

基于 YOLOv8 改进的轻量化接线端子姿态检测算法

秦 健^{1*}

QIN Jian

摘 要

为提升工业机器人在接线端子姿态识别与定位任务中的检测精度与推理效率,文章提出一种基于 YOLOv8n 模型结构优化的轻量化改进方法。首先,将原主干网络中的 C2f 模块替换为计算量更低的 C2f-faster 模块,在保持特征提取能力的同时有效降低了参数规模与运算开销;其次,在特征提取阶段引入 CGD (context guided down-sampling module) 模块,替代部分卷积 (Conv) 模块,增强了模型对局部与上下文信息的感知能力,提升复杂场景下的识别鲁棒性;最后,采用 SIoU 损失函数优化边界框回归过程,提升目标定位精度。在基于自建数据集的实验中,改进模型在保持轻量化和低资源消耗的基础上,实现了 99.4% 的 mAP@0.5,推理速度比未改进前提升约 17.8%。研究结果表明,所提方法在接线端子的高效识别方面具有良好的应用前景。

关键词

接线端子识别; YOLOv8; 轻量化; C2f-faster; CGD; SIoU

doi: 10.3969/j.issn.1672-9528.2025.12.045

0 引言

在智能制造背景下,接线端子的识别与抓取精度直接影响产线效率。传统人工检测效率低、误判率高,模板匹配方法受光照与姿态变化影响大,难以适应生产对高速节拍的要求。为此,本文提出一种基于 YOLOv8 的端子姿态识别方法,结合结构光双目视觉获取三维位姿,提升复杂背景下的识别鲁棒性和精度,判断不同姿态并为机器人路径规划提供可靠依据。Chen 等^[1]基于 PConv 进一步提出了 FasterNet,可以更高效地提取空间特征;Wu 等^[2]首次提出了 CG (Context Guided) 模块结构,有效地提高语义分割任务中轻量级网络对边界与细节的捕捉能力和整体表现;Lü 等^[3]将 YOLOv8 的下采样层替换为 CGD 模块,实现对图像特征的更完整保留;SIoU (SCYLLA-IoU) 损失函数由 Gevorgyan^[4]提出,通过引入方向匹配机制,解决了传统回归框“游荡”问题,在提升训练稳定性的同时显著增强了目标检测精度。尽管 FasterNet 与 CG 模块在特征提取与细节感知方面取得了显著进展,但二者尚未在统一框架中协同优化,难以兼顾检测精度与计算效率。为此,本研究在 YOLOv8n 主干中引入 C2f-faster 与 CGD 模块,并结合 SIoU 损失函数,构建出一套更适用于复杂工业场景下接线端子及其姿态的高效识别和定位方案,在保证轻量化的同时显著提升了边界感知能力与定位精度。

1 YOLOv8 模型

YOLOv8 是一款兼具高效性与精细化的实例分割模型,核心目标在于实现快速精准的目标检测。作为 YOLOv5 的重要升级,YOLOv8 具备更优的检测性能与泛化能力,广泛应用于工业检测、安防监控和自动驾驶等领域^[5]。

本课题旨在实现传送带运行过程中接线端子的实时姿态识别,选用轻量级 YOLOv8n 作为基础检测网络。该模型推理速度快、参数量少,便于部署于工业抓取等实时性强的任务中。YOLOv8n 网络结构由输入端、骨干网络、特征融合模块和检测头四部分组成^[6]。其中,输入端通过 Mosaic 增强和自适应缩放提升模型鲁棒性;Backbone 采用 Conv、C2f 与 SPPF 等模块提取图像特征;Neck 通过改进的 PANet 实现多尺度特征融合,增强对不同尺寸和复杂背景目标的感知;Head 则利用多尺度输出机制,提升模型在尺寸差异明显目标下的预测能力与稳定性。

