

基于嵌入式深度神经网络的杂草识别模型

曾 翊, 张积焯, 余 箫, 余吉雅, 张艳超*

浙江理工大学信息科学与工程学院, 浙江 杭州

收稿日期: 2023年8月14日; 录用日期: 2023年10月5日; 发布日期: 2023年10月16日

摘 要

杂草的生存能力强, 对环境要求低, 分布广泛且随机, 必将争夺农作物所需的水分和营养等生存条件, 导致农作物的质量和产量有一定程度的降低。为解决以上问题, 本文设计了一种基于嵌入式深度神经网络的杂草识别系统, 使用嵌入式设备查看农田的杂草分布情况并检测出杂草的具体位置和种类, 此时就可以有针对性地去掉杂草, 提高农作物质量和产量, 节约劳动成本, 同时保护生态环境。该系统选择YOLO系列算法进行模型训练, 并通过模型转换获得合适的模型部署在嵌入式设备上目标检测。具体来讲, 先筛选足够数量的杂草图像样本, 经过手动标记获得对应标签文件, 从而获得数据集; 而后使用数据集模型训练, 本文算法为YOLOv4、YOLOv4-tiny和YOLOx算法, 得到PTH模型; PTH模型加载在PC端, PTH模型经模型转换生成TRT模型部署在嵌入式端Jetson Nano上。实验结果表明, 采用F1值和mAP值进行模型准确度评估时, YOLOx最高, 准确度为80.48%以上; 采用计数准确度对目标检测正确率进行评估时, 该指标下三个算法均在94%以上; 采用单张处理时间进行速度评估时, YOLOv4-tiny作为轻量化网络仅需0.0068 s, 嵌入式端仅需0.0453 s, 相当于实时检测。

关键词

卷积神经网络, 目标检测算法, YOLO, 嵌入式设备, 杂草识别

Weed Recognition Model Based on Embedded Deep Neural Network

Yi Zeng, Jiye Zhang, Xiao Yu, Jiya Yu, Yanchao Zhang*

School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou Zhejiang

Received: Aug. 14th, 2023; accepted: Oct. 5th, 2023; published: Oct. 16th, 2023

Abstract

The strong survival ability, low environmental requirements, widespread and random distribu-

*通讯作者。

tion of weeds will inevitably compete for the living conditions such as water and nutrients needed by crops, leading to a certain degree of reduction in crop quality and yield. To solve these problems, this paper designs a weed recognition system based on embedded deep neural network. The system uses embedded devices to view the distribution of weeds in farmland, detect the specific location and species of weeds, and remove them in a targeted manner, improving crop quality and yield, saving labor costs, and protecting the ecological environment. This system selects the YOLO series algorithm for model training and deploys the appropriate model on embedded devices for target detection through model conversion. Specifically, a sufficient number of weed image samples are first selected and corresponding label files are obtained through manual labeling to obtain the dataset. Then, the dataset is used for model training. The algorithms used in this paper are YOLOv4, YOLOv4-tiny, and YOLOx, and the PTH model is obtained. The PTH model is loaded on the PC side, and the TRT model is generated through model conversion and deployed on the embedded Jetson Nano. The experimental results indicate that when evaluating model accuracy using F1 score and mAP (mean Average Precision), YOLOx achieves the highest accuracy, surpassing 80.48%. When evaluating object detection accuracy using counting accuracy, all three algorithms perform above 94% in this metric. For speed evaluation based on single-frame processing time, YOLOv4-tiny, as a lightweight network, requires only 0.0068 seconds, and on the embedded platform, it takes only 0.0453 seconds, equivalent to real-time detection.

Keywords

Convolutional Neural Network, Object Detection Algorithm, YOLO, Embedded Device, Weed Identification

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

深度神经网络是深度学习中的关键技术，在机器学习、语音识别、文本分析、自然语言处理、等领域应用广泛[1] [2] [3]，有些训练后智能机器的认知水平已经出现突破人类水平[4]。如今，我国的农业正不断向着更精准、更节约、更规模化的方向发展，对于具有自动化属性的智能农业设备的需求量也在不断升高[5]，深度学习和机器视觉在农业自动化方面取得了巨大的成就。例如，Chen 等人通过结合了立体视觉和同步定位与建图系统获得了一张果园的详细 3D 图。其中的轮廓信息提取并匹配后，会通过变换得到进一步候选答案，使用分类器进行训练，这一实验对很多类型的水果有较好的效果[6]；Gao 基于 YOLOv3-tiny 网络对甜菜田杂草进行研究以便于提高检测速度[7]；Liu 等人提出了一种基于 MobileNetv2-YOLOv3 模型的番茄叶斑辨认措施[8]，平衡番茄灰斑的准确度和速度；Mazzia 使用两台设备进行基准测试，以便于果园中苹果的实时监测和产量评估[9]。

