

基于 T-YOLO-LITE 树干检测的模型部署方法

高宗斌 崔永杰* 李凯

(西北农林科技大学机械与电子工程学院 陕西 杨凌 712100)

(农业农村部农业物联网重点实验室 陕西 杨凌 712100)

摘要 为解决猕猴桃采摘机器人视觉导航问题,提出基于 T-YOLO-LITE 的猕猴桃树干检测方法。通过保留 BN 层、调整输入图像尺寸、修改 anchor boxes、添加负样本对 YOLO-LITE 进行改进,并利用 Movidius 与 OpenCV-DNN 将改进后的模型分别部署在树莓派与 CPU 设备中。实验结果表明:通过改进优化使模型检测精度提升至 59.75%;利用 Movidius 与 OpenCV-DNN 部署后,模型检测速度分别达到了 2 帧每秒和 6 帧每秒。该模型在检测精度与 YOLOV2-TINY 持平的情况下,检测速度为 YOLOV2-TINY 的两倍,并在非 GPU 设备上完成近实时的树干检测任务。

关键词 YOLO 目标检测 模型部署 卷积神经网络

中图分类号 TP391 **文献标志码** A **DOI**:10.3969/j.issn.1000-386x.2021.02.023

MODEL DEPLOYMENT METHOD BASED ON T-YOLO-LITE TRUNK DETECTION

Gao Zongbin Cui Yongjie* Li Kai

(College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, Shaanxi, China)

(Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling 712100, Shaanxi, China)

Abstract In order to solve the problem of visual navigation of kiwifruit picking robot, a method for detecting kiwifruit trunk based on T-YOLO-LITE is proposed. YOLO-LITE was improved by preserving the BN layer, adjusting the input image size, modifying the anchor boxes, adding negative samples, and Movidius and Opencv-DNN were used to deploy the improved models in the Raspberry Pi and CPU devices. The experimental results show that the model detection accuracy was improved to 59.75% through improved optimization. After deployment with Movidius and Opencv-DNN, the model detection speed is 2 frames and 6 frames per second. The model has twice the detection speed of YOLOV2-TINY when the detection accuracy is the same as that of YOLOV2-TINY, and it completes the near real-time trunk detection task on non-GPU devices.

Keywords YOLO Object detection Model deployment Convolutional neural network

0 引言

中国是猕猴桃栽培面积最大的国家,猕猴桃采摘机器人^[1]可以有效改善劳动力短缺的现状,而自动导航是实现其自动化作业的关键技术之一。目前国内外普遍采用 GPS 进行导航,但是在猕猴桃果园独特的棚架式栽培模式下,由于棚架顶部用钢丝固定枝蔓,受钢丝和树冠层遮挡,导致无法精确接收 GPS 信号^[2],故

选择视觉方法检测猕猴桃树干实现移动设备的导航与定位。深度学习技术的迅猛发展使得目标检测算法的性能^[3]远优于传统图像处理算法,所以利用深度学习目标检测算法对猕猴桃树干进行检测有重要意义。

目前,许多学者利用传统图像处理方法尝试对果园特定目标进行检测或对图像进行分割。新西兰猕猴桃果园树干与支撑物苔藓附着严重,使得利用颜色特征将树干与背景分离显得尤为困难,Scarfe^[4]利用树干边缘特征,采用 Sobel 算子边缘检测和模板匹配方法

对树干以及支撑物进行检测,但是只有在有利光照条件下取得较好的效果。柑橘果园环境下,文献[5-6]利用颜色特征,分别采用 BP 神经网络分类以及聚类的方法将天空、树冠、土地进行图像分割,并取得较高的分割精度,但是柑橘果园场景相对单一,此方法并不适用于背景复杂的猕猴桃果园。中国新疆矮化密植枣园环境复杂,树干边缘以及颜色特征均不显著,彭顺正等[7]选用 B 分量图对其进行处理,提出“行阈值分割”方法分割树干,并引入了趋势线,但算法相对复杂,且准确率不高。由此可见,在果园较为复杂的环境中,常规的图像处理算法未取得较好的效果。

近年来,借助卷积神经网络进行图像分类的精度超过了人工分类[8],将其作为特征提取层的目标检测算法在检测精度和速度上均取得了巨大的提升,如 R-CNN 系列[9-11]、SSD[12]、YOLO[13-15]在许多不同的领域取得了较好的表现[16-17]。猕猴桃果园树干特征较为明显,部分学者对树干的检测进行了尝试:Zhang 等[18]针对苹果收获问题,利用 Kinect 深度图像与 R-CNN 目标检测算法检测苹果树干,但是因为环境的复杂性以及苹果树干本身的特征等因素导致检测精度较低;Shah 等[19]利用无人机搭载深度学习网络对树干进行检测和定位,同时标定相应的 GPS 信息,进行数据库的建立,但由于猕猴桃树干的特征与大多数树干的特征差异较大,无法利用现有的数据集进行检测。

