

文章编号: 2095-2163(2020)11-0182-05

中图分类号: TP181

文献标志码: A

# 基于YOLOV4的智能垃圾分类回收机器人

王伟杰, 姚建涛, 张敏燕, 王敏

(四川工商学院 计算机学院, 成都 611745)

**摘要:** 本研究基于Yolov4目标检测算法,将垃圾分类机器人和人工智能相结合,赋予机器人视觉系统,让其能够分辨垃圾的种类,并将其放入自带的相应垃圾桶中。传统扫地机器人的自动充电系统是基于红外线传感器,本研究采用计算机视觉系统代替红外线传感器,机器人能够主动寻找充电插座,使其更加智能化。

**关键词:** Yolov4; 机器人; 人工智能; 计算机视觉; 垃圾分类

## Intelligent garbage sorting and recycling robot based on YOLOV4

WANG Weijie, YAO Jiantao, ZHANG Minyan, WANG Min

(School of Computer Science, Sichuan Technology and Business University, Chengdu 611745, China)

**[Abstract]** Based on the Yolov4 target detection algorithm, this study combines garbage classification robot and artificial intelligence to give the robot a vision system that enables it to distinguish the types of garbage and put them into the corresponding garbage can. We have also improved its traditional charging system. The automatic charging system of the traditional sweeping robot is based on the infrared sensor, and we have replaced the infrared sensor with the computer vision system, so that our robot can actively find the charging socket, and at the same time make our robot more intelligent.

**[Key words]** Yolov4; Robotics; Artificial intelligence; Computer vision; Garbage classification

### 0 引言

随着国家对环境问题的重视,以及环保意识逐渐深入人心,垃圾分类的理念也逐渐步入人们的视野。然而对于垃圾分类存在的问题,也急需得到解决。目前市面上清理垃圾的机器人主要分为两类:一是家用的扫地机器人,二是大型的垃圾分拣机器人。前者依靠红外线或者超声波进行避障和充电,只能清扫垃圾,没有分类功能;后者成本太高,只适用于大型的垃圾回收站。基于这些问题,本文设计完成了一款基于图像识别的垃圾回收和计算机视觉智能充电机器人,既能识别和分类回收垃圾,智能避障,也能实现自动充电。

### 1 现状

由于人们对环境问题越来越重视,尽快解决垃圾分类的话题已被提到议事日程。国家为解决环境问题颁布了一些法律支持,上海市已于2019年1月31日率先通过了《上海市生活垃圾管理条例》,垃圾分类第一次被写入法律。目前中国的垃圾分类回收处理主要面临如下问题:前期垃圾分类阻力大、中期垃圾分类处置不合理、后期生活垃圾处理不够科

学<sup>[1]</sup>。中国的全民垃圾分类目前还处于推广阶段,居民垃圾分类意识不强,很多城市依然采用混合收集方式,这给垃圾处理的中期和后期带来不便。垃圾处理的中期是垃圾中转站对垃圾进行进一步分类,大部分地区没有出台强制性的垃圾分类政策,一部分地区的垃圾中转站并没有对回收的垃圾进行分类和再利用。国内市场上还没有一款针对前期垃圾分类的智能产品,垃圾分类依然需要居民手动分类。

人们生活水平的提高,生活垃圾种类也在不断增加,但市场上还没有适用于家庭和公共场所的人工智能垃圾回收机器人。目前国内市场上垃圾清理机器人技术主要有:激光雷达的传输和感应技术,通过光的折射或者反射对不同的垃圾进行数据的返回,根据射程的距离来实现垃圾分类。而人工智能的出现在增加传统扫地机器人功能的同时,也打破了机器人依靠多个传感器才能运行的传统模式。

### 2 系统设计

#### 2.1 功能设计

经调研分析,智能垃圾分类机器人的主要功能包括以下几部分:

**作者简介:** 王伟杰(1999-),男,本科生,主要研究方向:人工智能、云计算;姚建涛(1997-),男,本科生,主要研究方向:人工智能、云计算;张敏燕(1999-),女,本科生,主要研究方向:计算机科学与技术;王敏(1984-),男,硕士,副教授,主要研究方向:人工智能、数据挖掘。

**通讯作者:** 王敏 Email: 172128788@qq.com

收稿日期: 2020-09-03

(1) 垃圾识别。机器人会将视觉信息通过树莓派发送给服务器, GPU 服务器会将处理后的目标信息(包括垃圾的类型和位置)返回给机器人。

(2) 拾取垃圾并分类。树莓派会根据服务器返回的目标信息执行相应的命令控制机械臂夹取垃圾, 并将其放入到相应的垃圾桶中。

(3) 智能充电。当机器人的电量低于一定的值

时, 机器人将会通过摄像头自动检测周围是否有插座, 当识别到插座时, 机器人会通过机械臂将充电插头插入到插座中。

## 2.2 硬件设计

系统用到的硬件设备主要包含: 摄像头、机械臂(含机械臂驱动模块)、电机(含电机驱动模块)、树莓派、电池、底座(含履带)等。硬件结构如图 1 所示。

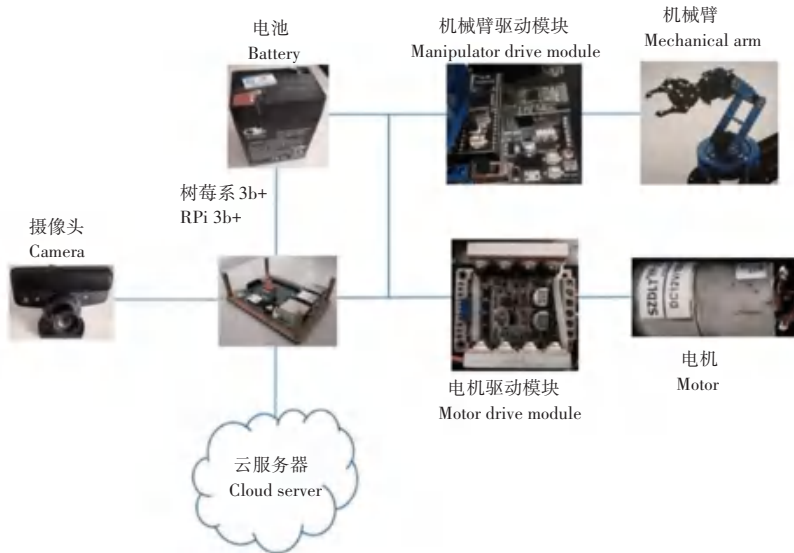


图 1 硬件结构图

Fig. 1 Hardware structure diagram

其中: 摄像头采用的是 1080p 高清 usb 工业摄像头; 机械臂驱动模块采用插卡式设计, 系统采用 STM32 控制板, 支持 PC 图形化、代码编程以及脱机运行, 提供正运动学分析和逆运动学分析; 电机驱动模块开关频率高, 可有效避免调试电机频率低带来的问题; 树莓派使用的是 Raspberry pi 3 Model B+, 板卡配备 64 位的 1.4GHz 四核 ARM Cortex-A53 处理器, 该设备的双频无线局域网具有模块化的兼容性; 系统使用铝电池, 主要为机械臂舵机、小车电机、以及树莓派供电; 电机型号为 985 低速版, 额定转速为 12 V 3 000 r/min; 机械臂由合金机械爪、6 个数字舵机(包含 3 个防烧防堵转舵机)、全金属旋转底盘以及若干铝合金支架构成。

## 2.3 软件架构

软件架构如图 3 所示。它主要有二种模式, 即垃圾识别分类模式和智能充电模式, 默认模式为垃圾识别分类模式。通过将摄像头获取的视觉信息传输给树莓派, 再由树莓派传输给系统后端, 后端 GPU 服务器将会对该视觉信息进行处理, 并将目标的位置信息和类别信息传输给树莓派。树莓派执行