庞大的深度神经网络所需环境不适合于生活场景，如何提升目标检测、图形分割等技术的性能指标，如何安全简易地部署在生活化设备中[10]，将是研究的必然方向。例如，Peng K 等人于 2019 年利用深度卷积神经网络对模拟电路进行故障排查及诊断[11]，提高其在实际应用场景中的性能，不断升级算法模型，使其在终端设备上的应用将更加广泛且精密，具备良好前景以及经济效益。

目前已有大量深度神经网络及其在边缘设备、边缘计算的农业应用，但关于如何在保证精确度的前提下提高效率的研究仍不理想。我国杂草众多且分布广泛，其会汲取农作物所需水分和营养等生存条件，

造成农作物质量降低且产量下降[12]。本研究基于 YOLO 各版本的目标检测算法进行杂草识别，不同模型呈现的数据对日后的智能杂草识别有参考意义，将模型部署在嵌入式设备上，同时获得相应数据进行评估，能更好地还原现实应用场景，帮助解决深度学习在农业应用的各种难题，如防止高低不齐的杂草遮挡农作物或绿色植被、精准识别解决机械去除无法实现的问题，监测固定区域内杂草的生长数量速度等情况，减少消耗的劳动力等，这将对环境保护和农作物产量都有着十分重要的意义。

2. 方法介绍

2.1. YOLOv4 算法

YOLOv4 算法在保持准确度的同时引入性能更优或更符合检测处理的函数和算法，提高算法计算效率。YOLO 系列算法都可以很形象地分为 backbone 脊柱、neck 颈部和 head 头部三部分，分别进行特征提取、特征融合增强以及目标预测，简化后的网络架构图如图 1 所示。

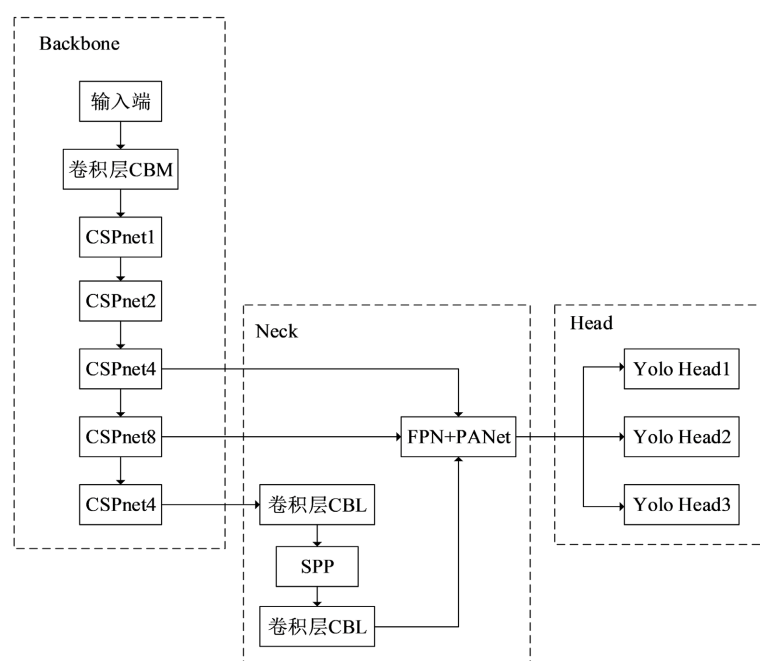


Figure 1. YOLOv4 Network architecture diagram

图 1. YOLOv4 网络架构图

YOLOv4 采用 CSPDarknet53 作为主干网络，通过 CSPnet 改变了残差块的堆叠方式，提高了网络重用特征的能力并减小了模型尺寸。Neck 部分包括 SPP 和 FPN + PAN 两个部分，SPP 结构进行最大池化增加感受野，FPN + PAN 结构反复提取特征以传达强语义特征和强定位特征。Head 部分利用前两部分的特征信息进行目标检测，采用 DIoU NMS 方法筛选出最优的预测框。损失函数由 box 位置损失、置信度损失、类别损失三部分构成，其中位置损失采用 CIOU 计算，置信度和类别损失采用二元交叉熵计算。

