计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(9):142−148 [doi: 10.15888/j.cnki.csa.007392]

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

基于树莓派与神经计算棒的特种车辆检测识别①

陈　璐[1], 管霜霜[2], 谢艳芳[1]

[1](上海浦东临港智慧城市发展中心, 上海 201306)
[2](阿里巴巴科技 (北京) 有限公司, 北京 100102)

摘　要: 目前随着深度学习技术的不断发展, 越来越多的智能化应用应运而生, 用于训练和演算的硬件设备通常以GPU 为主, 在实际部署和使用过程中会产生较高硬件采购成本和用电成本. 因此针对现有深度学习系统中成本与算法可用性的平衡问题, 本文提出以树莓派与 Movidius 神经元计算棒为计算平台, 通过改进的 SSD+MobileNet 算法实现对车辆目标进行识别和检测, 并在实际环境中对训练的模型进行测试和调优, 最终达到满足实际使用的效果, 处理速度为平均每秒 4 帧. 通过实验结果表明, 在树莓派这样计算能力较弱的平台上, 可以通过类似于 Movidius 神经元计算棒这样的 VPU 模块来实现算法的加速, 在满足实际使用的情况下还可以大幅度降低计算成本.

关键词: 树莓派; 深度学习; Movidius; 目标检测与识别; 低成本

引用格式: 陈璐,管霜霜,谢艳芳.基于树莓派与神经计算棒的特种车辆检测识别.计算机系统应用,2020,29(9):142–148. http://www.c-s-a.org.cn/1003-3254/7392.html

Truck Detection Method Based on Raspberry PI and Movidius Neural Computing Stick

CHEN Lu[1], GUAN Shuang-Shuang[2], XIE Yan-Fang[1]

[1](Shanghai Lingang Smart City Development Center, Shanghai 201306, China)
[2](Alibaba Technology (Beijing) Co. Ltd., Beijing 100102, China)

Abstract: With the rapid development of deep learning technology, more and more intelligent algorithms have been applied. The hardware equipment used for training and calculation is mainly GPU, which will incur high hardware procurement cost and power consumption cost in the actual deployment and use. Therefore, aiming at the high cost of the current deep learning system, this study proposes to use raspberry PI and Movidius neuron computing stick as the computing platform. SSD+MobileNet algorithm is adopted to realize the recognition and detection of vehicle targets, and the training model is tested and optimized in the actual environment to finally meet the effect of actual use, with a processing speed of 4 frames per second on average. The experimental results show that on the platform with weak computing power like raspberry PI, the algorithm can be accelerated by VPU modules like Movidius neuron computing stick, and the computing cost can be greatly reduced when it is in actual use.

Key words: raspberry PI; deep learning; Movidius; object detection and identification; low cost

车辆检测与识别技术是目标检测的重要分支, 通过监控摄像头对路面行驶的车辆进行检测和识别, 可以实现对不同车型进行分类和检测预警, 例如渣土车、油罐车以及公交车等特种车辆. 上海临港地区作为临港自贸新片区的重要载体, 处于高速建设发展过程中, 因此时常有非法渣土车等特种车辆进入主城区, 对于城运中心城市精细化管理而言, 需要对每一辆非法渣土车辆进行识别和跟踪, 以便为执法过程提供指

① 基金项目: 上海市临港地区产业专项 (RZ2018010201)
Foundation item: Special Fund for Industrialization of Shanghai Lingang Area (RZ2018010201)
收稿时间: 2019-10-04; 修改时间: 2019-10-29; 采用时间: 2019-11-14; csa 在线出版时间: 2020-09-04

挥调度和历史追溯. 但在实际使用过程中, 通常由于视频监控中的车辆目标受到光线照射、拍摄角度、复杂背景以及遮挡等多种因素, 使得车辆对象的检测和识别在计算机视觉存在一定难度[1]. 近些年由于大数据和云计算技术的快速发展, 算力水平得到进一步的提高, 深度学习技术也迎来快速发展. 在计算机视觉领域, 通过卷积神经网络来提取图像特征并实现目标回归检测已成为未来发展趋势, 但仍存在硬件成本和能耗较高等问题, 仍需进一步解决.