深度学习在数据集愈发庞大、网络结构愈发复杂的发展趋势下,精度不断提高,但计算量也随之增大,需要高性能 GPU 支持,在某些场合下无法投入实际应用。一些学者开始将轻量化模型部署在功耗较低的嵌入式设备[20-22]上,以减少计算量并提高实时性。本文对 YOLO-LITE[23] 模型进行优化,并利用 Movidius 与 OpenCV-DNN 两种方式将其部署在树莓派 3B+ 与 CPU 设备中,为算法真正投入使用打下基础,对实现猕猴桃采摘机器人视觉导航有重要意义。

1 YOLO 目标检测

1.1 图像采集与数据集构建

2018 年 9 月于陕西省眉县猕猴桃试验站进行图像采集。利用大疆御 Mavic Air 无人机拍摄视频。视频编码格式为 mp4,码率为 5 217 kbit/s,帧率为 30 帧每秒、分辨率为 1 280 × 720 像素,将视频进行分帧处理得到 JPG 格式图像。

本文从分帧处理所得到的 6 844 幅图像中随机取 5 128 幅图像作为训练集,剩余 1 716 幅图像作为测试

集。将猕猴桃果园树干和支撑钢管归为一类,统一标注为“tree”,标注示例如图 1 所示。



图 1 Labeling 标注示例

1.2 YOLO 目标检测原理

YOLO 算法的核心思想是将目标检测作为回归问题解决。将输入图像划分成 $S \times S$ 个栅格 (grid cell),若一个目标的中心点落在某个栅格里,那么这个栅格负责预测该目标。每个栅格需要预测边界框 (bounding box) 的四个值: x, y, w, h 。坐标 (x, y) 表示预测边界框中心与栅格边界的相对值;坐标 (w, h) 表示预测边界框的长与宽相对于整幅图像长与宽的比例。在预测边界框坐标的同时,会为每个边界框预测一个置信度 (Confidence scores),置信度则反映了此栅格中是否含有目标以及边界框与 Ground truth 的接近程度,其定义如下:

$$C_s = P_r(Object) \times IOU_{pred}^{truth} \quad (1)$$

如果栅格中不存在目标,则 $P_r(Object)$ 为 0,置信度为 0,所以此栅格中不对目标进行预测;反之,栅格中存在目标,则 $P_r(Object)$ 为 1,置信度为边界框与 Ground truth 的 IOU_{pred}^{truth} 。

置信度反映栅格中是否存在目标,而不反映属于哪一类目标,所以每个栅格还需要预测条件类别概率。本文只有一类目标,所以设置条件类别概率的数量为 1。最终预测时每个栅格的条件类别概率与边界框的置信度共同反映其预测精度,定义如下:

$$P_r(class_i | Object) \times P_r(Object) \times IOU_{pred}^{truth} = P_r(Class_i) \times IOU_{pred}^{truth} \quad (2)$$

条件类别概率 $P_r(class_i | Object)$ 与置信度 $P_r(Object) \times IOU_{pred}^{truth}$ 相乘,当栅格中包含目标时 $P_r(Object)$ 为 1,最终得到的 $P_r(Class_i) \times IOU_{pred}^{truth}$ 中 $P_r(Class_i)$ 反映了目标属于某个类别的概率, IOU_{pred}^{truth} 反映了所预测边界框的准确度。

YOLO 目标检测算法经过三个版本的更替,其性

能逐渐提高。YOLOV3 在 YOLOV2 基础上进行改进,在不降低检测速度的情况下,尽可能提升检测精度,但轻量级模型 YOLOV3-TINY 相比 YOLOV2-TINY 在检测速度方面下降幅度较大,所以在某些对精度要求不高,但需要保证检测速度的应用场合,选择在 YOLOV2-TINY 基础上进行优化的 YOLO-LITE 模型进行改进以及部署。

2 YOLO-LITE 改进与模型评价

为与 YOLO-LITE 原模型进行区分,将改进后的 YOLO-LITE 统称为 T-YOLO-LITE (TREE-YOLO-LITE)。本文保留 YOLO-LITE 提出的特征提取层,依据所要检测的类别数量,修改最后一个 1×1 卷积层。最终用于预测的向量深度为 $B \times (5 + C)$, B 为每个栅格内预测边界框的数量,5 个 anchor boxes 分别预测 5 个边界框, B 为 5; C 为所要检测的类别数量, C 为 1, 最终的向量深度为 $5 \times (5 + 1) = 30$ 。所以表 1 中 Conv7 的 Filters 为 30。

表 1 T-YOLO-LITE 特征提取层网络结构

Layer	Filters	Size	Stride
Conv1	16	3×3	1
Maxpooling1	—	2×2	2
Conv2	32	3×3	1
Maxpooling2	—	2×2	2
Conv3	64	3×3	1
Maxpooling3	—	2×2	2
Conv4	128	3×3	1
Maxpooling4	—	2×2	2
Conv5	128	3×3	1
Maxpooling5	—	2×2	2
Conv6	256	3×3	1
Conv7	30	1×1	1

