

# 基于改进 YOLOv8 的水稻病害检测<sup>①</sup>



聂俊<sup>1</sup>, 朱节中<sup>1,2</sup>

<sup>1</sup>(南京信息工程大学 自动化学院, 南京 210044)

<sup>2</sup>(南京信息工程大学 软件学院, 南京 210044)

通信作者: 聂俊, E-mail: 2352209710@qq.com

**摘要:** 本研究提出了一种改进的 YOLOv8 模型 (FCU-YOLOv8), 用于提升水稻病害检测的精度和效率, 以应对水稻病害种类繁多、背景复杂及病害间特征差异小等问题. 在 YOLOv8 主干网络的 C2f 模块基础上, 采用了 FasterNeXt 模块替换. FasterNeXt 模块通过优化网络结构减少了计算量和内存访问量, 同时提高了特征提取的效率, 从而降低模型的推理成本. 设计了 C3K 模块 (多尺度卷积模块) 和 CPSA 模块 (卷积注意力机制), 以进一步提升模型对病害区域的特征感知能力. C3K 模块允许模型通过灵活的卷积核选择适应不同尺度的病害特征, 而 CPSA 模块利用注意力机制增强模型对关键信息的捕捉. 为了提升检测框的质量和密集病害目标的检测效果, 模型采用了优化的 UIoU (unified intersection over union) 损失函数, 该函数在回归阶段通过平衡边界框的精确性与一致性来提升检测性能. 在自制的 8 种常见水稻病害图像数据集上, FCU-YOLOv8 在多个性能指标上相较于原始 YOLOv8 有显著提升, 其中  $mAP@0.5$  指标达到 94.7%, 相较于基线模型提升了 2.4%,  $mAP@0.5:0.95$  指标达到了 67.2%, 相较于基线模型提高 3.3%, 在轻量化方面, 模型参数相较于基线模型降低了 24.2%, 计算浮点数下降 28.7%. 与主流算法进行对比实验, 所提算法表现优于目前主流算法, 说明了该网络的有效性.

**关键词:** YOLOv8; 深度学习; 部分卷积; 注意力机制; UIoU

引用格式: 聂俊, 朱节中. 基于改进 YOLOv8 的水稻病害检测. 计算机系统应用, 2025, 34(5): 1-14. <http://www.c-s-a.org.cn/1003-3254/9894.html>

## Rice Disease Detection Based on Improved YOLOv8

NIE Jun<sup>1</sup>, ZHU Jie-Zhong<sup>1,2</sup>

<sup>1</sup>(School of Automation, Nanjing University of Information Science & Technology, Nanjing 210044, China)

<sup>2</sup>(School of Software, Nanjing University of Information Science & Technology, Nanjing 210044, China)

**Abstract:** An improved YOLOv8 model (FCU-YOLOv8) is proposed to enhance the accuracy and efficiency of rice disease detection, addressing the challenges of diverse rice diseases, complex backgrounds, and subtle differences in characteristics between diseases. The FasterNeXt module is used to replace the C2f module in the YOLOv8 backbone network. By optimizing the network structure, the FasterNeXt module reduces computation and memory access while improving feature extraction efficiency, thus lowering the inference cost of the model. The C3K module (multi-scale convolution module) and CPSA module (convolutional attention mechanism) are designed to further enhance the model's ability to perceive disease region features. The C3K module allows the model to adapt to disease characteristics at various scales through flexible convolutional kernel selection, while the CPSA module employs an attention mechanism to enhance the model's ability to capture key information. To improve the quality of detection boxes and the detection performance of dense disease targets, the optimized unified intersection over union (UIoU) loss function is adopted. This function improves detection performance by balancing the accuracy and consistency of bounding boxes during the

① 基金项目: 国家自然科学基金面上项目 (42275200)

收稿时间: 2024-10-24; 修改时间: 2025-01-15; 采用时间: 2025-01-24; csa 在线出版时间: 2025-04-01

regression phase. On a custom-made image dataset of eight common rice diseases, FCU-YOLOv8 demonstrates significant improvements over the original YOLOv8 in several performance metrics. The  $mAP@0.5$  index reaches 94.7%, a 2.4% improvement over the baseline model, and the  $mAP@0.5:0.95$  index reaches 67.2%, a 3.3% improvement. The model's parameters are reduced by 24.2%, and the calculated floating-point operations decrease by 28.7%, compared to the baseline model in terms of model lightweighting. Compared with mainstream algorithms, the proposed algorithm outperforms current leading algorithms, demonstrating the effectiveness of the network.

**Key words:** YOLOv8; deep learning; partial convolution; attention mechanism; UIoU

水稻是世界范围内最重要的粮食作物之一,其产量直接影响到全球数十亿人口的粮食供给安全.特别是在亚洲,水稻是主要的食物来源,中国的水稻产量约占全球总产量的40%<sup>[1]</sup>.然而,在生长过程中,水稻容易受到多种病害的侵袭,包括稻瘟病、纹枯病、白叶枯病、东格鲁病、褐斑病、鞘腐病、死心病、条纹病及稻霉病等.这些病害不仅会对水稻的产量造成严重威胁,还会影响其质量,带来显著的经济损失<sup>[2]</sup>.因此,及时且准确地检测水稻病害,对于控制病害的传播、制定有效的农业管理措施、以及提升水稻产量至关重要.

传统的水稻病害检测主要依赖于农学专家的肉眼观察和诊断.尽管这一方法在过去发挥了重要作用,但其局限性也十分明显.首先,人工检测耗时且耗力,尤其在大规模农业环境中,人工检测难以满足快速和高效的需求.其次,专家对病害的判断可能存在主观差异,导致结果的不确定性和误差.此外,农业专家的资源分布不均,尤其在农村和欠发达地区,资源匮乏进一步影响了病害诊断的效率和准确性<sup>[3]</sup>.

同时,基于机器学习的传统图像处理方法也被用于水稻病害检测.这些方法通常需要手动提取图像特征,再通过分类器(如支持向量机或随机森林)进行病害分类.然而,面对复杂的环境和多样的病害特征,这些方法的表现往往不理想.由于病害的多样性、病斑形状的变化以及不同病害之间的相似性,传统机器学习方法难以实现高精度的检测<sup>[4]</sup>.因此,当前的研究重点在于开发更加自动化且鲁棒性强的病害检测方法.

近年来,随着深度学习技术的飞速发展,卷积神经网络(CNN)在图像分类和目标检测等任务上取得了显著进展<sup>[5]</sup>.在农业领域,深度学习方法被广泛应用于作物病害的检测与分类.相比传统的图像处理技术,深度学习无需手动提取特征,而是能够自动从大量数据中学习复杂的病害特征,极大地提升了检测的准确性与

效率<sup>[6]</sup>.在两阶段目标检测器的研究中,2019年周飞燕等人<sup>[7]</sup>基于Faster R-CNN提出了一种方法,展示了其在水稻病害识别方面的优异表现,尤其在速度和精度上表现出色.2020年,Facebook研究团队提出了DETR(detection Transformer),将目标检测转化为预测问题,并采用统一的编码器-解码器架构来处理分类和定位任务. DETR简化了传统两阶段方法的流程,同时在性能上与Faster R-CNN相当<sup>[8]</sup>.2021年,Bari等人<sup>[9]</sup>提出了一种改进的区域建议网络(RPN)架构,大幅提升了目标定位的精度,进一步推动了基于Faster R-CNN的水稻叶片病害实时检测. Zhang等人<sup>[10]</sup>提出了一种针对水稻病害识别的改进型Mask R-CNN方法,通过优化特征融合机制和引入多尺度膨胀卷积层来扩大特征图的感受野,同时保持其分辨率,从而减少下采样过程中的细节损失.与标准Mask R-CNN相比,该方法在交并比阈值为0.6、0.7和0.8时,识别准确率分别提高了0.46%、0.66%和0.48%.2023年, Yang等人<sup>[11]</sup>设计了密集高层次组合特征金字塔网络(DHLC-FPN),通过将DETR的骨干网络ResNet替换为DHLC-FPN和Res2Net的组合,提出了DHLC-DETR算法.该算法增强了特征提取能力,并有效解决了传统卷积网络中高级和低级语义信息丢失的问题.在小目标检测方面, DHLC-DETR的 $mAP$ 提升了17.3%,尤其在微小病害检测中 $mAP$ 提高了9.5%.在水稻病害识别测试中,该算法的平均准确率达到97.44%.在一阶段目标检测器的研究中,严陈慧子等人<sup>[12]</sup>提出一种基于改进YOLOv4的水稻病害检测方法,使用MobileNetV3替代CSP-DarkNet-53,并通过在颈部网络中添加坐标注意力模块(CAM)来提升模型性能.周思捷等人在YOLOv5算法中引入了混合域注意力机制来校正特征,提升对水稻叶片和病斑位置信息的定位能力,并使用SIoU损失函数替换CIoU,改进算法在病害识别中的 $mAP$ 达到

了0.884,比原YOLOv5提升了2.9个百分点<sup>[13]</sup>.2023年,Yuan等人<sup>[14]</sup>提出一种基于改进YOLOv5的轻量化算法,用于识别杂草稻,为快速高效的田间监测提供了新方法.Jia等人<sup>[15]</sup>基于改进YOLOv7算法,提出了一个新的水稻病虫害识别模型,采用MobileNetV3进行特征提取,并引入坐标注意力机制(CA)和Siou损失函数,以提高精度.戴林华等人<sup>[16]</sup>基于YOLOv8提出了改进方法,使用ADown模块减少特征信息丢失,并引入SE注意力机制提升特征融合能力,同时使用WIoU损失函数进一步优化检测性能.

