基于 YOLO V4-TLite 的移动端君子兰病虫害检测方法

宋芝文[1]，李 伟[2]，谭 伟[3]，覃 涛[1]，刘 杰[4]，杨 靖[1,5※]

（1. 贵州大学 电气工程学院，贵阳 550025；2. 贵州大学 农学院，贵阳 550025；3. 贵州大学 林学院，贵阳 550025；4. 中国电建集团 贵州工程有限公司，贵阳 550025；5. 贵州省互联网+协同智能制造重点实验室，贵阳 550025）

摘 要：针对大棚和园林环境识别君子兰病虫害存在实时性差、检测精度低、过度依赖高算力和硬件功耗高等问题，提出一种面向移动端执行的 YOLO V4-TLite 君子兰病虫害检测方法。首先，以 YOLO V4-Tiny 为基础，使用低成本的部分卷积代替主干网络中的传统卷积。其次，使用逆残差网络结构，形成轻量化主干网络。再次，使用通道融合采样层机制，提升网络的鲁棒性和准确性。最后，将改进模型迁移部署在 ROCK 5B 移动端上，并针对君子兰 3 种典型病虫害叶枯病、黄斑病和介壳虫进行试验。试验结果表明，该改进模型的平均精度均值 (mean average precision, mAP) 为 78.5%，内存占用量仅为 4.8MB，浮点数运算量 (floating point operations, FLOPs) 为 1.3 G，最大卷积计算的随机存储器 (random access memory, RAM) 储存为 1 MB；桌面端单张检测速度为 0.005 s，功耗为 70 W；在移动端，CPU 单张检测速度为 0.239 s，功耗为 10 W，NPU 单张检测速度为 0.018 s，功耗为 7 W。YOLO V4-TLite 模型在低资源和低功耗的移动端进行君子兰病虫害检测，其相比于现有主流 YOLO 系列模型具有较好的竞争力。

关键词：君子兰；病虫害；YOLO V4-Tiny；轻量化；移动端部署

doi：10.11975/j.issn.1002-6819.202409169

中图分类号：S24；TP18 **文献标志码**：A **文章编号**：1002-6819(2025)-05-0175-07

宋芝文，李伟，谭伟，等. 基于 YOLO V4-TLite 的移动端君子兰病虫害检测方法[J]. 农业工程学报，2025，41(5)：175-181. doi: 10.11975/j.issn.1002-6819.202409169 http://www.tcsae.org

SONG Zhiwen, LI Wei, TAN Wei, et al. Detection method for Clivia Miniata pests and diseases on mobile terminal based on YOLO V4-TLite[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2025, 41(5): 175-181. (in Chinese with English abstract) doi: 10.11975/j.issn.1002-6819.202409169 http://www.tcsae.org

0 引 言

君子兰是石蒜科君子兰属植物，因其叶片翠绿、花朵雅致且具有一定的药用价值而得到广泛栽培[1-3]。但是，因君子兰种植环境的原因，难以部署具有高算力、高功耗硬件设备进行实时识别，导致君子兰病虫害的检测主要依赖人工观察和经验判断，难以满足大规模种植场景病虫害实时检测和诊断的需求。同时君子兰病虫害的深度学习诊断鲜有人研究，且无公共数据集。因此，利用可边缘部署的深度学习检测技术对君子兰病虫害进行高效准确地检测，是提高君子兰成活率和品质的关键。

在农业领域，单阶段目标检测算法 YOLO 系列以其高准确率和高效率而被广泛使用[4-6]。然而，由于 YOLO 为计算密集型和内存密集型算法，因此需要更大计算能力和存储资源[7]。而且，YOLO 算法是针对通用多类别场景提出，在简单的检测中存在较多的网络冗余[8]。这种网络冗余和较高的算力消耗，使得在计算资源和存储能力有限的移动端上运行 YOLO 算法时，计算时间过长，难以满足实时性的需求[9]。因此，构建轻量化的深度学习模型用于移动端农业检测是目前的研究热点[10-12]。

宋浩[13] 通过提取君子兰的颜色特征和纹理特征，使用各个特征组成特征向量，并使用 SVM 分类器对君子兰 3 种病虫害进行分类判断。王卫星等[14] 使用 GhostNet 网络替代了原 YOLO V4 模型中的主干网络，并使用注意力机制 CBAM，改进后模型缩小了 84%，速度提升了 38%，最后在自建荔枝病虫害数据集上，平均精度达到了 90%。LIU 等[15] 针对黄瓜病害的检测问题，提出具有特征融合模块和预测模块的 EFDet 模型，该模型鲁棒性好，参数和计算量少。

上述研究为桌面端的轻量化病虫害检测提供了解决方案，但在实际农业生产中，设备通常以计算能力有限的移动端设备为主，这使得现有的轻量化模型仍需进一步优化以适应移动端设备的资源约束[16]。LUO 等[17] 提出 LiteCNN 网络模型，该模型使用可深度分离卷积代替传统卷积，并使用分块状卷积方式进行数据加载。使用知识蒸馏后部署在 FPGA 板上，植物病害数据集上计算速度达到每张图片用时 0.071s，功率 2.31W。XU 等[18] 在 YOLO V5-S 的基础上使用 MobileNet和部分卷积对其

收稿日期：2024-09-23 修订日期：2025-02-09

基金项目：国家自然科学基金项目（No. 61640014）；贵州省教育厅工程研究中心（黔教技 [2022]040）；贵州省教育厅创新群体（黔教合 KY 字 [2021]012）；贵州省科技支撑计划（黔科合支撑 [2022] 一般 017，黔科合支撑 [2023] 一般 411，黔科合支撑 [2024] 一般 051）；贵州省科技成果转化项目：黔科合成果-LH[2024] 重大 028；贵州省基础研究计划（黔科合基础-zk[2025] 面上 596）

