

DOI: 10.3969/j.issn.1001-3881.2024.19.002

文献引用: 王众玄, 邹光明, 顾浩文, 等. 基于改进 YOLOv8 的堆叠零件实例分割研究[J]. 机床与液压, 2024, 52(19): 9-16.

Cite as: WANG Zhongxuan, ZOU Guangming, GU Haowen, et al. Research on stacked part instance segmentation based on improved YOLOv8 [J]. Machine Tool & Hydraulics, 2024, 52(19): 9-16.

基于改进 YOLOv8 的堆叠零件实例分割研究

王众玄^{1,2}, 邹光明^{1,2}, 顾浩文^{1,2}, 许艳涛^{1,2}, 李陈佳瑞^{1,2}

(1. 武汉科技大学冶金装备及其控制教育部重点实验室, 湖北武汉 430081;

2. 武汉科技大学机械传动与制造工程湖北省重点实验室, 湖北武汉 430081)

摘要: 为了实现复杂工业环境下机器人对堆叠零件的快速识别拣选, 构建一种改进的 YOLOv8s 实例分割模型, 并应用于堆叠零件实时识别分割中。针对堆叠工业零件不易分割的问题, 将原始模型的主干网络替换为提取特征能力更强的 PoolFormer 主干网络, 提升堆叠零件边缘分割效果; 为了更好地过滤掉多余背景信息, 保留关键信息, 引入了效果更好的 CARAFE 上采样模块。试验结果表明, 改进后模型的分割平均精度和预测框平均精度分别为 93.57% 和 97.47%, 相比原模型提升了 1.89% 和 1.23%, 且远高于同类型的 YOLACT++ 和 SOLOv2 模型, 验证了改进模型的有效性。

关键词: 堆叠零件; 实例分割; YOLOv8 网络; 主干网络; 上采样

中图分类号: TP242

Research on Stacked Part Instance Segmentation Based on Improved YOLOv8

WANG Zhongxuan^{1,2}, ZOU Guangming^{1,2}, GU Haowen^{1,2}, XU Yantao^{1,2}, LI Chenjiarui^{1,2}

(1. Key Laboratory of Metallurgical Equipment and Control Technology, Ministry of Education, Wuhan University of Science and Technology, Wuhan Hubei 430081, China, 2. Key Laboratory of Mechanical Transmission and Manufacturing Engineering of Hubei Province, Wuhan University of Science and Technology, Wuhan Hubei 430081, China)

Abstract: In order to achieve rapid recognition and selection of stacked parts by robots in complex industrial environments, an improved YOLOv8s instance segmentation model was constructed and applied to real-time recognition and segmentation of stacked parts. To address the issue of difficult segmentation of stacked industrial parts, the original model's backbone network was replaced with a PoolFormer backbone network with stronger feature extraction capabilities to improve the edge segmentation effect of stacked parts; in order to better filter out excess background information and retain key information, a better CARAFE upsampling module was introduced. The experimental results show that the average segmentation accuracy and prediction box accuracy of the improved model are 93.57% and 97.47%, respectively, which are 1.89% and 1.23% higher than the original model, and far higher than the YOLACT++ and SOLOv2 models with the same type, verifying the effectiveness of the improved model.

Keywords: stacked parts; instance segmentation; YOLOv8 network; backbone; up sample

0 前言

在工业生产中, 通常需要对输送线上零件进行识别和拣选操作, 但由于零件在运输过程中往往会出现堆叠现象, 对堆叠零件进行检测和分割是一个具有挑战性的问题。传统的视觉检测方法在处理堆叠零件时往往精度差、实用性不强, 而基于深度学习的实例分割技术能够通过精细的像素级预测, 高度准确地将每个零件分割出来, 实现对堆叠零件的精准识别。

目前, 基于深度学习的实例分割方法大致可以分为两类。一类是以检测和分割分段完成的两阶段实例分割方法, 最早由 GIRSHICK 等^[1] 提出并命名为

RCNN (Region-CNN), RCNN 算法分割质量较好, 但缺点是训练时间长且测试过程复杂。因此, REN 等^[2] 提出 Faster RCNN 模型, 它通过一个区域提案网络 (RPN) 来生成区域提案, 大大提高了检测效率。HE 等^[3] 在 Faster RCNN 上进行改进, 提出了 MASK R-CNN 算法, 实现了精度较高的分割。项荣、张茂琛^[4] 通过对 MASK R-CNN 掩码分支进行改进, 实现了番茄茎秆的准确分类。但由于两阶段的实例分割网络模型较大, 导致计算速度慢, 无法满足实时检测和分割要求。另一类则是检测和分割同步进行的单阶段实例分割方法, 其优点是模型简单, 能够保证精度和

收稿日期: 2023-10-12

作者简介: 王众玄 (1997—), 男, 硕士研究生, 研究方向为机器视觉。E-mail: 2587901935@qq.com。

通信作者: 邹光明 (1970—), 男, 教授, 主要研究方向为机器视觉、机器人。E-mail: zouguangming@wust.edu.cn。

实时性。YOLACT 作为最早的单阶段实时实例分割模型^[5]，通过并行地将图像分割分叉成两个子任务：生成原型掩膜和预测每个实例掩膜的掩膜系数，从而极大地提高了运行和检测速度，但在存在多个重叠目标的图像中，其分割精度稍显不足。为此，ZHOU^[6]在原模型基础上提出了 YOLACT++，提高其精度的同时也保证了实时性。肖珂等人^[7]通过改进 YOLACT++ 的主干网络和引入 NAM 注意力机制，实现了果树叶墙区域的实时检测。WANG 等^[8]提出的 SOLO 模型将实例分割定义为分类感知预测和实例感知掩码生成问题，通过将输入图像进行网格划分来预测其类别和分割实例。为了改进其掩码预测和运行效率，SOLOv2 通过引入非极大值抑制技术来减少推理时间和提高掩码精度^[9]，庄前伟等^[10]将 SOLOv2 主干网络进行特征融合，并引入可变形卷积 DCNv2，实现了穴盘幼苗的高效分割。