2.1 模型改进方法

本文通过保留 BN (Batch Normalization) 层、调整输入图像尺寸、修改 anchor boxes、添加负样本对 YOLO-LITE 进行优化,使之满足猕猴桃树干检测的需求。

(1) 保留 BN 层。YOLO-LITE 模型证明了在某些应用场合下,利用浅层网络进行预测时,BN 层是非必需的,而且剔除 BN 层可以减少计算量,提高实时性。但由于本研究中猕猴桃树干目标较小,并引入大量背

景信息,较难检测。去除 BN 层后训练出现梯度消失的问题,模型无法收敛,故本文保留了 YOLO-LITE 剔除的 BN 层。

(2) 调整输入图像尺寸。YOLO-LITE 模型为了提高实时性,将输入图像大小调整为 224×224 像素,但是输入图像过小导致检测精度降低,很难满足实际应用中的需求。本文将输入图像调整至 448×448 像素,保证模型部署后满足实际需求。

(3) 修改 anchor boxes。设置 anchor boxes 的目的是让边界框与 Ground truth 的 IOU 更大,以便提取目标中更多有用信息,而非背景,因此需要选择合适的 anchor boxes 使模型更加精准。采用 k-means 聚类方法对训练集 Ground truth 做聚类分析,选取 Ground truth 的 box 与聚类中心 box 之间的 IOU 作为距离指标:

$$d(box, centroid) = 1 - IOU(box, centroid) \quad (3)$$

最终聚类得到的五个 anchor boxes 为: (0.30, 0.86)、(0.40, 1.15)、(0.53, 1.57)、(0.70, 2.22)、(1.03, 2.96)。YOLO-LITE 原模型中提出的 anchor boxes 为: (1.08, 1.19)、(3.42, 4.41)、(6.63, 11.38)、(9.42, 5.11)、(16.62, 10.52), 原模型的 anchor boxes 尺度范围变化较大,适用于通用目标检测,而猕猴桃树干数据集目标整体较小,聚类后的 anchor boxes 尺度范围变化小,更易提取到目标。

(4) 添加负样本。本研究对误检测率要求较为严格,预测模型不能出现将行间杂草、地面以及树冠等检测为树干的情况,因此本文通过随机裁剪和人工分类的方式在训练样本中添加了 5 800 幅不同尺度的负样本,避免误检测。几类负样本示例如图 2 所示。

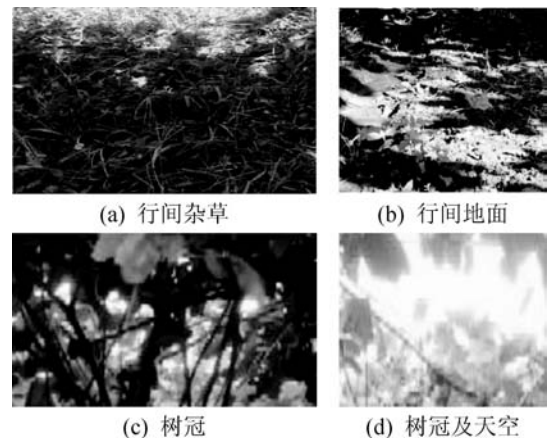


图 2 负样本示例

2.2 模型精度评价指标

为了验证猕猴桃树干检测精度,引入准确率 P (precision)、召回率 R (recall)、AP (average precision)、F1 值对训练后的模型进行评价。评价指标计算公式

如下:

$$P = TP/(TP + FP) \quad (4)$$

$$R = TP/(TP + FN) \quad (5)$$

$$F1 = 2PR/(P + R) \quad (6)$$

$$AP = \int_0^1 P(R) dR \quad (7)$$

式中: TP (True Positive) 为预测成功的数量,即正确检测到的树干数量; FP (False Positive) 为背景预测为树干的量,即误检为树干的量; FN (False Negative) 为将树干预测为背景的数量,即未检测到的树干数量。

检测时, TP 为边界框与 Ground truth IOU 大于 0.5 的目标数量, FP 为边界框与 Ground truth IOU 小于等于 0.5 的目标数量; FN 为没有检测到的 Ground truth 的数量。

3 模型部署

本文选择 Ubuntu 16.04 darknet 框架进行模型训练。硬件配置: 处理器 Intel Xeon E5-1650 v4, 主频 3.6 GHz, 内存 32 GB, 显卡 Nvidia Titan XP, 显示存储 12 GB。将标注完毕的数据集送入改进后的 T-YOLO-LITE 网络结构进行训练, 保存网络结构 T-YOLO-LITE. cfg 以及权重 T-YOLO-LITE. weights 文件。

3.1 Movidius

Movidius 是一款低功耗深度学习推理工具和独立的神经网络加速器。功耗 1 W, 浮点性能为 100 GFLOPs, 为广泛的边缘主机或嵌入式设备提供专用的神经网络推理功能。Movidius 通过 USB 接口连接至 PC 或树莓派等设备, 不能用于训练模型, 但是可以部署训练完毕的神经网络模型, 通过 Movidius 进行加速计算与推理。本文将 Movidius 通过 USB 连接至树莓派 3B+, 如图 3 所示。

