

基于机器人操作系统的智能移动小车目标识别研究

田 宇

(上海理工大学机械工程学院, 上海 200093)

✉893570276@qq.com



摘 要: 根据目前常见目标识别算法检测较为复杂、识别时间较长并容易出现错检漏检等方面的不足, 利用目前机器视觉技术, 提出一种适用智能移动小车系统的目标识别算法YOLOv3(一种深度学习目标检测方法), 并搭建智能移动小车的目标识别仿真系统。其目的是研究目标识别算法在ROS(机器人操作系统)中结合智能小车运动控制功能进行正确实现, 改善目标检测算法的漏检率、错误率等。分析目标识别算法在ROS系统中的精确度, 使用数据集进行检测实验。实验结果表明, 运用YOLOv3深度学习目标识别算法在提高识别目标速度的同时有较高的检测精度, 当目标丢失在视野中时利用ROS系统控制移动机器人重新使目标快速识别, 降低了识别目标的时间, 提高了移动机器人运动识别的效率。

关键词: ROS系统; 深度学习; YOLOv3识别算法; 运动控制

中图分类号: TP391.4 **文献标识码:** A

Research on Target Recognition of Intelligent Mobile Car based on Robot Operating System

TIAN Yu

(School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

✉893570276@qq.com

Abstract: Currently, common target recognition algorithms have shortcomings of complex detection, long recognition time and being prone to error detection and missed detection. In order to solve these problems, this paper proposes a target recognition algorithm YOLOv3 (A deep learning target detection method) that is suitable for intelligent mobile car system by using the current machine vision technology. Meanwhile, target recognition simulation system of the intelligent mobile car is built in order to study the correct implementation of target recognition algorithm in ROS (Robot Operating System) combined with intelligent car motion control function, thus reducing miss detection rate and error rate of target detection algorithm. Accuracy of target recognition algorithm in ROS system is analyzed and data set is used for detection experiment. Experimental results show that the proposed YOLOv3 deep learning target recognition algorithm can improve target recognition speed and detection accuracy. When a target is lost in the field of vision, ROS system is used to control the mobile robot to recognize the target quickly again, which reduces target recognition time and improves the efficiency of mobile robot motion recognition.

Keywords: ROS system; deep learning; YOLOv3 recognition algorithm; motion control

1 引言(Introduction)

随着人工智能和机器人技术的发展, 机器视觉成为机器人和计算机领域新兴的应用^[1], 而基于深度学习的视觉识别也成为目标识别中的主流, 特别是利用移动式、腿足式机器人进行动态识别成为当前的研究重点, 但腿足式机器人研究要结合复杂步态进行, 较为困难, 故现对移动式智能机器人小

车进行研究。在持续识别检测目标的过程中, 机器人与目标需要进行实时交互, 所以需要构建实时准确的机器人识别系统。而ROS能够提高机器人的功能开发效率, 受到机器人科研人员和开发人员的青睐, 在国内机器人研究中, 利用ROS对地图建设、路径计划、导航和模拟、运动控制和定位等进行了研究。使用ROS进行移动机器人的交互在检测跟踪方

面具有重要的研究意义,将基于深度学习的目标识别算法和ROS系统进行有效的结合,形成高效的实时仿真环境是值得研究的热门课题。

2 ROS平台简介(A brief introduction of ROS platform)

机器人操作系统(ROS)是一个基于机器人应用的开源系统框架,它包含大量工具软件、库代码和约定协议。使用ROS平台是为了提高机器人软件开发中的软件复用率,为机器人开发者提供底层驱动和控制、节点间的通信等功能,同时还包括获取、编译、封装运行程序所使用的工具和API(操作系统的应用程序接口)^[2]。它的五个特点为:(1)多语言的支撑;(2)框架简单,高集成度;(3)功能包、工具包种类繁多;(4)点对点的通信;(5)免费且开源。

