计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(11):227−231 [doi: 10.15888/j.cnki.csa.007594]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于深度学习的人手视觉追踪机器人①

林粤伟[1,2], 牟　森[1]

[1](青岛科技大学 信息科学技术学院, 青岛 266061)
[2](海尔集团博士后工作站, 青岛 266000)
通讯作者: 林粤伟, E-mail: linyuewei@qust.edu.cn

摘　要: 视觉追踪是智能机器人的核心功能之一, 广泛应用于自动驾驶、智慧养老等领域. 以低成本树莓派作为下位机机器人平台, 通过在上位机运行事先训练好的深度学习 SSD 模型实现对人手的目标检测与视觉追踪. 基于谷歌 TensorFlow 深度学习框架和美国印第安纳大学 EgoHands 数据集对 SSD 模型进行训练. 机器人和上位机的软件使用 Python 在 Linux 系统下编程实现, 两者之间通过 WiFi 进行视频流与追踪控制命令的交互. 实测表明, 所研制智能机器人的视觉追踪功能具有良好的稳定性和性能.

关键词: 深度学习; SSD 模型; 树莓派; 计算机视觉; 机器人

引用格式: 林粤伟,牟森.基于深度学习的人手视觉追踪机器人.计算机系统应用,2020,29(11):227–231. http://www.c-s-a.org.cn/1003-3254/7594.html

Human Hands Visual Tracking Robot Based on Deep Learning

LIN Yue-Wei[1,2], MU Sen[1]

[1](College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China)
[2](Postdoctoral Workstation of Haier Group, Qingdao 266000, China)

Abstract: Vision tracking is one of the core functions of smart robots, and widely used in automatic driving, intelligent pension and other fields. The low-cost Raspberry Pi is employed as the slave computer robot platform. The object detection and visual tracking of human hands is implemented through running the pre-trained deep learning SSD model on host computer. The SSD model is trained based on Google's TensorFlow deep learning framework and US Indiana University's EgoHands dataset. Both of the robot and host computer's software is written by Python in Linux systems. Video stream and tracking control commands are exchanged between robot and host via WiFi. The practical tests show that the vision tracking function of the developed smart robot has good stability and performance.

Key words: deep learning; SSD model; Raspberry Pi; computer vision; robot

智能机器人的开发是科学研究、大学生科技创新大赛的热点, 基于计算机视觉的目标检测技术在智能小车、无人机、机械臂等领域得到了广泛应用. 在企业界, 零度智控公司开发了 Dobby (多比)、大疆公司开发了 Mavic 等, 研发出了具有视觉人体追踪与拍摄功能的家用小四轴自拍无人机. 在学术界, 文献 [1] 从检测、 跟踪与识别三方面对基于计算机视觉的手势识别的发展现状进行了梳理与总结; 文献 [2] 基于传统的机器学习方法-半监督学习和路威机器人平台实现了视觉追踪智能小车; 文献 [3] 基于微软 Kinect 平台完

① 基金项目: 青岛科技大学教学改革研究面上项目 (2018MS44); 青岛市博士后应用研究项目
Foundation item: General Program of Education Reform of Qingdao University of Science and Technology (2018MS44); Post Doctorial Application Research of Qingdao City
收稿时间: 2020-01-08; 修改时间: 2020-02-08, 2020-03-17; 采用时间: 2020-03-24; csa 在线出版时间: 2020-10-29

成了视觉追踪移动机器人控制系统的设计; 文献 [4] 对服务机器人视觉追踪过程中的运动目标检测与跟踪算法进行研究并在 ROS (Robot Operating System, 机器人操作系统) 机器人平台进行实现.

上述视觉追踪功能的实现大多采用传统的目标检测方法, 基于图像特征和机器学习, 且所采用平台成本相对较高. 近年随着大数据与人工智能技术的兴起, 利用深度学习直接将分类标记好的图像数据集输入深度卷积神经网络大大提升了图像分类、目标检测的精确度. 国内外基于 Faster R-CNN (Faster Region-Convolutional Neural Network, 更快的区域卷积神经网络)、YOLO (You Only Look Once, 一种 single-stage 目标检测算法)、SSD (Single Shot multibox Detector, 单步多框检测器) 等模型的深度学习算法得到广泛应用, 如文献 [5] 将改进的深度学习算法应用于中国手语识别. 本文基于深度学习[6] 技术, 在低成本树莓派[7] 平台上设计实现了视觉追踪智能机器人 (小车), 小车能够通过摄像头识别人手并自动追踪跟随人手. 与现有研究的主要不同之处在于使用了更为经济的低成本树莓派作为机器人平台, 并且在目标检测的算法上使用了基于 TensorFlow[8] 深度学习框架的 SSD 模型, 而不是基于传统的图像特征和机器学习算法.

