

基于 TensorRT 的植物叶片病害实时检测分类模型优化^①



徐泽华¹, 李坚孝¹, 邓树源¹, 吴家隐², 高嘉晖¹, 潘明毅¹

¹(五邑大学 智能制造学部, 江门 529020)

²(广东邮电职业技术学院 计算机学院, 广州 510630)

通信作者: 吴家隐, E-mail: jiayinwu@foxmail.com

摘要: 为了提高边缘计算设备对植物叶片病害检测的识别速率, 本研究采用卷积神经网络搭建了植物叶片目标识别模型和植物叶片病害分类模型, 并且使用 OpenCV 将两个模型整合成植物叶片病害检测系统. 通过 SSD (single shot multibox detector) 算法对植物叶片的目标区域进行定位并裁剪, 再利用植物叶片病害分类模型对裁剪的植物叶片区域进行病害分类. 同时, 通过 TensorRT 加速推理对分类模型进行优化处理, 以及在同一台主机设备和 Jetson Nano 计算平台上, 对优化前后的模型进行了对比实验. 实验表明, 在同一主机设备上优化后的植物分类模型识别速率提升 22 倍. 同时, 优化后的分类模型使植物叶片病害检测系统识别速率提升 7 倍. 而将优化后的系统部署在 Jetson Nano 计算平台上, 对比优化前的植物叶片病害检测速率提升 10 倍, 实现了实时的植物叶片病害检测.

关键词: 卷积神经网络 (CNN); 植物叶片检测; OpenCV; TensorRT

引用格式: 徐泽华, 李坚孝, 邓树源, 吴家隐, 高嘉晖, 潘明毅. 基于 TensorRT 的植物叶片病害实时检测分类模型优化. 计算机系统应用, 2023, 32(2): 94-101. <http://www.c-s-a.org.cn/1003-3254/8977.html>

Optimization of Real-time Detection and Classification Model for Plant Leaf Diseases Based on TensorRT

XU Ze-Hua¹, LI Jian-Xiao¹, DENG Shu-Yuan¹, WU Jia-Yin², GAO Jia-Hui¹, PAN Ming-Yi¹

¹(Faculty of Intelligent Manufacturing, Wuyi University, Jiangmen 529020, China)

²(School of Computer, Guangdong Vocational College of Post and Telecom, Guangzhou 510630, China)

Abstract: In order to improve the recognition rate of plant leaf disease detection by edge computing devices, this study uses a convolutional neural network to build a plant leaf target recognition model and a plant leaf disease classification model and adopts OpenCV to integrate the two models into a plant leaf disease detection system. The target areas of plant leaves are positioned and clipped by the single shot multibox detector (SSD) algorithm, and then the plant leaf disease classification model is used to classify the clipped plant leaf areas according to diseases. At the same time, the classification model is optimized by TensorRT accelerated inference. In addition, on the same host device and Jetson Nano computing platform, a comparative experiment is carried out on the model before and after optimization. The experiment shows that the recognition rate of the optimized plant classification model on the same host device increases by 22 times. At the same time, the optimized classification model improves the recognition rate of the plant leaf disease detection system by seven times. Furthermore, the optimized system is deployed on the Jetson Nano computing platform, and the detection rate of plant leaf diseases is increased by 10 times compared with that before optimization, which thus realizes real-time plant leaf disease detection.

Key words: convolutional neural network (CNN); plant leaf detection; OpenCV; TensorRT

① 基金项目: 广东邮电职业技术学院校级质量工程项目 (202201); 广东省职业技术教育学会第四届理事会科研规划项目 (202103G93); 2022 年度广东省普通高校特色创新类项目 (2022KTSCX288)

收稿时间: 2022-06-20; 修改时间: 2022-09-07; 采用时间: 2022-09-19; csa 在线出版时间: 2022-11-18

CNKI 网络首发时间: 2022-11-21

目前,我国农业技术发展存在不平衡不充分的现象,农作物工作者主要通过肉眼目视病害叶片的方式,来观察农作物病害情况。然而,由于植物病害种类繁多,农作物工作者存在相关知识储备和农业病害防治能力不足的问题,在缺乏相关技术指导的情况下容易导致对植物病害的误诊。因此,及时识别农作物病害类型是防治农作物病害、减少损失的关键。近年来伴随着物联网技术、人工智能技术的蓬勃发展,人工智能技术也被运用到植物病害检测中。例如,史红栩等^[1]提出了一种香蕉远程病害诊断的解决方案,针对香蕉作物常见的7种病害类型。通过利用迁移学习对GoogLeNet进行训练来获取新的模型,并在此基础上开发了服务器和APP,进一步降低深度学习在植物病害检测领域的使用门槛。乔虹^[2]设计了一种基于深度学习的葡萄叶片病害动态监测系统,该系统由动态监测和疾病诊断两个部分组成。通过Faster R-CNN算法对监控视频进行逐帧处理,借助卡尔曼滤波算法对葡萄叶片进行动态跟踪,从而提取葡萄叶片的目标图像。再通过对目标图像进行全局和局部特征显著性检测,实现了对葡萄叶片病害类型诊断。

