

基于改进 YOLACT 的堆垛图像快速分割方法研究

苏铁明¹, 李鹏博¹, 徐志祥¹, 梁琛¹, 王宣平¹, 刘玮²

(1. 大连理工大学 机械工程学院, 辽宁 大连 116024;
2. 久亿航宇科技(大连)有限公司, 辽宁 大连 116085)

摘要: 针对堆叠密集的堆垛货箱出现的漏检情况以及难以分割出每个货箱的精确边缘而造成的难以准确抓取的问题, 对深度学习实例分割算法 YOLACT 进行了相应的改进; 使用工业相机采集货箱的堆垛图像, 利用 Labelme 标注图像制作数据集, 并且通过数据增强方法扩充数据集; 为了提高模型的分割准确率, 分别对掩码真值和 YOLACT 中的原型掩码输出分支 (Protonet) 的预测掩码使用 Canny 边缘检测算子, 并取二者的二值交叉熵损失作为损失函数加入到原网络中训练; 使用训练好的最优模型对测试集图像数据进行试验; 结果表明, 改进后的模型预测掩码 $mAP_{0.5,0.95}$ 可以达到 0.543, 比原模型提高 2.2%, 同时货箱边缘的分割精度也得到了提升, 模型推理速度可达 10.2 帧/秒, 可以满足精度要求和生产节拍要求。

关键词: 堆垛; 边缘检测; YOLACT; Canny; 损失函数

Research on Fast Segmentation Method of Stacking Image Based on Improved YOLACT

SU Tieming¹, LI Pengbo¹, XU Zhixiang¹, LIANG Chen¹, WANG Xuanping¹, LIU Wei²

(1. School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China;
2. Jiuyi Aerospace Technology (Dalian) Co., Ltd., Dalian 116085, China)

Abstract: Aimed at the problems of missing detection on densely stacked packing boxes and difficult to accurately capture due to segmenting the inexact edges of each packing box, the deep learning instance segmentation algorithm YOLACT was improved. The industrial camera was used to collect the stacking image of the packing box, the Labelme was used to annotate the image and create the dataset, and the dataset was expanded through the data enhancement method; In order to improve the segmentation accuracy of the model, the Canny edge detection operator was used for the mask truth value and predicted mask of the prototype mask output branch (Protonet) in YOLACT, respectively, and the binary cross-entropy loss of them was added to the original network as a loss function; The trained optimal model was used to test the image data of the test set. The results show that the predicted mask $mAP_{0.5,0.95}$ of the improved model can reach 0.543, which is 2.2% higher than that of the original model. At the same time, the segmentation accuracy of the packing box edge is also improved to a certain extent. The inference speed of the model can reach 10.2 frames/second (FPS), which can meet the requirements of the segmentation accuracy and cycle time in the production.

Keywords: stacked packing boxes; edge detection; YOLACT; Canny; loss function

0 引言

随着进出口贸易和电商的蓬勃发展, 货物运输、快递物流呈现了前所未有的爆发式增长^[1]。大量的货物依靠人工来搬运需要大量的劳动力^[2], 因此就需要自动化的搬运拆垛设备来代替人工。而自动化拆垛技术中视觉检测又是核心技术, 视觉技术的好坏决定着自动化拆垛工作能否顺利进行。针对这些问题, 研究者们开展了深入研究。文献 [3] 将激光三角测量视觉系统集成到机器人纸袋拆垛的任务中, 利用三维图像的光照不敏感性和目标三维几何的先验知识, 对不同颜色、材质、形状等目标进行拆垛。文献 [4] 将多分辨率面元模型用于拆垛任务中, 用来实现目标检测、定位和验证。文献 [5] 针对不稳定光照条件下的

编织袋定位问题利用区域生长方法提取原始图像中的整个目标区域, 建立图像强度与最佳区域生长阈值之间的关系模型; 然后, 采用渐进概率霍夫变换对编织袋进行定位。文献 [6] 提出一种基于单目视觉的工业机器人拆垛系统, 基于形状模板匹配法提出目标图像识别算法, 将三维视觉模型与机器人运动学模型进行信息融合从而引导机器人抓取散乱摆放的目标。文献 [7] 基于 RGB-D 相机采集到的深度信息, 设计了通过蜗壳摆放背景识别反推出各蜗壳所在位置的算法, 然后推导出深度图像像素坐标和机器人空间坐标的相对关系并使用带有反馈的抓取结构实现蜗壳工件的拆垛工作。