1 关键技术

1.1 系统架构

如图 1, 整个系统分为机器人小车 (下位机) 和主控电脑 (上位机) 两部分. 上位机基于深度学习卷积神经网络做出预测, 下位机负责机器人的行进以及视频数据采集与传输, 两者之间通过 WiFi 通信. 其中, 小车主控板为开源的树莓派 3 代 B 开发板, CPU (ARM 芯片) 主频 1.2 GHz, 运行有树莓派定制的嵌入式 Linux 操作系统, 配以板载 WiFi 模块、CSI 接口摄像头、底盘构成下位机部分. 上位机操作运行事先训练好的 SSD 模型[9]. 小车摄像头采集图像数据, 将其通过 WiFi 传输给上位机, 并作为 SSD 模型的输入. SSD 模型如果从输入的图像中检测到人手, 会得到人手在图像中的位置, 据此决定小车的运动方向和距离 (需要保持人手在图像中央), 进而向小车发送控制命令, 指示运动方向和距离. 小车收到上位机发来的远程控制命令后, 做出前进、转向等跟踪人手的动作. 智能小车和主控电脑两端皆运行用 Python[10] 编写的脚本程序.

1.2 深度学习 SSD 模型

SSD 模型全名为 Single Shot multibox Detector[9], 是一种基于深度学习的 one stage (一次) 目标检测模型. SSD 模型由一个基础网络 (base network) 的输出级后串行连接几种不同的辅助网络构成, 如图 2 所示. 不同于之前 two stage 的 Region CNN[11], SSD 模型是一个 one stage 模型, 即只需在一个网络中即可完成目标检测, 效率更高.

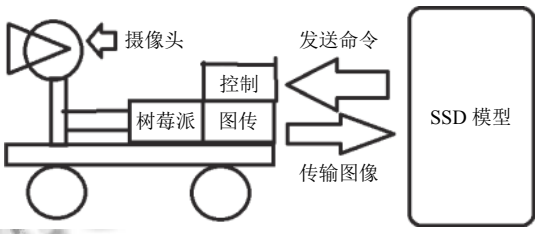

图 1 智能机器人系统架构

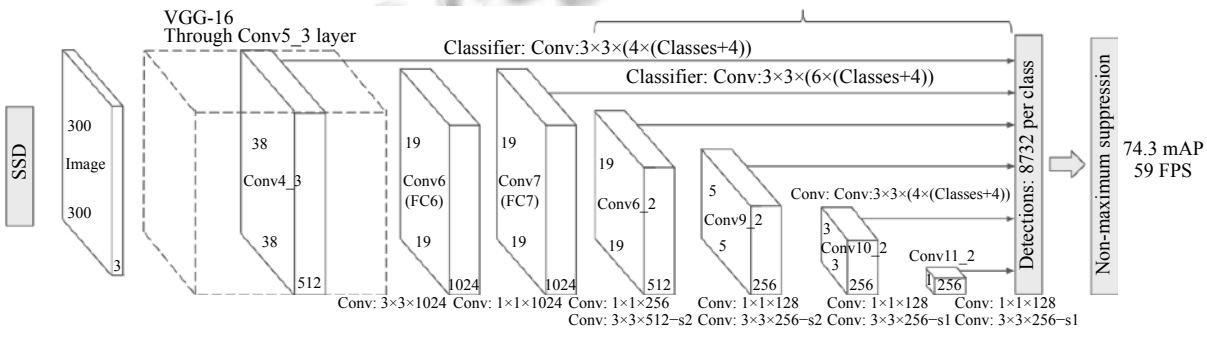

图 2 SSD 模型

SSD 模型采用多尺度特征预测的方法得到多个不同尺寸的特征图[9]. 假设模型检测时采用 m 层特征图, 则得到第 k 个特征图的默认框比例公式如式 (1):

$$S_k = S_{\min} + \frac{S_{\max} - S_{\min}}{m-1}(k-1), k \in \{1,2,\cdots,m\} \quad (1)$$

其中, S_k 表示特征图上的默认框大小相对于输入原图

的比例 (scale). 一般取 S_{min}=0.2, S_{max}=0.9. m 为特征图个数.

