基于 Movidius 神经计算棒的物体检测研究

张海清,张 生
(上海理工大学 光电信息与计算机工程学院,上海 200093)

摘 要:物体的实时检测和识别在计算机视觉领域还面临着许多挑战,尤其是在边缘计算设备上部署物体检测模型时,需要大量的内存与算力。Movidius 神经计算棒是用于深度学习推理的即插即用开发套件,能为低功耗嵌入式系统视觉设备提供深度神经网络加速功能。针对低功耗设备上的物体检测领域,提出一种基于 SSD MobileNetV2 神经网络结构的铁钉检测系统。首先,通过数据增强操作获取足够的训练样本,基于 TensorFlow 迁移学习训练铁钉检测模型;然后,结合 OpenVINO 对模型进行优化并生成专用网络,通过神经计算棒对部署在低功耗设备上的专用网络进行加速推理,并使用 Realsence D435 相机获取深度图像的深度值来计算铁钉的距离。实验结果表明,基于 Movidius 神经计算棒能显著提升在树莓派上的物体检测性能,在 UP Squared 平台上能够实现实时的铁钉检测与测距。

关键词:铁钉检测;Movidius;SSD MobileNetV2;OpenVINO;UP Squared
中图分类号:TP391.41 **文献标识码**:A **文章编号**:1673-629X(2020)10-0031-06
doi:10.3969/j.issn.1673-629X.2020.10.006

Research on Object Detection Based on Movidius Neural Computing Stick

ZHANG Hai-qing,ZHANG Sheng
(School of Optical-electrical and Computer Engineering,University of Shanghai for Science and Technology,
Shanghai 200093,China)

Abstract:Object detection and recognition in real time still face many challenges in the field of computer vision,especially in the deployment of object detection model on edge equipment,which requires a lot of memory and computational power. Movidius neural computing stick is a plug-and-play development suite for deep learning reasoning that provides deep neural network acceleration for low-power embedded system vision devices. A nail detection system based on SSD MobileNetV2 neural network is proposed for object detection in low-power devices. First of all,sufficient training samples are obtained through data enhancement operation,and the nails detection model is trained by TensorFlow-based transfer learning. Then combined with OpenVINO,the model is optimized and a special network is generated. The neural computing stick is used to accelerate the inference of the special network deployed on the edge equipment,and the depth value of the depth image obtained by the Realsence D435 camera is used to calculate the distance of the nails. The experiment shows that Movidius - based neural computing stick can significantly improve the object detection performance of Raspberry PI,and real-time nail detection and ranging can be realized on UP Squared platform.

Key words:nail detection;Movidius;SSD MobileNetV2;OpenVINO;UP Squared

0 引 言

随着深度学习与计算机硬件逐渐完善,神经网络彻底改变了机器智能的许多领域,基于深度学习的方法已经被广泛应用于实时目标检测,能够在图像识别领域实现较高的精确度,它们一般建立在卷积神经网络(CNN)的基础上[1]。然而,提高精确度需要大量的计算资源,需要进行大量数据计算,给移动设备和嵌入式系统带来了挑战。云平台能够提供训练深度学习模型的最佳环境,但是推理通常是在服务器,台式机,移动设备和边缘设备中完成。融合人工智能(AI)的应用程序结合硬件和软件来加速在云中训练的深度学习模型的推理。将 AI 部署到边缘设备上非常重要,可以避免数据传输的延迟以及保护用户数据隐私。这些移动端的应用对算法运行效率有较高的要求,在低功耗边缘设备上部署物体检测与识别深度学习网络的相关工作是近几年来积极研究的领域。

目前已提出多种目标检测模型,基于深度学习的物体检测方法可以分为两类:一阶段(one stage)检测

收稿日期:2019-11-04 **修回日期**:2020-03-05
基金项目:上海市科研计划项目(19511105103)
作者简介:张海清(1995-),男,硕士研究生,CCF 会员(4946M),研究方向为深度学习、目标检测;张 生,副教授,研究方向为计算机应用。