基于上述分析,水稻叶片病害检测面临多个挑战,如病害种类多样、背景复杂、不同病害的表型特征差异不显著,以及病害初期病斑较小等问题.本文以稻瘟病、纹枯病、白叶枯病、东格鲁病、褐斑病、鞘腐病、死心病、条纹病和假性病9种病害为研究对象,并基于YOLOv8网络提出一种改进模型(FCU-YOLOv8).在主干网络中,使用PConv替代普通卷积构建Faster-NeXt<sup>[17]</sup>,优化C2f结构以减少冗余计算和内存访问,从而更有效地提取空间特征.此外,增加了增强型卷积模块用于捕捉复杂特征.在Neck网络部分,设计了高效特征提取模块C2fK3,通过跨阶段结构和灵活模块选择,提升网络对密集病害目标的检测能力以及表示学习和计算效率.由于水稻叶片病害目标密集,且不同类别的表征差异较小,容易导致预测框质量参差不齐,影响检测框的精度,因此在包围框回归损失中加入UIoU(Unified-IoU)<sup>[18]</sup>,赋予高质量预测框更高权重,提升密集目标的检测精度.

## 1 数据材料

该研究所使用得到数据集共包含8种常见的水稻叶片病害:稻瘟病、白叶枯、胡麻叶斑病、枯心病、稻曲病、鞘病、叶纹病、东格鲁病.8种常见水稻病害和1种正常叶片作为背景类数据的数据集2253张.如图1所示,其中水稻病害数据集基于两个数据集构建,第1部分来自于Sethy等人<sup>[19]</sup>开源的4类别的水稻病害数据集Rice Leaf Disease Image Samples;第2部分是安徽农业大学和中国科学院合肥物质科学研究院官网提供的部分水稻样例数据.其中Rice Leaf Disease Image Samples数据集图像分辨率是640×640,第2部分数据图像分辨率是4032×3024.为了提升模型的训练效率和降低内存需求模型训练均采用640尺度为输入.



图1 水稻病害数据集部分

为了解决数据集中某些病害类别图像过多而其他类别图像过少的问题,防止模型训练时出现过度拟合或不足拟合的现象,并提升模型的稳定性和适应新数据的能力,我们实施以下图像处理策略:对图像进行水平或垂直翻转、随机旋转一定角度、调整亮度以模拟不同光照条件以及通过算法增强或降低图像的对比度.这些措施有助于丰富数据集,使模型训练更加全面.如表1所示,数据集以8:1:1的比例划分为训练集、验证集、测试集,训练集和验证集数据增强后共14039张图像,其中测试集中均为未进行增强的原始数据.

表1 水稻病害数据集构成

水稻病害	样本总数	训练集	验证集	测试集
稻瘟病	3859	3249	549	61
白枯病	2288	1908	342	38
胡麻叶斑病	1076	1026	135	15
枯心病	3067	2509	502	56
稻曲病	1430	1170	234	26
鞘病	1100	900	180	20
叶纹病	1040	880	144	16
东格鲁病	1179	999	162	18

## 2 水稻病害模型设计

### 2.1 YOLOv8 基线模型结构

YOLOv8是由Ultralytic公司推出的单阶段目标检测器,能够执行目标检测、物体分割、图像分类、人体姿态识别以及obb旋转目标检测等任务.与YOLOv5



的网络结构相似, YOLOv8 通过缩放系数提供了 n、s、m、l、x 这 5 种模型版本, 以适应不同的应用场景需求. 本文选择了速度和精度较为均衡的 s 版本作为

研究基线, 其网络结构如图 2 所示. 基线模型的架构分为 4 个主要部分: 输入端、主干网络、颈部网络和检测输出端.

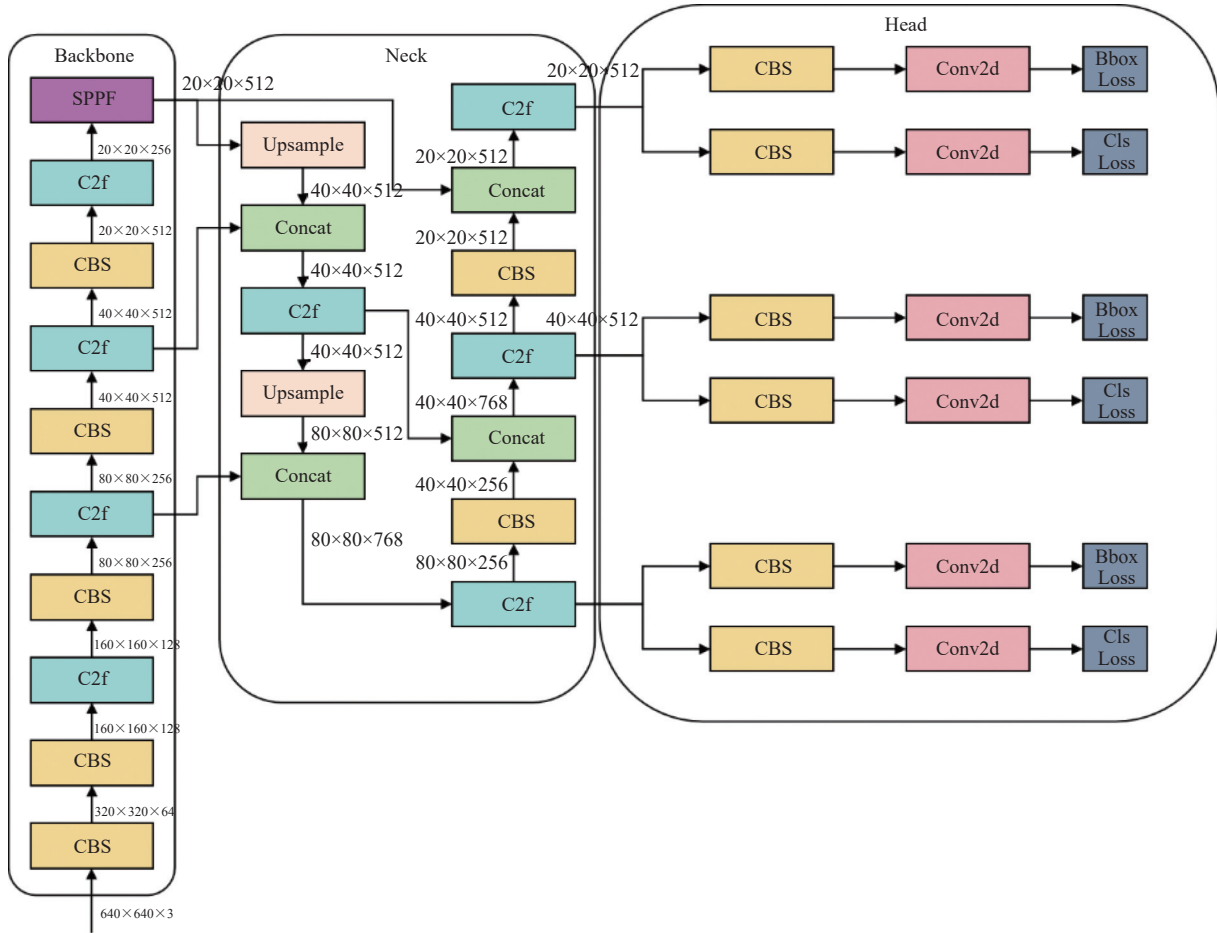


图 2 YOLOv8 网络结构图

输入端采用 640×640 尺寸的图像, 并通过 MOSAIC<sup>[20]</sup>、随机缩放、Mixup<sup>[21]</sup>、Albumentations、HSV 增强、水平翻转等技术进行在线数据增强. 这种方式能够在训练过程中动态生成新的样本, 无需离线数据增强, 节省了训练时间和资源. 同时, 这些增强技术还提高了模型在不同场景和目标下的适应能力, 增强了实际应用中的准确性和鲁棒性. 在训练的最后阶段, 借鉴 YOLOX<sup>[22]</sup> 的策略, 最后 10 个 epoch 关闭 MOSAIC 操作.

主干网络和颈部网络相比于 YOLOv5 进行了调整. 首先, 将第 1 个卷积模块的卷积核从 6×6 缩小为 3×3, 并将 C3 模块替换为 C2f 模块, 增加了跳跃连接和 split 操作. 此外, 主干网络中 C2f 模块的 Block 数从 3-6-9-3 调整为 3-6-6-3. 颈部网络采用了 PAN<sup>[23]</sup> 和 FPN<sup>[24]</sup> 的结构设计, 增强了多尺度特征融合能力, 提高

了特征提取的效果.

在检测输出端, YOLOv8 摒弃了耦合检测头, 采用解耦策略, 将输出分为分类和回归分支, 并从 YOLOv5 的基于 Anchor 方式转为 Anchor-Free. 取消了 objectness 分支, 仅保留了分类和回归分支, 其中回归分支使用了 Distribution Focal Loss 的积分形式表示法, 提升了收敛速度.

YOLOv8 的损失计算分为正负样本分配和 loss 计算两个部分. 正负样本分配采用动态策略 TaskAligned-Assigner<sup>[25]</sup>, 通过分类和回归分数加权来选择正样本. 总损失包括分类损失和回归损失, 其中分类损失使用 BCE 损失, 回归损失则由 CIoU<sup>[26]</sup> 和 DFL 组合. DFL 通过计算边界框和标签的分布概率损失, 优化边界框的每条边, 再通过 CIoU 损失对预测框整体进行优化.