本文对于堆叠密集的堆垛货箱的拆垛工作中的视觉

收稿日期: 2023-02-17; 修回日期: 2023-03-15。

作者简介: 苏铁明(1972-), 男, 博士, 讲师, 硕士生导师。

引用格式: 苏铁明, 李鹏博, 徐志祥, 等. 基于改进 YOLACT 的堆垛图像快速分割方法研究[J]. 计算机测量与控制, 2023, 31(12): 210-215.

引导方案选取深度学习实例分割算法 YOLACT 作为基础框架。实例分割的目的是得到图像上每个像素点所属类别以及相同类别的不同个体^[6], 是图像处理领域最具挑战的任务之一^[9]。YOLACT 由文献^[10]在 2019 年提出, 它在单阶段目标检测网络^[11-12]上增加了掩码生成分支, 是一种单阶段实例分割网络^[13], 相比于两阶段实例分割网络, YOLACT 具有更高的推理速度, 适用于工业场景下的实时检测任务^[14]。为了让 YOLACT 更适用本文工作, 引入 Canny 算子^[15]对 YOLACT 模型的原型掩码分支 Protonet 的输出和掩码真值进行边缘检测处理。Canny 是一种边缘检测算子, 具有易于运行^[16], 抗干扰能力强和边缘定位准确等优点^[17]。最后, 将 Protonet 的输出和掩码真值的边缘检测处理结果的二值交叉熵损失作为新的边缘损失函数加入到原网络进行训练以达到提升模型准确率的目的。

1 设备与工作流程

1.1 拆垛机器人结构及工作流程

本文的视觉方案基于的是一种桁架机器人结构, 如图 1 所示, 托盘对准货箱的边缘之后吸盘进行抓取, 前端的托盘和吸盘结构可以实现更高效的抓取。

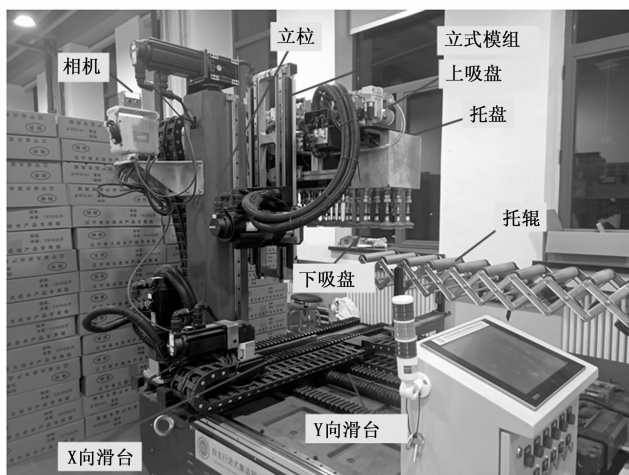


图 1 桁架机器人结构图

该机器人主要结构包括可以旋转的立柱、可以实现 x , y 向运动的滑台、可以实现 z 向运动的立式模组、托盘、上吸盘结构、下吸盘结构和托辊结构。首先相机进行拍照并处理计算出货箱的坐标值, 将坐标值发送给桁架机器人, 桁架机器人收到目标坐标值后, 桁架机器人的立柱旋转 90° 并向前移动到指定位置, 然后立式模组通过上下运动来使托盘运动到指定货箱的下边缘位置, 然后伸出吸盘进行抓取, 最后立柱再次旋转 90° 将货箱放置托辊上完成整个抓取流程, 工作原理如图 2 所示。

为了减少托盘结构在抓取过程中位置低于目标货物的边缘过多或高于目标货物的边缘过多而导致的抓取失败甚至是扯坏货箱的情况, 以及左右边缘与实际相差过大导致的抓偏情况, 拥有较高的边缘检测精度就显得十分重要, 本文在 YOLACT 中引入 Canny 边缘检测算子来提高掩码分

