remote sensing

MDPI

Article

Detection Based on Semantics and a Detail Infusion Feature Pyramid Network and a Coordinate Adaptive Spatial Feature Fusion Mechanism Remote Sensing Small Object Detector

Shilong Zhou [1,2] and Haijin Zhou [2,*]

[1] Institutes of Physical Science and Information Technology, Anhui University, Hefei 230601, China; q22301373@stu.ahu.edu.cn
[2] Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China
* Correspondence: hjzhou@aiofm.ac.cn

Abstract: In response to the challenges of remote sensing imagery, such as unmanned aerial vehicle (UAV) aerial imagery, including differences in target dimensions, the dominance of small targets, and dense clutter and occlusion in complex environments, this paper optimizes the YOLOv8n model and proposes an innovative small-object-detection model called DDSC-YOLO. First, a DualC2f structure is introduced to improve the feature-extraction capabilities of the model. This structure uses dual-convolutions and group convolution techniques to effectively address the issues of cross-channel communication and preserving information in the original input feature mappings. Next, a new attention mechanism, DCNv3LKA, was developed. This mechanism uses adaptive and fine-grained information-extraction methods to simulate receptive fields similar to self-attention, allowing adaptation to a wide range of target size variations. To address the problem of false and missed detection of small targets in aerial photography, we designed a Semantics and Detail Infusion Feature Pyramid Network (SDI-FPN) and added a dedicated detection scale specifically for small targets, effectively mitigating the loss of contextual information in the model. In addition, the coordinate adaptive spatial feature fusion (CASFF) mechanism is used to optimize the original detection head, effectively overcoming multi-scale information conflicts while significantly improving small target localization accuracy and long-range dependency perception. Testing on the VisDrone2019 dataset shows that the DDSC-YOLO model improves the mAP0.5 by 9.3% over YOLOv8n, and its performance on the SSDD and RSOD datasets also confirms its superior generalization capabilities. These results confirm the effectiveness and significant progress of our novel approach to small target detection.

Keywords: UAV; object detection; YOLOv8n; attention mechanism; feature fusion

check for updates

Citation: Zhou, S.; Zhou, H. Detection Based on Semantics and a Detail Infusion Feature Pyramid Network and a Coordinate Adaptive Spatial Feature Fusion Mechanism Remote Sensing Small Object Detector. *Remote Sens.* **2024**, *16*, 2416. https://doi.org/10.3390/rs16132416

Academic Editors: Jukka Heikkonen, Fahimeh Farahnakian, Pouya Jafarzadeh and Farshad Farahnakian

Received: 4 June 2024
Revised: 27 June 2024
Accepted: 27 June 2024
Published: 1 July 2024

Copyright: © 2024 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

With the advancement of UAV technology, UAVs have become an ideal platform for multi-target detection due to their superior payload capacity, ease of operation, and flexible mobility. UAVs play an important role in industrial risk monitoring [1], terrain exploration [2], crowd [3] and vehicle safety monitoring [4], and air and water quality monitoring [5], especially in applications where pollution sources are tracked. In these applications, the UAV's target acquisition and tracking technology is particularly important. However, due to the high altitude of remote sensing platforms such as drones, images often contain a large number of small objects [6], as shown in Figure 1. The detection of these small targets faces several challenges: low resolution, limited feature information, and susceptibility to clutter and occlusion in complex environments. These factors make small target detection a significant challenge.



Figure 1. Remote sensing images of various small objects.

With the rapid advancement of deep learning technology, the capabilities of object-detection models using this technology have improved significantly. Currently, mainstream object-detection algorithms can be categorized into three types: two-stage algorithms, one-stage algorithms, and vision transformers. The two-stage object-detection approach first generates candidate regions, followed by classification and regression analysis on these regions. Typical methods include the R-CNN series, such as Faster R-CNN [7], Mask R-CNN [8], Cascade R-CNN [9], R-FCN [10], etc. One-step object-detection methods consider object detection as a whole regression problem, and directly output object category and location information. The commonly used methods are some anchor-based detection methods. These include the You Only Look Once (YOLO) series [11–14], Single-Shot MultiBox Detector (SSD) [15], RetinaNet [16], etc. These algorithms determine the presence, location, and size of the target object by dividing the image into anchors and performing classification and regression processing on each anchor. Vision transformers divide the image into sequential blocks and process these sequences using a self-attention mechanism for object detection. Algorithms in this class include Swin transformer [17], detection transformer (DETR) [18], MViT [19], and its successors.

These algorithms have obvious differences in their design principles and architectures, and each has different advantages and disadvantages in terms of speed, accuracy, efficiency, and model size. For example, the two-stage algorithm has high accuracy, especially excellent performance in detecting small objects, and its detection accuracy can be further improved by methods such as multi-scale fusion and model distillation. However, the main drawback of this algorithm is the slow processing speed, and it needs to generate a large number of candidate regions, so this increases the amount of computation and time complexity. In contrast, vision transformers provide an end-to-end object-detection scheme that simplifies the overall architecture, but due to their high computational requirements, such algorithms are not suitable for use on embedded devices with limited resources. On the other hand, the one-step algorithm does not need to generate candidate regions, thus simplifying the detection process, and is very effective in scenes that require immediate

responses due to its fast and real-time performance, although it may be slightly insufficient in the accuracy of detecting small objects.

Balancing speed, accuracy, and model size is critical for UAV missions, so single-stage object-detection algorithms such as YOLO are preferred for their excellent real-time performance and high accuracy. Although the YOLO algorithm is mainly used to detect full-size objects, its performance can be limited when dealing with scenes with special dimensions. Various strategies have been developed by researchers to address the challenge of small object detection, with successful applications of YOLO series models. For example, Ding et al. [20] extended the YOLOv4 model by integrating the coordinate attention mechanism (CA) [21] into MobileNetV2 [22], replacing the original backbone, and using depth-separable convolution and the squeeze-and-excitation (SE) module [23]. These modifications improved the learning ability of the convolutional neural network, maintaining the accuracy of feature extraction while reducing complexity, resulting in better overall performance. However, YOLOv4 still contains some redundant structures. Similarly, Wu et al. [24] introduced a region proposal network and improved YOLOv5's small object detection capabilities by incorporating the multi-scale anchor mechanism from Faster R-CNN. This adaptation enabled YOLOv5 to exhibit high adaptability to images of different sizes. Zhu et al. [25] integrated transformer prediction heads (TPHs) into the YOLOv5 structure and proposed the TPH-YOLOv5 model. This model introduced an additional prediction head for detecting objects of different scales and used the self-attention mechanism of TPHs. It also incorporated the Convolutional Block Attention Module (CBAM) [26] into YOLOv5. These additions optimized the model's ability to discriminate cross-scale objects and effectively identify key regions in dense object scenes, although they introduced additional computational and parameter complexity, which affected detection efficiency. MS-YOLOv7, proposed by Zhao et al. [27], built on YOLOv7 by increasing the number of detection heads from three to four to better extract features at different scales. They also incorporated the Swin transformer, Window Multi-Head Self-Attention (W-MSA), Shifted Window Multi-Head Self-Attention (SW-MSA) [28], and CBAM attention mechanisms to improve the neck features of the network. In addition, the application of the soft NMS method improved the performance of NMS in densely distributed object detection. Lin et al. [29] proposed YOLOv8n-SLIM-CA, which uses mosaic data augmentation to generate many small target instances, thereby increasing the overall robustness of the network. The backbone network was augmented with the coordinate attention mechanism to improve focus on key regions in complex backgrounds and suppress interference from irrelevant features. In the neck network, they adopted a slim neck structure and added a small object detection layer to improve the model's ability to detect objects in complex backgrounds and small targets. Zhang et al.'s Drone-YOLO [30] used a three-layer path aggregation feature pyramid network (PAFPN) structure with a special small-object detection head. The sandwich fusion module was used to optimize semantic features. The detection head obtained feature vectors with high spatial resolution and accurate semantic information, improving the overall detection performance. The RepVGG reparameterized convolution module improved the model's ability to understand features of different scales, thereby improving the detection accuracy for small objects. Finally, Wang et al. [31] developed UAV-YOLOv8, which introduced a BiFormer attention mechanism to optimize the backbone network and improve the model's focus on critical information. They also introduced the Focal FasterNet Block (FFNB) for feature processing. This model added two new detection scales that effectively fuse shallow and deep feature information, significantly improving detection performance and reducing the miss rate of small objects.

These contributions collectively demonstrate the continuous progress in adapting YOLO models for improving small object detection. Due to the complexity of backgrounds, significant differences in spatial resolution, and the ubiquitous presence of irregularly arranged small objects, small objects need to be detected more effectively. To address these challenges and optimize the detection accuracy for small objects, this paper proposes an improved neural network model based on the YOLOv8n architecture. Our approach

enhances the C2f module, integrates advanced attention mechanisms, introduces a novel feature pyramid network, and strengthens the detection head to improve the feature representation of small object detection, thereby improving the overall detection performance. The proposed model structure and corresponding experimental results demonstrate the effectiveness of the model in detecting complex small objects. Comparative analysis with existing models indicates that this model exhibits superior performance in drone target detection.

The key contributions of this work encompass the following:

- Introducing the DualC2f module, which integrates 1 × 1 and 3 × 3 dual-convolution kernels to simultaneously process the identical input feature map channels. The use of group convolution techniques efficiently arranges the convolution filters, solving the problems of inter-channel communication and information preservation within the original input feature map, which significantly improves the accuracy of the model in identifying small targets.
- The proposed DCNv3LKA attention mechanism combines large convolution kernels and deformable convolutions to simulate receptive fields similar to those of self-attention. This approach adapts to the wide variations in targets while avoiding the high computational costs associated with traditional self-attention mechanisms.
- To mitigate the common problem of the misidentification and omission of small targets in aerial imagery, we developed the SDI-FPN feature pyramid network. This network architecture integrates both semantic and detailed information to improve target detection accuracy. Inspired by BiFPN, this approach achieves more efficient feature fusion by fully integrating shallow and deep features. By inserting four CBS blocks with a step of 1 and a kernel size of 1 between the spine and neck components, the storage and transmission of image feature information is improved. Furthermore, we introduce a novel detection scale optimized for the detection of small objects, which effectively mitigates the loss of contextual information within the model.
- We propose a spatial feature fusion method with coordinate adaptivity for optimizing the original detection head. This method first integrates features from different levels and adjusts them to the same resolution using weights, effectively filtering out spatial information conflicts. Then, it enhances the spatial expression of features by incorporating the coordinate attention (CA) mechanism. This mechanism assists the model in learning regions in the feature map that are relevant to the target position, thereby improving the model's localization accuracy. This optimization strategy enhances the model's ability to represent features of small-sized targets, especially in complex backgrounds, thereby increasing detection accuracy.