YOLO 作为单阶段实时目标检测任务的主流网络之一，随着 YOLOv1~v8 的不断演进，其在实例分割上也崭露头角。如 Poly-YOLO 便是在 YOLOv3 的思想上进行改进，通过限定多边形区域来执行实例分割^[11]。而 YOLOv8 作为 YOLO 系列的最新演进，将单阶段实时检测的优势与高质量的实例分割相结合，弥补了以往单阶段方法处理复杂场景精度不足的问题。由于 YOLOv8 在检测和分割中的出色表现，本文作者用它来进行堆叠零件的实例分割任务，使其能够在工业场景中快速而准确地识别堆叠零件，并通过改进其算法以满足堆叠零件实时实例分割的特殊需求。

1 堆叠零件数据集建立

1.1 堆叠零件图像采集

以某智能仓储内流水线上堆叠零件为研究对象，为方便后续对堆叠零件的拣选工作，需要设计一种准确高效的实例分割算法对需要拣选的零件进行识别分割。在零件选择上，选用常见的球头挂环（Q-7）、设备线夹（SYG-50）、平行挂板（PD-7）3 种不同零件进行堆叠试验，将 3 种不同的零件分别随机进行堆叠摆放，利用相机在不同角度和高度，采集到 3 种不同堆叠零件图片各 200 张，共计 600 张。其中部分采集图像如图 1 所示。

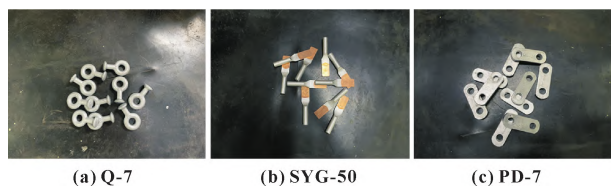


图 1 堆叠零件图像采集

Fig. 1 Stacked parts image acquisition: (a) Q-7; (b) SYG-50; (c) PD-7

1.2 数据集增强

为了扩展数据集、减少过拟合风险和提高网络的泛化能力，可以采用数据增强技术来增加已收集图像的数量。数据增强的方式有许多种，本文作者主要通过改变亮度、平移、镜像、增加椒盐噪声等方法，将采集到的图像扩增至 1 800 张，其中部分零件增强效果如图 2 所示。增强后的数据集按照 7:2:1 的比例划分为训练集、验证集和测试集。

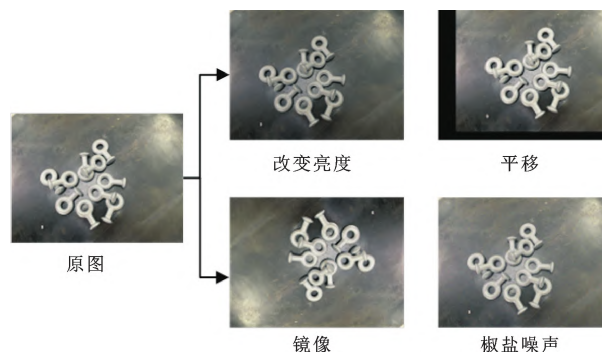


图 2 数据增强效果

Fig. 2 Dataset enhancement rendering

1.3 数据集标注

使用 Labelme 标注软件将划分好的训练集和验证集进行标注，部分标注样式如图 3 所示。在实际生产过程中，由于机械臂对堆叠零件进行拣选时遵循先从最上层完整零件进行拣选，因而在标注过程中只标注轮廓完整的零件。标注时将球头挂环、设备线夹和平行挂板这 3 种不同零件分别标注为 Q-7、SYG-50 和 PD-7（标注时只需标注出待拣选零件，其余默认为背景）。将数据集标注完成后使用编写好的 Python 程序将其转换为 YOLOv8 的数据集格式。



图 3 数据集标注示意

Fig. 3 Dataset annotation

2 YOLOv8 网络结构

YOLOv8 主要由 Input、Backbone、Neck、Prediction Header 组成，其网络结构如图 4 所示。其中，Input 为堆叠零件图像的输入环节，通过数据增强与自适应图像缩放等方法将输入的零件图像统一缩放到一个标准尺寸。Backbone 主要由 CBS、C2f 和 SPPF 模块组成，实现输入图像的特征提取操作。其中，CBS

由 Conv、BN 和 SILU 组成, 分别用于提取特征、规范化输出和增加模型的非线性能力, 以提高目标检测的准确性和稳定性。C2f 模块是 YOLOv8 网络中的重要组成部分, 其主要作用是增加网络的深度和感受野, 提高特征提取能力。SPPF 模块是一种池化模块, 主要实现输入数据的空间不变性和位置不变性, 提高