和两阶段(two stage)检测[2]。两阶段检测算法有基于区域建议(region proposal, RP)方法的 R-CNN 模型[3]、Fast R-CNN 模型[4]、Faster R-CNN 模型[5]、FPN 模型[6]。一阶段检测算法有:基于回归方法的 SSD 模型[7]以及 YOLO 模型[8]。SSD 模型是当中检测精确度相对较高的网络结构,传统的 SSD 算法采用 VGG16[9]卷积作为基础网络进行特征提取,前向传播计算时间大部分用在基本网络中。考虑到低功耗嵌入式平台运算能力有限,需要轻量化的卷积神经网络,以实现轻量化模型。针对不同领域和产品(如机器人,无人机,机器视觉,智能家居),Up squared 能够提供完美硬件解决方案,在性能良好的 Up squared 设备上能够实现基于深度学习的目标检测任务。

针对工业物体检测领域,该文提出了一种基于 SSD MobileNetV2 算法的铁钉检测方法,通过英特尔 Movidius 神经计算棒加速模型的实时推理,采用英特尔 Realsence D435 对检测的铁钉进行测距,实现在计算资源相对紧张的嵌入式边缘计算设备树莓派上进行铁钉检测,并在 Up squared 上进行实时的铁钉检测与测距。

1 SSD MobileNetV2

1.1 SSD 算法研究

单发多框目标检测(single shot multibox detector, SSD)是一个单级检测器,在保持实时效率的同时,其精度可与两级检测器媲美。相对于需要区域建议的两阶段检测算法更加简单,完全消除了区域建议和后续的像素或特征重采样阶段,并将所有计算封装在一个网络中。SSD 算法使用的是 Faster-RCNN 中的 anchor 机制(文献[8]中称之为 default box),与 Faster-RCNN 不同点为 SSD 算法每层的 feature maps 的感受野不同,能够获得不同的特征向量。SSD 算法在主干网络的特征图上,使用卷积滤波器预测锚框的分类置信度和目标边界框的位置偏移。SSD 算法使用不同比例的候选框,以多种尺度的特征图进行预测,它将 YOLO 算法的全连接层删除,将全连接层换成全卷积层,显著提升了检测速度[10]。

SSD 算法的主要流程如下:

· 使输入的图像经过一系列卷积层,以不同比例(例如 10×10、6×6、3×3 等)生成几个特征图。

· 对于特征图中的每个位置,使用 3×3 卷积滤波器输出一组默认边界框[11]。

· 对于每个边界框,同时预测边界框位置偏移和分类置信度。

· 在训练期间,将 ground_truth 框与这些基于 IOU 的预测框进行匹配。预测最准确的框将被标记为“正”。

SSD 算法基于前馈卷积网络会生成固定大小的边界框集合,并为这些框中存在的物体进行打分,然后采用非最大抑制算法来筛选最终检测的边界框[12]。

1.2 非极大值抑制 NMS

基于深度学习的目标检测算法在目标数据集训练好模型后,通过遍历的方式将图像输入到训练好的网络模型进行分类。首先,需要对输入的图像进行预处理,将输入调整为网络要求的尺寸大小并进行归一化操作。通过推理并返回位置信息和类别信息,对于预测的边框需要通过 NMS 算法选出有效检测结果,并在原图中画框。非极大值抑制本质为搜索局部极大值,抑制非极大值元素。NMS 算法的输入为:候选框 B,相应的置信度得分 S 和重叠阈值列表 N。输出为:检测框列表 D。NMS 算法过程如下:

(1)将置信度得分排序,选取置信度最高的候选框,将其从 B 中删除,并将其添加到最终输出列表 D 中,D 初始化为空。

(2)将该候选框与所有的候选框进行比较,计算该候选框与其他所有候选框的 IOU(intersection over union)。如果 IOU 大于阈值 N,则从 B 中删除该候选框。

