

[文章编号] 1003-4684(2022)04-0029-07

基于 YOLACT++ 的槟榔检测算法研究

舒 军¹, 王 祥², 舒心怡³

(1 湖北工业大学电气与电子工程学院, 湖北 武汉 430068;

2 湖北工业大学太阳能高效利用及储能运行控制湖北省重点实验室, 湖北 武汉 430068;

3 武外英中国际学校, 湖北 武汉 430000)

[摘 要] 设计了一个基于 YOLACT++ 深度学习算法的槟榔检测模型。针对传送带上采集的槟榔图片分割精度低和预测框不精确造成槟榔分级准确率低的问题。在模型主干网络中引入改进 Res2Net 模块, 改善槟榔掩模分割精度。在模型边界框回归损失中引入 CIOU 损失函数, 提高预测框的检测精度。结果表明, 改进模型的掩模 mAP 相较 YOLACT++、Mask R-CNN、SOLOv2 分别高出 5.20%, 4.09%, 2.37%。预测框 mAP 相较 YOLACT++、Mask R-CNN 分别高出 5.41%, 4.90%。相较于模型改进前分级准确率提升 2.12%。

[关键词] YOLACT++; 槟榔检测; Res2Net 模块; CIOU 损失函数

[中图分类号] TP391.4 **[文献标识码]** A

槟榔是海南的重要经济作物, 产量占全国 97% 以上, 曾列入抗击新冠肺炎中药配方, 具有较高药用价值^[1]。槟榔检测作为槟榔加工的重要步骤。目前, 主要采用最大类间方差法 (OSTU)^[2]、最大极值稳定区域 (MSER)^[3] 等传统图像处理方法将图片中的槟榔与背景分割开, 再根据槟榔图像最小外接矩形的长宽对槟榔进行分级。在槟榔图像采集过程中, 由于机械振动造成槟榔花蒂掉落、槟榔堆叠, 和打光原因造成的阴影生成。使得传统的图像处理方法无法准确识别单个槟榔实例, 给槟榔分级造成严重干扰。因此, 实现高可靠性的槟榔实例分割是目前亟需解决的问题。

实例分割具备目标检测和语义分割的特点, 既需要检测出同一种类的不同实例对象, 又需要做到像素层面分类。当前, 实例分割算法逐渐被应用于农业生产中, 并具有较高的应用价值, 姜红花等^[4]将 Mask R-CNN 运用于玉米叶与杂草的检测, 杂草检测率达 91%。陈佳等^[5]采用 YOLACT 算法将白羽鸡背部的分割面积与体质质量进行关联, 实现白羽鸡体质质量评估, 平均准确率 97.49%。刘文波等^[6]在 SOLOV2 网络中引入可变形卷积, 实现对番茄叶片病害的检测, 掩模分割精度达 42.3%。YOLACT++^[7]作为 YOLACT 改进算法^[8], 相较于 Mask R-CNN^[9]具有速度优势, 较 SOLOV2 算法^[10]在槟榔堆叠时具有边缘分割优势。选用 YOLACT++

作为槟榔实例分割模型, 仍然存在掩模预测错误与检测框回归不准确造成的部分槟榔掩模残缺的问题。

针对槟榔自动化加工中槟榔检测的需求, 需要实例分割速度达到 18 帧/s 以上, 同时要确保检测与分割准确性。在 YOLACT++ 主干网络中引入改进的 Res2Net^[11] 模块增强网络语义信息提取能力, 改善掩模预测错误问题。对于边界框不精确问题, 本文采用 CIOU^[12] 边界框回归损失函数, 改善了边界框预测精度。实验结果表明, 改进算法改善了槟榔实例分割效果, 进而提高了槟榔的分级准确率。

1 槟榔检测流程

槟榔检测主要由图像采集、图像处理和槟榔分拣三部分完成, 其检测系统模型见图 1a, 使用工业相机采集传送带上的槟榔图片, 使用工业光源增加槟榔与传送带对比度, 突出槟榔特征。服务器用训练好的槟榔检测模型对采集来的槟榔图片进行实例分割与等级判断, 结果发送到人机交互界面, 同时控制气泵, 将不同等级的槟榔果吹入不同的收集桶中。