神经网络的识别能力。Neck 部分主要由 PAN+FPN 模块组成, 对 Backbone 提取到的特征进行采样连接操作, 实现堆叠零件的特征融合。Prediction Header 部分主要采用解耦头结构 (Decoupled-Head), 将分类和检测分离, 最终输出堆叠零件的分类信息和分割掩码, 实现堆叠零件的实例分割。

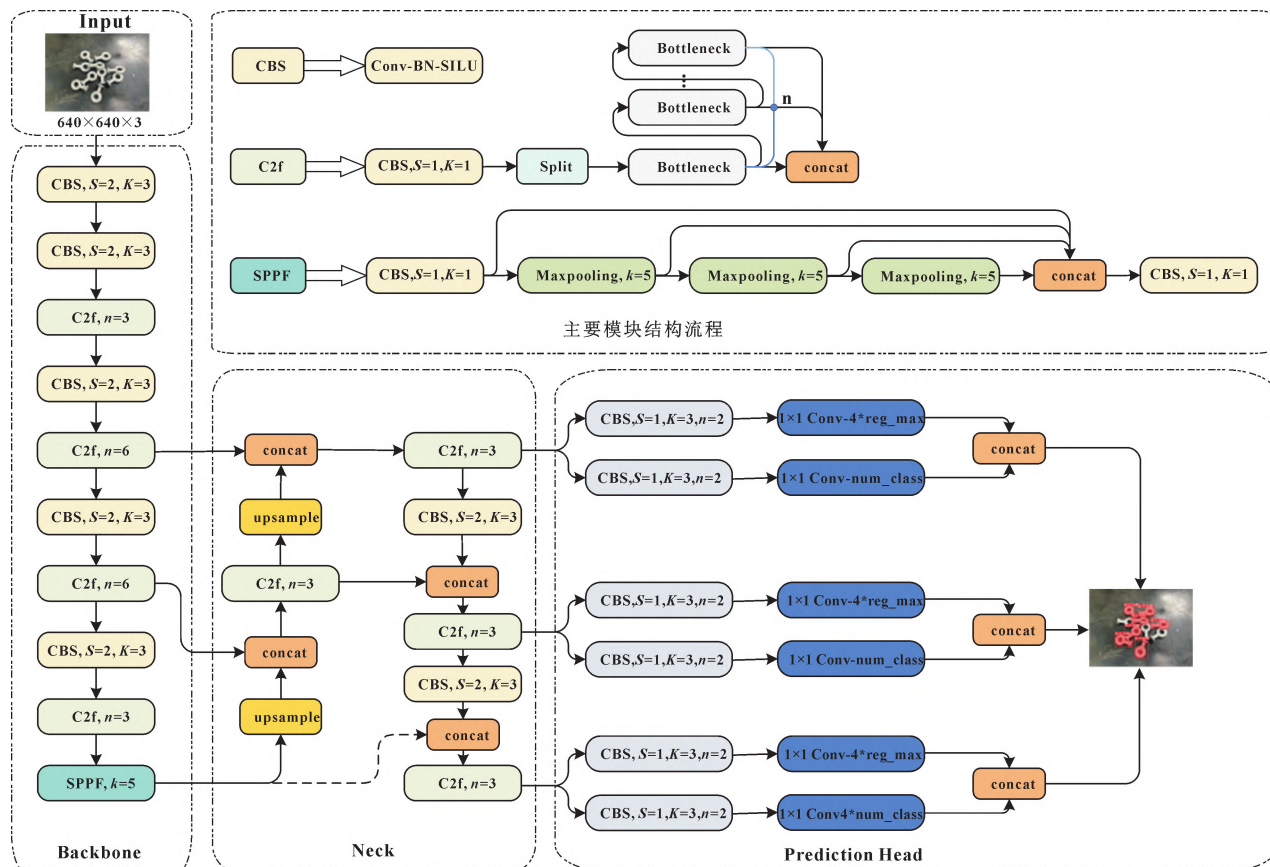


图 4 YOLOv8 网络结构

Fig. 4 YOLOv8 network structure

3 YOLOv8 网络改进

3.1 主干网络的改进

YOLOv8 主干网络主要借用了 Darknet-53 的思想, 使得网络在保证精度的同时又能得到不错的检测和分割效果。但随着机器视觉的不断发展, 基于 Transformer^[12] 的自注意力机制在图像处理领域取得了不错的成绩, 其架构相比传统卷积神经网络具有一些显著优点, 能够更准确有效地捕捉图像中的关键特征信息, 但缺点是其所需的训练量和参数量也远高于传统的卷积神经网络。通常来说, Transformer 的成功主要归功于其基于注意力的 token 混合器, 但 MetaFormer 将目光放在 Transformer 这个架构本身, 将其抽象为一个通用架构的 MetaFormer, 并且使用简单的非参数算子 pooling 作为 MetaFormer 的极弱 token 混合器, 从而构建了 PoolFormer 网络架构^[13], 如图 5 所示。