ROS的节点控制:节点是机器人实现应用的载体,各节点实现不同的目的,它们互相通过节点控制器进行连接通信。节点控制器通过管理如目标识别节点、传感器控制节点、键盘控制节点、运动控制节点等实现整个系统的实际应用,如图1所示。

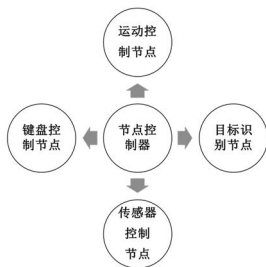


图1 系统节点连接图

Fig.1 System node connection diagram

3 基于深度学习的目标识别(Target recognition based on deep learning)

3.1 目标识别算法对比

比较流行的深度学习方法主要分为One-stage(SSD、YOLO系列)和Two-stage(R-CNN系列)。Two-stage包括:(1)R-CNN算法:①给出一张图片;②从图片中随机生成1,000多个方框;③对搜索到的每一个方框进行特征提取;④使用分类器对搜索到的方框进行分类;⑤通过回归的方法定义新的目标方框。但此方法有很明显的不确定性,它的特征提取有很大的冗余和重叠。(2)Fast R-CNN算法:在R-CNN的基础上进行深入研究,但选择性搜索仍然比较慢,计算量还是很大。(3)Faster R-CNN算法:在Fast R-CNN的基础上再进行改进^[3]。此时该方法把识别检测分为四个步骤:候选框选择、提取特征、提取后分类、Bounding Box(预测框)预测回归,但其缺陷是还需要花费不少的时间来训练。

VOC2007数据集上给出Faster R-CNN和YOLOv3的

检测模型,Faster R-CNN和YOLOv3在mPA(平均精度)和FPS(每秒所能达到的帧数)上的评估结果如表1所示。

表1 目标检测模型在VOC2007测试集上的表现评估结果
Tab.1 Performance evaluation results of target detection model in VOC2007 test set

目标检测模型	mAP	FPS
Faster R-CNN	0.785	7
YOLOv3	0.705	45

结果表明,YOLO方法解决了CNN(卷积神经网络)系列算法速度慢的问题,而且能保证平均精度在一定高度的范围内。它主要使用了回归的思想,通过对一张图片的整体进行研究,在一张图片的多个位置上回归出它的目标方框以及反映出目标的类别。由于在简单环境内需要检测的目标很少,因此在评估结果和保证实时方面,我们使用YOLOv3结合ROS进行此系统的目标识别实现。

3.2 YOLOv3算法网络框架

YOLO算法将所需检测的图片进行 $N \times N$ 的栅格化^[4](这是一种非常具有创新性的想法),然后将图片中一个目标的预测位置与目标中心位置所在最小单元格的Bounding Box进行相交。该算法大大减少了模型在图片中搜索的冗余和复杂度,将位置识别和类型检测封装到一个CNN网络中进行预测,判断图片中几种目标的所属类型和位置信息。CNN网络是一种深层次的网络结构^[5],CNN网络结构从输入层到输出层中间包含两个隐藏层,隐藏层里包含卷积层、池化层、全连接层,如图2所示。

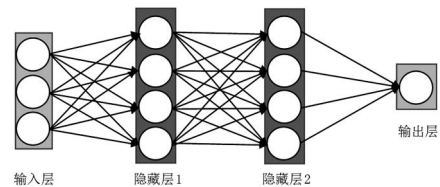


图2 CNN网络结构

Fig.2 CNN network structure

YOLOv3的主干特征提取网络的部分叫作Darknet-53^[6]。YOLOv3输入一个图片,主干特征提取网络就会对输入的图片不断地进行特征提取,从下到上为一系列卷积的过程。其中底层三个方向使用残差网络,Darknet-53中的残差卷积就是首先进行一次 3×3 的步长为2的卷积,然后保留该卷积层,再进行一次 1×1 的卷积和一次 3×3 的卷积,并把这个结果加上Layer作为输出的结果,在输出的三个方向上再进行残差网络(Residual)处理。YOLOv3尺度输出网络架构如图3(a)所示,可视化模型预测对网络的三个方向上的Box进行质心预测架构如图3(b)所示。