相应指令, 调整摄像头的位置(摄像头在机械臂上), 同时驱动电机靠近目标, 直到调整到合适的位置, 树莓派才会给机械臂驱动模块发送夹取的命令, 机械臂根据对应目标类别信息放入到自带的对应垃圾桶中。并且会再次判断目标是否夹取成功, 若失败将再次调整机器人与目标之间的位置。如果多次夹取失败, 树莓派会将其相关信息发送给服务器, 研究人员会对其进行分析 and 做相关处理。

智能充电的原理与垃圾识别分类的原理相似。当电池电量下降到相应值时(第一阈值), 树莓派自动将模式切换为智能充电模式。该模式下机器人会自动寻找插座, 当机器人识别到插座时, 树莓派会执行相应命令调整摄像头, 并驱动电机靠近插座, 机械臂再将机器人充电器夹起, 并调整机械臂位置, 直到充电器插入插座。树莓派会判断该插座是否有电, 如无电, 则取出充电器寻找其他的插座。若电量低于第二阈值时, 此时机器人的电量已经不够其完成自动充电, 机器人会停留在原地给后端发送位置坐标, 以便相关人员能够找到。软件逻辑关系如图 2 所示。

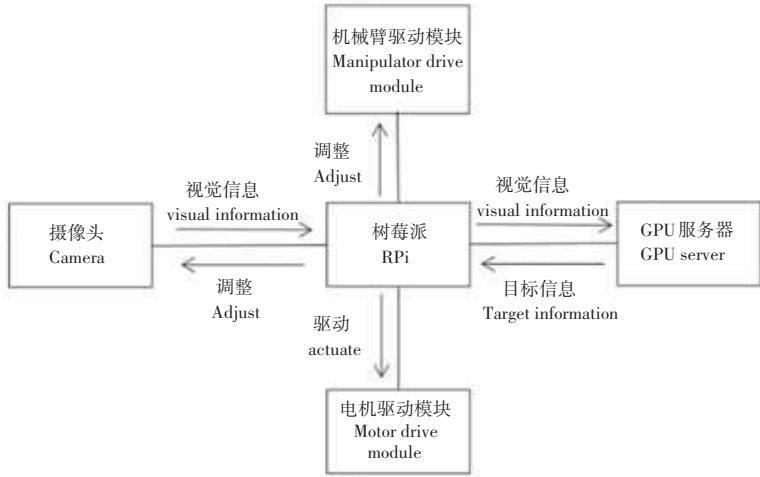


图 2 软件逻辑图

Fig. 2 Software logic diagram

2.4 算法设计

2.4.1 Yolov4 目标检测算法概述

Yolov4<sup>[2]</sup>是 Yolov3<sup>[3]</sup>的改进版,无论在精度还是速度上 Yolov4 都碾压了 Yolov3。YoLov4 原文作者在论文中提到,在进行 bounding box regression 时,传统的目标检测算法(如 Yolov3)都是根据预测框和真实框的中心点坐标,以及宽高信息设定 MSE(均方误差)损失函数, Yolov3 的总损失函数公式如式(1):

$$\begin{aligned}
 Loss = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} [(x_i - \hat{x}_i^j)^2 + (y_i - \hat{y}_i^j)^2] + \\
 & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} [(\sqrt{\omega_i^j} - \sqrt{\hat{\omega}_i^j})^2 + (\sqrt{h_i^j} - \sqrt{\hat{h}_i^j})^2] - \\
 & \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)] - \\
 & \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{noobj} [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)] - \\
 & \sum_{i=0}^{S^2} \sum_{c \in classes} I_{ij}^{obj} \sum_c ([\hat{P}_i^j \log(P_i^j) + (1 - \hat{P}_i^j) \log(1 - P_i^j)]).
 \end{aligned}
 \tag{1}$$

