

基于目标和关键点检测的单目托盘定位^①

周恒森, 朱 明

(中国科学技术大学 信息科学技术学院, 合肥 230026)

通信作者: 周恒森, E-mail: wzz233@mail.ustc.edu.cn



摘 要: 托盘的识别与定位是无人叉车中关键的问题之一. 当前托盘定位多采用目标检测的方法, 然而目标检测只能识别托盘在图像中的位置, 无法得到托盘的空间信息. 针对此问题, 本文提出了一种基于目标和关键点检测的单目托盘定位方法, 用于检测托盘并计算托盘当前的倾角和距离. 首先对托盘进行目标检测, 然后将检测的结果进行裁剪后输入到关键点检测网络中. 通过对托盘关键点的检测和托盘固有的几何外形特征, 设计边缘自适应调整, 得到高精度的托盘轮廓信息. 根据几何约束提出了基于轮廓点的托盘倾角与距离计算方法, 并采用 RANSAC 算法提升了计算结果的精度和稳定性, 解决了托盘的定位问题. 实验表明, 本文提出的算法在倾角计算上平均误差在 5° 以内, 水平距离计算上平均误差在 110 mm 以内, 能较好地定位托盘, 具有较高的实用价值.

关键词: 托盘定位; 位姿估计; 目标检测; 距离计算; 倾角计算

引用格式: 周恒森, 朱明. 基于目标和关键点检测的单目托盘定位. 计算机系统应用, 2023, 32(8): 180-188. <http://www.c-s-a.org.cn/1003-3254/9181.html>

Pallet Positioning Based on Target and Key Points Detection with Monocular Vision

ZHOU Heng-Sen, ZHU Ming

(School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China)

Abstract: Pallet recognition and positioning is one of the critical problems in unmanned forklift trucks. At present, target detection is mostly used for pallet positioning. However, target detection can only recognize the position of the pallet in the image and cannot obtain the spatial information of the pallet. To solve this problem, this study proposes a pallet positioning method based on target and key point detection with monocular vision, which is applied to detect the pallet and calculate the current dip angle and distance of the pallet. Firstly, target detection is carried out on the pallet. Then, the image will be cropped according to the detection result and input into the key points detection network. Through the detection of the key points and the inherent geometric features of the pallet, the edge adaptive adjustment is designed to obtain the high-precision profile information of the pallet. According to the geometric constraints, a method for calculating the dip angle and distance of the pallet based on contour points is proposed, and the RANSAC algorithm is adopted to improve the precision and stability of the calculation results, thus addressing the problem of pallet positioning. Experiments indicate that the average error of the proposed algorithm is less than 5° in the calculation of dip angle and less than 110 mm in the calculation of horizontal distance. It works well for pallet positioning and is of high practical value.

Key words: pallet positioning; pose estimation; target detection; distance calculation; dip angle calculation

1 引言

随着现代信息科技的发展, 人工智能技术突飞猛

进, 人们着手于将人工智能应用于机器人来解决诸如物流等领域中的问题. 在物流系统中, 货物一般通过托

① 基金项目: 科技创新特区计划 (20-163-14-LZ-001-004-01)

收稿时间: 2023-01-06; 修改时间: 2023-02-09; 采用时间: 2023-03-03; csa 在线出版时间: 2023-05-22

CNKI 网络首发时间: 2023-05-24

盘这一载体进行运输,移动托盘是自动化仓储中 AGV 设备的常见任务.2022 年《中共中央国务院关于加快建设全国统一大市场的意见》中也提到了“推广标准化托盘带板运输模式”^[1].因此,托盘定位是自动化仓储中不可或缺的环节.

根据采集数据所使用的传感器类型不同,托盘定位可以分为以下两类.

(1) 基于激光雷达的托盘定位.赵俊宏等人^[2]采用 2D 激光雷达获取数据,然后使用模板匹配的方法进行位姿估计,鲁棒性较差.Mohamed 等人^[3]采用深度学习结合 2D 激光雷达进行托盘定位,但因为激光数据不够密集,托盘位姿估计精度低.Bellomo 等人^[4]使用点云图像与激光雷达相结合的形式,由于点云数据处理需要耗费大量的时间,难以保证算法的实时性.并且激光雷达价格相对高昂,难以大规模普及.