Fang等^[3]指出的植物病害检测和鉴别的方法,可以通过直接和间接方法来实现。病害的直接检测包括了分子和血清学方法,但是需要选取大量样本用于高通量分析。在这些方法中,直接检测引起疾病的病原体,如细菌、真菌和病毒,用于提供准确的疾病/病原体的鉴别。常用的检测做法有PCR检测技术、酶联免疫吸附实验技术等。另一方面,间接方法是通过植株的形态变化、温度变化、蒸腾速率变化和挥发性有机化合物的释放量等参数来识别病害,常用的检测方法有高光谱检测技术、荧光成像技术,然后通过计算机图像分析辅助工作人员完成病害的检测。Cséfalvay等^[4]提出了一种基于叶绿素荧光(Chl-F)成像技术的检测方法,探讨该方法是否可用于揭示葡萄霜霉病感染的早期阶段,从而改进杀虫剂的治疗效果。Oppenheim等^[5]通过PlantVillage项目公开提供的14种作物的26种疾病状况的54 306张图像,借助深度学习方法研究探讨了卷积神经网络模型在植物病害检测中的可行性。同时对比分析了AlexNet^[6]和GoogLeNet^[7]的准确度和优劣性,探讨了将深度学习技术迁移至移动端设备上的可能性和便捷性。Gajjar等^[8]提出了一种基于嵌入式设备的植物叶片病害实时检测系统,通过借助英伟达的

Jetson TX1深度学习设备进行部署,从而做到了文献[5]中提出将深度学习迁移至移动端设备的想法。

综上所述,通过深度学习,从输入数据中提取和识别特征,可以进一步实现植物叶片病害的有效检测和识别。然而,由于实际检测的环境复杂,且深度学习对硬件性能要求较高,对现场图像进行实时检测的研究并不多,大多数局限于实验室环境下,以上的研究仍存在不足。本研究基于OpenCV实现了植物叶片病害检测系统的设计,并将系统部署到Jetson Nano计算平台,利用高清摄像头持续捕获病害叶片的视频流,再实时对视频流逐帧进行植物叶片病害检测,预测结果直接显示在屏幕。使用SSD叶片目标检测模型与植物叶片病害分类模型相结合的识别方式,大大提高了实时植物叶片病害检测的精确度。并且对训练完成的ResNet深度网络进行TensorRT优化,提高了植物叶片病害识别的速率,使得嵌入式系统边缘计算设备实现实时检测植物叶片病害成为可能。

1 植物病害实时检测系统实现流程

本研究的植物病害实时检测流程如图1所示,首先利用MobileNet-V2训练植物叶片目标检测模型,其次利用ResNet-152深度网络训练植物叶片病害分类模型。通过OpenCV对植物叶片目标检测模型和植物叶片病害分类模型整合成植物叶片病害检测系统,然后对优化模型前与优化模型后的植物叶片病害检测系统部署在主机设备和Jetson Nano设备上,进行实验比较,通过实验结果可以进一步分析模型优化前后的情况。

2 材料与方法

2.1 植物叶片目标检测模型的数据集

为了保证训练出来模型的鲁棒性和健壮性,增加数据的自发性。本实验的训练数据集样本来源多个数据集库。植物叶片检测模型数据来源:自建植物叶片数据集、加州大学欧文分校机器学习库、Mendeley数据集库。在自建数据集中通过高清摄像机拍摄植物叶片图像作为采集的训练和测试样本,数据集采集样本时间段为下午2点和6点。训练植物叶片目标检测模型的样本选择了包含不同大小、颜色和种类的叶片,保证了图像在光照、大小、聚类数量以及各种特征上都具有随机背景和亮度的变化,除了考虑植物病害叶片之外,还注重考虑正常植物叶片的数量。叶片目标检测

模型数据集来源具体如表 1 所示。

实验部分数据集样本图像如图 2 所示, 图 2(a) 和图 2(b) 是高清摄像机在下午 2 点拍摄; 图 2(c) 和图 2(d) 是摄像机于下午 6 点拍摄; 图 2(e) 是网络收集的实验

样本; 图 2(f) 是加州大学欧文分校机器学习库 (Folio data set) 的实验样本。图 2(g) 是 Mendeley_data 数据集中 Leaf Images 数据集的实验样本。图 2(h) 是本研究训练植物叶片病害分类模型的数据集的实验样本。