MSE 损失函数包括 3 个部分:即预测框的位置损失、预测框预测的类别损失和预测框置信度损失。MSE 损失函数将检测框中心点坐标和宽高等信息作为独立的变量对待,但实际上这些信息之间是有联系的。因此, Yolov4 的作者将 MSE 替换为 CIOU(Complete-IoU)。CIOU 考虑到 3 个几何因素,即重叠面积、中心点距离和长宽比。CIOU 的惩罚项是在 DIoU 的惩罚项基础上加了影响因子  $\alpha$  和  $v$ , 这个因子把预测框长宽比拟合目标框的长宽比考虑进去。其中  $\alpha$  是用做 trade-off 的参数,  $v$  是用来衡量长宽

比一致性的参数<sup>[4]</sup>,具体公式如式(2)、式(3):

$$v = \frac{4}{\pi^2} \left( \arctan \frac{\omega^{gt}}{h^{gt}} - \arctan \frac{\omega}{h} \right)^2, \tag{2}$$

$$\alpha = \frac{v}{(1 - IoU) + v}. \tag{3}$$

其中,  $\omega^{gt}$  和  $h^{gt}$  为真实框的宽和高,  $\omega$  和  $h$  为预测框的宽和高。如果真实框和预测框的宽和高一样,则  $v$  为 0, 即该惩罚项失效。其次 Yolov4 将 DarknetConv2D 的激活函数由 ReLU 修改成了 Mish, 如公式(4):

$$Mish = x \times \tanh(\ln(1 + e^x)). \tag{4}$$

相比 ReLU 激活函数, Mish 激活函数的输出更为平滑,而平滑的激活函数可以让更好的信息进入神经网络,从而使模型得到更好的泛化能力,避免过拟合。其次,是将 resblock\_body 的结构进行了修改,运用了 CSPnet 结构。运用这种结构的作用是在降低计算量的同时增加模型的精准度, CSPnet 结构如图 3 所示。

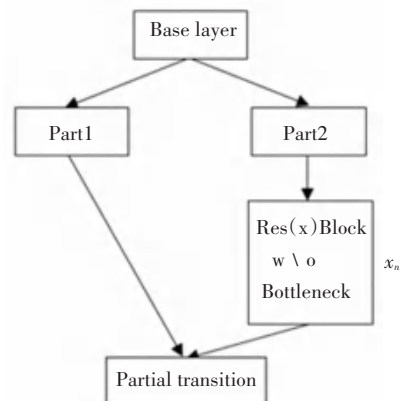


图 3 CSPnet 结构

Fig. 3 CSPnet structure

Yolov4 采用了 PANet(用于实例分割的路径聚合网络)<sup>[5]</sup>的结构,将低层与高层的特征进行融合,提高目标检测的精度,尤其是对小目标识别的精度。PANet 总体上是 Mask-Rcnn 的改进版<sup>[6]</sup>,整体思路为提高信息流在网络中的传输速度。其主要在 3 方面进行了改进:

(1) 提出了 Bottom-up Path Augmentation, 它能提高底层特征信息的利用率, 加快底层特征信息的传输效率;

(2) 提出了 Adaptive Feature Pooling, 在提高特征提取速度的同时也保证了特性信息的利用率;

(3) 提出了 Fully-connected Fusion, 提高了 mask 的生成质量。

为解决小目标检测准确度低的问题, Yolov4 作者在其算法中引入了马赛克数据增强算法。其原理是从数据集中每次读取 4 张图片, 然后对这 4 张图片进行翻转、缩放、色域变化等, 如图 4 所示。最后将其组合在一起, 如图 5 所示。