This paper is divided into five sections. In Section 2, we review related work relevant to our research. Section 3 provides a detailed explanation of the improvements and implementation details. Section 4 provides an overview of the experimental process, including the configurations and specific experimental methods, followed by a comparative analysis of the experimental results and a visualization of the results. Finally, Section 5 summarizes the current work and suggests future directions.

2. Related Work

2.1. Convolutional Neural Networks

Convolutional neural networks (CNNs) have emerged as the preferred feature encoders for numerous vision tasks, attributed to their ability to handle translation and variability and to exploit spatial locality. This architecture excels in extracting critical features from images, where a broad receptive field proves crucial for detecting small objects. Initially, CNNs utilized large kernel convolutions to achieve an extensive receptive field, facilitating the fusion of features over greater distances. AlexNet [32], the pioneering deep CNN, demonstrated the profound capabilities of deep CNNs in feature learning and generalization, employing a combination of deep convolutional and pooling layers, along with the introduction of the ReLU activation function and dropout technique. However, the in-

crease in network depth led to a rise in the number of model parameters and computational resource consumption. To mitigate this, VGG [33] introduced the use of multiple layers with smaller kernels (e.g., 3×3) instead of large kernels, effectively reducing parameters while preserving the receptive field. Despite its advantages, this approach risked information loss and training instability. Addressing these challenges, ResNet [33] incorporated cross-layer connections, significantly reducing the issues of gradient vanishing and exploding with deeper networks and preserving information integrity. MobileNetV2 further explored the role of large kernels through the design of the Linear Bottleneck and Inverted Residual structures, enhancing network representation by substituting large-weight kernels for small-weight ones. This achieved an extensive effective receptive field, mimicking human object shape recognition. In addition, Deformable Large Kernel Attention (D-LKA) [34] adopts the simplified attention mechanism of a large convolution kernel to enhance the adaptation of the model to geometric changes by introducing a learnable offset, which is similar to the operation in the receptive field of self-attention. Volumetric context can be fully understood while avoiding computational overhead. In summary, the utilization of large convolution kernels not only underscores their pivotal role in enhancing CNN performance, but also emphasizes the criticality of long-distance feature dependence for effective feature aggregation, particularly in small object detection tasks. This aspect provides robust support for addressing complex vision tasks.

2.2. Multi-Scale Features for Object Detection

Constructing an efficient feature pyramid network (FPN) [35] is crucial for enhancing the efficiency of multi-scale object detection. Deep networks, with their extensive receptive fields and robust semantic information representation capabilities, can effectively encode complex semantic data; however, their feature maps are of low resolution, leading to weaker spatial geometric information representation. In contrast, shallow networks, though limited by smaller receptive fields, possess higher resolution, which better captures the spatial geometric details, but lack semantic representation. By integrating multi-scale features from both deep and shallow layers, comprehensive information extraction is achieved, encompassing both global and local details. This fusion strategy significantly enhances the network's adaptability to targets of various scales, thereby improving detection precision and robustness. The FPN utilizes a top-down architecture with lateral connections to effectively incorporate semantic information from lower layers into higher-level features, facilitating information exchange across different scale feature maps and reducing semantic information loss caused by repeated convolution and pooling operations, which substantially increases the accuracy of object detection. PANet [36] extends the FPN by refining cross-layer feature fusion and semantic information transfer. It introduces bidirectional top-to-bottom and bottom-to-top paths to facilitate effective feature aggregation across layers. It also incorporates an adaptive feature pooling module to dynamically adjust receptive fields based on the target scale and shape, improving the model's adaptability to targets of different scales. The adaptive spatial feature fusion (ASFF) [37] strategy adjusts the fusion intensity dynamically based on feature weights at each position and modifies the fusion weights according to the target's location and size, enhancing model robustness in complex scenarios and against occlusions. BiFPN [38], a bidirectional feature pyramid network, allows features to merge in both top-down and bottom-up directions, effectively integrating features of different scales and adding weights to each input feature to optimize the fusion process, adapting to varying input resolutions and target sizes. However, during the layer-by-layer feature-extraction and spatial-transformation processes, these fundamental building blocks may still lose substantial information, particularly in detecting small targets.

3. Methods

The YOLOv8 detector consists of three primary components, the backbone, the neck, and the detector head, which are available in five variants, YOLOv8n, YOLOv8s, YOLOv8m,

YOLOv8l, and YOLOv8x, each varying in channel width, depth, and the maximum number of channels. All input images are resized to a uniform size of 640 × 640. The backbone network extracts feature maps from the input images through repeated convolutions, creating three layers of feature maps (80 × 80, 40 × 40, and 20 × 20). The backbone network uses Cross-Stage Partial Darknet (CSPdarknet) [12] for feature extraction, enriching gradient information by replacing the original Cross-Stage Partial (CSP) module with the C2f module. In addition, a Spatial Pyramid Pooling Fast (SPPF) module is used at the end of the backbone to pool the input feature maps to a fixed size to accommodate outputs of different sizes. The neck uses a PAN-FPN structure, forming a top-down and bottom-up network architecture. The detection head adopts a decoupled head structure, utilizing two independent branches for object classification and bounding box regression prediction, with distinct loss functions for each task: Binary Cross-Entropy (BCE) loss for class loss and Distribution Focal Loss (DFL Loss) [39] and Complete IoU Loss (CIOU Loss) [40] for regression loss. To address the detection of small and multi-scale objects within the YOLOv8 network, we propose the DDSC-YOLO network model, which is based on the YOLOv8n baseline combined with a multi-scale attention module, specifically for the detection of small targets in drone aerial images. The focus is on improving the spatial-feature-extraction capability for target detection and building an efficient multiscale feature fusion network to overcome the challenges posed by background clutter and large target variations in UAV aerial imagery. The detailed network structure is shown in Figure 2.

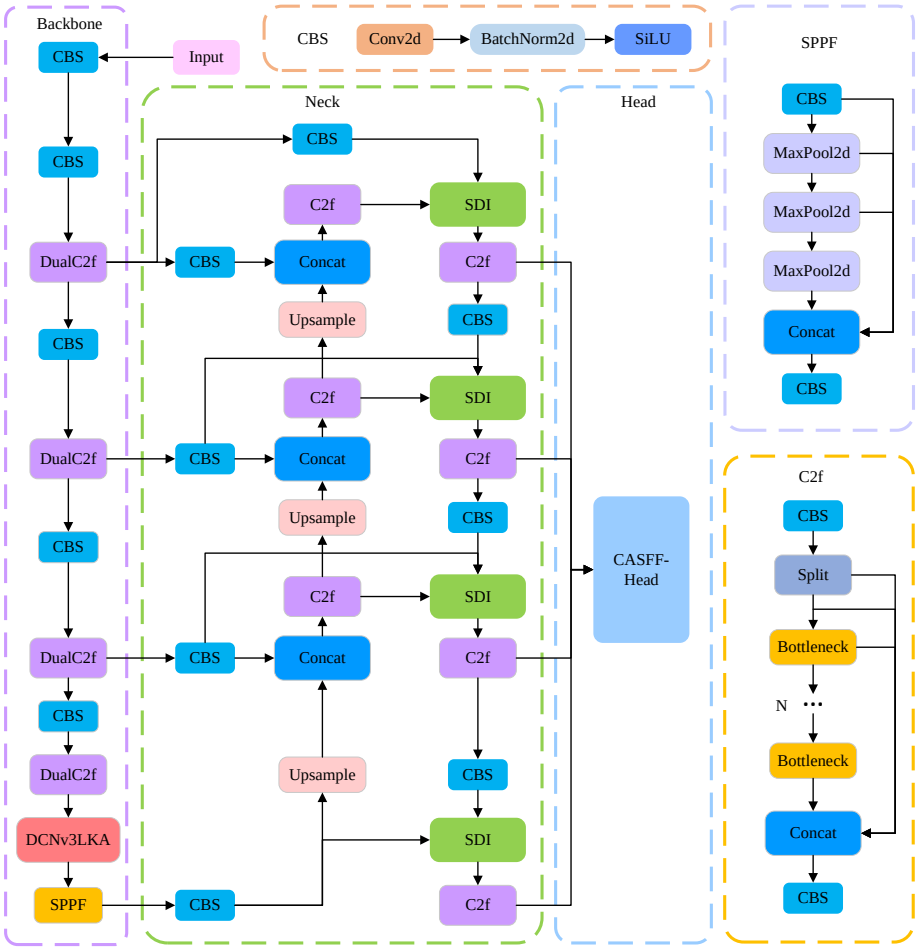


Figure 2. The overall structure of DDSC-YOLO.

3.1. DualC2f

In YOLOv8, the C2f module is employed to integrate low-level and high-level feature maps, facilitating the capture of rich gradient information flow. However, with the

proliferation of layers in convolutional neural networks, the semantic information within feature maps tends to be progressively extracted and aggregated, resulting in deep feature maps containing redundant information. Additionally, due to the weight-sharing mechanism of convolutional layers, convolution kernel parameters are shared across different positions of deep feature maps, further exacerbating redundancy. The bottleneck module, which consists of numerous complex convolutions, greatly increases the parameter size and computational complexity. To solve this problem, the DualC2f module is proposed to replace the bottleneck module with the newly designed DualBottleneck module. This structural design not only maintains the network depth and representational capability, but also reduces the redundancy of feature map information, as shown in Figure 3.

