doi:10.6041/j.issn.1000-1298.2026.02.021

基于轻量化 YOLO v8n 的自然环境下车前草病虫害检测方法

祝诗平　周　杰　张　越　唐茂杰　林　曦
（西南大学工程技术学院，重庆 400716）

摘要：为实现在自然环境下对车前草病虫害的快速准确检测，构建车前草虫害、车前草白粉病、车前草叶斑病和车前草花叶病 4 种常见病虫害的 2 290 幅图像数据集，并提出一种基于 YOLO v8n－Plantago 的车前草病虫害检测方法。利用轻量化模型 MobileNetV3 替换原模型的 Backbone，实现网络保持轻量化的同时提高模型响应速度；在模型的 Neck 部分使用 VoV－GSCSP 模块，替换第 15、18、21 层的 C2f 模块，以更高效地处理特征图并保留更多特征信息；在 Head 中引入 RepVGG 模块，加快模型的推理速度、减少模型内存占用量，进一步提高模型的准确率，实现对车前草病虫害快速、准确识别。在自建的车前草病虫害数据集上进行验证，结果显示，YOLO v8n－Plantago 在车前草病虫害检测中的 mAP@0.5 达 89.43%，相比于 YOLO v8n 模型提升 2.04 个百分点，并且浮点运算量下降 61.43%，模型内存占用量下降 3.53%，参数量下降 7.31%。将模型部署到边缘端设备上，推理速度提升 33.26%，后处理速度提升 8.08%。改进的 YOLO v8n－Plantago 模型有效地实现了车前草病虫害检测，为研制车前草病虫害自动识别与精准喷药装置奠定了基础。

关键词：车前草；病虫害检测；YOLO v8n；MobileNetV3；VoV－GSCSP；RepVGG
中图分类号：TP391.4；S24　　**文献标识码**：A　　**文章编号**：1000-1298(2026)02-0215-10　　**OSID**：

Detection Method for Pests and Diseases of *Plantago asiatica* L. in Natural Environments Based on Lightweight YOLO v8n

ZHU Shiping　ZHOU Jie　ZHANG Yue　TANG Maojie　LIN Xi
(*College of Engineering and Technology*, *Southwest University*, *Chongqing* 400716, *China*)

Abstract: Aiming to achieve fast and accurate identification of pests and diseases of *Plantago asiatica* L. in natural environments, a dataset comprising 2 290 images of four common diseases—insect pest, powdery mildew, leaf spot, and mosaic disease—was constructed, and a recognition method for pests and diseases of *Plantago asiatica* L. based on YOLO v8n－Plantago was proposed. The lightweight MobileNetV3 was utilized to replace the original model's Backbone, ensuring the network remained lightweight while enhancing response speed. The VoV－GSCSP feature network was applied in the Neck part of the model to replace the C2f modules in layers 15, 18 and 21, which can retain more feature information while processing the feature map more efficiently. The RepVGG module was introduced in the Head of the network to accelerate inference speed and reduce memory footprint, further improving model accuracy for rapid and accurate recognition of *Plantago asiatica* L. pests and diseases. Validation on the self-built dataset of pest and disease of *Plantago asiatica* L. showed that YOLO v8n－Plantago achieved mAP@0.5 of 89.43% for identifying *Plantago asiatica* L. pests and diseases, an increase of 2.04 percentage points compared with that of the YOLO v8n model, with FLOPs reduced by 61.43%, model memory usage decreased by 3.53%, and parameter count reduced by 7.31%. When deployed on an edge device, inference speed was increased by 33.26%, and processing speed was increased by 8.08%. The proposed YOLO v8n－Plantago model effectively facilitated the detection of *Plantago asiatica* L. pests and diseases, laying a foundation for developing automatic identification and precise spraying devices for *Plantago asiatica* L. pests and diseases.

Key words: *Plantago asiatica* L.; pest and disease detection; YOLO v8n; MobileNetV3; VoV－GSCSP; RepVGG

收稿日期：2024－10－01　修回日期：2024－12－24
基金项目：教育部中国高校产学研创新基金——德州专项(2021DZ005)
作者简介：祝诗平(1969—)，男，教授，博士，主要从事智能检测与控制研究，E-mail：zspswu@126.com

0 引言

车前草(*Plantago asiatica* L.)是一种中草药,在《神农本草经》记载车前草“主气癃”,具有“止痛、利水道小便、除湿痹”的功效[1]。近年来,车前草的药用价值已引起广泛关注。在大规模种植过程中,车前草易受到多种病虫害威胁,这些病虫害在不同生长阶段都具有强传染性,若不及时防治,将严重影响产量[2]。传统上,车前草病虫害的评估主要依赖专家经验,但这种方法具有较强的主观性,不利于标准化。

计算机视觉技术在作物病虫害识别中的应用发展迅速,显著提高了识别速度和准确性。目前的研究多集中于背景较简单、特征明显的农作物病虫害图像,通过这些图像优化模型结构和算法性能,使模型在不同作物病虫害识别任务中表现出高精度和快速推理的优势,有效推动了病虫害自动检测技术的发展[3-6]。然而,这类研究多基于受控或理想化环境下采集的数据,与实际农业环境中的复杂背景存在差异,可能影响模型识别效果。针对这一问题,研究者们在自然环境和复杂背景下对模型进行了优化,通过结构调整、特征增强或轻量化改进,使模型在复杂环境下的作物病害和虫害识别中表现出较高的精度和稳定性,提升了实际应用的可行性和可靠性[7-9]。该方向的研究不仅验证了模型在实验室外的适应性,也为智能病虫害检测和精准农业提供了重要支持。

综上,计算机视觉技术在农作物病虫害检测中表现优异,但针对车前草病虫害研究仍相对较少,显示出计算机视觉在该领域的巨大潜力。因此本文提出一种改进的 YOLO v8n 模型,并命名为 YOLO v8n-Plantago。通过使用 MobileNetV3 减少参数量和复杂度,引入 VoV-GSCSP(Vovnet-based ghost spatial cross stage partial module)增强特征提取能力,并在 Head 部分集成 RepVGG(Re-parameterizable vgg network)模块提升检测速度与精度。以期为车前草病虫害自动识别与精准喷药装置研发奠定基础。

1 材料与方法

1.1 图像采集

车前草病虫害图像数据集采集自四川省自贡市富顺县(104°77′N,29°33′E),采集时间为 2023 年 12 月—2024 年 4 月,覆盖车前草从幼苗期到结果期各生长阶段。拍摄场景包括晴天、阴天以及不同时间段(上午、下午),选择 4 种常见病虫害:车前草虫害[10]、车前草白粉病[11]、车前草叶斑病[12]、车前草花叶病[13]。拍摄设备为智能手机,图像格式为 JPEG,尺寸为 4 032 像素×3 240 像素,共采集 1 325 幅图像。此外,通过网络爬虫获取 800 幅图像,筛选后最终获得 1 897 幅有效病虫害图像,部分样本如图 1 所示。