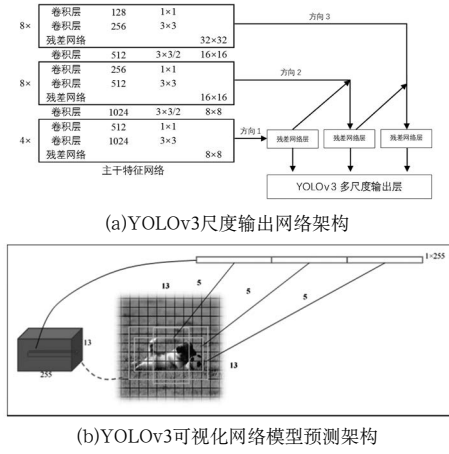


图3 YOLOv3网络架构

Fig.3 YOLOv3 network structure

残差网络最初是用在ResNet上,把残差网络分作一个个Block,每一个Block可以有输入也可以有输出,残差网络块对输入进行两部分的处理结构如图4所示。

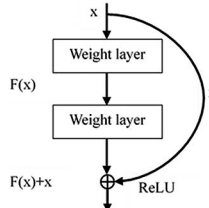


图4 残差网络结构

Fig.4 Residual network structure

3.3 YOLOv3算法的实现

YOLO的方法并非直接预测实际框架的宽度和高度(W, H),而是将预测的宽度和高度分为Anchor Box(Faster R-CNN中提出的预测锚框,后来演变为预测Bounding Box)的宽度和高度来连接。经过多次迭代训练,每个Bounding Box都会知道它是在预测什么样的Box。Bounding Box预测用式(1)和式(2)求得预测框的宽和高:

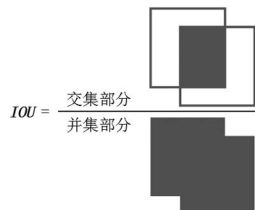
$$b_w = a_w e^{t_w} \quad (1)$$

$$b_h = a_h e^{t_h} \quad (2)$$

其中, a_w 和 a_h 为Anchor Box的宽和高, t_w 和 t_h 为Bounding Box直接预测出的宽和高, b_w 和 b_h 为转换后预测的实际宽和高,也就是最终预测中输出的宽和高。置信度如式(3)所示:

$$C_i^j = P_r(Object) \times IOU_{pred}^{truth} \quad (3)$$

其中, C_i^j 表示第 i 个最小单元格的第 j 个Bounding Box的置信度。 IOU 的求解如下:



当Bounding Box认为当前边框内有目标时,要检测的所有类型中每种类型的概率为对象条件类别概率(Conditional Class Probabilities),YOLOv3的对象条件类别概率中对背景的预测通过置信度来分类识别,所以它的输出是在置信度表示边框有目标的前提下,而它的条件概率为 $P_r(class_i | Object)$,所以置信度评测由 $P_r(class_i | Object) \times C_i^j$ 给出。

若一张图片被分为 $S \times S$ 个单元格,并且有 B 个Bounding Box,根据图3(b)预测的三个方向的Box,即每个Bounding Box内有5个位置参数含有1个置信度和 $classes$ 个类别概率,那么得到最终的输出维数如式(4)所示:

$$S \times S \times [B \times (4 + 1 + classes)] \quad (4)$$

根据YOLOv3算法进行直观检测,该算法的检测在图中能精确快速地识别。在单目标识别中有较精确的效果,如图5(a)所示;在多目标识别中存在极少数漏检情况,但识别精度仍然保持一定的高度,如图5(b)所示,而且有较快的识别速度。故将在该算法下研究ROS智能移动小车识别系统实验,为移动小车的智能应用提供基础,搭载YOLO识别功能包与ROS进行通信,达到准确高效的识别效果。