在 PoolFormer 架构中, 原始输入数据首先通过补丁嵌入 (InputEmb) 处理得到新的表示 X , 再将 X 送入到两个残差子块中。第一个子块由归一化 (Norm) 和池化 (Pooling) 模块组成, 主要将原始 Transformer 架构中的 Attention 模块替换为 1 个步长为 1 的池化块, 从而执行平均池化, 得到结果与原始输入 X 相加得到特征 Y 。第二个子块由归一化 (Norm) 和多层感知机 (MLP) 模块组成, 通过归一化、线性变换和非线性激活等操作, 将得到的特征与之前的特征 Y 相加获得最终的输出 Z 。主要计算过程如下:

$$X = \text{InputEmb}(L) \quad (1)$$

$$Y = \text{Pool}(\text{Norm}(X)) + X \quad (2)$$

$$Z = \sigma(\text{Norm}(Y)W_1)W_2 + Y \quad (3)$$

式中: L 为输入数据; Pool 表示平均池化; Norm 表示分组归一化; W_1 和 W_2 为 MLP 中的可学习参数; σ 为 GELU 激活函数^[14]。

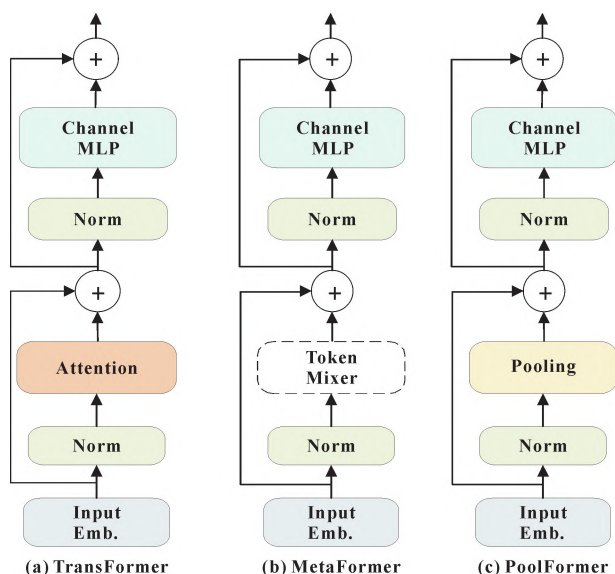


图 5 PoolFormer 网络架构对比

Fig. 5 PoolFormer network architecture comparison:

(a) Transformer; (b) MetaFormer;

(c) PoolFormer

虽然 PoolFormer 只是将 Transformer 架构中的 Attention 模块换成了简单的 Pooling 模块, 但大大减少了网络参数量, 并在某些数据集上表现效果更好。本文作者为了保证识别准确率和实时性, 选择以 YOLOv8s 模型进行堆叠零件实例分割试验, 因而将其主干替换为模型大小与之相近的 poolformer_s24 主干网络。它主要由 4 个 PoolFormer Blocks 堆叠而成, 堆叠比例为 [4, 4, 12, 4], 将其放到 YOLOv8 主干上能够有效增加网络的特征提取能力, 更好地捕捉图像中的语义信息, 提高模型在堆叠零件实例分割任务上的性能。改进后的 YOLOv8 网络结构如图 6 所示。

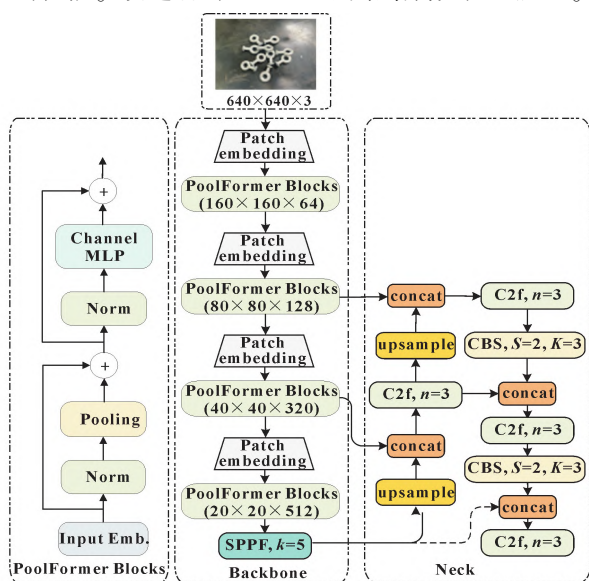


图 6 改进后的主干网络结构

Fig. 6 Improved backbone network structure

3.2 上采样模块改进

上采样模块的作用是将输入的特征图的尺寸增加, 从而增加图像或特征图的分辨率, 以便更好地捕捉物体的特征信息, 从而提高物体检测和分割性能。常用的上采样方法有最近邻插值、双线性插值、反池化等。YOLOv8 使用最近邻插值方法进行上采样, 将输入的特征图沿着高度和宽度两个维度进行 2 倍上采样, 以便与下一层的特征图进行连接操作。该算法简单、速度快, 其主要计算公式如下:

$$cX_s = X_g(W_s/W_g) \quad (4)$$

$$cY_s = Y_g(H_s/H_g) \quad (5)$$

式中: X_s 、 Y_s 分别为原图像某个点的横、纵坐标; X_g 、 Y_g 分别为目标图像某个点的横、纵坐标; W_s 、 H_s 分别为原图像的宽度和高度; W_g 、 H_g 分别为目标图像的宽度和高度。

由于最近邻插值算法仅通过像素点的空间位置决定上采样核, 并没有利用特征图的语义信息, 并且感知区域通常很小。而引入的 CARAFE 上采样算子^[15]在保持轻量化的同时具备较大的感受野, 能够更好地利用周围信息和特征图的语义信息。它主要由两个模块组成, 分别是上采样核预测模块和特征重组模块。