(2) 基于视觉的托盘定位.基于视觉的方法主要可以分为传统图像处理方法和基于深度学习的方法.在传统图像处理上 Chen 等人^[5]将图像从 RGB 空间转换到 HSV 空间,利用托盘的几何形状确定托盘位置.Cui 等人^[6]通过分析托盘的颜色、边缘、角点等信息直接检测托盘,这种方法只能适用于特定形状和颜色的托盘.武文汉^[7]使用 ToF 相机和纸标签进行 3D 点云模板匹配来进行托盘定位,虽然能保证准确性和鲁棒性,但无法满足实时性.Xiao 等人^[8]先对点云数据进行平面分割再使用模板匹配来估计托盘的位姿,但该算法对于相应的视觉传感器性能要求严格.利用传统图像算法来进行托盘定位鲁棒性低,实际场景下容易受到环境噪声干扰导致实用性不高.深度学习的方法主要是使用深度卷积神经网络完成对于托盘的检测.Mok 等人^[9]采用多任务分类的方式进行托盘的定位,将托盘按照偏转角度和距离分为多个类别,该方法只能用于特定场景,不具备实用性.叶伟^[10]使用了深度学习结合传统方法的想法,先使用 YOLOv4^[11]算法对托盘进行目标检测,得到托盘在图像中的位置后,再使用 GrabCut^[12]进行分割,以此来定位托盘,虽然该方法的托盘检测效果较好,但是无法获得空间信息.朱丹平等^[13]采用了改进 CenterNet^[14]的方法检测托盘,并利用托盘几何约束估计托盘偏转方向,效果较好,但关键点数量较少,无法得到托盘的距离和倾角信息.

在实际的托盘运输过程中,叉车需要完成对于托

盘的精准对接,这就需要知道托盘与叉车之间的距离以及托盘与叉铲的倾角,保证叉车可以正确调整车身实现对于托盘的安全叉取.针对此问题,本文提出了一种基于目标和关键点检测的单目托盘定位方法.由目标检测和关键点检测得到托盘的轮廓信息,通过轮廓信息与几何约束计算托盘的距离和倾角.

使用传统图像的方法在进行托盘检测容易受到光照等环境因素影响,导致检测鲁棒性低,托盘检测失败的情况发生.而深度学习的托盘检测由于使用深度神经网络提取特征,整体鲁棒性高,检测效果好.如图 1 所示,当有光照造成阴影在托盘表面时,传统方法的模板匹配无法检测托盘,而基于深度学习的 YOLOv5 网络在训练好模型后可以很好地进行托盘的检测.因此本文在选取托盘检测算法时,采取基于深度神经网络的 YOLOv5 算法.



(a) 传统图像处理方法检测托盘失败



(b) YOLOv5 成功检测托盘

图 1 托盘检测算法选取原因

本文的贡献如下:(1)针对托盘轮廓提取,本文提出了目标检测加关键点检测的方法,并设计了边缘自适应调整,得到高精度的轮廓信息.(2)针对托盘定位任务,本文根据轮廓信息和几何关系提出托盘的距离和倾角计算方法,并采用 RANSAC 算法^[15]提高计算精度和稳定性.(3)使用 RGB 摄像头拍摄并制作了托盘数据集.

输出结构来适应新的对象类别, 迁移能力较强, 所以本文选用 KPDA-Net 作为关键点检测网络. 该网络模型主要有以下 3 个部分构成: Stage1, Stage2 和 Back Propagation. Stage1 主要是提取输入图像的数据特征. 模型从经典的 ResNet^[19] 中借鉴了 short-cut connection 结构, 解决了训练时反向传播过程中出现梯度退化的问题, 保证了参数之间的区分度. Stage2 阶段进行特征图的数值坐标回归. 主要包括计算输出特征图 $S_j^l(p)$ 与热图标签的损失和特征图回归对象的关键点与对象的对应关系. Back propagation 进行关键点坐标预测值和真实值之间的损失计算. 将关键点预测坐标与真实坐标之间的 L2 范数定义为 f_S^l , 关系定义为 f_L^l , 损失函数的计算方式如下:

$$f_S^l = \sum_{j=1}^J \sum_p W(p) \cdot \|S_j^l(p) S_j^*(p)\|_2^2 \quad (3)$$

$$f_L^l = \sum_{c=1}^C \sum_p W(p) \cdot \|L_c^l(p) L_c^*(p)\|_2^2 \quad (4)$$

$$Total_loss = average_loss(f_S^l + f_L^l) \quad (5)$$

其中, $S_j^*(p)$ 是实际关键点的坐标位置, $L_c^*(p)$ 是实际标签中坐标中心位置生成的矢量图, $W(p)$ 是二值化掩码矩阵, 当 p 位置标签缺失时为 1, 否则为 0.