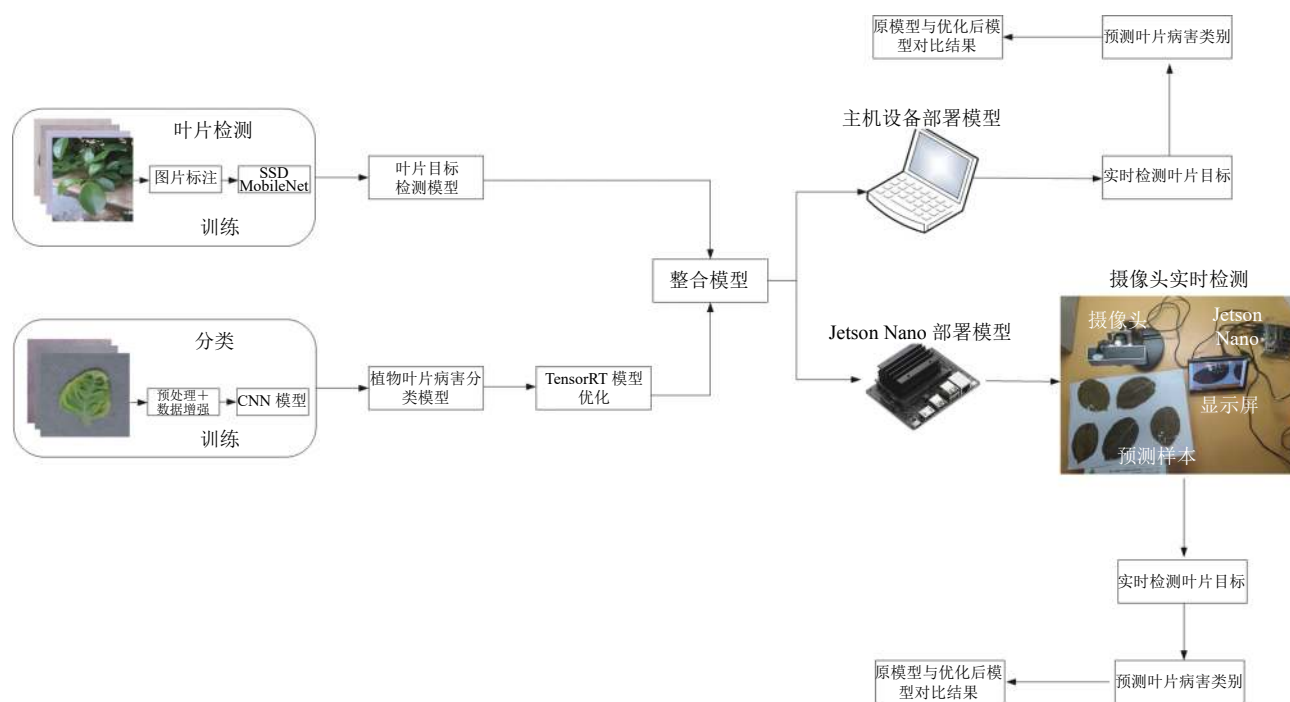


图 1 植物叶片病害检测实验流程

表 1 叶片目标检测模型数据集来源

叶片检测模型数据集来源	数量 (张)
自建植物叶片数据集	101
加州大学欧文分校机器学习库 (Folio data set) ^[9]	73
训练分类模型的数据集 (Mendeley) ^[10]	134
Leaf Images数据集 (Mendeley_data) ^[11]	724

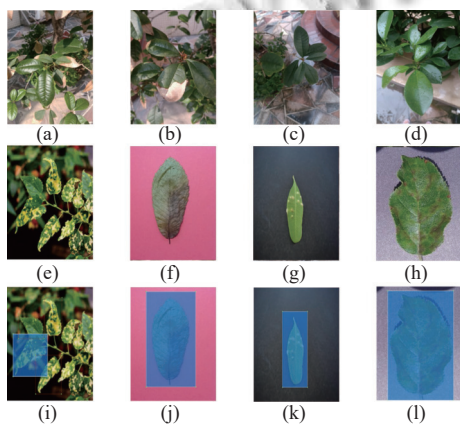


图 2 部分叶片检测模型数据集及样本叶片标注

2.2 LabelImg 对目标检测数据集标注

目标检测模型训练需要添加带有叶片坐标信息的 XML 文件, 即需要对含有叶子的图像区域进行人工标注. 在标注过程中, 用边界框标记叶子的类和区域. 使用 LabelImg 工具在叶子上绘制边界框, 并且标注为目标参数为 Leaves 类型, 生成包含每个图像中所有边界框各自坐标的 XML 文件. 数据集中经过标注的部分样本图像如图 2(i)、图 2(j)、图 2(k)、图 2(l) 所示. 在标注过程中, 舍弃了一些像素不佳、叶片形态不明显的叶片图像, 最终对剩下的 794 张叶片图像进行了标注。

2.3 MobileNet-V2 的 SSD 叶片目标检测模型

SSD 算法最先在 VGG-16 基础架构下实现了准确与快速地检测以及跟踪视频流. 原 VGG-16 网络只包含了两层原始的卷积层. 而 SSD 算法改造后的 VGG-16 网络, 增加了 4 个卷积部分. 如图 3 所示。

通过增加卷积部分, 使 SSD 网络对输入的图片进行特征提取, 提取出 $38 \times 38 \times 512$ 、 $19 \times 19 \times 1024$ 、 $10 \times$

10×512、5×5×256、3×3×256、1×1×256 这 6 个特征层. 特征层对输入的图像分割成不同数量的网格, 利用网格实现目标检测, 从而得到所需要的目标对象. SSD 目标检测算法的实现除了 VGG-16 外还有多种提取特

征的网络. 鉴于 MobileNet 网络拥有体积较小, 计算量可观的优点, 并且部署在嵌入式设备具有一定的速度优势^[12], 因此本研究的 SSD 目标检测算法, 基于 MobileNet-V2 网络实现目标检测.