同其他目标检测研究内容一样, 基于摄像头的车辆检测过程主要分为: 生成候选窗口、提取特征以及车辆目标分类. 在经典机器学习算法中, 车辆目标检测通常采用滑动窗口的方式生成候选区域, 随后利用人工创建特征的方式提取图像特征, 经典的方法包括梯度方向直方图 (Histogram of Oriented Gradient, HOG)[2]、不变尺度转换算法 (Scale-Invariant Feature Transform, SIFT)[3] 以及多尺度 Haar 小波特征[4] 等. 在分类识别的阶段, 经典算法主要采用的分类器主要为支持向量机 (Support Vector Machine, SVM)[4]、自适应集分类器 (AdaBoost)[5] 等. 由此可见, 上述方法大多是基于人工创建特征进行的识别, 特征提取的语义信息属于较低的层次, 适用性并不强. 在实际应用中, 传统检测方法针对特定场景需要投入大量的时间精力设计不同的特征, 面临很大的挑战. 近些年来, 随着深度学习技术的迅速发展, 在各领域中的应用也逐渐成熟, 自从 2014 年 Girshick R 等[6] 在目标检测领域成功应用深度神经网络后, 目标检测的研究方向就基本上被深度学习相关的算法和框架所占领. 此外, 从整体算法框架和处理思路上看, 深度学习在检测问题上的算法大致分为两种: 一种是基于候选区域生成的检测算法, 与传统目标检测算法相似, 主要分为两个步骤, 即生成候选区、利用深度神经网络对区域特征进行提取以及最后进行分类, 这类方法以 RCNN[7]、Fast-RCNN[8]、Faster-RCNN[9] 系列算法为代表; 另一种是基于回归的检测算法, 即通过深度学习框架直接回归出图像中的目标对象以及分类, 主要以 YOLO[10]、SSD[11] 等算法为代表.

本文主要基于树莓派 3 代 B+作为基础开发板[12], 由于树莓派是一种计算资源非常有限的设备, 一般的深度学习算法很难运行在该设备上, 因此通过 Movidius 神经元计算棒[13] 来加速视频处理, 基于 MobileNet+SSD 算法进行改进, 针对车辆目标在监控摄像头中的画面属于中等比例目标的特性, 在不明显增加计算量的前提下, 对网络层数和结构进行优化, 更进一步的提取高层次图像语义特征. 经过实验验证, 该网络可以在树莓派 3 代 B+开发板上达到平均 4 帧/秒的处理速度, 为了更好进行训练, 本文基于 MSCOCO 数据集[14], 同时结合自定义数据集制作适合应用场景的数据训练集. 通过使用该数据集进行训练, 本文取得了较好的识别效果, 并在实际使用中进行了验证.

总之, 本文的主要总结如下:

(1) 针对于有限嵌入式设备, 本文通过 Movidius 神经元计算棒进行加速, 并验证其有效性和低成本.

(2) 基于 MobileNet+SSD, 通过增加高层图像语义的提取特征来提高网络的准确性, 并在有限的计算资源中取得实际应用的效果, 能够满足渣土车等特种车辆的检测与识别效果.

1 车辆目标检测模型的构建

1.1 车辆检测网络框架

在本文中, 为了降低车辆检测与识别的硬件成本, 我们采用基于神经元计算棒的树莓派 3B+作为算法运行环境, 由于该运行环境在存储资源和计算资源方面都十分有限, 所以像 GoogleLeNet[15]、VGG[16] 以及 ResNet[17] 等层级较深的网络模型都不适用. 因此为了实现车辆目标的检测和识别, 需要合理的设计一个网络结构, 在保持最终识别准缺率不下降太多的情况, 让算法模型所需的计算量和存储量都得到进一步的降低.

1.2 通过 MobileNet 模型提取特征

MobileNet 网络模型[18] 是由 Google 公司在 2017 年提出的一种轻量级的卷积神经网, 其主要的模块结构叫做通道可分离卷积 (depthwise seperable convolutions), 单一卷积模块结构如图 1 所示.