(a) 车前草虫害 (b) 车前草白粉病 (c) 车前草叶斑病 (d) 车前草花叶病

图 1 4 种病虫害图像

Fig. 1 Images of four kinds of pests and diseases

1.2 数据集构建

将车前草图像数据按比例 7∶2∶1 划分为训练集(1 328 幅)、验证集(379 幅)和测试集(190 幅)。由于图像中存在土壤背景和叶片重叠等干扰,采用数据增强方法(如调整曝光度、旋转角度和添加高斯噪声),提高训练样本多样性,增强模型泛化能力。训练集增加至 1 721 幅,测试集和验证集保持不变,因此总数据集为 2 290 幅。数据标注采用 Makesense AI 工具,生成 YOLO 格式标注文件,其中标签包括“Insect Pest”“Powdery Mildew”“Leaf Spot”和“Mosaic Diseases”。最终数据集图像信息及对应的车前草标注框数量如表 1 所示。

表 1 车前草数据集详情

Tab. 1 Details of the *Plantago asiatica* L. dataset

数据集	图像数量/幅	病虫害叶片数量/片			
		车前草虫害	车前草白粉病	车前草叶斑病	车前草花叶病
训练集	1 721	1 636	3 166	4 027	651
验证集	379	470	905	673	111
测试集	190	224	457	345	51
总计	2 290	2 330	4 528	5 045	813

2 网络模型训练

2.1 实验环境及超参数设置

所有模型均使用相同超参数进行训练,其中训练轮数为 300;初始学习率设置为 0.001;批大小为 64;优化器为 Adam;边界框回归损失函数为 CIoU。改进模型和参与对比实验模型均在相同环境下进行。所用操作系统为 Windows10,CPU 型号为 Intel Core i5-12400F,GPU 型号为 NVIDIA GeForce RTX 4060Ti,运行内存为 16 GB,编程语言为 Python 3.9,深度学习框架为 Pytorch 2.0.1。

2.2　评价指标

使用精确率(Precision,P)、召回率(Recall,R)、平均精度(Average precision,AP)、模型内存占用量、参数量、平均精度均值(Mean average precision,mAP)、浮点运算量多指标评估模型。其中平均精度均值采用 mAP@0.5。

3　车前草病虫害检测模型构建

3.1　模型选择

在车前草病虫害数据集上训练并对比多种 YOLO 模型(YOLO v5n、YOLO v5s、YOLO v6n[14]、YOLO v7[15]、YOLO v8n[16]、YOLO v8s、YOLO v9c[17]、YOLO v10n[18] 和 YOLO 11n[19])。结果如表 2 所示,YOLO v7 在 mAP@0.5 上表现最佳,其次为 YOLO v9c、YOLO v8s、YOLO v5s 和 YOLO v8n。尽管 YOLO v7 和 YOLO v9c 精度较高,但由于模型内存占用量较大,不利于资源有限的边缘端设备部署。相比之下,YOLO v8n 在保持较高精度的同时,模型内存占用量和参数量较小,整体效果最优,因此选用 YOLO v8n 作为检测模型。

表 2　YOLO 系列模型结果对比

Tab.2　Comparison of results of YOLO series models

模型	精确率/%	召回率/%	mAP@0.5/%	浮点运算量	模型内存占用量/MB	参数量
YOLO v5n	84.24	77.69	86.24	4.20×10^{9}	3.90	1.77×10^{6}
YOLO v5s	87.21	78.62	87.62	1.60×10^{10}	13.70	7.03×10^{6}
YOLO v6n	86.40	77.57	87.32	1.14×10^{10}	9.97	4.63×10^{6}
YOLO v7	84.33	87.27	91.89	1.03×10^{11}	71.30	3.65×10^{7}
YOLO v8n	83.42	81.60	87.39	8.09×10^{9}	5.95	3.01×10^{6}
YOLO v8s	84.01	81.13	88.34	2.84×10^{10}	21.40	1.11×10^{7}
YOLO v9c	83.12	84.80	91.50	2.64×10^{11}	392.00	6.05×10^{7}
YOLO v10n	77.98	82.86	86.28	8.23×10^{9}	5.48	2.70×10^{6}
YOLO 11n	82.55	82.19	86.61	6.32×10^{9}	5.32	2.58×10^{6}

3.2　YOLO v8n 模型

YOLO v8 是 YOLO 系列中前沿目标检测算法,包括 5 种型号:YOLO v8n、YOLO v8s、YOLO v8l、YOLO v8m 和 YOLO v8x。模型由 4 部分组成:输入端(Input)、主干网络(Backbone)、颈部网络(Neck)和检测头(Head)。图像经过预处理后,Backbone 采用 C2f 模块,结合多个卷积层和残差连接,提升特征提取能力。Neck 部分整合特征金字塔网络(Feature pyramid network,FPN)和路径聚合网络(Path aggregation network,PAN)。其中,FPN 处理不同尺度的目标,PAN 增强位置特征,提升鲁棒性。在 Head 部分,采用解耦头结构,将分类和检测头分离,并从基于锚点的检测转变为无锚点的检测。

3.3　MobileNetV3 轻量化模型

MobileNetV3[20] 是一种轻量级卷积神经网络,具有良好的移动端适配性,可有效提升检测速度与准确性。MobileNetV3 利用神经架构搜索(Neural architecture search,NAS)设计深度神经网络结构,并通过 NetAdapt 算法优化卷积核通道数量,以在计算效率与模型性能之间取得最佳平衡。该网络包括 Small 和 Large 两个版本,选用 MobileNetV3 - Small 作为 YOLO v8n 模型特征提取主干网络,其核心网络结构 bneck 块如图 2 所示,能够有效提取输入图像中与任务相关的判别性特征,从而提升整体检测性能。

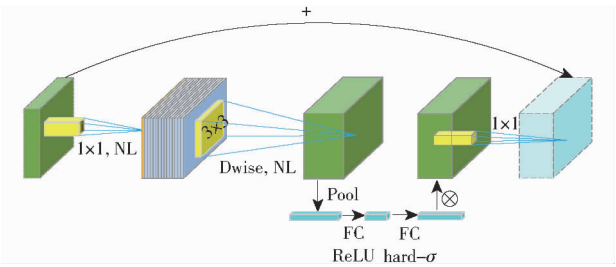


图 2　bneck 块网络结构图

Fig.2　bneck block network structure diagram

3.4　基于 GSConv 的 VoV - GSCSP 模块

GSConv[21] 是一种将标准卷积(Standard convolution,SC)、深度可分离卷积(Depthwise separable convolution,DSC)和 Shuffle 操作混合到一起的卷积,其网络结构如图 3 所示。在 GSConv 的基础上继续引入 GS bottleneck,采用聚合设计法设计跨阶段局部网络模块 VoV - GSCSP[22],其网络结构如图 4 所示。使用 VoV - GSCSP 代替 Neck 的第 15、18、21 层的 C2f,能够有效地提取浅层和深层网络特征信息,加以融合后使生成的输出特征图同时具备丰富的位置信息和语义信息。并且 VoV - GSCSP 结构在减少模型参数量与浮点运算量的同时,能够进一步提升模型检测精度。

