drones

MDPI

Article

MSUD-YOLO: A Novel Multiscale Small Object Detection Model for UAV Aerial Images

Xiaofeng Zhao, Hui Zhang *, Wenwen Zhang, Junyi Ma, Chenxiao Li, Yao Ding and Zhili Zhang

The National Key Laboratory of Optical Engineering, the Rocket Force University of Engineering, Xi'an 710025, China; xife_zhao@163.com (X.Z.); ww_z_@outlook.com (W.Z.); 15029904066@163.com (J.M.); 17692385658@163.com (C.L.); dingyao.88@outlook.com (Y.D.); 157918018@163.com (Z.Z.)
* Correspondence: zh17879839221@outlook.com

Abstract: Due to the objects in UAV aerial images often presenting characteristics of multiple scales, small objects, complex backgrounds, etc., the performance of object detection using current models is not satisfactory. To address the above issues, this paper designs a multiscale small object detection model for UAV aerial images, namely MSUD-YOLO, based on YOLOv10s. First, the model uses an attention scale sequence fusion mode to achieve more efficient multiscale feature fusion. Meanwhile, a tiny prediction head is incorporated to make the model focus on the low-level features, thus improving its ability to detect small objects. Secondly, a novel feature extraction module named CFormerCGLU has been designed, which improves feature extraction capability in a lighter way. In addition, the model uses lightweight convolution instead of standard convolution to reduce the model's computation. Finally, the WIoU v3 loss function is used to make the model more focused on low-quality examples, thereby improving the model's object localization ability. Experimental results on the VisDrone2019 dataset show that MSUD-YOLO improves mAP50 by 8.5% compared with YOLOv10s. Concurrently, the overall model reduces parameters by 6.3%, verifying the model's effectiveness for object detection in UAV aerial images in complex environments. Furthermore, compared with multiple latest UAV object detection algorithms, our designed MSUD-YOLO offers higher detection accuracy and lower computational cost; e.g., mAP50 reaches 43.4%, but parameters are only 6.766 M.

Keywords: multiscale small object; object detection; UAV aerial image; YOLOv10s; deep learning; lightweight CNN

Academic Editor: Joaquín Martínez-Sánchez

Received: 11 May 2025
Revised: 6 June 2025
Accepted: 10 June 2025
Published: 13 June 2025

Citation: Zhao, X.; Zhang, H.; Zhang, W.; Ma, J.; Li, C.; Ding, Y.; Zhang, Z. MSUD-YOLO: A Novel Multiscale Small Object Detection Model for UAV Aerial Images. *Drones* **2025**, *9*, 429. https://doi.org/10.3390/drones9060429

1. Introduction

A UAV (unmanned aerial vehicle) is unmanned by integrated remote control devices and autonomous program control systems. UAVs are widely used in military and civilian fields [1,2] with their advantages of small volume, long endurance, high concealment and easy manipulation. In military aspects, UAVs can be used in battlefield reconnaissance, information confrontation, cluster cooperative operations and other fields. On the civilian side, UAVs are mainly used in agricultural crop monitoring, power maintenance, urban traffic supervision, search and rescue and other fields [3–6]. UAV aerial image object detection mainly utilizes computer vision algorithms to classify and detect objects in UAV aerial images, so as to determine the precise location of objects and extract their feature information. However, UAV aerial images are more vulnerable to intricate environmental factors—like occlusion, light and weather—than ordinary images. Moreover, when images are captured by a UAV, objects of different categories often vary in size, which

leads to the multiscale feature of objects in images and affects detection accuracy. Furthermore, UAV aerial images often contain a lot of overlapping and dense small objects, making object features less prominent, resulting in false detection and missing detection problems. Previous methods, such as early manual feature detection [7–9], relied on manual work to design and select features to a large extent, and their accuracy, objectivity, robustness and generalization were all restricted to a certain degree, ultimately resulting in unsatisfactory detection precision and speed, which affect practical application. In contrast, deep learning methods offer the benefits of strong adaptability, end-to-end learning, parallel processing and so on. Therefore, researchers are increasingly inclined to use deep learning algorithms for extracting object features from UAV aerial images to improve detection accuracy.

At present, there are two categories of primary deep learning object detection methods: two-stage and one-stage. Two-stage algorithms, including R-CNN [10], SPPNet [11], Fast R-CNN [12] and Faster R-CNN [13], initially generate candidate regions, and they subsequently classify and regress them using convolutional neural networks. While these algorithms exhibit remarkable precision and robustness in handing complex scenarios and small object detection, their main disadvantages lie in their high computational demand and inference time. In contrast, one-stage algorithms treat object detection as regression problems. They directly predict the bounding box and object class right from the input image without generating region proposals. This approach leads to much faster detection speeds and keeps the computational complexity significantly lower. The YOLO series algorithm [14] is among the most representative. YOLOv1 [15], as the first formal one-stage algorithm, was faster than R-CNN but not good at detecting near and small objects due to fewer grids divided by feature maps and problems in matching strategy design. YOLOv2 [16] improved upon YOLOv1, focusing on addressing the issues of low recall rates and inaccurate positioning. YOLOv3 [17] improved the shortcomings of the previous YOLO algorithm and introduced a new backbone network, Darknet-53, which combines residual connections more deeply and efficiently. YOLOv4 [18] improved the model detection performance for objects of different scales by combining technologies, such as the Cross-Stage Partial (CSP) network [19] and multipath feature fusion based on PANet [20]. YOLOv5 [21] followed the design philosophy of YOLOv3, enhanced the learning process by automatically adjusting the anchor frame during training, applied the CSP module to the neck, and replaced SPP with Spatial Pyramid Pool Fast (SPPF). Specifically designed for industrial applications, YOLOv6 [22] adopted EfficientRep, based on the RepVGG [23] structure, as its backbone network to achieve efficient feature extraction. In addition, the detection head adopts an efficient decoupled head structure, optimizing computation and memory consumption without compromising precision. YOLOv7 [24] introduced a new Extended Efficiency Layer Aggregation Network (E-ELAN) and adopted an adaptive anchor mechanism, enabling model to better adapt to objects of varying sizes and proportions, thereby boosting the detection performance for small objects. YOLOv8 [25] abandoned the YOLOv5's C3 structure, opting instead for the more abundant C2f structure of gradient flow. Furthermore, it adjusted the numbers of different channel numbers for models of different scales, significantly enhancing the overall efficiency of the model. YOLOv9 [26] was an improved version of YOLOv7, mainly introducing two innovative technologies: Programable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN). YOLOv10 [27], the first real-time end-to-end object detection algorithm in the YOLO series, inherited the network structure from YOLOv8. Unlike previous YOLO algorithms, YOLOv10 introduced a consistent dual assignments strategy for training YOLO without NMS, enabling faster and more efficient object detection. Compared to previous YOLO

models, YOLOv10 markedly boosts speed and precision, making it suitable for real-time object detection. It is also the main reason for this paper to design a new network architecture based on YOLOv10.

Researchers have applied YOLO-series algorithms to object detection tasks in UAV aerial images. Jawaharlalnehru et al. [28] developed an improved YOLO algorithm, which uses methods such as object box dimension clustering, pre-trained network classification and multiscale detection training to address the challenges of slow detection speed and missed detections for multiscale objects in UAV image object detection. Sahin and Ozer et al. [29] introduced a new YOLODrone algorithm based on YOLOv3. This algorithm increases the number of detection layers from three various scales to five and enhances the localization ability of small objects. However, the detection accuracy of small targets still needs to be improved. Koay et al. [30] developed the YOLO-RTUAV algorithm. They not only used DIoU-NMS to reduce suppression errors and missed detections but also used multiple 1×1 convolution to reduce the model complexity. Bao [31] et al. proposed GCL-YOLO, a lightweight YOLO model based on GhostConv. Their method significantly reduces the network's parameters and calculation, enabling it to be better deployed on unmanned aerial vehicle mobile terminals for real-time object detection. However, this model has not solved the problem of multiscale object detection. Wang et al. [32] improved a lightweight detection model MFPYOLO by optimizing YOLOv5. They designed a multi-input inverted residual block (MIRB) and introduced a convolutional block attention module (CBAM), significantly boosting the model's performance under multiscale variations and complex backgrounds. Wang et al. [33] designed a new algorithm, DSAA-YOLO, which employs higher-quality training data through the super resolution data enhancement method (SRDA) and the super resolution module DRSR, so as to solve the problem of effective feature extraction for small object and low bounding box localization accuracy. Chen et al. [34] improved YOLOv8 by designing the feature fusion module FFNB, which significantly improves the model's detection performance, but the parameters still need to be reduced. Li et al. [35] proposed LUD-YOLO, a lightweight UAV small object detection model, introducing a novel multiscale feature fusion mode to achieve higher-quality and smaller semantic gap feature fusion. However, this fusion mode also leads to a relatively high computational complexity. Qi et al. [36] proposed a multi-strategy feature enhancement YOLO model by improving YOLOv8, aiming to solve the small-scale problem of aerial images taken by drones, but the detection performance of small targets in complex environments still needs to be enhanced. Zhang et al. [37] proposed a lightweight feature enhancement, fusion and context-aware YOLO detector, which solves the problem of low detection accuracy of small objects in aerial images. Luo et al. [38] enhanced the feature extraction capability of the model by integrating the shuffle block algorithm and the multiscale extended attention (MSDA) mechanism in the backbone. Liu et al. [39] designed a real-time pedestrian recognition model based on YOLOv5 to enhance the real-time detection performance of edge devices. They used global channel pruning to reduce the number of parameters in the model. Although the aforementioned models have made some progress in improving object detection accuracy and making the algorithms lightweight, the detection accuracy for multiscale objects and small objects in complex environments still needs to be further improved. Furthermore, for embedded devices with limited resources, the computational efficiency and speed of the model also need to be further improved to meet the requirements of practical applications.

In view of the problems in object detection from UAV aerial images mentioned above, this article designs a multiscale small object detection model for UAVs based on YOLOv10s, namely MSUD (UAV Aerial Image Multiscale Small Object Detection)-YOLO. It aims to

address the issues of low detection accuracy for multiscale objects and small objects in UAV aerial images, as well as the complexity of detection model. The key contributions of this article are as follows:

1. Aiming at multiscale issues in UAV aerial images, we have improved the neck network using the ASF structure. This structure achieves comprehensive fusion of different scale features and enhances detection capability by optimizing the feature pyramid and the path aggregation network. Concurrently, adding a small object prediction head P2 into the head makes the model focus more on low-level features, thereby enhancing its sensitivity to small objects.
2. A novel feature extraction module CFormerCGLU is designed to balance the model's detection speed and precision. The design of this module not only boosts the model's feature extraction ability but also solves the problem of large memory usage and high computational cost of the attention mechanism. In addition, GSConv is used instead of standard convolution to decrease the model's computation.
3. WIoU v3 is introduced into the boundary box regression loss, adopting a reasonable gradient gain allocation method, which can dynamically optimize the weight of high- and low-quality anchor boxes at a loss, so that the model focuses on ordinary-quality anchor boxes, thereby enhancing its generalization ability and overall performance.
4. Extensive experiments conducted on the VisDrone2019 dataset have shown that MSUD-YOLO achieves the best balance between lightweight and detection accuracy. Compared with the baseline model, YOLOv10s, our model has excellent detection performance in multiscale object and small object detection. Furthermore, in terms of computational efficiency, MSUD-YOLO is also superior to several of the latest YOLO algorithms.