作者简介：宋芝文，研究方向为智慧农业与嵌入式系统。

Email：gs.zwsong23@gzu.edu.cn

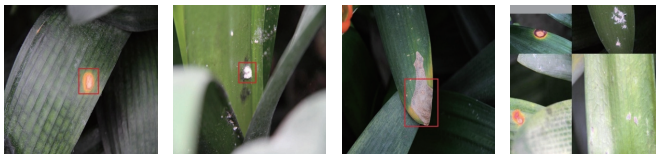
※通信作者：杨靖，教授，研究方向为物联网技术与应用与群体智能优化研究。Email：jyang7@gzu.edu.cn

进行 Backbone 和 Neck 网络的改进，改进模型达到 90%的精确率和 6.1G 的浮点数运算量，并将模型迁移部署移动端平台上，实现对苹果叶片部分的病害检测。

上述研究，重点关注如何提高模型在移动设备上的推理速度。然而，在边缘设备上，硬件功耗和模型的硬件兼容性也是必需考虑的因素[19]。特别是在移动平台计算环境下，模型需要具备较低功耗且高精度的检测能力，同时还应具备较小的内存和快速的推理等特点[20]。针对上述问题，面向资源有限的移动端进行君子兰病虫害检测的应用需求，本文提出一种改进的轻量化模型 YOLO V4-TLite，该模型保持着高硬件友好性的同时还拥有较好的检测精度。

1 数据集准备

本文使用的典型君子兰病虫害数据集采集于贵州省贵阳市花溪区，采集时间为冬季和春季两个季节，包括君子兰病虫害发病的早期和中期的图片，但由于早期病害不明显，观察难度高，所以本文主要面向君子兰病虫害发病中期进行测试和验证。同时通过使用 Mosaic 数据集增强功能[21]，使得图片语义信息更丰富。图 1 展现了该君子兰典型病虫害数据集类别和 Mosaic 数据集增强方法。



a. 黄斑病 a. Maculopathy　b. 介壳虫 b. Coccid　c. 叶枯病 c. Leaf blight　d. Mosaic方法 d. Mosaic method

图 1 数据集图片

Fig.1 Image of dataset

该数据集格式为 VOC 格式，叶枯病、黄斑病和介壳虫 3 种典型病虫害，原始数据集大小为 1 420 张，数据集增强后达到 2 155 张，每张图片分辨率为 800 × 780 像素，按照 8:1:1 的比例划分为训练集、验证集和测试集。数据集详细构建见表 1。

表 1 数据集病虫害标签分布

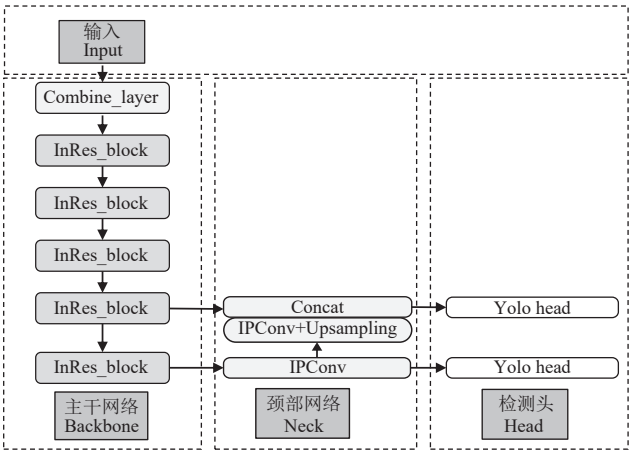
Table 1 Distribution of pest and disease labels in the dataset

类别名称 Class name	原始数据 Original data	增强数据 Enhanced data	总计 Total
介壳虫 Coccid	4 891	2 624	7 515
黄斑病 Maculopathy	1 572	947	2 519
叶枯病 Leaf blight	1 108	697	1 805
总计 Total	7 571	4 268	11 839

2 模型改进

2.1 YOLO V4-TLite 算法

为了在资源有限的移动端实现君子兰病害的实时识别，本文基于 YOLO V4-Tiny 目标检测网络，提出了 YOLO V4-TLite 网络模型，该网络做了如下改进：1）在模型的输入端加入采样卷积层（Combine_layer），实现冗余特征图的减少和精度的提升；2）使用深度可分离卷积对部分卷积进行改进得到一种改进的部分卷积（IPConv），并将其替换在主干网络和颈部网络中，实现模型检测速度的提升。3）使用逆残差网络结构和改进的部分卷积（IPConv）组合得到一种 InRes_block 卷积块，并将其替换在主干网络中，实现模型硬件兼容性的提升。总体改进结构如图 2 所示。



注：Upsampling 为上采样操作，Concat 为特征融合函数，Yolo head 是检测头模块。

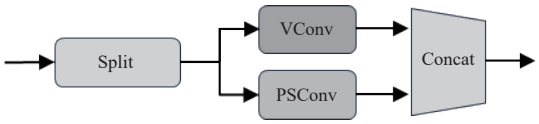
Note：Upsampling is upsampling, Concat is feature fusion function, Yolo Head is the detection head module.