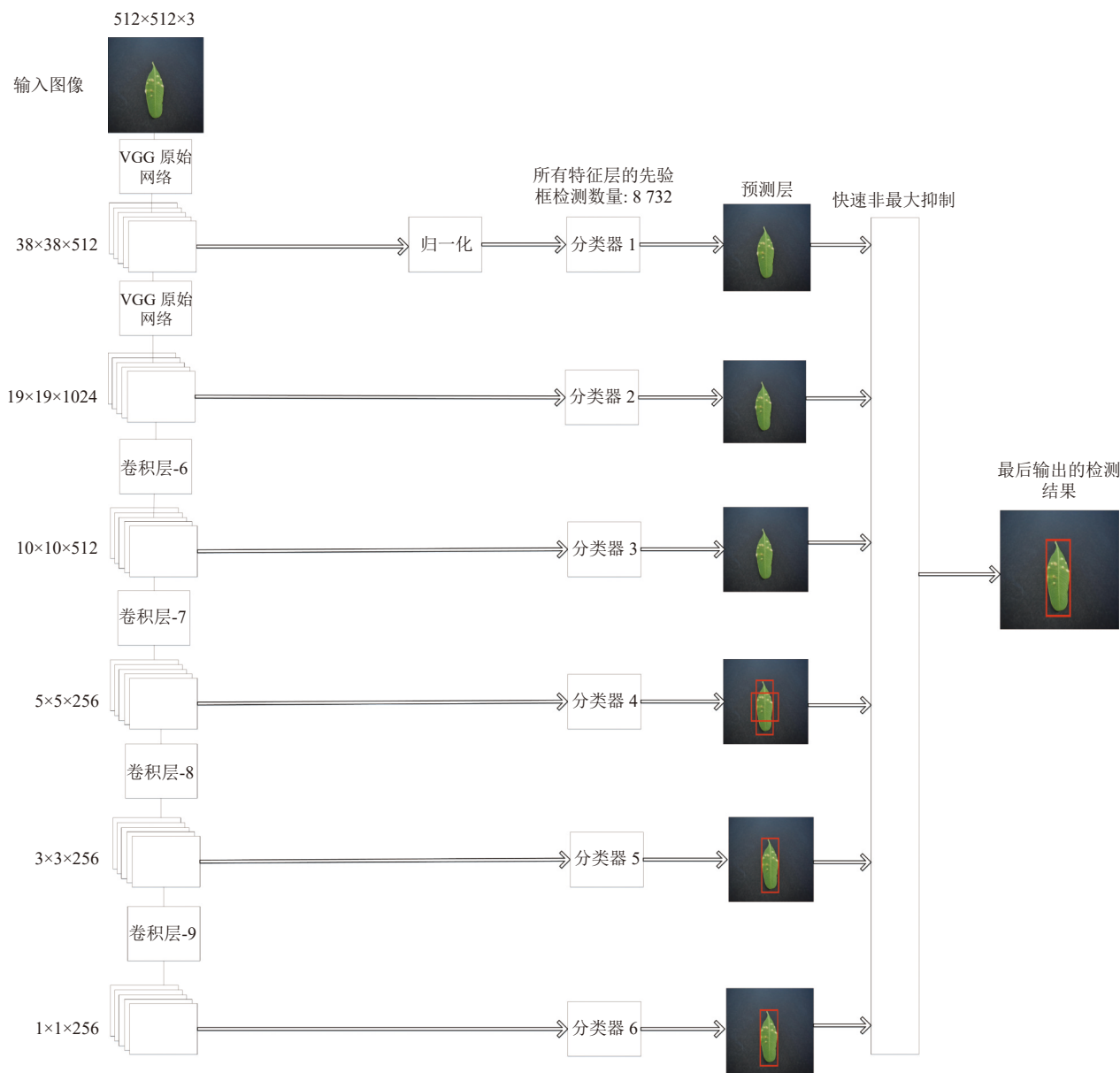


图3 基于 VGG-16 的 SSD 主干网络结构

在实现基于 MobileNet-V2 的 SSD 叶片目标检测模型之前, 需要对数据集进行一定比例的划分, 训练集与验证集的比例为 9:1, 训练集与验证集的总数量和测试集比例为 9:1, 则训练集为 642 张、验证集为 72 张、测试集为 80 张. 目标检测模型调用了 GPU 进行训练, 达到提高模型训练速度的目的. 训练前需要建立类别

对应的 txt 文件, 由于目标检测模型只有植物叶片一种类别, 则使用 txt 文件 leaves 标签. 在设定输入数据集的图片大小都调整为 512x512 尺寸. 此外, 选择相应的 MobileNet-V2 的权重参数模型, 对 SSD 植物叶片目标检测模型进行迁移学习. 本实验训练植物叶片目标检测的过程分成冻结训练和解冻训练两个阶段: 第 1 个

阶段使用权重参数进行训练,该阶段在设置目标检测模型的主干参数保持不变,仅对网络进行微调.训练进行50次迭代,鉴于硬件的性能,批处理 `batch_size` 的大小只设置为8.第2个阶段训练迭代的次数设置50次.模型的主干会在训练过程中发生变化,并且会引起特征提取网络改变.此阶段对显卡显存有较大占用,为了保证训练过程的持续进行,把批处理 `batch_size` 的值设置为4.植物叶片目标检测模型训练结束后,则使用测试集图像通过 SSD 叶片目标检测模型进行预测.将模型参数 `Confidence` 置信率设置为0.45,非极大抑制所用到的 `Nms_iou` 大小设置为0.45,结果如图4和图5所示,SSD 叶片目标检测模型不仅能够以较高的置信率识别到单个病害叶片,即使在比较光亮的环境仍然可以检测到叶片. SSD 叶片目标检测模型对于密集叶片中也可以检测出清晰的叶片.在复杂环境检测中,依然能够保持精准的目标检测.