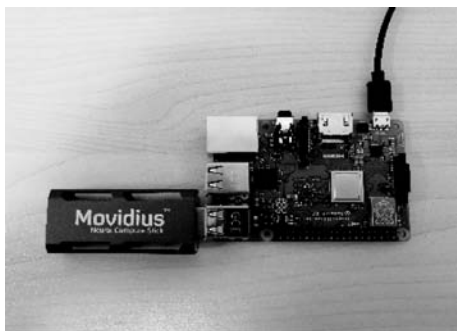


图 3 树莓派 3B+ 与 Movidius

Movidius 支持 Caffe 以及 Tensorflow 模型的转换与推理, 本文利用 darknet 框架训练模型, 所以要将训练

完毕的模型转换为 Caffe 或 Tensorflow 模型进行推理。利用: YoloV2NCS (<https://github.com/duanguan/YoloV2NCS>) 将 T-YOLO-LITE. cfg 转换为 T-YOLO-LITE. prototxt, 保存为 Caffe 格式的网络结构; 将 T-YOLO-LITE. weights 转换为 T-YOLO-LITE. caffemodel, 保存为 Caffe 格式的训练权重。

Movidius 运行 Caffe 模型前, 需要借助 NCSDK 中的 mvNCCompile 工具, 将 Caffe 模型 T-YOLO-LITE. prototxt 与 T-YOLO-LITE. caffemodel 转化为 Movidius 推理所需要的 graph 模型。最后利用在树莓派编译完毕的 NCSDK 中 inference 模块调用转换后的 graph 模型, 完成 T-YOLO-LITE 在树莓派设备中的部署。利用 graph 推断得到目标位置 (x, y, w, h) 、类别、置信度。模型转换流程如图 4 所示。

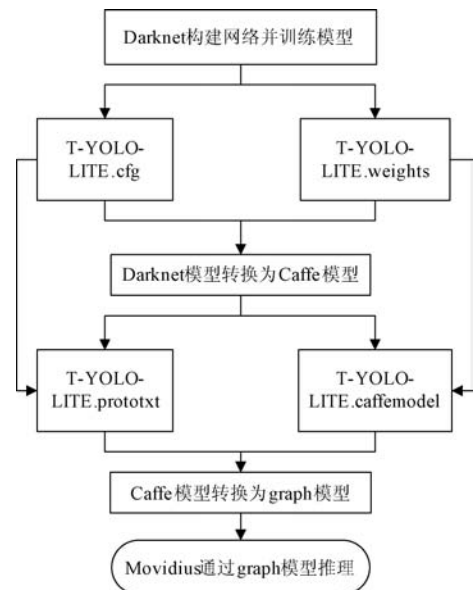


图 4 模型转换流程

Darknet 模型训练依赖高性能 GPU, 功耗较大。经过两次模型转换, 将其部署在树莓派 3B+ 与 Movidius 后, 只需要 5 V 的电源为树莓派供电即可, 满足实际应用中低功耗的需求。

3.2 OpenCV-DNN

OpenCV 是机器视觉领域运用较为广泛的视觉库, 随着深度学习迅速发展, OpenCV 3.1 版本 contrib 中加入了神经网络 DNN 模块, OpenCV 3.3 版本将 DNN 模块加入正式代码的主仓库。DNN 模块包含神经网络 Convolution、FullyConnected、MaxPooling、ReLU、BatchNormalization 等许多基本构成元素; 包含 Alexnet、GoogLeNet、VGG-16 等经典卷积神经网络以及 SSD、YOLO 等经典目标检测算法。DNN 模块只支持模型推理与加速, 不支持模型训练。

目前深度学习模型大多依赖 Caffe、Tensorflow、Py-torch 等框架,并需要 CUDA、CUDNN 等用于调用 GPU 加速训练。在某些实际应用中无须训练模型,硬件性能只需要保证神经网络的前向传播过程,进行推理即可。OpenCV-DNN 模块不依赖第三方库,也无须考虑深度学习框架以及算法的配置和编译,即不需要重新编译 Darknet,依靠 DNN 模块调用训练好的模型即可进行推理。由于不需要训练以及反向传播过程,针对此特点进行优化,避免不必要的计算,DNN 模块中 YOLO 模型在 CPU 上的推理速度要明显优于原版 Darknet。

OpenCV-DNN 调用 Darknet 模型包括以下几步:

1) Darknet 导出模型。模型经训练后,保存 T-YOLO-LITE. cfg 和 T-YOLO-LITE. weights 文件。

2) 调整初始化参数。调整网络的输入图像大小,设置为 448 像素 \times 448 像素。调整置信度阈值大小为 0.5。调整非极大值抑制阈值为 0.01。将非极大值阈值调整至一个较低的水平,用于判断网络的输出结果的误检测率,即让尽可能多的 boxes 输出,从而判断误检测率是否满足需求。