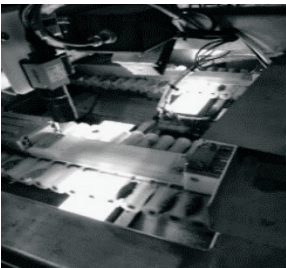
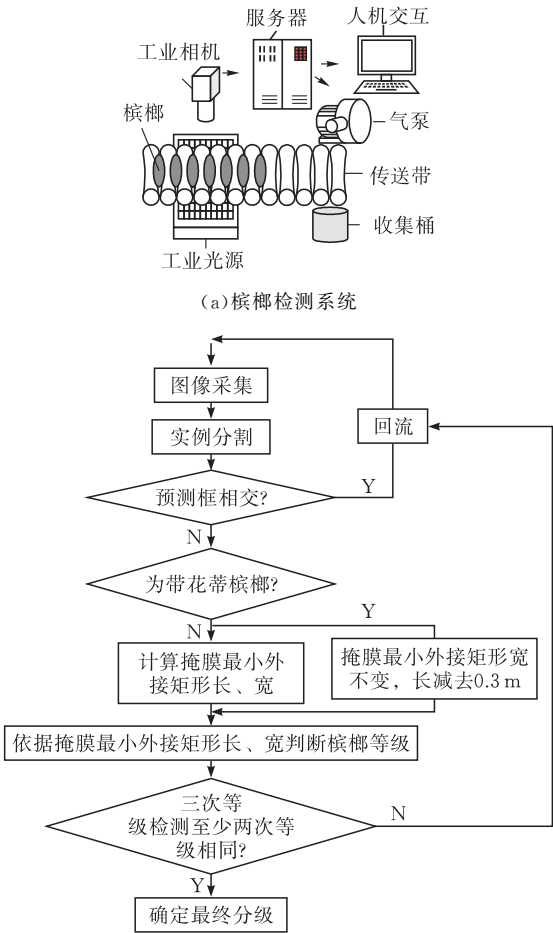
槟榔分级流程见图 1b, 工业相机完成槟榔图片的采集, 每张图片正常采集三颗槟榔, 用实例分割网络识别出带花蒂槟榔与无花蒂槟榔的类别、掩模和预测框位置, 若槟榔预测框相交, 可判断为槟榔堆叠, 进行回流处理。对于无花蒂的槟榔, 计算其掩模

[收稿日期] 2021-10-29

[第一作者] 舒 军 (1973-), 男, 湖北荆门人, 工学博士, 湖北工业大学副教授, 研究方向为机器视觉, 深度学习

[通信作者] 王 祥 (1995-), 男, 湖北孝感, 湖北工业大学硕士研究生, 研究方向为机器视觉, 深度学习

的小外接矩形的长和宽。对于带花蒂的槟榔,将其掩膜的最小外接矩形长度减去 30 mm,视为去花蒂处理,宽度保持不变。最后,依据最小外接矩形的长宽将槟榔分为 A、B、C、D 四个等级,槟榔分级指标见表 1。每颗槟榔均采集三个不同位置的图片,三次等级检测中选其中至少两次相同的等级作为最终分级。对三次等级检测均不同的情况进行回流处理。



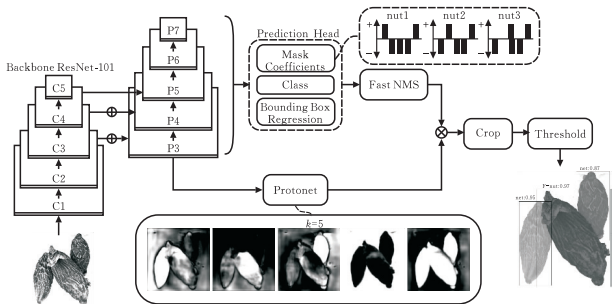
(c) 图像采集现场实验图

图 1 槟榔检测系统

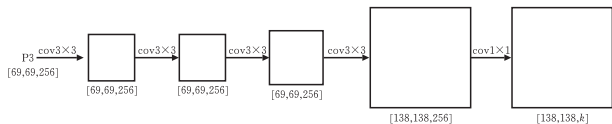
表 1 槟榔分级指标			mm
槟榔等级	最小外接矩形长度 L	最小外接矩形宽度 w	
A	$58 \leq L$	$21 \leq W$	
B	$54 < L \leq 58$	$16 < W \leq 21$	
C	$45 < L \leq 54$	$15 < W \leq 19$	
D	$L \leq 45$	$W \leq 15$	

2 YOLACT++ 网络结构

YOLACT++ 整体结构见图 2a, 包含主干 (Backbone)、预测头 (Prediction Header)、原型分支 (Protonet)、快速非极大值抑制 (Fast NMS)、裁剪模块 (Crop) 和阈值分割 (Threshold) 六个部分组成。基于 ResNet101 构造的主干网络对输入的图像进行特征提取, 生成 $C_1 \sim C_5$ 五个特征图。将 C_3, C_4, C_5 三个特征图作为特征金字塔的中间输入层, 经过多个尺度的特征融合生成 $P_3 \sim P_7$ 五个特征图, 再通过两个并行的分支任务。第一个分支将 P_3 特征图通过图 2(b) 所示的原型分支产生 k 个原型掩模。第二个分支在预测头中以特征图的每个像素点为锚点, 生成宽高比为 $[1, 1/2, 2, 1/3, 3]$ 的锚框, 用于检测是否存在实例。再分别进行原型掩模的系数预测 (mask Coefficients)、锚框分类 (Class) 与边界框回归 (Bounding Box Regression)。Fast NMS 对经过边界框回归后的锚框进行筛选, 获取最终的实例预测框。将原型掩模与掩模系数进行线性组合, 生成的图像经过预测框区域裁剪和阈值分割, 进而获得最终的实例掩模。