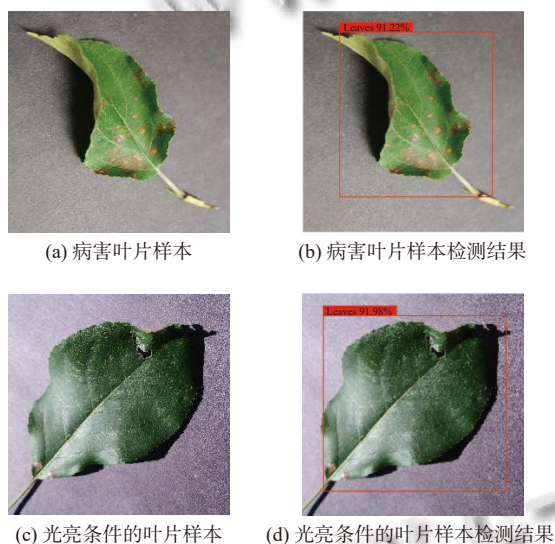


图4 单叶片使用 SSD 叶片目标检测模型检测

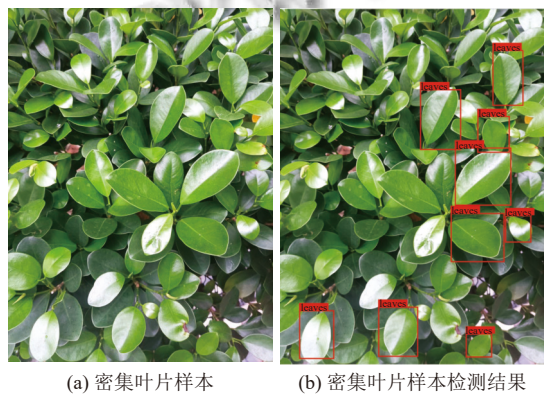


图5 多叶片使用 SSD 叶片目标检测模型检测

2.4 ResNet-152 的植物叶片病害分类模型

Geetharamani 等^[10]公开了有38种不同类别的植物叶片和一系列背景图像,此数据集使用了多种图像增强技术来增加数据集的数量,并使用增强数据的方式提高模型的分类能力.本研究在此增强数据集的基础上,挑选了苹果、樱桃、玉米、葡萄、桃子、灯笼椒、土豆、草莓、西红柿9种农业中常见的农作物叶片的33种健康或病害类型进行模型分类训练.选择上述9种农作物是在农业大量培植并供人食用或做工业原料的农作物,也建立了完整的病害叶片类型的防治图库,同时是目前在生活中常被人们食用的食物.本实验选择的此9种农作物作为植物叶片病害分类模型训练的数据集,目的是提升植物叶片病害检测系统的实用价值.本研究对数据集中33种健康和病害类型进行了大量的资料查证,收集到数据集中每个标签对应叶片具体病害类型.在训练分类模型之前,需要对数据集进行划分.植物叶片病害的训练集有36699张,验证集有9145张,每一种叶片病害的类型按照8:2的比例划分为训练集与验证集.植物叶片图片预处理之后,通过迁移学习方法构建和训练植物叶片病害分类模型.

本系统使用 Torchvision 库中的 ResNet-152 神经网络实现模型的预训练,采用已有模型的权重与偏置参数作为模型训练时初始化的参数.通过20次迭代的训练后,得到的最优叶片病害分类模型经过验证集测试后的精确度达到了95%,模型在训练集和验证集迭代后的损失率和准确率变化如图6所示.

2.5 OpenCV 对目标检测模型与分类模型整合

本研究合理调度了 SSD 植物叶片目标检测模型和植物叶片分类模型来完成对输入图像的植物叶片病害分类.并且结合了 OpenCV 对视频流进行实时的叶片病害识别.植物叶片病害检测系统运行流程如图7.

3 植物叶片病害分类模型的 TensorRT 优化

TensorRT 运用于深度学习推理优化器,能够给深度学习的应用带来少延时、高吞吐量的部署推理.基于英伟达的 TensorRT 应用程序,在推理过程中的运行速率比纯 CPU 平台要高效. TensorRT 支持 PyTorch 框架,同时提供了 Python 的接口程序,主要用来针对英伟达设计的 GPU 进行推理加速.本研究采用 TensorRT 对 PyTorch 的 Pth 模型的推理加速.通过降低少量识别

精度为代价,提高模型整体的检测效率,达到植物病害实时检测的效果。

TensorRT 模型优化方式,将模型神经网络的张量降为 FP16 或 INT8 的位数。为了更好地实现部署,推理的过程中不需要反向传播,从而减少模型神经网络的张量精度对模型造成大影响情况的发生。INT8 只有 8 个位数,仅表示 256 个数值范围,若 FP32 精度的模型转换成 INT8 精度的模型,会丢失大量精度信息,造成模型预测性能大幅下降。因此,为了保证模型性能的损失降到最小化,需要将分类模型的神经网络的张量精度转为 FP16。