SSD 模型的损失函数定义为位置损失与置信度损失的加权和[9], 如式 (2) 所示:

$$L(x,c,l,g)=\frac{1}{N}(L_{conf}(x,c)+\alpha L_{loc}(x,l,g)) \quad (2)$$

其中, N 表示与真实物体框相匹配的默认框数量; c 是预测框的置信度; l 为预测框的位置信息; g 是真实框的位置信息; α 是一个权重参数, 将它设为 1; $L_{loc}(x,l,g)$ 位置损失是预测框与真实框的 Smooth L1 损失函数; $L_{conf}(x,c)$ 是置信度损失, 这里采用交叉熵损失函数.

1.3 TensorFlow 平台

使用谷歌 TensorFlow 深度学习框架对 SSD 模型进行训练. TensorFlow 能够将复杂的数据结构传输至人工智能神经网络中进行学习和预测, 近年广泛应用于图像分类、机器翻译等领域. TensorFlow 有着强大的 Python API 函数, 而本文实现的智能小车和主控电脑端运行的程序皆为 Python 脚本, 可以方便的调用 Python API 函数.

2 设计与实现

系统主程序软件流程如图 3 所示. 上位机运行自行编写的 Python 脚本作为主程序, 接收下位机发来的图像, 并将其输入到事先训练好的深度学习 SSD 模型中, 以检测人手目标. 若检测到人手, 则产生、发送控制命令至下位机. 下位机运行两个自行编写的 Python 脚本, 其中一个脚本基于开源的 mjpg-streamer 软件采集、传输图像至上位机, 另一个接收来自上位机的控制命令并通过 GPIO 端口控制车轮运动.