2 改进的 YOLOv8n 模型

现有深度学习方法在接线端子姿态识别中,常因模型庞大、内存占用高,难以满足工业机器人等边缘设备的部署需求。为此,本文提出一种基于 YOLOv8n 结构优化的轻量化接线端子姿态识别算法。该方法在有效降低模型计算资源需求、提升推理速度的同时,尽可能保持识别精度,满足工业场景中对实时性与准确性的双重要求,改进后的算法结构如图 1 所示。

1. 南京工程学院机械工程学院 江苏南京 211167

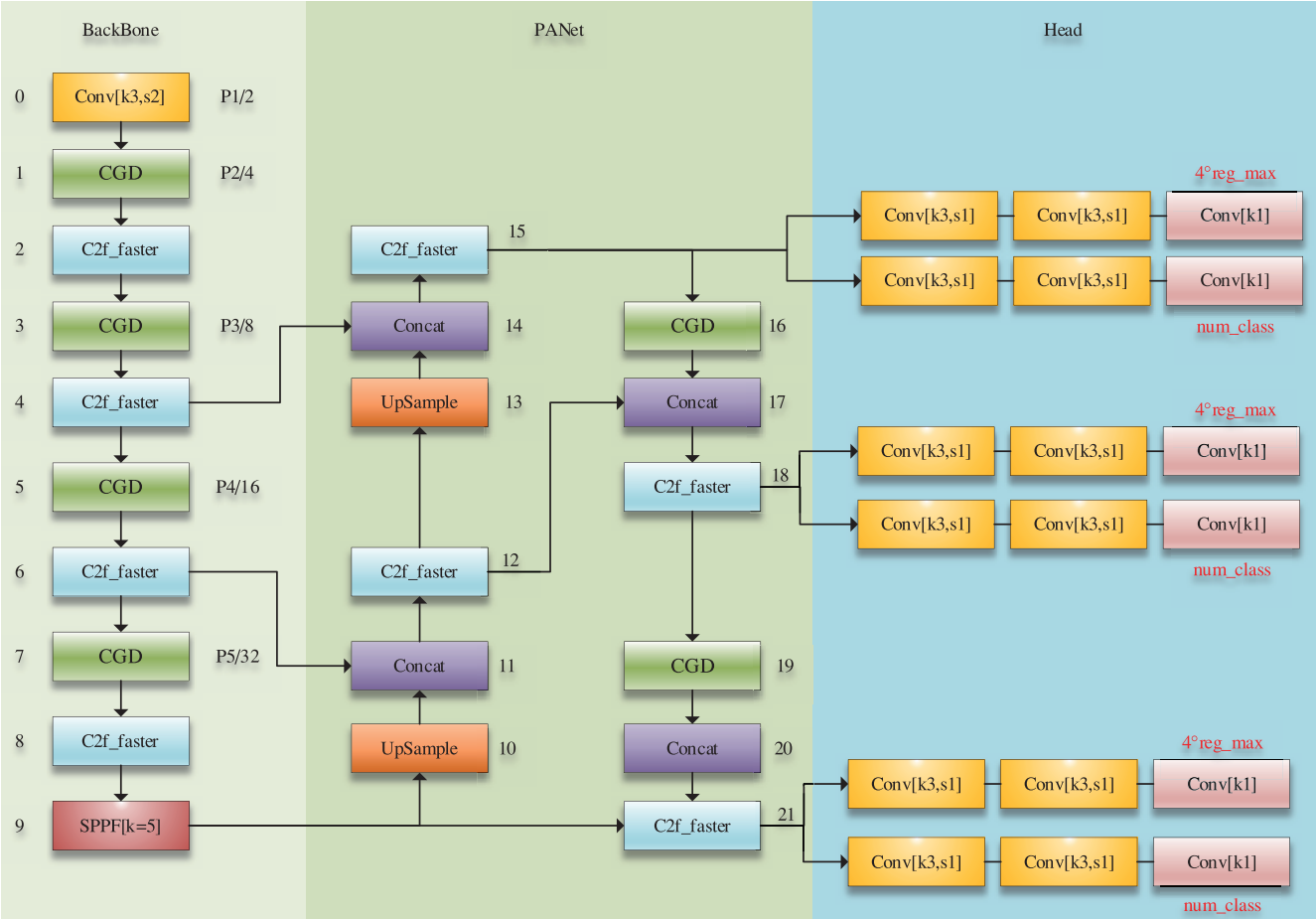


图1 改进后算法的网络结构图

具体工作如下：

- (1) 引入基于 PConv 进一步提出的轻量级主干网络 FasterNet，将原 C2f 模块中的 Bottleneck 替换为 FasterNet-Block，构建出新的 C2f-faster 模块；
- (2) 在特征提取阶段引入 CGD 模块，增强了网络对全图全局和上下文之间的联系；
- (3) 引入 SIoU 损失函数优化边界框回归过程，提高了目标定位的准确性。

2.1 C2f-faster

针对实际工业场景对模型实时性与部署效率的高要求，许多研究者提出了多种轻量化神经网络结构，如 MobileNets^[7]，ShuffleNets^[8] 或 Ghost-Net^[9] 等。这些网络大多通过引入深度可分离卷积或组卷积等方式来降低模型的计算量。然而，这类操作虽然有效减少了浮点运算次数（FLOPs），但由于频繁的内存访问，实际计算效率反而可能下降，影响推理速度，未能完全达到真正的轻量化目的。相比之下，PConv 结构则在保持模型精度的前提下，仅对部分输入通道进行标准卷积操作，显著降低计算量的同时也提升了执行效率，适用于本研究中的接线端子抓取或对响应速度要求较高

的任务中，PConv 的工作原理如图 2 所示。

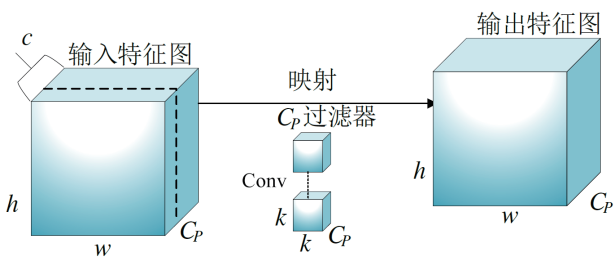


图2 PConv 工作原理图

