

桂余鹏,胡蓉华,崔艳荣,等. 基于轻量化 YOLO v8 - Rice 的水稻虫害检测方法[J]. 江苏农业科学,2024,52(20):277-284.

doi:10.15889/j.issn.1002-1302.2024.20.033

基于轻量化 YOLO v8 - Rice 的水稻虫害检测方法

桂余鹏, 胡蓉华, 崔艳荣, 贾瀛睿

(长江大学计算机科学学院,湖北荆州 434023)

摘要:针对真实场景下水稻虫害识别的背景复杂、模型计算量和参数量大以及难以在嵌入式设备或移动设备上部署等问题,在 YOLO v8 的基础上提出一种改进的轻量化的 YOLO v8 - Rice 水稻虫害检测算法。首先,采用 Context Guided Block 结构替换传统 YOLO v8 中 C2f 模块的 Bottleneck 结构,增强模型的上下文信息理解能力,压缩模型的权重;然后,使用深度可分离卷积代替传统 YOLO v8 中的标准卷积,以降低参数量、计算量;最后,将检测头重构为轻量级共享卷积检测头,以进一步降低参数量、计算量,并提高模型对多尺度虫害特征的定位和提取能力,使其能够更好地适应不同尺寸、复杂度的虫害状况。结果表明,相比于传统 YOLO v8,YOLO v8 - Rice 算法在计算量、参数量方面分别减小 70.5%、61.7%,模型的权重文件大小降低至 1.94 MB,仅为 YOLO v8n 的 32.4%,并且在平均精度上达到 94.1%,与其他模型相比明显提高。该算法在水稻虫害检测方面的性能取得了显著提升。借助轻量化网络模型及优化模型的部署,使其更适合在移动设备或嵌入式设备中部署,可为实际农业场景中的水稻虫害检测提供更可行的解决方案,可以准确地检测定位和分类水稻虫害。

关键词:水稻虫害检测;轻量化 YOLO v8 - Rice;Context Guided Block;深度可分离卷积;轻量级共享卷积检测头

中图分类号:S126;S435.112;TP391.41 **文献标志码:**A **文章编号:**1002-1302(2024)20-0277-08

水稻种植在确保我国的粮食供应、促进农业可持续发展方面扮演着至关重要的角色^[1]。然而,在水稻的生长周期中,众多病虫害频繁且严重的侵袭对水稻产量及其质量产生了较为恶劣的负面影响^[2]。有数据显示,2006—2015 年,我国水稻由病虫害导致的损失占比达 33.67%,但是通过有效的防治手段,能够成功挽回其中 55.18% 的潜在损失,表明综合性的植物保护策略对于维护水稻生产的稳定性具有重大贡献^[3]。

传统的水稻虫害识别和诊断主要依靠人工的直接观察与基于经验的判断,这种方式在实施上存在效率低下的问题,同时,由于其结果受到个人经验和主观判断的限制,可能导致准确性的波动。近年来,随着机器学习和深度学习领域的迅猛发展,这些先进的技术在农业害虫及疾病识别上实现了重大突破,给水稻种植业的可持续增长带来了希

望。江顺等提出了一种改进的水稻害虫识别技术,该技术以 AlexNet 为基础框架,通过引入额外的深层卷积层,去除局部响应归一化层和全连接层,并采用修正线性单元作为激活函数,以提高在多变环境下对水稻害虫的识别精度^[4]。梁万杰等针对水稻虫害识别问题,提出一种基于卷积神经网络模型的水稻二化螟虫害识别方法,通过对图像进行预处理、重新设计卷积神经网络模型,增强了网络模型对水稻二化螟虫害识别的抗干扰性和鲁棒性^[5]。郑果等以 MobileNet v3 作为骨干网络,对 YOLO v7 网络进行改进,并构建了基于特征金字塔的多尺度神经网络模型,提升了小个体害虫的识别精度,实现了水稻虫害远程实时自动化识别^[6]。曾伟辉等使用增强型超分辨率生成对抗网络对低分辨率图像进行数据增强,通过构建 SCResNeSt 网络并在骨干网络中使用 ResNeSt block 以进一步提升对图像中害虫信息获取的准确性,实现了野外实际场景下水稻害虫识别的应用需求^[7]。巨志勇等在传统 YOLO v5s 模型的基础上融合了 CA 注意力机制,并更换 PANet 为 BiFPN 结构,使用 CARAFE 结构来代替原始的上采样模块,替换损失函数为 EIoU 等操作,提出了一种高效快速的检测算法 YOLO - Rice,具有更高的准确率、鲁棒性^[8]。崔金荣等针对水稻

收稿日期:2024-03-23

基金项目:国家自然科学基金面上项目(编号:62077018);湖北省自然科学基金(编号:2023AFB082)。

作者简介:桂余鹏(1999—),男,湖北襄阳人,硕士研究生,研究方向为深度学习与目标检测。E-mail:2022710691@yangtzeu.edu.cn。

通信作者:胡蓉华,博士,讲师,研究方向为网络与信息安全。E-mail:hrh2016@yangtzeu.edu.cn。

病害识别方法准确度低、模型收敛速度缓慢的问题,在 MobileNet v3 基础上,通过微调的迁移学习策略完善了模型的训练,然后添加了 CA 模块,最终加入 SVM 多分类器,实现了高性能、易部署的水稻病害分类识别目标^[9]。