(a) YOLACT++ 整体结构



(b) protonet 结构

图 2 YOLACT++ 模型结构

3 算法改进

3.1 主干网络改进

YOLACT++ 采用基于 ResNet-101 的主干网络 (表 2), 其中 Bottleneck^[13] 残差结构作为 ResNet-101 的基本模块 (图 3a), 该结构在解决梯度消失同时增强网络的特征提取能力。为更进一步提取深层特征, 充分利用单层内特征, 本文采用改进的 Res2Net 模块替换主干网络中的 Bottleneck 模块, 以获得更强的分层多尺度表示能力。Res2Net 结构见图 3b。Res2Net 将经过 1×1 卷积特征图按通道

均匀分为 4 个特征子图,每个特征子图含有相同的空间大小和通道数。除第一个特征子图,其余每个特征子图都对应一个 3×3 卷积。从第三个特征子图开始,将特征子图与前一个经过卷积的特征子图输出相加,作为该特征子图对应卷积的输入,然后将 4 个特征子图的输出结果进行合并,并经过 1×1 卷积输出。通过对特征子图的分割和拼接实现特征重用,扩大感受野,有利于全局和局部信息提取。最后通过卷积输出。改进的 Res2Net 增加了 CBAM^[14] 注意力模块,对通道特征与空间特征进行加权,抑制无效特征,使得网络更关注于目标区域特征。

表 2 ResNet-101 主干网络

Layer name	Output size	ResNet-101
Input	[550,550,3]	—
C1	[275,275,64]	$7\times 7,64,\text{stride}2$
	[138,138,64]	$3\times 3,\text{maxpool},\text{stride}2$
C2	[138,138,256]	$\begin{bmatrix} 1\times 1,64 \\ 3\times 3,64 \\ 1\times 1,256 \end{bmatrix}\times 3$
C3	[69,69,512]	$\begin{bmatrix} 1\times 1,128 \\ 3\times 3,128 \\ 1\times 1,512 \end{bmatrix}\times 4$
C4	[35,35,1024]	$\begin{bmatrix} 1\times 1,256 \\ 3\times 3,256 \\ 1\times 1,1024 \end{bmatrix}\times 23$
C5	[18,18,2048]	$\begin{bmatrix} 1\times 1,512 \\ 3\times 3,512 \\ 1\times 1,2048 \end{bmatrix}\times 3$

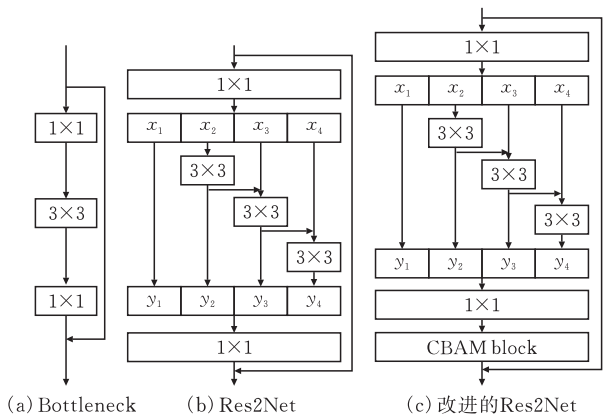


图 3 网络结构对比

CBAM 注意力模块结构见图 4。由通道注意力模块和空间注意力模块构成。分别为特征图生成一维的通道注意力权重和二维的空间注意力权重,用于加强网络的特征提取能力。

在通道注意力模块中将输入的特征图 U (大小 $H\times W\times C$) 经过全局最大池化和平均池化,获得两个 $1\times 1\times C$ 的特征图,其结果分别输入权重共享的