3) 读取网络结构。利用函数 cv.dnn.readNetFromDarknet(),读取两个传入参数:T-YOLO-LITE. cfg 和 T-YOLO-LITE. weights,加载训练完毕的网络。

4) 读取输入视频或图像。每帧图像通过 bolbFromImage() 函数将其处理为神经网络的输入类型 bolb,将图像像素归一化为 0 至 1 之间。

5) 网络输出。网络输出的每一个边界框都包含长度为 5 的向量,分别为 x, y, w, h 、置信度。利用坐标将边界框逐帧画在输入图像中。

本文在 CPU:i5-7300 HQ,内存 8 GB 的 PC 上配置 OpenCV 3.4.1 进行 T-YOLO-LITE 的推理加速计算。

4 实验

T-YOLO-LITE 网络训练超参数设定:batch = 64, subdivisions = 8, momentum = 0.9, decay = 0.0005, angle = 0, saturation = 1.5, exposure = 1.5, hue = 0.1, learning_rate = 0.001, burn_in = 1000, max_batches = 500000, policy = steps, steps = 400000, scales = 0.1。采用分步调整学习率的策略,初始学习率为 0.001,当迭代次数超过 400000 次时,学习率降低 10 倍,调整为 0.0001。

损失函数如图 5 所示。共迭代 500000 次,平均

loss 稳定在 0.13 附近,模型收敛较好。

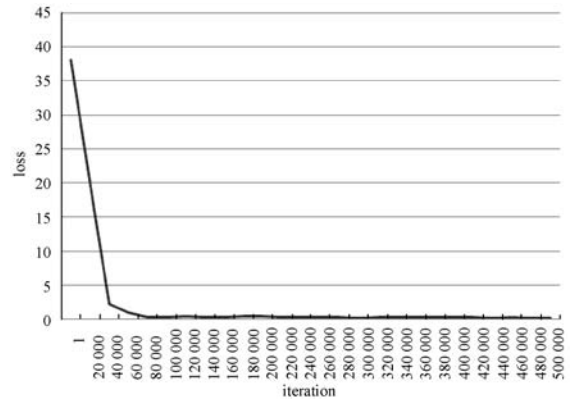


图 5 损失函数 loss 可视化

4.1 模型改进实验

利用第 2.1 节中提出的 4 个改进方法对 YOLO-LITE 进行优化,分别对应表 2 中改进(1)至改进(4),并针对每一个改进方法在测试集对检测精度进行测试。本文通过 AP 以及 F1 值两个评价标准对树干的检测精度进行评价。通过 FPS 对检测速度进行评价,在 Nvidia Titan XP 中进行计算。

表 2 YOLO-LITE 改进实验结果

	YOLO-LITE				T-YOLO-LITE
改进(1)		√	√	√	√
改进(2)			√	√	√
改进(3)				√	√
改进(4)					√
AP	0%	15.94%	42.65%	57.56%	59.75%
F1	0	0.36	0.58	0.61	0.63
FPS	240	230	200	200	200

注:表中改进(1)至改进(4)对应 2.1 节中四个改进方法;“√”表示采用此改进方法。

YOLO-LITE 原模型为了提高实时性,剔除了 BN 层,但在检测猕猴桃树干时,模型不收敛,无法检测到目标。如表 2 所示:YOLO-LITE 原模型 AP 为 0%,通过改进(1)保留 BN 层后,AP 提升至 15.94%,开始检测到目标;通过改进(2)增大输入图像尺寸,AP 提升至 42.65%,但经过两次改进导致检测速度降低,FPS 由 240 降低至 200;通过改进(3)修改 anchor boxes 以及改进(4)添加负样本,使 AP 进一步提高至 59.75%。F1 值与 AP 所反映的趋势一致。通过 4 个改进方法得到最终的 T-YOLO-LITE 模型。