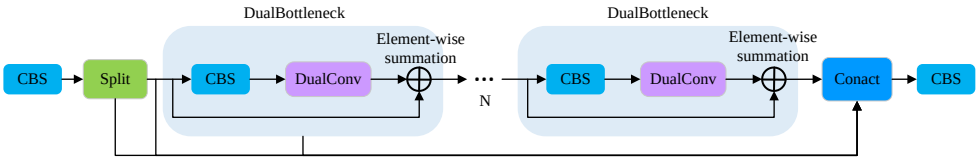


Figure 3. DualC2f module detail.

The DualConv [41] module optimizes the convolutional operations by employing group convolution and heterogeneous convolution techniques to process input feature map channels efficiently. Within this module, some kernels perform both 1×1 and 3×3 convolution operations, while others solely execute 1×1 convolution. The structure is shown in Figure 4, where M represents the number of input channels (i.e., the depth of the input feature map), N denotes the number of convolution filters and output channels (i.e., the depth of the output feature map), and G signifies the number of groups in group convolution and dual-convolution. The N convolution filters are partitioned into G groups, with each group processing the entire input feature map. Within these groups, M/G input feature map channels undergo simultaneous 1×1 and 3×3 convolution operations, while the remaining $(M - M/G)$ input channels are processed separately using 1×1 convolution kernels.

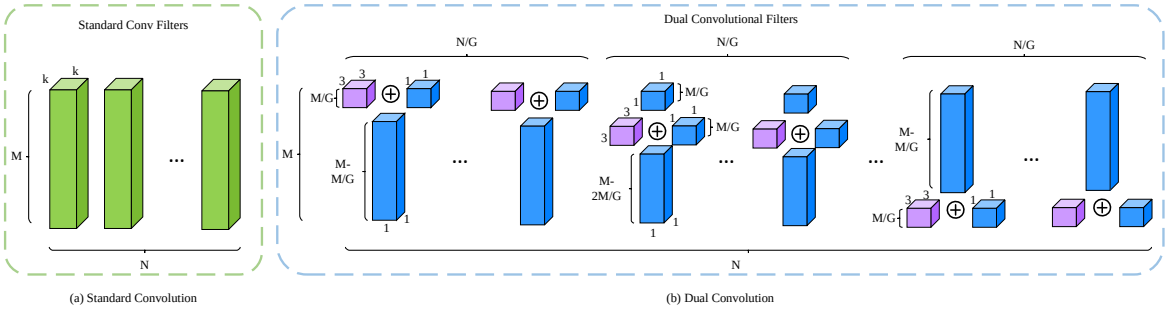


Figure 4. Comparison of the standard convolution kernel with the dual-convolution: (**a**) Standard convolution. (**b**) Dual-convolution.

The design of DualConv maintains the original information of the input feature map by employing the group convolution strategy. This approach fosters improved information sharing between convolutional layers and facilitates maximum cross-channel communication with M 1×1 convolutions, thereby achieving efficient information flow and integration across different feature map channels. Therefore, replacing the bottleneck structure in C2f with the above DualBottleneck enriches the gradient flow representation, improves the feature extraction ability, reduces the diversity of false detections and missed detections between network learning, and is more suitable for various small target scenarios.

3.2. DCNv3LKA Attention Mechanism

The target size in UAV aerial photography is small and highly variable, creating challenges for traditional convolution methods to extract features efficiently. As network complexity increases during training, critical inter-layer information is often lost, significantly affecting object detection accuracy. To tackle this issue, we integrated a DCNv3LKA attention mechanism, based on DCNv3 [42] and Deformable Large Kernel Attention, into the backbone network. This mechanism enhances the focus on crucial small-target information within the input features, suppresses non-essential details, and reduces the impact of background noise. The attention mechanism, which combines a large convolution kernel with deformable convolution, is illustrated in Figure 5. The DCNv3LKA module can be expressed as follows:

$$\text{Attn} = \text{Conv}_{1\times1}(\text{Dcnv3-DW-D-Conv}(\text{Dcnv3-DW-Conv}(S'))), \tag{1}$$

$$\text{Out} = \text{Conv}_{1\times1}(\text{Attn} \otimes S') \oplus S, \tag{2}$$

where the input feature is denoted by $S \in \mathbb{R}^{C\times H\times W}$ and $S' = \text{GELU}(\text{Conv}(S))$. The attention component, represented as an attention map in $\in \mathbb{R}^{C\times H\times W}$, assigns a relative importance to each corresponding feature. The operator $\otimes$ signifies matrix multiplication, while the $\oplus$ operator denotes elementwise summation.

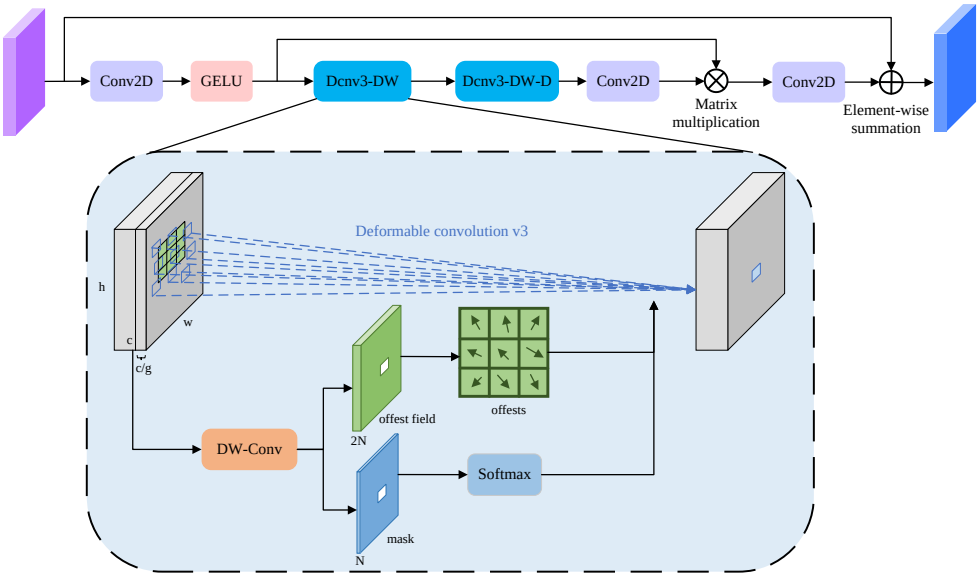


Figure 5. Structure of DCNv3LKA.

Deformable convolution v3 enables the sampling grid to deform adaptively using offsets learned from the features themselves, resulting in a flexible convolution kernel. Initially, the input feature maps are divided into g groups, with convolution performed on each group. This multi-group mechanism enhances the expressiveness of deformable convolution and achieves shared convolution weights w_g. Multiple sets of convolution kernel offsets and modulation factors are computed, with softmax normalization applied to the modulation scalars to stabilize the network training process. The output feature map is then calculated based on these sampling points and offsets. The formulation for deformable convolution v3 is as follows:

$$y(p_0) = \sum_{g=1}^{G}\sum_{k=1}^{K} w_g m_{gk} x_g(p_0 + p_k + \Delta p_{gk}), \tag{3}$$

where G is the total number of aggregation groups and K is the total number of sampling points. For the g-th group, $w_g \in \mathbb{R}^{C \times C'}$ represents the location-irrelevant projection weights of the group, where $C' = C/G$ represents the group dimension. $m_{gk} \in \mathbb{R}$ denotes the modulation scalar of the k-th sampling point in the g-th group, normalized by the softmax function along the k dimension. $x_g \in \mathbb{R}^{C' \times H \times W}$ represents the sliced input feature map. Δp_{gk} is the offset corresponding to the k-th grid sampling location p_k in the g-th group.

DCNv3LKA overcomes the limitations of traditional convolution concerning long-range dependencies and adaptive spatial aggregation, improving the network's ability to adapt to object deformation. This makes it more suitable for detecting small targets in complex natural backgrounds and enhances the model's accuracy in target detection.

3.3. Semantics and Detail Infusion Feature Pyramid Network

The original YOLOv8 network utilizes a path aggregation feature pyramid network (PAN-FPN) as the neck network. By bi-directionally aggregating features from both bottom-up and top-down pathways, it promotes the fusion of detailed and abstract information, effectively shortening the information path between them. During the upsampling and downsampling processes, features of the same dimension are stacked together to preserve information about small objects. However, for small targets with pixel-level resolution, due to insufficient attention to large-scale feature mappings, the demands for positional information are not fully met, reducing the quality of detection. Moreover, the reuse rate of features is low, and some information is lost after the long paths of upsampling and downsampling. Therefore, to enhance the performance in detecting low-pixel small targets and multidimensional feature fusion, we propose the SDI-FPN feature pyramid network.

First, we introduce four CBS modules between the backbone and the neck, with a step size and convolution kernel size of 1. These modules serve as containers for storing feature information from the backbone, aggregating both shallow feature maps (which have low resolution, but weak semantic information) and deep feature maps (which have high resolution, but rich semantic information). Feature information is transmitted along a designated path, enabling strong positional features from lower layers to be propagated upwards. These operations further enhance the expression capability of multi-scale features. Based on this, using these four containers as new inputs for backbone feature information in subsequent networks, we propose a new detection scale for small objects. This effectively addresses the problem of the loss of context information in the model, and thus improves the performance of small target detection with a low pixel size.

Furthermore, in the feature fusion process, due to different resolutions, rescaling is required, so the SDI mechanism is introduced to improve multidimensional feature fusion. The structure is shown in Figure 6.

