DOI:10.11918/202412071

基线网络轻量化的无人机小目标检测算法

杨 超[1],李新凯[2],张宏立[2],孟 月[2],王 聪[2]

(1.新疆大学 电气工程学院,乌鲁木齐 830049;2.新疆大学 智能科学与技术学院,乌鲁木齐 830049)

摘 要:为解决无人机端侧小目标检测中像素低、背景复杂、目标密集且算力受限的矛盾,提出基于改进YOLOv8s的轻量化基线(you only look once v8 small-lightweight,YOLOv8s-LW)算法。首先,利用幽灵卷积与深度卷积优化的高性能图形处理器网络取代YOLOv8s主干网络,通过参数共享与结构裁剪降低冗余,增强端到端可训练性;其次,在颈部以幽灵卷积增强型特征融合模块和快速块高效多尺度注意力融合模块替换跨阶段双卷积瓶颈模块,引入空间混合与多尺度注意力加权,强化多尺度特征聚合与泛化能力;再次,以焦点形状约束交并比损失函数取代原损失函数,从形状约束与匹配精度多维度优化小目标回归鲁棒性;最后,融合动态检测头,提升复杂背景下的小目标响应与定位一致性。实验结果表明:在VisDrone-2019数据集上,YOLOv8s-LW算法的mAP@0.5达到38.5%,参数量为6.4×10^6,计算量为19.2×10^9,推理速度为64.0帧/s;相较于基线YOLOv8s,参数量下降约43%,计算量下降约33%,且精度提升0.2%。所提算法在端侧无人机小目标检测中实现了低算力、高实时、稳精度的可用均衡,形成可直接部署的轻量化基线,并为复杂背景场景下的稳健检测与后续多任务扩展提供工程化支持与可迁移的结构设计范式。

关键词:无人机;小目标检测;YOLOv8s;轻量化;注意力融合模块;损失函数;检测头

中图分类号:TP391 **文献标志码**:A **文章编号**:0367-6234(2026)09-0070-13

Small target detection algorithm for unmanned aerial vehicles based on baseline network lightweighting

Yang Chao[1], Li Xinkai[2], Zhang Hongli[2], Meng Yue[2], Wang Cong[2]

(1. School of Electrical Engineering, Xinjiang University, Urumqi 830049, China;
2. School of Intelligent Science and Technology, Xinjiang University, Urumqi 830049, China)

Abstract: To alleviate the conflicts of low pixel resolution, complex backgrounds, dense targets, and limited compute in unmanned aerial vehicle (UAV) edge detection for small targets, this paper proposed a lightweight baseline algorithm based on the improved YOLOv8s, namely you only look once v8 small lightweight (YOLOv8s-LW). First, the algorithm replaced the backbone network of YOLOv8s with a high-performance graphics processing unit network optimized with ghost and depthwise convolutions, reducing redundancy through parameter sharing and structural pruning while enhancing end-to-end trainability. Second, it replaced the module of cross-stage partial bottleneck with two convolutions with the ghost-convolution-augmented feature fusion module and the fast block-based efficient multi-scale attention fusion module, which introduced spatial mixing and multi-scale attention weighting to strengthen multi-scale feature aggregation and generalization. Subsequently, the algorithm employed a focaler-shape-wise intersection over union loss function instead of the original loss function to improve the robustness of small target regression from the perspectives of shape constraint and matching accuracy. Finally, it integrated a dynamic detection head to enhance small target responses and localization consistency under complex backgrounds. Experiments on the VisDrone-2019 dataset show that YOLOv8s-LW achieves 38.5% mAP@0.5 with 6.4×10^6 parameters, 19.2×10^9 computations, and an inference speed of 64.0 frames per second. Compared with the baseline YOLOv8s, parameters and computation are reduced by about 43% and 33%, respectively, while accuracy is improved by 0.2%. The proposed algorithm achieves a favorable trade-off among computational cost, real-time performance, and accuracy for UAV edge detection for small targets and provides engineering support and a deployable lightweight baseline and transferable architectural design for robust detection and multi-task extensions in complex background scenarios.

Keywords: unmanned aerial vehicle; small target detection; YOLOv8s; lightweighting; attention fusion module; loss function; detection head

收稿日期:2024-12-24;**录用日期**:2025-03-04;**网络首发日期**:2025-11-24
网络首发地址:https://link.cnki.net/urlid/23.1235.t.20251121.1712.010
基金项目:国家自然科学基金(62263030);新疆维吾尔自治区自然科学基金(2023D01C187)
作者简介:杨 超(1999—),男,硕士研究生;张宏立(1972—),男,教授,博士生导师
通信作者:李新凯,lxk@xju.edu.cn

无人机凭借其卓越的灵活性和高效性，在大范围区域内实现实时监控，广泛应用于交通监控和智能交通系统等多个场景。无人机目标检测技术[1]结合了计算机视觉与机器学习等先进技术，相较于地面监控系统，能够从空中获得全局视野，提供更全面的交通流量分析和更精准的目标检测。这一特性有效降低了人力成本。然而，来自无人机的图像通常面临诸多挑战，其中小目标检测尤为突出。小目标通常具有较低的分辨率，并且位于图像的较小区域，传统的卷积神经网络往往难以充分提取这些目标的细节特征，特别是在采用轻量化网络结构时，网络深度和参数量的减少易导致信息丢失。因此，在追求高精度与高召回率的同时，如何平衡推理速度、算法大小和鲁棒性，已成为无人机目标检测技术优化的重要方向。

为解决小目标检测困难和算力受限问题，Cao 等[2]对 YOLOv8[3]的基础算法使用组洗牌卷积和双向特征金字塔网络进行改进，在提升多尺度特征融合效果的同时平衡了精度与速度，但在实际部署后，算法的算力需求仍然较高。Cao 等[4]在 YOLOv5[5]网络框架的基础上引入了多尺度特征融合模块，以提升对小目标的检测精度。但这种方法需要处理和融合来自不同尺度的特征图，从而导致算法的计算复杂度和推理延迟显著增加。凌强等[6]提出了 DN-YOLOv5 算法，该算法通过改进 YOLOv5，提升了检测精度并降低了算法复杂度，实现了高效的缺陷检测效果。杨志渊等[7]在 YOLOv8s 的基础上提出了轻量级目标检测算法，该算法通过引入非对称检测头，提升了小目标检测的精度和速度。

尽管现有研究在提升检测精度与优化计算复杂度方面取得了显著进展，但在小目标检测任务中依然存在检测精度与硬件算力之间难以平衡的挑战，往往需要在性能与实时性之间作出折中。为此，本文提出基于改进 YOLOv8s 的轻量化基线（you only look once v8 small-lightweight，YOLOv8s-LW）算法。通过结合幽灵卷积（ghost convolution，Ghost Conv）[8]和深度可分离卷积（depthwise separable convolution，DS Conv）[9]优化的网络高性能图形处理器网络（high-performance GPU net v2，HGNetV2）[10]来替换 YOLOv8s 的基线主干结构，以期降低计算参数量的同时提升特征表示能力；在跨阶段双卷积瓶颈（cross-stage partial bottleneck with two convolutions，C2f）部分，分别结合 Ghost Conv 和快速块高效多尺度注意力模块（fasterblock and efficient multi-scale attention，FasterBlock_EMA）对特征进行加权处理，在优化计算方式及提升计算效率的同时，增强模型的鲁棒性与检测稳定性；使用融合聚焦交并比（focaler intersection over union，Focaler-IoU）损失函数[11]、形状感知交并比（shape intersection over union，Shape-IoU）损失函数[12]和智能交并比（wise intersection over union，Wise-IoU）损失函数[13]的设计思想，构建焦点形状约束交并比（Focaler-Shape-Wise-IoU，FSW-IoU）损失函数替换原来的损失函数，对困难样本、目标形状和多样性进行优化；最后，利用动态检测头（dynamic head，DyHead）[14]提升复杂背景下的小目标检测与定位一致性。

1　基准算法选择与分析

基于深度学习的目标检测算法可大致划分为一阶段和两阶段两种架构。一阶段算法（如 YOLO 系列、SSD[15]、RetinaNet[16]等）通过单一神经网络进行端到端的类别识别和位置回归，这类算法结构简单且计算效率高，适用于对实时性要求较高的场景。两阶段算法如区域卷积神经网络系列，首先生成一组高质量的候选区域，再通过精细分类和精确定位来优化检测性能，虽然精度较高，但计算复杂度和推理时间较长，因此，更多地应用于对精度要求极高的任务。

作为典型的一阶段检测框架，YOLOv8 在精度与推理效率之间实现了合理的平衡，凭借较低的参数量和计算复杂度，成为实时目标检测任务中的理想选择。YOLOv8 的网络架构主要由主干（backbone）、颈部（neck）和头部（head）3 部分组成，且根据不同的需求，提供了多种子算法可供选择。通过调整网络宽度和深度，YOLOv8 衍生出了多个变体，分别为极小型（nano）、小型（small）、中型（medium）、大型（large）、超大型（extra-large），在算法规模和性能上呈现逐步递增的趋势。为了在精度和硬件性能之间取得较好的平衡，本文选取 YOLOv8s 作为基线网络。