图 2 YOLO V4-TLite 模型结构图

Fig.2 Model structure diagram of YOLO V4-TLite

2.1.1 卷积层改进

针对传统卷积层处理位置信息固定，导致模型模型的泛化能力差的问题，本文提出一种由权重共享卷积和传统卷积组合形成的全新采样卷积层（Combine_layer），概念计算如图 3 所示。权重共享通过在网络的不同位置共享权重，有效减少了模型的参数数量，从而提高了模型的泛化能力[22]。同时，传统卷积的保留具有一定程度的灵活性，可以根据具体任务需求灵活调整网络的结构，更好地适应不同的数据分布和任务类型。



注：Split 是特征切片函数，VConv 是传统卷积，PSConv 是权重共享卷积。

Note: Split is the feature slice function, VConv is the vanilla convolution, PSConv is the parameter sharing convolution.

图 3 全新采样层计算概念图

Fig.3 New sampling layer computing conceptual diagram

本文提出的全新采样层在参数量上小于传统卷积层，一般传统卷积层，需要的参数量L_{para}为

$$L_{para} = k^2 c_{in} c_{out} \tag{1}$$

式中，k 为卷积核大小，c_{in}为输入特征图通道数，c_{out}为输出特征图通道数，但在全新采样层中，需要的参数量L_{cpara}为

$$L_{cpara} = k^2 c_{in} \frac{c_{out}}{2} + k^2 \frac{c_{out}}{2} \tag{2}$$

由式（1）和式（2）可知，先进的采样层对比传统卷积层的压缩率r可表示为

$$r=\frac{k^2c_{in}\frac{c_{out}}{2}+k^2\frac{c_{out}}{2}}{k^2c_{in}c_{out}}=\frac{1}{2}\frac{c_{in}+1}{c_{in}}\approx\frac{1}{2}\tag{3}$$

对于卷积神经网络，通常具有较大的特征图通道数，因此可以认为式（3）中的$c_{in}=c_{in}+1$，全新采样层对比传统卷积层的参数量压缩率为 0.5。

采样层优化前后输出对比如图 4 所示，即当输入图片为 416 × 416 × 3 像素时，分别经过第一层卷积层后的特征图表现。图 4 中优化前特征图中黑色框的特征图具有较高的重合，过多的冗余特征图会影响到模型的鲁棒性。然而，经过优化后冗余特征图明显减少，模型的精度得到了提升。

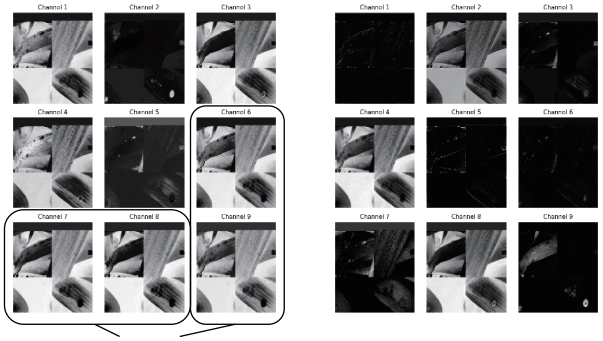


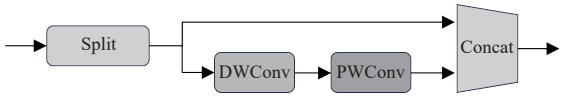
图 4　采样层改进前后输出对比

Fig.4　Output comparison before and after sampling layer improvement

2.1.2　卷积改进

针对传统卷积方式存在高运算和低灵活性而导致硬件功耗过高的问题，本文使用部分卷积[23]和深度可分离卷积组合得到一种改进的部分卷积（IPConv）。

IPConv 首先对输入特征图的部分通道进行深度可分离卷积计算，而其余部分则直接复制到输出层。这种设计不仅减少了计算量，还在网络结构中实现了跨层的信息传递，其计算示意图如图 5 所示。



注：DWConv 为逐通道卷积，PWConv 为逐点卷积。

Note: DWconv is the depthwise convolution, PWConv is the pointwise convolution.

图 5　改进部分卷积计算示意图

Fig.5　Schematic of improved partial convolution computation (IPConv)

对 IPConv 进行性能分析，输入特征图$I\in\mathbb{R}^{c\times h\times w}$，卷积核$W\in\mathbb{R}^{k\times k}$，输出特征图$O\in\mathbb{R}^{c\times h\times w}$，IPConv 在连续内存访问下使用部分的特征图通道进行卷积运算，这相比于传统卷积和传统部分卷积具有较低的（floating point operations, FLOPs）和较少的硬件内存访问数量（memory access cost, MAC）。传统卷积的 FLOPs 为

$$\text{FLOPs}=hwk^2c^2\tag{4}$$

未改进部分卷积 FLOPs*为

$$\text{FLOPs}^*=hwk^2c_p^2\tag{5}$$

改进的部分卷积的 FLOPs**为

$$\text{FLOPs}^{**}=hw\times3^2\times c_p+hw\times1^2\times c_p^2\tag{6}$$

传统卷积的硬件内存访问数量 MAC 为

$$\text{MAC}=hw\times2c+k^2c^2\tag{7}$$

未改进部分卷积的硬件内存访问数量 MAC*为

$$\text{MAC}^*=hw\times2c_p+k^2c_p^2\tag{8}$$

改进后部分卷积的硬件内存访问数量 MAC**为

$$\text{MAC}^{**}=hw\times2c_p+3^2c_p+1^2c_p^2\tag{9}$$

当$\frac{c_p}{c}=\frac{1}{4}$，$k=3$时，由式（4）和（5）可知，未改进部分卷积的 FLOPs 仅仅是传统卷积的$\frac{1}{16}$，由式（5）和（6）可知，改进后的部分卷积 FLOPs 仅为未改进部分卷积的$\frac{9+c_p}{k^2\times c_p}\approx\frac{1}{k^2}=\frac{1}{9}$；由式（7）和（8）可知，未改进部分卷积的内存访问数量约是传统卷积的$\frac{c_p}{c}=\frac{1}{4}$。由式（8）和（9）可知，改进后部分卷积的内存访问数约是未改进部分卷积的$\frac{h\times w\times2+c_p}{h\times w\times2+9c_p}<1$。