图 6 为 4 个改进方法对应的检测结果,改进(1)和改进(2)中白色椭圆标记的目标是明显的误检测结果,精度无法满足需求;改进(3)和改进(4)精度差距

较小,检测结果无明显差异。

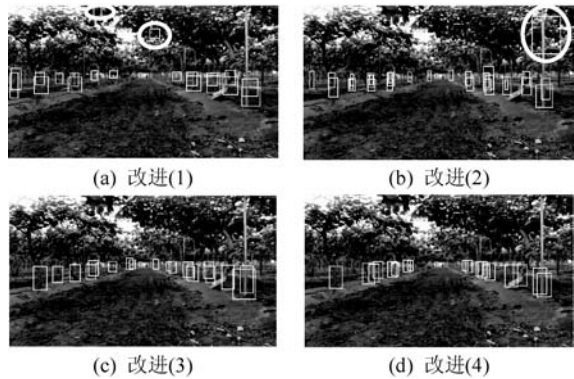


图 6 不同改进方法检测结果

4.2 模型性能对比

将 T-YOLO-LITE 模型与 YOLOV2 以及 YOLOV2-TINY 模型进行比较,结果如表 3 所示。特征提取层最深的 YOLOV2 模型取得了最高的检测精度,其 AP 为 69.54%;YOLOV2-TINY 与 T-YOLO-LITE 模型在检测精度上基本持平,AP 分别为 59.99%与 59.75%。检测速度上 YOLOV2-TINY 和 T-YOLO-LITE 的 FPS 分别为 YOLOV2 的 3.25 倍与 5 倍,由此可知轻量级模型在检测速度方面的优势明显强于深层模型。在模型大小方面,深层结构的 YOLOV2 最大,为 192 MB,浅层结构 YOLOV2-TINY 为 42 MB,而 T-YOLO-LITE 模型大小仅为 2 MB,远小于上述两种模型,但是 T-YOLO-LITE 却取得了与 YOLOV2-TINY 几乎一致的精度,并且在检测速度方面明显优于上述两种模型。

表 3 三种模型性能对比

模型名称	AP	F1	FPS	模型大小
YOLOV2	69.54%	0.73	40	192 MB
YOLOV2-TINY	59.99%	0.66	130	42 MB
T-YOLO-LITE	59.75%	0.63	200	2 MB

PR 曲线下方面积为该目标检测模型的 AP,三种模型的 PR 曲线如图 7 所示。可以看出,YOLOV2 的曲线完全高于 YOLOV2-TINY 与 T-YOLO-LITE,其面积最大,精度最高。YOLOV2-TINY 与 T-YOLO-LITE 曲线相互交错,面积基本一致,精度持平。虽然 YOLOV2 在精度上有着明显的优势,但从如图 8 所示的检测结果中可以看出:YOLOV2-TINY 与 T-YOLO-LITE 检测到的目标均在地垄附近,未出现将树冠、地面等检测为树干的情况,其检测精度已经可以满足猕猴桃树干检测的需求。YOLOV2-TINY 与 T-YOLO-LITE 两种轻量级模型检测到的目标与 Ground truth 重叠部分较少,IOU 较低,从而被判定为检测错误。大部分目标检测算法评价标准中,均将 IOU 小于 0.5 的目标判定为误

检测,而本研究中误检测定义为将树冠、地面等检测为树干,但与标准一致,依然保留此评价方法。

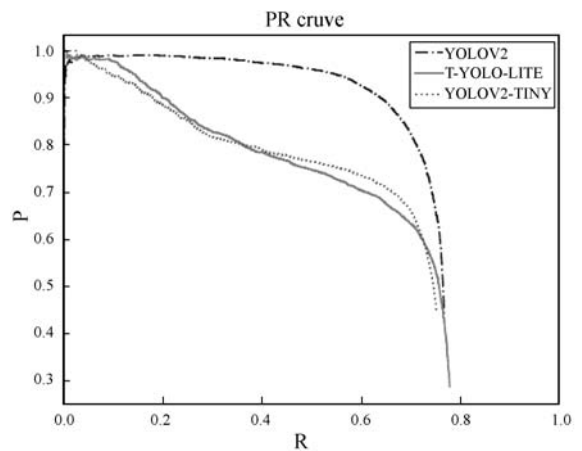


图 7 三种模型 PR 曲线

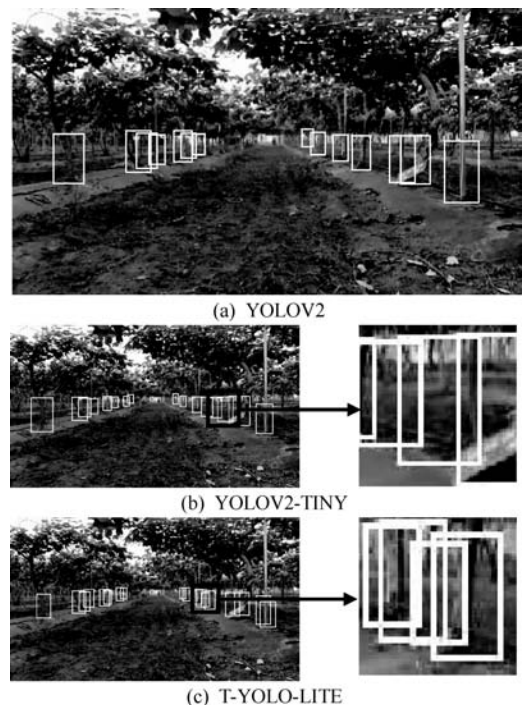


图 8 三种模型检测结果

从图 8(b)与图 8(c)可以看出,在猕猴桃树干检测任务当中,YOLOV2-TINY 与 T-YOLO-LITE 出现的 IOU 较低的目标位置依然处于地垄周围或树干之间,不影响拟合导航路径时的精度。虽然 T-YOLO-LITE 模型在 AP、F1 值等评价标准中处于较低的水平,但是完全符合实际应用需求,并且在检测速度以及模型大小两方面拥有显著优势。