图2中为识别模型中卷积特征提取模块结构示意图，输入特征图尺寸为 $w \times h \times C_p$ ，其中 w 、 h 、 C_p 分别表示宽度、高度和通道数^[10]；中间卷积核尺寸为 $k \times k \times C_p$ ，用于提取局部空间和通道特征；卷积操作后生成输出特征图，尺寸仍为 $w \times h \times C_p$ ，保持空间分辨率不变。该过程有助于增强模型对接线端子边缘与结构的感知能力，为后续姿态识别与抓取点提取提供基础支持。

相较于 C2f 模块中采用传统卷积的 Bottleneck 结构，C2f-faster 模块引入了计算更高效的 FasterNetBlock，采用 PConv 进行特征提取，显著降低了计算量与内存访问次数。通过将原 YOLOv8n 中的 Bottleneck 替换为 FasterNetBlock，

构建新的 C2f-faster 模块, 并替代原网络中的 C2f 模块, 增强了特征融合能力, 减少了模型的参数数量和计算复杂度, 其结构如图 3 所示。

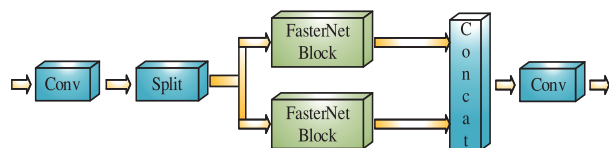


图 3 C2f-faster 网络结构图

2.2 用 CGD 模块替换 Conv 模块

主干网络中以 CGD 替代部分标准卷积结构, 从而增强模型对边界与细节的感知能力。CGD 模块首先通过 1×1 卷积压缩输入特征, 随后通过局部特征提取器 (LFE) 和周围上下文提取器 (SCE) 并行提取信息, 其中 LFE 利用标准卷积提取细节, SCE 采用扩张卷积扩展感受野以捕捉更丰富的上下文。联合特征提取器 (JFE) 对上述特征进行融合, 并经 BN 和 ReLU 激活以增强表达能力。全局上下文提取器 (GCE) 则通过全局平均池化与全连接网络生成通道注意力权重, 强化关键区域、抑制冗余干扰。该模块实现了对局部与全局特征的有效结合, 在复杂背景或目标形变情况下, 依然保持良好的识别鲁棒性与精度, 展现出轻量化与高性能的优良平衡, 其结构如图 4 所示。