PyTorch 模型经过 TensorRT 优化后生成的分类器,被英伟达官方命名为引擎。本研究对 TensorRT 优化的过程是基于 ResNet 的 PyTorch 模型先转换为 ONNX 模型再将其导入到 TensorRT 中生成高性能引擎,并将引擎从磁盘加载到内存中继续使用它进行推理。生成的引擎可转换成 Trt 格式并将其存储在磁盘上以供以后运行。上述的优化流程称为序列化引擎。将引擎从磁盘加载到内存中并继续使用它进行推理时,就会发生反序列化。如图 8 展示了引擎序列化和引擎反序列化的流程。

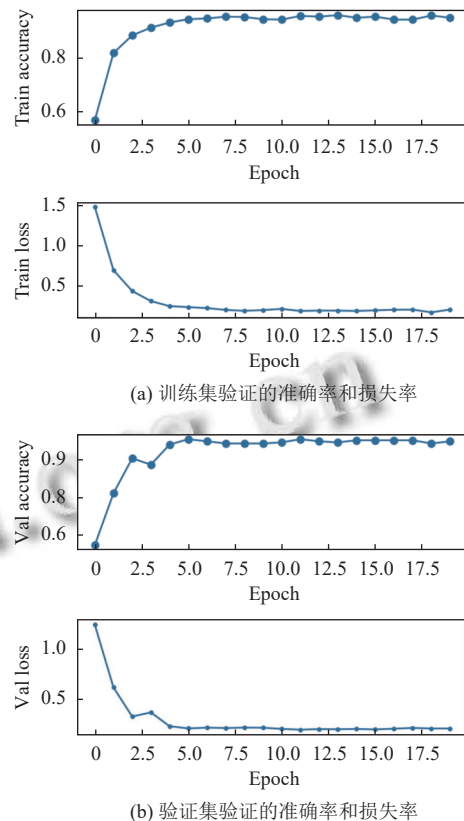


图 6 准确率和损失率变化图

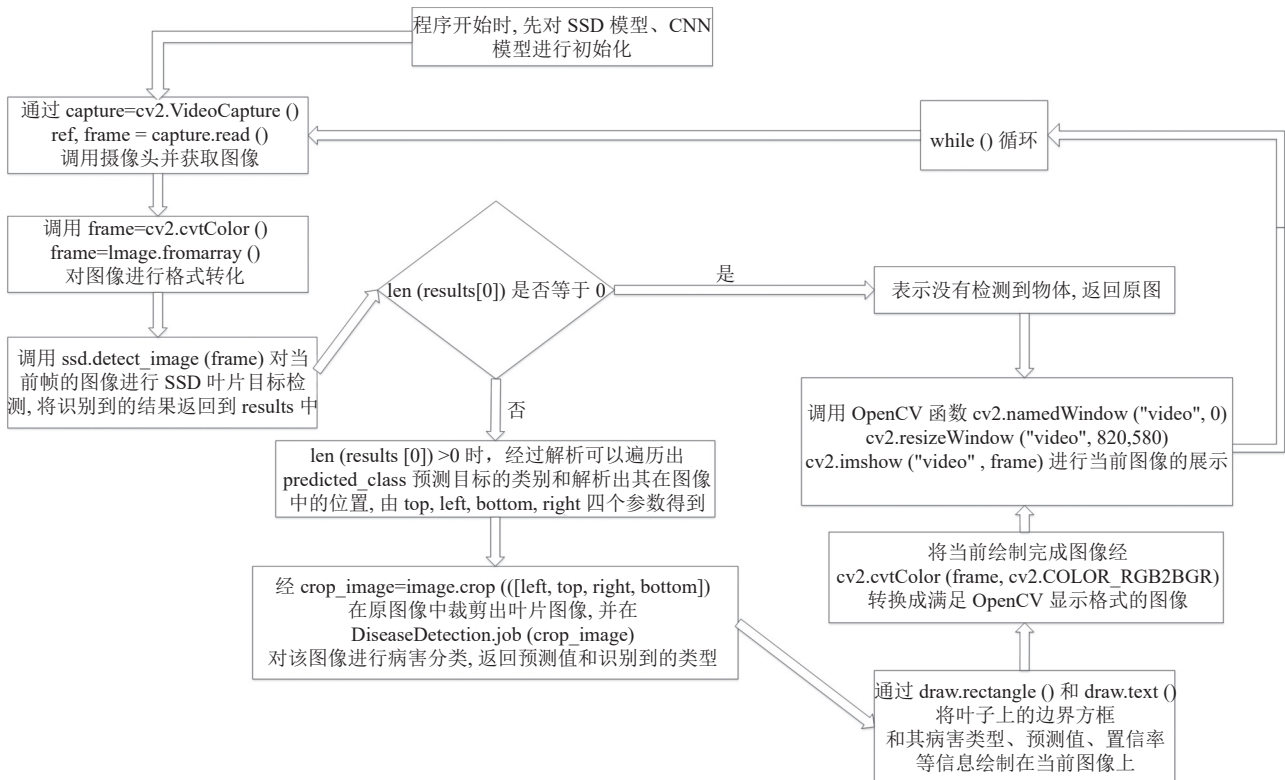


图 7 植物叶片病害检测系统运行流程

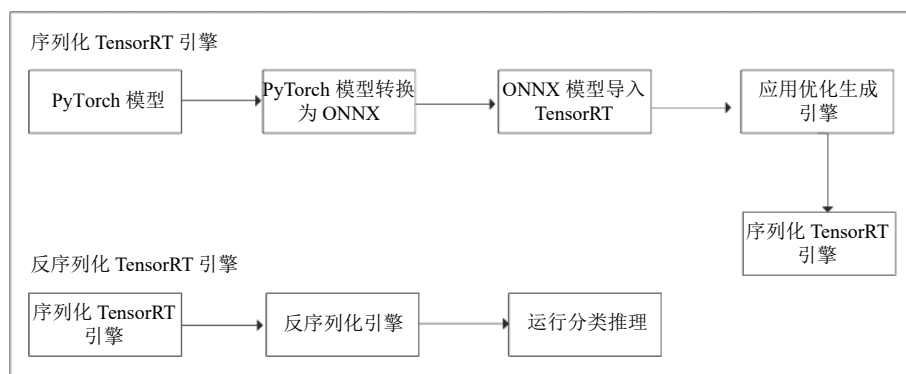


图8 引擎序列化和引擎反序列化流程

4 模型优化前后对比实验

本研究对 PyTorch 框架下调用植物病害分类模型与经过优化后的 TensorRT 引擎进行对比实验, 观察 TensorRT 引擎对于分类模型识别速度的提升效果. 再将优化前后模型分别结合 SSD 叶片目标检测模型对视频流进行实时的植物叶片病害检测, 观察 TensorRT 优化后对系统的整体检测速度提升效果. 为了保证实验的一致性, 对比实验在同一台主机上进行, 并且均使用 GPU 加载模型进行预测, 同时使用相同的验证数据集进行测试. 对比实验随机挑选验证集中的一张健康苹果 (Apple_healthy) 实验样本的图片数据进行预测, 对 PyTorch 框架下调用植物病害分类模型与经过优化后的 TensorRT 引擎的识别速度进行比较, 实验结果如表 2 所示. 可见两个模型都识别出来正确的类型, 虽然模型经 TensorRT 优化后, 其精度有所下降, 但是仅有 0.2% 的幅度, 影响微乎其微. 而识别速度却有质的提升, TensorRT 引擎相比于 PyTorch 模型的识别速度提升了 22 倍.

表 2 PyTorch 模型与 TensorRT 引擎对比

对比项目	PyTorch模型	TensorRT引擎
准确率 (%)	99.97266	99.78383
识别速度 (s)	0.6828470	0.0312399
识别类型	Apple_healthy	Apple_healthy