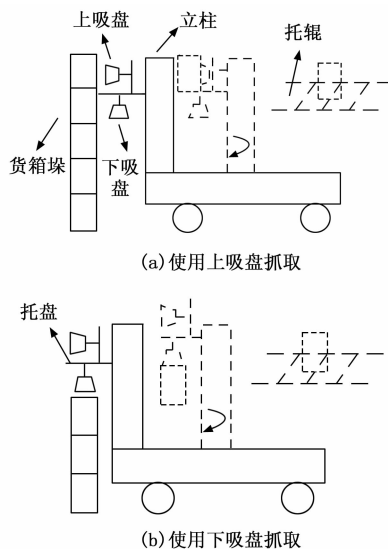


图 2 桁架机器人工作原理图

割的准确率以及掩码边缘的准确率。

本文的图像采集设备选用的是 SICK-Visionary-Tmini 工业相机, 可同时输出二维图像与三维点云, 分辨率为 512×424 , 接口为 gigE (千兆以太网)。

1.2 图像采集

本文采集的图像一共 407 张, 数据集包含着分别按照单一垛型堆放和混合垛型堆放的不同种类货箱。在采集时还针对同一种垛型采用不同的倾斜角度拍摄和采取不同的摆放方案来增加数据集的复杂性。收集好数据集后采用标注工具 Labelme 对数据集进行标注, Labelme 可以通过创建多边形来对要标注的目标进行轮廓的精准标注。将标注生成的 JSON 标注文件保存并转化成 COCO 数据集的格式, 在训练前对数据集进行随机按比例划分, 先按照 9:1 的比例划分为训练集和测试集, 再将训练集按 9:1 划分为真正的训练集和验证集, 最后训练集、验证集和测试集数量分别为 330 张, 37 张和 40 张。在训练时, 训练集和验证集在加载进网络时进行了随机色域调整 (为了适应不同光照条件时的检测)、将原图放入一张较大的图中 (利于小目标检测)、随机裁剪、镜像翻转和随机大小调整等 5 种数据增强操作, 将训练集扩充为 1 650 张, 验证集扩充为 185 张。

2 算法原理及改进

由于堆垛堆叠密集且摆放随意, 加之拍摄时受外界亮度、拍摄距离等因素的影响, 这样导致相机采集出来的堆垛图像存在着相邻的货箱之间分界模糊的现象, 难以在原始 YOLACT 算法模型的基础上很好地分割开, 这样产生的像素坐标误差就会带入到世界坐标误差, 最终会导致桁架机器人抓取失败。损失函数作为神经网络正向训练过程中的约束条件, 表示预测值和真实值之间的差距, 网络在训练过程中会不断缩小这个差距。在这里对 YOLACT 的 Protonet 分支的输出和掩码真值使用 Canny 边缘检测算子, 并对二者取二值交叉熵损失加入到网络中训练, 这样就约束

了边缘这一因素，不仅可以提高模型准确率还可以提高边缘的分割精度，总体思路如图 3 所示。

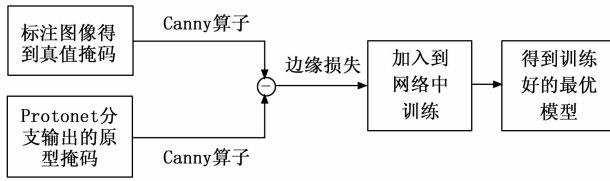


图 3 总体思路

2.1 YOLACT 网络结构

YOLACT 设计了两个分支结构 Prediction Head 和 Protonet 来并行的解决定位和分割这两个任务。Prediction Head 分支生成各候选框的类别置信度、锚框的定位和原型掩码的系数；Protonet 为每张图片生成 k 个原型掩码。原型掩码的数量和其系数的数量相等，并通过二者的线性组合得到最终的掩码 mask。

YOLACT 的主干网络是 ResNet-50，ResNet-50^[18] 是一种残差神经网络，由两个基本残差块 Conv Block 和 Identity Block 组成，Conv Block 用来改变网络的维度，Identity Block 用来加深网络，残差结构能将靠前的若干层的某一层数据输出直接跳出多层引入到后面数据层的输入部分，极大地消除了随着网络变深效果反而变差这种现象^[19]。