上采样核预测模块主要将输入的 $H \times W \times C$ 的特征图用一个 1×1 卷积压缩至 $H \times W \times C_m$, 减少后续步骤的计算量; 然后将压缩后的特征图用一个 $k_{\text{encoder}} \times k_{\text{encoder}}$ 卷积层来预测上采样核, 得到形状为 $\sigma H \times \sigma W \times k_{\text{up}}^2$ 的上采样核; 最后将得到的上采样核利用 softmax 函数进行归一化, 使得卷积核权重和为 1。

特征重组模块主要将输出特征图中的每个位置映射回输入特征图, 并取出以之为中心的 $k_{\text{up}} \times k_{\text{up}}$ 区域, 再将其与预测出的该点的上采样核作点积, 得到最终的输出值。

对于形状为 $H \times W \times C$ 的输入特征图 X , 如果上采样倍率为 σ , CARAFE 首先利用上采样核预测模块预测出上采样核, 然后利用特征重组模块完成上采样, 得到形状为 $\sigma H \times \sigma W \times C$ 的输出特征图。其主要计算过程如下:

$$W_l = \psi(N(X_l, k_{\text{encoder}})) \quad (6)$$

$$X'_l = \varphi(N(X_l, k_{\text{up}}), W_l) \quad (7)$$

式中: ψ 为上采样核预测模块; X_l 为以 l 为中心关于 X 的 $k \times k$ 子区域; k_{encoder} 表示内核大小为 k_{encoder} 的卷积层; W_l 表示预测的位置感知核; φ 为特征重组模块; k_{up} 为重组内核大小; X'_l 为输出特征值。

4 试验与结果分析

4.1 试验环境

文中试验软件环境由 ubuntu20.04、Python3.8、PyTorch1.10.0 和 Cuda11.3 搭建而成, 显卡为

NVIDIA GeForce RTX3080 10GB, 采用余弦退火动态调整学习率从而能够较快达到局部收敛效果。相关参数如表 1 所示。

表 1 试验相关模型参数

Tab. 1 Experimental related model parameters

参数	数值/类别
图像尺寸	640 像素×640 像素
优化器	Adam
批量大小	8
最大学习率	0.01
迭代次数	300
学习率调整方式	余弦退火

4.2 试验指标

因为文中的数据集是按照 COCO 数据集^[16] 格式制作而成, 为了更好地评价堆叠零件的实例分割效果, 故选用平均精度 AP (Average Precision) 来评估检测框和分割掩膜的精度, AP 值的计算如式 (8) — (10) 所示。由于 IOU 阈值的选择会对精确度和召回率产生影响, 因此文中主要采用 $\delta_{AP,all}$ (计算范围内多个 IOU 阈值的平均精度, 范围为 0.5 ~ 0.95, 步长为 0.05)、 $\delta_{AP,50}$ (IOU 阈值为 0.5) 和 $\delta_{AP,75}$ (IOU 阈值为 0.75) 作为评价指标。

$$P = \frac{n_{TP}}{n_{TP} + n_{FP}} \quad (8)$$

$$R = \frac{n_{TP}}{n_{TP} + n_{FN}} \quad (9)$$

$$\delta_{AP} = \int_0^1 p(R) dR \quad (10)$$

其中: n_{TP} 表示模型正确预测为正类别的样本数; n_{FP} 是模型错误预测为正类别的样本数; n_{FN} 是模型未能正确预测为正类别的样本数。

4.3 试验结果分析

4.3.1 主干网络改进前后热力图对比

将训练完成得到的权重利用 grad-cam 生成热力图, 得到如图 7 所示的主干网络改进前后热力图对比, 图中颜色越深表示主干网络对该区域的关注力越强。其中, 原模型主干网络关注区域包含堆叠零件区域和背景区域, 并且背景区域关注度明显高于堆叠零件区域。而改进后的主干网络特征由于提取能力更强, 因而更关注堆叠零件区域内的单个零件, 从而极大抑制了背景的干扰。

4.3.2 损失值对比

在深度学习中, 损失值是用来衡量模型性能的一种指标, 反映了模型的预测值与实际目标之间的差异, 损失值越小表示模型预测性能越高。将原始 YOLOv8s 主干和上采样模块替换后, 经过 300 次迭代后, 得到改进前后的模型损失值曲线如图 8 所示。可

以看到: 经过 300 次迭代后, 原模型损失值由 6.5 降到 0.31 左右, 改进模型损失值由 3.7 下降至 0.22 左右, 相比原模型收敛速度更快, 效果也更好。

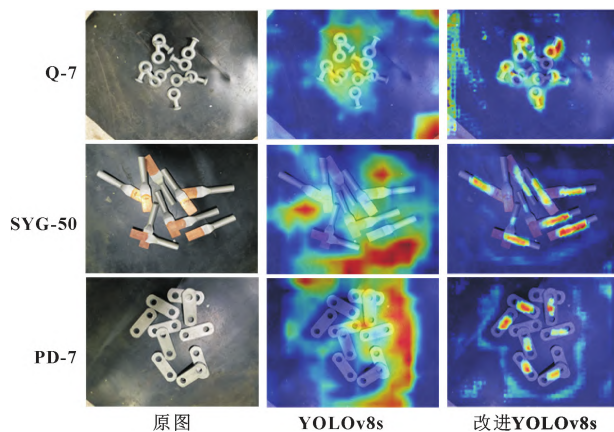


图 7 主干网络热力图对比

Fig. 7 Comparison of backbone network thermal maps

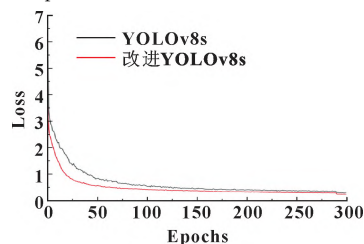


图 8 掩膜损失值曲线

Fig. 8 Mask loss value curves

4.3.3 消融试验

消融试验的目的在于深入研究模型或算法的性能, 通过逐步修改模型的组件、模块或超参数, 从而观察这些变化对模型性能的影响。为了比较不同改进方式时模型的分割效果, 文中统一了超参数, 并在验证集上进行消融试验, 得到的结果如表 2 所示。