4.3 模型部署后检测速度验证

利用第 3 节中的两种方法将模型分别部署在树莓派 3B+ 与 i5-7300HQ 上。通过表 4 所示的三种模型部署后的 FPS 对比可知:不利用高性能 GPU,三种模型的检测速度都出现很大程度的下降,其中深层模型 YOLOV2 表现最不理想。在无法利用高性能 GPU 的

场合,深层模型无法真正落地应用。相比之下,经过优化的 T-YOLO-LITE 在部署后的检测速度远远优于其他两种模型, T-YOLO-LITE 在 OpenCV-DNN 与 Movidius 中的推理 FPS 均达到了 YOLOV2-TINY 的 2 倍。

表 4 三种模型部署后 FPS 对比

模型名称	FPS	
	OpenCV-DNN	Movidius
YOLOV2	0.5	0.2
YOLOV2-TINY	3	1
T-YOLO-LITE	6	2

图 9 为模型部署后的检测结果, OpenCV-DNN 的检测结果要优于 Movidius, Movidius 经过两次模型转换后,边界框与 Ground truth 的接近程度下降。由于 Darknet 在 Maxpooling 时默认进行 padding 操作,转换后 Caffe 模型最终输出的特征图大小减少 1,造成精度损失。后续研究中,将 Darknet 模型转换为 Caffe 模型后,重新将模型进行训练,提升精度,最后再转换为 graph 进行推理。

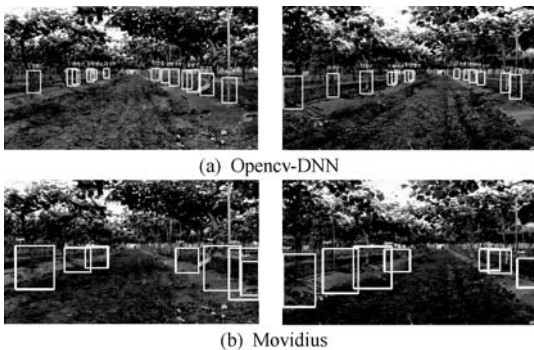


图 9 模型部署后检测结果

5 结 语

本文通过改进优化轻量级网络 YOLO-LITE,在检测精度和速度上做平衡,使其在精度满足猕猴桃树干的检测要求下,尽量提升检测速度。最终通过 Movidius 与 OpenCV-DNN 两种部署方式,将训练完毕的 T-YOLO-LITE 模型分别部署在树莓派 3B+ 与 CPU 中,达到了每秒 2 帧和每秒 6 帧的检测速度。本文为深度学习真正投入应用与后续研究中猕猴桃采摘机器人视觉路径的拟合打下了基础。

参 考 文 献

[1] Williams H A M, Jone M H, Nejadi M, et al. Robotic kiwifruit harvesting using machine vision, convolutional neural net-

works, and robotic arms [J]. Biosystems Engineering, 2019, 181: 140 - 156.

[2] Bell J, MacDonald B A, Ahn H S. Row following in pergola structured orchards [C] // 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016: 640 - 645.

[3] 张慧, 王坤峰, 王飞跃. 深度学习在目标视觉检测中的应用进展与展望 [J]. 自动化学报, 2017, 43 (8): 1289 - 1305.

[4] Scarfe A J. Development of an autonomous kiwifruit harvester [D]. New Zealand: Massey University, 2012.

[5] Joaquin T S, Patricio N. A new approach to visual-based sensory system for navigation into orange groves [J]. Sensors, 2011, 11 (4): 4086 - 4103.

[6] Sharifi M, Chen X Q. A novel vision based row guidance approach for navigation of agricultural mobile robots in orchards [C] // 2015 6th International Conference on Automation, Robotics and Applications. IEEE, 2015: 251 - 255.

[7] 彭顺正, 坎杂, 李景彬. 矮化密植枣园收获作业视觉导航路径提取 [J]. 农业工程学报, 2017, 33 (9): 45 - 52.

[8] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition [C] // 2016 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2016: 770 - 778.

[9] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C] // 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2014: 580 - 587.

[10] Girshick R. Fast R-CNN [C] // 2015 IEEE International Conference on Computer Vision. IEEE, 2015: 1440 - 1448.

[11] Ren S Q, He K M, 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.

[12] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector [C] // 2016 14th European Conference on Computer Vision (ECCV). Springer, 2016: 21 - 37.

[13] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection [C] // 2016 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2016: 779 - 788.

[14] Redmon J, Farhadi A. Yolo9000: Better, faster, stronger [C] // 2017 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 6517 - 6525.

[15] Redmon J, Farhadi A. Yolo3: An incremental improvement [EB]. arXiv:1804.02767v1, 2018.

[16] 朱敏超, 冯涛, 张钰. 基于 fd-ssd 的遥感图像多目标检测方法 [J]. 计算机应用与软件, 2019, 36 (1): 232 - 238.