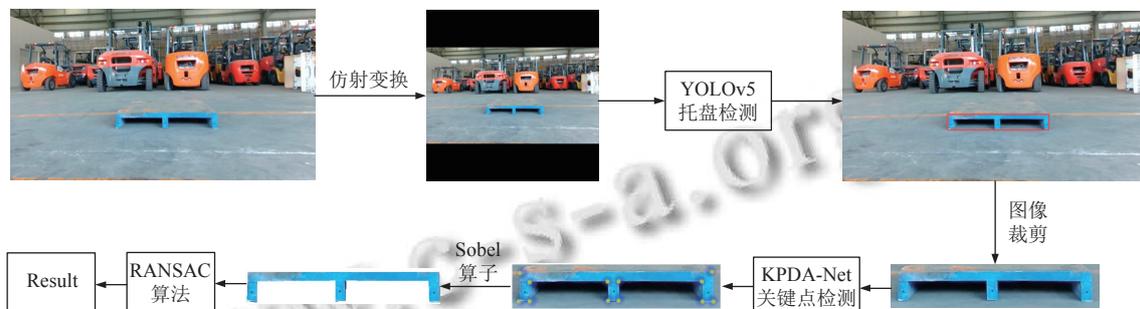


图3 算法流程图

3.2 K-means++预设锚框的托盘检测

在托盘检测的问题上, 本文使用了 YOLOv5 作为目标检测网络. YOLOv5 是有锚框 (anchor box) 的目标检测模型, 合理的锚框设置能够训练出效果更好的检测器. YOLOv5 中的预设锚框是在 COCO 数据集上使用 K-means 聚类算法计算得到, 而托盘检测作为特定的任务, 预设锚框在大多数情况下并不适用. 所以本文使用 K-means++ 算法对于托盘数据集中的标签进行聚类, 得到更适用于托盘检测任务的锚框预设值进行训

3 算法

3.1 算法框架

单目视觉的托盘定位算法可分为 3 部分: 托盘检测, 关键点检测与轮廓提取, 托盘距离与倾角计算. 设 YOLOv5 的模型为 YOLOv5det, 检测类别为托盘 (pallet), 返回信息为托盘的在图像中的位置信息 (x, y, w, h , 分别表示目标框的中心点和宽高); KPDA-Net 模型为 KPDA, 输入图像 img , 返回关键点 (keypoints) 的坐标 (coords), 具体算法流程如算法 1 所示, 整体的算法框架如图 3 所示.

算法 1. 托盘定位算法

输入: 摄像头采集的图片

输出: 托盘在图像中的位置 (x, y, w, h) 与托盘的距离 D 与倾角 θ

1. 使用仿射变换将图像转换成 $608 \times 608 \times 3$
2. **if** YOLOv5det['pallet'] != None **then**
3. $x, y, w, h = \text{YOLOv5}[\text{'pallet'}]$
4. 根据 x, y, w, h 对图像进行裁剪
5. keypoints = KPDA(img)
6. 基于几何关键初步得到托盘轮廓信息
7. 使用 Sobel 算子自适应调整得到轮廓
8. 根据式 (14) 和式 (15) 使用 RANSAC 算法计算距离 D 倾角 θ
9. **end if**
10. **return** x, y, h, w, D, θ

练. K-means++ 算法修改了 K-means 算法中初始化 k 个聚类中心的选取方法, 在 K-means++ 算法中, 距离已经选取聚类中心更远的点被选取为新的聚类中心的概率更大, 这让中心点在随机选取的过程中更趋于全局最优解而不是局部最优解. 通过 K-means++ 算法计算网络初始化的锚框大小, 使其与托盘的真实标记框大小更符合.

对托盘进行目标检测后, 按照检测结果, 对检测框四周各拓宽 $0.25h$ 像素值对图像进行剪裁 (h 表示目标

框在 y 方向的宽度), 输入关键点检测网络中进行关键点检测. 这样既可以保证剪裁过程中保留托盘的关键点以及轮廓信息, 同时又去除大量无关背景, 减少关键点检测网络的训练, 提升关键点的检测速度和准确率.