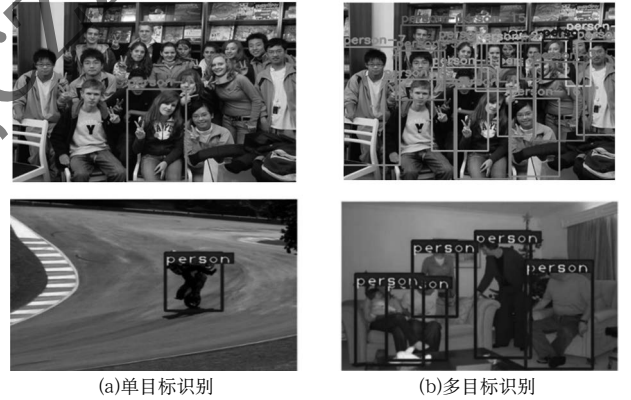


图5 检测结果效果图

Fig.5 Effect picture of test results

4 目标识别移动小车实验系统架构搭建 (Construction of experiment system architecture for target recognition mobile car)

此目标识别系统的功能包架构如图6所示。

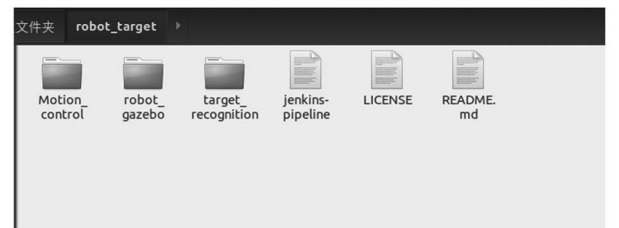


图6 目标识别系统整体目录架构

Fig.6 Overall directory architecture of target recognition system

robot_gazebo为小车三维仿真环境及其Gazebo(可视化仿真环境)世界的建模功能包; Motion_control为移动机器人小车运动控制功能包; target_recognition为封装的YOLOv3目标识别功能包,三者相互通信,形成一个以移动机器人小车为仿真实验平台的目标识别系统。

4.1 搭建仿真智能小车环境

该模型采用XACRO,是优化后的URDF模型,是一种精简、可复用、模块化的描述形式。模型文件的后缀名由.urdf变为.xacro,而且在模型标签中需要加入XACRO的声明,使模型与ROS进行连接:

```
<?xml version="1.0"?>
<robot name="mrobot" xmlns:xacro="http://www.
ros.org/wiki/xacro">
```

搭载简要移动机器人小车模型和其他外部环境需要定义各种参数,例如定义车身颜色、惯性矩阵、传动系统的关节连接,等等^[7]。此处利用mrobot_gazebo模型小车,如图7所示。

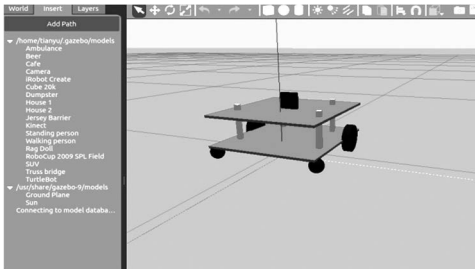


图7 ROS移动小车搭建

Fig.7 Construction of ROS mobile car

此环境搭建好后可以加入其他外部models搭建整体的world环境,为后续识别更多目标提供模型。此环境封装为robot_gazebo功能包。此环境加载摄像头插件camera.xacro,便于搭载视觉系统。

4.2 运动控制平台搭建

移动小车模型中已经加入了libgazebo_ros_diff_drive.so插件,可以使用差速控制器实现机器人运动,由于和控制运动有关的话题是/cmd_vel,故在终端输入roslaunch robot_control以便执行键盘运动控制节点,同时执行rostopic echo/cmd_vel来查询小车运动信息,此时控制键盘即可控制仿真小车的速度、方向等。小车的运动控制界面如图8所示。

```
Control mrobot!
-----
Moving around:
  u   i   o
  j   k   l
  m   ,   .

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
space key, k : force stop
anything else : stop smoothly
```