图 4 图片的翻转、缩放、色域变化  
Fig. 4 Flip, zoom, gamut change



图 5 图片的拼接结果

Fig. 5 The Mosaic result of the picture

#### 2.4.2 目标检测实现

本文基于 Yolov4 实现目标检测, Yolo Head 会根据前面提取的特征层进行预测, 不断调整先验框的大小和位置, 使其生成最后的预测框。经过主干

网络的特征提取, 可以得到 shape 为  $(N, 52, 52, 255)$ 、 $(N, 26, 26, 255)$  和  $(N, 13, 13, 255)$  的特征层, 而每个特征层上会有 3 个预设好的 anchor box, 总共会聚类 9 个 anchor box。对于输入图片为  $416 \times 416$  时, 在  $13 \times 13$  的特征图中由于其感受野最广, 适合检测大目标, 所以分配最大的 anchor box, 即  $(116 \times 90)$ 、 $(156 \times 198)$ 、 $(373 \times 326)$ 。在  $26 \times 26$  的特征图中, 它的感受野处于中等水平, 适合检测中等大小的目标, 所以分配中等大的 anchor box, 即  $(30 \times 61)$ 、 $(62 \times 45)$ 、 $(59 \times 119)$ 。在  $52 \times 52$  的特征图中, 感受野最小, 适合检测小目标, 所以分配最小的 anchor box, 即  $(10 \times 13)$ 、 $(16 \times 30)$ 、 $(33 \times 23)$ 。边框预测公式为式(5)~式(8):

$$b_x = \sigma(t_x) + c_x, \quad (5)$$

$$b_y = \sigma(t_y) + c_y, \quad (6)$$

$$b_w = p_w e^{t_w}, \quad (7)$$

$$b_h = p_h e^{t_h}. \quad (8)$$

通过上面的公式可以得到边界框相对于特征映射的位置和大小, 但网络实际学习的目标是  $t_x$ 、 $t_y$ 、 $t_w$ 、 $t_h$  这 4 个参数。由于 Yolov4 的输出是一个卷积特征图, 其中包含了特征图深度的边界框属性, 而边界框的属性又由彼此堆叠的单元格测出, 这种格式对输出值的处理很不方便。

其中  $t_x$  和  $t_y$  是预测的坐标偏移值,  $t_w$  和  $t_h$  是尺度缩放, 根据前面的公式可以求出真实的预测框坐标。求  $t_x$ 、 $t_y$ 、 $t_w$ 、 $t_h$  的公式如式(9)~式(12):

$$t_x = (G_x - P_x) / P_w, \quad (9)$$

$$t_y = (G_y - P_y) / P_h, \quad (10)$$

$$t_w = \log(G_w / P_w), \quad (11)$$

$$t_h = \log(G_h / P_h). \quad (12)$$

每张特征图上有 3 个先验框, 具体哪个先验框预测目标, 需要在训练中确定, 即由 IOU 最大的先验框来预测, 而其余的两个先验框不会和 ground truth 匹配。最后根据得到的  $t_x$ 、 $t_y$ 、 $t_w$ 、 $t_h$  来对先验框进行微调, 使其与 ground truth 重合, 得到最后的预测框。

### 3 实验结果与分析

#### 3.1 数据集制作

首先收集图片, 使用 Python 程序将所有图片重命名, 使用 Labelimg 给每个图片的 object 打上标签, 生成的目标信息保存于对应的“.XML”文件里。本研究采用的是 VOC 格式的数据集, 训练前将“.jpg”文件放在 VOC2007 文件夹中的 JPEGImages 目录, 标签文件放在 VOC2007 中的 Annotation 目录中。在 VOC2007 目录中运行 test.py, 会生成 4 个.txt 文



件,即 train.txt、val.txt、test.txt 和 trainval.txt。修改 voc\_label.py 中的相关参数(sets 和 classess)的值,然后运行该代码,会生成 3 个文件 2007\_train.txt、2007\_val.txt、2007\_test.txt,并在 VOC2007 中生成一个 labels 文件夹。

### 3.2 训练神经网络

上述生成的 labels 文件夹中 txt 文件,是训练真正用到的数据集,其中存放每张图片的绝对路径和真实框的位置。训练的命令也比较简单:./darknet detector train cfg/obj.data cfg/yolo-obj.cfg yolov4.conv.137。当神经网络的 loss 值几乎不变时,则可停止训练。