3.3 关键点检测与轮廓提取

KPDA-Net 进行关键点检测由两部分组成, 首先使用特征提取模块 KeyPointsModel 对输入图像 (img) 进行特征提取, 得到热图 (heatmap); 再使用数值坐标回归模块 dsntnn 对热图中的数组坐标 (coords) 进行回归得到关键点的坐标. 由于托盘具有固定的几何外形, 通过热力图回归得到的关键点按照一定顺序相连将直接获得托盘初步的轮廓信息. 但较差的关键点结果会影响托盘的轮廓信息进而影响最终的角度和距离测算. 针对此问题, 本文对边缘点设计了自适应调整的方法. 因为轮廓边缘是梯度变化最大的点, 所以本文采用 10×10 大小的采样框在初步轮廓连线上移动, 将采样框中图像转化为灰度图后对托盘的上下边缘计算 Sobel 算子 Y 方向梯度, 左右边缘计算 Sobel 算子 X 方向梯度, 按照梯度最大值自适应调整边缘, 如图 4 所示.

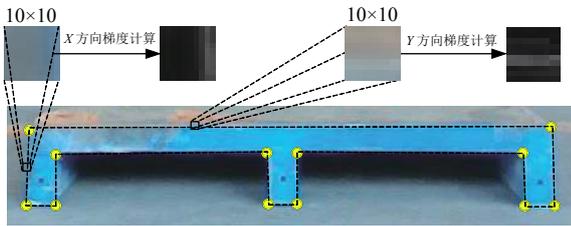


图4 Sobel 算子计算示意图

Sobel 算子是在边缘检测中常用的一种离散微分算子, 由两个 3×3 的卷积核构成, 分别是水平方向内核 G_x 和垂直方向内核 G_y .

$$G_x = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} \quad (6)$$

$$G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{pmatrix} \quad (7)$$

具体算法流程如算法 2 所示.

算法 2. 关键点检测与轮廓提取的算法

输入: 根据 YOLOv5 返回结果剪裁后的图片
输出: 托盘的轮廓信息

1. heatmap = KeyPointsModel(img)
2. coords = dsntnn(heatmap)

3. 将关键点按照几何位置相连得到初步轮廓边缘
4. for point in keypoints connection do
5. point = sobel(point range)
6. end for
7. return 托盘轮廓信息

3.4 基于几何约束的托盘倾角和距离计算方法

无人叉车在叉取托盘的过程中, 相机会安装在叉车上的固定位置, 托盘会摆放在地面或者货架上, 与地面平行. 根据以上条件, 将像素坐标系与叉车相机坐标系进行映射, 就可以对单目相机采集的托盘图像进行位姿估计. 像素坐标系与叉车相机坐标系的关系如图 5 所示. 根据针孔相机模型, 在叉车相机坐标系中, xz 平面平行于地面, z 轴方向为相机正前方, y 轴垂直于地面朝下. 归一化平面是相机前方 $z = 1$ 位置的平面. P 是托盘上的任意一点, 在叉车相机坐标系中的坐标为 (X, Y, Z) . p 是 P 在归一化平面中所对应的点, 像素坐标为 (x, y) . P 与 p 有如下映射关系:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \triangleq \frac{1}{Z} KP \quad (8)$$

其中, f_x, f_y 为归一化焦距, c_x, c_y 为像主点的坐标, K 为相机的内参矩阵. 托盘在空间中的位姿如图 5 所示. H 为托盘的物理高度, h 为叉孔上边缘的厚度, L 为托盘的长度, 定义角度 θ 为托盘正面与坐标系 x 轴到地面投影所成夹角, 定义距离 D 为托盘正面中心位置

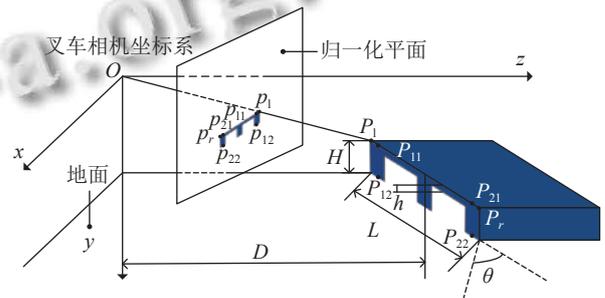


图5 针孔相机模型下托盘映射关系

到相机的水平距离, P_l, P_r 是托盘上边缘的左右端点, 坐标为 $(X_l, Y_l, Z_l), (X_r, Y_r, Z_r)$, p_l, p_r 为 P_l, P_r 在归一化平面内的映射点, 坐标为 $(x_l, y_l), (x_r, y_r)$, P_{i1}, P_{i2} 是托盘上下边缘 z 值相等的一对点, 坐标为 $(X_{i1}, Y_{i1}, Z_{i1}), (X_{i2}, Y_{i2}, Z_{i2})$, p_{i1}, p_{i2} 为 P_{i1}, P_{i2} 在归一化平面内的映射点, 坐标为 $(x_{i1}, y_{i1}), (x_{i2}, y_{i2})$. 当点对 P_{i1}, P_{i2} 处在非插孔所在边缘时, 有如下空间关系:

$$\begin{pmatrix} X_{i2} \\ Y_{i2} \\ Z_{i2} \end{pmatrix} = \begin{pmatrix} X_{i1} \\ Y_{i1} \\ Z_{i1} \end{pmatrix} + \begin{pmatrix} 0 \\ H \\ 0 \end{pmatrix} \quad (9)$$