2. YOLOv10s Algorithm

YOLOv10, developed by researchers at Tsinghua University utilizing the Ultralytics Python software package (8.2.30), is a new version in the YOLO series of object detection algorithms. It improves its performance and versatility by improving the model architecture and eliminating non-maximum suppression (NMS). Considering the hardware requirements of UAV object detection, this paper chooses YOLOv10 version YOLOv10s, with fewer parameters, as the basic model, and its structure is illustrated in Figure 1. Firstly, YOLOv10s uses a consistent dual assignment strategy for NMS-free training of YOLO, which results in both excellent performance and high reasoning efficiency. In addition, YOLOv10s comprehensively optimizes each module of the YOLO algorithm from the point of view of efficiency and accuracy, greatly reducing computational complexity and enhancing detection capabilities.

2.1. Consistent Dual Assignments for NMS-Free Training

During network training, YOLO typically uses a label assignment strategy to allocate multiple positive examples to each instance. The one-to-many assignment strategy provides rich supervisory signals, which is helpful for better learning of the model. However, it requires YOLO to rely on NMS post-processing, leading to poor reasoning efficiency. Therefore, YOLOv10s utilizes an NMS-free training strategy, as shown in Figure 2, to achieve high efficiency and good performance through using dual label assignments and consistent matching metrics.

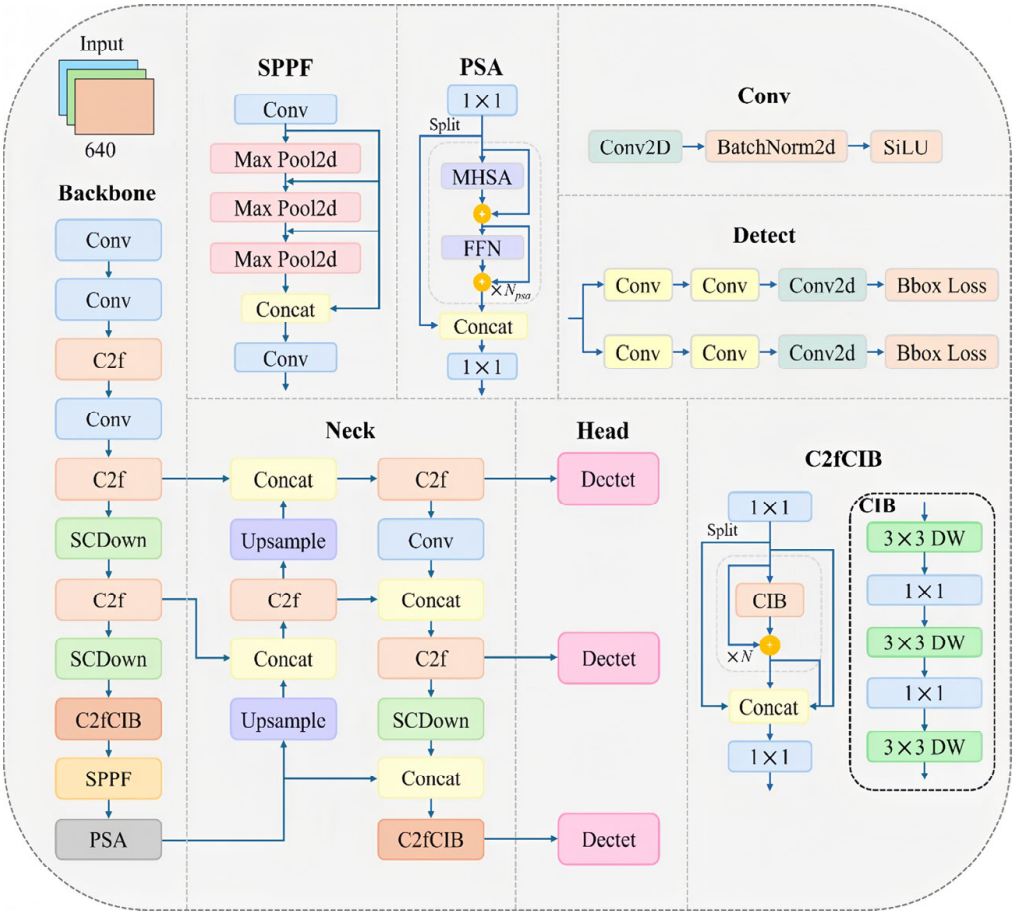

Figure 1. YOLOv10s algorithm structure.

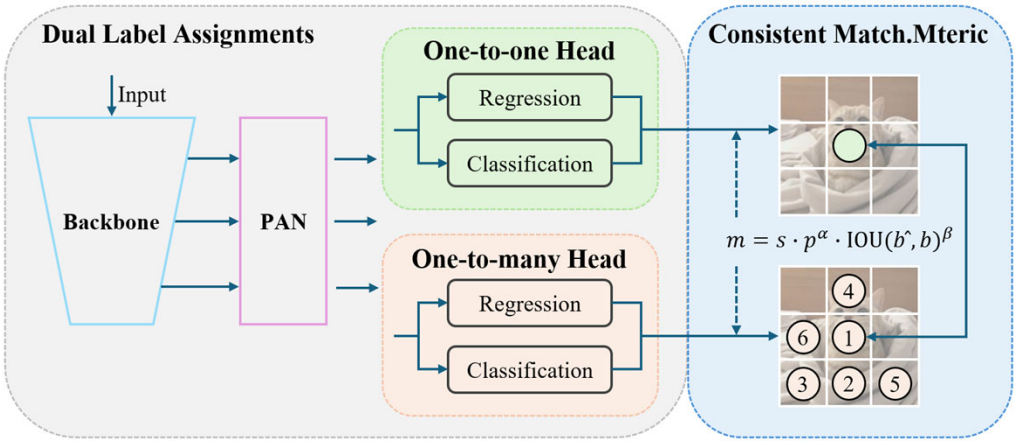

Figure 2. Consistent dual assignments for NMS-free training.

2.1.1. Dual Label Assignments

Differing from one-to-many assignments, one-to-one matching assignments have only a positive sample per true value and do not require post-processing by NMS. However, this results in weak supervision, which affects accuracy and the speed of convergence. Fortunately, a one-to-many assignment can compensate for this shortcoming. So, YOLOv10s uses a dual label assignment strategy, aiming to integrate the advantages of the two assignment methods. During the training phase, the two heads are optimized in conjunction with the main model, so that the backbone and neck fully benefit from the abundant supervisory information provided by one-to-many assignments. In the inference phase, only one-to-

one headers are used for prediction, enabling end-to-end deployment of YOLO without additional inference costs.

2.1.2. Consistent Matching Metric

One-to-one and one-to-many methods utilize metrics to quantitatively evaluate the level of agreement between predictions and instances during assignments. To achieve consistency between the two branches in the training process, a uniform matching index, the consistent matching metric, is adopted. The formula is as follows:

$$m = s \cdot p^{\alpha} \cdot \mathrm{IOU}\left(\hat{b}, b\right)^{\beta}, \tag{1}$$

where p is classification score, and $\hat{b}$ and b represent the predicted and true boundary box, respectively. s denotes the spatial prior indicating whether the predicted anchor is within the instance. α and β are two important hyperparameters that balance the effects of semantic prediction tasks and position regression tasks. Through consistent matching measures, the optimization direction of one head and one head are consistent, thereby enhancing the model's performance.

2.2. Efficiency and Accuracy Model Design

2.2.1. SCDown

YOLO typically employs a 3×3 standard convolution with a stride of 2 for spatial downsampling (from $H \times W$ to $\frac{H}{2} \times \frac{W}{2}$) and channel transformation (from C to $2C$). This leads to redundant computational costs $\frac{9}{2}HWC^2$ and the parameters $18C^2$. Therefore, YOLOv10s adopts the operations of spatial reduction and channel expansion for downsampling, which diminishes both computational cost to $\frac{9}{2}HWC^2$ and the parameters to $2C^2 + 18C$. At the same time, this maximizes the retention of the feature information in the downsampling process, thus achieving real-time performance while maintaining accuracy.

2.2.2. C2fCIB

YOLO typically uses the same basic building blocks across all network layers, such as the bottleneck block in YOLOv8. However, using the same basic building blocks tends to exhibit a lot of redundancy in the deep stages of the network and in large models. To this end, YOLOv10s has designed a compact inverted block (CIB) structure that aims to reduce the complexity of redundant stages through a compact architectural design. As shown in Figure 1, CIB employs cheap deep convolution for spatial mixing and cost-effective point convolution for channel mixing. It can be used as an efficient basic building block, embedded in ELAN structures. Furthermore, a rank-guided block allocation strategy is used to sort all phases of the model by intrinsic rank from lowest to highest and then check the performance change of the base block that replaces each phase with CIB. After comparison and experiment, YOLOv10s has the best performance in the deepest stage P5 (20×20) using the CIB structure.

2.2.3. Partial Self-Attention (PSA)

The principle of the PSA module is illustrated in Figure 1. Self-attention has been widely used in various visual tasks because of its remarkable global modeling ability. However, this often results in high computational complexity and memory footprint. To solve this problem, YOLOv10s adopts an efficient Partial Self-Attention (PSA) module. Specifically, the feature is first evenly split into two parts through 1×1 convolution. Then, only a part of these features is input into the NPSA block, which consists of a Multi-Head Self-Attention module (MHSA) and a Feed-Forward Network (FFN). The two parts of

the features are then joined and fused by 1 × 1 convolution. Secondly, the dimensions of the query and key in MHSA are set to half of the value, and LayerNorm is replaced with BatchNorm to speed up reasoning. To avoid excessive overhead caused by the quadratic complexity of attention, PSA is only placed after the P4 layer with the lowest resolution.