(3)从 B 中的其余候选框中选择置信度最高的,将其从 B 中删除并添加到 D 中。

(4)再使用 B 中的所有候选框计算该候选框的 IOU,并删除 IOU 高于阈值的框。重复此过程,直到 B 中没有剩余的候选框。

其中,IOU 表示两个候选框的交集与并集部分的面积比值。它主要是衡量模型生成的锚框(bounding box)和标注的真实框(ground truth box)之间的重叠程度,计算公式为:

$$\mathrm{IOU}(P,\mathrm{Gt}) = \frac{P \cap \mathrm{Gt}}{P \cup \mathrm{Gt}} \tag{1}$$

IOU 越高,预测框的位置越准确,IOU 阈值一般设为 0.3 ~ 0.5。

SSD 算法的损失函数由置信度损失(Softmax)与定位损失(Smooth L1)两部分组成,公式如下:

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

其中,x 为 X_{ij}^{p},表示第 i 个默认框(deafault box)和第 j 个真实框(ground truth box)两框面积之间的交并比,c 为 deafault box 的置信度大小,参数 l 为 default box,g 为 ground truth box。L_{loc} 是回归损失,表示预测框 l 和真实框 g 参数之间的 Smooth L1 损失。L_{conf} 是置信度损失,表示多个类别置信度 c 上的 softmax 损失。

1.3 MobileNetV2 网络

针对一些移动、嵌入式边缘设备,谷歌提出了 Mo-

bileNetV2 这种轻量级的深度卷积神经网络,其设计的宗旨是使移动设备支持图像分类与检测等。MobileNetV2[13]基于 MobileNetV1[14]的思想,使用深度可分离卷积作为有效的构建块。但是,MobileNetV2 向体系结构引入了两个新功能:(1)层之间的线性瓶颈;(2)瓶颈之间的快捷连接。MobileNetV2 体系结构如图 1 所示。