2　YOLOv8s-LW 算法

图 1 为 YOLOv8s-LW 算法的结构。起始特征抽取模块（HG Stem）的作用是将输入图像先做浅层卷积特征提取与下采样，将通道数提升至合适的规模，再将特征送入后面的各个阶段；幽灵层次特征块（Ghost_HG Block）由 Ghost Conv 模块与特征层次块（HG Block）结合构成，是核心特征提取单元；快速块高效多尺度注意力融合特征模块（C2f with fasterblock and efficient multi-scale attention，C2f_FB_EMA）用以加强小目标的显著性和上下文感知能力；通道拼接模块（Concat）用于沿通道维将多路特

征图拼接在一起;上采样模块(Upsample)将特征图放大至目标尺寸;快速空间金字塔池化(spatial pyramid pooling fast,SPPF)可增强特征的多尺度表达能力;幽灵卷积与动态卷积增强型特征融合模块(C2f with ghost and dynamic convolution,C2f_GDC)能以更低的计算量提升多尺度特征的表达与建模能力。

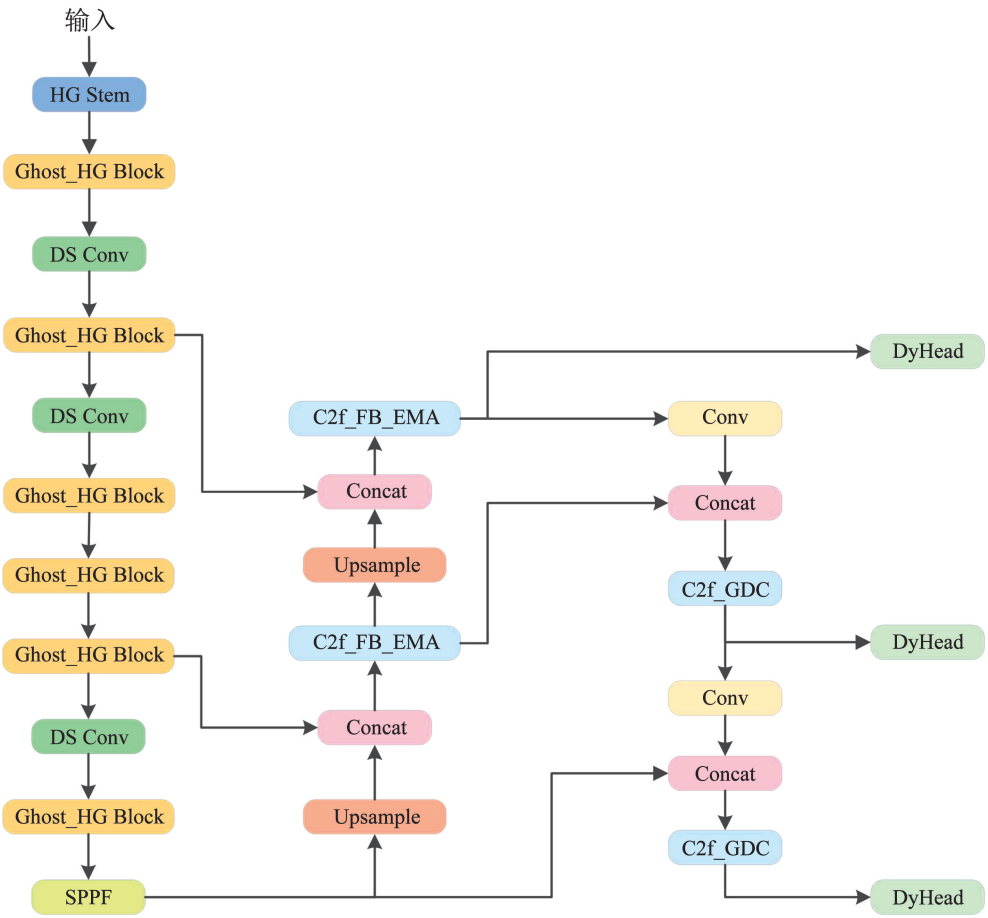


图 1 YOLOv8s-LW 结构

Fig. 1 YOLOv8s-LW architecture

2.1 轻量化主干网络设计

YOLOv8s 的主干网络通过卷积和池化操作从输入图像中提取特征,逐步降低空间维度并增加通道数以生成抽象且丰富的特征表示,但这些操作带来了高参数量和高计算量。为应对性能与效率的挑战,本文提出以幽灵卷积与深度卷积优化的高性能图形处理器网络(HGNetV2 with ghost and depthwise optimizations,GD_HGNetV2),旨在通过减少参数量的同时平衡检测准确性和效率。GD_HGNetV2 的整体结构如图 2 所示,其中的通道注意力模块(ESE)可在几乎不增加计算量的前提下提升特征表达能力。

HGNetV2 基于第一代高性能图形处理器网络架构,融合了可学习仿射变换层,通过引入缩放因子与偏移量增强算法的表达能力与适应性。网络由 4 个阶段组成,在阶段 2 ~ 4 中,采用可学习下采样层,通过组化操作降低计算复杂度来减轻硬件推理负担;为进一步提升轻量化特性,利用 DS Conv 替代传统池化层,优化网络的端到端训练;最后将 Ghost Conv 模块与 HG Block 结合构成 Ghost_HG Block,通过参数共享机制减少参数量。

DS Conv 操作分为深度卷积和逐点卷积两阶段。在深度卷积中输入特征图的维度为 $D_F \times D_F \times M$,其中 D_F 为输入特征图的空间尺寸,M 为通道数;$D_K \times D_K$ 为卷积核大小,卷积之后得到尺寸为 $A \times A \times M$ 的特征图,其中 A 为深度卷积输出特征图的边长。逐点卷积采用 1×1 的卷积核对深度卷积的输出再次进行操作,目的是组合来自不同通道的特征,如果有 N 个输出通道,则逐点卷积的参数量为 $M \times N$。综合两个阶段,DS Conv 模块的总参数量为 $M \times (D_K \times D_K + N)$,参数个数大约是常规卷积的三分之一,算法的参数量显著减少。

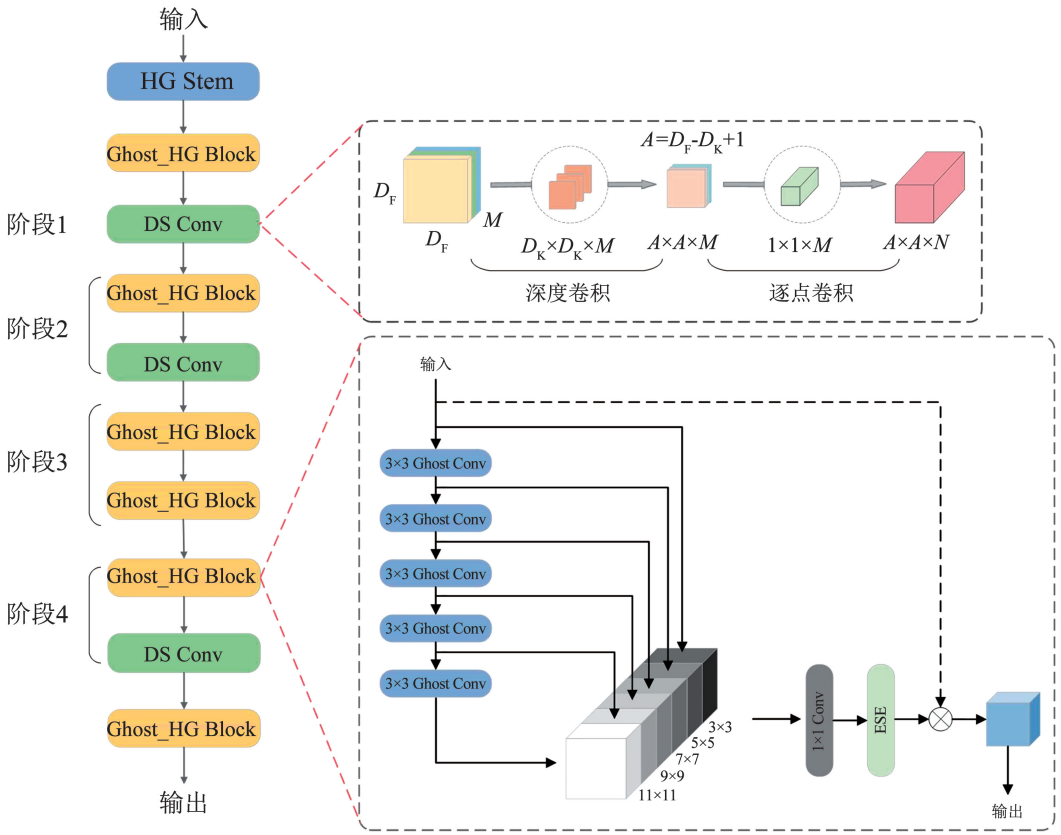


图 2　GD_HGNetV2 整体结构

Fig. 2　Overall architecture of GD_HGNetV2

Ghost Conv 模块通过减少卷积操作的次数来降低计算复杂度，在基本不牺牲检测性能的前提下加速推理并提高算力利用率。Ghost Conv 模块结构如图 3 所示。