主干网络进行特征提取后，对主干网络逐层卷积得到的三组不同维度的特征图 (C3、C4 和 C5)，C3、C4 和 C5 再进行上采样，融合等操作构成由 P3、P4、P5、P6 和 P7 组成的特征金字塔 FPN。FPN 因为融合了不同维度的特征层，而不同维度的特征层有着不同大小的感受野，这使得网络模型对于不同大小的待检测目标都有一个很好的适应性。P3、P4、P5、P6 和 P7 输入给 Prediction Head，Prediction Head 包含两层卷积结构，FPN 得到的特征层经过这两层卷积层结构会生成类别置信度、锚框定位和掩码系数；P3 输入给 Protonet；Protonet 由 4 个 3×3 的卷积层和一个 1×1 卷积层组成，经过最后的一层 1×1 卷积后会生成 k 个原型掩码，这 k 个原型掩码和 Prediction Head 输出的 k 个掩码系数进行线性组合，这个线性组合的结果再经过裁剪和阈值化等操作生成最终的掩码 mask。YOLACT 网络整体结构如图 4 所示。

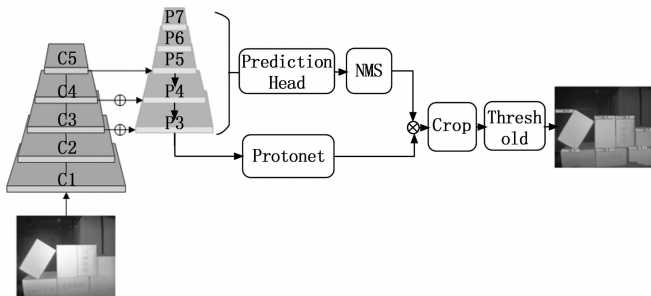


图 4 YOLACT 网络结构

2.2 Canny 算子原理

Canny 边缘检测算法是由 Canny 在 1986 年提出的一种多级边缘检测算法，具有定位准^[20]、精度高等优点。Canny 算子主要包含以下 4 个步骤：

2.2.1 高斯平滑滤波

高斯平滑滤波根据待滤波的像素点及其邻域点的灰度值按照一定的参数规则进行加权平均，这样可以有效滤去理想图像中的高频噪声。本文使用的高斯核函数为离散化二维高斯核函数：

$$Kernel = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

2.2.2 计算梯度大小和方向

Canny 算子中使用 Sobel 算子作为梯度算子，Sobel 算子含有 x 方向和 y 方向两个算子模板分别为：

$$S_y = \begin{bmatrix} 1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, S_x = \begin{bmatrix} 1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2)$$

将这两个模板分别与灰度图像矩阵 I 做互相关计算得到 x 方向像素梯度 $Grad_x$ 和 y 方向像素梯度 $Grad_y$ ，然后经式 (4) 得到该像素点梯度。梯度变化越大，说明相邻像素间差值较大，可能是边缘所在位置。

$$Grad_x = S_x * I, Grad_y = S_y * I \quad (3)$$

$$Grad = \sqrt{Grad_x^2 + Grad_y^2} \quad (4)$$

2.2.3 非极大值抑制

通过以上步骤得到了由多个像素组成的较粗的边缘，我们默认边缘是单个像素连成的，因此需要进行非极大值抑制来将抑制局部梯度值不是最大的点，主要包含以下步骤：将每一点梯度方向近似为 $[0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ]$ ，比较该像素点和该点正负方向的像素点梯度（比较范围一般为八邻域），如果该像素点梯度值最大则保留，否则置为 0。

2.2.4 双阈值连接

通过设定高低阈值来确定强弱边缘，若该像素点灰度值大于高阈值，则将该点灰度置为 255，认为是强边缘，若大于低阈值小于高阈值则认为是弱边缘，若小于低阈值则认为不是边缘舍去。