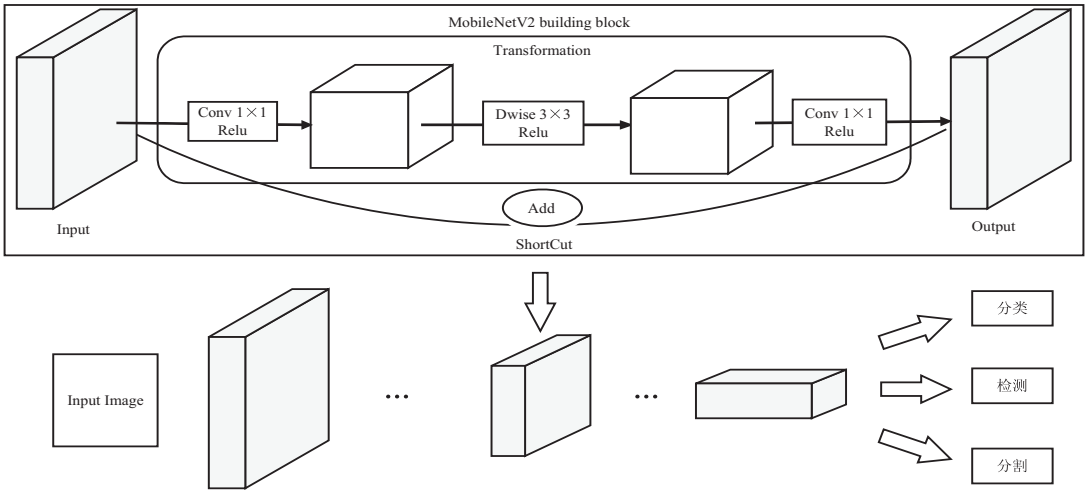

图 1　MobileNetV2 体系结构

MobileNetV2 体系结构基于反向残差结构,其中残差块的输入和输出是 Bottleneck 层,MobileNetV2 对网络输入进行扩展升维,深度分离卷积,再压缩降维,与传统残差模型相反[15],传统残差模型在输入中使用扩展表示形式。MobileNetV2 使用轻量级深度卷积来过滤中间扩展层中的特征。线性瓶颈(linear bottleneck)中包含了所有的必要信息,用 shortcuts 连接 linear bottleneck,可以提升梯度在乘积层之间的传播能力,提高内存的使用效率。图 1 中 Transformation 模块是 bottleneck 卷积的基本实现:先用 conv 1×1 变换通道,再用 ReLU6 激活。中间是深度卷积,后接 Relu;最后的 conv 1×1 之后不再使用 Relu,而是使用 linear bottleneck。

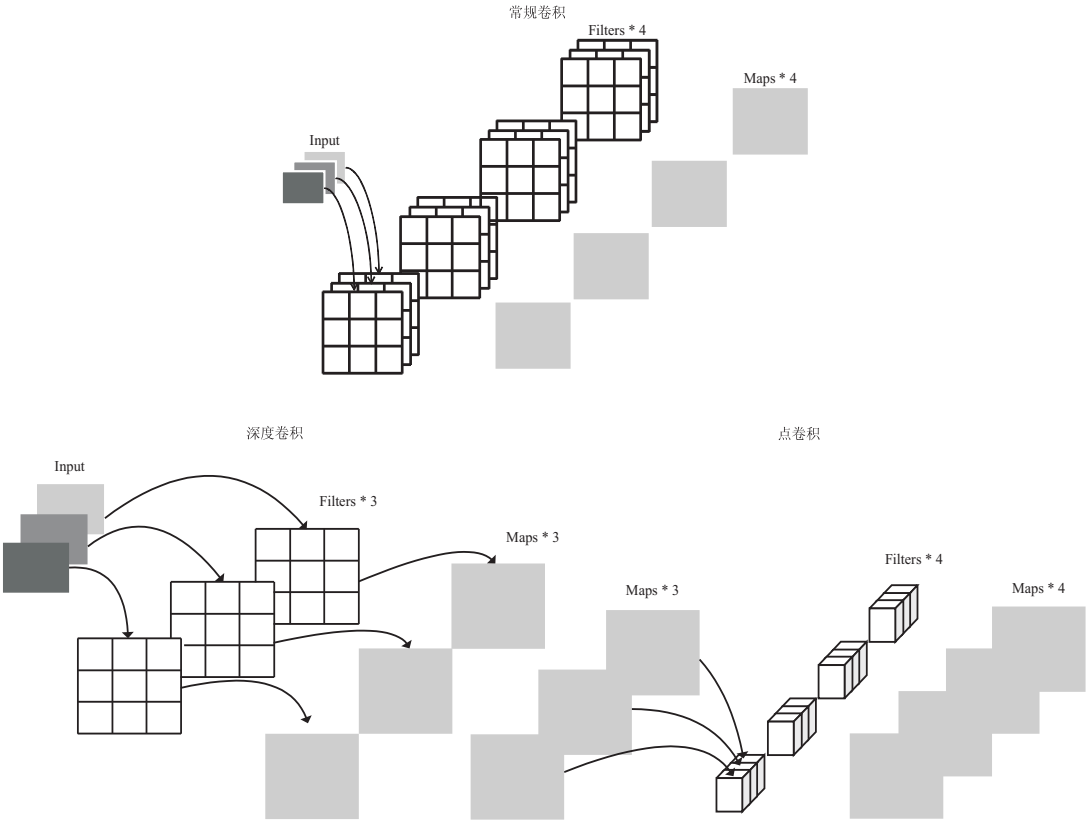

图 2　常规卷积与深度可分离卷积

MobileNetV2 共包含 28 个网络层,在该网络的卷积层中,第一层采用的常规卷积(Conv2d)运算,其他的卷积层中,均把常规的卷积运算拆分成了 depthwise 卷积(Conv dw)和 pointwise 卷积(Conv pw)过程,这两步运算被合称为深度可分离卷积(depthwise separable convolution)。深度卷积用来对每个输入通道应用单通道的轻量级滤波器,逐点卷积负责计算输入通道的线性组合构建新的特征。常规卷积与深度可分离卷积的过程的对比如图 2 所示。