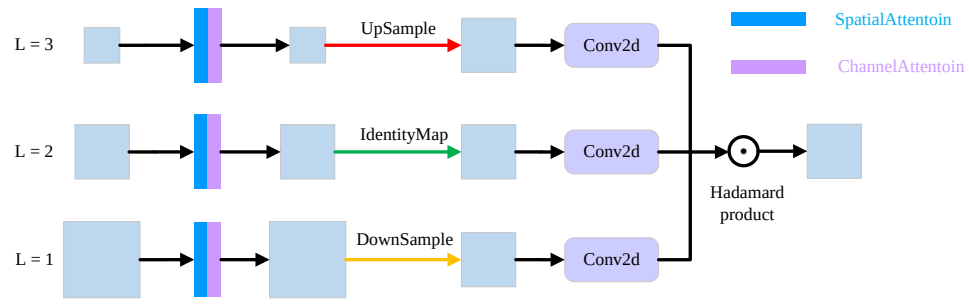


Figure 6. Structure of SDI.

For feature maps of various sizes, we begin by applying spatial and channel attention mechanisms to each level i feature f_i^0, where $1 \leq i \leq 3$, and we use CBAM to implement

the spatial and temporal attention. This process allows the features to incorporate both local spatial information and global channel information, expressed as:

$$f_i^1 = \phi_i(f_i^0), \tag{4}$$

where f_i^1 represents the processed feature map at the i-th level, and the calculation of the spatial and channel attentions at level i is denoted by ϕ_i. Subsequently, we apply a 1×1 convolution to adjust the channels of f_i^1 to c, resulting in the feature map $f_i^2 \in \mathbb{R}^{H_i \times W_i \times c}$, where H_i, W_i, and c are the height, width, and channels of f_i^2, respectively. We then resize the feature maps at each level j to match the resolution of f_i^2, as formulated below:

$$f_{ij}^3 = \begin{cases} (\mathrm{D})(f_j^2, (H_i, W_i)) & \text{if } j < i, \\ (\mathrm{I})(f_j^2, (H_i, W_i)) & \text{if } j = i, \\ (\mathrm{U})(f_j^2, (H_i, W_i)) & \text{if } j > i, \end{cases} \tag{5}$$

where (D), (I), and (U) stand for adaptive average pooling, identity mapping, and bilinear interpolation of f_j^2 to the resolution $H_i \times W_i$, respectively. In Equation (6), θ_{ij} represents the parameters of the smoothing operation, and f_{ij}^4 is the j-th smoothed feature map at the i-th level.

$$f_{ij}^4 = \theta_{ij}(f_{ij}^3), \tag{6}$$

Next, we enhance the i-th level features by applying the elementwise Hadamard product to all the resized feature maps, integrating semantic information and finer details as follows:

$$f_i^5 = H([f_{i1}^4, f_{i2}^4, \ldots, f_{iM}^4]), \tag{7}$$

where $H(\cdot)$ denotes the Hadamard product.

The SDI-FPN pyramid network further improves the quality of feature extraction after feature fusion, obtains richer gradient flow information, effectively compensates for the loss of context information in the model, and improves the accuracy of small target detection.

3.4. Coordinate Adaptive Spatial Feature Fusion Mechanism Detection Head

To further enhance the accuracy of small target detection and address the impact of conflicting information among targets during cross-scale fusion, we have improved the detection head of YOLOv8. This study introduces a novel coordinate adaptive spatial feature fusion mechanism (CASFF), as shown in Figure 7. Initially, inspired by the adaptive spatial feature fusion (ASFF) strategy, features of adjacent scales are resized to the same dimensions, and spatial importance weights representing four different layer feature mappings are calculated, effectively filtering out conflicting information and enhancing scale invariance. By using these learned weights for weighted fusion, we obtain a new multi-scale feature map tensor, calculated as:

$$y_{ij}^l = \alpha_{ij}^l \cdot x_{ij}^{P2 \to l} + \beta_{ij}^l \cdot x_{ij}^{P3 \to l} + \gamma_{ij}^l \cdot x_{ij}^{P4 \to l} + \delta_{ij}^l \cdot x_{ij}^{P5 \to l}, \tag{8}$$

where $P2, P3, P4, P5$ denote different feature layers, $x_{ij}^{n \to l}$ represents the feature vectors resized across layers, and $\alpha_{ij}^l, \beta_{ij}^l, \gamma_{ij}^l, \delta_{ij}^l \in [0, 1]$ are the learned weights. These weights are defined by using the softmax function with control parameters $\lambda_{\alpha_{ij}}^l$, $\lambda_{\beta_{ij}}^l$, $\lambda_{\gamma_{ij}}^l$, $\lambda_{\delta_{ij}}^l$, respectively, and δ_{ij}^l is defined as follows:

$$\alpha_{ij}^l = \frac{e^{\lambda_{\alpha_{ij}}^l}}{e^{\lambda_{\alpha_{ij}}^l} + e^{\lambda_{\beta_{ij}}^l} + e^{\lambda_{\gamma_{ij}}^l} + e^{\lambda_{\delta_{ij}}^l}}, \tag{9}$$

ensuring that $\alpha_{ij}^l + \beta_{ij}^l + \gamma_{ij}^l + \delta_{ij}^l = 1$.

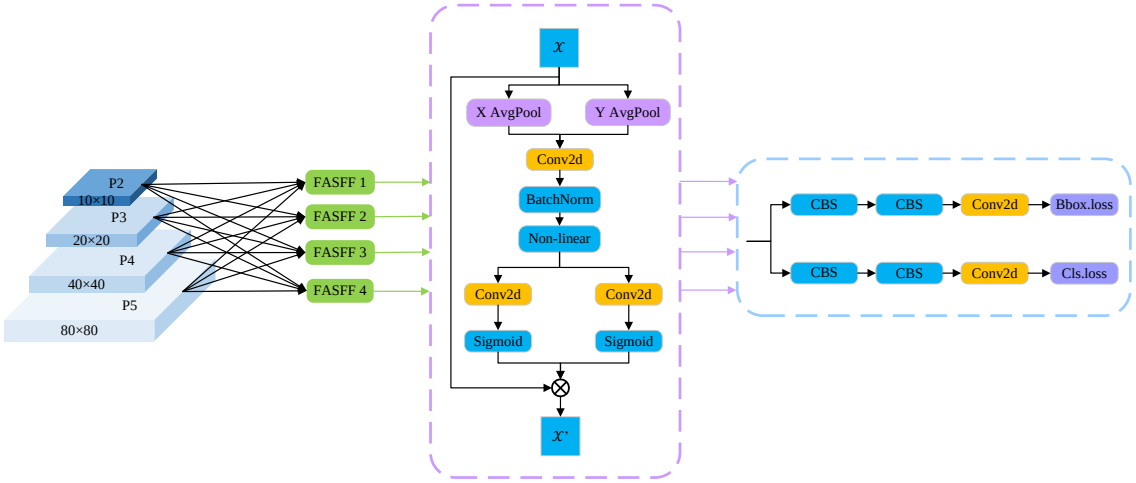


Figure 7. Structure of the coordinate adaptive spatial feature fusion mechanism detection head, where FASSF represents the features after spatial fusion according to the learned weight map.

To prevent the loss of spatial information during feature fusion, a CA mechanism is introduced to improve the network's focus on essential information, reducing false positives and missed detections. This mechanism involves two one-dimensional global pooling operations that aggregate the input features in vertical and horizontal directions, creating two independent direction-aware feature maps. These maps are then concatenated and processed through a convolutional layer, followed by a batch normalization (BN) layer and a non-linear activation layer. Subsequently, the feature maps are separated and processed through another convolutional layer, and the coordinate features are normalized using the Sigmoid function to obtain coordinate attention weights. These two direction-embedded feature maps are encoded into attention maps, each capturing long-range dependencies along one spatial direction of the input feature map. Finally, these attention maps are multiplied with the input feature map to enhance its representational power.

By integrating the adaptive spatial feature fusion (ASFF) strategy and the coordinate attention (CA) module into the detection head, the model not only leverages the multi-scale feature fusion advantages of the ASFF module, but also incorporates coordinate information through the CA attention mechanism. This combination significantly enhances the spatial expressiveness and precision of the final feature map. Consequently, the model's ability to perceive the positional information of small targets and long-range dependencies is greatly improved. This enables more accurate localization of small targets within detected images, thereby significantly enhancing the detection accuracy for small targets.

4. Experiments

Our experimental environment used Ubuntu 20.04.5 LTS as the operating system, with processing performed by an Intel(R) Xeon(R) CPU E5-2680 v4 and an NVIDIA GeForce RTX 3090 GPU (24GB). The software framework consisted of Python 3.8 and torch 2.0.0+cu117. We adopted the YOLOv8n model as our experimental baseline, employing a batch size of 8 for 200 training epochs. The optimizer chosen was Stochastic Gradient Descent (SGD), initialized with a learning rate of 0.01. The input images were standardized to a resolution of 640 × 640 pixels. Throughout the training procedure, mosaic data augmentation was applied to fortify the model's robustness.

4.1. Dataset and Evaluation Criteria

We assessed the efficacy of our model using the VisDrone2019 dataset [43], a UAV-based dataset tailored for object detection and tracking tasks, facilitating comprehensive evaluation and research of visual analysis algorithms for UAV platforms. This dataset offers a rich and diverse collection of images captured from various perspectives and tasks

utilizing drones. Noteworthy features encompass a wide array of detected objects, ranging from highly diverse to monotone, varying numbers and distributions of detected objects, and observations captured under both day and night lighting conditions. Comprising 10 categories, the training set consisted of 6471 images, while the validation set comprised 548 images. Additionally, the test set contained 1610 images, with the weight of each class in the training set being proportional to the number of labels. Notably, the dataset exhibited class imbalance and included small objects, as depicted in Figure 8. According to the COCO standard [44], we categorized the objects into three sizes: small, medium, and large. Specifically, objects with bounding box areas smaller than 32 × 32 pixels were classified as small objects, those with areas larger than 32 × 32 pixels, but smaller than 96 × 96 pixels were classified as medium objects, and those with areas larger than 96 × 96 pixels were classified as large objects. Table 1 shows the distribution of large, medium, and small objects for each category. As can be seen, small objects account for 60.49% of the targets in the VisDrone2019 dataset. This high proportion of small targets makes it an ideal dataset for evaluating the performance of small object detection.