2.1.3　主干网络改进

针对传统主干网络结构臃肿而导致模型在有限 RAM 的移动端无法部署的问题，本文提出一种由 IPConv 和逆残差网络块[24]组合形成的新型卷积块结构（InRes_block），解决传统深度学习网络由于主干网络过于复杂和庞大而导致硬件兼容性差的问题。

InRes_block 网络架构如图 6 所示，其利用跳跃连接和梯度传播机制有效减轻了梯度消失问题，并提高了模型的性能和泛化能力。改进后模型的网络主干参数信息如表 2 所示。

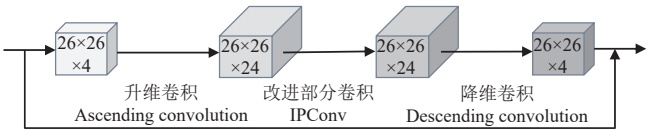


图 6　InRes_block 网络架构

Fig.6　InRes_block network architecture

表 2　改进后网络主干参数信息

Table 2　Improved network backbone parameter information

输入特征图大小 Input feature map size	操作 Options	t	c	n	m
$416^2\times3$	Combine_layer	1	32	1	2
$208^2\times32$	InRes_block	1	64	1	2
$104^2\times64$	InRes_block	2	128	2	2
$52^2\times128$	InRes_block	2	256	2	2
$26^2\times256$	InRes_block	1	512	1	2
$13^2\times512$	InRes_block	1	512	2	1

注：t 为逆残差网络中的放大倍数；c 为模型输出通道数；n 为该操作的重复数；m 为卷积步长。

Note: t is the magnification in the inverse residual network; c is the number of output channels of the model ; n is the number of repetitions of the operation ; m is the convolution step.

2.2　试验环境与评价指标

2.2.1　试验环境配置

本研究关于训练和部署用到的硬件平台和计算环境

如表 3 所示，训练迭代次数为 300 次，输入图像分辨率 416 × 416 像素。完成训练后，在测试集上进行综合性能测试，模型测试时的交并比 (IoU) 阈值设置为 0.5。

2.2.2 试验评价指标

为更加准确的评测模型性能，模型的速度指标为 1 000 幅图像推理用时平均值，计算出一秒能进行多少次图像推理（帧/s），模型的精度指标为在给定的 IoU 阈值为 0.5 时的平均精度均值（mAP），模型计算效率 FPJ（frames per joule）表示每焦耳能量处理图片的张数。其中对于改进前后模型性能参数提升百分比 I_U 和模型性能参数减少百分比 I_D 计算如下：

$$I_U = \left(\frac{I_p - O_p}{O_p}\right) \times 100\% \quad (10)$$

$$I_D = \left(\frac{O_p - I_p}{O_p}\right) \times 100\% \quad (11)$$

$$\mathrm{FPJ} = \frac{s^{-1}}{W} = \frac{1}{W \times s} \quad (12)$$

式中 I_p 为改进后性能参数；O_p 为改进前的性能参数；s 为处理一张图片的速度，s；W 为模型功耗，W；

表 3 模型训练与部署平台信息

Table 3 Model Training and Deployment Platform Information

类型 Type	名称 Name	配置 configuration
训练 Train	系统 System	Windows 11
	CPU	Intel i9-13900HX
	GPU	RTX 4 060
	内存 RAM	16 GB
部署 Deployment	系统 System	Ubuntu22.04 LTS
	CPU	Cortex-A76, A55
	GPU	Mali G610MP4
	内存 RAM	8 GB

3 结果与分析

3.1 模型训练与部署

3.1.1 模型训练与结果分析

试验以 YOLO V4-Tiny 为基础，结合不同改进策略，在君子兰病虫害数据集下进行开展 6 组试验，并选取平均精度均值和参数量作为评价指标。消融试验对比结果如表 4 所示。

表 4 YOLO V4-TLite 消融试验结果

Table 4 Comparison results of ablation experiments

模块设置 Module Settings			平均精度均值	参数量
Combine_layer	InRes_block	IPConv	mAP/%	Parameters/×10^{6}M
–	–	–	65.9	5.8
√	–	–	68.1	5.8
–	√	–	74.8	2.1
–	√	√	76.8	1.2
√	√	–	76.5	2.1
√	√	√	78.5	1.2

注：“√” 表示添加该结构；“–” 表示未添加该结构；

Note: “√” means added structure; “–” means unadded structure.