2.2. YOLOv4-Tiny 算法

YOLOv4-tiny 在 YOLOv4 基础上做减法，这种模型尺度的缩放提高了网络速度，能够适配在更多应用场景。但网络的学习能力会下降，更加适合嵌入式设备。该网络依然分三个部分，构架图如图 2 所示。

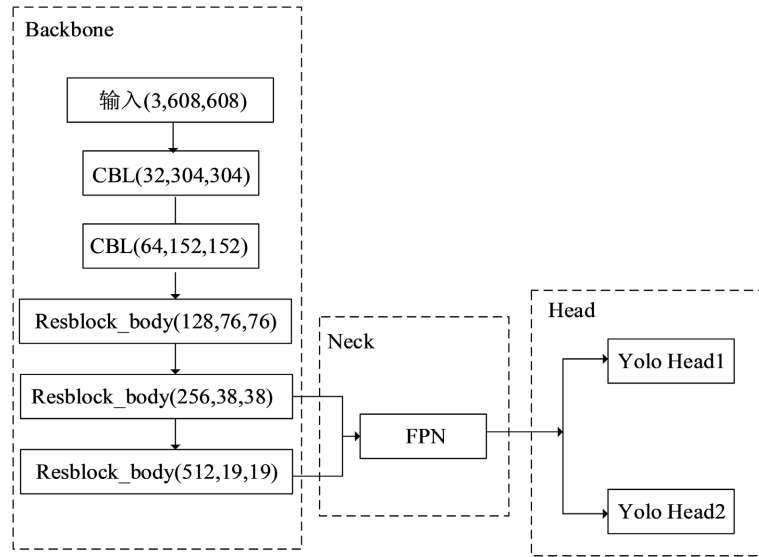


Figure 2. YOLOv4-tiny Network architecture diagram
图 2. YOLOv4-tiny 网络架构图

YOLOv4-tiny, 主干网络使用 3 个 DBL 和 3 个残差结构, 激活函数由 Mish 函数变更为 Leaky Relu 函数, 残差结构数量由 5 个变更为 3 个。特征融合部分只使用 FPN 特征金字塔, 其中 X1 和 X2 是最后两个残差结构输出的有效特征层。检测头仍然是 YOLO Head, 但只使用了两个有效特征层。与 YOLOv4 相比, YOLOv4-tiny 参数减少了 10 倍, 速度提高了 6~8 倍。但是, 简化主干网络并只进行下采样可能导致小目标或覆盖目标的漏检或错检, 需要在训练和推理时进行参数微调以确保有效信息的传递。

2.3. YOLOx 算法

考虑到本文需应用于终端设备进行边缘计算, 选择 YOLOx 参数量最少的 YOLOx-s 算法。算法仍然分为三部分, 与 YOLOv4 相比, 各部分都有不同程度的升级, 同时提高了算法的准确度和速度, YOLOx 算法简化架构图如图 3 所示。

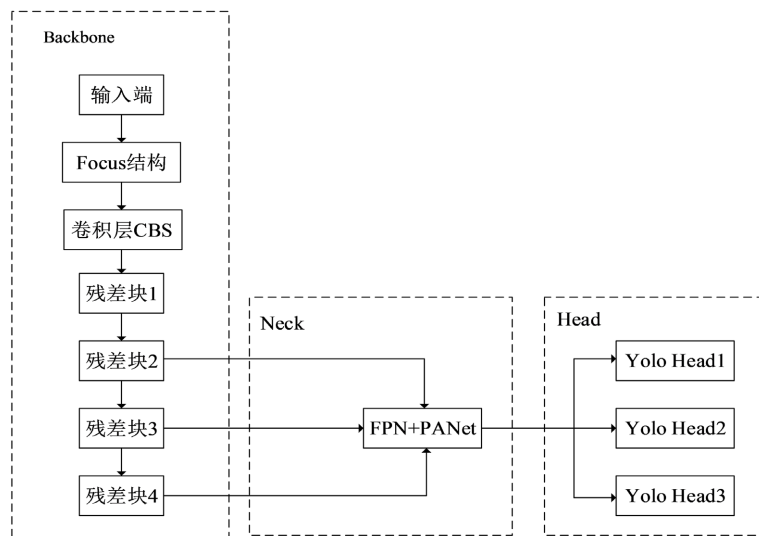


Figure 3. The YOLOx algorithm simplifies the structure diagram
图 3. YOLOx 算法简化结构图