综上所述, YOLOv8 在检测速度和精度上达到了较好的平衡, 且在多个领域展现出较强的泛化能力. 然而, 仍然存在一些需要改进的地方: (1) 由于 YOLOv8 采用了解耦检测头, 尽管提升了检测性能, 但相比耦合检测头, 模型的参数量增加了 55.5%, 浮点运算量增加了 73.3%. 这些增加使得模型变得更加庞大, 对资源有限的设备 (如嵌入式设备或移动端) 不太友好. 浮点运算量的增加也直接影响了计算复杂度, 推理时间变长, 进而降低了模型的实时性. 在计算资源有限或对低延迟有严格要求的场景中 (如自动驾驶), 模型可能无法满足实时检测需求; (2) 尽管 YOLOv8 已经取得了较好

的表现, 但其主干网络和颈部网络的特征提取仍有提升空间. 通过进一步引入注意力机制和改进卷积结构, 可以增强模型的特征提取能力, 同时实现更好的轻量化效果.

为了解决这些问题并提升 YOLOv8-s 算法的轻量化和特征提取能力, 本文对基线模型的主干网络、颈部网络及损失函数进行了改进, 提出了 FCU-YOLOv8 算法. 改进后的网络结构如图 3 所示, 改进后的模型在保持较高检测精度的同时, 减少了参数量和计算复杂度, 更适合资源受限的设备和实时性要求较高的应用场景.

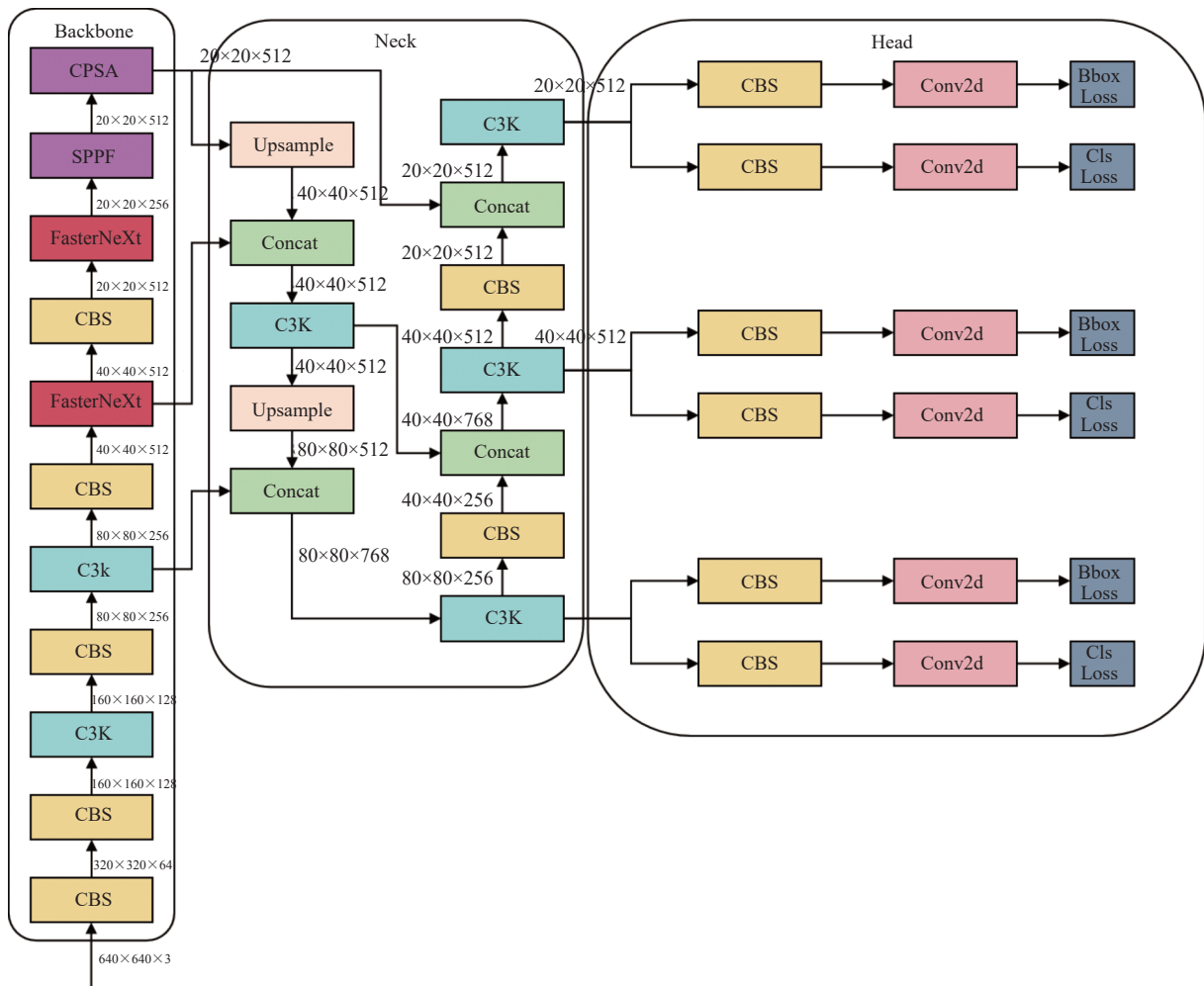


图 3 FCU-YOLOv8 网络结构图

## 2.2 FasterNeXt 模块

在水稻病害检测任务中存在检测背景复杂、病害表征多样的特点. YOLOv8 的主干网络中传统卷积模块存在计算量大、特征提取不完善的问题, 容易出现

目标漏检、误检问题. 为了更好地适配背景复杂性和目标多样性在主干网络中使用 FasterNeXt 替换指定层的 C2f 模块.

主干网络的 C2f 模块在特征提取过程中各个通道

的特征图都会进行卷积计算进行特征的提取,本文中  
使用 FasterNeXt 模块替换主干网络的 C2f 模块(第 6  
层和第 8 层)如图 4 所示,它是由 PConv<sup>[17]</sup>和 PWConv  
组成,由于不同的 channel 的特征图的特征存在高度相  
似性,为实现轻量化 PConv 卷积只在输入通道的一部  
分上进行常规的卷积计算实现空间的特征提取,而其

余通道数特征图保持不变.在 PConv 后连接两个逐点  
卷积 PWConv 充分有效地利用来自所有通道的信息.如  
图 4 右侧所示 FasterNeXt 的结构为 3 个卷积构成倒置  
残差块,其中中间层具有更多的通道数量,并且放置了  
shortcut 以重用输入特征.在两个 PWconv 模块之间加入  
BN 和 ReLU 激活函数保持网络的非线性表达的能力.

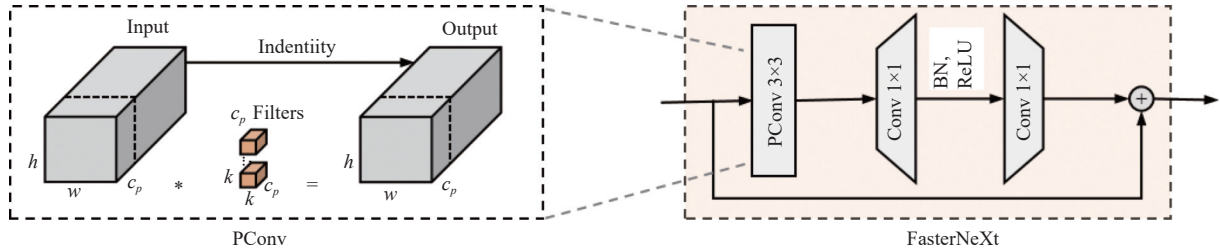


图 4 FasterNeXt 模块结构图

普通卷积模块的计算量 (FLOPs) 为:

$$h \times w \times k^2 \times c^2 \quad (1)$$

在 FasterNeXt 中部分卷积 PConv 的计算量 (FLOPs)  
为式 (2):

$$h \times w \times k^2 \times c_p^2 \quad (2)$$

其中,  $h, w$  是特征图的高和宽,  $k$  是卷积核的大小,  $c_p$  是  
部分卷积作用的通道数,  $c$  是普通卷积作用的通道数.  
在实际计算过程中  $\frac{c_p}{c} = \frac{1}{4}$ , 所以 PConv 计算量 FLOPs  
是普通卷积的 1/16.

在 PConv 计算过程中内存访问如下所示:

$$h \times w \times 2c_p + k^2 \times c_p^2 \approx h \times w \times 2c_p \quad (3)$$

其中,  $h, w$  是特征图的高和宽,  $k$  是卷积核的大小,  $c_p$  是  
部分卷积作用的通道数. PConv 的内存访问数量为常  
规卷积的 0.25 倍, 其余的  $c - c_p$  个通道不参与计算, 无  
需内存访问.

为了充分利用各个通道的特征信息, 在 PConv 后  
连接一个  $1 \times 1$  的卷积 PWConv 将输入特征图的有效感  
受野集中在特征图的中心位置区域, 作用等效于一个  
T-shaped Conv.

传统的 T-shaped Conv 的 FLOPs 如式 (4) 所示:

$$h \times w \times (k^2 \times c_p \times c + c \times (c - c_p)) \quad (4)$$

PConv+PWConv 组成的高效 T-shaped Conv 的  
FLOPs 如下:

$$h \times w \times (k^2 \times c_p + c \times c_p) \quad (5)$$

其中,  $c > c_p, c - c_p > c_p$  所以 PConv+PWConv 组合的卷  
积模块更加高效.