可以看到: 两种不同的改进方法都能够有效提升原模型的检测和分割精度。对比原模型可知, 当将主干网络改进为 PoolFormer 主干时, 模型的分割平均精度 $\delta_{AP,all}$ 、 $\delta_{AP,50}$ 、 $\delta_{AP,75}$ 分别提升 1.56%、0.27% 和 0.5%, 边界框平均精度 $\delta_{AP,all}$ 、 $\delta_{AP,50}$ 、 $\delta_{AP,75}$ 分别提升 0.92%、0.26% 和 0.25%; 当主干网络不变而改进上采样模块时, 分割的 $\delta_{AP,all}$ 和边界框的 $\delta_{AP,all}$ 分别提升 0.68% 和 0.27%, $\delta_{AP,75}$ 分别提升 0.42% 和 0.21%, 略低于主干网络改进效果。当两种不同改进方法同时进行, 模型的性能进一步提升, 最终改进后的 YOLOv8s 模型的分割平均精度 $\delta_{AP,all}$ 和边界框平均精度 $\delta_{AP,all}$ 分别达到 93.57% 和 97.47%, 相比原模型上升了 1.89% 和 1.23%, 分割的 $\delta_{AP,50}$ 和 $\delta_{AP,75}$ 提升了 0.37% 和 0.61%, 边界框的 $\delta_{AP,50}$ 和 $\delta_{AP,75}$ 提升了 0.38% 和 0.36%。

表 2 消融试验结果对比

Tab. 2 Comparison of ablation experiment results

模型	分割平均精度/%			边界框平均精度/%		
	$\delta_{AP,all}$	$\delta_{AP,50}$	$\delta_{AP,75}$	$\delta_{AP,all}$	$\delta_{AP,50}$	$\delta_{AP,75}$
YOLOv8s	91.68	97.96	97.65	96.24	97.96	97.96
YOLOv8s+PoolFormer	93.24	98.23	98.15	97.16	98.22	98.21
YOLOv8s+CARAFE	92.36	98.17	98.07	96.51	98.17	98.17
改进 YOLOv8s	93.57	98.33	98.26	97.47	98.34	98.32

为了判断模型是否满足实时实例分割需求,除了分析评估模型的检测精度和分割效果外,模型的大小和运行时间也是需要考虑的因素。将改进后的网络与原网络在测试集上进行对比,得到的结果如表 3 所示。可以看到:改进后的模型大小相比原模型增大了 12.9 MB,但单幅图片检测时间只增加了 4.8 ms,依旧能满足堆叠零件实时实例分割需求。

表 3 改进前后模型大小及检测时间对比

Tab. 3 Comparison of model size and detection time before and after improvement

模型	模型大小/MB	单幅图片检测时间/ms
YOLOv8s	11.8	24.0
改进 YOLOv8s	24.7	28.8

4.3.4 试验结果分析

改进原始 YOLOv8s 模型后,为了验证模型的实际分割性能,将先前划分好的 180 张验证集图片用来进行测试,得到 3 种不同堆叠零件实例分割效果,其中部分零件分割效果如图 9 所示。

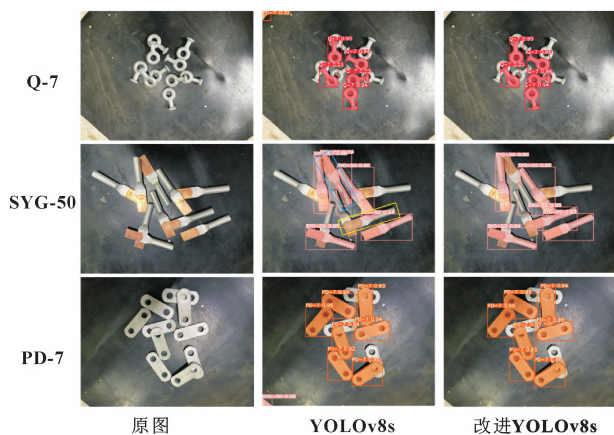


图 9 改进前、后模型分割效果对比

Fig. 9 Comparison of model segmentation effects before and after improvement

可以看到:原 YOLOv8s 模型对于堆叠零件背景区域存在误识别问题,并且当零件堆叠紧凑时会发生误分割(SYG-50 中蓝色方框)和未分割(SYG-50 中黄色方框)现象,并且边缘分割效果较差。通过改进主干网络和上采样算子后,改进 YOLOv8s 模型提取特征能力明显变强且更关注零件堆叠区域的识别和