YOLOx-s 网络的主干网络采用了 CSPDarknet 结构,并加入了 Focus 结构和 SPP 结构。CSPDarknet 网络包括三个部分: Focus、CSPnet 和 SPP 结构。其中, Focus 结构通过改变特征层获取的方式来扩大通道数, CSPnet 中的卷积层从 CBM 变为 CBS, 采用了 SiLU 函数代替 Mish 函数。SPP 结构在第四个残差块中使用,将其纳入主干网络中。在 Head 的结构上, YOLOx 采用了与之前版本不同的 YOLO Head, 分别进行分类和定位最后整合。YOLOx 采用 SimOTA 方法筛选正样本, 计算预测框与真实框的 IoU 并对位置进行判断, 最后计算代价矩阵。选取代价最低的 K 个点作为正样本, 其他点作为负样本, 然后用于计算损失函数, Reg 部分直接计算 IoU Loss, Obj 部分采用置信度损失算法, Cls 部分采用类别损失算法。

3. 基于嵌入式深度神经网络的杂草识别系统设计

基于嵌入式深度神经网络的杂草识别系统目的在于训练模型区分农作物和特定类别的杂草, 采集一定量的图像经标记后输入模型训练, 并将训练得到的模型以合适的格式部署在嵌入式设备上。系统可以分为两个部分, 分别为模型训练和模型部署, 系统流程图如图 4 所示。

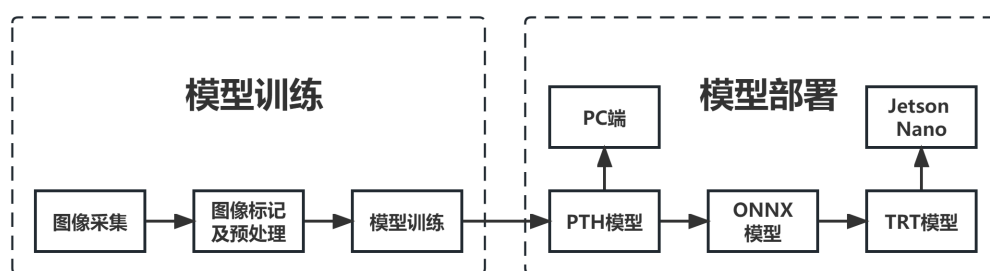


Figure 4. System flow chart
图 4. 系统流程图

3.1. 模型训练

3.1.1. 数据采集

本系统数据标记所需的数据集由 1118 个 .jpg 格式的图像文件以及 7893 个 .xml 格式的手动标记文件, 分别存放在两文件夹中。

其中, 图像包含 6 种农作物和 8 种杂草, 在手动标记过程中杂草被统一分类为 weed。农作物为甜菜根、胡萝卜、西葫芦、南瓜、萝卜和黑萝卜, 杂草为藜、八仙草、遏蓝菜、芥菜、野甘菊、野荞麦、野堇菜和牛膝菊。地点在拉脱维亚耶尔加瓦生命科学与技术大学植物保护研究科学研究所, 图像采用佳能 EOS 800D 拍摄, 相机镜头与植被表面保持 30 厘米距离, 从植被发育的第一阶段开始每日拍摄一次。

3.1.2. 数据标记

使用 LabelIng 软件进行手动标记, 在 Anaconda 命令提示符中输入 LabelIng 打开软件, 选择图像所在文件夹作为输入文件夹, 新建 JPEGImages 文件夹作为储存文件夹, 图像保存为 .xml 格式, 与 VOC 数据集使用规则兼容。

3.1.3. 迭代训练

将前两步中的图像和标签文件放置在 VOC2007 文件夹, 进行数据摆放。此外需要进行数据预处理。将图片文件分为训练集 train.txt、验证集 val.txt 和测试集 text.txt, 并保存在该路径的 ImageSets 文件夹下。

训练集用作模型训练, 验证集用于生成 2007_train.txt 和 2007_val.txt, 这也属于预处理, 测试集用于指标计算, 具体后文会有介绍。在训练开始前, 需要建立新的分类文件写入类别 weed, 同时设置为算法