### 2.3 C3K 模块和 CPSA 模块

在基线模型中 C2f 的结构是主干网络和颈部网络  
的最重要的计算单元, 该结构是 CSP 的衍生结构, 使用  
了跨层特征连接. 它在特征提取过程中使用较多的  
Bottleneck 模块, 每个 Bottleneck 模块内部包含残差连  
接和卷积操作, 通过引入跨通道的连接, 将输入特征图  
拆分为两部分, 一部分直接传递, 另一部分进行特征提  
取, 最后再将两者结合. 这种结构确保了在不同特征尺  
度上能提取到丰富的局部和全局特征. 这种计算方式  
在常规的目标检测场景中泛化能力很强. 在水稻病害  
检测任务中 C2f 固定感受野可能难以适配复杂的多尺  
度场景.

本文提出了一种新型特征提取模块 C3K, 其结构  
如图 5 所示. C3K 模块基于 CSP (cross stage partial) 设  
计, 通过灵活的特性增强了网络的适应性和性能. 其核  
心创新在于支持用户根据任务需求选择是否启用 C3n  
模块 (自定义卷积核大小的 Bottleneck). 当 C3n 参数  
为 false 时, C3K 等效于 C2f 模块, Bottleneck 采用传统  
形式; 当 C3n 参数为 true 时, Bottleneck 被替换为 C3  
模块, 从而提供更大的卷积核以提升特征提取能力.

相较于传统的 C3 模块, C3K 在特征提取能力、计  
算效率、扩展性和训练特性等方面均具备显著优势.  
首先, C3K 通过引入  $1 \times 1$  卷积实现特征图的分割与融  
合, 使得多尺度特征的提取更加精准和高效. 这种分割

策略能够动态调整不同区域的特征表示能力,尤其在复杂任务中显著提升对多尺度目标的适应性。相比之下, C3 模块虽然也使用  $1 \times 1$  卷积进行通道调整,但缺乏 C3K 的精细分割与融合机制,可能导致特征冗余或信息丢失。其次, C3K 优化了特征融合流程,通过  $1 \times 1$  卷积的分离与合并操作,显著降低了不必要的计算开销。在深层网络中, C3K 的模块化设计能够以较低的计算成本实现更强的特征表示能力,而 C3 模块的多次卷积操作则会增加计算负担。

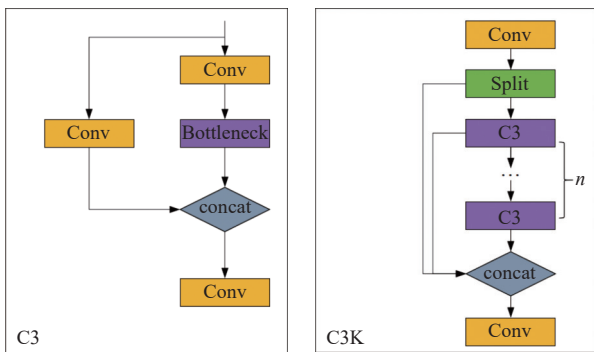


图5 C3K 模块结构图

此外, C3K 模块允许根据任务需求调整卷积核大小(通过选择 C3n 模块)。较大的卷积核能够有效扩大感受野,提升复杂场景中特征提取的深度和质量,从而在处理复杂任务时表现出更优的性能,而 C3 模块的固定结构限制了其适应性。最后, C3K 模块集成了残差连接设计,这对于深层网络的训练尤为重要。残差连接能够缓解梯度消失问题,提升训练的稳定性 and 收敛速度。在多层特征提取后, C3K 通过第 2 次  $1 \times 1$  卷积恢复通道数,有效避免了信息丢失,使模型在更深层次上仍能保持高效的特征表达能力。相比之下, C3 模块在信息恢复与融合上的处理相对粗糙,难以充分保留关键特征信息。

综上所述, C3K 模块通过精细化的特征分割与融合策略、灵活的卷积核配置以及高效的残差连接设计,在特征提取效率、计算成本和任务适应性方面显著优于 C3 模块。特别是在处理多尺度目标或复杂任务时, C3K 模块的性能提升尤为明显,展现了广泛的应用潜力。

在主干网络中的 SPPF 后增加一个 CPSA 模块结构图如图 6 所示,该模块是对 C2f 模块的扩展,结合 PSA (pointwise spatial attention)<sup>[27]</sup>用于增强特征提取和注意力机制。通过在标准 C2f 模块中引入 PSA 块, CPSA 实现了更强大的注意力机制,从而提高了模型对重要特征的捕捉能力。CPSA 扩展了 C2f,通过引入 PSA,

通过多头注意力机制和前馈神经网络来增强特征提取能力。它可以选择性地添加残差结构 (shortcut) 以优化梯度传播和网络训练效果。同时,使用 FFN 可以将输入特征映射到更高维的空间,捕获输入特征的复杂非线性关系,允许模型学习更丰富的特征表示。

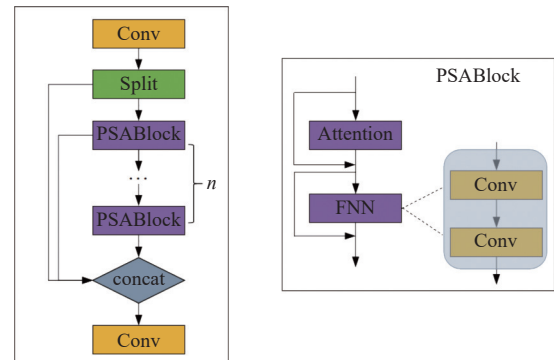


图6 CPSA 模块结构图

本文在主干网络中引入了 CPSA (channel and position attention) 模块,将卷积操作与注意力机制 PSA 融合,通过优化特征表示能力,增强网络的感知能力与空间注意力机制,从而显著提升检测性能。CPSA 在 PSA 的基础上引入了双重特征选择机制,结合了通道注意力与空间注意力,使网络能够对每个通道和每个位置进行联合加权。此机制进一步增强了特征表达能力,使网络在处理复杂场景时能够更加精确地聚焦于重要区域和特征通道,从而提高模型的精度和鲁棒性。

PSA 机制主要关注通道维度,通过对不同尺度的特征图进行下采样并加权,提升了对不同尺度信息的感知能力。然而, CPSA 模块不仅在通道维度进行加权,还在空间维度上对每个像素位置加权,使得网络能够更加细致地捕捉多尺度特征,强化对目标位置与形状变化的敏感性。在多尺度特征融合方面, CPSA 通过联合加权 (squeeze-and-excitation, SE) 优化了尺度信息的融合效果,不同于 PSA 仅对各尺度特征进行独立加权,未充分考虑尺度之间的空间位置关系。CPSA 能够更精细地融合不同尺度的特征,捕获上下文信息,从而显著提升对目标大小和位置变化的适应能力。

此外, CPSA 引入了自适应加权机制,根据图像的特征分布灵活调整权重。空间注意力有效减少了背景信息对目标检测的干扰,而通道注意力则帮助网络更好地聚焦于对任务有贡献的特征通道。这种自适应机制提升了模型在多样化场景(如小物体、大物体、高对比度和低对比度图像等)中的泛化能力,使其在应对



复杂背景、多目标场景时表现尤为出色。

综上所述, CPSA 模块通过双重特征选择机制显著提升了模型的多尺度信息融合能力、空间感知能力和通道表达能力. 与 PSA 相比, CPSA 不仅更精准地捕获多尺度信息, 还能够灵活分配注意力于关键特征区域, 在处理小物体、复杂背景和多目标任务时展现出更强的性能优势. 这些改进使得 CPSA 在目标检测任务中表现优于 PSA, 成为提升网络整体检测性能的重要组件.

## 2.4 损失函数优化

YOLOv8 采用 CIoU 组合 DFL 作为边界框回归损失函数, 然而水稻病害检测任务中, 由于拍摄角度和距离、叶片遮挡、病害目标密集等因素, 其中不可避免的包含一些低质量样本, 基线模型中使用 CIoU, 其中的几何因素会增加对这些低质量样本的惩罚导致模型泛化能力降低. 因此选择 UIoU 作为回归损失, 它更关注不同质量预测框之间的权重分配, 具体来说, 该损失函数以一种新颖的方式将模型的注意力从低质量的预测框动态地转移到高质量的预测框上, 以提升模型在高精度或密集数据集上的检测性能, 并在训练速度上达到平衡.

UIoU<sup>[18]</sup>是一种基于 IoU 的损失函数, 它在常规的 IoU 基础上, 集成了不同 IoU 变种的优势, 从而实现更强的几何约束和更精确的边界框回归. UIoU 的核心思想是, 通过动态调整边界框的尺寸, 并结合 IoU、中心点距离、宽高比、形状和角度等多个因素, 提升检测框和真实框的匹配效果.

UIoU 通过引入多个关键机制, 增强了边界框预测的精度, 尤其在长宽比、中心点位置等方面进行更为细致的约束. 它的工作原理可以分解为以下几个部分.

首先是动态缩放, 训练的早期阶段, 模型可能需要更多的边界框调整, 而在后期则需要更加精细的微调. UIoU 引入一个动态调整的缩放系数 *ratio*, 根据训练的周期 (epoch) 来动态缩放预测边界框的尺寸. 随着训练的推进, *ratio* 会从 2 缩小到 0.5, 使得早期更加关注粗略的边界框调整, 后期更关注细致的调整. 边界框的动态缩放的公式如下:

$$w' = w \times ratio \quad (6)$$

$$h' = h \times ratio \quad (7)$$

其中,  $w, h$  是原始宽高, *ratio* 随着训练周期进行变化.