从表 4 可知，单独加入 Combine_layer 模块时，模型 mAP 提升了 2 个百分点，但参数量不变。单独加入 InRes_block 模块时，模型 mAP 提升了 9 个百分点，同时模型参数量减少了 3.7×10^{6}M。同时加入 IPConv 和 InRes_block 模块时，相比于只加入 InRes_block 模块的模型精度提升了 2 个百分点，且参数量减少了 0.9×10^{6}M。

为评估改进模型的整体性能，本文引入目前主流算法 NanoDet、YOLO V7-Tiny[25]、YOLO V5-S、MobileNetV2、GhostNetV1[26]、YOLO V10-N[27]、YOLO V11-N[28]。使用 MobileNetV2 和 GhostNetV1 分别替换 YOLO V4-Tiny 模型的主干网络，得到 YOLO V4-TM 和 YOLO V4-TG 模型。由于目前硬件部署算法较少选择二阶段算法，本文采用一阶段算法进行对比。将改进模型与其他模型进行 mAP、训练集损失值和验证集损失值变化对比。各模型训练结果测试曲线如图 7 所示。

由图 7 可知，在 mAP 上，本文改进的模型 YOLO V4-TLite 比图中其他各模型表现更好，在验证集上的 mAP 最高的模型为 YOLO V4-TLite，达到了 82%，NanoDet 模型表现最差，仅仅达到 62%。各模型的更多性能参数如表 5 所示。

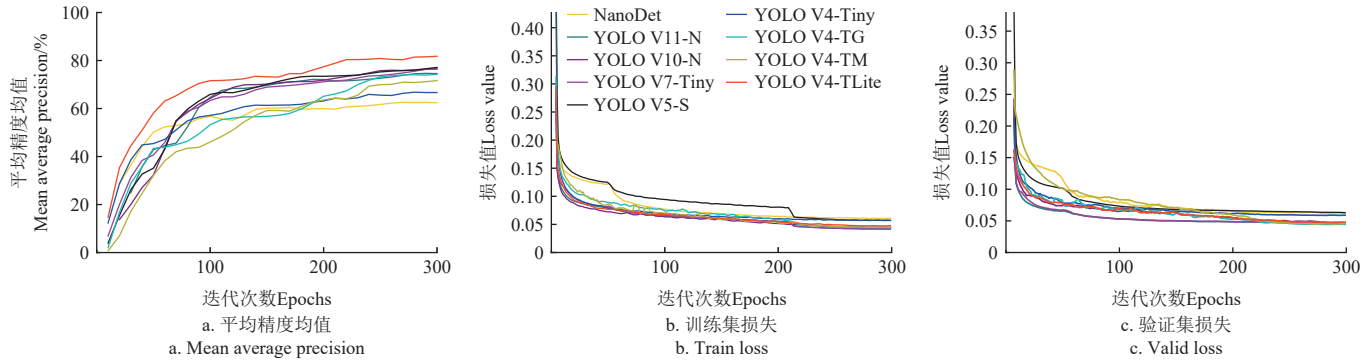


图 7 各模型训练结果测试

Fig.7 Testing of training results for each model

通过表 5 可知，YOLO V4-TLite 在 9 个模型中储存占用量最小，仅为 4.8 MB，相比于 YOLO V4-Tiny，改进模型大小减小了 78.57%，并且 mAP 为 9 个模型中最高，其中相比于 YOLO V4-Tiny 提升了 12.6 个百分点，与 YOLO V7-Tiny、YOLO V5-S 模型相差不大，但 YOLO V4-TLite 模型大小、RAM 储存和 FLOPs 均优于 YOLO V7-Tiny 和 YOLO V5-S 模型。虽然 YOLOV11-N 和 YOLO V10-N 具有较小的 RAM 消耗，但是其模型

大小，mAP 和运算量低于本文提出的 YOLO V4-TLite 模型。

表 5　各模型性能测试结果

Table 5　Performance test results for each model

模型 Model	模型大小 Model size/MB	随机存取存储器 Random access memory /MB	介壳虫 Coccid		叶枯病 Leaf blight		黄斑病 Maculopathy		平均精度均值 mAP/%	运算量 Computation/GFLOPs
			F1 分数 F1 score/%	召回率 Recall/%	F1 分数 F1 score/%	召回率 Recall/%	F1 分数 F1 score/%	召回率 Recall/%		
NanoDet	8.1	2.1	54.4	44.8	48.5	32.3	68.7	71.2	60.3	2.4
YOLO V11-N	5.3	0.6	64.0	58.2	67.8	65.5	86.1	82.5	74.6	6.3
YOLO V10-N	5.5	0.6	67.1	61.5	67.9	66.5	88.2	86.6	76.2	8.2
YOLO V7-Tiny	23.1	4.5	57.8	42.9	75.8	64.2	85.4	83.0	76.9	5.6
YOLO V5-S	27.1	4.5	63.2	50.3	74.2	62.7	84.2	83.0	77.2	7.0
YOLO V4-Tiny	22.4	9.0	54.5	45.2	45.1	30.2	78.5	76.6	65.9	6.8
YOLO V4-TM	14.9	1.2	64.6	59.6	67.6	64.1	87.1	87.1	74.8	2.3
YOLO V4- TG	11.1	0.9	65.1	60.3	66.0	65.1	82.8	88.0	73.0	1.5
YOLO V4-TLite	4.8	1.0	70.0	65.4	64.3	62.5	88.0	88.4	78.5	1.3

3.1.2　模型部署与结果分析

为检验改进模型在移动端的表现，硬件部署采用瑞芯微的 ROCK 5B 开发板，实物如图 8 所示。该硬件平台功耗低于 PC 机和工作站，但其有适中的算力和外设接口，且具有低的价格，适合作为边缘计算终端。

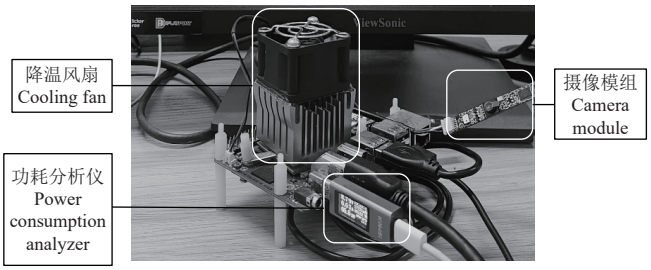


图 8　ROCK 5B 实物图

Fig.8　ROCK 5B physical picture

本文采用移动端 CPU 和 NPU 进行加速测试，并与桌面端部署进行比对，计算上述 9 个模型在不同平台的运算速度和功率消耗，得出各自的计算效率。模型在移动端的部署流程如图 9 所示。