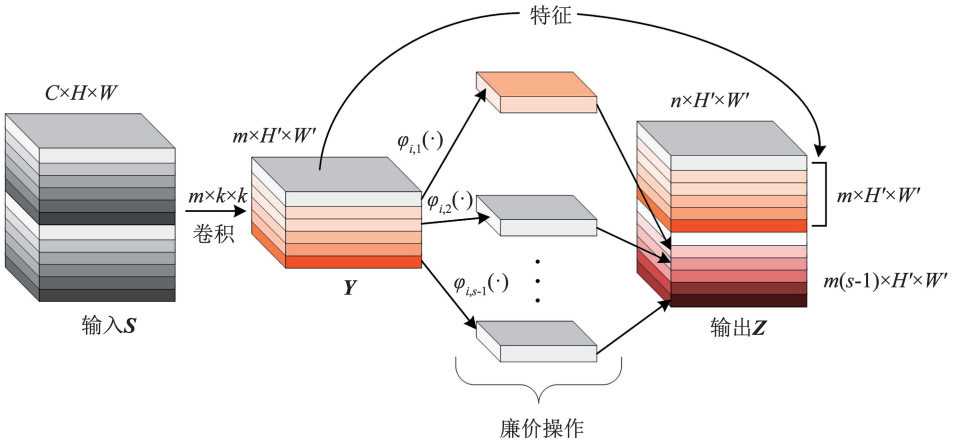


图 3　Ghost Conv 模块结构

Fig. 3　Ghost Conv module architecture

假设 Ghost Conv 模块的输入特征空间为 $\boldsymbol{S}\in\mathbf{R}^{C\times H\times W}$，其中 C、H、W 分别表示特征图的输入通道数、高度、宽度，Ghost Conv 操作整体分为两个部分：第一部分采用 m 个 $k\times k$ 卷积核 $\boldsymbol{f}$ 提取本征特征图 $\boldsymbol{Y}$（此处忽略偏置项），运算公式为

$$\boldsymbol{Y}=\boldsymbol{S}*\boldsymbol{f} \tag{1}$$

式中：$\boldsymbol{Y}\in\mathbf{R}^{m\times H'\times W'}$，$m$、$H'$、$W'$ 分别为特征图的输出通道数、高度、宽度；$*$ 为卷积算子。

第二部分对每个本征特征图 $\boldsymbol{Y}_i$ 施加 $s-1$ 个廉价线性变换 $\varphi_{i,j}(\cdot)$，生成幽灵特征图 $\boldsymbol{Y}'_{i,j}$，运算公式为

$$\boldsymbol{Y}'_{i,j}=\varphi_{i,j}(\boldsymbol{Y}_i),i=1,\cdots,m;\ j=1,\cdots,s-1 \tag{2}$$

从而共得到 m 个本征特征图和 $m(s-1)$ 个幽灵特征图，将二者在通道维拼接可得到输出 $\boldsymbol{Z}\in\mathbf{R}^{n\times H'\times W'}$，其中 $n=m\cdot s$，n 为输出通道数，s 为每个本征特征图产生的特征图个数。

普通卷积计算量 C_{std} 的计算式为

$$C_{\text{std}} = n \times h' \times w' \times c \times k^2 \tag{3}$$

式中：h'、w' 分别为输出特征图的高度、宽度，c 为卷积操作的输入通道数，k 为普通卷积核的空间尺寸。

Ghost Conv 计算量 C_{ghost} 的计算式为

$$C_{\text{ghost}} = \frac{n}{s} \times h' \times w' \times c \times k^2 + (s-1)\frac{n}{s} \times h' \times w' \times l^2 \tag{4}$$

式中 l 为廉价线性变换所使用卷积核的空间尺寸，通常有 $l \approx k$。

普通卷积与 Ghost Conv 计算量比值 R_{s} 为

$$R_{\text{s}} = \frac{C_{\text{std}}}{C_{\text{ghost}}} = \frac{n \times h' \times w' \times c \times k^2}{\frac{n}{s} \times h' \times w' \times c \times k^2 + (s-1)\frac{n}{s} \times h' \times w' \times l^2} \approx s \tag{5}$$

普通卷积参数量 P_{std} 的计算式为

$$P_{\text{std}} = n \times c \times k^2 \tag{6}$$

Ghost Conv 参数量 P_{ghost} 的计算式为

$$P_{\text{ghost}} = \frac{n}{s} \times c \times k^2 + (s-1)\frac{n}{s} \times l^2 \approx \frac{n}{s} \times k^2 \times (c+s-1) \tag{7}$$

普通卷积与 Ghost Conv 参数量比值 R_{c} 为

$$R_{\text{c}} = \frac{P_{\text{std}}}{P_{\text{ghost}}} = \frac{n \times c \times k^2}{\frac{n}{s} \times k^2 \times (c+s-1)} \approx s \tag{8}$$

由式(5)和式(8)可知，计算量比值 R_{s} 和参数量比值 R_{c} 均为 s，表明 Ghost Conv 能在保持输出通道数不变的前提下显著降低了计算量与参数量。

对基线网络和改进后网络输入相同大小图像进行训练，通过对比替换前后的网络架构，整体参数量从 11.2×10^6 减少至 8.4×10^6，计算量从 28.6×10^9 降至 23.5×10^9，充分体现了所设计网络的轻量化优势；但轻量化设计会伴随着精度的下降，所以接下来的工作将聚焦于提升检测精度，同时有效缓解轻量化带来的性能损失。

2.2 基于 C2f 的改进模块

C2f 模块在小目标检测任务中并不能充分融合多尺度特征，导致在检测时获取的特征信息不足，从而造成检测精度下降。因此本文在 C2f 模块中分别结合 Ghost Conv 和 FasterBlock_EMA，设计新的模块 C2f_GDC 和 C2f_FB_EMA。改进后的 C2f_GDC 结构如图 4 所示，其中特征切分模块(Split)用于沿通道维将输入特征划分为多路子特征，供后续分支处理。

C2f_GDC 通过使用 Ghost Conv 替代传统的瓶颈模块构成幽灵模块(Ghost Module)，以减少计算量并保持特征表达能力，从而提高计算效率，使其更适用于计算资源受限的场景。

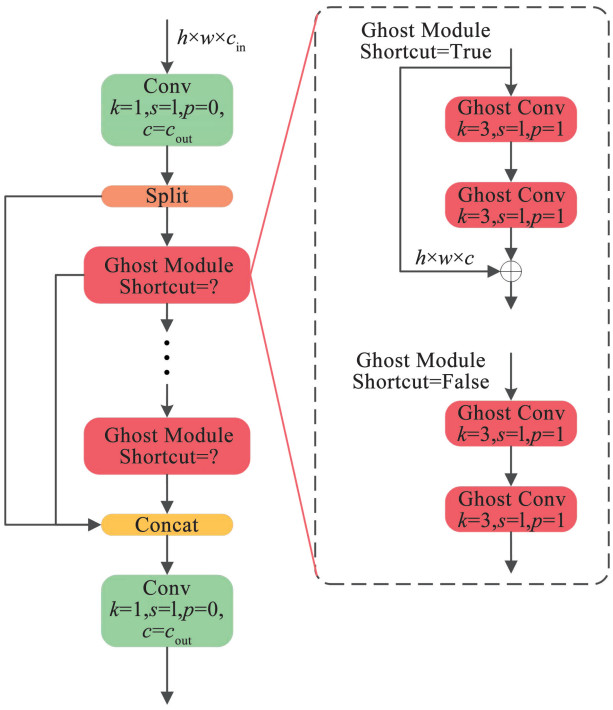


图 4 C2f_GDC 模块结构

Fig. 4 C2f_GDC module architecture

C2f_FB_EMA 模块是将 C2f 中瓶颈模块替换为 FasterBlock_EMA，而 FasterBlock_EMA 是由高效多尺度注意力(efficient multi-scale attention，EMA)[17]集成到快速块(FasterBlock)[18]的部分卷积(partial convolution，PConv)前构成的，这样的设计可以提升网络在处理复杂输入场景时的多尺度特征表达能力。C2f_FB_EMA 模块的结构如图 5 所示，其中分组注意力模块(Groups)用于沿通道维将输入特征划分为多组，各组独立计算注意力以降低整体计算开销；水平全局平均池化(Avg Pool(P))与垂直全局平均池化(Avg Pool(V))分别沿水平方向和垂直方向对特征做一维全局平均池化，提取横向与纵向的上下文信息；拼接与 1 × 1 卷积操作(Concat + 1 × 1 Conv)将两条池化分支在通道维拼接并通过 1 × 1 卷积完成通道融合，Sigmoid 激活函数将融合向量压缩至[0，1]以生成方向门控权重图，通道重标定(Re-weight)利用该权重对组内特征逐点重加权，实现方向敏感的通道增强，随后由组归一化(GroupNorm)对重标定后的特征进行组归一化以稳定小批量训练；通过行列相关性池化(Avg Pool)得到低维的行、列相关性表示，经注意力权重归一化(Softmax)转换为行/列注意力权重，并由长程依赖聚合(Matmul)对特征进行加权聚合以建模长程空间依赖，最终通过空间注意力掩码生成得到像素级空间注意力图，对输出特征图进行逐像素重加权。