图8 移动小车运动控制界面

Fig.8 Mobile car motion control interface

由于小车为两驱轮式驱动,以直角坐标系 XOY 为基础,小车运动学方程^[8]如式(5)所示:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{lv} \\ v_{rv} \end{bmatrix} \quad (5)$$

左右两轮角速度如式(6)和式(7)所示:

$$V_{\omega v(\text{左})}(t) = \frac{V_L(t)}{R(t) - \frac{L}{2}} \quad (6)$$

$$V_{\omega v(\text{右})}(t) = \frac{V_R(t)}{R(t) + \frac{L}{2}} \quad (7)$$

在 t 时刻姿态位置变化如式(8)所示,姿态位置随时间变化如式(9)和式(10)所示:

$$\theta(t) = \theta(0) + \int_0^t \frac{V_R(t) - V_L(t)}{L} dt \quad (8)$$

$$x_w(t) = x_w(0) + \int_0^t \frac{V_R(t) + V_L(t)}{2} \cos \theta(t) dt \quad (9)$$

$$y_w(t) = y_w(0) + \int_0^t \frac{V_R(t) + V_L(t)}{2} \sin \theta(t) dt \quad (10)$$

其中, V_{lv} 为移动小车质心线速度, V_{rv} 为小车转动角速度, L 为两驱动轮的距离, θ 为小车转弯的角度, V_L 和 V_R 分别为小车左右驱动轮的线速度, R 为小车某时刻回转半径。启动rostopic相关话题查询如图9所示的移动小车运动信息显示界面。

```
tianyu@tianyu-vpc:~$ rostopic echo /cmd_vel
linear:
  x: 0.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
--
linear:
  x: 0.0
  y: 0.0
  z: 0.0
```

图9 移动小车运动信息界面

Fig.9 Mobile car movement information interface

了解到小车运动方位和姿态的位置信息,将此环境封装为Motion_control功能包。搭建移动小车仿真平台后进行目标识别模块的连接和通信,运动识别整体流程框架如图10所示。首先初始化移动小车,随后订阅小车运动的/cmd_vel节点来获得小车的速度、角度、方向等信息,若小车视野内存在要识别的目标则小车位置不发生改变,只获取识别到的信息;若小车视野内目标位置偏移或者不存在目标,则调整小车速度、方向等进行小车的运动控制,直至识别到目标重新出现在视野范围内。

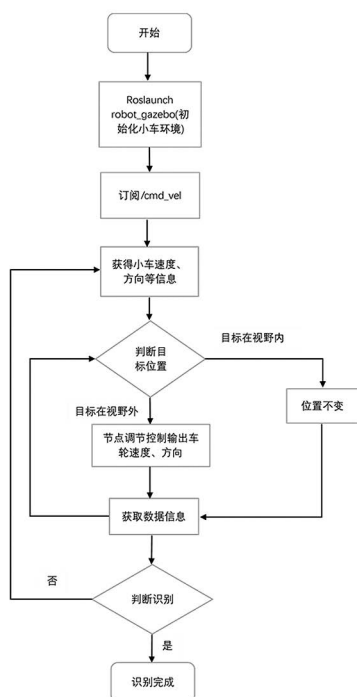


图10 运动识别流程图

Fig.10 Flow chart of motion recognition

4.3 目标识别在ROS中的实现

将YOLO算法与ROS进行通信, 需要建立移动小车仿真环境、运动控制和目标识别间的节点通信^[9], 使用数据集进行实验。将目标识别封装为target_recognition功能包, 通过Gazebo节点进行三类节点的控制通信, 利用/cmd_vel进行运动控制节点连接, 利用camera/image_raw进行ROS中目标识别算法节点连接, 利用/joint_states进行移动小车仿真环境节点连接, 三者构成基于ROS移动小车的目标识别系统^[10]。目标识别小车系统节点通信如图11所示。