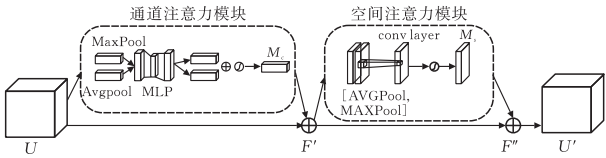


图 4 CBAM 结构

两层感知器(MLP),将输出的特征相加经过激活函数,生成通道注意力权重 M_c 。最后将输入特征图与通道注意力权重相乘,生成通道注意力特征图 F' 。计算过程如下:

$$Z_{\text{avg}} = \text{AvgPool}(U)$$

$$Z_{\text{max}} = \text{MaxPool}(U)$$

$$M_c = \sigma[f_{c2}(\delta(f_{c1}(Z_{\text{avg}}))) + f_{c2}(\delta(f_{c1}(Z_{\text{max}})))]$$

$$F' = M_c \cdot U$$

式中: AvgPool 表示全局平均池化,求取每个通道的像素平均值; MaxPool 表示全局最大池化,保留每个通道的特征图的像素最大值; f_{c1} 全连接层,输入通道 C ,输出通道 $C/16$; f_{c2} 全连接层,输入通道 $C/16$,输出通道 C ; δ 表示 ReLU 函数; σ 表示 sigmoid 函数。

在空间注意力模块中将通道注意力特征图 F' 作为输入先进行通道维度的平均池化和最大池化,获得两个 $H\times W\times 1$ 的特征图,再将两个特征图拼接后通过 7×7 的卷积层,结果经过 sigmoid 函数,得到二维单通道的空间注意力权重 M_s ,将空间注意力权重与通道注意力特征图相乘,生成同时具有空间与通道注意力的特征图 F'' 。计算过程如下:

$$Z_{\text{avg}} = \text{AVGPool}(U)$$

$$Z_{\text{max}} = \text{MAXPool}(U)$$

$$M_s = \sigma(f_{c3}(Z_{\text{avg}}, Z_{\text{max}}))$$

$$F'' = M_s \cdot F'$$

式中: AVGPool 表示通道维度平均池化,对每个通道的特征图对应像素相加取平均值; MAXPool 表示通道维度最大池化,保留各个通道特征图对应位置的最大像素值; f_{c3} 表示 7×7 的卷积,输出通道为 1。

3.2 边界框回归损失函数改进

边界框回归损失函数用来衡量预测框与真实框的位置关系。YOLACT++ 采用 Smooth L1 损失函数作为边界框回归损失函数,分别对预测框的长、宽、中心点横坐标和纵坐标的偏置进行损失计算,具体计算过程如下:

$$L_{\text{SmoothL1}} = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L1}(t_i^* - t_i)$$

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherswise} \end{cases}$$

$$\mathbf{v} = (t_x^*, t_y^*, t_w^*, t_h^*), \mathbf{u} = (t_x, t_y, t_w, t_h)$$

$$t_x^* = (x^* - x_a) / w_a, t_y^* = (y^* - y_a) / h_a$$

$$t_w^* = \ln(\omega^*/\omega_a), t_h^* = \ln(h^*/h_a)$$

式中： x_a, y_a, ω_a, h_a 表示 anchor 的中心点坐标、长和宽； x^*, y^*, ω^*, h^* 表示真实框的中心点坐标、长和宽； \mathbf{u} 表示网络预测的 anchor 偏置矩阵； \mathbf{v} 表示真实框与 anchor 的偏置矩阵。

由于 Smooth L1 作为边界框回归损失函数对四个偏置量分别计算损失值，算法无法准确描述预测框与真实框之间位置关系。采用 CIoU 损失函数将有助于解决上述问题。边界框回归损失示意图见图 5，定义边界框损失函数：

$$L_{CIoU} = 1 - P_{IoU} + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$
$$P_{IoU} = IoU(b, b^{gt}) = \frac{|b \cap b^{gt}|}{|b \cup b^{gt}|}$$
$$\alpha = \frac{v}{(1 - P_{IoU}) + v}$$
$$v = \frac{4}{\pi^2} \left(\arctan \frac{\omega^{gt}}{h^{gt}} - \arctan \frac{\omega}{h} \right)^2$$

式中： b, b^{gt} 分别表示预测框与真实框的几何中心； ρ 表示两个中心点之间的欧式距离； c 表示预测框和目标框最小外接矩形的对角线长度； α 为用于平衡比例的参数； v 用来衡量预测框和目标框之间的比例一致性。

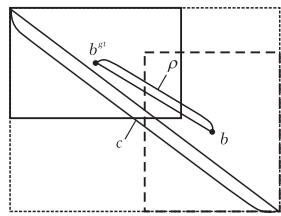


图 5 边界框回归损失计算

网络先进行 anchor 偏置量预测，再将 anchor 经过平移缩放成为预测框 b ，完成边界框回归。利用预测框 b 与真实框 b^{gt} 计算边界框回归损失函数，CIoU 损失函数考虑到了预测框与真实框的交并比 (IoU)、最小外接矩形、几何中心距离 ρ 和长宽比例。Smooth L1 损失函数针对预测框的偏移量进行损失计算。只考虑几何中心距离与长宽比，缺乏对交并比和最小外接矩形的考量。