然后是 IoU 权重和置信度结合, 为了进一步增强对框定位和分类置信度的平衡, UIoU 将 IoU 与置信度信息结合, 用于调整 *Focal Loss*.

$$FocalIoU = IoU^\alpha \times confidence^\gamma \quad (8)$$

*Focal Loss*<sup>[28]</sup>的置信度通过调整 IoU 和置信度的权重, 提升对困难样本的关注. 其中,  $\alpha$  控制 IoU 的权重,  $\gamma$  控制置信度的权重.

总体来说, UIoU 提供了更全面、更灵活的边界框回归损失函数, 在保留中心点距离、形状一致性、角度等几何因素的同时增加动态缩放和 IoU 权重和置信度结合, 提高了检测模型的预测精度, 特别适用于边界模糊或目标较小精度要求较高的场景.

综上所述, 本文在 YOLOv8 的基础上, 提出了针对水稻病害检测任务的多项改进. 首先, 采用了 FasterNeXt 模块替换主干网络中的 C2f 模块, 显著减少了计算量和内存访问, 提升了特征提取效率. 其次, 提出了 C3K 和 CPSA 模块, 增强了特征提取的灵活性和对复杂场景的适应能力, 并结合注意力机制进一步提高了检测性能. 最后, 在损失函数方面, 引入了 UIoU, 优化了边界框回归过程, 使模型更具泛化能力. 总体而言, 这些创新点有效提升了 YOLOv8 在水稻病害检测任务中的轻量化水平和检测精度.

## 3 实验结果与分析

### 3.1 YOLOv8 实验平台与参数设置

本实验硬件配置使用 Intel Core i7-14700KF CPU@5.60 GHz 处理器, 16 核 24 线程, NVIDIA GTX 4080 super 显卡、32 GB RAM、2TB 机械硬盘, 操作系统是 Windows 11 64 位系统. 编程语言为 Python, 深度学习框架为 PyTorch, GPU 加速库为 CUDA 11.8 和 CUDNN 8.5. 模型输入图像大小为 640×640, 优化器选用 SGD, 动量设置为 0.937, 初始学习率为 0.01, 训练周期为 200 轮, 批量大小设为 64, 其他训练参数采用官方推荐参数.

### 3.2 评价指标

为了全面评估模型在水稻病害检测任务中的性能, 本文采用精确率 precision、召回率 recall、F1 值、平均精度 (mean average precision, *mAP*)、参数量、浮点计算量作为评价指标. *mAP*@0.5 表示 IoU 设为 0.5 时的平均精确度, *mAP*@0.5:0.95 表示 IoU 从 0.5 到 0.95,



步长为 0.05 时的平均精确度, 参数量表示模型体积, 浮点计算量表示算法复杂度. 其他性能指标的具体含义如下.

精确率 (precision,  $P$ ): 用于评估分类算法在预测为正类的样本中, 实际为正类的比例. 它通过将正确预测为正类的样本数量除以所有预测为正类的样本数量来计算. 公式如式 (9) 所示:

$$P = \frac{TP}{TP+FP} \quad (9)$$

召回率 (recall,  $R$ ): 用于衡量分类算法在实际为正类的样本中, 成功预测为正类的比例. 具体来说, 它通过将正确预测为正类的样本数量除以所有实际为正类的样本数量来计算. 公式如式 (10) 所示:

$$R = \frac{TP}{TP+FN} \quad (10)$$

$F1$  值: 是同时结合精确率和召回率的综合性指标, 通过计算它们的调和平均值来体现二者的平衡. 具体计算公式如式 (11) 所示.

$$F1 = \frac{2 \times P \times R}{P + R} \quad (11)$$

平均精度 ( $mAP$ ): 这是用于评估目标检测任务的指标, 综合了各类目标的精确率和召回率.  $mAP$  通过计算每个类别的精确率-召回率曲线下的面积来获得每个类别的平均精度, 最后对所有类别的平均精度取均值. 公式如式 (12) 所示:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (12)$$

### 3.3 实验结果对比和分析

为了验证本文设计的水稻病害识别网络 FCU-YOLOv8 中 FasterNeXt、C3K、CPSA 模块以及损失函数 UIoU 的有效性, 以 YOLOv8s 模型为基线在水稻病害数据集上进行消融实验, 为了加快模型的收敛训练时均加载预训练模型, 使用相同的实验环境和参数设置, 模型的输入为  $640 \times 640$ , 迭代的次数为 200. 实验结果如表 2 所示.

表 2 消融实验

模型	$P$	$R$	$mAP@0.5$	$mAP@0.5:0.95$	Params (MB)	计算量 (GFLOPs)
YOLOv8	0.906	0.877	0.902	0.639	11.2	28.6
YOLOv8+CPSA	0.908	0.864	0.911	0.63	13.4	36.2
YOLOv8+C3K+CPSA	0.923	0.866	0.923	0.626	9.4	21.3
YOLOv8+FasterNeXt	0.905	0.861	0.927	0.645	8.5	20.5
YOLOv8+UIoU	0.911	0.893	0.914	0.642	11.2	28.6
FCU-YOLOv8 (本文)	0.905	0.892	0.926	0.672	8.5	20.4

由表 2 中的消融实验结果可知, 模型 2 是在主干网络中最后一层增加 CPSA 引入多头注意力机制, 提高模型的表达能力, 与此同时会带来计算参数和浮点运算数量的增大, 因此需要搭配其他轻量化模块使用. 模型 3 是在对主干网络和颈部网络中 C2f 模块替换为 C3K 模块, 在主干网络 SPPF 模块后增加 CPSA 模块. 相比较与基线模型在平均精度均值  $mAP@0.5$  上提升了 2.1%, 准确率  $P$  提升了 1.7%, 与此同时模型的参数量 Params 降低了 16%, 计算浮点数 FLOPs 降低了 25.5%. 模型 4 是在主干网络中使用 FasterNeXt 模块替换部分 C2f 模块, 以求提升特征提取能力的同时优化卷积计算实现主干网络的轻量化, 相比较与基线模型在平均精度均值  $mAP@0.5$  上提升了 2.5%, 在  $mAP@0.5:0.95$  上提升了 0.6%, 与此同时模型的参数量 Params 降低了 24.2%, 计算浮点数 FLOPs 降低了 28.3%. 模型 5 中使用 UIoU 替换了基线模型的 Clou 损失函数, 对低质量的目标进行权重重新分配. 由实验结果可知,

模型 5 与基线模型相比准确率  $P$  提升 0.5%, 召回率  $R$  提升 1.6%,  $mAP@0.5$  提升 1.2%. 模型 6 是本文提出的 FCU-YOLOv8 模型, 同时融合 C3K、CPSA、FasterNeXt 和 UIoU 的改进, 与基线模型对比  $mAP@0.5$  提升了 2.4%,  $mAP@0.5:0.95$  提升了 3.3%, 在轻量化方面, 模型参数降低了 24.2%, 计算浮点数下降 28.7%, 综合对比在精度和速度上 FCU-YOLOv8 是最佳的模型.

表 3 为 SOTA 模型对比, 通过对比不同算法的实验结果, 本文提出的 FCU-YOLOv8 在多个关键性能指标上表现出色. FCU-YOLOv8 的  $mAP@0.5$  为 0.926,  $mAP@0.5:0.95$  为 0.672, 均超过了大多数对比算法, 其中  $mAP@0.5:0.95$  比 YOLOv8 高出 3.3 个百分点, 比 YOLOv5 高出 11 个百分点. 此外, FCU-YOLOv8 在保持较低的参数量 (8.5 MB) 和计算量 (20.4 GFLOPs) 的同时, 依然能达到 0.90 的  $F1$  值, 与 YOLOv9-c 持平, 但 FCU-YOLOv8 的模型更轻量化, 参数量和计算量分别比 YOLOv9-c 减少了 2.5 倍和 4 倍. 相比之下,

YOLOv3 虽然在精度 ( $P=0.919$ ) 上稍占优势, 但其  $mAP@0.5:0.95$  仅为 0.597,  $F1$  值也较低 (0.86), 且参数量 (98.4 MB) 和计算量 (261.8 GFLOPs) 极大, 严重影响实际应用中的推理效率. YOLOv5 尽管  $F1$  值 (0.90)

较高, 但其  $mAP@0.5:0.95$  为 0.562, 在精度上明显不如 FCU-YOLOv8. YOLOv6 的性能相对平衡, 但其  $mAP@0.5:0.95$  仍低于 FCU-YOLOv8, 且计算量更大 (42.8 GFLOPs).

表3 SOTA 模型对比实验

方法	$P$	$R$	$mAP@0.5$	$mAP@0.5:0.95$	Params (MB)	计算量 (GFLOPs)	$F1$
YOLOv8	0.906	0.877	0.902	0.639	11.2	28.6	0.89
YOLOv3*	0.919	0.832	0.89	0.597	98.4	261.8	0.86
YOLOv5	0.861	0.88	0.883	0.562	7.2	16.4	0.90
YOLOv6 <sup>[29]</sup>	0.876	0.866	0.915	0.612	15.9	42.8	0.87
YOLOv9-c <sup>[30]</sup>	0.915	0.891	0.93	0.657	21.1	82.7	0.90
YOLOv10 <sup>[31]</sup>	0.856	0.822	0.893	0.615	8.1	24.5	0.84
FCU-YOLOv8 (本文方法)	0.905	0.892	0.926	0.672	8.5	20.4	0.90

注: \*表示该算法由于模型参数量限制无法与其他算法保持在batch为128训练, 采用的batch参数为32

综上所述, FCU-YOLOv8 通过网络结构的改进, 展现了优异的检测性能和实时性, 特别是在资源有限的场景中, FCU-YOLOv8 凭借较低的参数量和计算量实现了高精度与高效性, 具备更强的实际应用价值.