本实验还使用通过植物叶片病害检测系统调用优化前后的植物病害分类模型对视频流进行实时病害检测的对比实验. 实验的设计思路是在同一主机设备下调用摄像头分别对验证集中的同一样本叶片类型进行实时病害的识别检测, 分别观察优化前后的两个模型结合 SSD 叶片目标检测模型对样本叶片的检测速度和叶片识别的准确性. 实验挑选了灯笼椒辣椒疮痂病 (Pepper_Bell_Bacterial_Spot) 病害类型的一个样本进行测试. 为观察检测速度的快慢, 引进帧率进行衡量,

帧率即每秒帧数, 简称 FPS, 1 s 内可以处理的帧数越多, 则处理的图像就越多, FPS 就会越高, 屏幕上的显示就越流畅. 较低的帧速率会显得卡顿或检测的延迟较高, 使用帧率是评估模型性能和识别速度的重要指标. 图 9 显示了在同一主机设备上未经优化的 PyTorch 分类模型与经过优化的 TensorRT 引擎在结合 SSD 叶片目标检测模型进行实时的植物叶片病害检测时识别速率的对比, 图 9(a) 和图 9(b) 显示的是未经优化的 PyTorch 分类模型在结合 SSD 叶片目标检测模型的识别速率的表现, 可见实时检测装置没有检测到叶片时, 识别的速率是在 18 FPS 左右浮动, 而识别到叶片后, 在成功识别到正确病害类型的情况下, 仅有 1.28 FPS 的速率. 而从图 9(c) 和图 9(d) 展示的经过优化的 TensorRT 引擎结合 SSD 叶片目标检测模型的识别速率来看, 在没有识别到叶片的存在时, 其识别的速率有 17 FPS 左右, 与未经优化的 PyTorch 分类模型在结合 SSD 叶片目标检测模型的识别速率在相同情况下不相上下. 在识别到叶片后, 依旧可以成功识别到正确病害类型, 识别的速率为 9.71 FPS 左右, 帧率比未经优化的 PyTorch 分类模型在同等情况下相比提高了 7 倍, 同时在实验过程中植物叶片病害检测系统运行流畅. 因此, 经过优化的 TensorRT 引擎对系统的识别速率提升幅度十分明显. 而 TensorRT 引擎的加速效果在 Jetson Nano 计算平台上有更加明显的体现, 如图 10 所示. 图 10(a) 和图 10(c) 表示在没有识别到叶片的存在时, 优化前后植物叶片病害检测系统在 Jetson Nano 上运行时, 识别的帧率都有 3.6 FPS 左右, 而在对叶片的类型进行检测时, 未经优化的植物叶片病害检测系统在 Jetson Nano 的识别的帧率仅有 0.21 FPS, 如图 10(b) 所示. 经过优化的植物叶片病害检测系统在 Jetson Nano 的识别的帧率达到了 2.11 FPS, 提升了 10 倍的帧率, 如图 10(d)

所示. 由此可知, 优化后的植物叶片病害检测系统在 Jetson Nano 计算平台上达到实时检测叶片病害的效果.