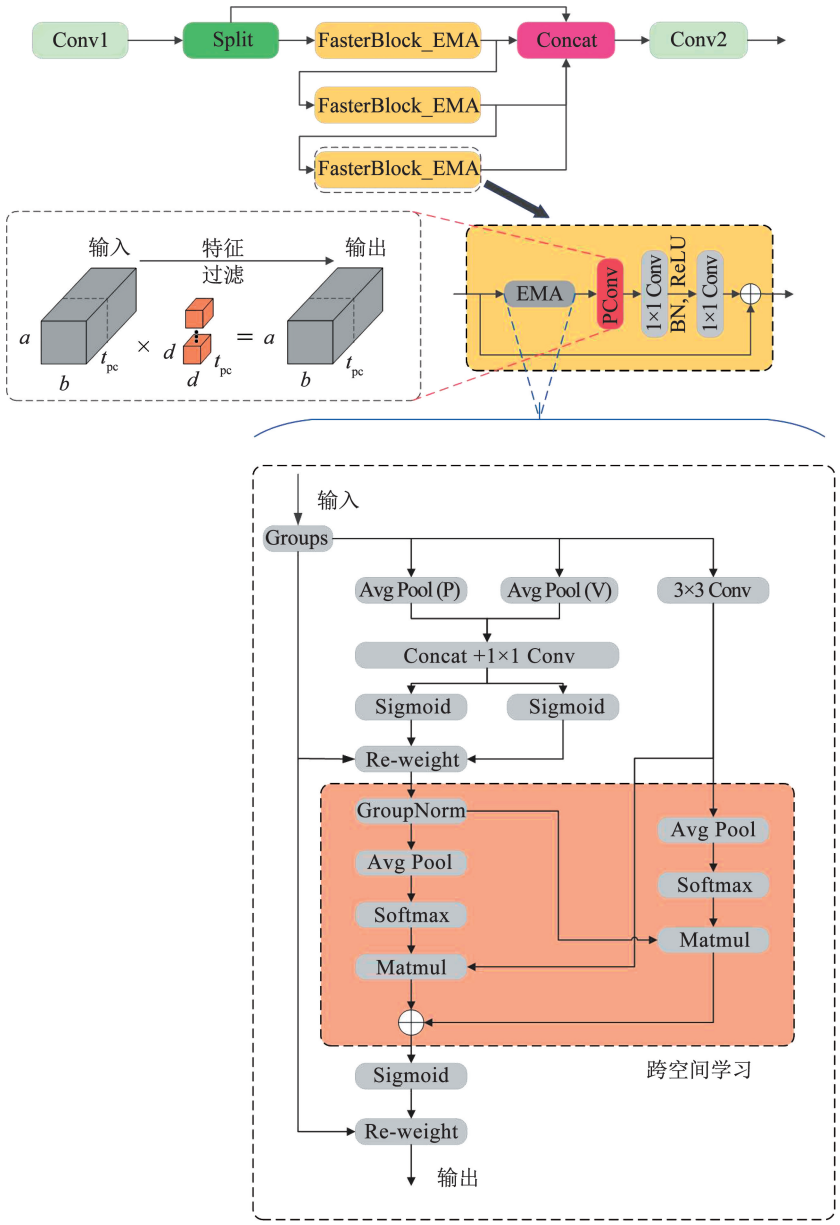


图 5　C2f_FB_EMA 模块结构

Fig. 5　C2f_FB_EMA module architecture

EMA 模块以输入特征 $\boldsymbol{F}\in\mathbf{R}^{E\times G\times Q}$ 为基础，其中 E、G、Q 分别为特征图的输入通道数、高度、宽度，先按通道分组，在每一组的内部分别沿水平与垂直方向进行一维全局平均池化，以提取方向敏感的上下文信息。对于第 e 个通道，在高度为 G、宽度为 Q 的特征图上，沿水平方向（宽度维）的一维全局平均池化可得到针对第 e 个通道的按行聚合后的纵向响应序列 $z_e^g(g)$，其表达式为

$$z_e^g(g)=\frac{1}{Q}\sum_{p=0}^{Q-1}r_e(g,p),\ g=0,\cdots,G-1 \tag{9}$$

式中 $r_e(g,p)$ 表示在第 e 个通道、行索引为 g、列索引为 p 处的输入特征值。

类似地，沿垂直方向（高度维）的一维全局平均池化可得到针对第 e 个通道的按列聚合后的纵向响应序列 $z_e^q(q)$，其表达式为

$$z_e^q(q)=\frac{1}{G}\sum_{u=0}^{G-1}r_e(u,q),\ q=0,\cdots,Q-1 \tag{10}$$

式中 $r_e(u,q)$ 表示在第 e 个通道、行索引为 u、列索引为 q 处的输入特征值。

在 C2f_FB_EMA 中，PConv 仅对部分通道执行逐点卷积，从而降低计算量与内存访问开销。假设 PConv 中特征图的空间尺寸为 $a\times b$，卷积核尺寸为 $d\times d$，参与运算的通道数为 t_{pc}（通常有 $t_{\mathrm{pc}}=t/4$），则 PConv 每秒执行的浮点运算次数 T_{opc} 与内存访问量 T_{mem} 可分别表示为：

$$T_{\mathrm{opc}}=a\cdot b\cdot d^2\cdot t_{\mathrm{pc}}^2 \tag{11}$$

$$T_{\mathrm{mem}}=a\cdot b\cdot 2t_{\mathrm{pc}}+d^2\cdot t_{\mathrm{pc}}^2\approx a\cdot b\cdot 2t_{\mathrm{pc}} \tag{12}$$

相比在全部 t 个通道上执行的标准逐点卷积，

当 $t_{pc}=t/4$ 时，PConv 的浮点运算次数约为常规卷积的 1/16，内存访问量约为 1/4。因此，将 EMA 与 FasterBlock 结合，可以在增强多尺度与长程依赖建模能力的同时有效抑制参数量和计算复杂度的增加，在精度与效率之间取得较为理想的平衡。

C2f_GDC 在计算效率上具有优势，但其在某些场景下可能会损失部分重要的特征信息，特别是在需要精细特征的复杂场景下；而 C2f_FB_EMA 在精度上有很大提升，但相应的计算开销也会有所增加。两种模块对检测算法指标影响的实验结果见表 1。由表 1 可知，若同时引入 C2f_GDC 和 C2f_FB_EMA 两种模块，能够在不同维度上实现较为理想的平衡，优化了参数量、计算量与检测精度之间的相互关系，这一组合方案显著提升了检测精度，同时有效控制了算法的计算开销和参数增长，避免过度复杂化。

表 1 两种模块对检测算法指标的影响

Tab. 1 Influence of two modules on indicators of detection algorithm

模块	mAP@0.5/%	参数量/10^6	计算量/10^9
C2f_GDC	36.3	6.85	19.7
C2f_FB_EMA	37.1	7.12	20.9
C2f_GDC + C2f_FB_EMA	36.8	6.93	20.2

2.3 多维融合损失函数

YOLOv8s 基线网络中的损失函数在小目标的尺寸、形状变化和背景干扰较大时，执行的效果较差，其主要关注边界框的重叠度、位置和长宽比，对于形状不规则或尺寸较小的目标优化效果有限，且缺乏智能加权机制。为在小目标检测中同时兼顾样本难度、框形状以及锚框质量，本文在基础交并比（intersection over union，IoU）损失基础上引入 Focaler-IoU[11]、Shape-IoU[12] 与 Wise-IoU[13] 思想，并构造综合的 FSW-IoU 损失，用于替代原始边界框回归损失。

2.3.1 基础 IoU 损失

设预测框为 B，真实框为 B^{GT}，基础 IoU 定义和对应的基础 IoU 损失函数分别为：

$$\mathrm{IoU}=\frac{|B\cap B^{GT}|}{|B\cup B^{GT}|} \tag{13}$$

$$L_{IoU}=1-\mathrm{IoU} \tag{14}$$

2.3.2 Focaler-IoU

Focaler-IoU 的定义为

$$\mathrm{IoU}_{Focaler}=\begin{cases}0, & \mathrm{IoU}<L_{down}\\ \dfrac{\mathrm{IoU}-L_{down}}{L_{up}-L_{down}}, & L_{down}\ll\mathrm{IoU}\ll L_{up}\\ 1, & \mathrm{IoU}>L_{up}\end{cases} \tag{15}$$

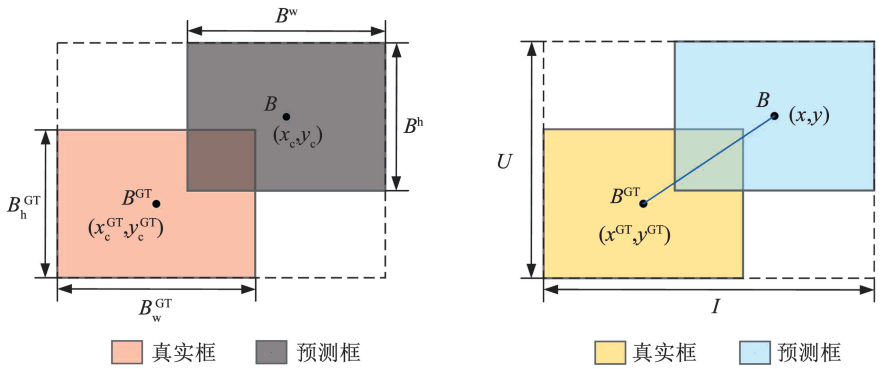
式中：$\mathrm{IoU}\in[0,1]$，L_{down}、L_{up} 分别为映射区间的下界、上界。