假设常规卷积使用 4 个大小为 3 * 3 的卷积核,每个卷积核将对包含 3 个通道的输入进行卷积运算得到特征图,进行一次卷积需要 4 * 3 * 3 * 3 = 108 个要学习的参数。MobilenetV2 将传统的 convolution 换成 depthwise 卷积和 pointwise 卷积,进行深度可分离卷积时,depthwise 卷积的过程是使用 3 个二维的卷积核分别与输入的 3 个通道进行卷积,得到的 3 张特征图并与 1×1×3 大小的卷积核进行 pointwise 卷积输出特征图,但卷积过程中的参数却只有 3×3×3+1×1×3×4 = 39 个。其中 depthwise 卷积使用 3 * 3 大小卷积核进行 3 次卷积运算,pointwise 卷积使用 1 * 1 * 3 大小卷积核进行 1 次卷积,总学习参数为 3 * 3 * 3+1 * 1 * 3 = 39。深度可分离卷积通过减少了卷积过程中的学习参数来降低计算复杂度和模型的大小。

MobileNetV2 通过增大 depthwise 卷积的步幅来实现对输入特征的下采样,除了最后的全连接层外,其他各个网络层的输出都先进行了一次批标准化,再使用 Relu 函数进行激活,实现加快模型的收敛速度。所有空间卷积核尺寸使用 3×3 卷积核大小,在 32 个卷积核的全卷积层之后接上 17 个反向残差瓶颈模块,并采用 Relu6 作为非线性激活函数,确保低精度计算的鲁棒性。

2 基于 Movidius 的开发应用流程

2.1 NCSDK 开发流程

英特尔 Movidius 神经计算棒(NCS)是一款深度学习加速计算设备,旨在为低功耗移动和嵌入式视觉等边缘设备上加速 AI 推理,例如树莓派或 Up Squared board。目前已推出 NCS1 和 NCS2 两个系列。1 代神经计算棒 NCS 基于英特尔 Myriad 2 VPU(视觉处理单元),2 代神经计算棒 NCS2 基于 Myriad X VPU,性能显著优于 1 代。1 代需要依赖 NCSDK 环境中进行编译、部署,以实现加速网络计算,NCSDK 包括一组用于编译,分析和验证深度神经网络的软件工具,还包括用于以 C/C++或 Python 开发应用程序的 Intel Movidius NCAPI。基于 NCSDK 的开发流程如图 3 所示。

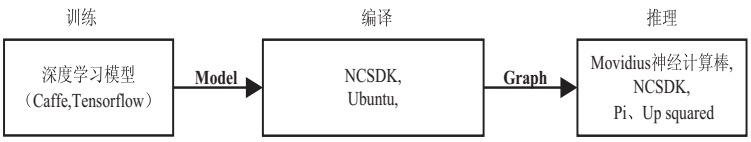

图 3 NCSDK 应用开发流程

2.2 OpenVINO 开发流程

2 代神经计算棒在 NCSDK 上不支持,而是使用 OpenVINO 代替 NCSDK。OpenVINO 是英特尔发布的视觉推理和神经网络优化软件开发套件(SDK),旨在为英特尔视觉产品(支持 AI 的处理器和加速器的产品组合)之间扩展工作负载并优化性能。OpenVINO 详细的开发应用流程如图 4 所示。