3. Proposed MSUD-YOLO

3.1. Overall Framework

Figure 3 demonstrates the MSUD-YOLO model proposed in this article, which is optimized and improved on the basis of YOLOv10s. Firstly, the ASF structure is incorporated into the neck of the network to boost the model's capability for multiscale feature fusion. Furthermore, the small object prediction head P2 is added on the shallow feature map, so that the model leverages more local information, which improves the detection precision of small objects. Secondly, a novel feature extraction module named CFormerCGLU is designed in the feature extraction network. The CFormerCGLU module leverages the attention mechanism to enhance the model feature extraction capabilities, while also solving the problem of excessive memory consumption and computational cost associated with the attention mechanism. In the meantime, GSConv is used instead of standard convolution to decrease the model computational load. Finally, WIoU v3 is introduced as the loss function, which offers a wise gradient gain allocation method. This method enhances its generalization ability and overall performance. The following will introduce the structural framework and principle of each module.

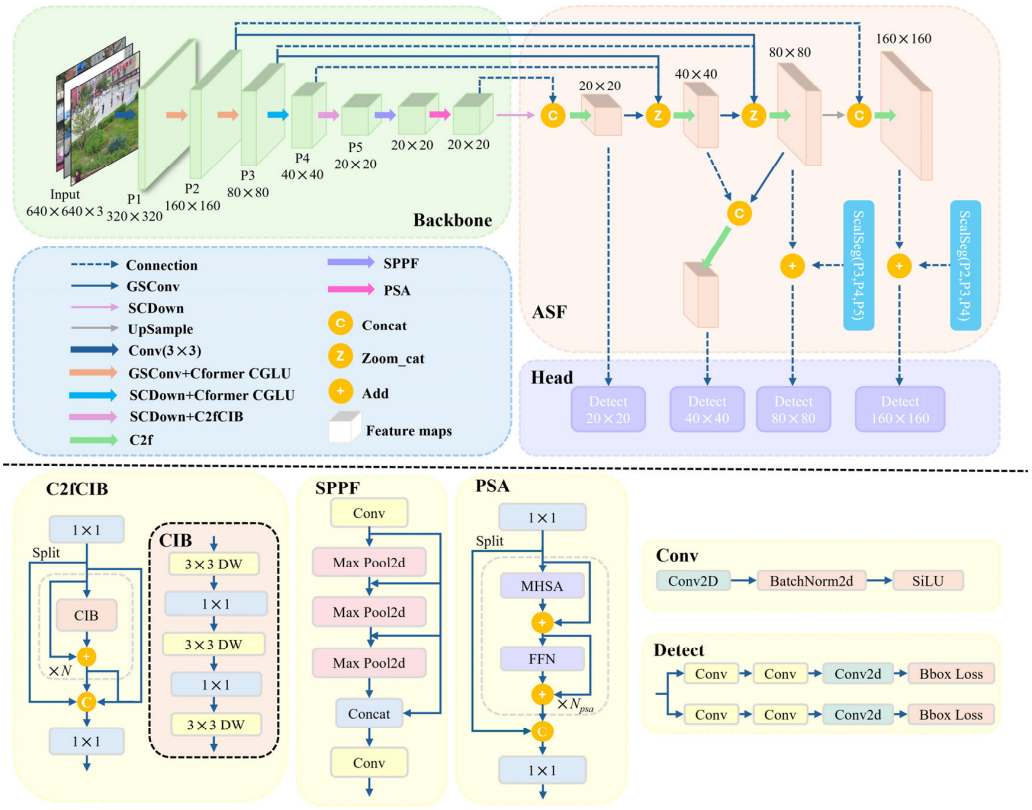

Figure 3. MSUD-YOLO model framework.

3.2. ASF Neck Structure

To solve the multiscale problem in UAV aerial images, we introduced the ASF (Attentional Scale Sequence Fusion) structure. ASF [40] primarily consists of the Scale Sequence Feature Fusion (SSFF) module, Triple Feature Encoding (TFE) module and Channel and Position Attention Mechanism (CPAM), as illustrated in Figure 4.

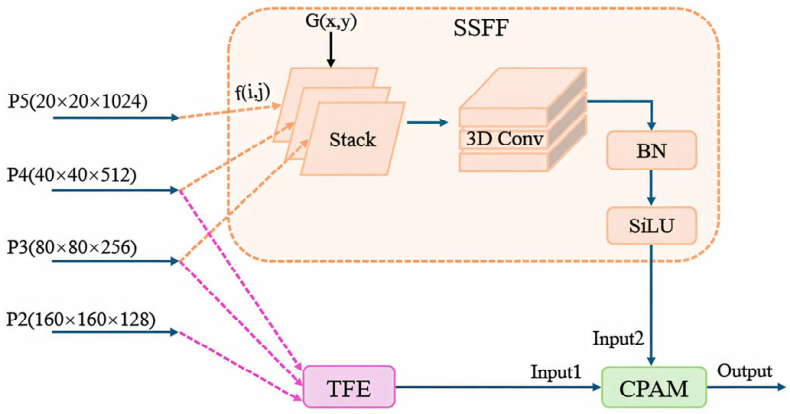

Figure 4. Attentional Scale Sequence Fusion structure.

3.2.1. Scale Sequence Feature Fusion (SSFF) Module

Aiming at multiscale issues in UAV aerial images, current feature fusion methods mostly adopt the accumulation or superposition of feature pyramid structure for fusion. However, these structures fail to fully take advantage of correlations among all feature maps. To address this, a novel Scale Sequence Feature Fusion (SSFF) module (the ScaleSeq block in Figure 3) is introduced to the structure, which better integrates the multiscale feature map, thereby enhancing the multiscale information extraction ability of the network.

Figure 4 depicts the specific process of the SSFF module. First, the feature maps P3, P4 and P5 generated from the backbone are convolved with a series of Gaussian kernels with incrementally increasing standard deviations [41–43], aiming to keep the scale of the feature maps consistent. The following formula is used:

$$L_\sigma(x,y) = f(x,y) * G_\sigma(x,y), \tag{2}$$

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}, \tag{3}$$

where $L_\sigma(x,y)$ is a function of the scale of an image, σ is the scale parameter of the Gaussian function, f denotes a 2D feature map, and $G_\sigma(x,y)$ is a variable-scale Gaussian function.

Subsequently, feature maps are stacked horizontally, and 3D convolution is used to extract their scale sequence features. After 3D convolution, the data is normalized using batch normalization (BN) to stabilize the data distribution and accelerating the training process. Finally, the model's nonlinear processing capability is enhanced by the SiLU activation function. This process effectively combines the feature information at different scales, enhancing understanding and expression capabilities for data.

3.2.2. Triple Feature Encoding (TFE) Module

For the problem of small object recognition with dense overlapping in UAV images, we can enlarge images to refer to and contrast alterations in shape or appearance across various scales. Because of the different dimensions of the varying feature layers in the backbone, the traditional FPN fusion mechanism can only upsample small-size feature maps and split or add them to the features of the previous layer, neglecting the rich details in the larger-scale feature layer. TFE module can split large, medium and small features, integrate them into large-scale feature maps, and amplify features to capture detailed feature information more comprehensively. This module corresponds to the Zoom_cat block in Figure 3.

Figure 5 shows the structural principle of the TFE module. Before encoding features, a composite layer (ConvBNSiLU) is first applied for preprocessing to adjust feature channel number, aligning them with the main scale feature. For the Large feature map, we first use

a convolutional module to unify the channel number to 1C. Subsequently, max pooling and average pooling are employed for downsampling, helping decrease the spatial dimension of features, thereby enhancing the robustness of the network to spatial changes and shifts of input images. For the Small feature map, we similarly adjust the channel number to 1C using a convolutional module. Secondly, nearest-neighbor interpolation is employed in upsampling to match the "Medium" feature map. This helps preserve local feature information and prevents small object feature loss. Lastly, the three feature maps of the same dimension are convolved and connected according to the following formula:

$$F_{TFE} = concat(F_l, F_m, F_s), \tag{4}$$

where F_{TFE} represents output feature map by the TFE module, and *Concat* denotes the connection operation. F_l, F_m and F_s represent large-, medium- and small-size feature maps, respectively.

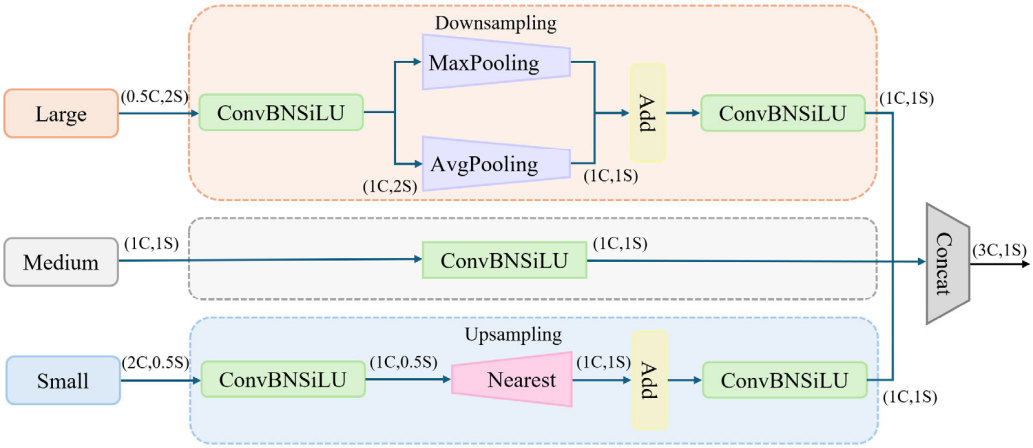

Figure 5. Triple feature encoding module.

3.2.3. Channel and Position Attention Mechanism (CPAM)

The CPAM (the Add block in Figure 3) can extract key feature information across channels, integrating detailed and multiscale feature information of SSFF and TFE outputs, using channel attention to enhance object features and suppress irrelevant background, while position attention helps the model accurately locate and distinguish key areas in the image in spatial position. This mechanism enables the model to better perform object detection and recognition in a small-scale and complex environment.

The structure of the CPAM is illustrated in Figure 6, which consists of a channel attention network and location attention network. In channel attention networks, the input feature maps are first pooled averagely to retain channel information while reducing spatial dimension. Then, two fully connected layers and a nonlinear Sigmoid function are used to generate channel weights, which are then applied to the original feature map to highlight important spatial areas. In the position attention network, the position attention mechanism firstly applies average pooling on input feature maps along the horizontal (p_w) and vertical (p_h) axes to preserve the spatial structure information of feature maps. The pooling formula is as follows:

$$p_w(i) = \frac{1}{H} \sum_{0 \le j \le H} E(i, j), \tag{5}$$

$$p_h(j) = \frac{1}{W} \sum_{0 \le i \le W} E(i, j), \tag{6}$$

where W and H are the width and height of the input feature map, respectively. $E\ (i, j)$ are the values at the position (i, j) of the input feature map.