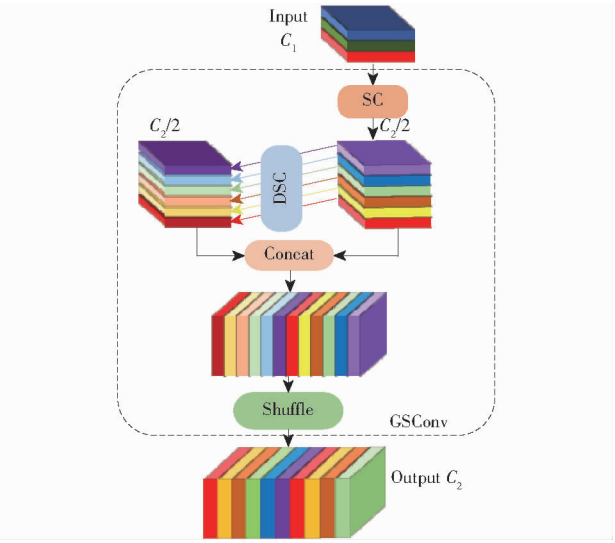


图 3 GSConv 网络结构图

Fig. 3 GSConv network structure diagram

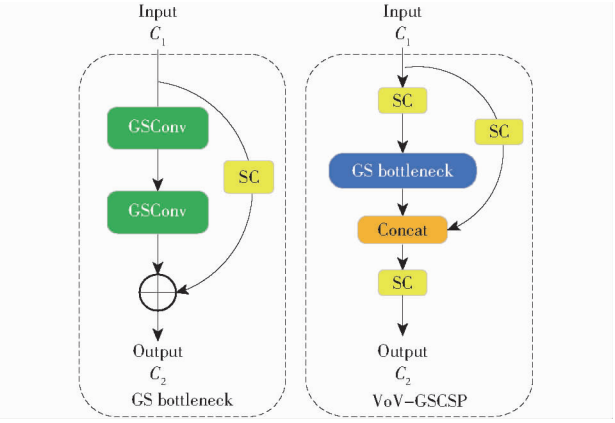


图 4 VoV - GSCSP 网络结构图

Fig. 4 VoV - GSCSP network structure diagram

3.5 重参数化的高效卷积网络 RepVGG

在搭建网络模型时,复杂的多分支模型虽然能提高准确率,但会降低推理速度、增加模型内存占用量,且难以在移动端部署。RepVGG[23]通过重参数化策略(Re-parameterization strategy,RPS)解决了这些问题,在训练阶段引入多分支模块以增强特征学习能力,推理阶段将这些模块合并为单一卷积结构,从而实现高效推理性能。RepVGG 网络结构如图 5 所示。图 5a 为原始 ResNet 结构,图 5b 为训练阶段 RepVGG 结构,图 5c 为推理阶段结构。

3.6 YOLO v8n - Plantago 网络模型

为提升车前草病虫害识别效果并适应移动端部署,对 YOLO v8n 模型进行了改进。替换 Backbone 为轻量化的 MobileNetV3,以减小模型内存占用量并提升响应速度;在 Neck 层集成 VoV - GSCSP 模块,并替换第 15、18、21 层的 C2f 模块,增强特征提取能力并降低计算成本;在 Head 部分引入 RepVGG 模块,利用重参数化加速推理过程,提高部署实用性,形成的 YOLO v8n - Plantago 模型结构如图 6 所示。

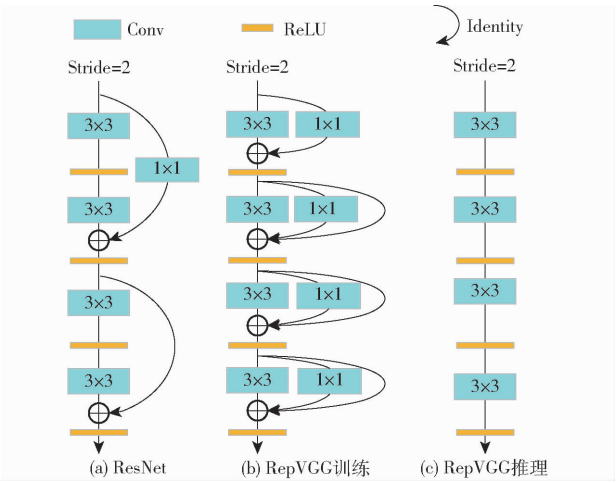


图 5 RepVGG 网络结构图

Fig. 5 RepVGG network structure diagram

4 结果与分析

4.1 模型检测效果

为验证 YOLO v8n - Plantago 模型在车前草病虫害识别中的性能,将其与原始 YOLO v8n 模型进行对比。各种类别的检测结果对比如表 3 所示,YOLO v8n - Plantago 模型在测试集上的 mAP@0.5 为 89.43%,比原始模型提升 2.04 个百分点。YOLO v8n - Plantago 在车前草虫害、车前草白粉病和车前草花叶病的 mAP@0.5 分别提升 7.02、0.87、1.23 个百分点,显示了较为明显的性能提升。

YOLO v8n - Plantago 与 YOLO v8n 在测试集上的部分检测结果对比如图 7 所示,蓝圈标注了原始 YOLO v8n 模型未能成功识别的病虫害,而 YOLO v8n - Plantago 模型能够准确检测到这些目标。蓝圈所示区域突显了改进模型在处理复杂背景和多目标重叠时的准确率提升。

4.2 不同轻量化 Backbone 对比

在 YOLO v8n 网络结构中使用轻量化 Backbone 可以有效降低模型内存占用量及参数量。不同 Backbone 结果对比如表 4 所示,在替换 ShuffleNetV2[24]、MobileNetV3、MobileNeXt[25]、GhostNetV2[26]和 EfficientNet[27]轻量级网络后,mAP@0.5都有所下降,然而这种下降的同时也带来了浮点运算量和模型内存占用量缩减,这表明随着网络深度减少,模型识别能力会下降。在不同 Backbone 结果对比中,MobileNetV3 模型效果最优,浮点运算量和模型内存占用量只有 5.72×10^9 和 4.77 MB,相比于原模型有显著下降;且 mAP@0.5 为 83.88%,相比于原模型下降较少,在检测方面具