为了充分展现本文所提算法在轻量化方面的显著成效, 本文将其与当前目标检测领域的最新 SOTA 算法进行了对比, 这些算法涵盖了 DEIM、D-FINE、YOLOv11 等前沿技术. 从表 4 中可以看出, YOLOv11、D-FINE 以及 DEIM 的同级别 s 模型在参数量 (Params) 方面, 分别为 9.4 MB、10.0 MB、10.0 MB; 在浮点运算次数 (GFLOPs) 方面, 分别为 21.7、25.0、25.0. 而本文所提出的 FCU-YOLOv8 算法, 在参数量和浮点运算次数这两个关键指标上, 均达到了最低水平, 分别为 8.5 MB 和 20.4 GFLOPs. 具体来看, FCU-YOLOv8 算法在参数量方面, 相比 YOLOv11 降低了约 9.57%, 相比 D-FINE 和 DEIM 降低了 15%; 在浮点运算次数方面, 相比 YOLOv11 降低了约 6.0%, 相比 D-FINE 和 DEIM 降低了 18.4%. 这些数据充分证明 FCU-YOLOv8 算法在实现网络轻量化方面的显著优势, 使其在资源受限的场景下更具应用优势, 能够以更少的参数和更低的运算量达到优异的检测效果, 为实际部署提供了更高效、更经济的解决方案.

为了更清晰地对比不同模型在训练过程中的表现, 本文展示了各个模型的精度曲线图如图 7 所示和损失曲线图如图 8 所示. 这些图表能够更直观地理解模型的收敛情况、训练的稳定性以及性能提升的速度. 然而, 需要特别注意的是, 损失曲线存在部分缺失的情况. 这是由于模型间的差异导致的: 具体而言, YOLOv3 和

YOLOv5 模型中不包含 DFL (distribution focal loss) 损失, 而其他模型则不包含 obj (objectness) 损失. 这种差异在一定程度上影响了损失曲线的完整性, 但对于整体趋势的理解依然具有重要参考价值. 尽管如此, 精度曲线仍然能够全面展示各模型的表现, 使得模型间的对比分析更加有效.

表4 轻量化模型与 SOTA 模型参数对比

方法	Params (MB)	计算量 (GFLOPs)
YOLO11s <sup>[32]</sup>	9.4	21.7
D-FINE-S <sup>[33]</sup>	10.0	25.0
DEIM-D-FINE <sup>[34]</sup>	10.0	25.0
FCU-YOLOv8 (本文方法)	8.5	20.4

从损失曲线来看, 不同模型在训练和验证过程中的损失 (box\_loss, dfl\_loss, cls\_loss) 均呈现出随训练轮数增加而下降的趋势. YOLOv3 的损失值在训练初期最高, 表明其初始定位误差较大, 但随着训练的进行损失逐渐下降. FCU-YOLOv8 (本文方法) 和 YOLOv9-c 均表现出较低的 box\_loss, 尤其是 FCU-YOLOv8, 在训练和验证集上的 box\_loss 都低于其他模型, 表明其定位精度较高. FCU-YOLOv8 和 YOLOv9-c 的 dfl\_loss (距离方差损失) 较低, 进一步验证了这两种模型在预测框架方面表现更为精准. YOLOv3 和 YOLOv10 的 dfl\_loss 显著高于其他模型, 尤其在训练初期收敛较慢. FCU-YOLOv8 和 YOLOv9-c 的分类损失较低, 表明这两种方法在目标分类上也具有较高的准确性, 验证集上的 cls\_loss 也相对稳定. YOLOv10 的 cls\_loss 下降速度慢, 且在后期仍未完全收敛, 说明其在目标分类方面存在一定的问题.

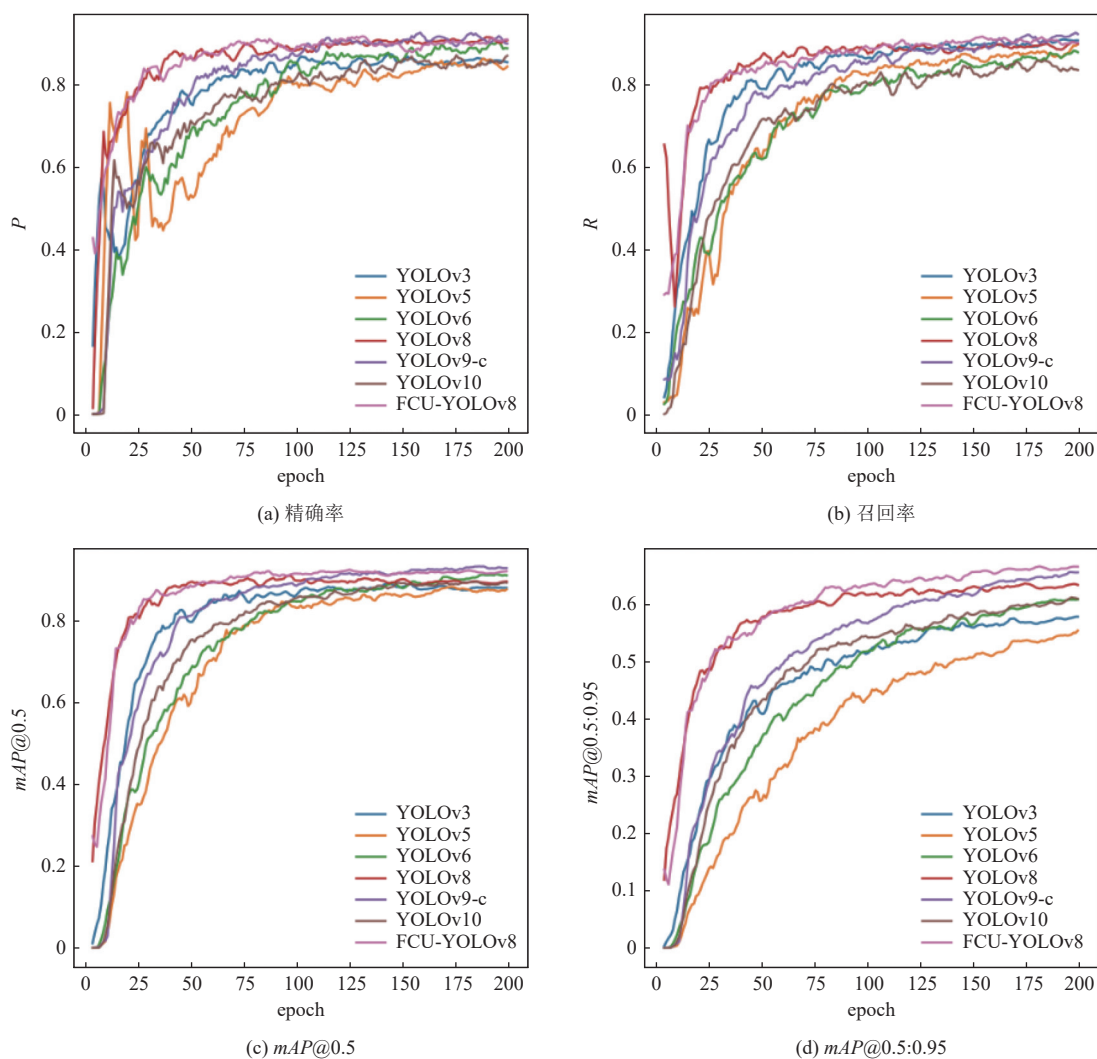


图7 SOTA模型对比实验精度曲线图

在精度 (precision)、召回率 (recall) 以及平均精度 ( $mAP@0.5$  和  $mAP@0.5:0.95$ ) 的表现上, 各模型呈现明显的差异: FCU-YOLOv8 (本文方法) 和 YOLOv9-c 的精度在训练过程中迅速提升, 并稳定在较高的水平, 表明这两种模型在检测到的目标中, 正确预测的比例较高. YOLOv10 的精度曲线波动较大, 收敛性差, 表明其在精度上存在不足. FCU-YOLOv8 的召回率表现尤为突出, 始终高于其他模型, 说明其能够捕捉更多的目标. YOLOv6 和 YOLOv9-c 的召回率也表现较好, 但相较于 FCU-YOLOv8 稍逊一筹. FCU-YOLOv8 在  $mAP@0.5$  和  $mAP@0.5:0.95$  上均取得了最佳结果, 特别是在  $mAP@0.5:0.95$  上超越其他模型, 表明其在多个  $IoU$  阈值下具有更好的综合表现.

从以上损失和精度曲线的分析可以看出, 本文提出的 FCU-YOLOv8 在目标检测任务中具有显著的性

能优势. 其在精度、召回率和  $mAP$  指标上均优于其他模型, 尤其在  $mAP@0.5:0.95$  上表现最为突出. 同时, FCU-YOLOv8 的损失曲线表明该模型不仅在训练过程中收敛迅速, 而且在定位和分类任务上的误差也较小, 验证了其在目标检测任务中的卓越表现.

### 3.4 可视化检测结果

将改进后的算法应用于模型训练后, 获得了最佳模型, 并使用该模型对图像进行预测. 图9展示了模型改进前后预测结果. 从图中可以看出, 改进算法在轻量化的前提下依然具备出色的检测性能. 各类目标, 如密集点病害或大幅叶片病害, 准确定位和标注, 优于改进前, 表明 FCU-YOLOv8 算法的精度和鲁棒性优异. 此外, FCU-YOLOv8 算法模型在不同场景下均表现出较强的泛化能力, 能够适应复杂的背景与多样化的样本输入, 展示了其在实际应用中的潜力.