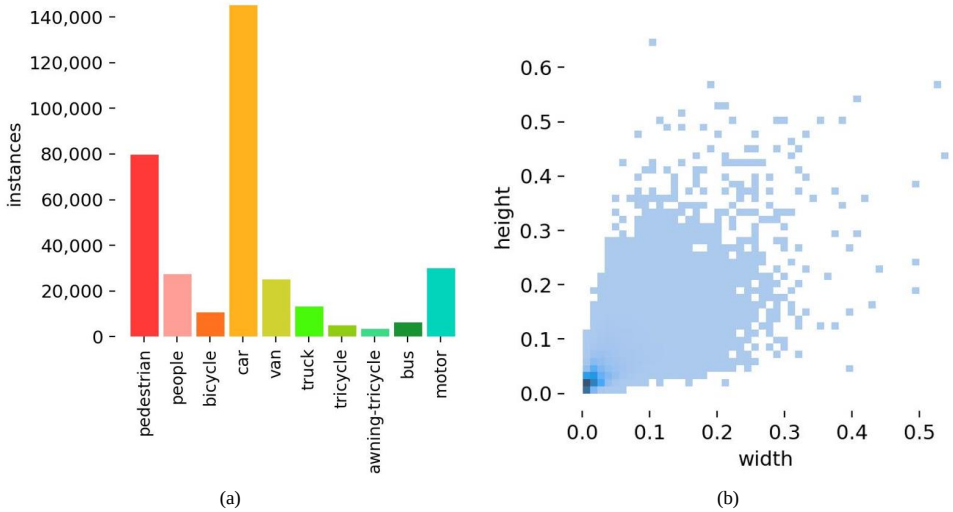


Figure 8. The distribution of the objects in the dataset: (**a**) distribution of the number of classes; (**b**) the width and height distribution of the target; the concentration of the distribution is indicated by the color gradient from light white to dark blue, indicating that the distribution becomes more and more concentrated.

Table 1. Distribution of target sizes in the VisDrone2019 dataset.

Category	Small		Medium		Large	
	Number	Proportion	Number	Proportion	Number	Proportion
pedestrian	65,227	82.22%	13,804	17.40%	306	0.39%
people	23,497	86.84%	3463	12.80%	99	0.37%
bicycle	7111	67.85%	3244	30.95%	125	1.19%
car	69,862	48.22%	63,079	43.54%	11,926	8.23%
van	10,751	43.08%	11,517	46.15%	2688	10.77%
truck	3982	30.93%	6725	52.23%	2168	16.84%
tricycle	2176	45.22%	2425	50.39%	211	4.38%
awning tricycle	1360	41.90%	1699	52.34%	187	5.76%
bus	1661	28.03%	3265	55.10%	1000	16.87%
motor	21,986	74.16%	7395	25.00%	266	0.90%
Total	207,613	60.49%	116,616	33.98%	18,976	5.53%

To assess the effectiveness of our proposed enhanced detection model, we employed precision (P), recall (R), average precision (AP), mean average precision at IoU threshold 0.5 (mAP@0.5), and mean average precision from IoU 0.5 to 0.95 (mAP@0.5:0.95) as the

evaluation metrics. Precision denotes the ratio of correctly detected targets to the total number of predicted targets, while recall indicates the ratio of correctly detected targets to the total number of actual targets. The formulas for these metrics are as follows:

$$R = \frac{TP}{TP + FN} \tag{10}$$

$$P = \frac{TP}{TP + FP} \tag{11}$$

where TP (True Positive) represents the count of instances where the model correctly identifies positive-class samples as positive, FP (False Positive) signifies the count of instances where the model incorrectly identifies negative-class samples as positive, and FN (False Negative) denotes the count of instances where the model incorrectly classifies positive-class samples as negative.

The F1 score, ranging from 0 to 1, represents the harmonic mean of precision and recall. It takes into account the significance of both precision and recall, effectively assessing algorithm performance. The formula is as follows:

$$F1 = \frac{2 \times P \times R}{P + R} \tag{12}$$

The average precision (AP) evaluates the performance of object detection models for a particular class. It is computed as the area under the precision–recall (P–R) curve, with recall plotted on the x-axis and precision on the y-axis:

$$AP = \int_0^1 P(R)\, dR \tag{13}$$

The mean average precision (mAP) signifies the average precision across all categories, serving as a comprehensive performance metric for multi-class detection tasks:

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{14}$$

Specifically, mAP@0.5 denotes the mean average precision at an IoU threshold of 0.5, while mAP@0.5:0.95 represents the mean average precision for IoU thresholds ranging from 0.5 to 0.95, in increments of 0.05.

4.2. Experimental Results

Remotely sensed images, such as those from UAVs, have significantly different characteristics from images captured by ground personnel. As a result, unmanned aerial vehicle image recognition is more challenging than traditional image recognition. To confirm the superiority of the proposed DDSC-YOLO model over the YOLOv8n model in remote sensing detection tasks, comparative experiments were conducted on the VisDrone2019 validation set to assess the performance of both models. To better evaluate the model's object-detection capabilities on a small scale, the AP_s indicator was added. The results, presented in Table 2, indicate that, compared to the baseline model, our strategy achieved significant improvements of 9.2% in P, 7.4% in R, 8.3% in F1, 9.3% in mAP0.5, and 6.4% in mAP0.5:0.95. Additionally, AP_s shows significant growth compared to YOLOv8n, YOLOv8s, and YOLOv8m. These significant improvements demonstrate an overall enhancement in detection accuracy, indicating that the enhanced model significantly boosts the detection accuracy for small targets.

Table 2. The comparison results with the baseline model YOLOv8n, YOLOv8s, and YOLOv8m.

Model	P	R	F1	AP_s	mAP0.5	mAP0.5:0.95	Parameters
YOLOv8n	0.437	0.329	0.376	0.089	0.329	0.191	3.01M
YOLOv8s	0.49	0.385	0.431	0.126	0.393	0.235	11.13M
YOLOv8m	0.523	0.414	0.462	0.141	0.422	0.254	25.86M
DDSC-YOLO	0.529	0.403	0.459	0.155	0.422	0.255	4.99M

The precision–recall (P–R) curve illustrates the trade-off between precision and recall across various thresholds. A larger area under the P–R curve signifies better overall performance, reflecting higher average precision (AP). This means the model achieves greater accuracy and recall in classification tasks. As depicted in Figure 9, the area under the P–R curve for the DDSC-YOLO model (0.422) surpasses that of YOLOv8n (0.329), confirming the superior detection capability of our enhanced model over the original YOLOv8n.

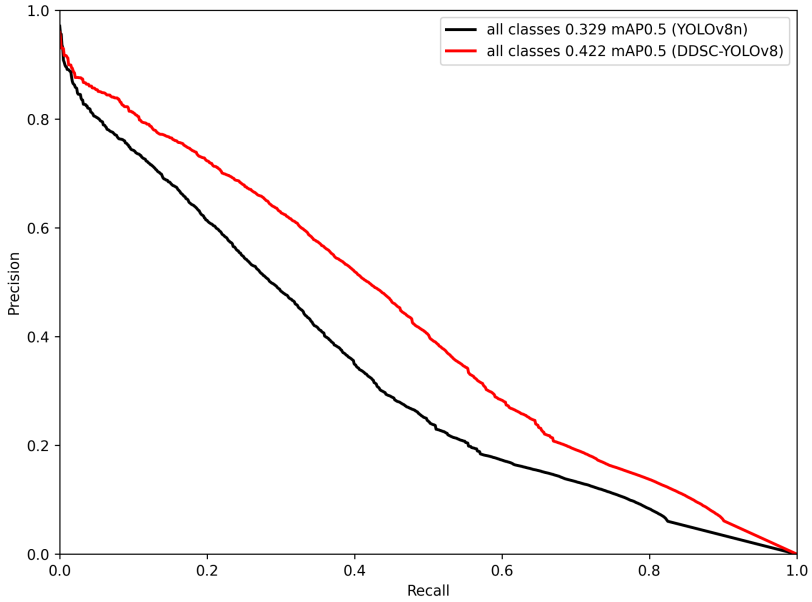
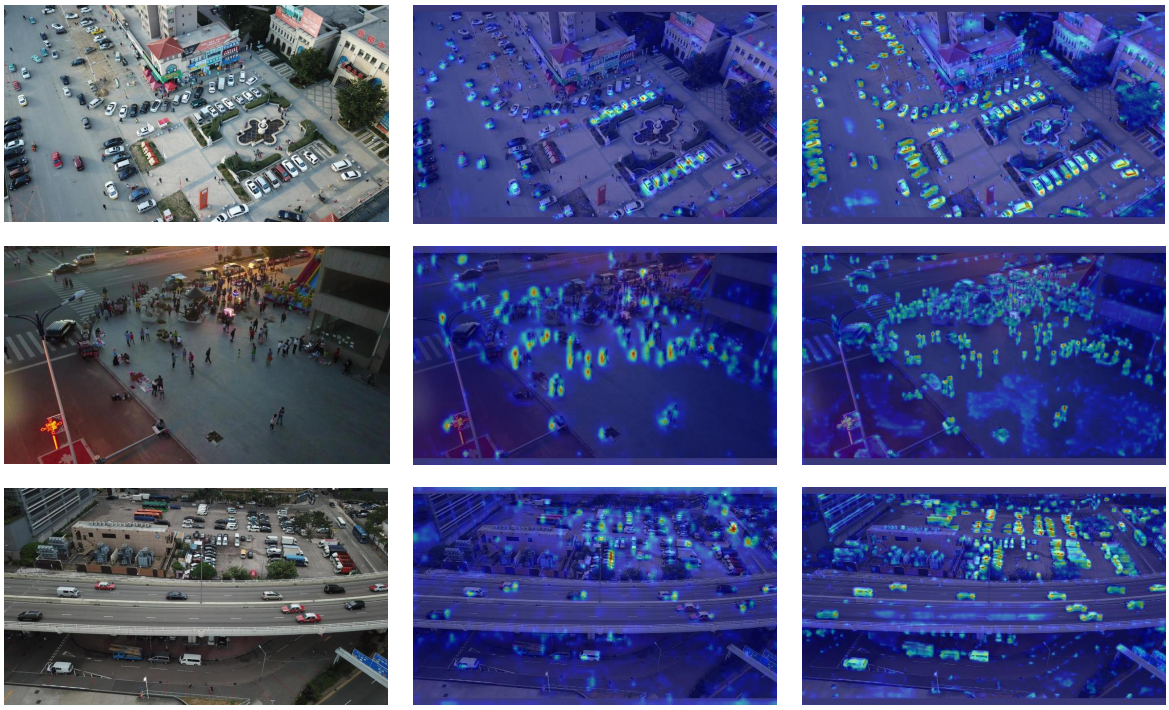


Figure 9. P–R curves of YOLOv8n (black) and DDSC-YOLO (red).