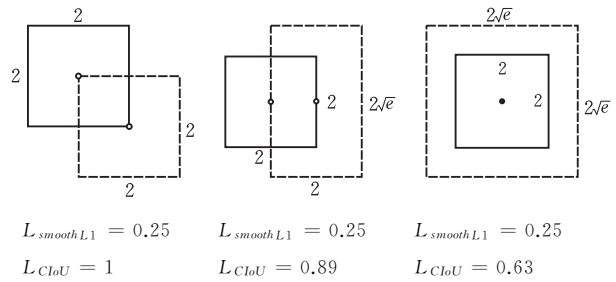


图 6 不同预测框对应的损失函数

图 6 中实线框为真实框，虚线框为预测框。当偏移量 $\mathbf{u} = (0, 0, 0, 0)$ ，即预测框与 anchor 重合时，

用两种边界框损失函数分别计算其对应损失值，可以发现采用 Smooth L1 损失函数计算的损失值均相同，但采用 CIoU 损失函数计算的损失值均不同，预测框与真实框交并比越大、外接矩形越小、几何中心距离越短、长宽比越接近，其损失值越小，预测框与真实框越接近，即边界框回归越准确。故可采用 CIoU 损失函数更加精确地衡量边界框回归性能。

4 实验

4.1 槟榔图片采集与数据集建立

本研究以槟榔果作为研究对象。数据采集地点在益阳市口味王槟榔厂，图像采集设备选用德国 acA1280-60gc Basler ace 工业相机。对每颗槟榔采集如图 7 所示左、中、右三个位置对应不同角度的图片，共采集槟榔图片 1500 张，并对图片采用添加噪声、随机翻转、随机亮度、色彩通道转换这 4 种数据增强技术，将数据集扩充到 6000 张。用 LabelMe 标注工具完成图像的标注工作，将图像的语义分为三类：背景(background)、带花蒂槟榔果(F-nut)、无花蒂槟榔果(nut)对应的标签分别为 0、1、2。标记后的数据集包含原始图像和与之对应 json 标签文件，部分数据处理后的图片见图 8。

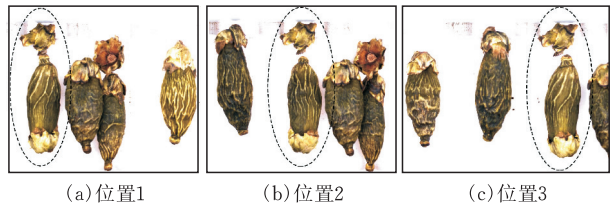


图 7 同一颗槟榔不同位置图像采集

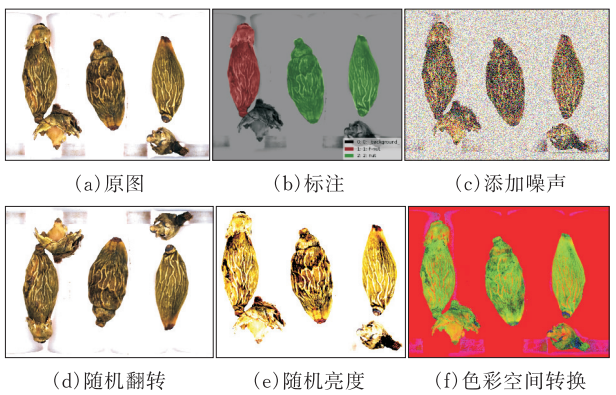


图 8 数据集处理

4.2 实验环境

槟榔实例分割实验硬件配置：操作系统 Windows 10，CPU 为 Intel Core i5-11400，GPU 加速 NVIDIA GTX1080ti 与 CUDA10.0 cuDNN7.0，运行环境 Python3.7，深度学习框架 Pytorch1.2.0。

4.3 评价指标

选用 AP(Average Precision)作为预测框和实

例分割掩模的评价指标。AP 通过计算精确度-召回率曲线(P-R)下方面积得到,由于 IoU 阈值的选取会影响精确度和召回率大小,选用 AP75(IoU=0.75),mAP(IoU=0.50;0.05;0.95)作为评价指标,计算公式如下:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, AP = \int_0^1 p(r)dr$$

式中: TP 表示模型识别为带花蒂槟榔或无花蒂槟榔且与真实标注类别匹配的样本数量;FP 表示模型预测为带花蒂槟榔或无花蒂槟榔且与真实标注类别不匹配的样本数;FN 表示预测为背景但真实标注为带花蒂槟榔或无花蒂槟榔的样本数量。