上述方法虽然都可以较为精准地识别研究目标,但是很少考虑到所提方法是否能够兼顾精度、参数量、计算量及模型大小。为了解决此类问题,本研究提出一种基于 YOLO v8 - Rice 的水稻虫害检测算法。该算法通过改进 YOLO v8,旨在实现模型轻量化的同时,保证模型的性能,以满足在嵌入式设备或移动设备终端进行实时监测的需求。本研究的主要贡献有 3 点:(1) 替换 C2f 中的 Bottleneck 为 Context Guided Block^[10],增强网络处理局部特征、周围上下文和全局上下文信息的能力,并有效融合这些信息,提高模型对特征的定位精度及对其周围环境的理解能力;(2) 替换标准卷积为深度可分离卷积^[11],有效减少了模型的参数量、计算量,在保持模型检测性能稳定性的同时,提高了模型的运行效率;(3) 轻量级共享卷积检测头,

重新设计检测头结构,通过使用共享卷积,大幅减小模型的参数量、计算量,进一步压缩模型的权重文件大小,提升检测头的定位和分类性能,并且能够提升模型对多尺度特征的理解能力。

1 材料与方法

1.1 数据集介绍

本研究所用数据集样本主要来源于 2 个不同的途径,一是 Kaggle 在线数据科学平台,二是利用 Baidu、Google、Bing 等多个搜索引擎进行检索。经过筛选和整理,建立了 1 个水稻虫害的数据集,该数据集共有 2 078 张图片,涵盖 7 种不同的水稻虫害类型,具体包括稻螟蛉卵、稻螟蛉虫、稻螟蛉蛾、稻大螟幼虫、点斑 I 型稻绿蝽、点斑 II 型稻绿蝽、全绿型稻绿蝽等,部分水稻害虫如图 1 所示。本研究所用训练模型为 YOLO 系列格式的数据集,因此,采用 LabelImg 图像工具标记害虫类型及其位置,生成 xml 类型的标注文件,再将 xml 格式的文件转换为 YOLO 训练的 txt 格式的文件。按照 7 : 2 : 1 的比例将数据集随机划分为训练集、测试集、验证集。



图1 部分水稻害虫示意

1.2 YOLO v8 网络结构

YOLO v8 是 Ultralytics 公司在 2023 年 1 月 10 日开源的 YOLO v5 的下一个重大更新版本,YOLO v8 继承了 YOLO 系列^[12-19]之前的成果,并在图像分类、目标检测和实例分割等多个领域得到了广泛应用。YOLO v8 发布时推出了 5 种不同尺寸的网络模型,标记为 n、s、m、l 和 x,这些模型的大小依次递增,精度也相应提高。表 1 中展示了 YOLO v8 5 个版本的计算量和参数量。

考虑到移动设备和嵌入式设备的资源限制及对轻量化的需求,本研究采用参数量、计算量最小的 YOLO v8n 模型。如图 2 所示,YOLO v8 的主要

表 1 不同版本 YOLO v8 模型的比较

模型	参数量 (M)	计算量 (G)
YOLO v8n	3.2	8.7
YOLO v8s	11.2	28.6
YOLO v8m	25.9	78.9
YOLO v8l	43.7	165.2
YOLO v8x	68.2	257.8

结构包括骨干网络、颈部网络、头部网络 3 个部分。在骨干网络部分,YOLO v8 使用了梯度流更丰富的 C2f 结构,C2f 主要负责特征的融合工作。该结构的核心在于它能够融合不同层级的特征图,从而使模

型获取更加丰富的信息,这种融合机制显著增强了模型对图像内容的识别能力,有助于减少在处理过程中的信息损耗。C2f 模块的设计使得 YOLO v8 能够更有效地捕捉和利用图像中的各种细节,从而提高了目标检测的准确性。在颈部网络部分,YOLO v8 采用了 C2f 模块、PANet 结构来进行特征聚合,由

浅到深逐层聚合信息。在头部网络部分,YOLO v8 采用了主流的解耦头结构,将分类和检测头分离,同时采用了 Anchor-Free 策略。通过以上改进策略,YOLO v8 在损失计算和网络结构方面取得了显著进展。