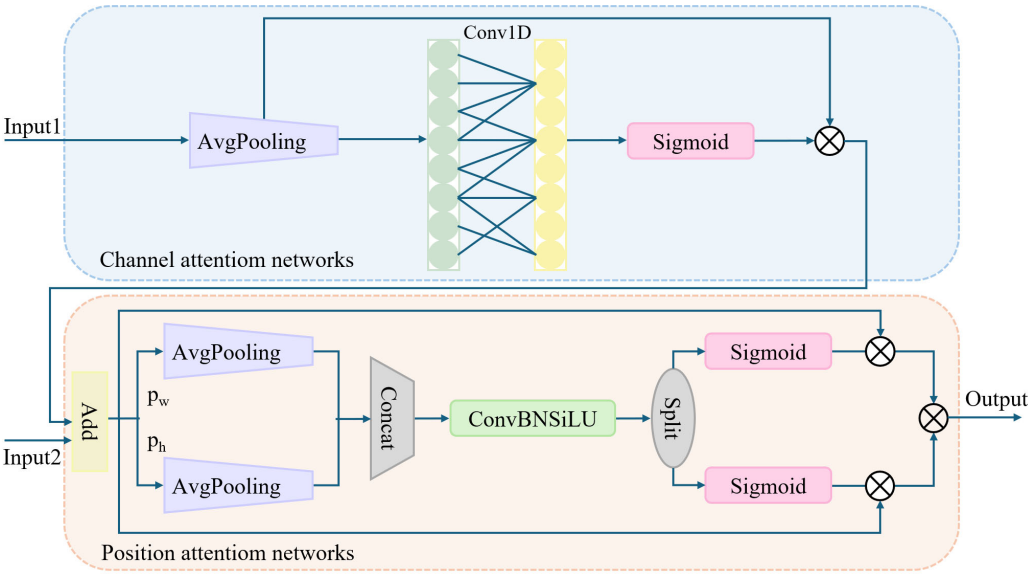

Figure 6. Channel and position attention mechanism module.

Then, when generating position attention coordinates, concatenation and convolution operations are performed on the horizontal and vertical axes:

$$P(a_w, a_h) = Conv[Concat(p_w, p_h)], \tag{7}$$

where a_w and a_h are position attention coordinates, $P(a_w, a_h)$ represents the output of the position attention coordinates, *Conv* represents 1×1 convolution, and *Concat* means concatenation.

Finally, the attention features are split into two parts to calculate the position weights by the Sigmoid function. When splitting the attention features, the following formulas generate pairs of feature maps related to position:

$$s_w = Split(a_w), \tag{8}$$

$$s_h = Split(a_h). \tag{9}$$

where s_w and s_h are the width and height of the splitting output, respectively.

The final formula of the CPAM is as follows:

$$F_{CPAM} = E \times s_w \times s_h, \tag{10}$$

where E denotes the weight matrix of the channel and position attentions.

This integrated channel and position attention mechanism makes models focus on relevant channels and spatial information more accurately, significantly improving processing efficiency and model performance.

3.3. Tiny Prediction Head-P2

YOLOv10 has three prediction heads (80×80, 40×40, 20×20), which have a larger receptive field for medium- and high-level features, focusing more on semantic information of features. However, its ability to express positional and detailed information is limited, leading to low-level features easily being ignored. Therefore, selecting a suitable receptive field size or considering multiscale receptive fields can effectively preserve more local feature information of small objects when extracting features of small objects. Most of the objects in UAV aerial images are tiny, and primordial multiscale detection structures often miss such object.

To solve this problem, this article adds a new type of tiny prediction head, P2, as shown in Figure 3, which increases the resolution of the detection feature map (160 × 160) for detecting small objects larger than 4 × 4 pixels in value. Small detection pixels are capable of capturing richer location information about objects, which is essential to enhance the detection accuracy of small objects and provide valuable inputs to other layers during feature fusion. The newly introduced P2 concentrates on low-level features, thus improving the model's sensitivity to small objects.

3.4. GSConv Module

GSConv (Grouped Spatial Convolution) [44] is a lightweight convolution technique, aiming at reducing computation while maintaining sufficient precision. The GSConv principle is illustrated in Figure 7. First, we input the downsampled feature map of a standard convolution (SC), then Depth-Separable Convolution (DSC) is utilized to the output of the previous step. The results from these two steps are concatenated, and we finally use shuffling to infiltrate the features produced by SC into each section of the features produced by DSC. Shuffling represents an evenly mixed strategy. This approach facilitates the integration of SC information into the output of DSC through the uniform exchange of local features across different channels, and the entire process does not require cumbersome steps. Compared with traditional convolution techniques, GSConv significantly enhances the nonlinear representation of lightweight detectors by combining the Depth-Separable Convolution layer and a shuffle layer. This design strengthens the network's ability to handle complex feature while keeping low computing requirements.

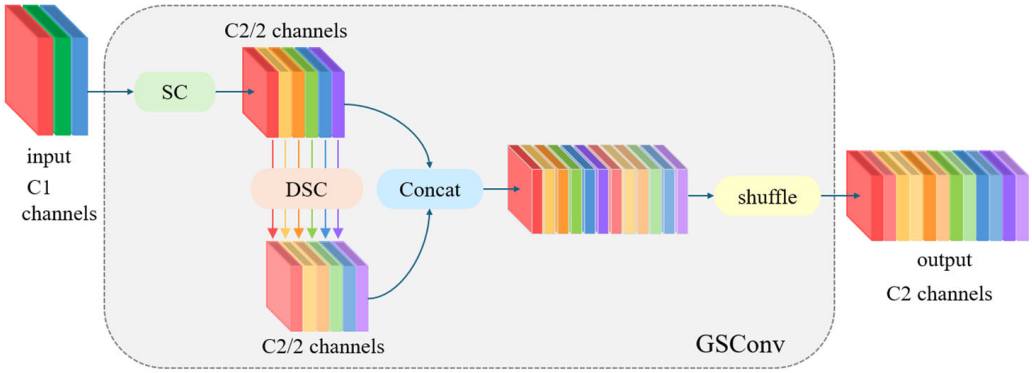

Figure 7. GSConv module.

3.5. CFormerCGLU Module

The backbone of YOLOv10 mainly uses the C2f module for feature extraction. The C2f module fully takes into account abundant gradient flow information, achieving more power feature extraction capability. Nevertheless, C2f is essentially a local operator and still built on convolutional neural networks. By contrast, a notable attribute of attention is its global receptive field, which enables it to effectively capture long-range dependencies between features. Therefore, we propose the new feature extraction module CFormerCGLU in the backbone, which boosts the model's feature extraction ability, while addressing the issues of large memory usage and high computational cost of the attention mechanism.

ConvFormer Block [45] is designed to allow CNN branches to learn how to extract high-quality semantic information from transformer branches, only using convolution operations to capture long-distance context information just like Transformer blocks. The ConvFormer Block structure resembles that of a typical Transformer encoder, as shown on the left in Figure 8, detailed below:

$$y = Norm(f + FFN(f)), f = Norm(x + ConvAttention(x)), \tag{11}$$

where $Norm(\cdot)$ denotes batch normalization, and f, x and y represent the hidden feature, input and output, respectively.

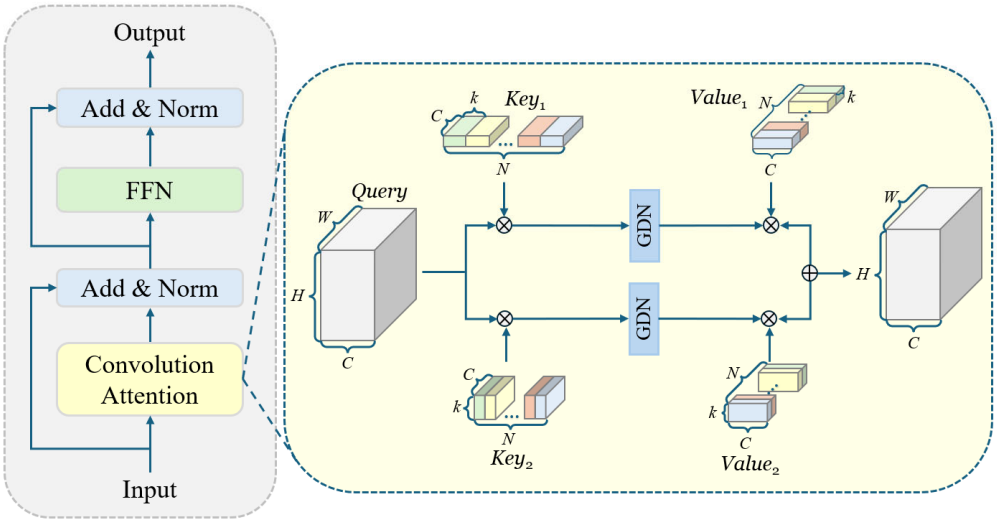

Figure 8. ConvFormer module (**left**) and the detail of convolutional attention (**right**). GDN denotes grouped double normalization. $\otimes$ denotes convolution operation, $\oplus$ denotes add.

Convolutional GLU (ConvGLU) [46] is a channel mixer with gated channel attention that closes the gap between the GLU and SE mechanisms. It allows each token to gain channel attention based on its nearest neighbor image feature, which enhances local modeling capability and robustness. Figure 9 (right) depicts the structure of the CGLU. Nevertheless, the weight adjustment between channels in the CGLU is implicit and lacks the ability to model global context information. ConvFormer makes up for the deficiency. In addition, when keeping the parameter volume consistent with the Convolutional Feed-Forward (CFFN) module with an expansion ratio of R and a convolution kernel size of $k \times k$, the computational complexity of ConvGLU is $2RHWC^2 + \frac{2}{3}RHWCk^2$, which is less than the $2RHWC^2 + RHWCk^2$ of CFFN (Given an input $X \in \mathbb{R}^{C \times H \times W}$). Therefore, we integrate ConvFormer and the CGLU to achieve the attention of the channel mixer with less parameter and computational complexity, thus making the model more effectively extract global features.

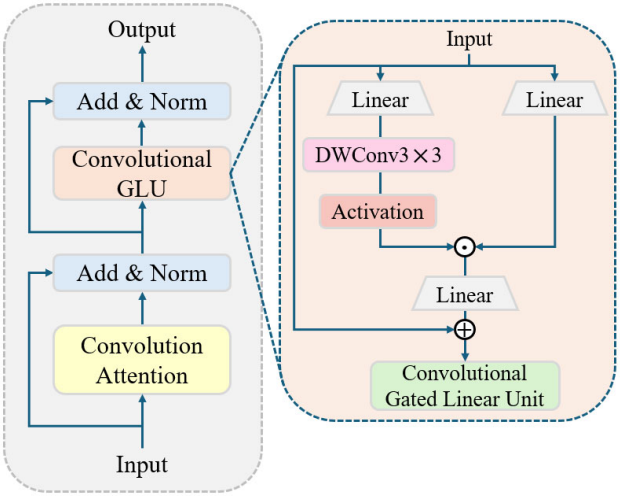

Figure 9. CFormerCGLU (**left**) and detail of CGLU (**right**).