对网络模型运算速度及模型复杂度采用参数量(Parameter),以及浮点计算量(FLOPs)作为评价标准完成评价。对于一次卷积运算,其模型参数和浮点计算量公式如下:

$$Parameter = (K^2 C_{in}) C_{out} + C_{out}$$

$$FLOPs = [2(C_{in} K^2) C_{out} + C_{out}] H W$$

其中 C_{in}, C_{out} 分别为输入输出特征图通道数, K 为卷积核的大小, $H \times W$ 为输出特征图大小。考虑到激活函数(activation function)以及偏置(bias)等方面的不同,对 FLOPs 的计算方式不尽相同。为便于对比分析,本文采用 Pytorch 中的第三方库 thop,对模型参数量、浮点计算量进行统一计算。

4.4 实验结果分析

实验数据集内共包含槟榔片样本 6000 张,其中用作训练集 4800 张,测试集 1200 张。训练 epoch 为 800 次, GPU 每次处理 4 张图片,使用随机梯度下降(SGD)作为训练时的优化器,初始动量设置为 0.9,学习率为 0.001,权重衰减系数 0.0001。

4.4.1 参数量与计算量对比结果 主干网络的改进将同一特征层的不同通道特征图进行多次多尺度的特征融合,使得高分辨率特征图与低分辨特征图信息相互补充,使得主干能充分提取图片的语义信息,同时造成部分主干网络模型尺寸与参数的增加,边界框回归损失函数改进也在一定程度上增加模型的计算量。网络改进前后网络参数量及总浮点运算量 FLOPs 量化结果见表 3。可以看出网络参数、模型大小、浮点运算量相较于改进前提升 9.8%, 1.1%, 6.6%

表 3 网络改进前后参数量与计算量对比

Model	Parameter Size/MB	Model Size/MB	FLOPs
YOLACT++	42.14	207.6	21.35GFLOPs
本文算法	46.26	209.8	22.76GFLOPs

4.4.2 主干网络改进前后热点图对比 针对主干网络的改进,采用 Grad-CAM 生成主干网络的热力

图(heatmap)用来对主干网络的输出进行可视化分析,生成的主干网络改进前后的热力图见图 9。颜色越亮表示主干网络对于该区域的关注度越强,原主干网络关注的区域包含槟榔的部分区域和背景区域,存在关注区域错误的问题。本文提出改进的主干特征提取网络,加深网络对于图像深层信息的提取能力,采用 CBAM 注意力机制,使得主干网络更关注于槟榔区域,并极大的抑制了背景阴影和掉落花蒂的干扰。

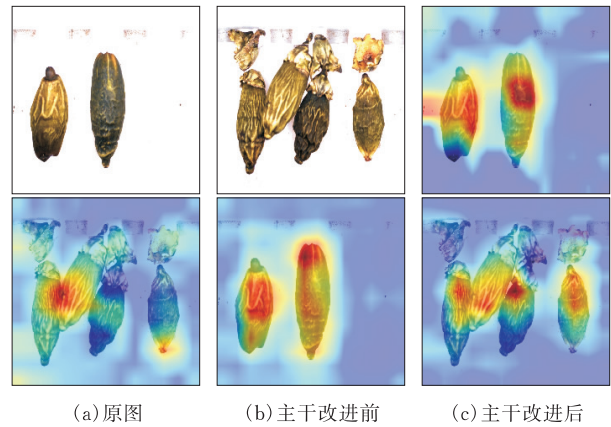


图 9 主干网络热点图

4.4.3 网络分割效果对比 损失值用来衡量网络预测结果与标注之间的误差,损失值越小表示模型预测得越准,网络每一次迭代训练过程中生成损失值,并记录于训练日志中。网络改进前后的掩模损失函数相同,均采用预测掩模与标注掩模的二值交叉熵作为损失函数,故可通过掩模损失值作为衡量网络改进前后的掩模质量。掩模损失折线图见图 10,损失值随着迭代次数的增多而下降,原网络进行 800 次迭代后 loss 值稳定在 0.08 左右。改进后网络损失值最终稳定在 0.01 左右,改进后的模型具有更加良好掩模预测性能。

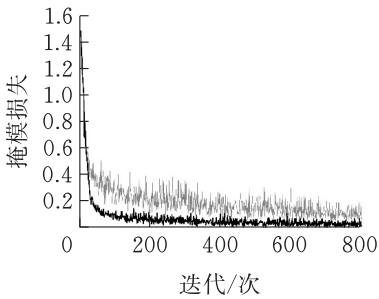


图 10 掩模损失值曲线

从测试集中挑选 3 幅不同条件下的图片来测试槟榔实例分割模型改进前后的分割效果。图 11 可以看出对正常采集图像进行实例分割,掩模会受到预测框区域的限制,SmoothL1 损失函数作为边界框回归的损失函数造成部分槟榔果预测框定位不准确,导致掩模的部分区域无法识别出来。图 12 和图