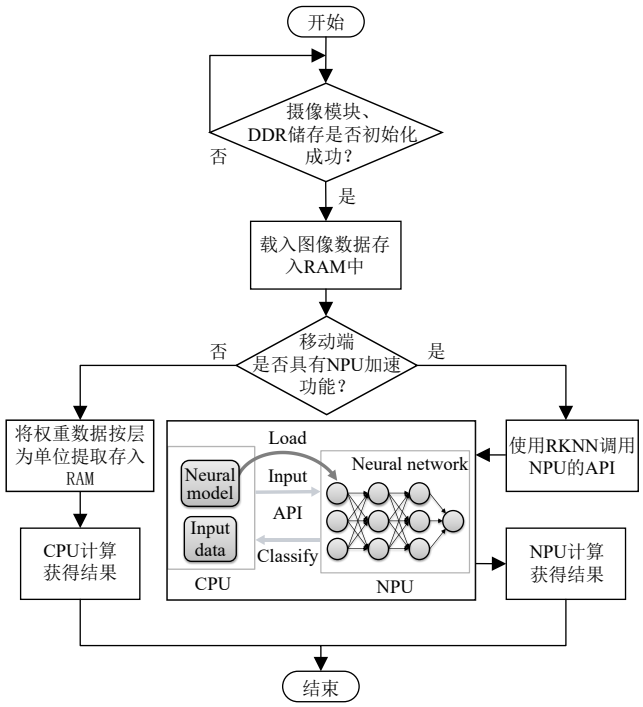


图 9　移动端部署流程图

Fig.9　Mobile deployment flowchart

部署完成后，进行速度和功耗测试，使用单张图片进行循环推理，桌面端和移动端 NPU 平台采用 100 次循环测试，移动端 CPU 平台采用 10 次循环测试。其中各模型在移动端运行的速度和功耗试验数据如表 6 所示。

表 6　模型部署后的综合性能对比

Table 6　Total performance comparison of models after deployment

模型 Model	平台 Platform	速度 Speed/s	功耗 Consumption/W	效率 Efficiency/(张·J^{-1})	价格 Price/元
NanoDet	桌面端	0.005	75	2.67	≥8 499
	移动端-CPU	0.241	11	0.38	≥649
	移动端-NPU	0.025	8	5.00	
YOLO V11 -N	桌面端	0.006	76	2.19	≥8 499
	移动端-CPU	0.235	12	0.36	≥649
	移动端-NPU	0.032	8	3.91	
YOLO V10- N	桌面端	0.006	78	2.14	≥8 499
	移动端-CPU	0.242	12	0.34	≥649
	移动端-NPU	0.035	8	3.57	
YOLO V7-Tiny	桌面端	0.006	80	2.08	≥8 499
	移动端-CPU	0.235	12	0.35	≥649
	移动端-NPU	0.028	9	3.97	
YOLO V5-S	桌面端	0.008	76	1.65	≥8 499
	移动端-CPU	0.251	12	0.33	≥649
	移动端-NPU	0.038	9	2.92	
YOLO V4-Tiny	桌面端	0.005	96	2.08	≥8 499
	移动端-CPU	0.214	12	0.39	≥649
	移动端-NPU	0.026	9	4.27	
YOLO V4-TM	桌面端	0.006	71	2.35	≥8 499
	移动端-CPU	0.271	11	0.34	≥649
	移动端-NPU	0.021	8	5.95	
YOLO V4- TG	桌面端	0.010	65	1.54	≥8 499
	移动端-CPU	0.252	11	0.36	≥649
	移动端-NPU	0.017	8	7.35	
YOLO V4-TLite	桌面端	0.005	70	**2.86**	≥8 499
	移动端-CPU	0.239	10	**0.42**	≥649
	移动端-NPU	0.018	7	**7.94**	

注：移动端-CPU 为移动端通用计算的中央处理器；移动端-NPU 为移动端专用加速神经网络计算的神经网络计算器。

Note : Mobile-CPU is the central processor of general computing; Mobile-NPU is the dedicated neural network calculator that accelerates neural network computing.

改进模型桌面端单张检测速度为 0.005 s，功耗为 70 W，计算效率 2.86，为 9 个模型最优。在桌面端上，改进模型的计算效率提升了 35.7 个百分点，在移动端上，CPU 计算效率提升了 7.4 个百分点，NPU 计算效率提升了 85.7 百分点。移动端 CPU 和NPU 上改进模型检测速度分别为 0.239 和 0.018 s 的，高于其他模型。

3.2　模型应用测试

为验证改进模型的实际表现，本文在贵阳市花溪区

某君子兰种植基地进行了实际应用测试，测试图片包含逆光拍摄和正侧面角度拍摄。

模型实际部署运行结果如图 10 所示。对于常见君子兰病虫害，改进模型均表现出较好的检测效果，虽然对于小目标的介壳虫在复杂环境下存在着漏检的问题，但是已经能达到了实际使用需求效果。

a. 正面拍摄 b. 逆光拍摄 c. 右侧面拍摄
a. Frontal shooting b. Backlight shooting c. Right side shooting