当区间靠近低 IoU 区域时，算法更关注困难样本；当区间靠近高 IoU 区域时，算法更关注高质量样本。其对应的 Focaler-IoU 损失函数为

$$L_{Focaler\text{-}IoU}=1-\mathrm{IoU}_{Focaler} \tag{16}$$

2.3.3 Shape-IoU

Shape-IoU 和 Wise-IoU 示意如图 6 所示。



(a) Shape-IoU示意 (b) Wise-IoU示意

图 6 Shape-IoU 和 Wise-IoU 示意

Fig. 6 Shape-IoU and Wise-IoU illustration

Shape-IoU 定义为：

$$\eta_N=\frac{2\times(B_w^{GT})^{\vartheta}}{(B_w^{GT})^{\vartheta}+(B_h^{GT})^{\vartheta}} \tag{17}$$

$$\eta_Z=\frac{2\times(B_h^{GT})^{\vartheta}}{(B_w^{GT})^{\vartheta}+(B_h^{GT})^{\vartheta}} \tag{18}$$

$$\phi_{shape}=\eta_Z\times\frac{(x_c-x_c^{GT})^2}{v^2}+\eta_N\times\frac{(y_c-y_c^{GT})^2}{v^2} \tag{19}$$

$$\Gamma_{shape}=\sum_{o=B^w,B^h}(1-\mathrm{e}^{-\zeta_o})^{\varphi},\begin{cases}\zeta_{B^w}=\eta_Z\times\dfrac{|B^w-B_w^{GT}|}{\max(B^w,B_w^{GT})}\\ \zeta_{B^h}=\eta_N\times\dfrac{|B^h-B_h^{GT}|}{\max(B^h,B_h^{GT})}\end{cases} \tag{20}$$

式中：η_N、η_Z 分别为水平方向、垂直方向的权重系

数，其数值与真实框的形状有关；$B_{\mathrm{w}}^{\mathrm{GT}}$、$B_{\mathrm{h}}^{\mathrm{GT}}$ 分别为真实框的宽度、高度；ϑ 为比例因子，与数据集中目标的比例有关；ϕ_{shape} 为考虑框的形状因素的距离项；x_{c}、y_{c} 为预测框中心坐标；$x_{\mathrm{c}}^{\mathrm{GT}}$、$y_{\mathrm{c}}^{\mathrm{GT}}$ 为真实框中心坐标；v 为预测框与真实框的最小外接框的尺度因子；Γ_{shape} 为形状损失项；B^{w}、B^{h} 分别为预测框的宽度、高度；φ 为固定的比例系数，一般取 4；$\zeta_{B^{\mathrm{w}}}$、$\zeta_{B^{\mathrm{h}}}$ 分别为宽度方向、高度方向的形状损失因子。

Shape-IoU 的损失函数为

$$L_{\text{Shape-IoU}} = 1 - \text{IoU} + \phi_{\text{shape}} + 0.5 \times \Gamma_{\text{shape}} \tag{21}$$

2.3.4　Wise-IoU

Wise-IoU 通过引入动态聚焦因子，根据锚框中心与真实框中心的相对位置，对 IoU 损失做再加权，从而抑制低质量样本的有害梯度，避免高质量样本过度竞争，并将更多梯度分配给普通质量样本。Wise-IoU 的定义为

$$\text{IoU}_{\text{Wise}} = \exp\left(\frac{(x - x^{\mathrm{GT}})^2 + (y - y^{\mathrm{GT}})^2}{(I^2 + U^2)^{\tau}}\right) \tag{22}$$

式中：x、y 为预测框中心坐标；x^{GT}、y^{GT} 为真实框中心坐标；I、U 分别为最小外接框的宽度、高度；τ 为梯度分离算子，用于阻断 I、U 的梯度回传，防止 IoU_{Wise} 产生阻碍收敛的梯度。在此基础上，Wise-IoU 的损失函数为

$$L_{\text{Wise-IoU}} = L_{\text{IoU}}\text{IoU}_{\text{Wise}} \tag{23}$$

2.3.5　FSW-IoU

在 Shape-IoU 的基础上引入 Focaler-IoU 的思想，得到聚焦形状感知交并比（Focaler-Shape-IoU）损失函数，其表达式为

$$L_{\text{Focaler-Shape-IoU}} = L_{\text{Shape-IoU}} + \lambda(\text{IoU} - \text{IoU}_{\text{Focaler}}) \tag{24}$$

式中 λ 为调整系数，用于动态调节损失函数，在增强算法对背景复杂或难以检测目标的关注的同时提高对预测偏差较大边界框的敏感性，从而优化困难目标的检测性能。

保持 Wise-IoU 对锚框质量动态聚焦，同时引入 Shape-IoU 的形状、尺度约束以及 Focaler-IoU 的样本难度重构思想，得到最终的 FSW-IoU 损失函数。

$$L_{\text{FSW-IoU}} = L_{\text{Wise-IoU}} + L_{\text{Focaler-Shape-IoU}} \tag{25}$$

为验证 3 个模块递进融合的性能，对其进行简单的消融实验。将 YOLOv8s 的基础损失函数分别替换为 Shape-IoU、Focaler-Shape-IoU、FSW-IoU，通过 mAP@0.5 和 mAP@0.5:0.95 衡量模块的性能，实验结果见表 2。从表 2 可以看出，针对复杂背景下的小目标检测任务，递进融合后的模块 FSW-IoU 具有最好的检测性能。

表 2　模块递进融合实验结果

Tab. 2　Experimental results of progressive module integration

%

模块	mAP@0.5	mAP@0.5:0.95
YOLOv8s	38.3	22.5
+ Shape-IoU	38.8	22.9
+ Focaler-Shape-IoU	39.2	23.2
+ FSW-IoU	**39.4**	**23.4**

2.4　DyHead 模块

YOLOv8s 的检测头提供了优异的速度和实时性，适合大多数标准检测任务，但在小目标检测上仍存在不足。主要问题在于其固定的卷积核尺寸限制了对小目标细节和边界的捕捉能力，导致特征融合效果不佳且回归精度不足。与之相比，DyHead 通过动态卷积和自适应回归机制，能够灵活调整卷积核的尺寸和形状，增强特征提取并减少背景干扰，尤其是在小目标的尺寸、形状及分辨率较低时表现更为优异，所以本文采用 DyHead 替换基线网络的检测头。DyHead 模块结构如图 7 所示。其中硬 Sigmoid 激活函数（Hard Sigmoid）用于将 1×1 卷积后的输出压缩至[0,1]，作为每个尺度的门控权重；线性整流激活函数（ReLU）用于增强尺度权重的非线性表达能力；采样偏移量（Offset）用于根据预测的偏移从不同空间位置自适应聚合特征，实现空间感知注意力；索引生成单元（Index）用于根据当前特征图的位置生成基础坐标网格，为偏移采样提供参考；归一化处理单元（Normalize）用于对任务权重向量进行归一化，防止某一任务权重过大导致训练不稳定；基准任务权重向量（1,0,0,0）用于表示原始检测头的默认任务权重，在此基础上叠加学习到的任务偏移以获得最终的任务组合权重。

假设当前有特征张量 $\boldsymbol{\Phi} \in \mathbf{R}^{K \times X \times O}$，$K$ 为尺度特征层数，X 为空间位置数，O 为通道数，DyHead 模块将函数转换为 3 个连续的注意力（尺度感知注意力、空间感知注意力和任务感知注意力，其门控权重向量、聚合权重向量和任务组合权重向量分别为 $\boldsymbol{\Omega}_{\mathrm{KL}}$、$\boldsymbol{\Omega}_{\mathrm{XL}}$ 和 $\boldsymbol{\Omega}_{\mathrm{OL}}$），每个注意力只关注一个视角。DyHead 对输入特征张量施加 3 个注意力之后得到的整体变换结果为 $\boldsymbol{\Theta}(\boldsymbol{\Phi})$，其表达式为：

$$\boldsymbol{\Theta}(\boldsymbol{\Phi}) = \pi_O(\pi_X(\pi_K(\boldsymbol{\Phi}) \cdot \boldsymbol{\Phi}) \cdot \boldsymbol{\Phi}) \cdot \boldsymbol{\Phi} \tag{26}$$

$$\pi_K(\boldsymbol{\Phi}) \cdot \boldsymbol{\Phi} = \sigma\left(\partial\left(\frac{1}{XO}\sum_{\gamma=1}^{X}\sum_{\varepsilon=1}^{O}\iota_{\gamma,\varepsilon}\right)\right) \cdot \boldsymbol{\Phi} \tag{27}$$

$$\sigma(\xi) = \max\left(0, \min\left(1, \frac{\xi + 1}{2}\right)\right) \tag{28}$$

$$\pi_X(\boldsymbol{\Phi})\cdot\boldsymbol{\Phi}=\frac{1}{K}\sum_{\delta=1}^{K}\sum_{l=1}^{\zeta}J_{\delta,l}\cdot\boldsymbol{\Phi}(\delta;\omega_l+\Delta\omega_l;\varepsilon)\cdot\Delta V_l \tag{29}$$

$$\pi_O(\boldsymbol{\Phi})\cdot\boldsymbol{\Phi}=\max(\alpha_1(\boldsymbol{\Phi})\cdot\boldsymbol{\psi}_O+\beta_1(\boldsymbol{\Phi}),\alpha_2(\boldsymbol{\Phi})\cdot\boldsymbol{\psi}_O+\beta_2(\boldsymbol{\Phi})) \tag{30}$$