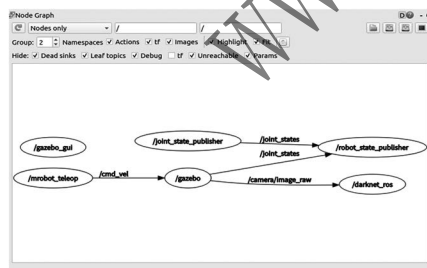


图11 ROS目标识别小车系统节点通信示意图

Fig.11 Node communication diagram of ROS target recognition car system

通过虚拟机的Ubuntu系统中ROS各个节点控制, 完成仿真系统的研究, 以YOLOv3和ROS结合为主要识别算法进行实验, 与R-CNN系列算法进行比对, 得出实验结果。虽然在虚拟机下实现识别有一定的不确定性, 但是在简单识别目标的前提下有很好的效果, 在错检、漏检和速度方面有较大改善。在ROS中的识别实验效果如图12所示。

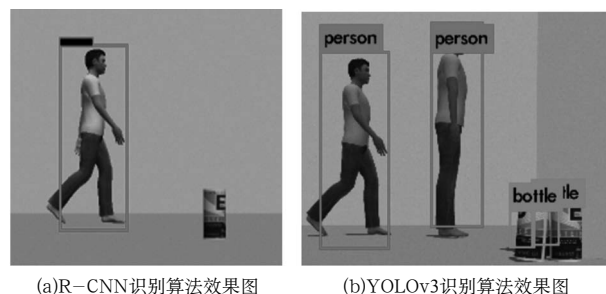


图12 仿真实验效果图

Fig.12 Simulation experiment effect diagram

5 结论(Conclusion)

本文以虚拟机Ubuntu 18.04系统的ROS平台, 创建移动小车XACRO模型, 封装运动控制、键盘驱动功能包, 以YOLOv3为识别算法, 分析算法的优点, 搭建相机插件, 进行仿真移动小车目标识别实验。该深度学习算法在ROS中错误识别、遗漏识别的情况很少出现, 表现出较好的精度和速度。后续将在高效的GPU平台上针对复杂环境, 对检测实时性开展进一步的研究。

参考文献(References)

- [1] 高娟娟, 梁中豪, 宋亚青. 机器视觉技术研究和应用现状及发展趋势[J]. 中国传媒科技, 2020(07):21-22.
- [2] 詹润哲, 姜飞. 基于ROS与深度学习的移动机器人目标识别系统[J]. 电子测试, 2018(15):70-71, 64.
- [3] 曹之君, 张良. 基于Faster-RCNN的快速目标检测算法[J]. 航天控制, 2020, 38(04):49-55.
- [4] REDMON J, FARHADI A. YOLOv3: An incremental improvement[J]. IEEE Conference on Computer Vision and Pattern Recognition, 2018(04):89-95.
- [5] 王琦. 卷积核相关性与CNN网络结构优化的关系研究[D]. 兰州: 兰州大学, 2019.
- [6] 王辉, 张帆, 刘晓凤, 等. 基于DarkNet-53和YOLOv3的水果图像识别[J]. 东北师大学报(自然科学版), 2020, 52(04):65-70.
- [7] 张玉林, 陈勇林. 基于ROS平台的移动机器人运动分析与仿真[J]. 机器人技术与应用, 2018, 185(05):37-40.
- [8] 许松清, 吴海彬, 杨兴裕. 两轮驱动移动机器人的运动学研究[J]. 电子机械工程, 2005, 21(06):31-34.
- [9] 张亚斌. 基于ROS的轮式移动机器人行人目标视觉感知与跟踪系统研究[D]. 徐州: 中国矿业大学(江苏), 2019.
- [10] 贾浩东. 基于ROS平台的机器人目标识别系统的设计[J]. 电子测试, 2018(02):49-50, 59.

作者简介:

田 宇(1994-), 男, 硕士生. 研究领域: 机器视觉.