.; Chen, Y.; An, P.; Hong, H.; Hu, J.; Huang, T. UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. *Sensors* **2023**, *23*, 7190. [CrossRef]
57. Zhang, Z. Drone-YOLO: An Efficient Neural Network Method for Target Detection in Drone Images. *Drones* **2023**, *7*, 526. [CrossRef]
58. Li, Y.; Li, Q.; Pan, J.; Zhou, Y.; Zhu, H.; Wei, H.; Liu, C. SOD-YOLO: Small-Object-Detection Algorithm Based on Improved YOLOv8 for UAV Images. *Remote Sens.* **2024**, *16*, 3057. [CrossRef]
59. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J.J.A. Object Detection in Optical Remote Sensing Images: A Survey and A New Benchmark. *arXiv* **2019**, arXiv:1909.00133. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.