2.3 引入 SIoU 边界框回归损失函数

传统目标检测算法多采用 GIoU、DIOU 或 CIOU 等回归损失函数优化目标框位置, 但这类方法仅考虑边界框间的重叠面积或中心点距离, 未充分利用框之间的方向、角度与形状信息, 限制了定位精度的进一步提升。通过在回归分支中引入 SIoU 损失函数, 该方法综合考虑预测框与真实框之间的距离、重叠面积、宽高比和角度方向等几何因素, 通过引入角度损失项强化对偏斜目标的定位约束, 特别适用于姿态变化明显的工件检测任务。引入 SIoU 后, 来加速两个预测

框之间距离的收敛, 从而减少推理时间和计算成本^[4], 其结构如图 5 所示。

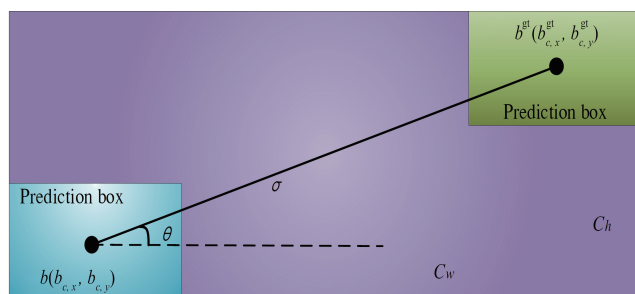


图 5 SIoU 参数示意图

图 5 中, $b(b_{c,x}, b_{c,y})$ 与 $b^g(b_{c,x}^g, b_{c,y}^g)$ 分别表示预测框和真实框的中心点的坐标; σ 表示两框中心的距离; w 和 h 分别表示预测框与真实框最小包围框的宽和高; θ 表示两框中心连线与水平方向的夹角。

3 数据集的制作及划分

本研究使用 Intel RealSense D435 双目相机采集 3 种姿态下的接线端子图像, 将分辨率统一为 $640 \text{ px} \times 480 \text{ px}$, 并采用旋转、遮挡等方法扩增至 2 703 张, 按 7:2:1 划分训练集、测试集与验证集, 使用 LabelImg 工具完成数据标注, 输出格式为 YOLO, 分为 6 个类别: “pose1、pose2 和 pose3” 表示接线端子为正反面姿态; “pose1-p、pose2-p 和 pose3-p” 表示对应姿态下的抓取框, 其标签如图 6 所示。

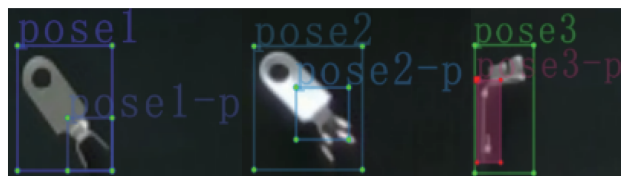


图 6 三种姿态数据样本图

4 实验及结果分析

实验平台采用 Windows 11 操作系统, 搭载 Intel(R)

Core(TM)i5-13500HX2.50 GHz 处理器, 构建了基于 GPU 的 PyTorch 环境用于 YOLOv8 模型训练。图 7 展示了训练数据的整体分布情况: (a) 为样本数量分布; (b) 为标注框的大小与数量统计; (c) 为目标中心在图像中的位置分布; (d) 为目标在图像中所占的相对高宽比例。整体来看, 样本类别较为均衡, 但目标框大小存在较大差异。