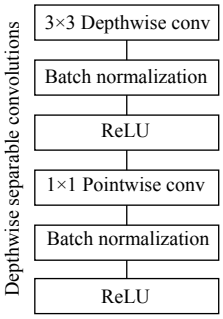


图 1 MobileNet 的通道可分离卷积块

MobileNet 的网络结构是基于深度级可分离卷积块的堆叠设计, 其网络结构基本思想就是将通道间的相关性和空间相关完全分离出来, 同时降低计算量和所需参数量. 与传统的卷积网络结构有所不同, MobileNet 属于深度可分卷积, 其最主要的特点就是对特征图中的各个通道进行卷积操作, 然后将卷积操作之后的各个特征图通道进行合并, 通过 1×1 卷积降低其通道数. MobileNet 网络架构中的关键是深度可分卷积, 因为其极大地降低了算法的复杂度, 适用于嵌入式设备的应用. 如图 2(b) 所示, 深度可分卷积通过将每个常规卷积层分成两部分, 深度卷积层和逐点卷积层使计算复杂度更适合于的移动智能设备

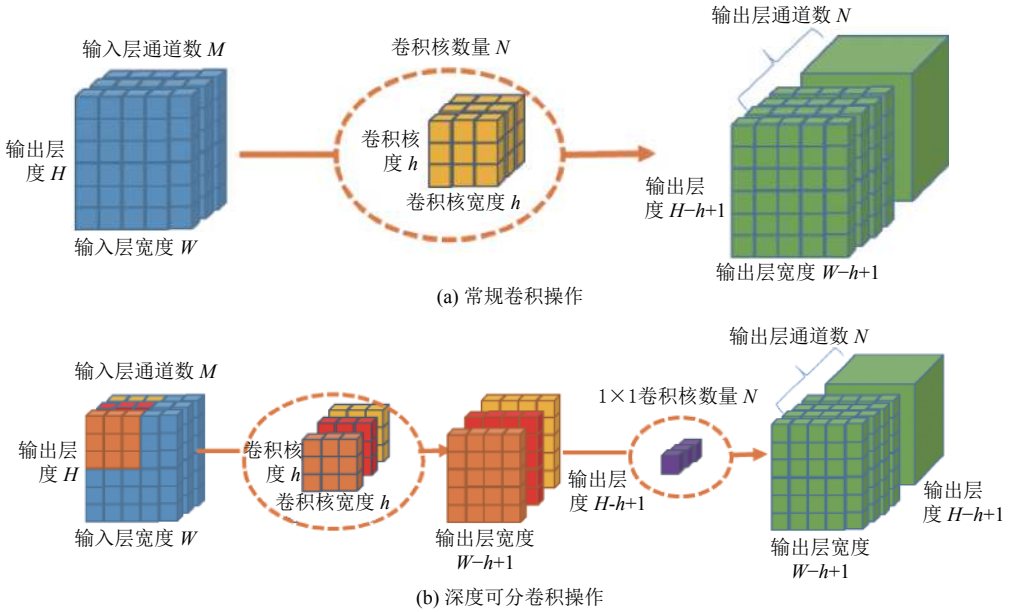


图 2 常规卷积与深度可分卷积对比

通过对比传统卷积和深度可分卷积, 如图 2(a) 所示, 传统卷积输入层为$F \in R^{W\times H\times M}$, 其中 W 为输入层宽度, H 为输入层高度, M 为输入层通道数, 卷积核大小为 h, 经过 N 个卷积核进行处理后, 得到输出层$G \in R^{(W_c\times H_c\times N)}$, 其中输出层宽度为$W_c = (W-h+1)$, 输出层高度为$H_c = (H-h+1)$, 输出层通道数为 N. 由此, 计算常规卷积操作的时间复杂度如式 (1):

$$\mathrm{O}(Conv) = N\times h\times h\times M\times W_c\times H_c \tag{1}$$

深度可分离卷积主要由两部分组成, 深度卷积层与逐点卷积层. 深度卷积层由 M 个卷积核组成, 分别针对输入层进行卷积操作; 逐点卷积层则利用1×1卷积进行逐点卷积, 降低卷积操作的计算量, 其时间复杂度如式 (2):

$$\mathrm{O}(Depth) = N\times h\times h\times 1\times W_c\times H_c + N\times 1\times 1\times M\times W_c\times H_c \tag{2}$$

由式 (2) 与式 (1) 进行相比运算, 得到式 (3). 深度可分离卷积与 2D 卷积之间的乘法运算次数之比为:

$$\frac{\mathrm{O}(Depth)}{\mathrm{O}(Conv)} = \frac{1}{N} + \frac{1}{h^2} \tag{3}$$

由式 (3) 可知, 对于目前大部分网络模型来说, 输出层通常不止 3 个通道, 通常几百甚至几千个通道, 即 $N \gg h^2$. 如果使用的卷积核大小为3×3, 经典卷积操作中的乘法运算要比深度可分离卷积多 9 次, 若使用大小为5×5的卷积核, 则要多运算 25 次.