2.3 损失函数的改进

YOLACT 在正向训练时用到的损失函数一共包括四部分组成，分别是定位损失，掩码系数损失，掩码损失和语义损失。

定位损失函数为 Smooth-L1 函数，如下所示：

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & |x| \geq 1 \end{cases} \quad (5)$$

$$L_{loc} = \sum_{i \in \{x, y, w, h\}} smooth_{L1}(t_i - v_i) \quad (6)$$

$v = (v_x, v_y, v_w, v_h)$ 为真实定位框的中心点坐标和长宽。

$t = (t_x, t_y, t_w, t_h)$ 为预测框的中心点坐标和长宽。

掩码系数损失函数如下：

$$y_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (7)$$

$$L_{\text{coef}} = H_{y_i}^{\wedge}(y_i) = - \sum_i y_i \log(y_i) \quad (8)$$

式 (8) 中, $y_i = (Y_1, Y_2, \dots, Y_n)$ 为预测的每一个类别的概率, 将这个输出与真实的标签值 \hat{y} 做交叉熵, 显然 y_i 越接近 1 结果值越小。

掩码损失函数和语义损失函数均为二值交叉熵函数:

$$L_{(\text{mask}, \text{semantic})} = - \frac{1}{n} \sum_{i=1}^n y_i \cdot \log \hat{y} + (1 - y_i) \cdot \log(1 - \hat{y}) \quad (9)$$

n 为预测的类别个数, 其余符号与掩码系数损失函数符号意义相同。所以, 原始 YOLACT 的总体损失函数如下:

$$L_{\text{total}} = 1.5 \times L_{\text{loc}} + L_{\text{coef}} + 6.125 \times L_{\text{mask}} + L_{\text{semantic}} \quad (10)$$

最后, 在 YOLACT 损失函数的基础上再加入边缘损失函数, 由于边缘是在掩码的基础上得到的, 因此边缘损失函数仍然采用二值交叉熵函数, 函数公式中的原本预测掩码和真值掩码的地方替换成 Canny 算子提取到的预测值边缘和真实值边缘, 如式 (11) 所示:

$$L_{\text{edge}} = - \frac{1}{n} \sum_{i=1}^n C_{y_i} \cdot \log C_{\hat{y}_i} + (1 - C_{y_i}) \cdot \log(1 - C_{\hat{y}_i}) \quad (11)$$

C_{y_i} 为对掩码损失中的预测掩码进行边缘提取之后的结果。

$C_{\hat{y}_i}$ 为对掩码损失中的真值掩码进行边缘提取之后的结果。

特别注意的是, 边缘也是相当于一种细化的掩码, 因此边缘损失应当享有和掩码损失相同的权重系数。综上所述, 得到改进后的 YOLACT 损失函数:

$$L_{\text{total}} = 1.5 \times L_{\text{loc}} + L_{\text{coef}} + 6.125 \times L_{\text{mask}} + L_{\text{semantic}} + 6.125 \times L_{\text{edge}} \quad (12)$$

3 试验结果与分析

3.1 试验条件

本文的改进 YOLACT 实例分割算法训练和测试的模型均运行在 Windows10 操作系统的个人计算机上, 配置为英特尔 2.60 GHz 六核 CPU, 16 GB 内存, 显卡为 NVIDIA GeForce RTX2070S, 深度学习框架为 TensorFlow1.14 和 Keras2.1.5。

3.2 评价指标

为了衡量改进后的模型是否有提升, 这里的评价指标采用 mAP 均值平均精度, mAP 的计算需要使用到查准率 (Precision) 和查全率 (Recall)。查准率代表预测为真的样本中正样本的比例, 查全率代表在所有真实情况为正的样本中预测为真的样本数比例。查准率和查全率计算好得到一条 $P-R$ 曲线, 这条曲线的面积就是这一类的准确率, 也就是这一类的 mAP 。 mAP 通常需要在限定一个阈值的情况下使用, 这个阈值就是 IoU 交并比, 表示预测值和真实值的重叠程度, 只有重叠程度大于这个阈值才会被认为是真

的正样本。具体公式如下:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (13)$$

$$\text{Recall} = \frac{TN}{TN + FP} \quad (14)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N \int_0^1 P(R) dR \quad (15)$$