3.6. WIoU v3

In UAV aerial image object detection tasks, small objects constitute the majority. Therefore, a suitable loss function is crucial for enhancing detection accuracy. YOLOv10 employs DFL and CIoU for bounding box regression loss computation, yet CIoU has the following defects: Firstly, CIoU falls short in addressing the problem of sample imbalance. Secondly, CIoU utilizes the aspect ratio as a penalty factor, but in some cases, this penalty factor may not fully reflect the true difference between the predicted box and the ground truth box. Thirdly, the CIoU calculation formula involves inverse trigonometric functions, increasing the model's computational complexity. Currently, mainstream loss functions (such as EIoU [47], SIoU [48], etc.) adopt a static focusing mechanism, whereas WIoU considers factors such as overlapping area, central point distance, aspect ratio and so on, while introducing a dynamic non-monotonic focusing mechanism. The three WIoU versions [49] are described below.

WIoU v1 incorporates distance as a key attention standard. When there is a certain degree of overlap between the object box and the predicted box, decreasing the penalty for geometric factors significantly enhances the model's generalization capability. The WIoU v1 formula is as follows:

$$\mathcal{L}_{WIoU\ v1} = \mathcal{R}_{WIoU}\mathcal{L}_{IoU}, \tag{12}$$

$$\mathcal{L}_{IoU} = 1 - IoU, \tag{13}$$

$$\mathcal{R}_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{\left(W_g^2 + H_g^2\right)^*}\right). \tag{14}$$

where (x_{gt}, y_{gt}) is the true box center coordinate, and W_g and H_g are the smallest enclosing box size, while * denotes that W_g and H_g are detached from the computational map.

WIou v2 reduces the proportion of simple samples in the loss calculation by applying the monotonic focusing coefficient $\mathcal{L}_{IoU}^{\gamma*}$ to WIou v1. The WIou v2 formula is as follows:

$$\mathcal{L}_{WIoU\ v2} = \left(\frac{\mathcal{L}_{IoU}^*}{\mathcal{L}_{IoU}}\right)^{\gamma}\mathcal{L}_{WIoU\ v1}, \gamma > 0, \tag{15}$$

where the gradient gain is $r = \mathcal{L}_{IoU}^{\gamma*} \in [0, 1]$.

WIou v3 introduces an outlier degree β for measuring anchor box quality and constructs a non-monotonic focusing coefficient based on β. The smaller β denotes high-quality anchor boxes and a small r is allocated to it, allowing the bounding box regression (BBR) to focus on ordinary-quality anchor boxes. The larger β denotes low-quality anchor boxes and a smaller r is allocated to it, preventing large harmful gradients from low-quality examples. This allocation method enables the model to focus more on ordinary quality anchor boxes, which enhances the model's localization ability. The WIou v3 formula is as follows:

$$\mathcal{L}_{WIoU\ v3} = r\mathcal{L}_{WIoU\ v1}, r = \frac{\beta}{\delta\alpha^{\beta-\delta}}, \tag{16}$$

$$\beta = \frac{\mathcal{L}_{IoU}^*}{\mathcal{L}_{IoU}} \in [0, +\infty). \tag{17}$$

where δ and α are hyperparameters and can be adjusted to adapt to different models.

Therefore, we use WioU v3 as a loss function. WioU v3 incorporates the advantages of EIoU and SIoU, comprehensively taking into account factors such as overlapping area, central point distance, aspect ratio and so on. Additionally, WIoU v3 employs a dynamic non-monotonic strategy to assess the quality of anchor box, addressing the issue of bal-

ancing BBR between samples of good and poor quality. For object detection tasks in UAV aerial images, the high proportion of small objects increases detection challenges, while WIoU v3 can dynamically adjust the loss weights of small objects, enhancing the model's generalization capability and comprehensive performance.

4. Experiment

4.1. Dataset

The VisDrone2019 dataset [50], collected and published by Tianjin University, is a large-scale dataset of aerial images captured by drone, containing 288 video clips and 10,209 static images. This dataset covers a variety of scenarios including cities, rural areas, highways, construction sites, etc. It also includes data gathered in diverse conditions, such as different weather conditions, different object scales and different UAV platforms (i.e., different UAV models), enabling algorithms to be more adaptable and robust when facing various challenges. Additionally, the dataset defines ten common object categories, such as people, cars, trucks, and motors, etc., which have important research value for tasks such as intelligent traffic management, pedestrian recognition and vehicle tracking.

Figure 10 shows part of the images from the dataset (a), the count of tags (b) and the size of the tags (c). These figures illustrate the following characteristics and difficulties of the dataset in object detection and recognition. As illustrated in Figure 10a, the objects to be detected are numerous, small and easily confused. In the example scenario, it is difficult to distinguish between the pedestrian category and the people category. From the perspective of drone photography, the scale of objects varies greatly, and mutual occlusion is more pronounced, posing a great challenge to the performance of the object detection algorithm. Secondly, from Figure 10b, we can see that there is a significant difference in tag number among various categories. The tag number for the car category reached 144,867, while the awning–tricycle category was only 3246. The imbalanced examples among categories proposes higher requirements for the robustness of detection models. Finally, considering the size distribution of anchor points comprehensively, as illustrated in Figure 10c, in addition to the large number of large-size anchors in some categories, most anchors are within the range of 150 $\times$ 150 pixels, particularly the small objects, which are concentrated within the 50 $\times$ 50 pixel range. Therefore, this dataset puts forward higher requirements on the model's ability for detecting small objects.

The whole dataset is partitioned into a training, validation and testing set, as shown in Table 1. Specifically, the training set contains 6471 images, which are utilized for model learning and parameter adjustment; the validation set contains 548 images, utilized for the preliminary evaluation of model performance and hyperparameter tuning; and the testing set consists of 1610 images, employed for the final assessment of the model's generalization ability and detection precision.

Table 1. The division of the dataset.

	Image Number	Tags Number
All	8629	457,063
Train set	6471	343,205
Val set	548	38,756
Test set	1610	75,102

4.2. Experimental Environment and Experimental Parameters

To evaluate the efficacy of MSUD-YOLO in UAV aerial image detection tasks, ablation experiments on the MSUD-YOLO model and comparison experiments with other similar algorithms are designed based on the VisDrone2019 dataset. To ensure the repeatability

and accuracy of the experiment, Tables 2 and 3 provide the version and configuration information of the environment required for the experiment, along with the related parameter settings throughout the training phase.

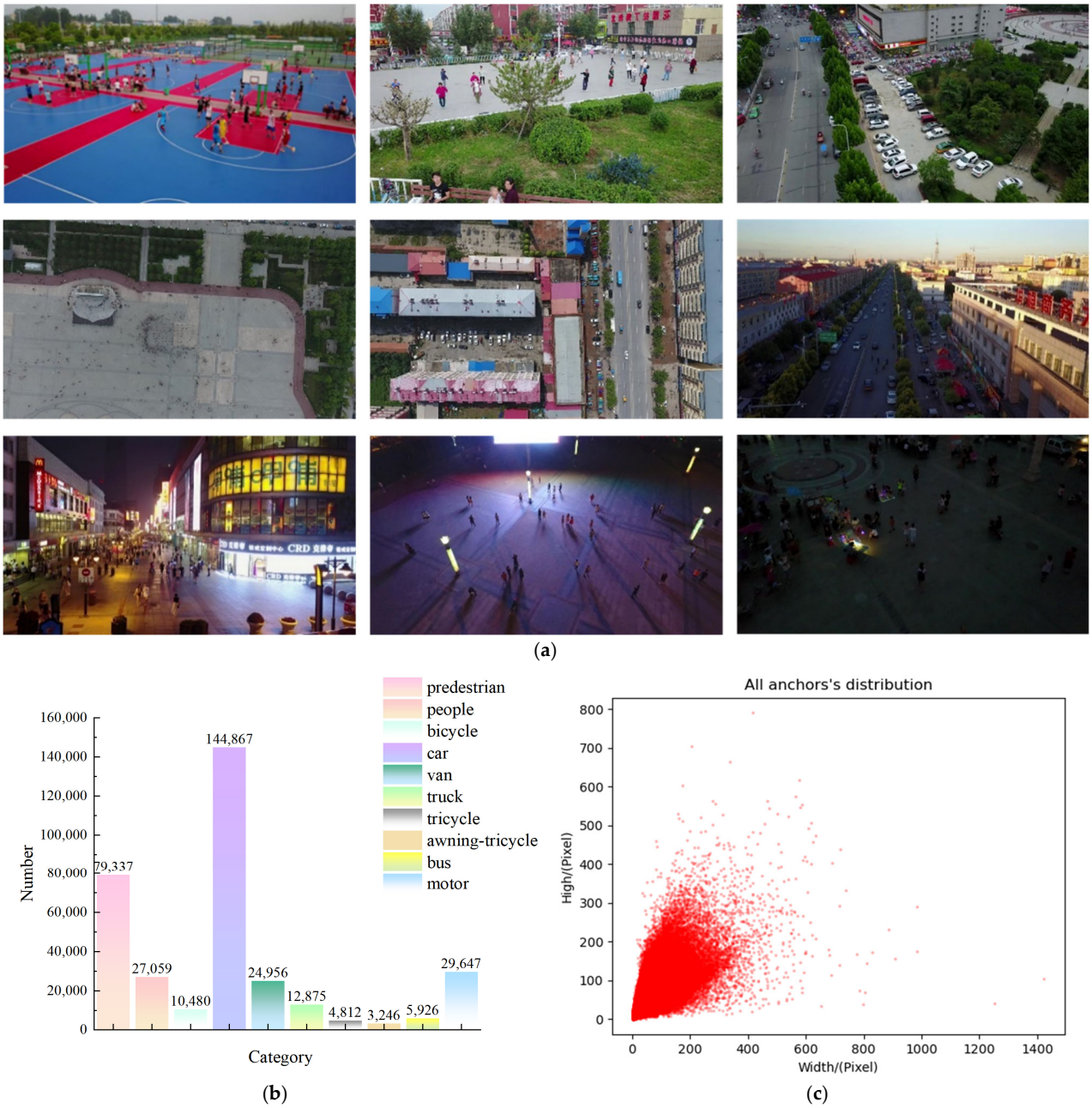

Figure 10. Analysis of VisDrone2019 dataset. (**a**) Part of the dataset images. (**b**) Distribution of tag quantity. (**c**) Overall distribution of anchor sizes.