1.3 基于 SSD 算法进行车辆目标检测与识别

基本的 SSD 模型是通过 VGG 网络模型用来提取特征, 通过将不同的卷积层特征进行融合来实现对不同类型目标的特征进行表达, 提升目标检测的效率.

在 SSD 模型中特征的提取主要采用的是逐层提取和抽象的思想, 较低层级的特征主要针对占比较小的目标, 而高层特征主要对应占比较大的目标[19]. 因此 SSD 模型算法如式 (4)、式 (5) 所示.

$$T_n = S_n(T_{n-1}) = S_n(S_{n-1}(\cdots S_1(I))) \tag{4}$$

$$R = D(d_n(T_n),\cdots,d_{n-k}(T_{n-k})),\ n > k > 0 \tag{5}$$

其中, T_n标识第n层的特征向量, S_n表示由第$n-1$层特征向量经过非线性的预算得到的第n层特征向量, $S_1(I)$则表示对于输入的图像 I, 经过非线性运算后得到的第 1 层的特征向量; $D(\cdot)$表示所有检测的中间结果集合后的最后输出. 由式 (4)、式 (5) 可以看出, 第$n-1$层的特征向量将决定第n层特征向量, 因此如果要更加准确的检测出目标对象就需要获取足够量的特征信息.

由于 SSD 算法可以不用产生候选区域, 直接就生成了物体的类别概率以及定位坐标, 经过单次检测运算即可获得最终检测结果. 本文将 SSD 与 MobileNet 激进型融合, 使用3×3的卷积核进行深度可分卷积操作, 在不增加运算量的前提下保证了模型的准确率. 在实际应用过程中, 由于监控摄像头受到安装点位、日照光线以及地点不同, 使得图像的背景较为复杂, 为了让训练的模型更适用于当前正在使用的监控摄像头, 针对监控摄像头中的特种车辆目标进行改进.

车辆目标在监控画面中属于中型占比目标, 因此本文以 SSD 为主要检测器, 通过将 MobileNet 与 SSD 进行结合, 通过对多个高层特征信息的提取, 通过以新样本为补充从而对网络进行再训练以此提高模型对车辆目标的监测能力. 另外, 由于 MobileNet 网络具有更少层级、更少参数, 可以使训练后的网络模型更加小巧、运算量远小于传统卷积神经网络[20], 具体网络结构如图 3 所示.

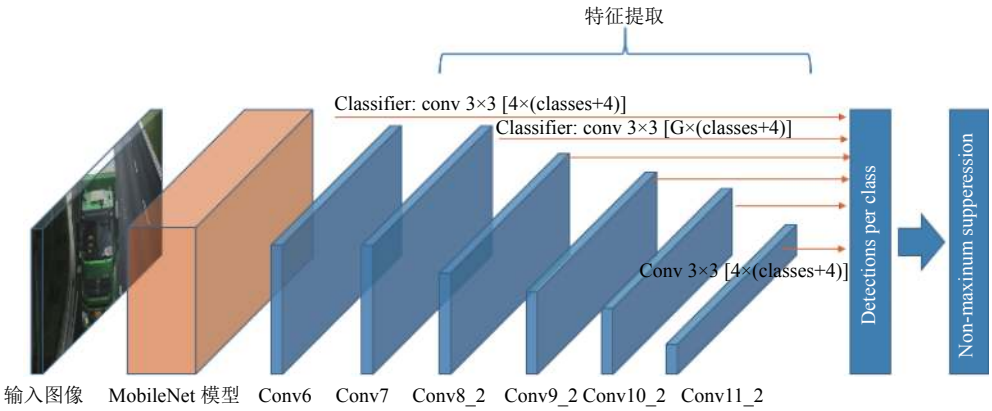


图 3 基于 MobileNet+SSD 的车辆检测网络结构

对于输入的视频流图像首先进入 MobileNet 模型中, 然后通过增加辅助卷积层结构来获取分层特征.

2 基于神经元计算棒的特种车辆检测

2.1 树莓派与神经元计算棒

将上文中介绍的改进模型在完成训练后, 最终在树莓派上进行部署和应用, 如图 4 所示.