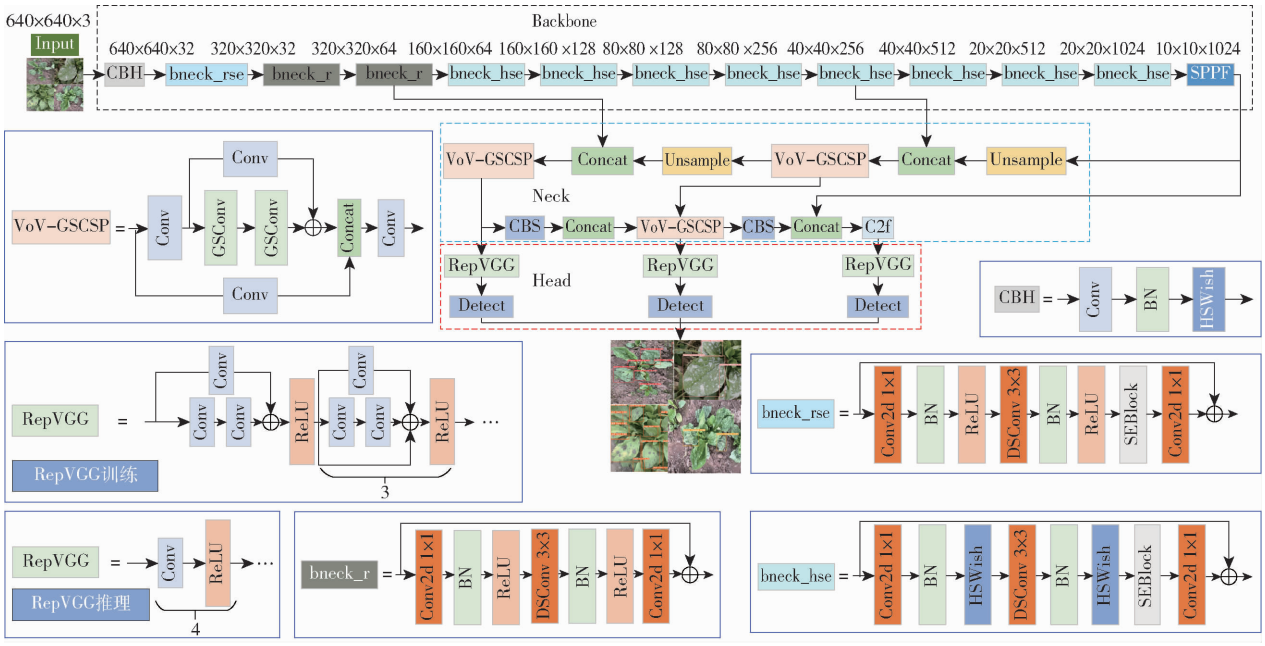


图 6　YOLO v8n－Plantago 网络结构图

Fig. 6　YOLO v8n－Plantago network structure diagram

表 3　YOLO v8n－Plantago 与 YOLO v8n 结果对比

Tab. 3　Comparison of results between YOLO v8n－Plantago and YOLO v8n

病虫害种类	mAP@0.5/%	
	YOLO v8n－Plantago	YOLO v8n
车前草虫害	83.15	76.13
车前草白粉病	94.23	93.36
车前草叶斑病	92.39	93.35
车前草花叶病	87.94	86.71
平均值	89.43	87.39

有优势，因此选择 MoileNetV3 作为主干网络。

4.3　Backbone 与 Neck 组合实验

采用 MobileNetV3 作为模型的 Backbone，并在 Neck 部分中引入 CARAFE[28]、ODConv[29]、VoV－GSCSP 和 EVC[30] 进行特征融合。实验结果表明，Neck 中特征融合的位置对模型性能具有显著影响。经过多次实验，确定了在 Neck 部分实现特征融合的最优方案：将 Neck 中的第 15、18、21 层的 C2f 模块替换为 VoV－GSCSP 模块以实现轻量化，同时保留第 24 层的 C2f 模块以充分学习高层语义特征。尽管 VoV－GSCSP 模块参数较少，难以支撑高层特征，但与保留的高层 C2f 模块结合后效果最佳，精确率为 79.56%、召回率为 80.52%、mAP@0.5 为 86.75%、浮点运算量为 5.07×10^9，模型内存占用量为 4.66 MB，不同 Neck 部分结果对比如表 5 所示。

4.4　Backbone、Neck 与特征增强模块组合实验

将 MobileNetV3 作为 Backbone，并在 Neck 中引入 VoV－GSCSP 模块，同时加入特征增强模块（Feature enhancement module，FEM）以提升检测精度。特征增强模块被置于Neck部分的第 18、19 层

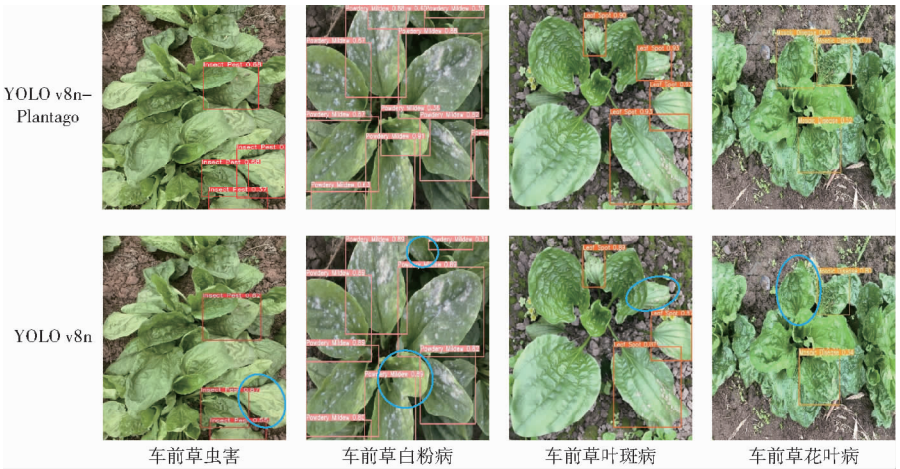


图 7　YOLO v8n－Plantago 与 YOLO v8n 检测结果对比

Fig. 7　Comparison of detection results between YOLO v8n－Plantago and YOLO v8n

表 4 不同 Backbone 结果对比

Tab. 4 Comparison results of different Backbones

模型	主干网络	精确率/%	召回率/%	mAP@0.5/%	浮点运算量	模型内存占用量/MB
YOLO v8n	CSPDarknet	83.42	81.60	87.39	8.09×10^9	5.95
	ShuffleNetV2	75.04	69.59	77.71	4.96×10^9	3.49
	MobileNetV3	79.69	78.80	83.88	5.72×10^9	4.77
	MobileNeXt	73.81	74.56	81.65	5.87×10^9	4.23
	GhostNetV2	74.41	84.08	82.91	7.90×10^9	7.80
	EfficientNet	76.94	81.89	82.74	5.65×10^9	3.93