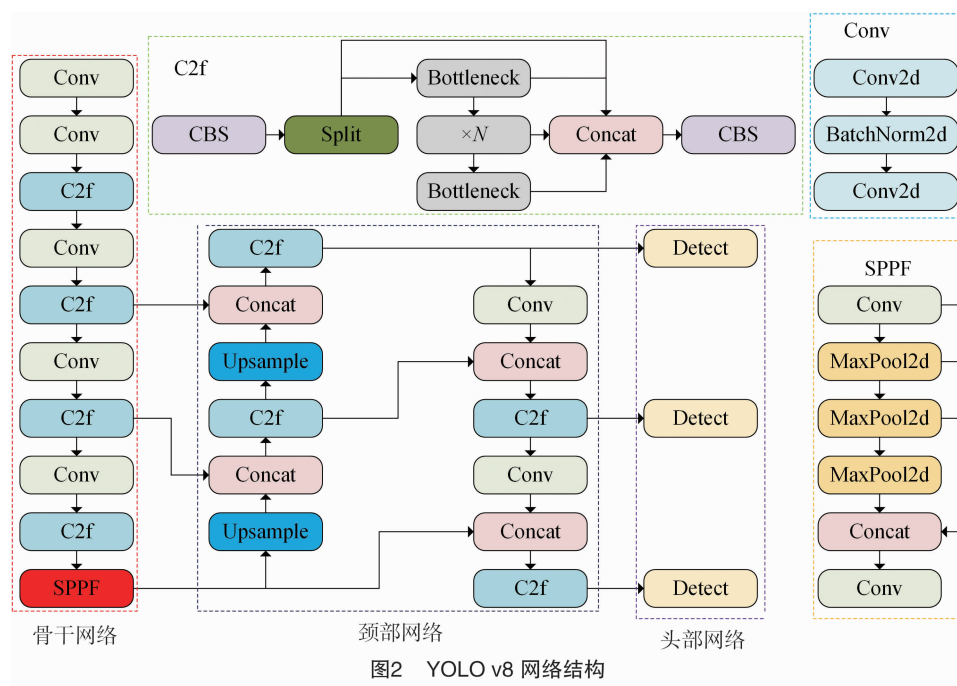


图2 YOLO v8 网络结构

1.3 YOLO v8 - Rice 网络结构

本研究算法 YOLO v8 - Rice 是在 YOLO v8 的基础上作出的改进,其结构如图 3 所示。首先,将 Context Guided Block 取代 C2f 中的 Bottleneck 模块,形成 C2f_ContextGuided,以替代原网络中的 C2f 模块;然后,将原网络中的标准卷积替换为深度可分离卷积;最后,将检测头重新设计为轻量级共享卷积检测头,形成 Detect_LSCD,以代替原网络中的检测头结构。

1.3.1 Context Guided Block 模块 在 YOLO v8 中,C2f 模块由多个 Bottleneck 块组成,每个块都包含 2 个卷积层。这些卷积层对输入特征图进行变换,提取出更高级别的特征表示。C2f 模块采用 Bottleneck 的设计理念,虽然可以使主干网络学习到多尺度特征,但是同时也会使模型变大。在 YOLO v8 - Rice 算法中,为了压缩模型大小,同时提高主干网络对上下文信息的理解能力,采用 Context Guided Block 结构替换 C2f 中的 Bottleneck 结构。

Context Guided Block 结构如图 4 所示,主要由局部特征提取器、周围上下文特征提取器、联合特

征提取器和全局上下文特征提取器组成。传统的下采样技术在处理图像时可能会遇到一些问题,如信息损失、分辨率降低和感受野受限等,这些问题可能会影响模型对图像中细节、整体结构的把握,特别是在需要精确识别的任务(如虫害检测)中,这些限制可能会对模型的性能产生不利影响。为了解决这些问题,特引入 Context Guided Block 模块,以帮助模型更好地捕捉害虫与其周围环境之间的关系。通过结合全局和局部的上下文信息,该模块有助于提高模型在害虫识别方面的准确性。

Context Guided Block 模块主要包括 2 个步骤:

(1)局部特征的提取。使用了 1 个相当于 3×3 标准卷积层的局部特征提取器,它从周围的 8 个相邻特征向量中学习局部特征。同时,为了获取更大感受野并有效学习周围上下文信息,还使用 1 个相当于 3×3 空洞卷积的周围上下文特征提取器。然后,联合特征提取器,将局部特征提取器和周围上下文特征提取器的输出特征结合起来,并通过批量归一化、参数化 ReLU 激活函数进行处理。