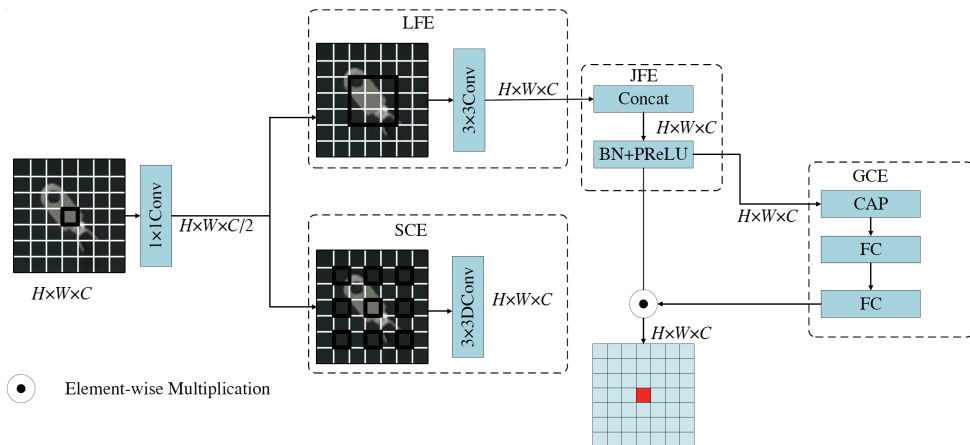


图 4 上下文引导下采样模块的总体框架

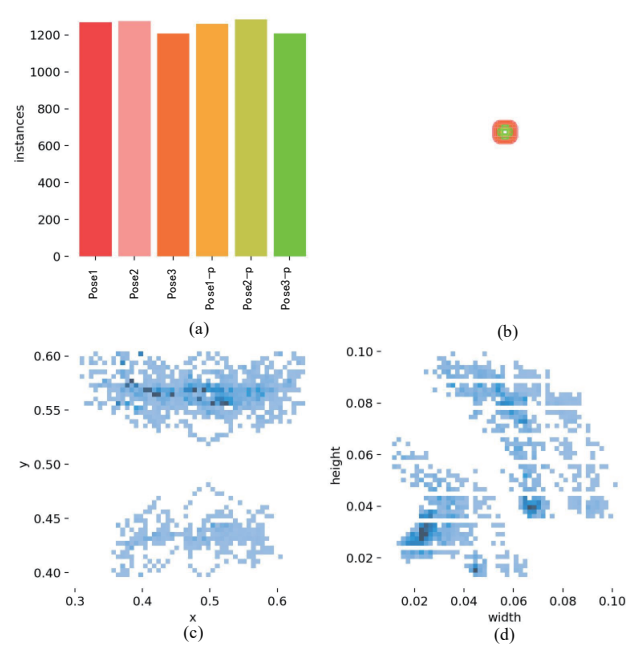


图 7 数据集分布情况

4.1 改进算法消融实验

通过在 YOLOv8n 模型基础上依次引入 3 项改进：（1）将 C2f-faster 模块替换原有的 C2f 模块；（2）使用 CGD 模块代替 Conv 模块；（3）并在损失函数层面，在回归分支中引入 SIoU 损失函数。为验证各改进组件的有效性，设计逐步添加改进项的消融实验，并与完整改进模型进行对比，实验结果如表 1 所示。

表 1 改进算法消融实验结果表

	C2f-faster	CGD	SIoU	Precision/%	Recall/%	mAP@0.5/%
1	×	×	×	95.6	95.6	97.1
2	√	×	×	98.3	97.1	98.2
3	√	√	×	99.5	98.5	99.3
4	√	×	√	98.0	98.2	98.4
5	√	√	√	99.2	99.3	99.4