本文采用的树莓派版本为 3B+开发版, 该版本树莓派的处理器为 ARMv7 1.2 GHz、内存为 1 GB RAM, 其计算能力一般, 但由于其成本低、适用性强等特性, 应用范围较广. 因此, 为了在计算能力有限的树莓派上运行较为复杂的深度学习算法, 还需要采用 Movidius 神经元计算棒进行加速运算. Movidius 神经元计算棒是 Intel 研发的 VPU 模块, 可以通过 USB 端口进行挂载, 针对移动端和嵌入式设备的计算能力不足的缺点, Movidius 神经元计算棒能够将深度学习网络, 如 Caffemodel 编译为可执行的 Graph 格式, 实现深度学习算法加速, 支持在 Tensorflow[21] 和 Caffe[22] 框架下进行模型的训练和预测.

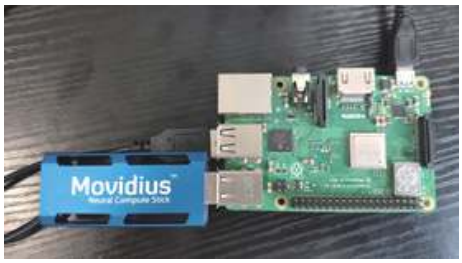


图 4 搭建树莓派与神经元计算棒开发环境

2.2 训练数据集的制作

本文为了提高训练数据的准确性, 首先采用数据集图像质量较高并同时标注完善的 MS COCO2014 作为初始的训练集. MS COCO 数据集是计算机视觉领域中著名的数据集之一, 包含日常生活中常见的 91 个类

别, Truck 就是其中一类. 与 PASCALVOC、ImageNet 等数据集相比, 由于 MSCOCO 数据集中图片背景更加复杂, 目标数量较多, 同时目标尺寸更小, 因此通过 MSCOC 数据集进行预训练, 提取只有 Truck 类别的目标图片, 制作 COCO_Truck 数据集, 最终一共包含 5000 张训练图片和 2000 张验证图片. 同时, 为了保证训练的数据涵盖最终使用的场景, 另外通过卡口的摄像头补充 1500 张训练图片, 和 500 张验证图片. 新补充的数据集如下所示, 其中包括夜间场景、日间场景以及相似车辆的场景 (负样本), 如图 5 所示.

(a) 夜间场景 (b) 日间补充 (c) 相似车辆 (负样本)

图 5 补充样本数据

2.3 开发与训练过程

在开发和训练阶段需要对数据进行预处并在最后训练结束时将算法模型编译成神经元计算棒能够执行的格式, 具体流程图如图 6 所示.

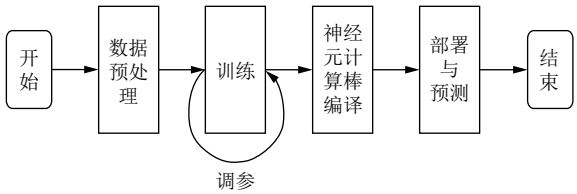


图 6 开发与训练过程流程图

第 1 步. 数据预处理

将 MS COCO 和新添加的数据集制作成训练集和测试集, 再将数据集转化为 LMDB 格式, 便于算法进行读取和使用.

第 2 步. 训练

在数据集制作完成后, 基于 Caffe 框架进行渣土车的算法设计, 利用带有 GPU 运算能力的服务器进行训练, 最终得到 Caffe 格式的检测模型, 即 Caffemodel 权重文件. 在进行算法训练环节中, 通过使用配备显存为 11GBRTX2080Ti 的服务器进行训练, 设置的训练迭代次数为 250 000 次. 同时, 为了提高算法识别的准确率, 采用 multistep 学习率衰减策略, 学习率设置为 0.006.

第 3 步. 神经元计算棒编译

通过训练所得检测模型并不能直接使用 Movidius 神经元计算棒进行计算, 需要将模型编译成其可执行的格式, 即 Graph 格式. 运行 Movidius 神经元计算棒 SDK 的 mvNCCompile 模块将 Caffemodel 权重文件编

译成 Movidius 神经元计算棒可执行的 graph 文件.