13 可以看到,原始的 YOLACT++ 算法对传送带上的阴影与脏污和掉落的槟榔花蒂容易造成误分割。本文提出改进的主干特征提取网络,加深网络对于图像深层信息的提取能力,同时使得主干网络更关注于槟榔花蒂与槟榔籽,减少背景阴影与脏污干扰,算法使用 CIoU 作为边界框回归损失函数,使得边界框回归更加准确,一定程度上提升了掩模的分割质量。

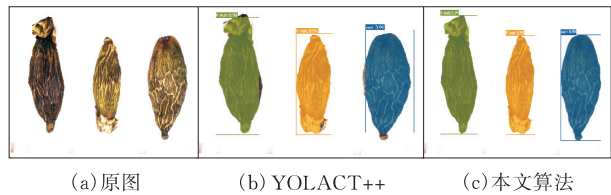


图 11 正常采集

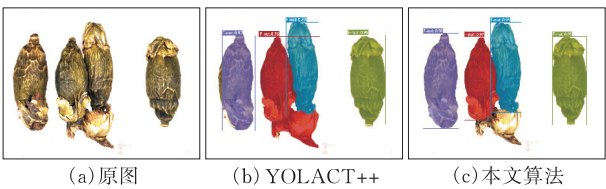


图 12 花蒂掉落及槟榔堆叠

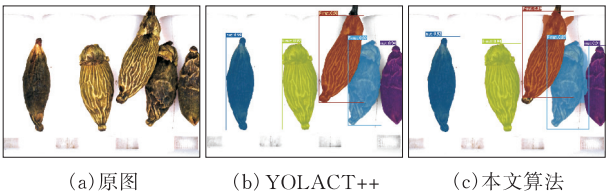


图 13 背景阴影及槟榔堆叠

4.4.4 不同改进点对比 为了验证不同改进点的有效性,做了四组对比实验,结果见表 4。

表 4 不同改进点对比实验结果

实验	主干网络引入改进的 Res2net	边界框回归损失引入 CIoU	检测速度 (帧/s)	预测框 mAP/%	预测框 AP75/%	掩模 mAP/%	掩模 AP75/%
1	否	否	27.1	71.80	82.36	70.61	77.82
2	是	否	26.3	73.53	84.83	72.54	79.71
3	否	是	23.6	74.74	85.29	72.14	79.14
4	是	是	22.4	77.21	87.84	75.81	82.54

通过对比实验 1 与实验 2 可以发现,主干网络的改进对于预测框与掩模的 mAP 均分别提升 1.73%,1.93%。对比实验 1 与实验 3,边界框回归中 CIoU 的引入明显提高了预测框的平均精度,相较于改进前上升 2.94%。对比实验 1 与实验 4,可以发现主干网络与边界框整体改进相较于原算法在预测框 mAP 与掩模 mAP 上分别提升 5.41% 与 5.20%,但检测速度有所损失,降低了 4.7 帧/s。

4.4.5 样本预测结果 用训练好的网络模型进行分级测试,将 A、B、C、D 四个等级的槟榔各两百颗送入传送带进行实例分割与槟榔分级测试,改进前后实验测试结果见表 5。

表 5 改进前/后识别结果

槟榔 样本 级别 数据	改进前/后识别结果				改进前/后 准确率/%
	A 级	B 级	C 级	D 级	
A 级 200	196/199	3/1	1/0	0/0	98.0/99.5
B 级 200	2/0	188/195	6/3	4/2	94.0/97.5
C 级 200	0/0	2/1	192/197	6/2	96.0/98.5
D 级 200	0/0	0/0	2/0	198/200	99.0/100.0

表中改进前/后平均准确率均为 96.75%/98.87%

通过对比可以发现 YOLACT++ 网络的实例掩模由于受到预测框精度限制,导致无法实现完整的掩模分割,造成部分槟榔等级被误分为低等级。由于打光造成的阴影被误分割为槟榔区域,造成槟榔被误分为高等级。而本文算法很好的改善了上述问题,使得槟榔的平均准确率提升 2.12%,达到 98.87%。

4.4.6 不同算法对比 为进一步验证本文算法的实例分割效果,利用槟榔数据集分别对 Mask R-CNN、SOLOv2、YOLACT++ 与本文算法进行训练,用训练好的模型验证槟榔的检测与分割效果。Mask R-CNN 作为 R-CNN 最具代表性的双阶段实例分割算法,SOLOv2 作为单阶段实时实例分割代表,训练结果对比见表 6。从表 6 可以看出 Mask R-CNN 检测速率不佳,SOLOv2 检测精度逊于本文算法,本文算法相较于 Mask R-CNN、SOLOv2 在掩模 mAP 上分别高出 4.09%,2.37%。本文算法相较于 Mask R-CNN 在预测框 mAP 上分别高出 4.90%。