式中：π_K、π_X、π_O 分别为作用于尺度维、空间维和通道维的注意力函数；$\sigma(\xi)$ 为一个严格激活函数；$\partial(\cdot)$ 为由 1×1 卷积层近似的线性函数；$\iota_{\gamma,\varepsilon}$ 为在第 γ 个空间位置、第 ε 个通道上沿尺度维度聚合得到的特征；ζ 为稀疏采样位置的数量；l 为稀疏采样点的索引；δ 为尺度层索引；$J_{\delta,l}$ 为空间聚合的加权系数；$\omega_l+\Delta\omega_l$ 为通过自学习空间偏移 $\Delta\omega_l$ 移动的位置，通过其使得注意力聚焦于需要判别的区域；ΔV_l 为位置 ω_l 处的自学习标量；$\boldsymbol{\psi}_O$ 为第 O 个通道的输入特征；α_1、β_1、α_2、β_2 分别为第 1 段、第 2 段线性映射的斜率参数、偏置参数。

DyHead 将尺度、空间和任务信息统一到同一注意力机制中，在提升特征表达能力的同时增强多尺度和任务自适应性，并通过简化网络设计减少任务间冲突和冗余。

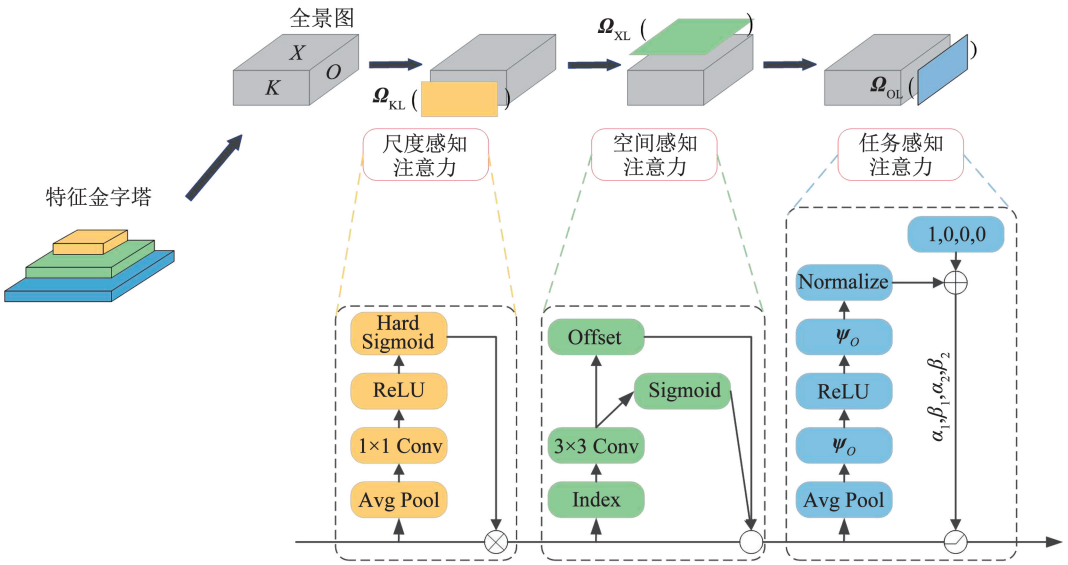


图 7 DyHead 模块结构

Fig. 7 DyHead module architecture

3 实验结果与分析

3.1 数据集

VisDrone-2019[19] 数据集专为无人机视角下的小目标检测任务构建，包含多种复杂场景中的高分辨率图像，重点聚焦远距离行人、车辆、建筑物等小尺寸目标的检测。由于无人机拍摄存在远距离视角、斜视成像以及目标遮挡等问题，目标在图像中只占据极小区域，且常常被复杂背景或其他目标遮挡，使得小目标在特征表达和定位精度上均面临很大挑战。

VisDrone-2019 数据集涵盖多种场景与天气条件下的图像和视频序列，共有 10 个预定义类别，并提供精确的边界框与类别标注。图像部分划分为训练集(6 471 张图像)和验证集(548 张图像)，其中小尺寸目标占比较高。以 people 类别为例，超过 77% 的目标属于小目标，且遮挡面积超过 50% 的样本接近一半。多分辨率、复杂背景与多视角共同叠加，使该数据集成为评估和优化无人机端小目标检测算法的重要基准平台。

3.2 实验设备及评价指标

实验硬件环境为 NVIDIA GeForce RTX 3080 显卡，内存 30 G；软件环境为 PyTorch 1.11.0、Python 3.8、cuda 11.3 等。性能评价指标包括平均检测精度 mAP@0.5、参数量、计算量和推理速度。

3.3 消融实验及可视化结果

在 VisDrone-2019 数据集上进行算法性能验证，采用改进后的模块替换基线 YOLOv8s 算法中的模块进行消融实验，结果见表 3，其中， 表示未添加此模块，√表示添加了此模块。

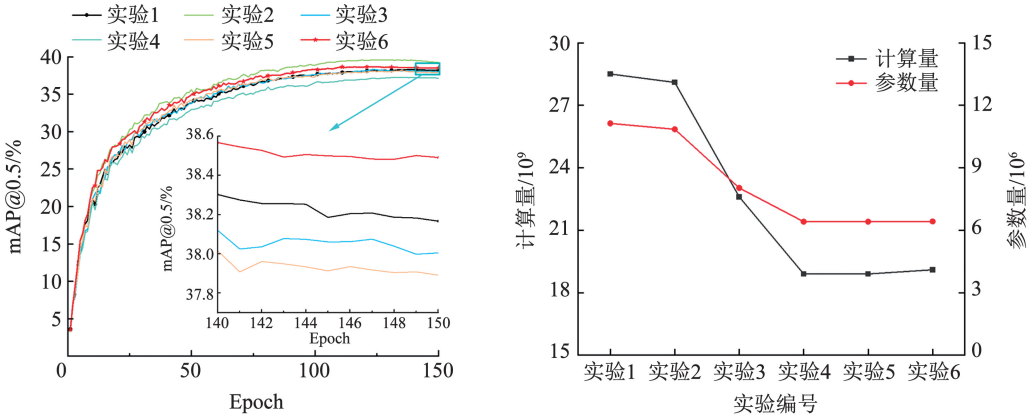
表 3　消融实验结果

Tab. 3　Results of ablation experiment

编号	DyHead	GD_HGNetV2	C2f_GDC	FSW-IoU	C2f_FB_EMA	mAP@0.5/%	参数量/10^6	计算量/10^9	推理速度/(帧·s^{-1})
实验 1(YOLOv8s)						38.3	11.13	28.5	83.0
实验 2	√					39.4	10.85	28.1	64.0
实验 3	√	√				38.0	8.03	22.6	64.4
实验 4	√	√	√			37.2	6.41	18.9	66.9
实验 5	√	√	√	√		37.8	6.41	18.9	66.1
实验 6(YOLOv8s-LW)	√	√	√	√	√	38.5	6.40	19.2	64.0

由表 3 可以看出：从实验 2 开始，引入 DyHead 模块替代原有的检测头，提升了算法的检测精度与灵活性；在实验 3 中，采用 GD_HGNetV2 主干网络替换了基线网络的主干部分，从而有效增强了特征提取能力；在实验 4 ~ 实验 6 中，依次引入了 C2f_GDC、FSW-IoU 和 C2f_FB_EMA 模块，每一个模块的加入均在不同程度上优化了算法的性能与鲁棒性。

根据表 3 中的实验数据，改进后的 YOLOv8s-LW 相较于原始 YOLOv8s，在复杂背景下的小目标检测任务中展现出了更为优异的表现，检测精度提高了 0.2%。同时，得益于模块化的优化设计，YOLOv8s-LW 算法的参数量减少了近 43%，计算量降低了近 33%。尽管推理速度略有下降(降幅为 19.0 帧/s)，但依然充分满足了实时检测任务的要求，确保了高效推理与精确检测的平衡。最终，通过一系列模块的引入，YOLOv8s-LW 展现了更强的适应性与优化潜力，在保证精度的同时，进一步提升了计算效率。图 8 为消融实验的每轮检测精度、计算量与参数量。



(a) 每轮检测精度　　(b) 计算量与参数量

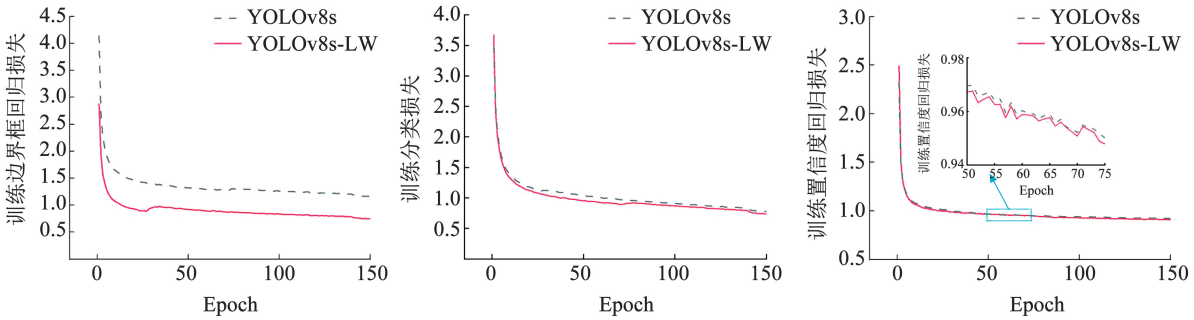
图 8　消融实验的每轮检测精度、计算量与参数量

Fig. 8　Detection accuracy of each round, computational cost, and parameter quantity of ablation experiment