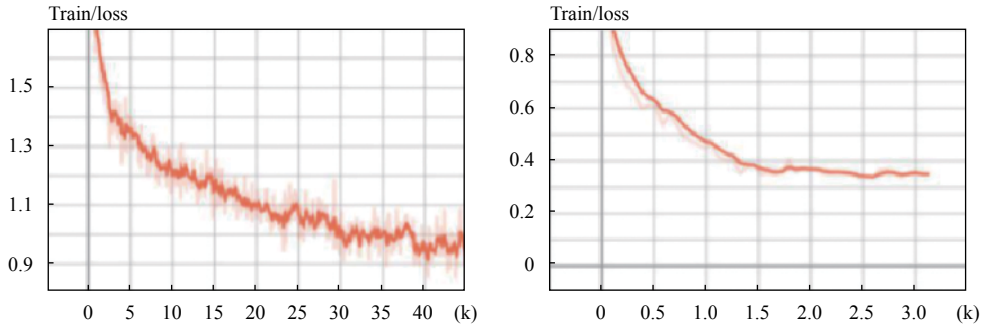
第 4 步. 部署与预测

将检测模型部署在树莓派 3B+开发板中, 将树莓派通过 WIFI 连接到专有网络, 基于 RSTP[23] 协议实现视频监控影像的实时数据回传, 并通过 inference 检测模块, 实现对 graph 模型的运行来对回传影像进行处理, 实现特种车辆的检测与识别.

3 实验结果

本文主要通过改进的 MobileNet+SSD 算法, 并将其运行在计算资源和存储资源有限的树莓派+神经元计算棒平台上, 以此来实现更低成本的渣土车辆的检测与识别. 为了提高算法的适用性, 本文主要基于 MS COCO 数据集进行训练, 同时新增了如渣土车的负样本、夜间场景、部分出现与遮挡等监控图片和标注. 图 7 为训练 MSCOCO 数据集与自定义的人工标注数据集的训练 loss 曲线, 可以看出 MSCOCO 数据集的基础上进行训练, 最后得到稳定的收敛结果.

为了比对本文选取的方法与采用 GPU 的检测识别算法的差异, 本文主要采用的指标为 FPS 与 mAP 进行分析, FPS 表示每秒的识别帧数, mAP 表示平均准确度. 通过使用平均精度均值 (mean Average Precision, mAP)、通过 mAP 指标来衡量检测出的目标中正确的目标所占比率, 测试结果比对如表 1 所示.



(a) 训练 MSCOCO 数据集 (b) 自定义的人工标注数据集

图 7 Loss 曲线

表 1 本文算法与其他算法比对测试结果

算法	FPS	mAP
HOG+SVM (树莓派环境)	0.5	55
Yolo-tiny (树莓派环境)	6	70.8
SSD (GPU环境)	20	87.9
本文算法(树莓派环境)	**4**	**85.8**

在对比实验中, 本文主要从两个方面进行分析: 首先在相同的运算环境下, 采用传统机器学习算法 HOG+SVM、深度学习算法 Yolo-tiny 与本文算法进行比较分析, 通过对比实验可以看出本文算法要优于以上两个算法, 虽然在 FPS 方面要略逊于 Yolo-tiny, 但在实际应用过程中可以忽略本文算法的不足; 此外, 通过将部署在 GPU 环境的 SSD 算法与本文算法将进行比较, 可以看出在 GPU 环境下 FPS 可以达到 20, 而 mAP 与本算法是相近的, 但 GPU 环境下的耗电量以及采购成本都远远超过本文提出的算法和架构, 可见在实际工程应用中, 本文提出的算法和架构更具优势.

同时, 为了验证本文提出的算法和架构在实际使用中的有效性, 在实验中选取白天 (非高峰时段)、夜间以及多车辆场景进行对比验证, 具体如表 2 所示.

表 2 本文算法在不同场景下的对比结果

场景	本文算法mAP	Tiny-yolomAP
夜间	81.2	72.6
白天	92.1	84.8
白天(多车辆)	84.4	81.1

图 8 为基于树莓派+神经元计算棒的运算环境下使用本文提出的 MobileNet+SSD 的算法识别效果, 可以看到在白天、夜间场景和多车辆的场景下以及部分出现与遮挡的场景中, 都取得了不错的效果.