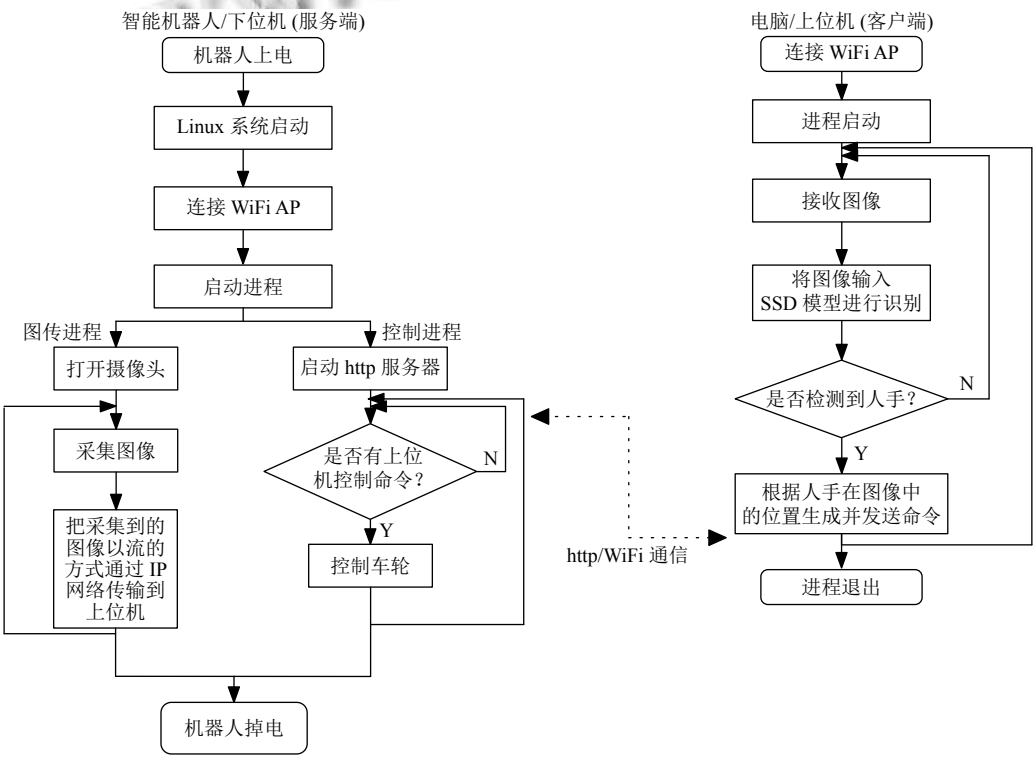

图 3 主程序软件流程

2.1 深度学习 SSD 模型训练

上位机电脑和 CPU 型号为联想 Thinkpad E540 酷睿 i5 (第 4 代), 操作系统为 Ubuntu 16.04 LTS 64 位, TensorFlow 版本为 v1.4.0, 采用 TensorFlow Object Detection API 的 SSD MobileNet V1 模型. 训练数据直接使用了美国印第安纳大学计算机视觉实验室公开的 EgoHands 数据集, 该数据集是一个向外界开放下载的 1.2 GB 的已经标注好的数据集, 用谷歌眼镜采集第一视角下的人手图像数据, 例如玩牌、下棋等场景下人手的姿态. 首先对数据集进行数据整理, 将其转换为 TensorFlow 专有的 TF Record 数据集格式文件, 然后修改 TensorFlow 目标检测训练配置文件 ssd_

mobilenet_v1_coco.config. 训练全程在电脑上由通用 CPU 运行, 共运行 26 小时. 结束训练后将 protobuf 格式的二进制文件 (真正的 SSD 模型) 保存下来以便下文介绍的上位机 Python 主程序调用.

2.2 上位机设计

考虑到小车回传视频的帧数比较高, 且深度学习神经网络的计算也是一件耗时的任务, 在上位机主程序 (Python 脚本) 中建立了两个队列, 一个输入队列用来存储下位机传来的原始图像, 一个输出队列用来存储经神经网络运算处理之后带有标注结果的图像. 上位机通过开源软件 OpenCV 的 cv2.videoCapture 类用文件的方式读取视频信息. 运行 SSD 目标检测模型进行人手识别时, 会得到目标的标注矩形框中心, 当中心落到整幅图像的左侧并超出一定距离时, 产生 turnleft 左转指令; 当中心落到整幅图像右侧且超出一定距离的时, 产生 turnright 右转指令; 当中心落到图像的上半部分并超过一定距离时, 产生 forward 前进指令. 距离值默认设定为 60 个像素, 该参数可修改. 预测小车行进方向功能的伪代码如算法 1 所示.

算法 1. 上位机预测行进方向伪代码

```
Require: 距离阈值 (默认为 60 像素)
  while 全部程序就绪 do
    if 没有识别到目标:
      Continue;
    else if 识别到目标:
      if 识别到目标面积过大, 目标离摄像头太近:
          Send("stop");
       else:
          if 目标中心 x<640/2–距离阈值:
            Send("turnleft");
          if 目标中心 x>640/2+距离阈值:
            Send("turnright");
          else:
            Send("forward");
  end while
```

2.3 下位机设计

下位机基于低成本树莓派平台实现, 使用开源软件 Bottle 部署了一个多线程的 HTTP 服务器, 该服务器接收上位机发出的 HTTP POST 请求, 提取其中的控制命令进行运动控制. 使用开源软件 mjpg-streamer 控制网络摄像头采集图像, 并将图像数据以视频流的方式通过 IP 网络传输到上位机客户端.

3 测试结果与评估

搭建局域网环境 (也支持广域网), 使上位机和下位机接入同一无线路由器. 当摄像头采集到的画面右侧出现人手时, 如图 4 所示的实时图像中, 标注方框标记出了检测到的人手的位置, 同时控制台输出 turnright (右转) 控制命令, 此时小车向右侧做出移动. 当屏幕中没有人手时, 画面上面没有用彩色画出的区域, 上位机的终端也不打印输出任何控制命令.