中训练的分类路径 `classes_path`，最后引入权重文件，第一次训练可下载预训练后的权重文件，这会使用后的训练可引入上次训练中数据最好的权重文件，提高训练模型的目标检测能力。此外，需根据电脑显存允许的 `batch_size` 进行训练，迭代次数也可灵活改变。

3.2. 模型部署

经过训练后模型可在电脑上进行预测，新生成的 `.pth` 格式权重文件同理应用在预测的模型路径中。预测图像可先保存在 `img` 文件夹下，运行预测代码时输入图像路径来调用。但嵌入式设备应用 `PTH` 模型进行边缘计算时可能出现预测速度过慢甚至无法成功加载的情况，因此需要转化为 `TRT` 模型。

本系统采用嵌入式设备 `Jetson Nano`，实物图如图 5 所示。



Figure 5. Jetson Nano physical picture
图 5. Jetson Nano 实物图

设备由 `NVIDIA` 推出，体积小，功耗低至 `5 W`，可运行多种并行神经网络模型，支持 `TensorRT` 软件库和 `PyTorch` 框架，用于目标检测、图像分割和语音处理等。具体参数如表 1 所示。

Table 1. Jeston Nano performance parameter table
表 1. Jeston Nano 性能参数表

| | |
|--------------|---------------------------------------|
| GPU | 128 核 Maxwell 架构 GPU |
| CPU | 四核 ARM A57 @ 1.43 GHz |
| 内存 | 4 GB 64 位 LPDDR4 25.6 GB/s |
| 外部存储 | MicroSD 卡座(不含卡片) |
| 相机接入 | 2x MIPI CSI-2 DPHY lanes |
| Connectivity | Gigabit Ethernet, M.2 KeyE |
| 显示 | HDMI and display port |
| USB | 4x USB 3.0, USB 2.0 Micro-B |
| 其他 I/O | GPIO, PC, PS, SPI, UART |
| 机械特性 | 69 mm × 45 mm, 260-pin edge connector |

格式化 `SD` 卡后从官网下载镜像系统并使用 `balenaEtcher` 写入镜像。`NVIDIA` 提供环境的 `cuda` 版本为 `10.2`，使用 `Jetpack4.4` 安装 `PyTorch3.7`，并下载 `4.1.1` 版本 `opencv` 库以及 `7.1` 版本 `TensorRT` 库。

3.3. 系统指标

本文为评价训练模型引用 mAP 、 F_1 和 FPS 三个参数作为指标, 其中 mAP 、 F_1 可评估准确度, FPS 可以评估效率。

3.3.1. F_1

模型样本检测时会出现三种情况, TP 为正确检测的数量、 FP 为误检情况的数量、 TN 为错检情况的总数量、 FN 为漏检情况的数量。通过统计可以得到这四个数据并计算出以下 4 个参数, 公式如下:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

式中的精度 $Precision$ 能够表示真正的杂草样本在区域检测区域中被检测出的概率, 召回率 $Recall$ 能够表示检测的图像种杂草总数。准确度 $Accuracy$ 和 F_1 值都能对准确度进行评估, 本文采用后者, 值越接近 1 表示准确度越高。

3.3.2. mAP

模型某一类的准确度使用 AP 平均精度表示, 而所有类用 mAP 平均精度均值, 表达式如(5)和(6):

$$AP = \int_0^1 P(D) dR \quad (5)$$

$$mAP = \frac{\sum_{i=1}^N AP_i}{N} \quad (6)$$

式中, N 为需要识别的目标的所有类别总数。因为召回率和精度会呈现反相关趋势, 为兼顾两者引入 P-R 曲线, 召回率为横坐标, 精度为纵坐标, 面积为 AP 值。本文选择 mAP 值对准确度进行评估, 值同样越接近 1 越好。

3.3.3. 单张处理时间

模型能以一定速度检测图片, 每一秒检测的总数量就可以用 FPS 表示, 用于检测速度。单张处理时间 T 定义为 FPS 的倒数, 用于表示模型每处理一张图片所需要的时间, 同样可用于检测速度, 公式如下:

$$T = \frac{1}{FPS} \quad (7)$$

4. 系统实现及结果分析

基于嵌入式神经网络的杂草识别系统主要目的在于识别粮食作物种的杂草, 系统实现后可以模型的预测结果以及相应的性能指标。实验使用 YOLOv4、YOLOv4-tiny、YOLOx 算法进行模型训练和目标检测, 因此得到的数据结果能够进行横向对比, 观察不同版本的模型性能。