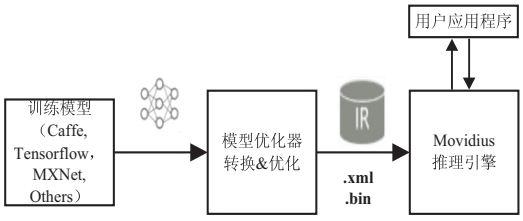

图 4 OpenVINO 应用开发流程

3 实 验

3.1 硬件设备

实验主要硬件设备如图 5 所示,实验中分别采用了树莓派 3B+和 UP Squared 开发板,UP Squared 采用英特尔赛扬™,奔腾™和凌动™处理器,是工业物联网边缘设备的理想选择。搭载 Intel Movidius™ Myriad™ 2 VPU 的 UP Squared 只需很少的功耗即可实现本地深度学习和计算视觉算法。UP Squared board 大小与树莓派相同,可以运行一般的 Windows、Linux 或 Android 系统。

图 5 实验主要硬件设备

3.2　数据集预处理

深度学习模型需要大量的训练样本,由于很难具有足够采样的数据集,可通过数据增强方法扩充数据集,数据增强策略在提高模型的精度与泛化方面非常重要。为此,对所有训练图像进行随机采样,通过对输入数据进行变换,自动生成新的训练样本。例如,移位、随机裁剪、旋转、垂直和水平翻转图像。实验中需要检测的物体只有一类,原始采集的图像有一百张,通过数据增强扩充数据集,总共 200 张,其中,90% 用于训练,10% 用于测试。

标注图像使用的是 labelImg 工具,每标注一张图片后,会产生一个. xml 文件,由于检测物体只有一类(钉子,nail),标签映射 pbtxt 中只需一个 item,id 设置为 1,name 设置为 nail。之后将标注图像生成的所有. xml文件转换为. csv 格式,再将. csv 文件转换为. Record文件,用于 TensorFlow 物体检测 API 进行训练。实验过程中使用 Tensorboard 查看模型训练指标,例如损失和精确度。

3.3　训练模型

随着计算机视觉在无人驾驶汽车、人脸识别、智能驾驶系统等领域的用例日益增长,用户希望能够建立定制的机器学习模型以实现物体的检测与识别。但是,从头开始构建训练模型需要大量的专业知识,时间和计算资源。为减少深度学习入门障碍,Google 发布了 TensorFlow 物体检测 API 和 TensorFlow Hub 平台,使人们能够通过迁移学习来快速构建自定义模型。

该文使用 Google 推出的 TensorFlow 深度学习框架来训练模型,TensorFlow 物体检测 API 能够轻松构建、训练和部署物体检测模型。通过物体检测 API 结合给定的图像数据集即可训练自定的目标检测模型。目前在庞大的数据集上已有许多可用的预训练模型,通过迁移学习在 SSD MobileNetV2 模型上进行微调,只需数小时即可完成模型的训练,进行推理即可获得较好的结果。SSD MobileNetV2 模型在 COCO 数据集上经过预训练,数据集包含 32.8 万张带有 250 万个标注的实例图片,共 91 类物体。通过迁移学习,基于 Google 预训练的 SSD MobileNetV2 模型进行微调,采用随机梯度下降法(stochastic gradient descent,SGD)进行训练,动量因子为 0.9,衰减因子为 0.000 5。初始学习率为 0.005,并使用指数衰减学习率衰减策略。在显存为 8 GB NVIDIA RTX2070 的主机上训练,整个训练过程为 200 000 次迭代。

3.4　转换与部署模型

基于 TensorFlow 训练导出的模型包含 ckpt 格式和固化的. pb 格式文件,对于 1 代的神经计算棒 NCS,需要通过安装 NCSDK 将训练好的模型编译为 graph 格式。如果是 2 代神经计算棒 NCS2,则训练结束后要将导出的模型通过 OpenVINO 工具包中的模型优化器转化为中间表示 IR 格式,它由. xml 和. bin 两个文件组成。xml 文件中保存了模型网络结构信息,bin 文件中保存了模型的权重。之后结合神经计算棒将转换后的专用网络部署在边缘设备上进行物体检测。