在托盘轮廓上任取两对点 P_{11}, P_{12} 和 P_{21}, P_{22} , 对点 P_{11} 和 P_{21} , 有如下空间关系:

$$\begin{pmatrix} X_{21} \\ Y_{21} \\ Z_{21} \end{pmatrix} = \begin{pmatrix} X_{11} \\ Y_{11} \\ Z_{11} \end{pmatrix} + \begin{pmatrix} W \cos \theta \\ 0 \\ W \sin \theta \end{pmatrix} \quad (10)$$

$$W = L \frac{x_{11} - x_{21}}{x_1 - x_2} \quad (11)$$

其中, W 为点 P_{11} 和 P_{21} 的空间距离. 根据式(8)反向求解 Y 可得:

$$Y = \frac{Z}{f_y} (y - c_y) \quad (12)$$

将式(12)带入式(9)求解 Z 可得:

$$Z_{i1} = Z_{i2} = \frac{f_y H}{y_{i2} - y_{i1}} \quad (13)$$

将式(13)带入式(10)可得:

$$\theta = \arcsin \left(\frac{f_y H (x_1 - x_2)}{L (x_{11} - x_{21})} \left(\frac{1}{y_{22} - y_{21}} - \frac{1}{y_{12} - y_{11}} \right) \right) \quad (14)$$

将式(13)和式(14)带入式(10)可得:

$$D = \frac{f_y H}{\Delta y_i} + \frac{f_y H (x_{i1} - 0.5 \Delta x)}{L (x_{11} - x_{21})} \left(\frac{1}{\Delta y_2} - \frac{1}{\Delta y_1} \right) \quad (15)$$

$$\Delta y_i = y_{i2} - y_{i1} \quad (16)$$

$$\Delta x = x_1 - x_2 \quad (17)$$

由式(14)和式(15)即可算出托盘当前与摄像头之间的距离和倾角.

3.5 基于 RANSAC 算法的托盘倾角和距离计算

如果只取检测的关键点用于计算, 样本数据量小, 微小的误差就可能产生不准确的计算结果, 针对此问题, 本文根据轮廓信息, 采用 RANSAC 算法提高计算精度, 算法流程如图6所示.

通过选取多组点对进行计算增加样本的数据量, 同时使用 RANSAC 算法剔除错误的点对. 由于通过对点对计算得到的倾角和距离是估计值而非固定值, 所以就必须要考虑存在误差. 设定算法阈值为:

$$|d\theta| \leq 3^\circ \quad (18)$$

$$|dD| \leq 0.2 \text{ m} \quad (19)$$

其中, $d\theta$ 为最优角度与当前计算角度的差, dD 为最优距离与当前计算距离的差.