表 5 不同 Neck 部分结果对比

Tab. 5 Comparison results of different Neck configurations

模型	颈部网络	精确率/%	召回率/%	mAP@0.5/%	浮点运算量	模型内存占用量/MB
YOLO v8n + MobileNetV3	①CARAFE	80.87	77.91	84.51	5.97×10^9	4.83
	②CARAFE	78.92	76.18	84.43	5.96×10^9	4.97
	③CARAFE	80.68	78.96	85.54	6.21×10^9	5.03
	④ODConv	79.76	77.69	84.10	5.53×10^9	4.82
	⑤ODConv	80.03	79.76	85.25	5.61×10^9	5.63
	⑥ODConv	80.43	80.04	85.99	5.49×10^9	5.85
	⑦VoV-GSCSP	77.52	80.07	83.76	4.85×10^9	4.42
	⑧VoV-GSCSP	79.56	80.52	86.75	5.07×10^9	4.66
	⑨EVC	80.79	76.47	83.73	1.91×10^{10}	13.00

注:①CARAFE 表示替换 Neck 层中的第 1 个 Unsample(改进模型的第 13 层);②CARAFE 表示替换 Neck 层中的第 2 个 Unsample(改进模型的第 16 层);③CARAFE 表示替换 Neck 层中的全部 Unsample(改进模型的第 13、16 层);④ODConv 表示替换 Neck 层中的第 1 个 CBS(改进模型的第 19 层);⑤ODConv 表示替换 Neck 层中的第 2 个 CBS(改进模型的第 22 层);⑥ODConv 表示替换 Neck 层中的全部 CBS(改进模型的第 19、22 层);⑦VoV-GSCSP 表示替换 Neck 层中的全部 C2f(改进模型的第 15、18、21、24 层);⑧VoV-GSCSP 表示替换 Neck 层中的第 1、2、3 个 C2f(改进模型的第 15、18、21 层);⑨EVC 表示插入第 1 个 Unsample 后(第 13 层后)。

之间和 Head 层,其中 Neck 部分的第 18、19 层之间适合增强中等目标特征,而 Head 层则有助于识别小目标的全局特征。实验对比了多种特征增强模块,包括基于卷积结构的模块(如 RepVGG、SCConv[31]、LSKNet[32])和基于注意力机制的模块(SimAM[33]、EMA[34]、GlobalContext[35]、TripletAttention[36]、Gather-Excite[37])。根据模块性质,确定最佳的添加位置,实现了模型性能的最优效果,不同特征增强模块组合的结果对比如表 6 所示。

表 6 不同 FEM 结果对比

Tab. 6 Comparison results of different FEMs

模型	特征增强模块	精确率/%	召回率/%	mAP@0.5/%	浮点运算量	模型内存占用量/MB
YOLO v8n + MobileNetV3 + VoV-GSCSP	②RepVGG	83.27	85.02	89.43	3.12×10^9	5.74
	①SimAM	82.07	79.79	85.10	5.07×10^9	4.67
	②SimAM	81.88	75.91	85.04	5.07×10^9	4.67
	①EMA	78.61	84.26	86.94	5.14×10^9	4.68
	②EMA	83.08	76.35	86.34	5.14×10^9	4.69
	①GlobalContext	81.85	82.36	87.43	5.43×10^9	5.10
	②GlobalContext	79.28	85.45	86.59	5.08×10^9	4.70
	①LSKNet	81.66	80.39	86.94	5.21×10^9	4.70
	②LSKNet	81.86	78.86	86.04	3.16×10^9	4.87
	①SCConv	76.03	79.81	82.82	5.52×10^9	5.11
	②SCConv	79.94	78.83	85.04	5.08×10^9	4.90
	①TripletAttention	81.96	76.98	85.12	5.11×10^9	4.67
	②TripletAttention	80.03	77.76	84.82	5.11×10^9	4.67
	①Gather-Excite	77.68	81.17	85.92	5.42×10^9	5.10
	②Gather-Excite	81.12	78.76	85.86	5.08×10^9	4.69

注:①X(X 为特征增强模块)表示添加的位置为改进模型的第 18、19 层之间;②X 表示添加的位置为 Head 层中。

由表 6 可知，这些模块显著影响精确率、召回率、mAP@0.5、浮点运算量和模型内存占用量。其中 RepVGG 的 mAP@0.5 最高，为 89.43%，浮点运算量较低，为 3.12×10^9，整体表现最优，适合边缘设备部署。由于 RepVGG 可优化推理速度，因此添加在 Head 层以提升高层特征处理效率。而 Neck 层主要负责特征融合和空间信息的保留，对灵活性和多尺度特征处理的要求较高，RepVGG 的固定结构不完全适用于此，因此未在 Neck 中使用。

4.5　消融实验

为验证 YOLO v8n－Plantago 模型的性能，设计消融实验，结果如表 7 所示。实验 1 与实验 2 对比，使用 MobileNetV3 替换原始 Backbone 后，浮点运算量从 8.09×10^9 降到 5.72×10^9，模型内存占用量从 5.95 MB 降到 4.77 MB，说明 MobileNetV3 能有效减少模型参数；实验 1 与实验 3、实验 4、以及实验 2 与实验 5、实验 6 的对比表明，VoV－GSCSP 和 RepVGG 模块能显著提高 mAP@0.5 并减少浮点运算量。尤其是实验 3 引入 VoV－GSCSP 后，mAP@0.5 从 87.39% 提升至 90.58%，浮点运算量降至 7.44×10^9；实验 4 使用 RepVGG 后，mAP@0.5 提升至 90.99%，浮点运算量降至 6.14×10^9；实验 7 同时引入 VoV－GSCSP 和 RepVGG 后，mAP@0.5 提升至 91.22%，浮点运算量降至 5.49×10^9。结果表明，替换 Backbone 为 MobileNetV3 能有效减小模型复杂度，而 VoV－GSCSP 和 RepVGG 模块进一步提升精度并减少运算复杂度。

4.6　轻量化模型改进比较

为比较 YOLO v8n－Plantago 与主流轻量化目标检测模型性能，在相同实验条件下对 YOLO v5n、YOLO v10n 和 YOLO 11n 进行了相同改进，实验结果如表 8 所示。YOLO v5n 在改进后，mAP@0.5 提高 1.61 个百分点，浮点运算量减少 2.1×10^9，模型内存占用量减少 0.29 MB，参数量下降 1.0×10^4。YOLO v10n 的 mAP@0.5 提升 0.44 个百分点，浮点运算量下降 5.71×10^9，参数量减少 1.2×10^5。YOLO 11n 的 mAP@0.5 提升 0.62 个百分点，浮点运算量减少 3.79×10^9，模型内存占用量减少 0.61 MB，参数量下降 2.9×10^5。上述结果表明，改进算法在轻量化目标检测模型中显著提升了计算效率并降低了模型复杂度。其中，YOLO v8n－Plantago 的综合性能表现最佳，充分验证了改进方法的优越性。