根据表 1 改进算法消融实验结果可知，逐步引入各模块对模型性能产生了明显影响。将原始 YOLOv8n 结构中的 C2f 模块替换为轻量化的 C2f-faster 后，模型的 Precision 和 Recall 均有明显提升，mAP@0.5 从 97.1% 上升至 98.2%，说明 C2f-faster 模块在降低模型复杂度的同时保持了较高的检测性能。在此基础上引入 CGD 模块，进一步提升了模型的检测精度，mAP@0.5 达到 99.3%，表明该模块对目标特征提取具有积极作用。然而在加入 C2f-faster 模块和不加入 CGD 模块的情况下加入 SIoU 损失函数后，mAP@0.5 下降了 0.3%，说明在当前结构下该损失函数的引入未带来预期的效果。最终集成 C2f-faster、CGD 模块和 SIoU 模块后，模型达到 99.4%

的 mAP@0.5，性能最优。整体来看，本研究提出的改进策略在保持较高精度的同时，实现了模型结构的轻量化，验证了其在工业检测任务中的实用价值。综上所述，在保持检测精度基本不变的前提下，改进算法有效实现了模型的轻量化。

4.2 YOLOv8n 检测模型训练

通过使用 YOLOv8n 的配置文件，设置的参数如下：输入图片尺寸为 640 px×480 px，训练轮次为 200，batch-size 为 16，Workers 为 0。从图 8 中的 mAP@0.5 曲线可以看出，模型在第 200 轮时已趋于收敛，平均 mAP@0.5 达到 97.1%。部分检测后的图像展示在图 9 中。

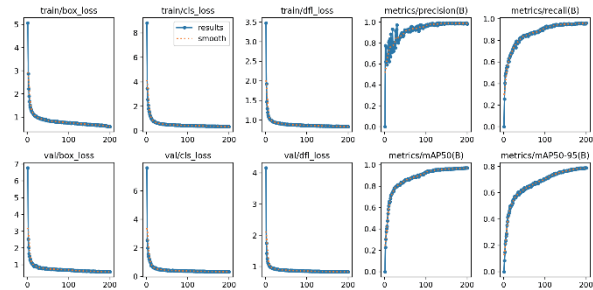


图 8 训练各值曲线图



图 9 YOLOv8 n 部分检测后的图像

4.3 改进 YOLOv8n 检测模型训练

改进后的 YOLOv8n 检测模型在与原模型相同的训练设置下进行训练。从图 10 可见，mAP@0.5 在约 200 轮时趋于稳定，最终达到 99.4%，改进前后模型的 mAP@0.5 值随着迭代次数变化的曲线如图 11 示。部分检测后的图像展示在图 12 中。