表 6 不同算法分割数据对比

算法	Backbone	Box Loss	检测速度 (帧/s)	预测框 mAP/%	预测框 AP75/%	掩模 mAP/%	掩模 AP75/%
Mask R-CNN	ResNet-101	Smooth-L1	5.8	72.31	83.42	71.72	78.96
SOLOv2	ResNet-101	无	31.2	无	无	73.44	80.26
YOLACT++	ResNet-101	Smooth-L1	27.1	71.80	82.36	70.61	77.82
本文算法	改进的 ResNet-101	L_{CIoU}	22.4	77.21	87.84	75.81	82.54

5 结论

提出了一种基于 YOLACT++ 的槟榔检测模型,对采集的槟榔图像进行标注并采用数据增强方法扩增数据集。在主干网络中引入改进的 Res2Net 模块,加深了主干特征提取能力。并改进了边界框回归损失函数,进而改善了掩模分割效果。测试实验结果表明,训练出的模型在处理花蒂掉落、槟榔堆叠和复杂背景等情况下的槟榔图片均有较好的检测效果,训练后的模型掩模分割效果优于 Mask R-CNN、SOLOv2 网络,槟榔实例分割速度达到 22.4 帧/s,槟榔分级准确率提升了 2.12%,满足工业现场需求。对模型部分参数稍作修改仍可应用于其他工业生产线产品检测,具有较高的应用价值。由于网络的改进造成部分速度损失,后续工作将在保证网络分割质量的前提下优化网络结构,提升分割速度。

[参 考 文 献]

[1] 唐敏敏,宋菲,陈华,等. 槟榔籽提取物抗炎作用及化学成分研究[J]. 热带作物学报,2021,42(7):2035-2042.

[2] OTSU N. A threshold selection method from gray-level histograms [J]. IEEE Transactions on Systems, Man, and Cybernetics, 1979, 9(1): 62-66.

[3] MATAS J, CHUM O, URBAN M, et al. Robust wide-baseline stereo from maximally stable extremal regions [J]. Image and Vision Computing, 2004(10): 761-767.

[4] 姜红花,张传银,张昭,等. 基于 Mask R-CNN 的玉米田间杂草检测方法[J]. 农业机械学报,2020,51(6): 220-228.

[5] 陈佳,刘龙申,沈明霞,等. 基于实例分割的白羽肉鸡体

质量估测方法[J]. 农业机械学报,2021,52(4): 266-275.

[6] 刘文波,叶涛,李硕. 基于改进 SOLO v2 的番茄叶部病害检测方法[J]. 农业机械学报,2021,52(8):213-220.

[7] BOLYA D, ZHOU C, XIAO F, et al. YOLACT++: better real-time instance segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 18(3): 530-536.

[8] BOLYA D, ZHOU C, XIAO F, et al. YOLACT: real-time instance segmentation [C]// proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, 9156-9165.

[9] HE K, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 386-397.

[10] WANG X, ZHANG R, KONG T, et al. SOLO v2: dynamic and fast instance segmentation[C]. In Advances in Neural Information Processing Systems, 2020, 17 721-17 732.

[11] GAO S H, CHENG M M, ZHAO K, et al. Res2Net: a new multi-scale backbone architecture [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(2): 652-662.

[12] 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), 12 993-13 000.

[13] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, 770-778.

[14] WOO S, PARK J, LEE J Y, et al. Cbam: Convolutional block attention module[C]// Proceedings of the European Conference on Computer Vision (ECCV), 2018, 3-19.

Research on Segmentation Algorithm of Areca
Nut Case Based on YOLACT++

SHU Jun¹, WANG Xiang², SHU Xinyi³

(1 School of Electrical and Electronic Engineering, Hubei Univ. of Tech., Wuhan 430068, China ;
2 Hubei Key Laboratory for High-efficiency Utilization of Solar
Energy and Operation Control of Energy Storage System, Wuhan 430058, China ;
3 Wuhan Britain-China School, Wuhan 430000, China)

Abstract: A betel nut detection model based on YOLACT++ deep learning algorithm is designed, aiming at the problem of low accuracy of betel nut grading caused by low segmentation accuracy and inaccurate prediction frame of the betel nut images collected on the delivery belt. The improved Res2Net module is introduced into the model backbone network to improve the accuracy of the betel nut mask segmentation. The CIoU loss function is introduced into the model bounding box regression to improve the detection accuracy of the prediction box. The results show that the mask mAP of the improved model is 5.20%, 4.09%, and 2.37% higher than that of YOLACT++, Mask R-CNN and SOLOv2, respectively. Compared with YOLACT++ and Mask R-CNN, the prediction frame mAP is 5.41% and 4.90% higher respectively. Compared with the model improvement before the classification accuracy rate is increased by 2.12%

Keywords: YOLACT++ ; betel detection ; res2net module ; ciou loss function

[责任编校: 张岩芳]