表 7　消融实验结果

Tab.7　Ablation experiments

实验序号	MobileNetV3	VoV－GSCSP	RepVGG	精确率/%	召回率/%	mAP@0.5/%	浮点运算量	模型内存占用量/MB
1	×	×	×	83.42	81.60	87.39	8.09×10^9	5.95
2	√	×	×	79.69	78.80	83.88	5.72×10^9	4.77
3	×	√	×	82.72	86.35	90.58	7.44×10^9	5.85
4	×	×	√	85.78	83.97	90.99	6.14×10^9	7.03
5	√	√	×	79.56	80.52	86.75	5.07×10^9	4.66
6	√	×	√	80.87	82.39	86.34	3.77×10^9	5.85
7	×	√	√	83.56	89.60	91.22	5.49×10^9	6.94
8	√	√	√	83.27	85.02	89.43	3.12×10^9	5.74

注：√表示采用此模块，×表示不采用此模块。

表 8　轻量化模型改进比较

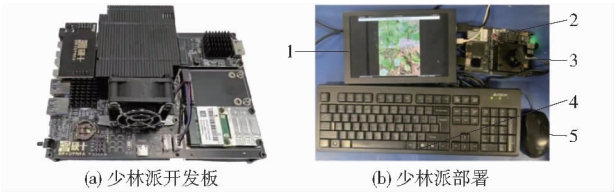
Tab.8　Comparison of improvements for lightweight models

模型	精确率/%	召回率/%	mAP@0.5/%	浮点运算量	模型内存占用量/MB	参数量
YOLO v5n	84.24	77.69	86.24	4.20×10^9	3.90	1.77×10^6
YOLO v5n－Plantago	85.38	81.16	87.85	2.10×10^9	3.61	1.76×10^6
YOLO v8n	83.42	81.60	87.39	8.09×10^9	5.95	3.01×10^6
YOLO v8n－Plantago	83.27	85.02	89.43	3.12×10^9	5.74	2.79×10^6
YOLO v10n	77.98	82.86	86.28	8.23×10^9	5.48	2.70×10^6
YOLO v10n－Plantago	82.54	79.66	86.72	2.52×10^9	5.48	2.58×10^6
YOLO 11n	82.55	82.19	86.61	6.32×10^9	5.32	2.58×10^6
YOLO 11n－Plantago	79.80	84.76	87.23	2.53×10^9	4.71	2.29×10^6

4.7　算法部署

为验证改进算法在实际应用中的可行性，选用基于国产人工智能芯片的边缘算力开发板——少林派。通过将 Pytorch 生成的预训练模型（.pt 文件）

转换为 ONNX 格式,少林派开发板就可高效运行深度学习模型。少林派开发板及部署如图 8 所示,性能比较如表 9 所示,检测结果如图 9 所示。



(a) 少林派开发板 (b) 少林派部署

图 8 少林派开发板及部署

Fig. 8 Shaolin school development version and deployment

1. 检测结果 2. 少林派 3. 摄像头 4. 键盘 5. 鼠标

表 9 改进前后模型部署性能对比

Tab. 9 Comparison of model deployment performance before and after improvement

模型	模型内存占用量/MB	推理速度/(幅·ms^{-1})	后处理速度/(幅·ms^{-1})
YOLO v8	5.95	18.82	3.96
YOLO v8 - Plantago	5.74	12.56	3.64

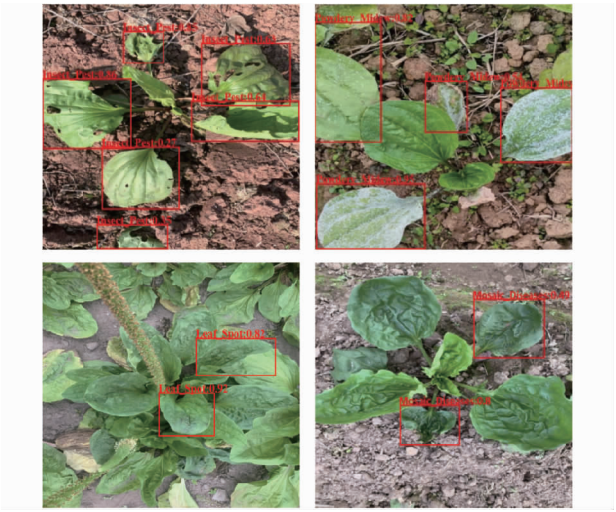


图 9 在少林派上检测结果

Fig. 9 Detection results on Shaolin pai

性能测试结果主要对比推理速度(即模型输出预测结果的过程)和后处理速度(即对检测框进行筛选与结果生成的过程),结果对比显示,YOLO v8n - Plantago 模型在边缘端设备上的推理速度提升了 33.26%,后处理速度提升了 8.08%。因此综合考虑模型内存占用量、速度和检测效果,改进模型满足实际应用对准确性和高效性的要求,具有较高应用价值。

5 结论

(1)针对自然条件下车前草病虫害识别问题,提出一种 YOLO v8n 模型的轻量化改进策略。在原始 YOLO v8n 架构基础上,利用 MobileNetV3 替换原始 Backbone,显著减少了模型参数量和计算复杂度,同时保持较高稳定性和速度。随后引入 VoV - GSCSP 模块来替代 Neck 的第 15、18、21 层的 C2f 模块,保持模型识别精度的同时降低计算成本。最后,在 Head 层引入 RepVGG 模块,通过训练时使用多分支卷积结构来提升模型的表现力,以简化计算和提高推理效率。通过以上改进,形成了轻量化的 YOLO v8n - Plantago 模型。

(2)通过在车前草病虫害数据集上对 YOLO v8n - Plantago 模型进行训练和测试,得到精确率、召回率、mAP@0.5、浮点运算量、模型内存占用量和参数量分别为 83.27%、85.02%、89.43%、3.12×10^{9}、5.74 MB 和 2.79×10^{6}。与 YOLO v8n 模型相比,精确率略微下降,但召回率提升 3.42 个百分点,mAP@0.5 提升 2.04 个百分点,浮点运算量下降 61.43%,模型内存占用量下降 3.53%,参数量下降 7.31%。与主流网络结构方法相比,该模型在识别精度和模型内存占用量上均取得更优表现。

(3)将 YOLO v8n - Plantago 模型部署在边缘端设备上,验证模型的应用价值。结果表明,推理速度比原模提升 33.26%,后处理速度比原模型提升 8.08%。因此,这些改进策略不仅简化了模型结构,而且有效提高模型的识别速度,为研制车前草病虫害自动识别与精准喷药装置奠定了基础。