(2)全局上下文特征的聚合。首先,使用 1 个全

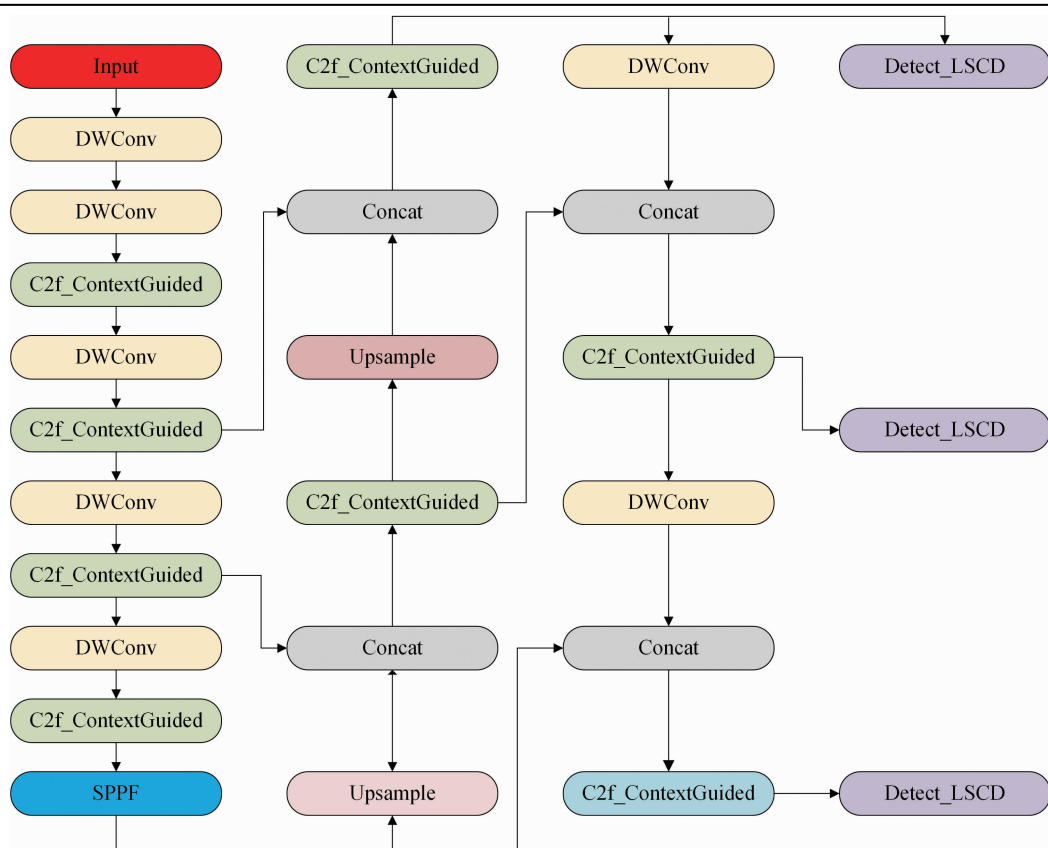


图3 YOLO v8-Rice 网络结构

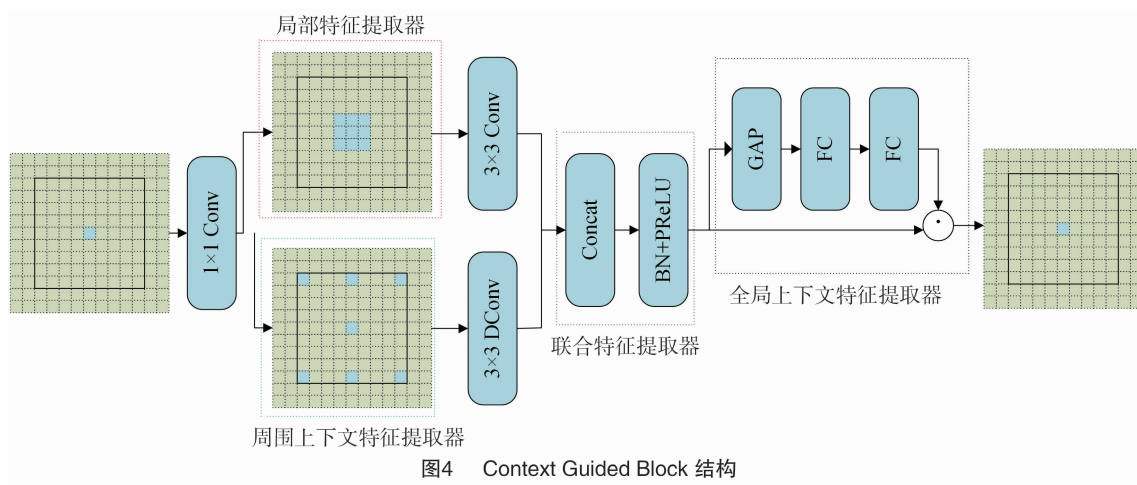


图4 Context Guided Block 结构

局平均池化层来收集全局上下文信息。然后,通过 1 个多层感知机进一步提取这些全局上下文特征。最后,通过加权操作得到输出的特征图。

YOLO v8 - Rice 算法采用 Context Guided Block 结构替代 C2f 中的 Bottleneck 结构,从而提高模型对多尺度、多场景的适应能力,增强模型对上下文信息的提取能力,提高模型在虫害检测任务中的性能和准确性,在后面的试验中也得到了证实。

1.3.2 深度可分离卷积 YOLO v8 中的标准卷积

利用若干个多通道卷积核对输入的多通道图像进行处理,既提取了通道特征,又提取了空间特征。虽然在目标检测模型中起到关键作用,但是其计算量大,容易造成参数冗余,并且占用的资源较多。

为了解决上述问题,在 YOLO v8 - Rice 算法中引入深度可分离卷积代替标准卷积。深度可分离卷积由逐深度卷积和逐点卷积组成,逐深度卷积用于提取空间特征,逐点卷积用于提取通道特征,显著减少了计算量、参数数量。

深度可分离卷积的结构见图 5。在逐深度卷积过程中,通道数不变,宽度或高度改变,由于逐深度卷积的卷积核为单通道模式,需要对输入的每个通道进行卷积,得到与输入特征图通道数一致的输出特征图,因此在逐深度卷积中,输入特征图通道数、