由图 8 可以看出，在仅引入 DyHead 模块的情况下，算法的平均检测精度达到了最高水平，说明 DyHead 模块在提升检测性能方面具有显著优势。随着后续实验中逐步引入轻量化主干网络 GD_HGNetV2 以及轻量化模块 C2f_GDC，虽然算法的检测精度略有下降，但算法的参数量和计算复杂度也显著减少，展现了轻量化设计在优化计算效率方面的有效性。在最后两个实验中，分别加入了新设计的 FSW-IoU 和 C2f_FB_EMA 模块，这些模块不仅成功提升了检测精度，还有效维持了算法的参数量和计算量水平。更为重要的是，尽管进行了多次优化，算法的推理速度仍稳定保持在 60.0 帧/s 以上，满足了实时检测任务的需求。综上所述，经过模块化设计并逐步优化，改进后的算法在保证精度的同时，成功平衡了计算效率与实时性，展现了强大的实际应用潜力。图 9 为算法改进前后训练损失值对比曲线。

由图 9(a)可以看出，在训练初期(0 ~ 10 轮)，两种算法的训练边界框回归损失相近，但随着训练迭代的深入，YOLOv8s-LW 在训练边界框回归方面显著优于 YOLOv8s，体现了其在回归精度上的持续优化。从图 9(b)和图 9(c)可以看出，尽管 YOLOv8s-LW 在训练分类损失和训练置信度回归损失上未表现出大幅下降，然而随着训练的进一步进

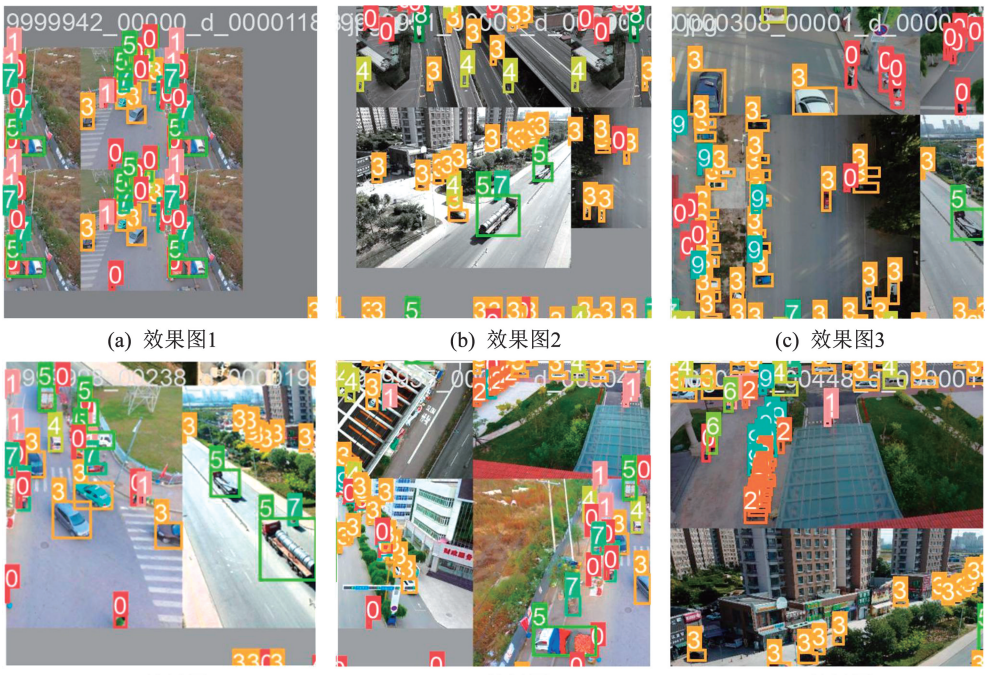
行,该算法仍表现出一定的优势。综合对比图 9 可知,轻量化设计可能带来某些损失,但本文提出的改进算法有效缓解了轻量化过程中可能出现的性能下降,提升了基线网络在精度上的表现。因此,改进后的算法能够在保证计算效率的同时,较好地维持和提升检测任务的整体性能,充分展示了其在优化轻量化网络中的潜力和优势。图 10 和图 11 分别展示了 YOLOv8s-LW 的检测效果和热力图。



(a) 训练边界框回归损失 (b) 训练分类损失 (c) 训练置信度回归损失

图 9 训练损失值对比曲线

Fig. 9 Comparison curves of training loss values

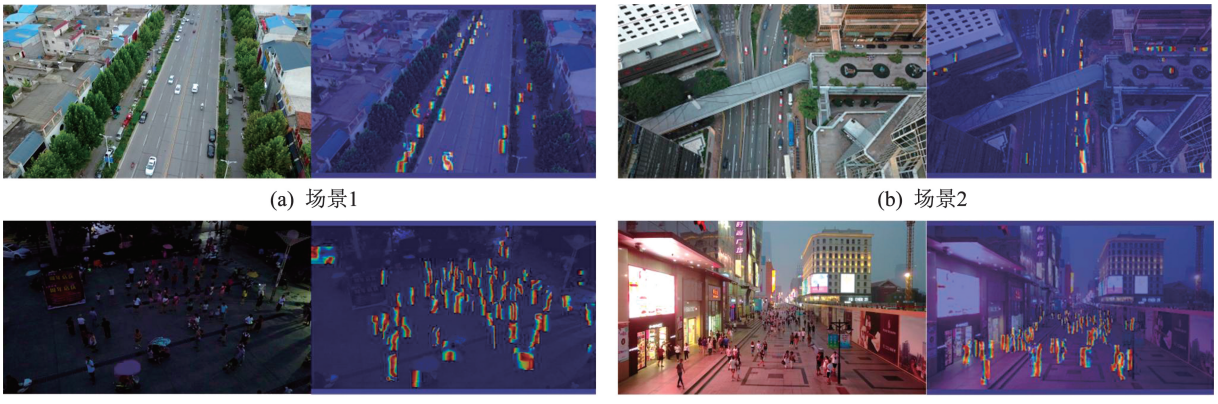


(a) 效果图1 (b) 效果图2 (c) 效果图3

(d) 效果图4 (e) 效果图5 (f) 效果图6

图 10 YOLOv8s-LW 检测效果

Fig. 10 YOLOv8s-LW detection effect



(a) 场景1 (b) 场景2

(c) 场景3 (d) 场景4

图 11 YOLOv8s-LW 检测热力图

Fig. 11 YOLOv8s-LW detection heat maps

从图 11 可以看出，约 90% 的被检测目标的整体区域集中在较高的关注度范围内，且预测的边界框恰好围绕该区域，表明 YOLOv8s-LW 在进行区域选择时展现了较高的合理性与精准度。针对小目标（定义为尺寸在 20 × 20 ~ 50 × 50 像素之间），根据检测结果与热力图的综合分析，本文算法能够准确地识别并定位所有小目标，证明了其在处理尺度较小目标时的有效性和鲁棒性。