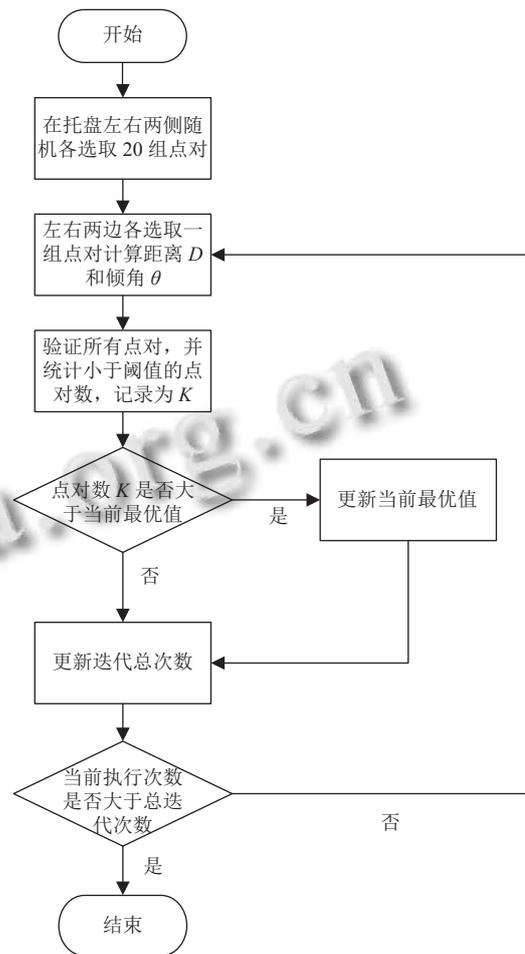


图6 RANSAC 算法流程图

4 实验分析

4.1 数据集

本文采用的数据集是笔者自己采集并建立的数据集. 考虑到目前没有公开托盘检测数据集, 笔者自己使用 1980×1080 分辨率的 RGB 摄像头拍摄了一个托盘数据集. 整个数据集分别在室内和室外多个场景下拍摄放置于地面和货架上的托盘, 采集总时长为 70 min 的视频. 剔除质量不高和变化不大的图片, 选取 1 680 帧图像作为实验数据集. 其中训练集占比 80%, 验证集占比 10%, 测试集占比 10%.

4.2 实验过程

本实验所用机器配置如表1所示. 为了提高目标检测的鲁棒性, 本文训练时对输入图像进行随机反转, 并加入一个服从正太分布的随机噪声, 将数据增强后的 2 688 张图像输入网络进行训练.

实验有以下评价指标: 对于托盘的目标检测, 使用

召回率 Recall 和平均 AP 值 mAP. 召回率越高, 表示托盘被正确检测的概率越高; mAP 越高, 表示托盘检测精度越高. 对于托盘倾角和距离计算, 使用平均角度误差 $\theta_{average}$ 和最大角度误差 θ_{max} 作为角度计算的评价指标; 使用平均距离误差 $D_{average}$ 和最大距离误差 D_{max} 作为距离计算的评价指标.

$$\theta_{average} = \frac{\sum_{j=1}^J |\theta_j - \theta|}{J} \quad (20)$$

$$\theta_{max} = MAX(|\theta_j - \theta|) \quad (21)$$

$$D_{average} = \frac{\sum_{j=1}^J |D_j - D|}{J} \quad (22)$$

$$D_{max} = MAX(|D_j - D|) \quad (23)$$

其中, θ_j 为第 j 帧图像中托盘角度计算值, θ 为托盘实际角度, D_j 为第 j 帧图像中托盘距离计算值, D 为托盘实际距离.

经过 K-means++ 算法对托盘数据集进行聚类后得到的 9 个锚框的大小分别为 (62, 19), (68, 22), (76, 27),

(80, 23), (89, 34), (91, 27), (102, 30), (112, 31), (119, 35). 学习率设置为 0.003 2, 模型训练的 epoch 设置为 300, 模型每次迭代一个 epoch 都会在验证集上进行验证, 并保存效果最好的一个模型. 由于数据集数量较小, 为了防止出现过拟合的情况, 当训练过程中 Loss 不再下降时笔者会停止训练. 利用 YOLOv5 的预训练权重作为初始训练权重, 使用自己的数据集进行训练. 训练过程中 Loss, Recall 和 mAP 的变化曲线如图 7 所示, 其中, 深色曲线为当前坐标轴的曲线变化, 浅色曲线为放大坐标轴的曲线变化.

表 1 训练所用机器配置表

类型	型号	参数
系统	Ubuntu	16.04.2 LTS
CPU	Intel(R) Core(TM) i9-9900	8核
GPU	Nvidia GeForce GTX 1080Ti	11 GB
内存	DDR4	32 GB

由图 7 可知, 随着训练的进行, Loss 值逐渐下降, Recall 和 mAP 值逐渐上升. 当训练到一定的程度时, 模型收敛, 此时 Loss, Recall 和 mAP 的值在一个稳定的区间, 停止训练, 得到训练好的模型.

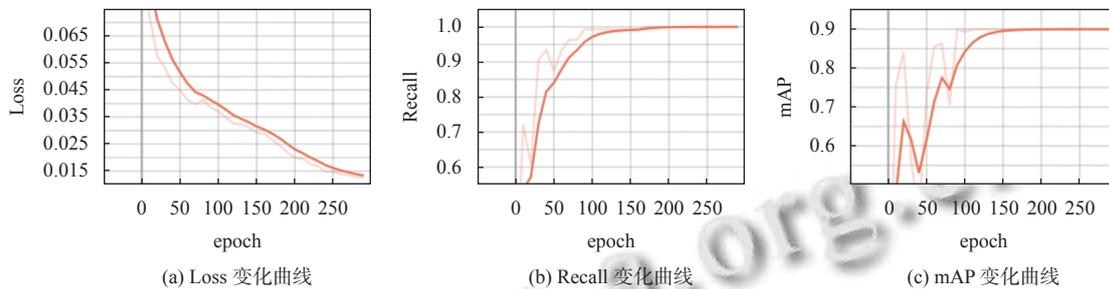


图 7 训练过程曲线变化图

4.3 实验结果及分析

将设定锚框后的网络与经典的目标检测网络 Faster R-CNN^[20], YOLOv5s^[16] 以及文献 [10] 和文献 [13] 中的方法在测试集上进行实验结果比对, 如表 2 所示. 表中对比不同网络的召回率和平均精度, 从表中可以看出, 本文方法对比其他目标检测网络效果较好, 更利于后续对于托盘的位姿计算.