卷积核数及输出特征图数保持一致,因此可能会导致输出的特征图数过少,影响信息的有效性,此时就需要进行逐点卷积。逐点卷积的实质是用 1×1 的卷积核来控制维度,将所有通道进行聚合。在逐点卷积的过程中,宽度或高度不变,通道数改变。

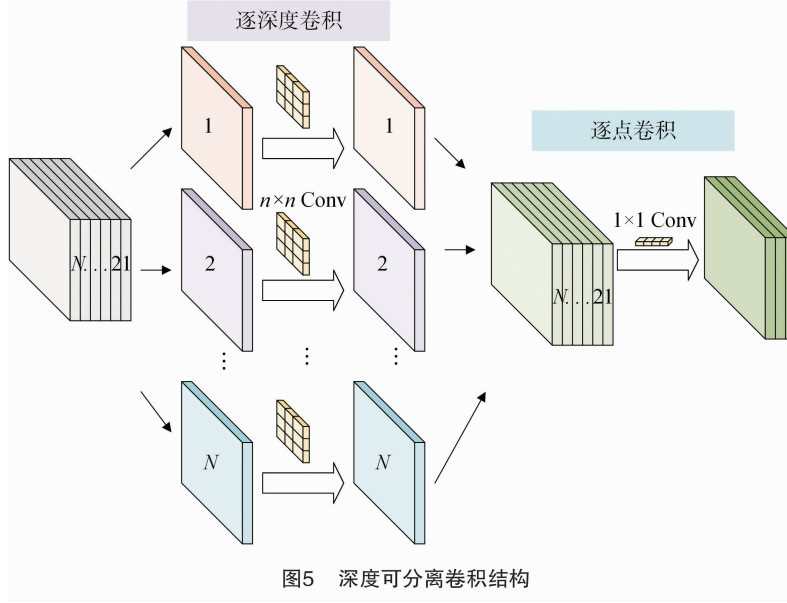


图5 深度可分离卷积结构

假设卷积核大小为 $D_K \times D_K$, 输入通道为 M , 输出通道为 N , 输出的特征图尺寸为 $D_F \times D_F$, 通过标准卷积后, 参数量的计算公式如下:

$$Params(Conv) = D_K \times D_K \times M \times N. \quad (1)$$

标准卷积浮点运算量为:

$$FLOPs(Conv) = D_K \times D_K \times M \times N \times D_F \times D_F. \quad (2)$$

通过深度可分离卷积的参数量为:

$$Params(DWConv) = D_K \times D_K \times M + M \times N. \quad (3)$$

深度可分离卷积浮点运算量为:

$$FLOPs(DWConv) = D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F. \quad (4)$$

对比深度可分离卷积和标准卷积的参数量和浮点运算量,有:

$$\frac{Params(DWConv)}{Params(Conv)} = \frac{D_K \times D_K \times M + M \times N}{D_K \times D_K \times M \times N} = \frac{1}{N} + \frac{1}{D_K^2}; \quad (5)$$

$$\frac{FLOPs(DWConv)}{FLOPs(Conv)} = \frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2}. \quad (6)$$

由此可知,深度可分离卷积的参数量、浮点运算量远远小于标准卷积。由于深度可分离卷积的浮点运算量更小,能够进一步精简计算模型,后续的试验也为此提供了证明。

1.3.3 轻量级共享卷积检测头 YOLO v8 的检测头模块采用 Decoupled Head, 其结构如图 6 所示。检测头将回归分支和分类分支分离, 每个分支都会进行 2 个 3×3 卷积核的 Conv 模块和 1 个 1×1 卷积核的 Conv2d 模块, 这样会产生很多计算量、参数量。Conv 模块采用 BatchNorm2d 结构, 可以解决内部协变量偏移, 缓解梯度饱和问题。但是 BatchNorm2d 受限于批尺寸, 对于较小的批尺寸, BatchNorm2d 的表现可能较差。为了解决这些问题, YOLO v8 - Rice 算法中重新设计了一种轻量级共享卷积检测头, 其结构如图 7 所示。轻量级共享卷积检测头首先采用 GroupNorm2d 代替 BatchNorm2d, GroupNorm2d 将通道分为若干组, 然后对组内的通道进行归一化处理。由于 GroupNorm2d 是在组级别进行归一化处理, 因此它可以显著减小模型的计算量, 并且它不依赖于批尺寸的大小, 因此它适用于较小的批尺寸。这样可以保证在保持归一化效果的同时, 减小模型的计算量。然后, 先采用 1×1 的卷积核来将输出的特征图