- [17] 陈江昀. 一种基于深度学习的新型小目标检测方法[J]. 计算机应用与软件, 2017, 34(10): 227-231, 247.
- [18] Zhang J, He L, Karkee M, et al. Branch detection with apple trees trained in fruiting wall architecture using stereo vision and regions-convolutional neural network (R-CNN) [C]//ASABE Annual International Meeting. Spokane: American Society of Agricultural and Biological Engineers, 2017: 2-11.
- [19] Shah U, Khawad R, Krishna K M. Detecting, localizing, and recognizing trees with a monocular MAV: Towards preventing deforestation [C]//2017 IEEE International Conference on Robotics and Automation. IEEE, 2017: 1982-1987.
- [20] 陆维娜, 胡瑜, 叶靖, 等. 面向卷积神经网络加速器吞吐量优化的 FPGA 自动化设计方法[J]. 计算机辅助设计与图形学学报, 2018, 30(11): 2164-2173.
- [21] 张洋硕, 苗壮, 王家宝, 等. 基于 Movius 神经计算棒的行人检测方法[J]. 计算机应用, 2019, 39(8): 2230-2234.
- [22] 张雲轲, 刘丹. 基于小型 Zynq SoC 硬件加速的改进 TINY YOLO 实时车辆检测算法实现[J]. 计算机应用, 2019, 39(1): 192-198.
- [23] Rachel H, Jonathan P, Cuixian C. Yolo-LITE: A real-time object detection algorithm optimized for non-gpu computers [EB]. arXiv:1811.05588v1, 2018.

(上接第 113 页)

负责转发流程和四层负载均衡, 服务器负责七层负载均衡, 将原有 F5 负载均衡的能力分布到了交换机与服务器集群中。同时, 还充分利用加密卡、25 Gbit 智能网卡等硬件设备的加速特性以提升整体集群的性能。

参 考 文 献

- [1] 李伟. 银行业金融机构实施分布式架构的思考[J]. 金融电子化, 2016(12): 19-21.
- [2] 瞿秋平. 分布式架构在证券业信息化建设中的规划思考[J]. 金融电子化, 2016(12): 17-18.
- [3] 王和. 分布式, 未来的必然选项——中国人保财险的思考与实践[J]. 金融电子化, 2016(12): 14-16.
- [4] 柴洪峰. 中国银联专用云的探索与实践[EB/OL]. 第七届中国云计算大会演讲实录. (2015-06-03). <https://www.csdn.net/article/2015-06-03/2824858>.
- [5] McKeown N. Software-defined networking [C]//International Conference on Computer Communications, 2009.
- [6] Bosshart P, Daly D, Gibb G, et al. P4: Programming protocol-independent packet processors [J]. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 87-95.
- [7] Lee J, Miao R, Kim C, et al. Stateful layer-4 load balancing in switching ASICs [C]//SIGCOMM Posters and Demos,

2017: 133-135.

- [8] 徐敏. 从阿里金融云的实践看金融上云的意义和路径[J]. 清华金融评论, 2016(11): 104-106.
- [9] 陈磊. 5.0V—基于 SDN 的腾讯次世代数据中心网络架构[EB/OL]. (2018-01-30). <https://cloud.tencent.com/developer/article/1032288>.
- [10] 闫浩. SDN 网络与传统网络对比分析[EB/OL]. (2017-03-16). <https://www.sdnlab.com/18725.html>.

(上接第 131 页)

- [11] Musci M, Feitosa R Q, Costa G A O P, et al. Assessment of binary coding techniques for texture characterization in remote sensing imagery [J]. IEEE Geoscience and Remote Sensing Letters, 2013, 10(6): 1607-1611.
- [12] Cai D, He X, Han J. Speed up kernel discriminant analysis [J]. The VLDB Journal—The International Journal on Very Large Data Bases, 2011, 20(1): 21-33.
- [13] Brown M, Süssstrunk S. Multi-spectral SIFT for scene category recognition [C]//Computer Vision and Pattern Recognition (CVPR), 2011: 177-184.
- [14] Sande K V D, Gevers T, Snoek C. Evaluating color descriptors for object and scene recognition [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(9): 1582-1596.
- [15] Yang Y, Newsam S. Bag-of-visual-words and spatial extensions for land-use classification [C]//18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2010: 270-279.
- [16] Zhao L J, Tang P, Huo L Z. Land-use scene classification using a concentric circle-structured multiscale bag-of-visual-words model [J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2014, 7(12): 4620-4631.
- [17] Zhao L, Tang P, Huo L. A 2-D wavelet decomposition-based bag-of-visual-words model for land-use scene classification [J]. International Journal of Remote Sensing, 2014, 35(6): 2296-2310.
- [18] Chen S, Tian Y L. Pyramid of spatial relations for scene-level land use classification [J]. IEEE Transactions on Geoscience and Remote Sensing, 2015, 53(4): 1947-1957.
- [19] Cheryadat A M. Unsupervised feature learning for aerial scene classification [J]. IEEE Transactions on Geoscience and Remote Sensing, 2014, 52(1): 439-451.
- [20] Shao W, Yang W, Xia G S, et al. A hierarchical scheme of multiple feature fusion for high-resolution satellite scene categorization [C]//9th International Conference on Computer Vision Systems, 2013: 324-333.