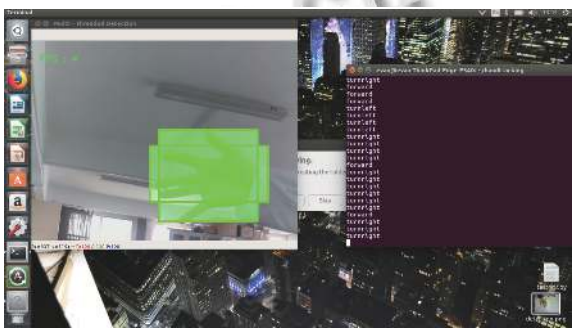

图 4 人手目标检测功能测试结果

功能方面, 针对人手在小车视野的不同位置情况进行所研制小车人手视觉追踪的功能测试. 比如, 当人手在小车前方且完整出现时, 上位机应发出 forward (前进) 命令, 进而小车收到该命令后向前行进. 当小车视野里没有或只有部分人手时, 应当无命令输出, 小车原地不动. 功能测试用例如表 1 所示, 测试结果均为预期的正常结果. 性能方面, 所采用基于深度学习 SSD 模型的人手目标检测算法的准确性与实时性较好, 算法的 mAP (平均精准度) 为 74%, 检测速率 40 fps 左右, 可以较好的满足系统要求.

表 1 人手视觉功能测试结果

测试用例	输出命令	小车动作
手在小车正前方60 cm处	forward	向前行进
手在小车左前方60 cm处	turnleft	向左行进
手在小车右前方60 cm处	turnright	向右行进
手在小车正前方130 cm处	forward	向前行进
手在小车左前方130 cm处	turnleft	向左行进
手在小车右前方130 cm处	turnright	向右行进
视野里只有半只手	无输出	原地不动
手在小车视野下方	无输出	原地不动

小车 (机器人平台) 外观如图 5 所示. 另外, 由于动态视频文件无法在论文中展示, 这里展示的是录制好

的测试视频中 2 个帧的截图, 如图 6 所示, 从小车的位置变化可以看出其可以追踪人手.

图 5 机器人外观

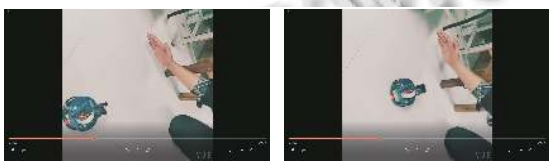

图 6 追踪人手

4 结论

本文利用深度学习 SSD 目标检测模型对目标进行识别, 将识别的结果用于修正智能小车机器人的行进路线, 满足了智能机器人的视觉追踪功能需求. 其特色主要在于采用了低成本树莓派, 以及深度学习而非传统的神经网络识别算法, 省去了设置特征的步骤. 系统暂时只能用来识别人手, 小车能够跟随人手移动, 功能稳定性与性能良好. 若要识别追踪其他物体, 可以使用其他自己制作或第三方数据集对 SSD 模型进行训练, 以把网络的识别对象训练成拟追踪的目标类型. 未来也可应用 5G 通信模块, 进行更为稳定低时延的视频传输与控制.

参考文献

1 Rautaray SS, Agrawal A. Vision based hand gesture recognition for human computer interaction: A survey. Artificial Intelligence Review, 2015, 43(1): 1–54. [doi: 10.1007/s10462-012-9356-9]

2 张子洋, 孙作雷, 曾连荪. 视觉追踪机器人系统构建研究. 电子技术应用, 2016, 42(10): 123–126, 130.

3 王道全. 基于视觉的智能追踪机器人的设计研究 [硕士学位论文]. 青岛: 青岛科技大学, 2016.

4 周燕秋. 服务机器人视觉追踪技术研究 [硕士学位论文]. 上海: 上海师范大学, 2018.

5 周舟, 韩芳, 王直杰. 改进 SSD 算法在中国手语识别上的应用. 计算机工程与应用: 1–7. http://kns.cnki.net/kcms/detail/11.2127.TP.20191207.1137.006.html. [2020-03-19].

6 Goodfellow I, Bengio Y, Courville A. Deep Learning 深度学习. 赵申剑, 黎彧君, 符天凡, 等译. 北京: 人民邮电出版社, 2017.

7 许艳, 孟令军, 王志国. 基于树莓派的元器件检测系统设计. 电子技术应用, 2019, 45(11): 63–67, 71.

8 郑泽宇, 梁博文, 顾思宇. TensorFlow: 实战 Google 深度学习框架. 2 版. 北京: 电子工业出版社, 2018.

9 Liu W, Anguelov D, Erhan D, *et al*. SSD: Single shot MultiBox detector. Proceedings of the 14th European Conference on Computer Vision. Amsterdam. 2016. 21–37.

10 Chun W. Python 核心编程. 孙波翔, 李斌, 李晗, 译. 3 版. 北京: 人民邮电出版社, 2016.

11 Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA. 2014. 580–587.