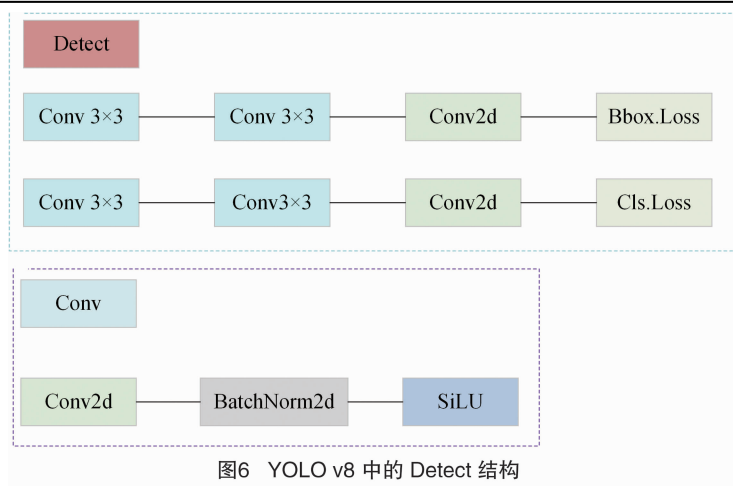


图6 YOLO v8 中的 Detect 结构

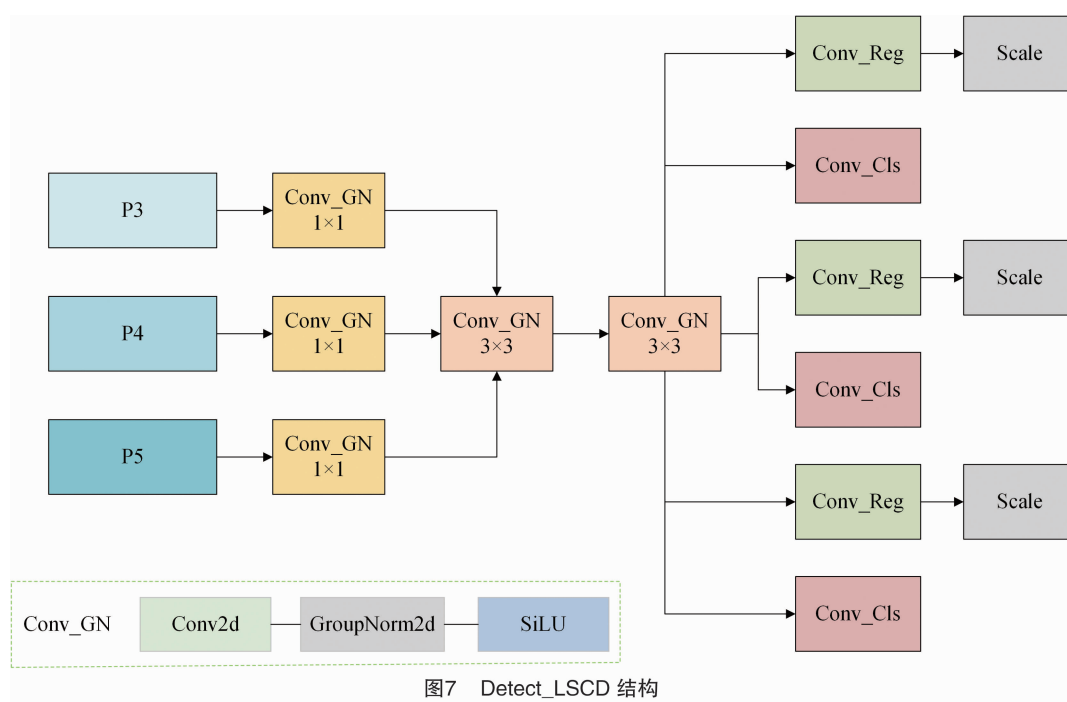


图7 Detect_LSCD 结构

维度固定为 256×256 , 以减小模型的计算量, 再将 2 个 3×3 的 Conv_GN 卷积进行串联, 统一经过这 2 个卷积实现参数共享, 减小参数量。最后, 为了应对每个检测头检测的目标尺度不一致的问题, 使用 Scale 层对特征进行缩放, 从而有效提高模型的多尺度检测能力。

2 结果与分析

2.1 评价指标

本研究采用的评价指标有平均精度 ($mAP@0.5$)^[20]、参数量、计算量和模型权重。

mAP 是目标检测任务中常用的性能评估指标之一, 用于衡量模型在不同类别上的平均精度。 $mAP@0.5$ 表示当 IoU 阈值设定为 0.5 时的平均精

度, 即当 IoU 阈值 ≥ 0.5 时, 认为预测框和真实框是匹配的。 mAP 的计算公式如下:

$$mAP = \frac{1}{N} \int_0^1 Precision(Recall) d(Recall) \quad (7)$$

式中: $Precision$ 表示准确率; $Recall$ 表示召回率; N 表示预测的类别数。

参数量是衡量模型复杂度、计算成本的一个重要指标, 指的是模型中可训练参数的总数, 较低的数值通常表示模型更轻量化, 更利于在资源受限的情况下部署。

计算量是模型进行推理所需的浮点运算次数的度量, 它反映了模型在进行目标检测时的计算复杂度。计算量越大, 表明模型通常越复杂, 需要更

多计算资源和时间。

模型权重大小即模型所有层参数所占用的存储空间,影响着模型的加载速度、存储成本、能耗及部署的便利性。较小的模型通常意味着模型更轻量化及更容易完成部署。

2.2 试验环境的配置

本研究通过搭建 PyTorch 深度学习框架,对自定义数据集进行训练和测试。操作系统选用 Windows 专业工作站版 21H2, CPU 为 Intel i5 - 12490F, GPU 为 GeForce RTX 3080Ti, 显存为 12 GB,内存为 32 GB, 训练环境配置为 Python 3.9.18, Pytorch 2.1.0, Cuda 11.8, YOLO v8 版本为 8.1.9。试验时间为 2023 年 9 至 2024 年 2 月,试验地点为长江大学东校区 3 号教学楼。