其中: TP 表示预测为真的正样本, FN 表示预测为假的负样本, FP 表示预测为假的正样本, N 表示取间隔点的个数。

3.3 试验结果分析

本试验选用了两种模型进行比对: 使用原始 YOLACT 算法, 用 YOLACT 表示; 使用 Canny 边缘检测算子优化的 YOLACT, 用 YOLACT-Canny 表示。在这里为了保证试验结论可靠性, 对两种模型采取相同的超参数设置, 超参数设置如表 1。

表 1 超参数设置表

超参数	数值/类型
学习率	2×10^{-3}
Momentum	0.937
优化器	SGD
迭代次数	8 200
Batchsize	2
主干网络	ResNet-50

接下来对两种模型的训练过程在验证集上的损失值可视化处理如图 5 所示。

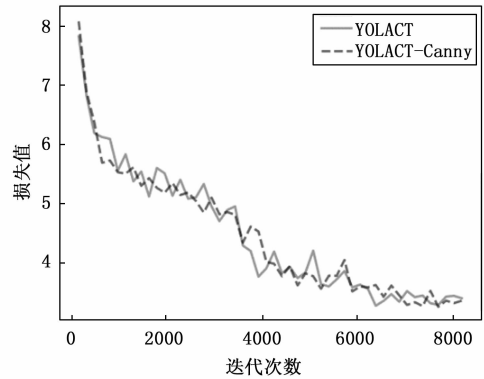


图 5 Loss 值随着迭代次数变化曲线图

可以看出两种模型大约都在 6 000 次迭代后逐渐趋于稳定。这里取两种模型在训练过程中每隔相同步长保存的 10 个权重对测试集进行评估, 二者的评估指标在这里采用 $mAP_{0.5,0.95}$ 和 $mAP_{0.5}$ 。 $mAP_{0.5,0.95}$ 代表在不同 IoU 阈值 (从 0.5 到 0.95, 步长 0.05) 上的平均准确率, $mAP_{0.5}$ 则是代表 IoU 阈值大于 0.5 时的模型准确率, 二者的对比如图 6~9 所示。

从图 6 和图 7 可以看出, YOLACT-Canny 模型的预测掩码的准确率数值不论是以 $mAP_{0.5,0.95}$ 为指标, 还是以 $mAP_{0.5}$

表 2 二者最优模型性能比较

模型	$mAP_{0.5;0.95}(\text{mask})$	$mAP_{0.5}(\text{mask})$	$mAP_{0.5;0.95}(\text{box})$	$mAP_{0.5}(\text{box})$	推理速度/FPS	模型大小/MB
YOACT	0.521	0.863	0.672	0.897	10.2	117
YOACT-Canny	0.543	0.876	0.675	0.907	10.2	117

为指标，在训练迭代的过程中整体都是高于 YOACT 的。YOACT-Canny 的最优模型掩码 $mAP_{0.5;0.95}$ 比 YOACT 的最优模型要高 2.2%， $mAP_{0.5}$ 比 YOACT 的最优模型高 1.3%；而从图 8 和图 9 中可以看出预测框 mAP 值与 YOACT 相比几乎没有改变，说明本文中的改进方法只对掩码有效，并没有对预测框产生影响。二者训练过程中得到的最优模型性能比较如表 2 所示，可以看出本文提出的这种修改损失函数的方法可以在不改变模型大小、不影响推理速度的前提下提升模型的准确率。分别取各自最优模型的部分测试效果对比如图 10 所示。