4.1. 系统实现

本文使用杂草数据集, 操作系统、图像处理器、处理器分别为 64 位 Windows10、RTX2060 和酷睿

i5-10300H8 核处理器。NVIDIA 提供的 CUDA 和 CUDNN 版本为 11.1 和 8.0.4。该实验以 PyTorch7 为训练框架，该框架简洁灵活，高效迅速且容易上手，适合用于构建深度学习网络。系统实物图如图 6 所示。



Figure 6. Physical system diagram
图 6. 系统实物图

4.2. 结果分析

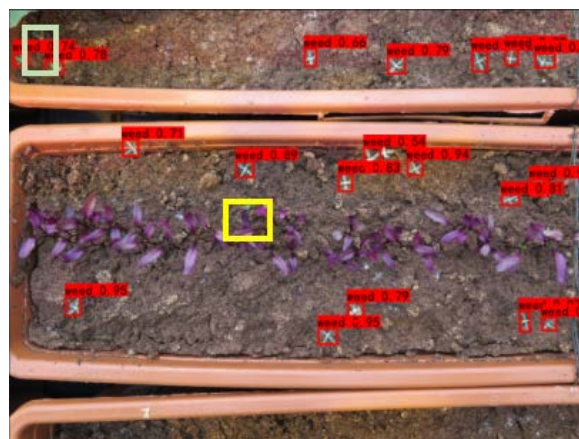
4.2.1. PTH 模型数据结果及分析

YOLOv4 算法训练时冻结阶段 batch size 设置为 4，解冻阶段 batch size 设置为 2，共训练 40 次，分别迭代 1 个 100 次和 39 个 400 次。YOLOv4-tiny 算法训练时冻结阶段 batch size 设置为 16，解冻阶段 batch size 设置为 8，共训练 40 次，分别迭代 1 个 100 次、25 个 400 次和 14 个 1000 次。YOLOx 算法训练时冻结阶段 batch size 设置为 16，解冻阶段 batch size 设置为 8，共训练 30 次，分别迭代 1 个 100 次和 29 个 300 次。室内和室外原始图像和预测结果分别如图 7、图 8 所示。该模型的数据集比较巧妙，将 8 种杂草统一标记为 weed，因为室内外图像中的杂草品种不同，因此本文各取一张图像作为输入图像。

三个室内图像目标检测中，YOLOv4-tiny 表现最优，预测框置信度基本为 1，能准确识别目标类别并定位，YOLOx 其次，能完整检测出该类目标但位置有偏差，YOLOv4 存在 2 处漏检分别如图黄框和绿框位置，其余目标置信度与 YOLOx 相似。



(a) 室内原始图像



(b) YOLOv4 PC 端

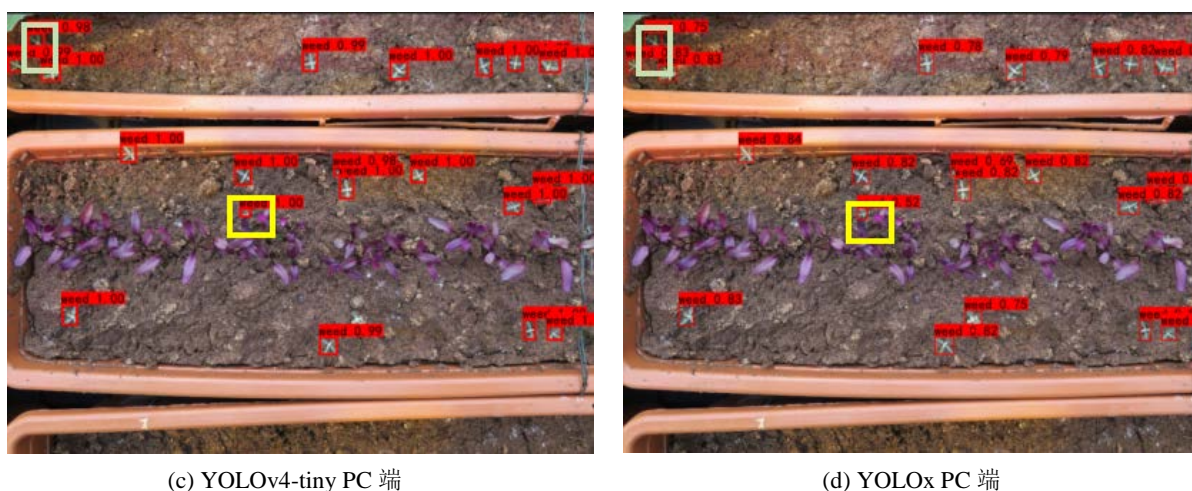


Figure 7. Comparison of indoor image target detection results of three models at PC end
图 7. 3 种模型 PC 端室内图像目标检测结果对比