Table 2. Configuration experimental environment.

Environment	Parameters
GPU	Intel(R) Xeon(R) Platinum 8488C
CPU	NVIDIA A100
GPU memory size	80 G
Operating system	Win 10
Language	Python 3.8.20
Frame	Pytorch 2.4.1
CUDA version	Cuda 12.1

Table 3. Training parameters setting.

Parameters	Setup
Epochs	300
Input image size	640 × 640
Batch size	16
Optimizer	SGD
Initial learning rate	0.01
Final learning rate	0.0001

4.3. Assessment Indicators

This article utilizes precision (P), recall (R), mAP (mean average precision), parameters, FPS, model size and time (inference latency time) as evaluation metrics to assess the model's performance. Among them, inference latency represents the time spent by the model in processing each frame of the image, including the time of preprocessing, inference and postprocessing. The calculation formula of assessment indicators is as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{18}$$

$$Recall = \frac{TP}{TP + FN}, \tag{19}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N}(P(R)dR)_i. \tag{20}$$

where TP, FP and FN denote true positive, false positive and false negative, respectively, and N indicates the number of categories.

4.4. Ablation Experiment

This article first introduces the ASF structure and P2 head based on YOLOv10s and then utilizes the GSConv and CFormerCGLU modules to lighten the network, striving to decrease the network parameters and complexity without sacrificing precision. Finally, the WIoU v3 loss function is introduced to enhance overall detection performance of the model. To verify the validity of all proposed improvements, we used the Visdrone2019 dataset to assess the influence of each improvement unit on the model on YOLOv10s in turn. This ablation results of the proposed MSUD-YOLO are presented in Table 4 and Figure 11, with the superior results appearing in boldface.

Table 4. Ablation results.

ASF	P2	GSConv	CFormer-CGLU	WIoU v3	P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	FPS	Params /Million	Model Size /MB	Time (ms/Frame)
					51.4	38.9	40.0	23.7	131.5	7.222	15.8	11.1
√					52.0	38.9	41.0	24.3	188.9	7.379	16.1	9.3
	√				52.6	42.2	43.9	26.3	93.3	7.409	16.3	10.7
		√			50.3	38.6	39.5	23.5	**229.4**	7.106	15.6	**6.9**
			√		49.9	38.9	39.6	23.4	132.7	6.768	**14.9**	10.0
				√	50.5	39.7	40.2	24.0	131.5	7.222	15.8	9.9
√	√				52.6	42.5	**44.1**	**26.4**	158.0	7.435	16.3	9.7
√	√	√			52.1	**42.6**	43.6	26.2	157.0	7.226	15.9	10.9
√	√	√	√		51.0	41.5	42.8	25.5	131.4	6.766	15.1	11.5
√	√	√	√	√	**53.0**	42.0	43.4	25.6	134	**6.766**	15.1	11.3

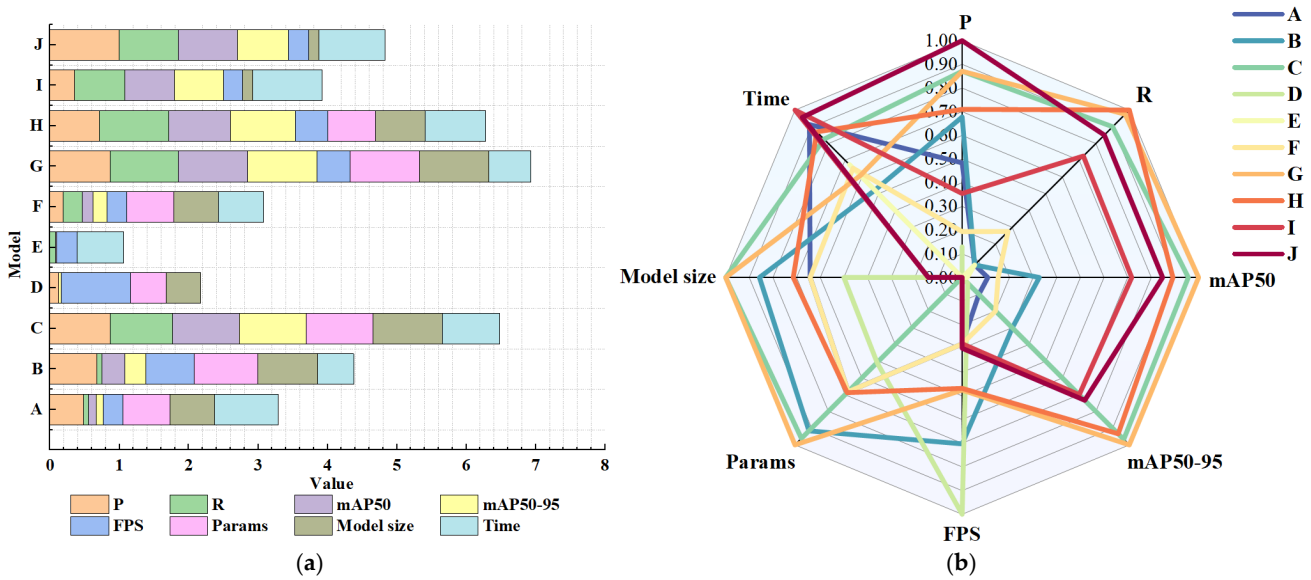

Figure 11. Normalized effect diagram of all indexes. (**a**) Normalization histogram of ablation experiment. (**b**) Performance diagram of ablation experiment.

It can be seen from Table 4 that as ASF and P2 are added into the baseline model separately, the accuracy metrics P, R, mAP50 and mAP50-95 exhibit a consistent upward tendency. That is, ASF can ensure the multi-dimensional fusion quality of feature maps of different scales, while increasing the attention of the model to multiscale features and detailed features in complex backgrounds. To better illustrate the contribution of the ASF structure, we provide a more detailed analysis, as shown in Table 5. The addition of layer P2 makes the model more focused on low-level features, thereby enhancing its sensitivity to small objects. Additionally, the design of the GSConv module and CFormerCGLU module enables the model to obtain excellent accuracy while reducing parameters and model size. Finally, WIoU v3 utilizes its dynamic non-monotonic approach to make our model more focused on low-quality samples, leading to a boost in model performance, with P, R and mAP50 improved by 3.9%, 1.2% and 1.4%, respectively, and FPS improved by 2.6 frames per second.

Table 5. In the ablation study of ASF, "ASF w/o TEF and CPAM" indicates the removal of the TEF and CPAM module, and "ASF w/o SSFF and CPAM" indicates the removal of the SSFF and CPAM module, while ASF w/o CPAM indicates that only the CPAM module is removed.

Model	P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	Params/Million
ASF w/o TEF and CPAM	51.0	38.8	40.2	23.7	7.337
ASF w/o SSFF and CPAM	51.3	39.2	40.7	24.0	7.263
ASF w/o CPAM	51.6	38.7	40.8	24.2	7.365
ASF	52.0	38.9	41.0	24.3	7.379

Overall, the optimized model parameters and model size outperformed the baseline model, with mAP50 improving by 8.5% and FPS improving by 2.5 frames per second. Among all the improvement strategies, P and parameters have the best performance. It shows that this suggested strategy is effective when the detection scene needs to consider both precision and speed.

In addition, Figure 11 intuitively presents the performance of each indicator of MSUD-YOLO and the comprehensive performance of the entire model. (All data in the figure are normalized. The closer the parameters of model size and time indexes are closer to 0, the better. The closer P, R, mAP50, mAP50-95 and FPS are to 1, the better.) A to J denote

the ten models in Table 3, respectively. It is clear from Figure 11a that while the improved model integrating ASF and P2 boasts the most outstanding detection precision, it leaves room for enhancement regarding its parameters and model size. Model J proposed in this article demonstrates the most balanced performance results. In particular, the radar chart in Figure 11b illustrates that model J has the best overall performance.

4.5. Compare Experiment

To further evaluate the superiority of MSUD-YOLO, this paper selects to compare it against the classical two-stage detection algorithm Faster R-CNN, the one-stage detection algorithm CenterNet [51] and several enhanced versions of YOLO. The latter includes YOLOv3, YOLOv4s, YOLOv5s, YOLOv8s, ASF-YOLO, MFFCI-YOLOv8 [52], LUD-YOLO [35] and YOLOv10s. Table 6 summarizes the experimental results of 11 comparison algorithms, including the proposed MSUD-YOLO in this paper, and the optimal outcomes appear in boldface. All experimental models were trained without setting pre-training weights, which can effectively avoid overfitting during training. Due to inconsistencies in experimental environments and hardware conditions among some models, leading to unaligned comparison dimensions such as FPS metrics, etc. we therefore only use the parameters and mAP50 (for 10 categories) data of each model for comparison in our experiments to better reflect the experimental data under the same environment. On the whole, these two indexes can reflect the diversity in precision and complexity of each model.

Table 6. Performance comparison between MSUD-YOLO and other algorithms.

Model	Params /Million	mAP50 (%)										
		Pedestrain	People	Bicycle	Car	Van	Truck	Tricycle	ATricycle	Bus	Motor	All
Faster R-CNN	41.100	21.4	15.6	6.7	51.7	29.5	19.0	13.1	7.7	31.4	20.7	21.7
CenterNet	41.700	22.6	20.6	14.6	59.7	24.0	21.3	20.1	**17.4**	37.9	23.7	26.2
YOLOv3	61.546	**49.6**	39.7	**19.2**	78.4	40.9	**38.4**	26.6	13.6	55.4	45.8	40.7
YOLOv4s	9.135	42.4	32.9	8.4	74.2	33.2	24.8	17.3	9.9	36.2	31.5	31.8
YOLOv5s	9.153	42.9	32.8	12.8	79.4	44.8	36.4	29.5	14.2	55.3	44.4	39.3
YOLOv8s	11.129	42.9	32.9	14.5	79.6	44.9	34.6	27.3	15.2	55.3	44.4	39.2
MFFCI-YOLOv8	10.260	42.7	33.3	13.4	80.5	46.9	38.4	31.4	16.6	58.6	44.6	40.6
ASF-YOLO	7.600	48.4	39.0	13.9	82.1	47.2	33.3	28.5	16.4	57.0	48.5	41.4
LUD-YOLO	10.340	44.8	34.3	14.5	80.9	**48.4**	29.8	29.8	16.9	**62.2**	46.2	41.7
YOLOv10s	7.222	43.4	34.7	13.8	80.1	45.3	36.0	28.3	16.4	55.8	45.7	40.0
MSUD-YOLO	**6.766**	48.0	**39.8**	17.5	**84.0**	47.0	37.4	**30.9**	15.8	60.0	**52.7**	**43.4**