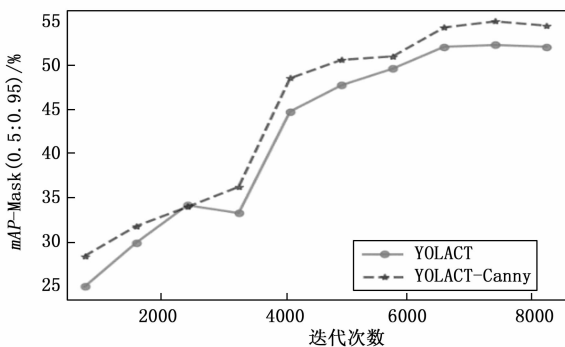


图 6 掩码 $mAP_{0.5;0.95}$ 随迭代次数变化

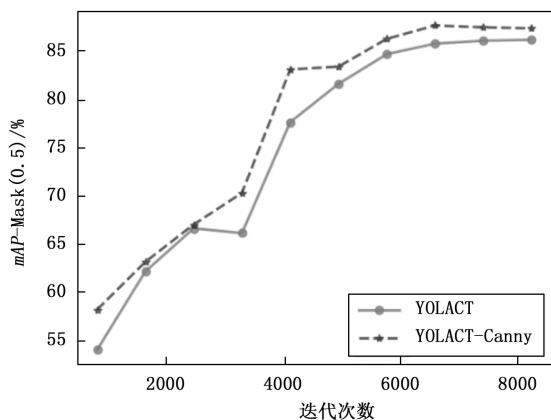


图 7 掩码 $mAP_{0.5}$ 随迭代次数变化

从图 10 (a) 和 (b) 中的前两组可以看出，二者分割出的货箱个数和位置一致，但 YOACT-Canny 对于边缘的描绘明显比 YOACT 要更准确；首先图 (a) 的第一组图中的中心位置附近的箱子存在着和附近的箱子之间的垂直方向间隙过大的问题，这样会导致桁架机器人使用上吸盘或下吸盘抓取的时候出现托盘或下吸盘无法与箱子边沿对齐而导致抓取失败的问题，而 YOACT-Canny 由于边缘

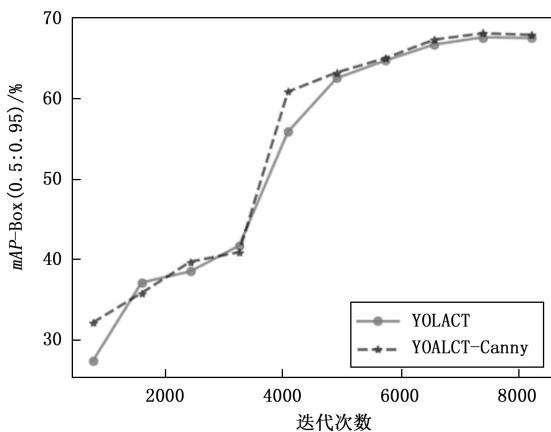


图 8 预测框 $mAP_{0.5;0.95}$ 随迭代次数变化

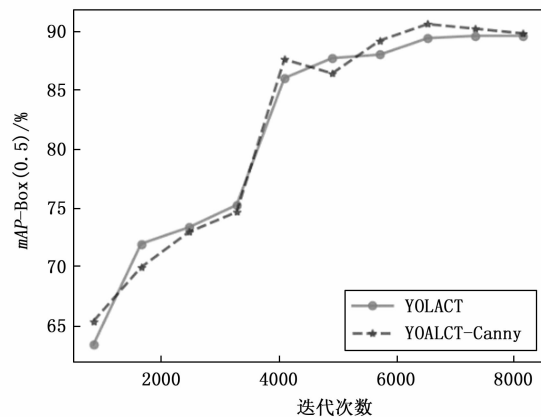


图 9 预测框 $mAP_{0.5}$ 随迭代次数变化

损失函数的存在使得这部分的箱子边缘平滑且准确。在图 (a) 第二组图中，有箱子的右侧出现了一部分掩码缺失，这样会影响桁架机器人对箱子的横向定位，横向的运动直接会影响抓取箱子时吸盘附着的个数，吸盘附着的少导致吸力不足会抓取失败。在后两组测试图中，YOACT-Canny 与 YOACT 对于边缘的描绘相差不多，但是 YOACT 分别在两组图中都存在漏识别现象，而 YOACT-Canny 则不存在漏识别现象。漏识别会对生产节拍产生影响，一旦出现漏识别就需要人工来进行干预，这在生产制造的过程中是影响生产效率的，而 YOACT-Canny 解决了这一问题，从后两组测试效果图中可以看出，无论是在堆放规则的单一垛型还是在堆放随意的混合垛型中，YOACT-Canny 都可以很好地解决漏识别这一问题。