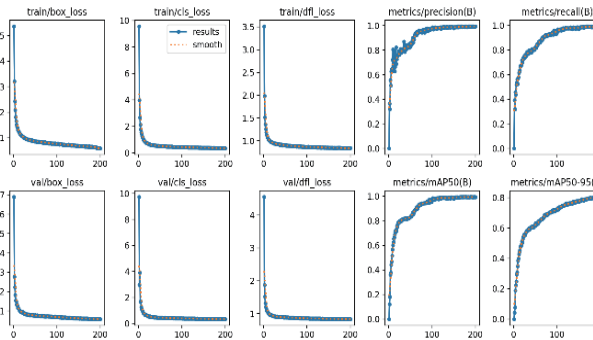


图 10 训练各值曲线图

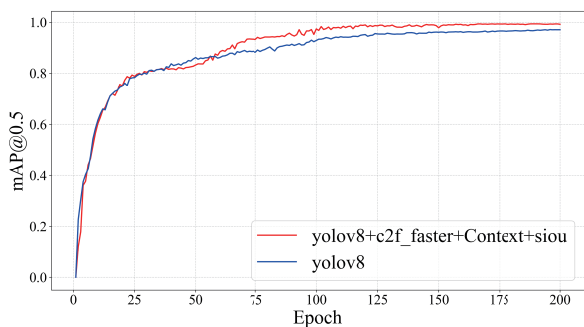


图 11 算法改进前后训练过程中 mAP@0.5 变化对比图



图 12 改进后 YOLOv8n 部分检测后的图像

4.4 不同算法对比实验

为验证本算法的有效性,表 2 对比实验结果表明,本文方法在各项指标上均优于 YOLOv8n 和 YOLOv10n。具体而言,本方法的召回率达到 99.3%,明显高于 YOLOv8n 的 95.6% 和 YOLOv10n 的 93.9%; mAP@0.5 达到 99.4%,同样优于 YOLOv8n 的 97.1% 和 YOLOv10n 的 98.2%。此外,本文方法的检测速度(帧/s)为 159.88,远高于 YOLOv8n 的 135.68 和 YOLOv10n 的 132.60,而 GFLOPs 仅为 7.3,低于 YOLOv8n 的 8.1 和 YOLOv10n 的 8.2。综上所述,所提方法在检测准确率和效率方面均表现出明显的综合性能优势,实现了更高精度与更快速度的良好平衡。

表 2 不同算法对比实验结果表

模型	Recall/%	mAP@0.5/%	检测速度 (帧·s ⁻¹)	GFLOPs
YOLOv8n	95.6	97.1	135.68	8.1
YOLOv10n	93.9	98.2	132.60	8.2
Ours	99.3	99.4	159.88	7.3

5 结语

通过围绕工业场景中接线端子及其姿态的高效识别和定位,提出了针对 YOLOv8n 模型的结构改进方法。在保持模型检测精度的基础上,通过采用 C2f-faster 模块实现主干网络的轻量化设计,利用 CGD 模块替代常规卷积操作强化上下文特征提取,并引入 SIoU 损失函数优化边界框回归策略,从而在准确性与实时性之间取得了良好平衡。实验结果表明,

该改进模型在多项关键指标上均优于原始 YOLOv8n,在资源受限的工业部署环境中具有更强的实用性和部署适应性。

参考文献:

- [1]CHEN J R, 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. Piscataway: IEEE, 2023: 12021-12031.
- [2]WU P Y, ZHU K, LIU Y, et al. Improved video VAE for latent video diffusion model[C]//Proceedings of the Computer Vision and Pattern Recognition Conference. Piscataway:IEEE,2025: 18124-18133.
- [3]LÜ K W, WU R B, CHEN S Y, et al. CCI-YOLOv8n: enhanced fire detection with CARAFE and context-guided modules[EB/OL]. (2025-07-04)[2025-11-10].<https://doi.org/10.48550/arXiv.2411.11011>.
- [4]GEVORGYAN Z. SIoU Loss: more powerful learning for bounding box regression[EB/OL].(2022-05-25)[2025-11-03].<https://doi.org/10.48550/arXiv.2205.12740>.
- [5]陈爽. 基于目标检测与跟踪的机动车实时速度检测算法研究[D]. 哈尔滨: 东北林业大学, 2024.
- [6]王宏, 韩晨, 袁伯阳, 等. 基于改进 YOLOv5 的拥挤行人检测算法[J]. 科学技术与工程, 2023, 23(27): 11730-11738.
- [7]HOWARD A G, ZHU M L, CHEN B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017-04-17)[2025-10-25].<https://doi.org/10.48550/arXiv.1704.04861>.
- [8]MA N N, ZHANG X Y, ZHENG H T, et al. ShuffleNet V2: practical guidelines for efficient CNN architecture design[C]// Proceedings of the European conference on computer vision (ECCV). Berlin:Springer, 2018: 116-131.
- [9]HAN K, WANG Y H, TIAN Q, et al. GhostNet: more features from cheap operations[C]//Proceedings of the IEEE/ CVF conference on computer vision and pattern recognition. Piscataway: IEEE, 2020: 1580-1589.
- [10]蒋端, 彭龔. 基于 YOLOv8 改进的轻量化森林火灾检测算法[J]. 兰州工业学院学报, 2025, 32(2): 32-37.

【作者简介】

秦健(1999—), 通信作者(email:qinjian13814348552@163.com), 男, 江苏盐城人, 硕士研究生, 研究方向: 机器视觉。

(收稿日期: 2025-06-26 修回日期: 2025-12-04)