为了验证本文对于托盘倾角和距离计算方法及其改进的有效性, 选取室内放置于地面场景下的托盘, 角度为 $-60^{\circ} \sim 60^{\circ}$, 每间隔 10° 为一组图片; 距离为 1 600–3 200 mm, 每间隔 100 mm 一组图片进行实验验证. 笔者使用 $\theta_{average}$, θ_{max} , $D_{average}$ 和 D_{max} 作为评价指标, 对算

法自身进行了纵向对比实验. 将关键点检测后直接进行倾角和距离计算设为原算法, 自适应边缘调整设为改进 1, 使用 RANSAC 算法设为改进 2. 实验所得角度与真实角度的误差如表 3 所示, 表中所有数据单位为 $^{\circ}$; 所得距离与真实距离的误差如表 4 所示, 表中所有数据单位为 mm.

表 2 不同网络结果对比实验表 (%)

模型	Recall	mAP
FasterRCNN	98.1	78.3
YOLOv5s	98.8	89.8
叶伟 ^[10]	98.3	87.4
朱丹平等人 ^[13]	94.1	88.2
Ours	99.0	90.1

表3 托盘计算角度误差实验表(°)

角度	原算法		改进1		改进2	
	θ_{average}	θ_{max}	θ_{average}	θ_{max}	θ_{average}	θ_{max}
-60	7.4	13.1	7.1	12.8	4.1	8.1
-50	5.6	10.7	5.4	10.6	3.7	7.7
-40	5.4	9.3	5.1	8.9	3.0	6.8
-30	4.9	9.5	4.6	9.2	2.4	5.9
-20	5.3	8.6	5.2	8.6	2.8	5.6
-10	4.8	9.0	4.6	8.9	2.6	5.4
0	5.2	7.8	5.0	7.3	2.9	4.8
10	4.8	9.4	4.5	9.1	2.6	6.3
20	4.5	9.3	4.3	9.1	2.2	5.9
30	4.7	8.3	4.6	8.2	2.0	6.4
40	5.2	10.7	5.0	10.5	3.0	8.9
50	5.9	10.4	5.6	10.3	3.4	5.6
60	6.5	11.4	6.3	11.2	3.8	7.5

表4 托盘计算距离误差实验表(mm)

距离	原算法		改进1		改进2	
	D_{average}	D_{max}	D_{average}	D_{max}	D_{average}	D_{max}
1600	81	136	77	131	47	65
1700	84	131	79	127	52	68
1800	86	126	83	119	55	66
1900	87	146	84	143	58	79
2000	96	134	92	128	65	81
2100	95	145	92	140	65	82
2200	104	143	101	138	77	94
2300	110	152	106	152	80	100
2400	121	163	119	161	90	110
2500	114	150	112	148	86	100
2600	112	163	108	162	82	110
2700	127	161	123	159	96	113
2800	120	166	117	163	91	112
2900	125	177	121	171	95	116
3000	127	187	123	182	98	125
3100	133	176	127	168	101	117
3200	138	201	133	192	108	123

根据表3和表4的实验结果可知,改进后的算法精度和稳定性显著提升,对于托盘的角度计算误差均值小于5°,最大值不超过10°;距离计算误差均值小于110 mm,最大值不超过130 mm,不足真实距离的5%。实验结果,本文方法的托盘距离和倾角计算对于单目相机而言具有较高的精度,能够提供有效的托盘位姿信息。

5 结论与展望

本文提出了一种基于目标和关键点检测的托盘定位方法。首先通过目标检测网络YOLOv5对于托盘进行目标检测,将得到的结果剪裁后输入KPDA-Net中

进行关键点检测。然后根据托盘的几何外形,设计了自适应边缘调整提取轮廓。在得到托盘轮廓的基础上,提出了托盘倾角与距离的计算方法并进行了实验验证。与其他托盘定位的方法对比,本文解决了目前托盘定位中目标检测算法无法得到托盘空间信息的问题,算法表现优异,下一步计划将本文方法部署在无人叉车系统上,更好更精确的实现托盘的叉卸。