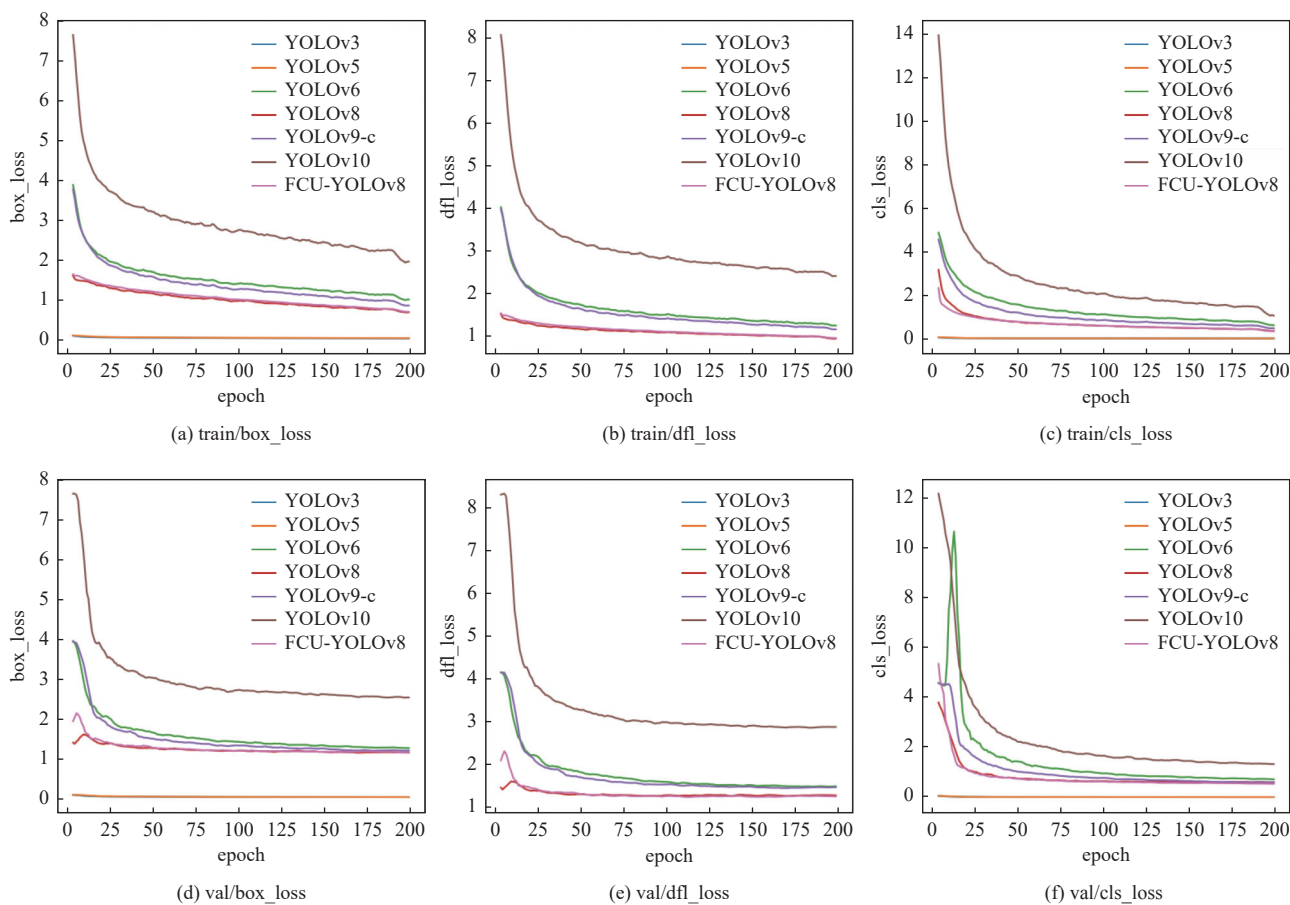


图8 SOTA模型对比实验精度损失曲线图

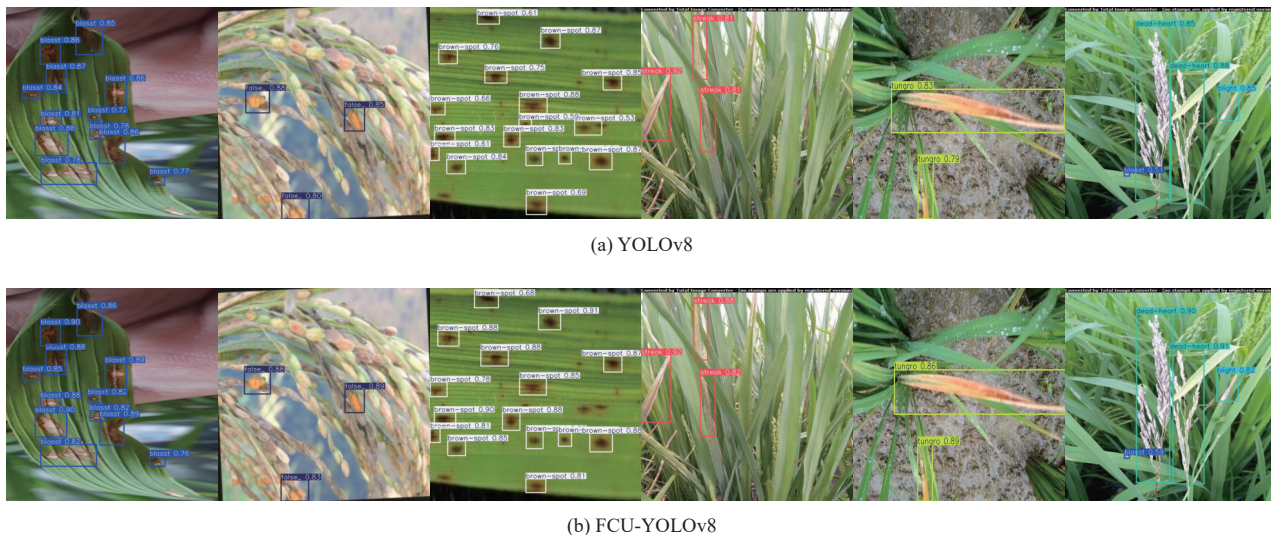


图9 FCU-YOLOv8与YOLOv8检测效果对比

这种性能的提升不仅在检测精度上有所体现,同时也显著减少了计算资源的消耗,为轻量级目标检测任务提供了高效解决方案。

为了深入分析改进后的FCU-YOLOv8网络在目标检测性能方面的提升,本研究对比了改进前后多层特征图的输出,并进行了可视化分析如图10所示。

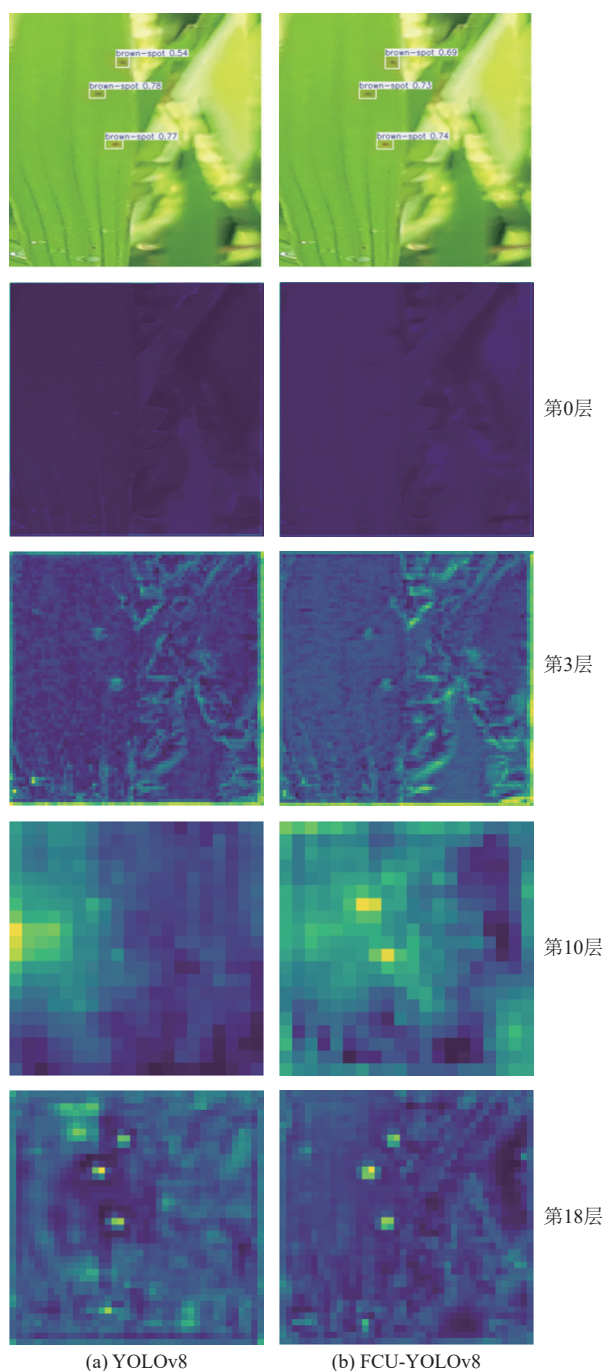


图10 FCU-YOLOv8(右侧)与YOLOv8特征图可视化对比

改进后的网络在特征图表现上展现了显著优势,具体表现在:首先,改进后的网络特征图中,热点区域的分布更加集中(主干网络的第10、18层所示),表明网络在定位目标区域方面更加精确,热点集中度的增强有效减少了误检与漏检现象,从而提升了目标检测的准确性;其次,特征图中热点区域与非热点区域之间的对比度显著提高(第18层),表明网络在特征区分能力上有所提升,增强的特征区分度使得网络能够更有