4　实验结果与分析

表 1 和表 2 为实验中基于 Movidius 神经计算棒的铁钉检测结果。本实验中,基于树莓派的铁钉检测,USB 摄像头的性能稍好于 Pi 摄像头。测试分辨率为 640×480,将 Pi Camera 的分辨率降低到 320×240,可以得到约 5 FPS 的图像,这表明可以通过降低输入图像的分辨率的大小以提高帧速率。与单个线程处理当前帧再等待处理下一帧相比,通过线程优化处理的检测效果有所提升。检测性能还会受帧中检测到的物体数量的影响,当有多个物体检测时,NCS 需要对输出进行更多的反序列化和图像处理。与树莓派 CPU 相比,基于 Movidius 神经计算棒的检测推理速度获得明显提升。

表 1　基于 Movidius 的树莓派检测性能

平台	FPS
Pi 3B+,CPU,Pi Camera	0.9
Pi 3B+,CPU,USB camera	1.0
NCS,Pi 3B+,Pi Camera	4.2

表 2　基于 OpenVINO 和 Movidius 的检测性能

平台	FPS
Pi 3B+ CPU	0.9
NCS 2,OpenVINO,Pi 3B+	8.3
NCS 2,OpenVINO,Up squared	23.4

在搭载 Movidius 的 UP Squared 设备上测试(见图 6),实时检测帧率约 23.4 FPS,通过 Realsence D435 相机获取深度图像的深度值来计算被检测铁钉的距离,能够实时地检测铁钉并进行测距。

5　结束语

实验在低功耗设备上进行基于 SSD MobileNetV2 的铁钉检测,比较了 1 代神经计算棒和 2 代神经计算棒在低功耗设备上的铁钉检测性能,并利用 Realsence D435 相机获取深度图像的深度值来计算铁钉的距离,在 UP Squared 平台上对检测的铁钉进行实时测距。实验结果表明,基于 Movidius 神经计算棒能够显著提升树莓派的物体检测性能。通过 OpenVINO 优化基于 SSD MobileNetV2 网络模型结构,结合英特尔 2 代神经计算棒与 Reasence D435 相机,能够在 Up Squared