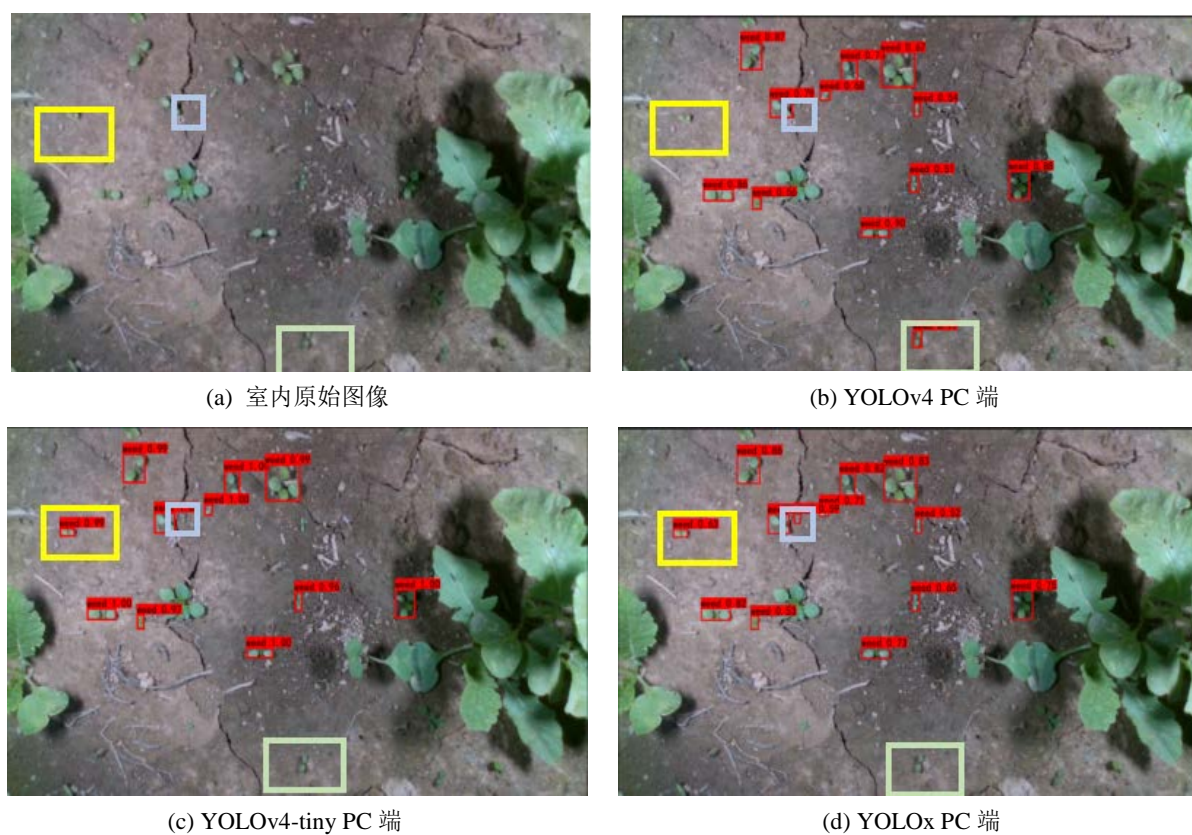


Figure 8. Comparison of outdoor image target detection results of three PC models
图 8. 3 种模型 PC 端室外图像目标检测结果对比

室外图像总体置信度比室内图像低，可能存在光线、背景颜色等因素。三个室外图像目标检测中，三个算法性能仍然为上图分析的顺序，YOLOv4 存在两处漏检分别如图蓝框和黄框标注，YOLOv4-tiny 两处漏检分别如图蓝框和绿框标注，YOLOx 存在一处漏检如图绿框标注，三个算法均能识别小目标、模糊目标和阴影中的目标。