2.3 消融试验

为了更好地验证 YOLO v8 - Rice 算法的有效性,本研究进行了消融试验,共试验了 3 种不同改进方法,使用同样的数据集划分进行测试,epoch 设置为 200。

由表 2 可以看出,改进 1 通过将 YOLO v8 中

C2f 模块的 Bottleneck 结构替换为 Context Guided Block 结构,模型的 $mAP@0.5$ 基本不变,同时参数量、计算量分别降低了 30.1%、28.4%,模型权重大小压缩了 1.7 MB,说明模型融合信息的能力有所增强,对上下文的理解能力有所提升,这一改进对于水稻虫害检测的轻量化非常有帮助。改进 2 在改进 1 的基础上,将 YOLO v8n 中的标准卷积替换为深度可分离卷积,模型的 $mAP@0.5$ 略微回升,同时参数量、计算量大幅降低,模型权重大小得到进一步压缩,这一改进有效减小了模型的参数量、计算量,提升了模型检测性能的稳定性。改进 3 在改进 2 的基础上,将检测头重构为轻量级共享卷积检测头,提升了模型对多尺度特征的理解能力,模型的 $mAP@0.5$ 进一步得到提升,同时参数量、计算量分别降低至基础模型的 29.5%、38.3%,模型权重大小仅为 1.94 MB,这一改进提升了模型的定位和识别水稻虫害的能力,实现了模型的轻量化。通过上述消融试验的全面评估,验证了在基础模型上,不同改进都表现出优异性能。

表 2 消融试验结果

模型	改进方法	$mAP@0.5$ (%)	参数量 (M)	计算量 (G)	模型权重 (MB)
基础模型	YOLO v8n	93.8	3.007	8.1	5.98
改进 1	YOLO v8n + C2f_ContextGuided	93.3	2.102	5.8	4.28
改进 2	YOLO v8n + C2f_ContextGuided + DWConv	93.4	1.532	4.7	3.19
改进 3	YOLO v8n + C2f_ContextGuided + DWConv + Detect_LSCD	94.1	0.888	3.1	1.94

2.4 对比试验

为了进一步验证 YOLO v8 - Rice 算法的效果。本研究选取 YOLO v3 - tiny、YOLO v5n、YOLO v6n、YOLO v8n 及常见的 RepViT、FasterNet、EfficientViT 来优化 YOLO v8n 的骨干网络,与 YOLO v8 - Rice 算法进行对比试验,试验模型的对比结果如表 3 所示。为了保证试验结果的严谨,本研究中所有模型的参数设定都保持一致。

由表 3 可知,YOLO v8 - Rice 算法在目标检测任务中具有显著的优越性。首先,YOLO v8 - Rice 的 $mAP@0.5$ 达到了 94.1%,明显高于其他模型,表明其在水稻虫害中定位害虫、识别害虫方面具有出色的准确性。与 YOLO v8n 相比,YOLO v8 - Rice 算法的参数量、计算量分别降低了 70.5%、61.7%,模型的权重文件大小压缩了 67.6%,并且 $mAP@0.5$ 还有 0.3% 的提升。上述结果表明,YOLO v8 - Rice 在保持高识别精度的同时,大幅度减小了参数量、计算

表 3 不同模型的对比试验结果

模型	$mAP@0.5$ (%)	参数量 (M)	计算量 (G)	模型权重 (MB)
YOLO v3 - tiny	94.0	12.131	18.9	23.20
YOLO v5n	91.9	2.504	7.1	5.05
YOLO v6n	93.4	4.234	11.8	8.31
YOLO v8n	93.8	3.007	8.1	5.98
YOLO v8n + RepViT	91.9	6.700	18.5	13.30
YOLO v8n + FasterNet	93.4	1.532	4.7	3.19
YOLO v8n + EfficientViT	93.5	4.010	9.4	8.36
YOLO v8 - Rice	94.1	0.888	3.1	1.94