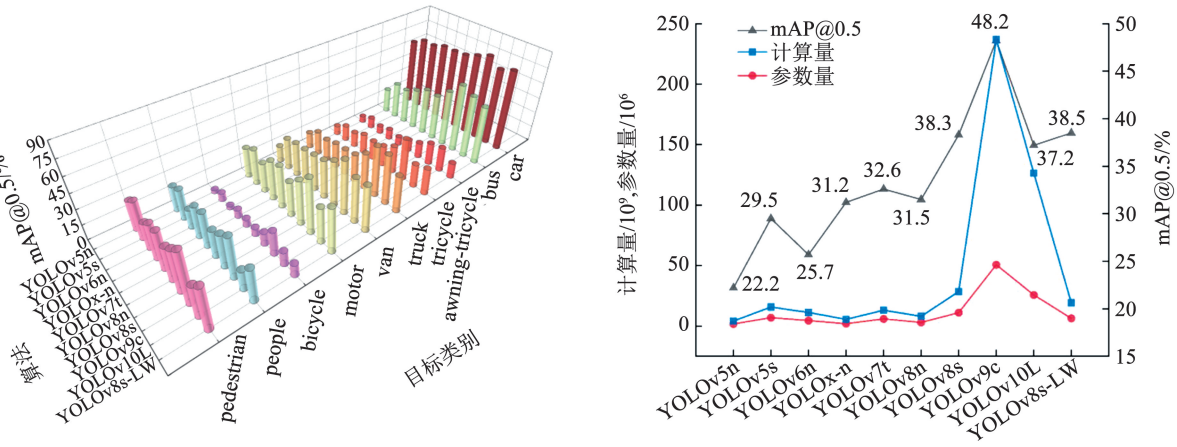
3.4　对比实验

为验证本文提出的算法在无人机执行复杂场景中的小目标检测任务时的轻量性和准确性，将本文所提算法与一些经典算法进行对比，实验结果见表 4 和图 12。

表 4　YOLOv8s-LW 与其他算法的对比结果

Tab. 4　Comparison results of YOLOv8s-LW with other algorithms

算法	pedestrian	people	bicycle	motor	van	truck	tricycle	awning-tricycle	bus	car	mAP@0.5/%	参数量/10^6	计算量/10^9
YOLOv5n	29.2	24.6	2.9	29.4	15.9	15.4	8.8	6.1	23.9	65.5	22.2	1.8	4.2
YOLOv5s	37.6	29.6	7.7	35.1	30.5	22.9	15.1	10.2	35.2	71.3	29.5	7.0	15.8
YOLOv6n	26.0	18.7	3.0	28.6	32.8	21.5	14.3	7.3	34.4	70.8	25.7	4.6	11.3
YOLOx-n	34.7	30.6	7.6	38.4	33.8	24.3	18.8	9.8	39.6	74.3	31.2	2.0	5.6
YOLOv7t	37.4	33.6	7.1	41.1	35.4	25.1	18.2	8.1	45.5	74.2	32.6	6.0	13.1
YOLOv8n	32.9	26.2	7.2	33.4	37.5	26.9	20.1	11.3	44.8	75.0	31.5	3.0	8.1
YOLOv8s	41.1	32.2	12.0	43.1	44.3	35.1	26.5	15.0	54.9	79.0	38.3	11.1	28.5
YOLOv9c	50.8	40.3	23.0	53.4	52.8	49.4	40.8	21.3	66.6	83.4	48.2	50.7	236.7
YOLOv10L	30.5	18.9	13.2	34.5	43.7	48.5	22.5	23.7	61.3	75.5	37.2	25.7	126.4
YOLOv8s-LW	**41.4**	**31.8**	**11.5**	**43.6**	**44.6**	**35.9**	**25.7**	**14.8**	**56.6**	**79.3**	**38.5**	**6.4**	**19.2**



(a) 对比实验结果　　(b) mAP@0.5、参数量和计算量

图 12　对比实验结果及 mAP@0.5、参数量和计算量

Fig. 12　Comparison of experimental results and mAP@0.5, parameter quantity, and computational cost

在 VisDrone-2019 数据集上，大多数现有算法的检测精度普遍较低，主要原因在于该数据集图像中目标尺度小、背景复杂且遮挡严重。YOLOv9c 算法在该数据集上表现出了最高的平均检测精度，达到了 48.2%。YOLOv8s-LW 的检测精度为 38.5%，尽管相较于 YOLOv9c 低了 9.7%，但 YOLOv9c 的参数量和计算量分别为本文算法的 7.9 倍和 12.3 倍。

通过对小目标检测的平均精度、参数量、计算量、推理速度的综合分析发现，本文提出的 YOLOv8s-LW 算法在保持较低计算开销的同时，能够高效完成实时检测任务。该算法在显著减少计算资源消耗的前提下，仍能保持较为优异的检测性能，充分体现了其在性能与效率平衡方面的优势。

4　结　论

1）本文提出的轻量化基线 YOLOv8s-LW 以主干降算、颈部重标定、稳健回归、动态检测头的组合范式，在 GD_HGNetV2 基础上压缩冗余，在 C2f_

GDC/C2f_FB_EMA 中引入空间混合与多尺度注意力,以 FSW-IoU 强化回归鲁棒性,并配合 DyHead 提升复杂背景下的小目标响应与定位一致性,形成结构清晰、可插拔、可迁移的轻量化框架。

2)在 VisDrone-2019 数据集上,YOLOv8s-LW 的 mAP@0.5 达到 38.5%,参数量为 6.4×10^6、计算量为 19.2×10^9、推理速度为 64.0 帧/s;相较基线 YOLOv8s,参数量降低了约 43%、计算量降低了约 33%,同时精度提升了 0.2%,实现了更优的精度与效率权衡并满足端侧实时性需求。

3)消融实验表明,各子模块均带来稳定且可叠加的收益,GD_HGNetV2 降算提速,C2f_GDC 和 C2f_FB_EMA 强化多尺度特征表达,DyHead 与 FSW-IoU 提升定位与回归稳定性。整体方案训练与部署门槛低、与主流推理框架兼容性好,可作为资源受限无人机场景的通用轻量化基线,具备良好的工程落地性与复现性。

参考文献

[1] Wu Xin, Li Wei, Hong Danfeng, et al. Deep learning for unmanned aerial vehicle-based object detection and tracking: a survey[J]. IEEE Geoscience and Remote Sensing Magazine, 2022, 10(1): 91. DOI:10.1109/MGRS.2021.3115137

[2] Cao Yukang, Pang Dandan, Zhao Qianchuan. Improved YOLOv8-GD deep learning model for defect detection in electroluminescence images of solar photovoltaic modules[J]. Engineering Applications of Artificial Intelligence, 2024, 131: 107866. DOI: 10.1016/j.engappai.2024.107866

[3] Varghese R, Sambath M. YOLOv8: a novel object detection algorithm with enhanced performance and robustness [C]// Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS 2024). Chennai: IEEE, 2024: 1. DOI: 10.1109/ADICS58448.2024.10533619

[4] Cao Shihai, Wang Ting, Li Tao, et al. UAV small target detection algorithm based on an improved YOLOv5s model[J]. Journal of Visual Communication and Image Representation, 2023, 97: 103936. DOI:10.1016/j.jvcir.2023.103936

[5] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016: 779. DOI:10.1109/CVPR.2016.91

[6] 凌强, 刘宇, 王春举, 等. DN-YOLOv5 的金属双极板表面缺陷检测算法[J]. 哈尔滨工业大学学报, 2023, 55(12): 104
Ling Qiang, Liu Yu, Wang Chunju, et al. DN-YOLOv5 algorithm for detecting surface defects of metal bipolar plates[J]. Journal of Harbin Institute of Technology, 2023, 55(12): 104. DOI: 10.11918/202212004

[7] 杨志渊, 罗亮, 吴天阳, 等. 改进 YOLOv8 的轻量级光学遥感图像船舶目标检测算法[J]. 计算机工程与应用, 2024, 60(16): 248
Yang Zhiyuan, Luo Liang, Wu Tianyang, et al. Improved lightweight ship target detection algorithm for optical remote sensing images with YOLOv8[J]. Computer Engineering and Applications, 2024, 60(16): 248. DOI:10.3778/j.issn.1002-8331.2403-0217

[8] Han Kai, Wang Yunhe, Tian Qi, et al. GhostNet: more features from cheap operations[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle: IEEE, 2020: 1577. DOI: 10.1109/CVPR42600.2020.00165

[9] Nascimento M G D, Prisacariu V, Fawcett R. DSConv: efficient convolution operator[C]//Proceedings of the 17th IEEE/CVF International Conference on Computer Vision (ICCV). Seoul: IEEE, 2019: 5148. DOI: 10.1109/ICCV.2019.00525

[10] Zhao Yian, Lü Wenyu, Xu Shangliang, et al. DETRs beat YOLOs on real-time object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle: IEEE, 2024: 16965. DOI: 10.1109/CVPR52733.2024.01605

[11] Zhang Hao, Zhang Shuaijie. Focaler-IoU: more focused intersection over union loss[PP/OL]. V2. arXiv (2024-01-19)[2024-11-20]. https://doi.org/10.48550/arXiv.2401.10525

[12] Zhang Hao, Zhang Shuaijie. Shape-IoU: more accurate metric considering bounding box shape and scale[PP/OL]. V2. arXiv (2024-01-12)[2024-11-20]. https://doi.org/10.48550/arXiv.2312.17663

[13] Tong Zanjia, Chen Yuhang, Xu Zewei, et al. Wise-IoU: bounding box regression loss with dynamic focusing mechanism[PP/OL]. V3. arXiv (2023-04-08)[2024-11-20]. https://arXiv.org/abs/2301.10051

[14] Dai Xiyang, Chen Yinpeng, Xiao Bin, et al. Dynamic head: unifying object detection heads with attentions[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville: IEEE, 2021: 7373. DOI:10.1109/CVPR46437.2021.00729

[15] Liu Wei, Anguelov D, Erhan D, et al. SSD: single shot multibox detector[C]//14th European Conference on Computer Vision (ECCV 2016). Amsterdam: Springer, 2016: 21. DOI:10.1007/978-3-319-46448-0_2

[16] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV). Venice: IEEE, 2017: 2999. DOI:10.1109/ICCV.2017.324

[17] Ouyang Daliang, He Su, Zhang Guozhong, et al. Efficient multi-scale attention module with cross-spatial learning[C]//48th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023). Rhodes Island: IEEE, 2023: 1. DOI: 10.1109/ICASSP49357.2023.10096516

[18] Chen J, Kao S H, He H, et al. Run, don't walk: chasing higher FLOPS for faster neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver: IEEE, 2023: 12021. DOI: 10.1109/CVPR52729.2023.01157

[19] Du Dawei, Zhu Pengfei, Wen Longyin, et al. VisDrone-DET2019: the vision meets drone object detection in image challenge results[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). Seoul: IEEE, 2019: 213. DOI:10.1109/ICCVW.2019.00030