上进行实时铁钉检测与测距。

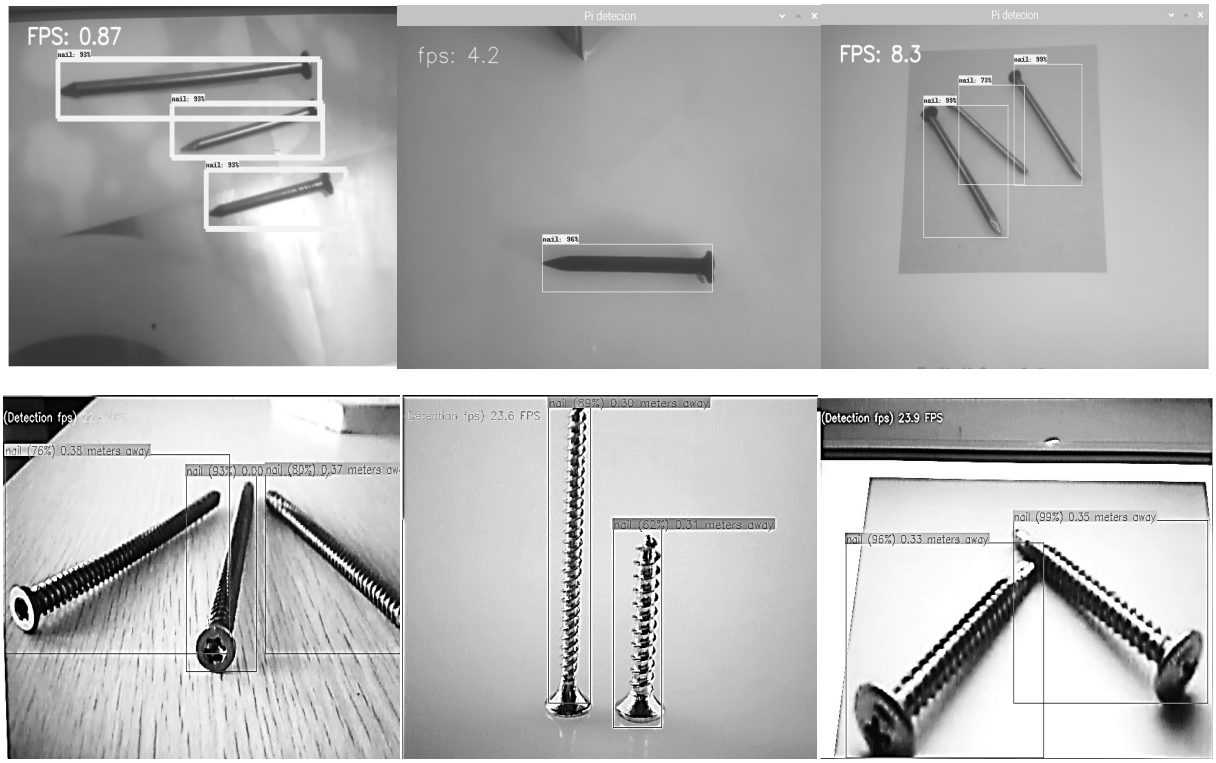

图 6 基于 Movidius 神经计算棒的树莓派与 Up Squared 实时铁钉检测

参考文献：

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]// Proceedings of the 25th international conference on neural information processing systems. Lake Tahoe, Nevada: Curran Associates Inc., 2012: 1097-1105.

[2] 唐 聪,凌永顺,杨 华,等. 基于深度学习物体检测的视觉跟踪方法[J]. 红外与激光工程,2018,47(5):0526001-1-0526001-11.

[3] HUANG J, GUADARRAMA S, MURPHY K, et al. Speed/accuracy trade-offs for modern convolutional object detectors[C]//IEEE international conference on computer vision. [s. l.]: IEEE, 2016: 3296-3297.

[4] GIRSHICK R. Fast R-CNN[C]//IEEE international conference on computer vision. [s. l.]: IEEE, 2015: 1440-1448.

[5] REN Shaoqing, HE Kaiming, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[6] LIN T Y, DOLLAR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]//IEEE conference on computer vision and pattern recognition. Italy: IEEE, 2017: 936-944.

[7] LIU W, ANGUELOV D, ERHAN D, et al. SSD: single shot multibox detector[C]//European conference on computer vision. Cham: Springer, 2016: 21-37.

[8] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]//Computer vision and pattern recognition. [s. l.]: IEEE, 2016: 779-788.

[9] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[C]//International conference on learning representations. San Diego, CA: Computational and Biological Learning Society, 2015: 1150-1210.

[10] 寇大磊,权冀川,张仲伟. 基于深度学习的目标检测框架进展研究[J]. 计算机工程与应用,2019,55(11):25-34.

[11] XU J. 斯坦福目标检测深度学习指南[J]. 机器人产业,2017(6):18-24.

[12] NEUBECK A, GOOL L J V. Efficient non-maximum suppression[C]//18th international conference on pattern recognition (ICPR2006). Hong Kong: IEEE, 2006: 850-855.

[13] SANDLER M, HOWARD A, ZHU M, et al. MobileNetV2: inverted residuals and linear bottlenecks[C]//IEEE conference on computer vision and pattern recognition. Salt Lake City: IEEE, 2018: 4510-4520.

[14] HOWARD A G, ZHU M, CHEN B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications[C]//IEEE conference on computer vision and pattern recognition. Hawaii: IEEE, 2017: 1-9.

[15] HAN D, KIM J, KIM J. Deep pyramidal residual networks[C]//Proceedings of the 2017 IEEE conference on computer vision and pattern recognition. Piscataway: IEEE, 2017: 6307-6315.