分割,减少了背景的干扰,增强了分割准确率,对堆叠零件边缘部位的分割效果也更好。

为了显示改进后的模型能否真实有效地识别出需要拣选的零件,统计这两种模型在测试集中识别出的零件数量,并计算识别零件的准确率,准确率越高证明模型效果越好。其中,分割的准确率用式(11)计算,最终测试得到的结果如表 4 所示。

$$P_1 = \frac{A}{B + C} \quad (11)$$

式中: P_1 为识别准确率; A 为正确识别零件数量; B 为需要正确识别零件数量; C 为错误及未识别零件数量。

由表 4 和图 9 可知,改进后的 YOLOv8s 模型对这 3 种不同堆叠零件分割准确率都有所提升,提升率分别为 2.36%、4.25% 和 1.69%,并且 SYG-50 类堆叠零件相比原模型提升效果最明显,侧面证明了改进后的模型能够有效完成 3 种不同堆叠零件的识别分割任务。

表 4 改进前后零件识别准确率对比

Tab. 4 Comparison of part recognition accuracy before and after improvement

模型	零件类别	需要正确识别零件数量	正确识别数量	错误及未识别零件数量	识别准确率/%
YOLOv8s	Q-7	375	369	6	96.85
	SYG-50	318	315	24	92.11
	PD-7	351	345	3	97.46
改进 YOLOv8s	Q-7	375	375	3	99.21
	SYG-50	318	318	12	96.36
	PD-7	351	351	3	99.15

4.3.5 不同深度模型试验结果分析

为了进一步验证和分析不同深度模型对堆叠零件的实例分割性能,将改进前后的 YOLOv8s 模型和 YOLACT++、SOLOv2 进行比较,得到的分割效果如图 10 所示。可知:对于不同类型的堆叠零件图像,YOLACT++模型分割效果表现最差,不能有效地分割出待拣选零件,导致堆叠图像中存在部分误分割和未分割零件;而 SOLOv2 算法虽然能够正确分割出需要拣选的零件,但通过与文中算法对比可以看到,该模

型对于堆叠零件的边缘分割效果较差，不能够有效分割出零件的边缘轮廓，因而文中模型比其他模型有更好的检测效果。同时，为了更好比较不同模型的分割效果，将这 4 种模型的测试结果绘制成表 5。

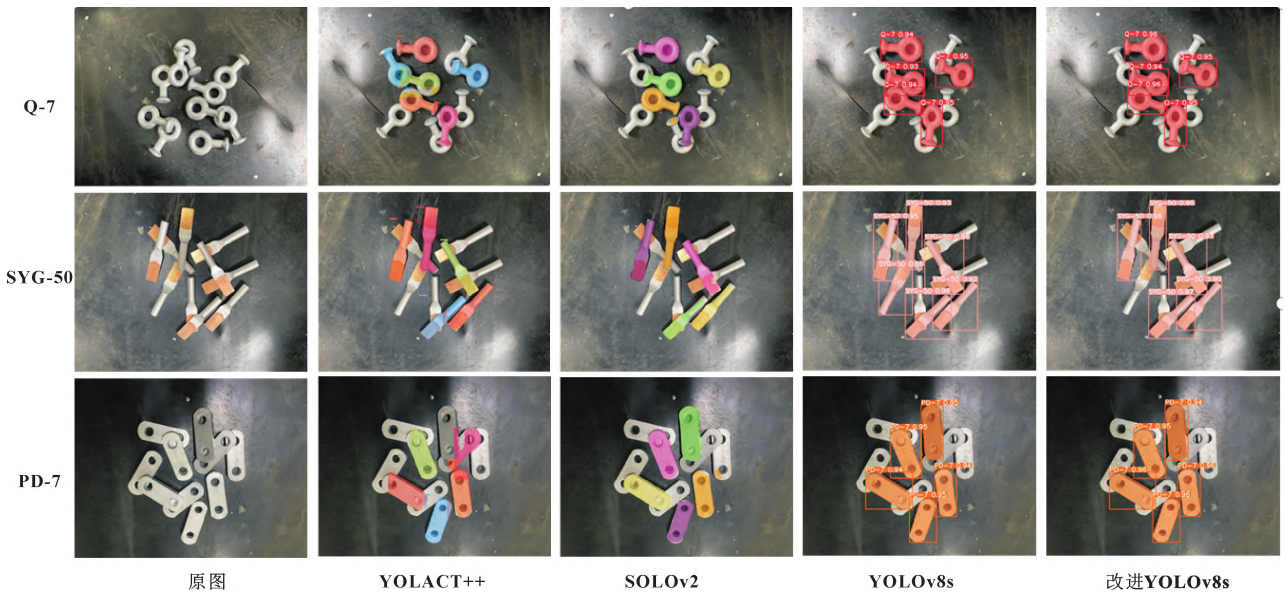


图 10 不同深度模型分割效果对比

Fig. 10 Comparison of segmentation effects among different depth models

表 5 不同深度模型测试结果对比

Tab. 5 Comparison of test results for different depth models

模型	分割平均 精度/%		边界框平均 精度/%		单幅图 片检测 时间/ms
	$\delta_{AP,all}$	$\delta_{AP,75}$	$\delta_{AP,all}$	$\delta_{AP,75}$	
YOLOv8s	91.68	97.65	96.24	97.96	24
YOLACT++	76.96	83.26	78.75	86.28	14
SOLOv2	83.43	87.02	\	\	22
改进 YOLOv8s	93.57	98.26	97.47	98.32	28.8