4 结束语

本文提出了针对一种特殊的托盘式拆垛结构的视觉引



图 10 改进前后 YOLACT 分割效果展示

导部分工作的新方法, 将实例分割网络 YOLACT 作为基础检测算法, 引入 Canny 边缘检测算子对真值掩码和 YOLACT 的 Protonet 输出的预测掩码分别进行边缘检测, 并将二者的结果做二值交叉熵处理得到新的边缘损失函数加入到网络中训练, 最终对改进后的算法模型进行试验。试验结果表明, 改进后的模型在测试集上的预测掩码 $mAP_{0.5;0.95}$ 可以达到 0.543, $mAP_{0.5}$ 可以达到 0.876, 分别比原模型提高 2.2% 和 1.3%; 推理速度平均可达 10.2 帧/秒, 检测效率高, 检测准确率高, 可以满足生产节拍要求以及识别的准确性要求。

参考文献:

- [1] 徐凯翔. 基于三维视觉技术的混合拆垛分析与定位方法研究 [D]. 哈尔滨: 哈尔滨工业大学, 2019.
- [2] 马继成. 新型交叉带分拣系统的设计与研究 [D]. 合肥: 安徽理工大学, 2020.
- [3] ZHANG B, SKAAR S B. Robotic de-palletizing using uncalibrated vision and 3D laser-assisted image analysis [C] // Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009: 3820 - 3825.
- [4] HOLZ D, TOPALIDOU-KYNIASOPOULOU A, STÜCKLER J, et al. Real-time object detection, localization and verification for fast robotic depalletizing [C] // Proceedings of the 2015

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015: 1459 - 1466.

- [5] LI J, KANG J, CHEN Z, et al. A workpiece localization method for robotic de-palletizing based on region growing and PPHT [J]. IEEE Access, 2020, 8: 166365 - 76.
- [6] 闫九祥, 赵永国, 张艳芳, 等. 基于单目视觉的工业机器人拆垛系统设计与实验 [J]. 机床与液压, 2020, 48 (3): 18 - 23.
- [7] 杨平, 王博, 骆凯鑫. 基于机器视觉的蜗壳智能识别拆垛系统 [J]. 组合机床与自动化加工技术, 2022 (11): 38 - 43.
- [8] LI L, FANG M, FU F. Instance Segmentation Based on Improved YOLACT [C] // 2020 International Conference on Virtual Reality and Visualization (ICVRV), 2020: 165 - 170.
- [9] HARIHARAN B, ARBELAEZ P, GIRSHICK R, et al. Simultaneous detection and segmentation [C] // Computer Vision-ECCV 2014, PT VII. 2014: 297 - 312.
- [10] BOLYA D, ZHOU C, XIAO F, et al. YOLACT: real-time instance segmentation [C] // 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019: 9156 - 9165.
- [11] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection [C] // Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 779 - 788.
- [12] REDMON J, FARHADI A. YOLO9000: Better, Faster, Stronger [C] // Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 6517 - 6525.
- [13] LEE S, KIM H. MPQ-YOLACT: mixed-precision quantization for lightweight YOLACT [C] // Proceedings of the 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020: 1 - 4.
- [14] 徐胜军, 李康平, 韩九强, 等. 基于多尺度特征注意 Yolact 网络的堆叠工件分拣算法 [J]. 计算机测量与控制, 2022, 30 (9): 184 - 192, 200.
- [15] CANNY J. A computational approach to edge detection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, 8 (6): 679 - 98.
- [16] 刘宇涵, 闫河, 陈早早, 等. 强噪声下自适应 Canny 算子边缘检测 [J]. 光学精密工程, 2022, 30 (3): 350 - 62.
- [17] JI D, LIU Y, WANG C. Research on image edge detection based on improved canny operator [C] // 2022 3rd International Conference on Information Science, Parallel and Distributed Systems (ISPDS), 2022: 229 - 232.
- [18] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C] // Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 770 - 778.
- [19] 郭明秀, 杨伟, 刘琦, 等. 残差网络研究综述 [J]. 计算机应用研究, 2020, 37 (5): 1292 - 7.
- [20] 余波, 吴静, 周琦宾. 一种基于改进 Canny 算子的边缘检测算 [J]. 制造业自动化, 2022, 44 (8): 24 - 26, 43.