To visually compare the models before and after optimization, we used Gradient-weighted Class Activation Mapping (GradCAM) to visualize the output layers for small objects from both YOLOv8n and DDSC-YOLO. As shown in Figure 10, without improvements to YOLOv8n, the model is affected by densely packed objects and complex backgrounds, resulting in less focus on small and distant targets.

After applying the above optimization steps to YOLOv8n, the model shows an improved focus on small target regions. For example, compared to the baseline, the improved model better highlights distant cars in the background of the square in the first image, people and vehicles under night lighting conditions in the second image, and distant cars in the background of the elevated bridge in the third image. The baseline method shows a weak classification ability when dealing with edge, clustered, or blurred objects, often misclassifying background as foreground targets. In contrast, our proposed approach shows promising performance, indicating that the improved model effectively prevents information loss when downsampling small targets. Our method has distinct advantages in handling small targets and overall image understanding, effectively suppressing irrelevant background interference. As a result, we have improved the detection accuracy of small targets.



(a) original images (b) heat maps of YOLOv8n (c) heat maps of DDSC-YOLO

Figure 10. Compare thermal maps generated for different model detection results: (**a**) Original images. (**b**) Heat maps of YOLOv8n. (**c**) Heat maps of DDSC-YOLO.

4.3. Ablation Experiment

To evaluate the impact of the newly added and modified modules on the performance of the baseline model, a series of controlled experiments was conducted for evaluation and comparison. Using YOLOv8n as the baseline, we performed training and testing under consistent experimental conditions—employing the VisDrone2019 dataset and identical training parameters. The experimental outcomes are summarized in Table 3.

Table 3. Ablation experiments on the VisDrone2019 dataset.

Model	SDI-FPN	DCNv3LKA	DualC2f	CASFF-Head	P	R	mAP0.5	mAP0.5:0.95	AP_s
1					0.437	0.329	0.329	0.191	0.089
2	✓				0.489	0.38	0.387	0.232	0.138
3	✓	✓			0.493	0.388	0.395	0.237	0.144
4	✓	✓	✓		0.491	0.39	0.401	0.244	0.146
5	✓	✓	✓	✓	0.529	0.403	0.422	0.255	0.155

As shown in Table 3, each enhancement strategy applied to the baseline model resulted in varying degrees of improved detection performance. In particular, replacing the original path aggregation feature pyramid network with the SDI-FPN structure and adding a detection head specifically for small targets resulted in significant improvements in all metrics: the precision (P), recall (R), mAP0.5, mAP0.5:0.95, and AP_s increased by 5.2%, 5.1%, 5.8%, 4.1%, and 4.9%, respectively. These improvements are attributed to effectively preventing the loss of contextual information during the multi-level fusion process, facilitating the integration of low-level and high-level information, and significantly enhancing the overall performance metrics of the model. To adapt to the wide variation in target sizes, more emphasis was placed on critical information in small targets within the input features, while

suppressing non-critical feature information. The inclusion of the DCNv3LKA module in the feature-extraction process resulted in a 0.8% increase in the mAP, and the AP_s improved by 0.6%. Using the DualC2f module instead of the C2f module improved the mAP0.5 by 0.6% by reducing the redundancy of feature map information and promoting information sharing between convolutional layers, thus enriching the gradient flow. In addition, the introduction of the coordinate adaptive spatial feature fusion mechanism in the detection head effectively mitigated conflicts between different feature levels, filtered out conflicting information, and improved the model's ability to recognize images at different scales, with increases of 3.8%, 1.3%, 2.1%, 1.1%, and 0.9% in the P, R, mAP0.5, mAP0.5:0.95, and AP_s, respectively. These experimental results demonstrate that detailed optimizations at different stages of the algorithm can significantly improve the model's learning efficiency and confirm the effectiveness of each optimization measure. The optimized network structure effectively addresses the challenges of detecting small-sized targets.

In the detection head, the addition of the CA attention mechanism was compared with commonly used attention mechanisms. Table 4 shows the performance of the model after incorporating different attention mechanisms. Compared to SGE [45], SimAM [46], and the Convolutional Block Attention Module (CBAM), the CA attention mechanism demonstrated the best detection performance. The inclusion of the CA allows the model to extract features more effectively, particularly for small and dense objects that often blend into blurred edges and backgrounds in the dataset. This significantly reduces the influence of irrelevant information on detection results, thereby enhancing the model's ability to detect small targets.

Table 4. Results of the comparison of different attention mechanisms.

Model	P	R	mAP0.5	mAP0.5:0.95
+SGE	0.516	0.401	0.416	0.251
+SimAM	0.531	0.397	0.419	0.254
+CBAM	0.505	0.399	0.413	0.25
+CA (Ours)	0.529	0.403	0.422	0.255

4.4. Comparative Experiment

To assess the enhancements made to the YOLOv8 network model, we compared the performance of our improved model with that of current mainstream object-detection algorithms. The experimental results from this quantitative analysis and comparison are presented in Table 5.

On the mAP50 metric, our proposed DDSC-YOLO model achieved significant performance improvements: it outperformed Faster R-CNN by 20.5%, RetinaNet by 28.3%, Cascade-RCNN by 19%, and CornerNet by 24.8%. Faster R-CNN had low resolution feature maps extracted by their backbone networks, resulting in relatively poor detection accuracy for small objects. RetinaNet and CornerNet can suffer when dealing with dense objects and tend to ignore the overlap of extremely small objects, reducing detection accuracy. Cascade R-CNN's multi-level detection enhances overall performance, but adds complexity and training difficulty, necessitating further optimization. In contrast, our proposed model delivers superior results, avoiding the limitations of retinal networks.

Compared to YOLOv3-tiny, YOLOv4, YOLOv5s, YOLOv5m, CDNet, YOLOv6, YOLOv7-tiny, YOLOv8n, and YOLOv8s, the improvements were 18.6%, 1.8%, 9.2%, 4.9%, 8%, 13%, 6.7%, 9.3%, and 2.9%, respectively. YOLOv3-Tiny and YOLOv7-Tiny achieved lightweight models, but lost much of their detection accuracy. YOLOv4 had good detection accuracy, but performed poorly in terms of model size. CDNet showed high detection accuracy in two categories, tricycle awning and tricycle, but showed severe misses and false detections in other categories. YOLOv6 had poor detection performance and complexity. YOLOv5s and YOLOv8n had a small model size. These two models cannot meet the detection requirements of a high proportion of small objects. Notably, the DDSC-YOLO model has only 4.99 million model parameters, representing an 80.7% reduction compared to the 25.9 million parameters of YOLOv8m. Despite

the significantly smaller parameter count, the detection accuracy remained on par with that of YOLOv8m, demonstrating its efficiency. Compared to improved algorithms such as DBAI-Det, YOLOv5s-pp, LW-YOLO v8, DC-YOLOv8, and Drone-YOLO (nano), which are designed to detect small targets in UAV and other remote sensing images, our method showed significant improvements. Specifically, our approach achieved increases of 14.2%, 0.5%, 5.9%, 0.7%, and 4.1%, respectively.

Specifically, in four categories—people, car, van, and motor—the DDSC-YOLO model demonstrated exceptional detection performance. This indicates that the model significantly outperformed other classical algorithms in small-target-detection tasks. This achievement is attributed to the proposed feature pyramid network, which enhances the fusion of semantic and detailed information, improves the overall feature representation and network performance, and implements four-scale detection and enhanced feature extraction capabilities. Moreover, the integration of a coordinate adaptive spatial feature fusion mechanism in the detection head optimizes the model's recognition capabilities across different image scales. Comprehensive experimental results showed that the DDSC-YOLO model performed exceptionally well in detecting targets in aerial images captured by drones. The model effectively addressed the challenges posed by complex backgrounds in drone detection and considered the issue of class imbalance in the dataset, demonstrating robustness.

Table 5. Compare results with other models on VisDrone2019-val; '--' denotes difficult to obtain data. The black bold numbers in the table indicate the best results.