量,模型的权重文件大小也得到了极大压缩,证明了本研究中轻量化改进的有效性,能够满足水稻虫害识别的实际需求。

综上所述,本研究提出的 YOLO v8 - Rice 算法在 $mAP@0.5$ 、参数量、计算量及模型权重等多个关键性能指标方面均展现了显著的优势,具有轻量

化、高性能特性,使其在水稻虫害检测中具有广泛的应用潜力。

3 结论

本研究针对水稻虫害识别的背景复杂、模型计算量和参数量大,以及难以在嵌入式设备或移动设备上部署等问题,在 YOLO v8 的基础上,提出了一种改进的轻量化 YOLO v8 - Rice 模型。通过将 YOLO v8 中 C2f 模块的 Bottleneck 结构替换为 Context Guided Block 结构,将 YOLO v8n 中的标准卷积替换为深度可分离卷积,将检测头重构为轻量级共享卷积检测头,试验结果的 $mAP@0.5$ 高达 94.1%,参数量、计算量分别为 0.888 M、3.1 G,模型的权重文件大小为 1.94 MB,表明 YOLO v8 - Rice 在识别水稻虫害方面有相对更好的效果,实现了对 7 类常见水稻虫害的有效识别。YOLO v8 - Rice 模型更加适应真实环境,多尺度特征学习能力有所增强,与 YOLO v8n 相比, $mAP@0.5$ 提升了 0.3 个百分点,并且参数量、计算量分别降低了 70.5%、61.7%,模型的权重文件大小压缩了 67.6%,验证了改进的有效性,并且该方法可以推广到其他农作物虫害的识别问题上。

综上所述,本研究提出的 YOLO v8 - Rice 算法实现了模型的轻量化,并且在目标检测任务中取得了显著的性能优势,为真实场景下水稻虫害检测的应用提供了更经济、准确、高效的解决方案。未来的研究可以进一步优化算法、提高虫害的检测速度、进一步丰富数据集以提升模型的泛化性能。

参考文献:

- [1] 吕亮,常向前,张舒. 水稻害虫绿色防控技术应用研究现状及展望[J]. 环境昆虫学报,2021,43(3):623-632.
- [2] 赵景,蔡万伦,沈栋阳,等. 水稻害虫绿色防控技术研究的发展现状及展望[J]. 华中农业大学学报(自然科学版),2022,41(1):92-104.
- [3] 刘万才,刘振东,黄冲,等. 近 10 年农作物主要病虫害发生危害情况的统计和分析[J]. 植物保护,2016,42(5):1-9.
- [4] 江顺,黄红星,莫里楠,等. 基于改进 AlexNet 的岭南水稻虫害识别方法研究[J]. 江苏农业科学,2023,51(23):187-195.
- [5] 梁万杰,曹宏鑫. 基于卷积神经网络的水稻虫害识别[J]. 江苏农业科学,2017,45(20):241-243,253.
- [6] 郑果,姜玉松,沈永林. 基于改进 YOLO v7 的水稻害虫识别方法[J]. 华中农业大学学报(自然科学版),2023,42(3):143-151.
- [7] 曾伟辉,张文凤,陈鹏,等. 基于 SCResNeSt 的低分辨率水稻害虫图像识别方法[J]. 农业机械学报,2022,53(9):277-285.
- [8] 巨志勇,易成,周重臣,等. YOLO - Rice:一种基于 YOLO v5 的水稻虫害检测[J/OL]. 控制工程,2004;1-10(2024-03-15)[2024-05-30]. <https://doi.org/10.14107/j.cnki.kzgc.20231000>.
- [9] 崔金荣,魏文钊,赵敏. 基于改进 MobileNet v3 的水稻病害识别模型[J]. 农业机械学报,2023,54(11):217-224,276.
- [10] Wu T Y, Tang S, Zhang R, et al. CGNet: a light-weight context guided network for semantic segmentation[J]. IEEE Transactions on Image Processing, 2020,30:1169-1179.
- [11] Chollet F. Xception: deep learning with depthwise separable convolutions[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE, 2017: 1251-1258.
- [12] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016: 779-788.
- [13] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI: IEEE, 2017: 7263-7271.
- [14] Redmon J, Farhadi A. YOLO v3: an incremental improvement[EB/OL]. (2018-04-08)[2024-01-20]. <http://arxiv.org/abs/1804.02767>.
- [15] Bochkovskiy A, Wang C Y, Liao H Y M. YOLO v4: optimal speed and accuracy of object detection[EB/OL]. [2024-01-20]. <http://arxiv.org/abs/2004.10934>.
- [16] Ge Z, Liu S T, Wang F, et al. YOLOX: exceeding YOLO series in 2021[EB/OL]. (2021-07-18)[2024-01-20]. <http://arxiv.org/abs/2107.08430>.
- [17] Li C Y, Li L L, Jiang H L, et al. YOLO v6: a single-stage object detection framework for industrial applications[EB/OL]. (2022-09-07)[2024-01-20]. <http://arxiv.org/abs/2209.02976>.
- [18] Wang C Y, Bochkovskiy A, Liao H Y M. YOLO v7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver: IEEE, 2023: 7464-7475.
- [19] Xu X Z, Jiang Y Q, Chen W H, et al. DAMO - YOLO: a report on real-time object detection design[EB/OL]. (2022-11-23)[2024-01-20]. <http://arxiv.org/abs/2211.15444>.
- [20] 杨佳昊,左昊轩,黄祺成,等. 基于 YOLO v5s 的作物叶片病害检测模型轻量化方法[J]. 农业机械学报,2023,54(增刊1):222-229.