4 结语

本文基于树莓派与 Movidius 神经元计算棒作为计算平台, 通过将 MobileNet 与 SSD 进行结合提出能够在计算资源有限的平台上运行的车辆目标检测与识别算法, 在更低成本和更低能耗的条件下可以实现实时目标检测, 并实际应用中进行验证. 未来通过对车辆

目标检测网络进行持续优化, 实现针对车辆目标的结构化输出, 以及跨摄像头的目标检索.

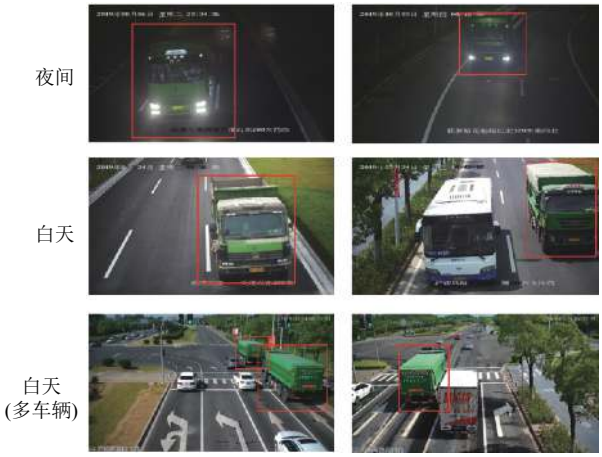


图 8 本文提出的 MobileNet+SSD 算法识别效果

参考文献

1 许洁琼. 基于视频图像处理的车辆检测与跟踪方法研究 [博士学位论文]. 青岛: 中国海洋大学, 2012.

2 Dalal N, Triggs B. Histograms of oriented gradients for human detection. Proceedings of 2018 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, CA, USA. 2005.886–893.

3 Lowe DG. Object recognition from local scale-invariant features. Proceedings of the 7th IEEE International Conference on Computer Vision. Kerkyra, Greece. 1999. 1150–1157.

4 张学工. 关于统计学习理论与支持向量机. 自动化学报, 2000, 26(1): 32–42.

5 Viola P, Jones MJ. Robust real-time face detection. International Journal of Computer Vision, 2004, 57(2): 137–154. [doi: 10.1023/B:VISI.0000013087.49260.fb]

6 Gupta S, Girshick R, Arbeláez P, *et al*. Learning rich features from rgb-d images for object detection and segmentation. Proceedings of the 13th European Conference on Computer Vision – ECCV 2014. Zurich, Switzerland. 2014. 345–360.

7 Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA. 2014.580–587.

8 Girshick R. Fast R-CNN. Proceedings of 2015 IEEE International Conference on Computer Vision. Santiago, Chile. 2015.1440–1448.

9 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]

10 Redmon J, Divvala S, Girshick R, *et al*. You only look once: Unified, real-time object detection. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 779–788.

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

12 李龙棋, 方美发, 唐晓腾. 树莓派平台下的实时监控系统开发. 闽江学院学报, 2014, 35(5): 67–72. [doi: 10.3969/j.issn.1009-7821.2014.05.014]

13 Ionica MH, Gregg D. The movidius myriad architecture's potential for scientific computing. IEEE Micro, 2015, 35(1): 6–14. [doi: 10.1109/MM.2015.4]

14 Vinyals O, Toshev A, Bengio S, *et al*. Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(4): 652–663. [doi: 10.1109/TPAMI.2016.2587640]

15 Szegedy C, Liu W, Jia YQ, *et al*. Going deeper with convolutions. Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA. 2014. 1–9.

16 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv: 1409.1556, 2014.

17 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA. 2015. 770–778.

18 Howard AG, Zhu ML, Chen B, *et al*. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv: 1704.04861, 2017.

19 张洋硕, 苗壮, 王家宝, 等. 基于 Movidius 神经计算棒的行人检测方法. 计算机应用, 2019, 39(8): 2230–2234. [doi: 10.11772/j.issn.1001-9081.2018122595]

20 吴广伟. 基于移动终端的轻量级卷积神经网络研究与实现 [硕士学位论文]. 西安: 西安电子科技大学, 2018.

21 Abadi M, Agarwal A, Barham P, *et al*. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. arXiv: 1603.04467, 2016.

22 Bahrampour S, Ramakrishnan N, Schott L, *et al*. Comparative study of deep learning software frameworks. arXiv: 1511.06435, 2015.

23 Schulzrinne H, Rao A, Lanphier R. Real Time Streaming Protocol (RTSP). RFC 2326, 1998.