Model	mAP0.5	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awning Tricycle	Bus	Motor
Faster R-CNN	0.217	0.214	0.156	0.067	0.517	0.295	0.19	0.131	0.077	0.314	0.207
RetinaNet	0.139	0.13	0.079	0.014	0.455	0.199	0.115	0.063	0.042	0.178	0.118
Cascade-RCNN	0.232	0.222	0.148	0.076	0.546	0.315	0.216	0.148	0.086	0.349	0.214
CornerNet [47]	0.174	0.204	0.066	0.046	0.409	0.202	0.205	0.14	0.093	0.244	0.121
YOLOv3-tiny	0.236	0.189	0.143	0.044	0.606	0.302	0.229	0.15	0.072	0.403	0.219
YOLOv4	0.404	0.434	0.33	0.161	0.789	0.442	0.393	0.285	0.163	**0.609**	0.442
YOLOv5s	0.33	0.402	0.322	0.103	0.733	0.395	0.275	0.181	0.107	0.415	0.397
YOLOv5m	0.373	0.455	0.357	0.15	0.763	0.402	0.327	0.245	0.119	0.487	0.423
YOLOv5s-pp [48]	0.417	**0.517**	0.396	**0.19**	0.821	0.441	0.36	0.263	0.147	0.553	0.482
DBAI-Det	0.28	0.367	0.128	0.147	0.474	0.38	**0.414**	0.234	0.169	0.319	0.166
CDNet [49]	0.342	0.356	0.192	0.138	0.558	0.421	0.382	**0.33**	**0.254**	0.495	0.293
YOLOv6 [50]	0.292	0.314	0.251	0.043	0.737	0.348	0.242	0.171	0.1	0.391	0.321
YOLOv7-tiny	0.355	0.396	0.362	0.096	0.775	0.383	0.303	0.194	0.102	0.496	0.445
YOLOv8n	0.329	0.347	0.282	0.0815	0.757	0.381	0.282	0.207	0.124	0.471	0.362
YOLOv8s	0.393	0.421	0.329	0.125	0.795	0.451	0.36	0.282	0.158	0.569	0.443
YOLOv8m	0.422	0.459	0.358	0.155	0.811	0.47	0.393	0.316	0.177	0.607	0.474
LW-YOLO v8 [51]	0.363	0.393	0.319	0.092	0.785	0.421	0.308	0.238	0.15	0.506	0.415
DC-YOLOv8 [52]	0.415	--	--	--	--	--	--	--	--	--	--
Drone-YOLO (nano)	0.381	--	--	--	--	--	--	--	--	--	--
DDSC-YOLO	**0.422**	0.489	**0.403**	0.151	**0.831**	**0.478**	0.35	0.276	0.163	0.591	**0.486**

4.5. Generalization Test

To further evaluate the effectiveness of our method, we conducted comparative analyses with the YOLO series object-detection algorithms on the SSDD [53] and RSOD datasets [54]. For a fair comparison, we used the same hyperparameters and training policies as those employed for the VisDrone dataset. The SSDD dataset, designed for ship detection in satellite imagery, includes 1160 high-resolution remotely sensed images with a single category—ships—comprising a total of 2456 instances. This dataset is divided into training, validation, and testing sets with an 8:1:1 ratio, resulting in 928 images for training, 116 for validation, and 116 for testing. The RSOD dataset is tailored to object detection in remote sensing images and features four target categories: airplanes, oil tanks, playgrounds, and overpasses. It contains 4993 aircraft instances, 1586 oil tank instances,

191 playground instances, and 180 overpass instances. This dataset was also split into the same 8:1:1 ratio. Both datasets contain small and very small objects, making detection difficult and prone to false positives and false negatives.

According to Table 6, the Drone-YOLO algorithm proposed in this paper outperformed Faster R-CNN, YOLOv3-tiny, YOLOv5s, YOLOv6, YOLOv7-tiny, and YOLOv8n in terms of the precision, recall, mAP50, and mAP0.5:0.95 on both the SSDD and RSOD datasets. Specifically, compared to YOLOv8n, the mAP0.5 metric increased by 1.3% and 3.2% on the SSDD and RSOD datasets, respectively. This demonstrates that the DDSC-YOLO model introduced in our research is versatile and exhibits superior detection capabilities across various remote sensing scenarios, not being limited to specific datasets.

Table 6. Comparison results on other datasets.

Dataset	Model	P	R	mAP0.5	mAP0.5:0.95
SSDD	Faster R-CNN	0.824	0.856	0.926	0.649
	YOLOv3-tiny	0.947	0.840	0.938	0.666
	YOLOv5s	0.956	0.957	0.978	0.707
	YOLOv6	0.955	0.927	0.976	0.695
	YOLOv7-tiny	0.930	0.917	0.957	0.588
	YOLOv8n	0.940	0.931	0.969	0.701
	DDSC-YOLO	0.955	0.930	0.982	0.694
RSOD	Faster R-CNN	0.918	0.915	0.899	0.574
	YOLOv3-tiny	0.938	0.661	0.730	0.465
	YOLOv5s	0.926	0.920	0.950	0.605
	YOLOv6	0.881	0.933	0.939	0.645
	YOLOv7-tiny	0.912	0.908	0.915	0.586
	YOLOv8n	0.925	0.873	0.919	0.608
	DDSC-YOLO	0.918	0.914	0.951	0.609

4.6. Visualization

Figure 11 shows the visualization results of the DDSC-YOLO model in some complex scenes and dense object areas in the VisDrone2019 dataset, with bounding box labels and confidence scores displayed. The results indicate that the DDSC-YOLO model can accurately detect small targets even in these challenging scenarios, maintaining high confidence scores and demonstrating its good performance in handling complex natural backgrounds and dense object detection.

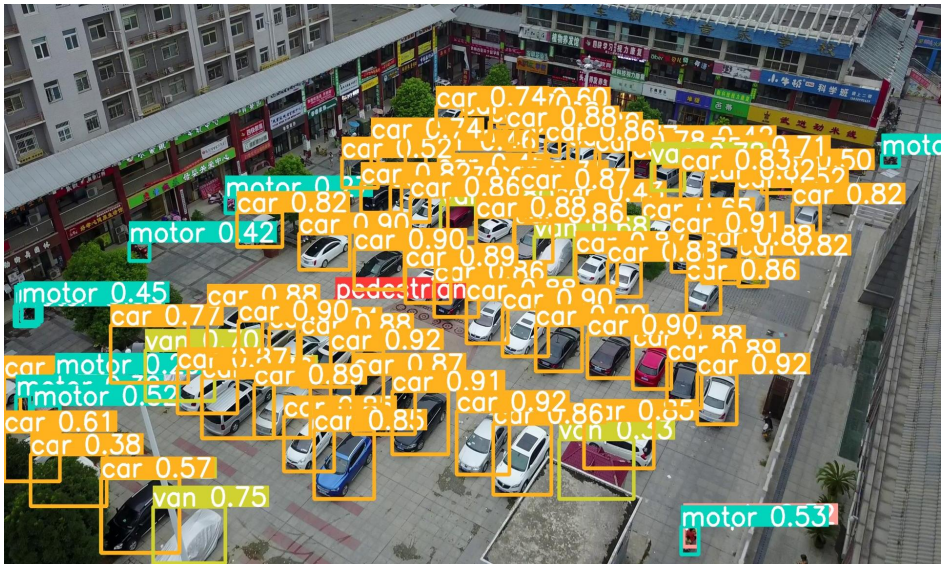
To clearly demonstrate the detection performance of our method, we compared the detection results of DDSC-YOLO and the baseline method using visualization techniques. We selected some representative images from the VisDrone, SSDD, and RSOD datasets as experimental data. These images include small objects under different lighting conditions, backgrounds, and density levels, making them suitable for comparative analysis experiments.

Figure 12 presents these comparative results. As objects in drone and remote sensing images are typically small in size, we enlarged some samples to ensure the clear display of their differences. The results clearly show that our method had higher effectiveness and robustness in various complex situations compared to the YOLOv8n method. For instance, when comparing group a, the baseline method was significantly impacted by occlusion from pedestrians and vehicles, whereas our method demonstrated notable superiority under low-light conditions. In groups b and e, where numerous small and obstructive objects are present, the baseline method experienced a significant decrease in detection accuracy due to the influence of dense objects, while our model showed advanced detection capabilities in complex scenes. In groups c and d, our model showed a lower false negative rate compared to YOLOv8n due to the small size of the boats and the blurred remote sensing images. In addition, in group f, YOLOv8 showed significant false positives, identifying a single aircraft as two detection boxes, while our method accurately identified the object. Through these visualizations, DDSC-YOLO can effectively guide the detector to focus on challenging regions, effectively suppress non-critical information, and enhance the focus

on critical information. It demonstrates good detection performance in practical scenarios with varying lighting conditions, backgrounds, and object scales.



(a)



(b)

Figure 11. Visualization results of DDSC-YOLO model on VisDrone2019 dataset. (**a**) shows a densely populated basketball court area. (**b**) shows a parking lot with vehicles arranged in an orderly fashion.