从表 5 可以看到：改进后的 YOLOv8s 模型的单幅图片检测时间最长，比 YOLACT++ 和 SOLOv2 模型分别高 14.8 ms 和 6.8 ms，主要原因还是 transformer 架构导致的模型参数量和计算量的增加；但是在分割平均精度上，改进 YOLOv8s 的 $\delta_{AP,all}$ 和 $\delta_{AP,75}$ 与 YOLACT++ 和 SOLOv2 相比，分别高 16.61%、10.14% 和 15%、11.24%，并且边界框平均精度远高于 YOLACT++ 模型。以上分析表明，本文作者所提出的改进模型在实时实例分割模型中具有较大优势，能够满足堆叠零件实时实例分割需求。

5 结论

(1) 为实现堆叠工业零件的快速拣选需求，提出一种基于改进 YOLOv8s 的实例分割模型。在原始 YOLOv8s 模型基础上，通过对主干网络进行改进，将其替换为网络结构更深、特征提取能力更强的 PoolFormer 主干网络，能够更好地分割堆叠零件轮廓边缘。同时为了过滤掉多余的背景区域，保留需要分割的区域，将原始上采样模块替换为 CARAFE 模块，

减少了模型对零件的误检测概率。试验结果证明，改进后的模型分割和检测平均精度均高于原 YOLOv8s、YOLACT++ 和 SOLOv2 模型，并对未训练过的样本进行测试，得出的分割效果远好于其他模型，有效证明了改进模型的可行性。

(2) 尽管改进后的 YOLOv8s 模型在处理堆叠零件分割方面表现出了出色的性能，但仍然存在部分误识别和误分割的问题。这些问题主要源于堆叠场景的复杂性，以及堆叠可能引起的光照不足、阴影和遮挡等因素。在未来的研究中，可以通过改善照明条件、进一步优化实例分割算法，以增强模型的鲁棒性和适用性。

参考文献

[1] GIRSHICK R,DONAHUE J,DARRELL T,et al.Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition.Columbus,OH,USA:IEEE,2014:580-587.

[2] REN S Q,HE K M,GIRSHICK R,et al.Faster R-CNN:towards real-time object detection with region proposal networks[J].IEEE Transactions on Pattern Analysis and Machine Intelligence,2017,39(6):1137-1149.

[3] HE K M,GKIOXARI G,DOLLÁR P,et al.Mask R-CNN [C]//Proceedings of 2017 IEEE International Conference on Computer Vision.Venice,Italy:IEEE,2017:2980-2988.

[4] 项荣,张茂琛.基于改进 Mask R-CNN 的番茄茎秆分类方法[J].湖南大学学报(自然科学版),2023,50(2):31-39.

- XIANG R,ZHANG M C.Tomato stem classification method based on improved mask R-CNN[J].Journal of Hunan University(Natural Sciences),2023,50(2):31-39.
- [5] BOLYA D,ZHOU C,XIAO F Y,et al.YOLACT:real-time instance segmentation[C]//Proceedings of 2019 IEEE/CVF International Conference on Computer Vision.Seoul,Korea (South):IEEE,2019:9156-9165.
- [6] ZHOU C.Yolact++ better real-time instance segmentation[M].Davis:University of California,2020.
- [7] 肖珂,梁聪哲,夏伟光.基于改进 YOLACT 的果树叶墙区域实时检测方法[J].农业机械学报,2023,54(4):276-284.
- XIAO K,LIANG C Z,XIA W G.Real-time detection method of fruit leaf wall area based on improved YOLACT[J].Transactions of the Chinese Society for Agricultural Machinery,2023,54(4):276-284.
- [8] WANG X L,KONG T,SHEN C H,et al.SOLO:segmenting objects by locations[C]//Proceedings of European Conference on Computer Vision,2020:649-665.
- [9] WANG X L,ZHANG R F,KONG T,et al.SOLOv2:dynamic and fast instance segmentation[EB/OL].2020:2003.10152.<https://arxiv.org/abs/2003.10152v3>
- [10] 庄前伟,王志明,吴龙貽,等.基于改进 SOLOv2 的穴盘幼苗图像分割方法[J].南京农业大学学报,2023,46(1):200-209.
- ZHUANG Q W,WANG Z M,WU L Y,et al.Image segmentation method of plug seedlings based on improved SOLOv2[J].Journal of Nanjing Agricultural University,2023,46(1):200-209.
- [11] HURTIK P,MOLEK V,HULA J,et al.Poly-YOLO:higher speed,more precise detection and instance segmentation for YOLOv3[J].Neural Computing and Applications,2022,34(10):8275-8290.
- [12] VASWANI A,SHAZEER N,PARMAR N,et al.Attention is all you need[C]//Proceedings of 31st Conference on Neural Information Processing Systems,2017.
- [13] YU W H,LUO M,ZHOU P,et al.MetaFormer is actually what you need for vision[C]//Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).New Orleans,LA,USA:IEEE,2022:10809-10819.
- [14] HENDRYCKS D,GIMPEL K.Gaussian error linear units (GELUs)[EB/OL].2016:1606.08415.<https://arxiv.org/abs/1606.08415v5>
- [15] WANG J Q,CHEN K,XU R,et al.CARAFE:content-aware ReAssembly of FEatures[C]//Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV).Seoul,Korea (South):IEEE,2019:3007-3016.
- [16] LIN T Y,MAIRE M,BELONGIE S,et al.Microsoft COCO:common objects in context[C]//Proceedings of Computer Vision - ECCV 2014:13th European Conference,Zurich,Switzerland.Cham:Springer International Publishing,2014:740-755.