As shown in Table 6, in comparison to Faster R-CNN and CenterNet, MSUD-YOLO's parameters are only 1/6 of those of these two models, but the overall mAP50 is increased by 100% and 65.6%, respectively. So, it illustrates that our proposed model obtains optimal outcomes regarding mAP50 and parameters. Meanwhile, in contrast to typical YOLO models such as YOLOv3, YOLOv4s, YOLOv5s, YOLOv8s and YOLOv10s, the parameters of the proposed MSUD-YOLO model in this article are smaller than the aforementioned models, a mere 6.766 MB, yet the overall mAP50 reaches 43.6%, outperforming the comparison models YOLOv3 (+2.73), YOLOv4s (+11.6), YOLOv5s (+4.1), YOLOv8s (+4.2) and YOLOv10s (+3.43), respectively. Compared with YOLOv10s, MSUD-YOLO showed a slight decrease in mAP on the awning–tricycle category, which has the fewest labels, but the mAP50 of the other nine categories improved, suggesting that the model has better robustness when detecting objects of different scales. Likewise, in relation to other state-of-the-art models, such as ASF-YOLO, MFFCI-YOLOv8 and LUD-YOLO, the proposed MSUD-YOLO model in this paper demonstrates better performance in terms of parameters and mAP50, outperforming ASF-YOLO (−0.834, +2.0), MFFCI-YOLOv8 (−3.494, +2.8) and LUD-YOLO (−3.574, +1.7). In particular, although we adopted the key modules of ASF-YOLO, our model has smaller

parameters and higher accuracy compared to ASF-YOLO, and the detection accuracy of most categories is superior to that of ASF-YOLO.

To clearly display the assessment outcomes of the aforementioned 11 algorithms for various categories, this study employs the normalized histogram of indicators to illustrate the comparative experiment, as depicted in Figure 12 (the evaluation criteria of indicators in the figure are the same as those in Figure 11). We can obviously see that the mAP50 and parameters of the MSUD-YOLO model are far superior to the other 10 comparison models, which further indicates that the proposed strategy can improve precision while reducing model complexity.

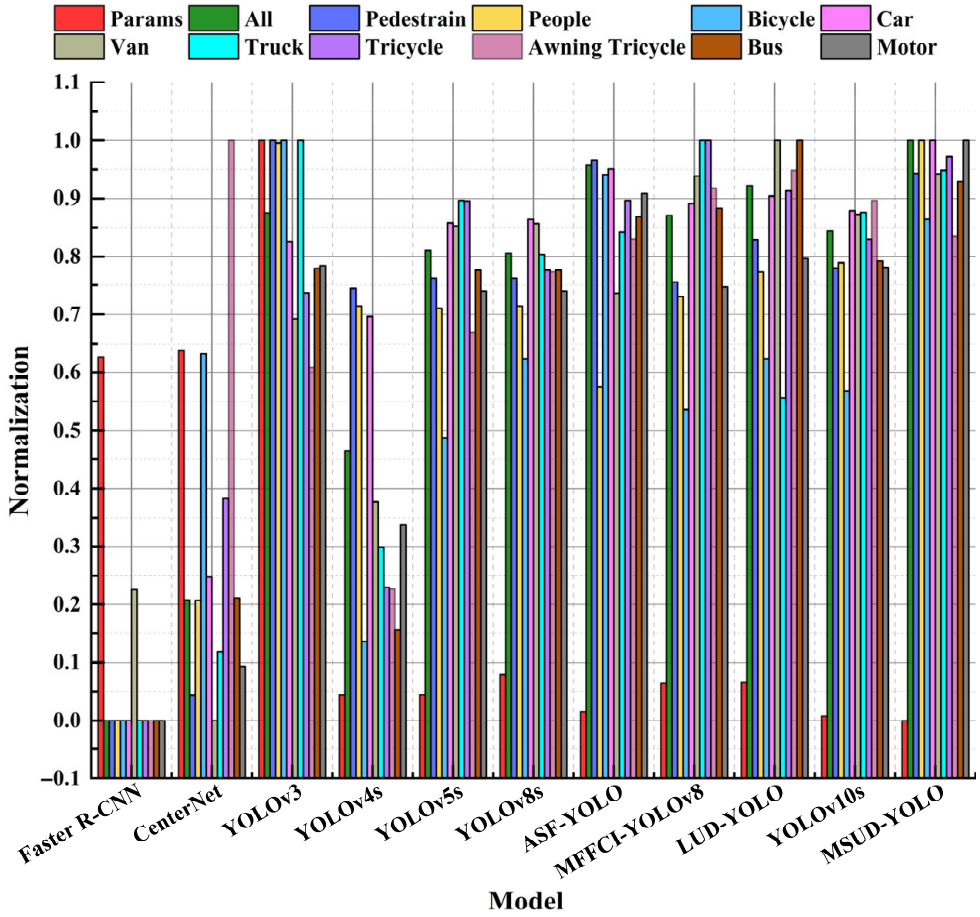

Figure 12. Normalization histogram of comparison experiment.

In summary, the MSUD-YOLO algorithm proposed in this article has high detection accuracy, strong adaptability and robustness when dealing with UAV aerial image object detection tasks. Meanwhile, deployment in drone embedded devices is also easier, which has high potential and advantages in practical applications.

4.6. Visual Experiment

To further evaluate the efficacy of MSUD-YOLO, we conducted visual experiments on the testing set, with the results displayed in Figure 13. In the experiments, various types of detection scenes including complex backgrounds (e.g., occlusion, nighttime, dark light), dense overlapping, small objects, different shooting heights and angles were selected as detection samples to compare the detection performance of YOLOv10s and MSUD-YOLO. The red dashed box in the figure indicates the comparison area for this experiment. It can be seen that the proposed model in our paper can well complete the detection task of various objects, and it can correctly recognize and locate objects in various scenarios.

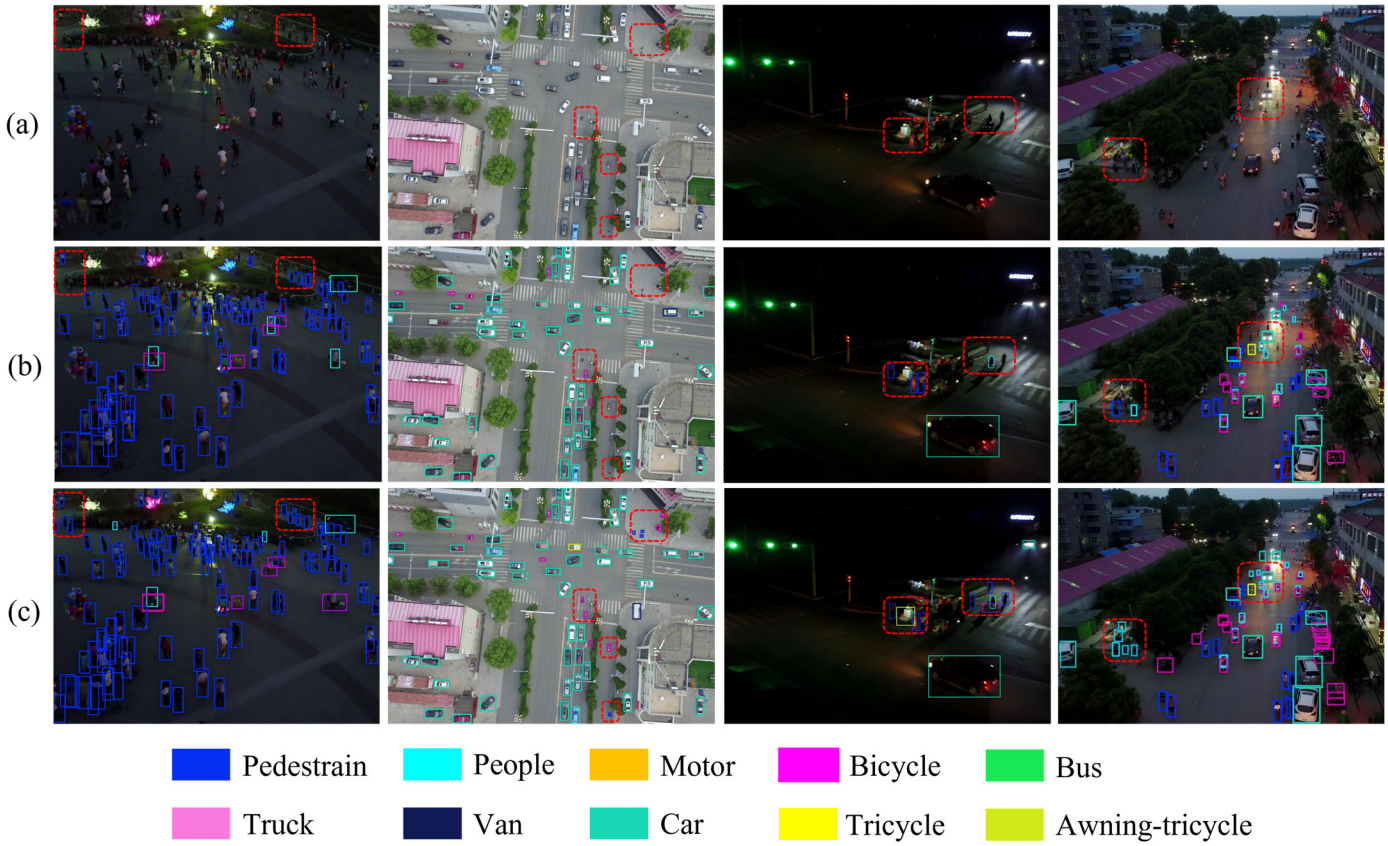

Figure 13. Comparison results of visualization experiments on VisDrone2019 dataset. (**a**) Initial image. (**b**) YOLOv10s. (**c**) MSUD-YOLO.

As illustrated in Figure 13, the first column is the scenario with densely overlapping objects. From Figure 13b,c, it is obvious that when objects present as dense and overlapping, YOLOv10s only recognizes a small number of objects, or even no objects. However, the improved model accurately identified pedestrians with dense overlaps. Even in corners with poor light conditions, MSUD-YOLO has excellent detection results. The second column is the intersection scene, in which the object to be detected presents multiscale changes. As can be seen from the red dotted line box in the figure, YOLOv10s missed the detection of many smaller objects, while our model accurately identified small-scale pedestrians and motors, thus verifying the effectiveness of our model in solving multiscale problems. The third and fourth columns are complex scenes such as night and occlusion, respectively. As shown in Figure 13a,b of the third column, YOLOv10s failed to recognize objects in complex backgrounds, misidentified pedestrians as crowds, and missed tricycles in poor lighting places. The improved model not only correctly identifies the object category but also has higher accuracy. In addition, our model also has a good detection effect in the occlusion scene, which proves its adaptability in different environmental conditions.