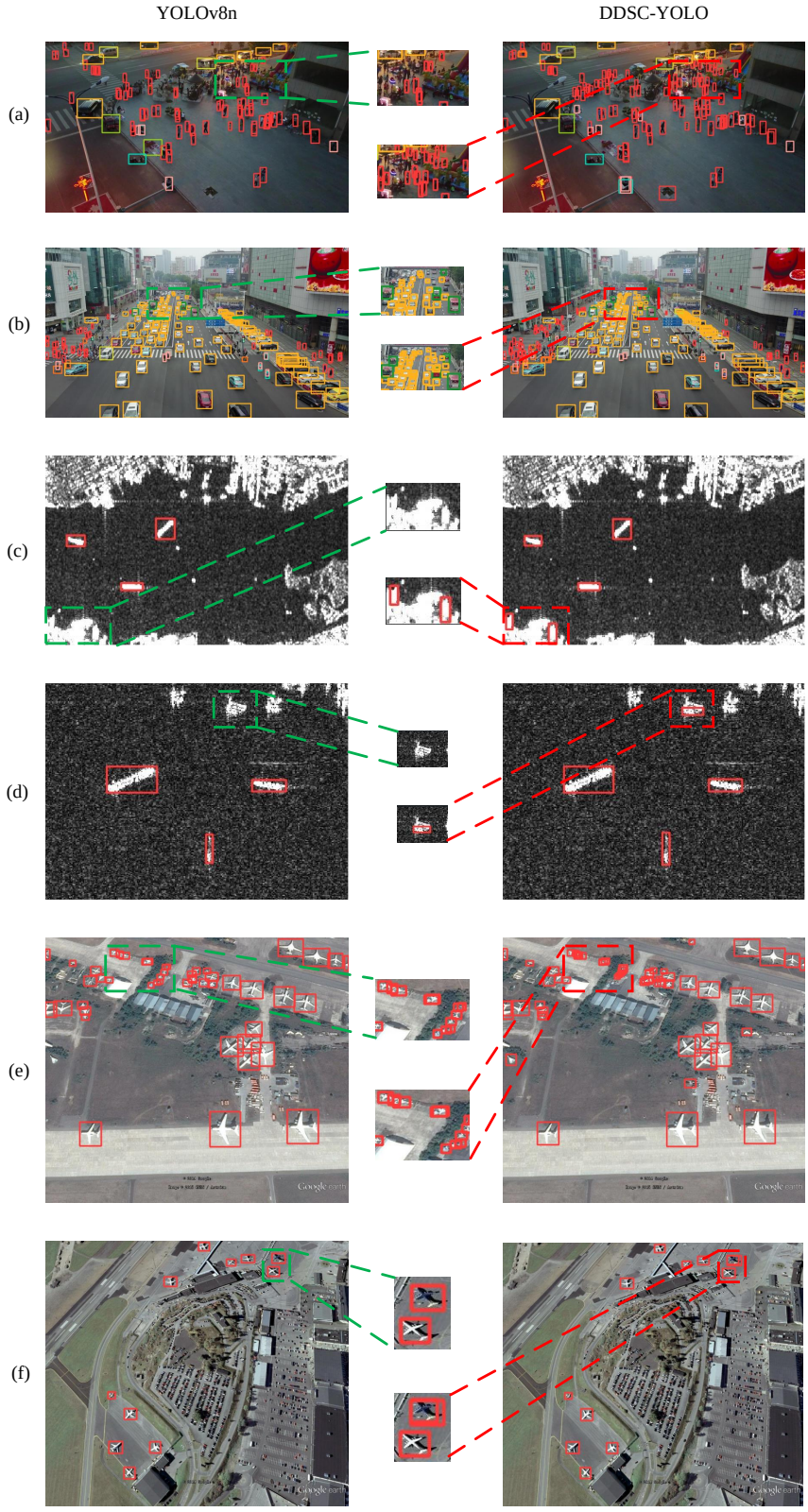


Figure 12. The visualization results between DDSC-YOLO and YOLOv8n are compared on VisDrone2019, SSDD, and RSOD datasets. (**a**,**b**) are from the VisDrone2019 dataset, (**c**,**d**) are from the SSDD dataset, and (**e**,**f**) are from the RSOD dataset.

5. Conclusions

This study focuses on the problems of the difficult detection of small targets in UAV aerial scenes, the high missed detection rate, and the frequent false detection rate, based on the YOLOv8n model. We propose a small-target-detection model specifically designed for UAV aerial scenes, called DDSC-YOLO. First, DualC2f addresses cross-channel communication issues in feature maps while preserving critical information. In addition, the DCNv3LKA attention mechanism improves interlayer information retention and reduces non-critical details, which is critical for detecting small targets against complex backgrounds. The SDI-FPN structure refines multi-level feature fusion, preventing the loss of contextual information. This integration enables seamless blending of low-level and high-level information, which is critical for accurate detection of small targets. In addition, a novel coordinate-adaptive spatial feature fusion mechanism in the sensor head improves feature representation across different target scales.

Experimental validation on the VisDrone2019 dataset demonstrates the superior performance of our approach. Compared to YOLOv8n, our DDSC-YOLO increases the mAP0.5 by 9.3% and AP_s by 6.6%, effectively improving small target detection accuracy and outperforming other comparison models on the mAP0.5 and mAP0.5-0.95 metrics. Our model also demonstrates robust generalization across the SSDD and RSOD datasets. Going forward, our research will focus on further improving the model's lightness, detection accuracy, and processing speed to make it more effective and improve small target detection performance. We plan to explore efficient network optimization algorithms such as quantization compression and sparse training to adapt to the characteristics of high-resolution images, thereby accelerating network training speed and improving validation accuracy. In addition, we will consider applying model pruning and knowledge distillation techniques to eliminate redundant and unnecessary network connections. This will further improve the efficiency and speed of lightweight models, opening up new possibilities for object-detection technology in remote sensing image scenes.

Author Contributions: Conceptualization, S.Z.; methodology, S.Z.; investigation, S.Z. and H.Z.; validation, H.Z.; formal analysis, S.Z. and H.Z.; data curation, S.Z.; writing—original draft preparation, S.Z.; writing—review and editing, S.Z. and H.Z.; visualization, S.Z.; supervision, H.Z.; project administration, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Key R&D Program of China under Grant No. 2022YFB3904805.

Data Availability Statement: The data are contained within the article.

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

References

1. Aiello, G.; Hopps, F.; Santisi, D.; Venticinque, M. The employment of unmanned aerial vehicles for analyzing and mitigating disaster risks in industrial sites. *IEEE Trans. Eng. Manag.* **2020**, *67*, 519–530. [CrossRef]
2. Ropero, F.; Muñoz, P.; R-Moreno, M.D. TERRA: A path planning algorithm for cooperative UGV–UAV exploration. *Eng. Appl. Artif. Intell.* **2019**, *78*, 260–272. [CrossRef]
3. Motlagh, N.H.; Bagaa, M.; Taleb, T. UAV-based IoT platform: A crowd surveillance use case. *IEEE Commun. Mag.* **2017**, *55*, 128–134. [CrossRef]
4. Outay, F.; Mengash, H.A.; Adnan, M. Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges. *Transp. Res. Part A Policy Pract.* **2020**, *141*, 116–129. [CrossRef] [PubMed]
5. Koparan, C.; Koc, A.B.; Privette, C.V.; Sawyer, C.B. In situ water quality measurements using an unmanned aerial vehicle (UAV) system. *Water* **2018**, *10*, 264. [CrossRef]
6. Zhao, D.; Shao, F.; Liu, Q.; Yang, L.; Zhang, H.; Zhang, Z. A Small Object Detection Method for Drone-Captured Images Based on Improved YOLOv7. *Remote Sens.* **2024**, *16*, 1002. [CrossRef]
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef]
8. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.

9. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
10. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 379–387.
11. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
12. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
13. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
14. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
15. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Cham, Switzerland, 2016; pp. 21–37.
16. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
17. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
18. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
19. Fan, H.; Xiong, B.; Mangalam, K.; Li, Y.; Yan, Z.; Malik, J.; Feichtenhofer, C. Multiscale vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 6824–6835.
20. Ding, P.; Qian, H.; Zhou, Y.; Chu, S. Object detection method based on lightweight YOLOv4 and attention mechanism in security scenes. *J. Real-Time Image Process.* **2023**, *20*, 34. [CrossRef]
21. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 13713–13722.
22. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
23. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
24. Wu, W.; Liu, H.; Li, L.; Long, Y.; Wang, X.; Wang, Z.; Li, J.; Chang, Y. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image. *PLoS ONE* **2021**, *16*, e0259283. [CrossRef] [PubMed]
25. 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 IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
26. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
27. Zhao, L.; Zhu, M. MS-YOLOv7: YOLOv7 based on multi-scale for object detection on UAV aerial photography. *Drones* **2023**, *7*, 188. [CrossRef]
28. Zhang, C.; Wang, L.; Cheng, S.; Li, Y. SwinSUNet: Pure transformer network for remote sensing image change detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5224713. [CrossRef]
29. Lin, B. Safety Helmet Detection Based on Improved YOLOv8. *IEEE Access* **2024**, *12*, 28260–28272. [CrossRef]
30. Zhang, Z. Drone-YOLO: An efficient neural network method for target detection in drone images. *Drones* **2023**, *7*, 526. [CrossRef]
31. 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] [PubMed]
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1106–1114. [CrossRef]
33. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
34. Azad, R.; Niggemeier, L.; Hüttemann, M.; Kazerouni, A.; Aghdam, E.K.; Velichko, Y.; Bagci, U.; Merhof, D. Beyond self-attention: Deformable large kernel attention for medical image segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2024; pp. 1287–1297.
35. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
36. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.

37. Liu, S.; Huang, D.; Wang, Y. Learning spatial fusion for single-shot object detection. *arXiv* **2019**, arXiv:1911.09516.
38. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
39. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21002–21012.
40. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
41. Zhong, J.; Chen, J.; Mian, A. DualConv: Dual convolutional kernels for lightweight deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 9528–9535. [CrossRef]
42. Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 14408–14419.
43. 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 IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
44. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Cham, Switzerland, 2014; pp. 740–755.
45. Li, X.; Hu, X.; Yang, J. Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. *arXiv* **2019**, arXiv:1905.09646.
46. Yang, L.; Zhang, R.Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 11863–11874.
47. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
48. Xu, H.; Zheng, W.; Liu, F.; Li, P.; Wang, R. Unmanned Aerial Vehicle Perspective Small Target Recognition Algorithm Based on Improved YOLOv5. *Remote Sens.* **2023**, *15*, 3583. [CrossRef]
49. Zhang, Z.D.; Tan, M.L.; Lan, Z.C.; Liu, H.C.; Pei, L.; Yu, W.X. CDNet: A real-time and robust crosswalk detection network on Jetson nano based on YOLOv5. *Neural Comput. Appl.* **2022**, *34*, 10719–10730. [CrossRef]
50. 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.
51. Huangfu, Z.; Li, S. Lightweight You Only Look Once v8: An Upgraded You Only Look Once v8 Algorithm for Small Object Identification in Unmanned Aerial Vehicle Images. *Appl. Sci.* **2023**, *13*, 12369. [CrossRef]
52. Lou, H.; Duan, X.; Guo, J.; Liu, H.; Gu, J.; Bi, L.; Chen, H. DC-YOLOv8: Small-size object detection algorithm based on camera sensor. *Electronics* **2023**, *12*, 2323. [CrossRef]
53. Zhang, T.; Zhang, X.; Li, J.; Xu, X.; Wang, B.; Zhan, X.; Xu, Y.; Ke, X.; Zeng, T.; Su, H.; et al. SAR ship detection dataset (SSDD): Official release and comprehensive data analysis. *Remote Sens.* **2021**, *13*, 3690. [CrossRef]
54. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2486–2498. [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.