参 考 文 献

[1] 刘娟, 夏伟, 袁泽焕, 等. 冬柏通淋合剂治疗慢性前列腺炎的临床观察[J]. 中华男科学杂志, 2024, 30(4): 336 - 341.
LIU Juan, XIA Wei, YUAN Zehuan, et al. Dongbai tonglin mixture for the treatment of chronic prostatitis[J]. National Journal of Andrology, 2024, 30(4): 336 - 341. (in Chinese)

[2] 刘鹏凯. 车前草穗枯病菌分子鉴定及野生车前草种质资源抗性筛选[D]. 南昌: 江西农业大学, 2011.
LIU Pengkai. Molecular identification of pathogen of Plantain head blight and resistance screening of germplasm of wild Plantain [D]. Nanchang: Jiangxi Agricultural University, 2011. (in Chinese)

[3] 张惠莉, 代晨龙, 任景龙, 等. 基于 YOLO v8 - GSGF 模型的葡萄病害识别方法研究[J]. 农业机械学报, 2024, 55(11): 75 - 83.
ZHANG Huili, DAI Chenlong, REN Jinglong, et al. Grape disease identification method based on YOLO v8 - GSGF model [J]. Transactions of the Chinese Society for Agricultural Machinery, 2024, 55(11): 75 - 83. (in Chinese)

[4] 袁杰, 谢霖伟, 郭旭, 等. 基于改进 YOLO v7 的苹果叶片病害检测方法[J]. 农业机械学报, 2024, 55(11): 68 - 74.
YUAN Jie, XIE Linwei, GUO Xu, et al. Apple leaf disease detection method based on improved YOLO v7[J]. Transactions of the Chinese Society for Agricultural Machinery, 2024, 55(11): 68 - 74. (in Chinese)

[5] LI R, LI Y, QIN W, et al. Lightweight network for corn leaf disease identification based on improved YOLO v8s[J]. Agriculture, 2024, 14(2): 220.
[6] 时雷, 杨程凯, 雷镜楷, 等. 基于改进 YOLO v8s 的小麦小穗赤霉病检测研究[J]. 农业机械学报, 2024, 55(7): 280 – 289.
SHI Lei, YANG Chengkai, LEI Jingkai, et al. Wheat spikelet detection of fusarium head blight based on improved YOLO v8s [J]. Transactions of the Chinese Society for Agricultural Machinery, 2024, 55(7): 280 – 289. (in Chinese)
[7] CHEN Z, WU R, LIN Y, et al. Plant disease recognition model based on improved YOLO v5[J]. Agronomy, 2022, 12(2): 365.
[8] 廖娟, 刘凯旋, 杨玉青, 等. 基于 RDN – YOLO 的自然环境下水稻病害识别模型研究[J]. 农业机械学报, 2024, 55(8): 233 – 242.
LIAO Juan, LIU Kaixuan, YANG Yuqing, et al. Rice disease recognition in natural environment based on RDN – YOLO[J]. Transactions of the Chinese Society for Agricultural Machinery, 2024, 55(8): 233 – 242. (in Chinese)
[9] 左昊轩, 黄祺成, 杨佳昊, 等. 基于改进 YOLO v5s 的作物黄化曲叶病检测方法[J]. 农业机械学报, 2023, 54(增刊 1): 230 – 238.
ZUO Haoxuan, HUANG Qicheng, YANG Jiahao, et al. Improved YOLO v5s-based detection method for crop yellow leaf curl virus disease[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(Supp. 1): 230 – 238. (in Chinese)
[10] 曾建军, 欧阳群, 李晓红, 等. 车前不定芽直接诱导及再生体系的建立[J]. 安徽农业科学, 2008(5): 1810 – 1811.
ZENG Jianjun, OUYANG Qun, LI Xiaohong, et al. Direct induction of adventitious buds and establishment of regeneration system in *Plantago asiatica* L.[J]. Anhui Agricultural Sciences, 2008(5): 1810 – 1811. (in Chinese)
[11] 张庆琛, 裴冬丽, 丁锦平, 等. 车前草白粉病的病原菌鉴定[J]. 贵州农业科学, 2012, 40(9): 106 – 108.
ZHANG Qingchen, PEI Dongli, DING Jinping, et al. Identification of *Plantago asiatica* powdery mildew[J]. Guizhou Agricultural Sciences, 2012, 40(9): 106 – 108. (in Chinese)
[12] 严吉明, 叶华智, 秦芸, 等. 四川药用植物病害调查与病原鉴定Ⅱ. 四川药用植物病害[J]. 西南农业学报, 2008, 21(2): 359 – 363.
YAN Jiming, YE Huazhi, QIN Yun, et al. Investigation on the disease of medicinal plants in Sichuan Province and pathogen identification Ⅱ. A list of medicinal lant diseases[J]. Southwest China Journal of Agricultural Sciences, 2008, 21(2): 359 – 363. (in Chinese)
[13] 刘博, 付海滨, 李鑫, 等. 车前草花叶病毒对我国百合产业的风险评估[J]. 中国植保导刊, 2023, 43(9): 102 – 105.
LIU Bo, FU Haibin, LI Xin, et al. Risk assessment of introduction of *Plantago asiatica* mosaic virus on lily industry in China [J]. China Plant Protection, 2023, 43(9): 102 – 105. (in Chinese)
[14] LI C, LI L, JIANG H, et al. YOLOv6: a single-stage object detection framework for industrial applications[J]. arXiv Preprint, arXiv:2209.02976, 2022.
[15] WANG C Y, BOCHKOVSKIY A, LIAO H Y M. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, Canada: IEEE, 2023: 7464 – 7475.
[16] VARGHESE R, SAMBATH M. YOLOv8: a novel object detection algorithm with enhanced performance and robustness[C]// 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS). Chennai, India: IEEE, 2024: 1 – 6.
[17] WANG C Y, YEH I H, LIAO H Y M. Yolov9: learning what you want to learn using programmable gradient information[J]. arXiv Preprint, arXiv: 2402.13616, 2024.
[18] WANG A, CHEN H, LIU L, et al. Yolov10: real-time end-to-end object detection[J]. arXiv Preprint, arXiv: 2405.14458, 2024.
[19] KHANAM R, HUSSAIN M. Yolov11: an overview of the key architectural enhancements[J]. arXiv Preprint, arXiv:2410.17725, 2024.
[20] HOWARD A, SANDLER M, CHU G, et al. Searching for Mobilenetv3[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 1314 – 1324.
[21] LI H, LI J, WEI H, et al. Slim-neck by GSConv: a lightweight-design for real-time detector architectures[J]. Journal of Real-Time Image Processing, 2024, 21(3): 62.
[22] 刘晓文, 曾雪婷, 李涛, 等. 基于改进 YOLO v7 的生猪群体体温自动检测方法[J]. 农业机械学报, 2023, 54(增刊 1): 267 – 274.
LIU Xiaowen, ZENG Xueting, LI Tao, et al. Automatic detection method of body temperature in herd of pigs based on improved YOLO v7[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(Supp. 1): 267 – 274. (in Chinese)
[23] DING X, ZHANG X, MA N, et al. Repvgg: making vgg-style convnets great again[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ: IEEE, 2021: 13733 – 13742.
[24] MA N, ZHANG X, ZHENG H T, et al. Shufflenetv2: practical guidelines for efficient cnn architecture design[C]// Proceedings of the European Conference on Computer Vision (ECCV). Munich, Germany: IEEE, 2018: 116 – 131.
[25] ZHOU D, HOU Q, CHEN Y, et al. Rethinking bottleneck structure for efficient mobile network design[C]// Computer Vision-ECCV 2020: 16th European Conference. Springer International Publishing, 2020: 680 – 697.
[26] TANG Y, HAN K, GUO J, et al. GhostNetv2: enhance cheap operation with long-range attention[J]. arXiv Preprint, arXiv:

2211.12905, 2022.
[27] TAN M. Efficientnet: rethinking model scaling for convolutional neural networks[C]// Proceedings of the 36th International Conference on Machine Learning. Long Beach, USA: PMLR, 2019: 6105 - 6114.
[28] WANG J, CHEN K, XU R, et al. Carafe: content-aware reassembly of features[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, South Korea: IEEE, 2019: 3007 - 3016.
[29] LI C, ZHOU A, YAO A. Omni-dimensional dynamic convolution[J]. arXiv Preprint, arXiv:2209.07947, 2022.
[30] QUAN Y, ZHANG D, ZHANG L, et al. Centralized feature pyramid for object detection[J]. IEEE Transactions on Image Processing, 2023, 32: 4341 - 4354.
[31] LI J, WEN Y, HE L. Scconv: spatial and channel reconstruction convolution for feature redundancy[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver, Canada: IEEE, 2023: 6153 - 6162.
[32] LI Y, HOU Q, ZHENG Z, et al. Large selective kernel network for remote sensing object detection[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. Paris, France: IEEE, 2023: 16794 - 16805.
[33] YANG L, ZHANG R Y, LI L, et al. Simam: a simple, parameter-free attention module for convolutional neural networks[C]// Proceedings of the 38th International Conference on Machine Learning, PMLR, 2021: 11863 - 11874.
[34] OUYANG D, HE S, ZHANG G, et al. Efficient multi-scale attention module with cross-spatial learning[C]// ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway, NJ: IEEE, 2023: 1 - 5.
[35] CAO Y, XU J, LIN S, et al. Gcnet: non-local networks meet squeeze-excitation networks and beyond[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. Seoul, South Korea: IEEE, 2019: 1971 - 1980.
[36] MISRA D, NALAMADA T, ARASANIPALAI A U, et al. Rotate to attend: convolutional triplet attention module[C]// Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Online Virtual Conference, 2021: 3139 - 3148.
[37] HU J, SUN G, LI X, et al. Gather-excite: exploiting feature context in convolutional neural networks[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS). Montreal, Canada, 2018: 9401 - 9411.

(上接第 202 页)

[19] 杨鸣娟, 张坤生. 番茄品种、成熟度和热处理对番茄中番茄红素含量的影响[J]. 食品研究与开发, 2003(3): 99 - 101.
YANG Mingjuan, ZhANG Kunsheng. Effects of tomato variety, maturity and heat treatment on lycopene content in tomatoes [J]. Food Research and Development, 2003(3):99 - 101. (in Chinese)
[20] 庞淑艳. 关于番茄果实颜色的主要研究进展[J]. 现代农业, 2015(2):119 - 120.
PANG Shuyan. Major research progress on tomato fruit color[J]. Modern Agriculture, 2015(2):119 - 120. (in Chinese)
[21] 陈文静. 番茄果实成熟过程中番茄红素含量变化及关键基因的表达分析[D]. 石家庄:河北科技大学, 2020.
CHEN Wenjing. Changes in lycopene content and key gene expression during tomato fruit ripening[D]. Shijiazhuang:Hebei University of Science and Technology, 2020. (in Chinese)
[22] 袁奋杰, 周晓, 丁军, 等. 基于 FPGA 的 RGB 和 HSV 色空间转换算法实现[J]. 电子器件, 2010, 33(4): 493 - 497.
YUAN Fenjie, ZHOU Xiao, DING Jun, et al. Implementation of RGB and HSV color space conversion algorithm based on FPGA[J]. Electronic Devices, 2010, 33(4):493 - 497. (in Chinese)
[23] 马玲, 张晓辉. HSV 颜色空间的饱和度与明度关系模型[J]. 计算机辅助设计与图形学学报, 2014, 26(8):1272 - 1278.
MA Ling, ZHANG Xiaohui. Relationship model of saturation and brightness in HSV color space[J]. Journal of Computer-Aided Design and Computer Graphics, 2014, 26(8):1272 - 1278. (in Chinese)
[24] 陶华伟, 赵力, 奚吉, 等. 基于颜色及纹理特征的果蔬种类识别方法[J]. 农业工程学报, 2014, 30(16): 305 - 311.
TAO Huawei, ZHAO Li, XI Ji, et al. Fruit and vegetable classification method based on color and texture features[J]. Transactions of the CSAE, 2014, 30(16):305 - 311. (in Chinese)
[25] 李天华, 孙萌, 丁小明, 等. 基于 YOLO v4 + HSV 的成熟期番茄识别方法[J]. 农业工程学报, 2021, 37(21): 183 - 190.
LI Tianhua, SUN Meng, DING Xiaoming, et al. Tomato maturity recognition method based on YOLO v4 + HSV[J]. Transactions of the CSAE, 2021, 37(21):183 - 190. (in Chinese)
[26] ZHENG Z, WANG P, LIU W, et al. Distance - IoU loss: faster and better learning for bounding box regression[C]// Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 12993 - 13000.
[27] LI X, WANG W, WU L, et al. Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection[J]. Advances in Neural Information Processing Systems, 2020, 33: 21002 - 21012.
[28] GEVORGYAN Z. SIoU loss: more powerful learning for bounding box regression[J]. arXiv preprint arXiv:2205.12740, 2022.
[29] 王晓明,陈智宇,董文涛,等. 基于 YOLO v7x 的接触网吊弦缺陷检测方法[J]. 华东交通大学学报, 2024,41(3):65 - 73.
WANG Xiaoming, CHEN Zhiyu, DONG Wentao, et al. Catenary dropper defect detection method based on YOLO v7x[J]. Journal of East China Jiaotong University, 2024,41(3):65 - 73. (in Chinese)
[30] ZHANG Y F, REN W, ZHANG Z, et al. Focal and efficient IoU loss for accurate bounding box regression[J]. Neurocomputing, 2022, 506: 146 - 157.