5. Conclusions

At present, the challenges of multiple scales, small objects, complex backgrounds and dense overlapping often arise when using UAVs for object detection. To address these problems, this paper proposes MSUD-YOLO for precise and efficient object detection in UAV aerial images. MSUD-YOLO replaces the PANet structure of YOLOv10s with the ASF, which merges spatial and scale features to achieve precise detection of multiscale objects in complex backgrounds. Meanwhile, the feature structure and prediction head specially designed for small objects are combined to solve the detection problem of small

objects in aerial images. Furthermore, by integrating the CFormerCGLU and GSConv modules, the improved model not only improves its feature extraction capability but also lowers computational cost. Lastly, the WIoU v3 loss function is used, incorporating a gradient allocation method to make our model more focused on low-quality examples, which improves the model's overall performance. The experimental results illustrate that compared with YOLOv10s, MSUD-YOLO has achieved better detection performance on the VisDrone2019 dataset. Specifically, mAP50 is increased by 8.5%, mAP50-95 is increased by 7.4%, the speed is 8.0% faster, the parameters are decreased by 6.3% and the model size is reduced by 4.4%. Furthermore, in comparison with multiple latest UAV object detection algorithms, our MSUD-YOLO model demonstrates the best comprehensive performance, providing a more efficient solution for the high-precision detection of UAV aerial image object under complex background.

This paper proposes a multiscale small object detection model suitable for UAVs, but it is only applicable to scenarios with a large amount of labeled data. In the future, we will explore methods of few-shot learning or unsupervised learning and study how to improve the model's detection accuracy with a small amount of labeled data or even no labeled data, thereby reducing the model's dependence on a large amount of labeled data. In addition, we will continue to be committed to enhancing the lightweight and processing speed of the model, so that when the model is embedded in unmanned aerial vehicles to perform related tasks, it can achieve faster, more accurate and more scenario object detection.

Author Contributions: Conceptualization, X.Z.; Validation, H.Z.; Formal analysis, H.Z.; Data curation, H.Z.; Writing—review and editing, W.Z.; software, J.M.; Visualization, C.L.; Supervision, Y.D.; Resources, Z.Z.; Project administration, X.Z.; Funding acquisition, X.Z. and Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 41404022 and in part by the National Foundation for Enhancing Fundamental Sciences in China under Grant 2021-JCJQ-JJ-0871.

Data Availability Statement: Data related to the current study are available from the corresponding author upon reasonable request. The codes used during the study are available from the corresponding author upon request.

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

References

1. Wang, Q.; Zhan, Y.; Zou, Y. UAV recognition algorithm for ground military targets based on improved Yolov5n. *Comput. Meas. Control* **2024**, *32*, 189–197.
2. Rao, J.; Xiang, C.; Xi, J.; Chen, J.; Lei, J.; Giernacki, W.; Liu, M. Path planning for dual UAVs cooperative suspension transport based on artificial potential field-A* algorithm. *Knowl.-Based Syst.* **2023**, *277*, 110797. [CrossRef]
3. Bhadra, S.; Sagan, V.; Sarkar, S.; Braud, M.; Mockler, T.C.; Eveland, A.L. PROSAIL-Net: A transfer learning-based dual stream neural network to estimate leaf chlorophyll and leaf angle of crops from UAV hyperspectral images. *ISPRS J. Photogramm. Remote Sens.* **2024**, *210*, 1–24. [CrossRef]
4. Duo, C.; Li, Y.; Gong, W.; Li, B.; Qi, G.; Zhang, J. UAV-aided distribution line inspection using double-layer offloading mechanism. *IET Gener. Transm Distrib.* **2024**, *18*, 2353–2372. [CrossRef]
5. Hang, R.; Xu, S.; Yuan, P.; Liu, Q. AANet: An ambiguity-aware network for remote-sensing image change detection. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5612911. [CrossRef]
6. Wan, M.; Gu, G.; Qian, W.; Ren, K.; Maldague, X.; Chen, Q. Unmanned aerial vehicle video-based target tracking algorithm using sparse representation. *IEEE Internet Things J.* **2019**, *6*, 9689–9706. [CrossRef]
7. Cheng, G.; Han, J. A survey on object detection in optical remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28. [CrossRef]
8. Weber, J.; Lefevre, S. A multivariate hit-or-miss transform for conjoint spatial and spectral template matching. In Proceedings of the 3rd International Conference, Image and Signal Processing, Cherbourg-Octeville, France, 1–3 July 2008; pp. 226–235.

9. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of online learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
10. 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.
11. Song, X.; Fang, X.; Meng, X.; Fang, X.; Lv, M.; Zhuo, Y. Real-time semantic segmentation network with an enhanced backbone based on Atrous spatial pyramid pooling module. *Eng. Appl. Artif. Intel.* **2024**, *133*, 107988. [CrossRef]
12. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
13. 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]
14. Terven, J.; Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 and beyond. *Compu. Vis. Pattern Recognit.* **2023**, arXiv:2304.00501.
15. 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.
16. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
17. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767, 1–6.
18. Bochkovskiy, A.; Wang, C.; Liao, H. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
19. Wang, C.; Liao, H.; Wu, Y.; Chen, P.; Hsieh, J.; Yeh, I. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
20. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the 2018, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
21. Jocher, G.; Liu, C.; Hogan, A.; Yu, L.; Rai, P.; Sullivan, T. *Ultralytics/YOLOv5: Initial Release*; Zenodo: Geneva, Switzerland, 2020.
22. 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.
23. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
24. Wang, C.; Bochkovskiy, A.; Liao, H. 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, 17–24 June 2023; pp. 7464–7475.
25. Gallagher, J. *How to Train an Ultralytics YOLOv8 Oriented Bounding Box (OBB) Model*; Roboflow: Des Moines, IA, USA, 2024.
26. Wang, C.; Yeh, I.; Liao, H. YOLOV9: Learning What You Want to Learn Using Programmable Gradient Information. In *European Conference on Computer Vision*; Springer Nature: Cham, Switzerland, 2024; pp. 1–21.
27. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J. YOLOV10: Real-Time End-to-End Object Detection. *Comput. Vis. Pattern Recognit.* **2025**, *37*, 107984–108011.
28. Jawaharlalnehru, A.; Sambandham, T.; Sekar, V.; Ravikumar, D.; Loganathan, V.; Kannadasan, R.; Khan, A.; Wechtaisong, C.; Haq, M.; Alhussen, A.; et al. Target object detection from unmanned aerial vehicle (UAV) images based on improved YOLO algorithm. *Electronics.* **2022**, *11*, 2343. [CrossRef]
29. Sahin, O.; Ozer, S. YOLODrone: Improved YOLO architecture for object detection in drone images. In Proceedings of the 2021 44th International Conference on Telecommunications and Signal Processing (TSP), Brno, Czech Republic, 26–28 July 2021; pp. 361–365.
30. Koay, H.; Chuah, J.; Chow, C.; Chang, Y.; Yong, K. YOLO-RTUAV: Towards real-time vehicle detection through aerial images with low-cost edge devices. *Remote Sens.* **2021**, *13*, 4196. [CrossRef]
31. Cao, J.; Bao, W.; Shang, H.; Yuan, M.; Cheng, Q. GCL-YOLO: A ghostconv-based lightweight YOLO network for UAV small object detection. *Remote Sens.* **2023**, *15*, 4932. [CrossRef]
32. Wang, J.; Zhang, F.; Zhang, Y.; Liu, Y.; Cheng, T. Lightweight object detection algorithm for UAV aerial imagery. *Sensors* **2023**, *23*, 5786. [CrossRef]
33. Hui, Y.; Wang, J.; Li, B. DSAA-YOLO: UAV remote sensing small target recognition algorithm for YOLOv7 based on dense residual super-resolution and anchor frame adaptive regression strategy. *J. King Saud Univ. Comput. Inf. Sci.* **2024**, *36*, 101863. [CrossRef]
34. 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]

35. Fan, Q.; Li, Y.; Deveci, M.; Zhong, K.; Kadry, S. LUD-YOLO: A novel lightweight object detection network for unmanned aerial vehicle. *Inf. Sci.* **2025**, *686*, 121366. [CrossRef]
36. Qi, S.; Song, X.; Shang, T.; Hu, X.; Han, K. MSFE-YOLO: An Improved YOLOv8 Network for Object Detection on Drone View. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 1–5. [CrossRef]
37. Zhang, Y.; Ye, M.; Zhu, G.; Liu, Y.; Guo, P.; Yan, J. FFCA-YOLO for Small Object Detection in Remote Sensing Images. *IEEE Tran. Geosci. Remote Sens.* **2024**, *62*, 1–15. [CrossRef]
38. Luo, X.; Zhu, X. YOLO-SMUG: An Efficient and Lightweight Infrared Object Detection Model for Unmanned Aerial Vehicles. *Drones* **2025**, *9*, 245. [CrossRef]
39. Liu, L.; Huang, K.; Li, Y.; Zhang, C.; Zhang, S.; Hu, Z. Real-time pedestrian recognition model on edge device using infrared vision system. *J. Real-Time Image Process.* **2025**, *22*, 1–11. [CrossRef]
40. Kang, M.; Ting, C.; Ting, F.; Raphael, C. ASF-YOLO: A novel yolo model with attentional scale sequence fusion for cell instance segmentation. *Image Vis. Comput.* **2024**, *147*, 105057. [CrossRef]
41. Lindeberg, T. *Scale-Space Theory in Computer Vision*; Springer: Cham, Switzerland, 1994; pp. 10–11.
42. Lowe, D.J. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
43. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
44. Lin, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Slim-neck by GSConv: A lightweight-design for real-time detector architectures. *Comput. Vis. Pattern Recognit.* **2024**, *21*, 62.
45. Peng, Z.; Huang, W.; Gu, S.; Xie, L.; Wang, Y.; Jiao, J.; Ye, Q. Conformer: Local Features Coupling Global Representations for Visual Recognition. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 367–376.
46. Shi, D. TransNeXt: Robust Foveal Visual Perception for Vision Transformers. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024. arXiv:2311.17132.
47. Zhang, Y.; 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]
48. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740.
49. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
50. 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; pp. 213–226.
51. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6569–6578.
52. Xu, S.; Song, L.; Yin, J.; Chen, Q.; Zhan, T.; Huang, W. MFFCI–YOLOv8: A Lightweight Remote Sensing Object Detection Network Based on Multiscale Features Fusion and Context Information. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 19743–19755. [CrossRef]