图 10 模型应用测试结果

Fig.10 Model application test results

4 结 论

本文提出了一种适用于移动端运行的君子兰病虫害检测方法 YOLO V4-TLite，主要结论如下：

1）本文提出的 YOLO V4-TLite 模型具有全新采样层，低成本的部分卷积和轻量化的主干。在桌面端上，改进模型的计算效率提升了 37.5 个百分点，在移动端上，CPU 计算效率提升了 7.7 个百分点，NPU 计算效率提升了 85.9 百分点。

2）本文提出的 YOLO V4-TLite 模型移动端推理时间为 0.018 s。较快的推理速度和较低的功耗，为君子兰的病虫害实时识别提供了支撑。

[参 考 文 献]

[1] 王冲，纪艺，王鸣谦，等君子兰属植物种及品种分类研究[J]. 植物遗传资源学报，2023，24（3）：692-700.
WANG Chong, JI Yi, WANG Mingqian, et al. Study on taxonomy of clivia species and cultivars[J]. Journal of Plant Genetic Resources, 2023, 24(3): 692-700. (in Chinese with English abstract).

[2] OMORUYI S I, DELPORT J, KANGWA T S, et al. In vitro neuroprotective potential of Clivia miniata and Nerine humilis (Amaryllidaceae) in MPP+-induced neuronal toxicity in SH-SY5Y neuroblastoma cells[J]. South African Journal of Botany, 2021, 136: 110-117.

[3] FU L, ZHENG Y, ZHANG P, et al. Development of an electrochemical biosensor for phylogenetic analysis of Amaryllidaceae based on the enhanced electrochemical fingerprint recorded from plant tissue[J]. Biosensors and Bioelectronics, 2020, 159: 112212.

[4] 周桂红，马帅，梁芳芳. 基于改进 YOLOv4 模型的全景图像苹果识别[J]. 农业工程学报，2022，38(21)：159-168.
ZHOU Guihong, MA Shuai, LIANG Fangfang. Recognition of the apple in panoramic images based on improved YOLOv4 model[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2022, 38(21): 159-168. (in Chinese with English abstract).

[5] ZHENG Z, XIONG J, LIN H, et al. A method of green citrus detection in natural environments using a deep convolutional neural network[J]. Frontiers in Plant Science, 2021, 12: 705737.

[6] XU Y, CHEN Q, KONG S, et al. Real-time object detection method of melon leaf diseases under complex background in greenhouse[J]. Journal of Real-Time Image Process, 2022, 19(5): 985-995.

[7] HAN S, LIU X, MAO H, et al. EIE: Efficient inference engine on compressed deep neural network[J]. ACM SIGARCH Computer Architecture News, 2016, 44(3): 243-254.

[8] LU Y, ZHANG L, XIE W. YOLO-compact: An efficient YOLO network for single category real-time object detection[C]//2020 Chinese Control and Decision Conference (CCDC). Hefei, China: IEEE, 2020: 1931-1936.

[9] CHOUDHARY T, MISHRA V, GOSWAMI A, et al. A comprehensive survey on model compression and acceleration[J]. Artificial Intelligence Review, 2020, 53: 5113-5155.

[10] LYU S, LI R, ZHAO Y, et al. Green citrus detection and counting in orchards based on YOLOv5-CS and AI edge system[J]. Sensors, 2022, 22(2): 576.

[11] XU L, WANG Y, SHI X, et al. Real-time and accurate detection of citrus in complex scenes based on HPL-YOLOv4[J]. Computers and Electronics in Agriculture, 2023, 205: 107590.

[12] 范天浩，顾寄南，王文波，等. 基于改进 YOLOv5s 的轻量化金银花识别方法[J]. 农业工程学报，2023，39(11)：192-200.
FAN Tianhao, GU Jinan, WANG Wenbo, et al. Lightweight honeysuckle recognition method based on improved YOLOv5s[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2023, 39(11): 192-200. (in Chinese with English abstract).

[13] 宋浩. 君子兰病虫害诊断模型构建及智能专家系统实现[D]. 天津：天津理工大学，2016：28-34.
SONG Hao. Clivia Plant Diseases and Pests' Diagnosis Model Building and the Realization of Intelligent Expert System [D]. Tianjin: Tianjin University of Technology, 2016: 28-34. (in Chinese with English abstract).

[14] 王卫星，刘泽乾，高鹏，等. 基于改进 YOLO v4 的荔枝病虫害检测模型[J]. 农业机械学报，2023，54(5)：227-235.
WANG Weixing, LIU Zeqian, GAO Peng, et al. Detection of litchi diseases and insect pests based on improved YOLO v4 Model[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(5): 227-235. (in Chinese with English abstract).

[15] LIU C, ZHU H, GUO W, et al. EFDet: An efficient detection method for cucumber disease under natural complex environments[J]. Computers and Electronics in Agriculture, 2021, 189: 106378.

[16] ZENG T, LI S, SONG Q, et al. Lightweight tomato real-time detection method based on improved YOLO and mobile deployment[J]. Computers and Electronics in Agriculture, 2023, 205: 107625.

[17] LUO Y, CAI X, QI J, et al. FPGA–accelerated CNN for real-time plant disease identification[J]. Computers and Electronics in Agriculture, 2023, 207: 107715.

[18] XU W, WANG R. ALAD-YOLO: An lightweight and accurate detector for apple leaf diseases[J]. Frontiers in Plant Science, 2023, 14: 1204569.

[19] 刘忠，卢安舸，崔浩，等. 基于改进 YOLOv8 的轻量化荷叶病虫害检测模型[J]. 农业工程学报，2024，40(19)：1-9.
LIU Zhong, LU Ange, CUI Hao, et al. A lightweight lotus leaf diseases and pests detection model using improved YOLOv8[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2024, 40(19): 1-9. (in Chinese with English abstract).