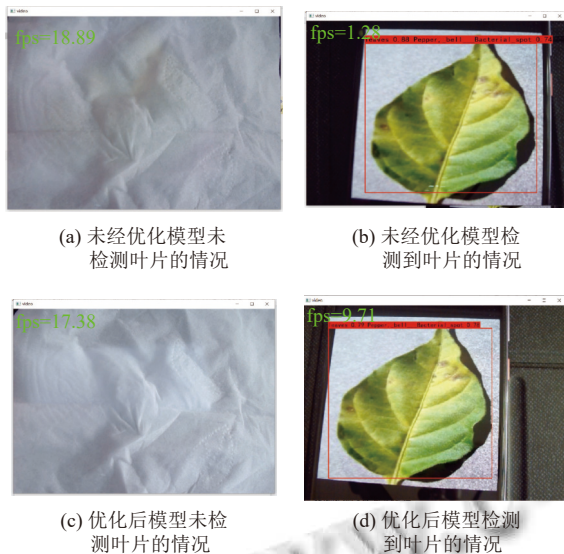


图9 主机设备上的模型优化对系统提升对比结果

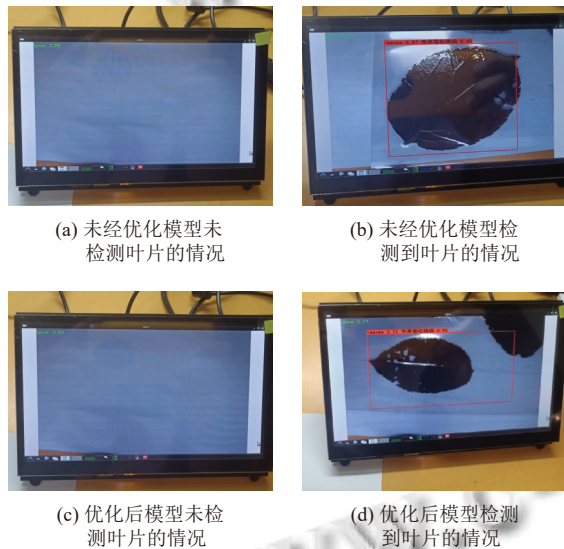


图10 Jetson Nano 上的模型优化对系统提升对比结果

5 结论与展望

本研究首先通过迁移学习的机器学习方法构建了叶片目标检测模型和植物叶片病害分类模型, 并通过 OpenCV 对两个模型进行整合, 完成植物叶片病害检测系统的实现. 为了再实现系统的实时检测, 对植物叶片病害分类模型进行了 TensorRT 优化, 并且对模型优化前后进行对比实验. 实验表明, 在同一主机设备上, 优化后的植物叶片病害分类模型对叶片类型的识别速度相较于优化前提升了 22 倍, 植物叶片病害检测系统的

识别帧率相较于优化前提升了 7 倍, 而在 Jetson Nano 计算平台运行植物叶片病害检测系统的识别帧率相较于优化前也提升了 10 倍, 达到了边缘计算设备进行实时检测的标准, 从而使采用廉价的硬件设备来实现实时植物叶片病害检测成为可能. 在未来的研究中, 我们将继续展开植物叶片病害实时检测系统在 Jetson Nano 计算平台上的优化实验, 使实时检测系统在面向复杂的环境中具备高效率的植物叶片病害识别.

参考文献

- 1 史红棚, 李修华, 李民赞, 等. 基于深度学习的香蕉病害远程诊断系统. 华南农业大学学报, 2020, 41(6): 92-99. [doi: 10.7671/j.issn.1001-411X.202004027]
- 2 乔虹. 基于深度学习的葡萄叶片病害动态监测 [硕士学位论文]. 兰州: 甘肃农业大学, 2019.
- 3 Fang Y, Ramasamy RP. Current and prospective methods for plant disease detection. Biosensors, 2015, 5(3): 537-561. [doi: 10.3390/bios5030537]
- 4 Cséfalvay L, Di Gaspero G, Matouš K, et al. Pre-symptomatic detection of *Plasmopara viticola* infection in grapevine leaves using chlorophyll fluorescence imaging. European Journal of Plant Pathology, 2009, 125(2): 291-302. [doi: 10.1007/s10658-009-9482-7]
- 5 Oppenheim D, Shani G, Erlich O, et al. Using deep learning for image-based potato tuber disease detection. Phytopathology, 2019, 109(6): 1083-1087. [doi: 10.1094/PHYTO-08-18-0288-R]
- 6 Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2012. 1097-1105.
- 7 Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2818-2826.
- 8 Gajjar R, Gajjar N, Thakor VJ, et al. Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform. The Visual Computer, 2022, 38(8): 2923-2938. [doi: 10.1007/s00371-021-02164-9]
- 9 Asuncion A, Newman DJ. UCI machine learning repository. Technical Report, Irvine: Irvine University of California, 2007.
- 10 Geetharamani G, Arun Pandian J. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Computers & Electrical Engineering, 2019, 76: 323-338.
- 11 Siddharth SC, Singh UP, Kaul A, et al. A database of leaf images: Practice towards plant conservation with plant pathology. Mendeley Data, 2019, V4. [doi: 10.17632/hb74ynkjcjcn.4]
- 12 Howard AG, Zhu ML, Chen B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861, 2017.

(校对责编: 孙君艳)