参考文献

- 1 中共中央国务院关于加快建设全国统一大市场的意见. 人民日报, 2022-04-11(001). [doi: 10.28655/n.cnki.nrmrb.2022.004032]
- 2 赵俊宏, 吕恩利, 陆华忠, 等. 基于激光雷达的托盘位姿识别算法及验证. 仪器仪表学报, 2017, 38(10): 2469–2476. [doi: 10.3969/j.issn.0254-3087.2017.10.015]
- 3 Mohamed IS, Capitanelli A, Mastrogianni F, *et al.* Detection, localisation and tracking of pallets using machine learning techniques and 2D range data. *Neural Computing and Applications*, 2020, 32(13): 8811–8828. [doi: 10.1007/s00521-019-04352-0]
- 4 Bellomo N, Marcuzzi E, Baglivo L, *et al.* Pallet pose estimation with LIDAR and vision for autonomous forklifts. *IFAC Proceedings Volumes*, 2009, 42(4): 612–617. [doi: 10.3182/20090603-3-RU-2001.0540]
- 5 Chen G, Peng R, Wang ZC, *et al.* Pallet recognition and localization method for vision guided forklift. *Proceedings of the 8th International Conference on Wireless Communications, Networking and Mobile Computing*. Shanghai: IEEE, 2012. 1–4.
- 6 Cui GZ, Lu LS, He ZD, *et al.* A robust autonomous mobile forklift pallet recognition. *Proceedings of the 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*. Wuhan: IEEE, 2010. 286–290.
- 7 武文汉. 基于视觉的仓储托盘检测方法研究 [硕士学位论文]. 上海: 上海交通大学, 2018. [doi: 10.27307/d.cnki.gsjtu.2018.002684]
- 8 Xiao JH, Lu HM, Zhang LL, *et al.* Pallet recognition and localization using an RGB-D camera. *International Journal of Advanced Robotic Systems*, 2017, 14(6): 1729881417737799.
- 9 Mok C, Baek I, Cho YS, *et al.* Pallet recognition with multi-task learning for automated guided vehicles. *Applied Sciences*, 2021, 11(24): 11808. [doi: 10.3390/app112411808]
- 10 叶伟. 基于视觉的无人叉车导航和托盘定位方法研究与实现 [硕士学位论文]. 合肥: 中国科学技术大学, 2021. [doi:

- [10.27517/d.cnki.gzkju.2021.001422](https://doi.org/10.27517/d.cnki.gzkju.2021.001422)]
- 11 Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934, 2020.
 - 12 Rother C, Kolmogorov V, Blake A. “GrabCut”: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics, 2004, 23(3): 309–314. [doi: [10.1145/1015706.1015720](https://doi.org/10.1145/1015706.1015720)]
 - 13 朱丹平, 朱明, 周恒森. 基于改进 CenterNet 的托盘定位方法. 计算机系统应用, 2022, 31(10): 303–309. [doi: [10.15888/j.cnki.csa.008728](https://doi.org/10.15888/j.cnki.csa.008728)]
 - 14 Zhou XY, Wang DQ, Krähenbühl P. Objects as points. arXiv:1904.07850, 2019.
 - 15 Fischler MA, Bolles RC. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 1981, 24(6): 381–395. [doi: [10.1145/358669.358692](https://doi.org/10.1145/358669.358692)]
 - 16 Jocher G, Stoken A, Borovec J, *et al.* Ultralytics/YOLOv5: V3.0. <https://zenodo.org/record/3983579#.ZAG7Sh9By61>. (2020-08-13).
 - 17 Wu J, He ZS, Yan K, *et al.* Key points detection algorithm of object based on full convolution network. Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE). Beijing: IEEE, 2020. 158–162. [doi: [10.1109/ICAICE51518.2020.00037](https://doi.org/10.1109/ICAICE51518.2020.00037)]
 - 18 Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE, 2015. 3431–3440. [doi: [10.1109/CVPR.2015.7298965](https://doi.org/10.1109/CVPR.2015.7298965)]
 - 19 He KM, Zhang XY, Ren SQ, *et al.* Deep residual learning for image recognition. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778. [doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90)]
 - 20 Ren SQ, He KM, Girshick R, *et al.* Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149. [doi: [10.1109/TPAMI.2016.2577031](https://doi.org/10.1109/TPAMI.2016.2577031)]

(校对责编: 牛欣悦)