[20] LIU Z, ABEYRATHNA R M R D, SAMPURNO R M, et al.

Faster-YOLO-AP: A lightweight apple detection algorithm based on improved YOLOv8 with a new efficient PDWConv in orchard[J]. Computers and Electronics in Agriculture, 2024, 223: 109118.

[21] DEGADWALA S, VYAS D, CHAKRABORTY U, et al. Yolo-v4 deep learning model for medical face mask detection[C]//2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). Coimbatore IEEE, 2021 209-213.

[22] 王慧霭，王传旭，刘豪，等. 基于双分支通道空间依赖和非对称权重共享卷积的目标检测优化结构[J]. 计算机应用研究，2023，40(5)：1565-1570.
WANG Huiru, WANG Chuanxu, LIU Hao, et al. Dual channel spatial interdependent and asymmetric weight-sharing convolution for object detection[J]. Application Research of Computers, 2023, 40(5): 1565-1570. (in Chinese with English abstract).

[23] CHEN J, KAO S H, HE H, et al. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks [C]// 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada: IEEE, 2023: 1201-12031.

[24] SANDLER M, HOWARD A, ZHU M, et al. MobileNetV2: inverted residuals and linear bottlenecks [C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA: IEEE, 2018: 4510-4520.

[25] WANG C Y, BOCHKOVSKIY A, LIAO H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern. Vancouver, BC, Canada: IEEE, 2023: 7464-7475.

[26] HAN K, WANG Y, TIAN Q, et al. GhostNet: more features from cheap operations. [C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA: IEEE, 2020: 1577-1586.

[27] WANG A, CHEN H, LIU L, et al. YOLOv10: Real-Time End-to-End Object Detection[EB/OL]. (2024-05-23) [2025-2-13]. https://arxiv.org/abs/2405.14458

[28] KHANAM R, HUSSAIN M. Yolov11: An overview of the key architectural enhancements [EB/OL]. (2024-10-23) [2025-2-13]. https://arxiv.org/abs/2410.17725

Detection method for Clivia Miniata pests and diseases on mobile terminal based on YOLO V4-TLite

SONG Zhiwen[1] , LI Wei[2] , TAN Wei[3] , QIN Tao[1] , LIU Jie[4] , YANG Jing[1,5※]

(1. *Electrical Engineering College*, *Guizhou University*, *Guiyang* 550025, *China*;　2. *Agricultural College*, *Guizhou University*, *Guiyang* 550025, *China*;　3. *Forestry College*, *Guizhou University*, *Guiyang* 550025, *China*;　4. *China Electric Construction Group Guizhou Engineering Company Limited*, *Guiyang* 550025;　5. *Guizhou Provincial Key Laboratory of Internet + Intelligent Manufacturing*, *Guiyang* 550025, *China*)

Abstract: An accurate and rapid identification is required for the Clivia miniata pests and diseases in the greenhouse and garden in recent years. In this study, the YOLO V4-TLite algorithm was proposed to detect the Clivia miniata pest using a mobile terminal. The real-time performance and high accuracy were also achieved to reduce the over-reliance on the high-computing and high-power hardware. Firstly, the dataset of Clivia diseases and insect pests was collected from the real planting in the greenhouse. The images were captured at the early and middle stages of Clivia diseases and insect pests in winter and spring. Secondly, a low-cost improved partial convolution was used to replace the traditional one in the backbone network using the YOLO V4-Tiny model. The improved model was then obtained with the high speed of operation and the low consumption of memory. Thirdly, an improved structure of the inverse residual network was used to form a lightweight backbone network. The hardware compatibility was also enhanced to reduce the large consumption of random storage in the depth of the backbone network in the YOLO V4-Tiny model. The high operation speed of the model was obtained with the compatibility of the mobile terminal with the limited resources. Fourthly, the weight-sharing convolution was combined with the conventional convolution for channel fusion. The high robustness and accuracy of the network were obtained to reduce the redundant feature maps and their attention distraction in the traditional convolution layer of the YOLO V4-Tiny model. Finally, the improved model was deployed on the ROCK 5B mobile. Three types of Clivia miniata pests were then tested: leaf blight, maculopathy, and coccid. The experimental results showed that the better performance of the improved model was achieved with the mean average precision (mAP) of 78.5% at an intersection over union (IoU) ratio of 0.5, memory usage of only 4.8MB, and the floating point operations (FLOPs) of 1.3 G. The desktop single detection speed was 0.005 s with 70 W power consumption. On the mobile side, the CPU single detection speed was 0.239 s with 10 W power consumption. The NPU single detection speed was 0.018 s with 7 W power consumption. Compared with the original YOLO V4-Tiny model, the mAP50 of the YOLO V4-TLite model increased by 12.6 percentage point, whereas, the model size decreased by 78.6%. The computational efficiencies of the YOLO V4-TLite model were improved by 37.5 and 85.9 percentage point on the desktop and mobile side, respectively. While the power consumption demands were reduced by 26 and 2 W, respectively. The mAP50 values were 3.9, 2.3, 1.6, and 1.3 percentage point higher, respectively, compared with the target detection models of YOLOV11-N, YOLO V10-N, YOLO V7-Tiny, and YOLO V5-S. The YOLO V4-TLite model can be expected to detect the Clivia Miniata pest and disease on the low resource and power mobile. Better performance was also achieved, compared with the existing mainstream YOLO series models.

Keywords: Clivia miniata; pest and disease; YOLO V4-Tiny; lightweight; mobile deployment