效地识别和区分不同目标,进而提升了目标检测的鲁棒性。另外,改进后的网络在背景区域的热点数量较少,说明其在抑制背景干扰方面表现更佳,减少了复杂背景下的误检现象,从而提高了检测精度。同时,改进后的特征图呈现了更多细节信息,表明网络能够捕捉到更多关键特征,丰富的细节信息增强了网络对图像内容的理解能力,进一步提升了检测的准确性和鲁棒性;最后,改进后的网络在不同层之间展示了更高的一致性,表明网络在特征传递过程中保持了更稳定的特征信息,特征一致性的提高有助于各层之间更有效地传递稳定的特征信息,从而提升了整体检测性能的稳定性的。

#### 4 结语

在本研究中,针对水稻叶片病害检测任务提出的改进YOLOv8模型(FCU-YOLOv8)展示了优异的性能。通过引入FasterNeXt、C3K和CPSA模块,结合UIoU损失函数的优化,模型在精度、召回率以及推理速度等关键指标上均得到了显著提升。实验结果表明,FCU-YOLOv8能够更有效地检测复杂场景下的水稻病害,尤其是在背景干扰强、病害特征差异小的情况下,展现出更强的鲁棒性和适应性。

这一改进不仅为农业病害检测提供了一种高效的解决方案,也为未来的相关领域研究提供了重要参考。未来的工作可以进一步探索如何优化模型的轻量化设计,使其能够在低功耗设备上高效运行,促进在实际生产中的广泛应用。同时,结合多源数据的融合分析或迁移学习的策略,也有望进一步提升病害检测的广泛适用性,为现代智能农业提供技术支持。

#### 参考文献

- Lu Y, Yi S J, Zeng NY, *et al.* Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, 2017, 267: 378–384. [doi: 10.1016/j.neucom.2017.06.023]
- 朱文杰, 孟鑫, 李根, 等. 水稻病虫害目标检测技术研究进展. *农业工程*, 2024, 14(6): 39–46.
- Annabel LSP, Annapoorani T, Deepalakshmi P. Machine learning for plant leaf disease detection and classification—A review. *Proceedings of the 2019 International conference on communication and signal processing (ICCSP)*. Chennai: IEEE, 2019. 538–542.
- Phadikar S, Sil J, Das AK. Classification of rice leaf diseases based on morphological changes. *International Journal of Information and Electronics Engineering*, 2012, 2(3): 460–463.
- Xu YZ, Yu GZ, Wang YP, *et al.* Car detection from low-

- altitude UAV imagery with the faster R-CNN. *Journal of Advanced Transportation*, 2017, 2017(1): 2823617.
- 6 周飞燕, 金林鹏, 董军. 卷积神经网络研究综述. *计算机学报*, 2017, 40(6): 1229–1251. [doi: [10.11897/SP.J.1016.2017.01229](https://doi.org/10.11897/SP.J.1016.2017.01229)]
  - 7 Carion N, Massa F, Synnaeve G, *et al.* End-to-end object detection with Transformers. *Proceedings of the 16th European Conference on Computer Vision*. Cham: Springer, 2020. 213–229.
  - 8 Zhou GX, Zhang WZ, Chen AB, *et al.* Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion. *IEEE Access*, 2019, 7: 143190–143206.
  - 9 Bari BS, Islam MN, Rashid M, *et al.* A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *PeerJ Computer Science*, 2021, 7: e432. [doi: [10.7717/peerj-cs.432](https://doi.org/10.7717/peerj-cs.432)]
  - 10 Zhang Z, Jiang L, Hong QQ, *et al.* Rice disease identification method based on improved mask R-CNN. *Proceedings of the 2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI)*. Kunming: IEEE, 2021. 250–253.
  - 11 Yang H, Deng XQ, Shen H, *et al.* Disease detection and identification of rice leaf based on improved detection Transformer. *Agriculture*, 2023, 13(7): 1361. [doi: [10.3390/agriculture13071361](https://doi.org/10.3390/agriculture13071361)]
  - 12 严陈慧子, 田芳明, 谭峰, 等. 基于改进 YOLOv4 的水稻病害快速检测方法. *江苏农业科学*, 2023, 51(6): 187–194.
  - 13 周思捷, 刘天奇, 陈天华. 基于改进 YOLOv5 算法的水稻病害识别研究. *中国农机化学报*, 2024, 45(8): 246–253.
  - 14 Yuan CC, Liu TH, Gao FY, *et al.* YOLOv5s-CBAM-DMLHead: A lightweight identification algorithm for weedy rice (*Oryza sativa f.spontanea*) based on improved YOLOv5. *Crop Protection*, 2023, 172: 106342.
  - 15 Jia LQ, Wang T, Chen Y, *et al.* MobileNet-CA-YOLO: An improved YOLOv7 based on the MobileNetV3 and attention mechanism for rice pests and diseases detection. *Agriculture*, 2023, 13(7): 1285. [doi: [10.3390/agriculture13071285](https://doi.org/10.3390/agriculture13071285)]
  - 16 戴林华, 黎远松, 石睿. 基于改进 YOLOv8n 算法的水稻叶片病害检测. *湖北民族大学学报(自然科学版)*, 2024, 42(3): 382–388. [doi: [10.13501/j.cnki.42-1908/n.2024.09.008](https://doi.org/10.13501/j.cnki.42-1908/n.2024.09.008)]
  - 17 Chen JR, Kao SH, He H, *et al.* Run, don't walk: Chasing higher FLOPs for faster neural networks. *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vancouver: IEEE, 2023. 12021–12031. [doi: [10.1109/CVPR52729.2023.01157](https://doi.org/10.1109/CVPR52729.2023.01157)]
  - 18 Luo XJ, Cai ZH, Shao B, *et al.* Unified-IoU: For high-quality object detection. *arXiv:2408.06636*, 2024.
  - 19 Sethy PK, Barpanda NK, Rath AK, *et al.* Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture*, 2020, 175: 105527. [doi: [10.1016/j.compag.2020.105527](https://doi.org/10.1016/j.compag.2020.105527)]
  - 20 Wang W, Howard A. MOSAIC: Mobile segmentation via decoding aggregated information and encoded context. *arXiv:2112.11623*, 2021.
  - 21 Zhang HY, Cisse M, Dauphin YN, *et al.* Mixup: Beyond empirical risk minimization. *Proceedings of the 6th International Conference on Learning Representations*. Vancouver: OpenReview.net, 2018. 1–13.
  - 22 Ge Z, Liu ST, Wang F, *et al.* YOLOX: Exceeding YOLO series in 2021. *arXiv:2107.08430*, 2021.
  - 23 Lin TY, Dollár P, Girshick R, *et al.* Feature pyramid networks for object detection. *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu: IEEE, 2017. 936–944. [doi: [10.1109/CVPR.2017.106](https://doi.org/10.1109/CVPR.2017.106)]
  - 24 Liu S, Qi L, Qin HF, *et al.* Path aggregation network for instance segmentation. *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018. 8759–8768. [doi: [10.1109/CVPR.2018.00913](https://doi.org/10.1109/CVPR.2018.00913)]
  - 25 Feng CJ, Zhong YJ, Gao Y, *et al.* TOOD: Task-aligned one-stage object detection. *arXiv:2108.07755*, 2021.
  - 26 Zheng ZH, Wang P, Ren DW, *et al.* Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybernetics*, 2022, 52(8): 8574–8586. [doi: [10.1109/TCYB.2021.3095305](https://doi.org/10.1109/TCYB.2021.3095305)]
  - 27 Liu HJ, Liu FQ, Fan XY, *et al.* Polarized self-attention: Towards high-quality pixel-wise regression. *arXiv:2107.00782*, 2021.
  - 28 Lin TY, Goyal R, Girshick R, *et al.* Focal loss for dense object detection. *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, 2017. 2999–3007.
  - 29 Li CY, Li LL, Jiang HL, *et al.* YOLOv6: A single-stage object detection framework for industrial applications. *arXiv:2209.02976*, 2022.
  - 30 Wang CY, Yeh IH, Liao HYM. YOLOv9: Learning what you want to learn using programmable gradient information. *Proceedings of the 18th European Conference on Computer Vision*. Milan: Springer, 2024. 1–21.
  - 31 Wang A, Chen H, Liu LH, *et al.* YOLOv10: Real-time end-to-end object detection. *Proceedings of the 38th Conference on Neural Information Processing Systems*. Vancouver: NeurIPS, 2024. 1–28.
  - 32 Jocher G, Qiu J, Chaurasia A. Ultralytics YOLO (Version 8.0.0) [Computer software]. <https://github.com/ultralytics/ultralytics>. [2024-10-12].
  - 33 Peng Y, Li H, Wu P, *et al.* D-FINE: Redefine regression task of DETRs as fine-grained distribution refinement. *arXiv:2410.13842*, 2024.
  - 34 Huang SH, Lu ZC, Cun XD, *et al.* DEIM: DETR with improved matching for fast convergence. *arXiv:2412.04234*, 2024.

(校对责编: 张重毅)