### 3.3 结果分析

实验使用 2 500 张图片来测试模型的精度,具体数据见表 1。

表 1 模型测试精度表  
Tab. 1 Model test accuracy table

名称	类型	数量	精度
塑料瓶	可回收垃圾	500	85.00%
食物罐头	可回收垃圾		
牛奶盒	可回收垃圾		
卫生纸	干垃圾	500	83.50%
一次性筷子	干垃圾		
杯子	干垃圾		
橙皮	湿垃圾	500	86.40%
香蕉皮	湿垃圾		
西瓜皮	湿垃圾		
废电池	有害垃圾	500	86.20%
医用棉签	有害垃圾		
口服液瓶	有害垃圾		
插座		500	89.20%

(上接第 181 页)

### 2.2 结果分析

在起始点与目标点的距离为 5 000m~5 500 m,本文方法与原有的二种方法飞行耗时实验对比数据见表 1。

表 1 飞行耗时实验对比数据

Tab. 1 Flight time-consuming experiment comparison data

起始点与目标点的距离/m	飞行耗时/min		
	本文方法	传统方法 1	传统方法 2
5 000	20.31	26.32	27.32
5 100	22.36	27.25	28.35
5 200	23.21	28.36	29.33
5 300	24.15	28.58	29.96
5 400	25.01	29.32	30.27
5 500	25.36	29.63	30.88

从表中可以看出,模型对插座的识别精度最高,对干垃圾的识别精度最低。可能是选取干垃圾的训练图片特征不太明显或图片量太少而导致模型对该类别的收敛不及其它类别。因此,可以寻找特征更为明显的干垃圾图片,加入到训练集中,这样其识别精度应该会有所提高。另外,本模型的平均精度为 86.06%,能识别的垃圾种类很少,离理想的模型还是有一定的距离,需扩充训练集图片的数量和种类,优化识别算法,使模型更加完善。

### 4 结束语

本研究是基于 Yolov4 目标检测算法,将图像信息传输给树莓派,再由树莓派传输给服务器,服务器会根据相应算法进行处理后,将处理的数据传回到树莓派中。机器人通过芯片的返回数据执行相应的操作,将垃圾进行分类。自动充电和自动避障也是相同的原理来实现的。该机器人获取外界信息靠的是摄像头,而非各种传感器,这种设计不仅能减轻机器人的质量,提升机器人的灵活性,也大大减少了使用其它传感器产生的费用。

### 参考文献

- [1] Joseph Redmon, Ali Farhadi. YOLOv3: An Incremental Improvement [J]. Cornell University. 2018.arXivpreprint arXiv:1804.02767.
- [2] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection [J]. Cornell University. 2020.arXiv preprint arXiv:2004.10934.
- [3] LIU Shu, QI Lu, QIN Haifang, et al. Path Aggregation Network for Instance Segmentation [J]. Cornell University. 2020. arXiv preprint arXiv:1803.01534.
- [4] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. Mask R-CNN [J]. Cornell University. 2020. arXiv preprint arXiv:1703.06870.

根据表 1 数据可知,在起始点与目标点距离为 5 000 m~5 500 m,本文提出方法的飞行耗时低于原有二种方法。

### 3 结束语

基于多目标搜索的无人机协同轨迹智能规划方法,实现了特定距离内飞行耗时的降低,对于无人机任务的协同执行有很大意义。

### 参考文献

- [1] 王渊,魏政磊,黄长强,等. 不确定条件下无人机编队在线协同轨迹规划研究 [J]. 西北工业大学学报, 2018, 36(6): 126-136.
- [2] 杨勇,丁勇,黄鑫城. 改进 APF 与 Bezier 相结合的多无人机协同避障航路规划 [J]. 电光与控制, 2018, 25(11): 36-41.
- [3] 段海滨,霍梦真,范彦铭. 仿鹰群智能的无人机集群协同对抗飞行验证 [J]. 控制理论与应用, 2018, 35(12): 111-118.