1) 准确度评估

准确度使用 F_1 值和 mAP 值进行评估, 得到的模型准确度指标对比如表 2 所示。

Table 2. Comparison of accuracy of three models

表 2. 3 种模型准确度对比

| 网络 | F_1 | mAP (%) |
|-------------|-------|-----------|
| YOLOv4 | 0.68 | 74.07 |
| YOLOv4-tiny | 0.72 | 75.36 |
| YOLOx | 0.75 | 80.48 |

由表 2 可知, F_1 值和 mAP 值的结果趋势相同。YOLOx 模型 F_1 值为 0.75, 最接近 1; 其次是 YOLOv4-tiny 模型, F_1 值为 0.72 与 YOLOx 仅差 0.03, 准确度相近; 最后是 YOLOv4 模型的 F_1 值 0.68, 与最高值 0.75 相差了 0.07, F_1 值相对偏低。 F_1 值越接近 1 说明准确度越高, 因此通过 F_1 值对比后三种算法中 YOLOx 算法准确度最高。再观察 mAP 值, YOLOx 模型为 80.48%, 为三个模型中的最高值; YOLOv4-tiny 和 YOLOv4 模型的 mAP 值均在 75 左右, 分别为 75.36% 和 74.07%, 因此还有提高的空间。

结合两个指标进行评估可知, YOLOx 算法准确度都最高, YOLOv4-tiny 其次, YOLOv4 最低。但本实验训练过程的其他参数不能控制相同, 如由于电脑 CPU 限制, 三种算法的 `batch_size` 和迭代次数都不同, 因此需要更强大且精准的设备进行模型训练, 这是本次实验的不足之处需要改进。

2) 速度评估

速度使用单张处理时间进行评估。使用同一张照片, 得到的速度指标对比如表 3 所示。

Table 3. Speed comparison of three models

表 3. 3 种模型速度对比

| 网络 | 单张处理时间(s) |
|-------------|-----------|
| YOLOv4 | 0.0360 |
| YOLOv4-tiny | 0.0068 |
| YOLOx | 0.0209 |

由表 3 可知, 轻量化网络 YOLOv4-tiny 的单张处理时间最短, 较其他两者有很大优势, 仅 YOLOv4 算法的 20% 左右; 其次是 YOLOx 算法, 与前者相差 0.0141 s; 最后是 YOLOv4 算法, 在三个算法中表现最差。反观, 该数据有较大差异的原因有迹可循, YOLOv4-tiny 减少了主干网的层级结构, 同时将特征增强部分简化到仅有经典的 FPN 结构, 特征层和参数的减少使得其速度提高较多。此外, YOLOx 算法以 YOLOv5 算法为基础改进, 旨在保证准确度的基础上提高速度, 而 YOLOv5 算法以 YOLOv4 算法为基础改进, 主干网和预测头都有较大改变, 因此 YOLOx 算法速度更高是一个必然的结果。

3) 计数准确度评估

本训练集图片中共有 weed 类别目标 730 个, 不同算法正确检测的该类别目标数量以获取, 可进行计数准确度评估, 结果如表 4 所示。

由下表可知, 三个模型的计数准确度均在 90% 以上, 其中 YOLOx 模型最高, 为 97.81%; YOLOv4 适中, 为 96.85%, 与 YOLOx 差不到 1%; 最低的是 YOLOv4-tiny 模型, 计数准确度为 94.93%。可以看

出算法在检测过程中，能在保持正确性的同时，识别极大部分的目标，但和 F_1 、 mAP 值联合起来会发现识别的准确率仍然有很大调整空间。

Table 4. Comparison of counting accuracy indexes of three models

表 4. 3 种模型的计数准确度指标对比

| 网络 | 真实总数(个) | 检测计数结果(个) | 计数准确度(%) |
|-------------|---------|-----------|----------|
| YOLOv4 | | 707 | 96.85 |
| YOLOv4-tiny | 730 | 693 | 94.93 |
| YOLOx | | 714 | 97.81 |

4.2.2. TRT 模型数据结果及分析

YOLOv4 算法和 YOLOv4-tiny 算法的 PTH 模型转换至 TRT 模型后部署在 Jetson Nano 上，同样使用这两张图像作目标检测，室内、室外图像目标检测结果分别如图 9、图 10 所示，因 Jetson Nano 显示不清晰，因此图像保留较大尺寸。

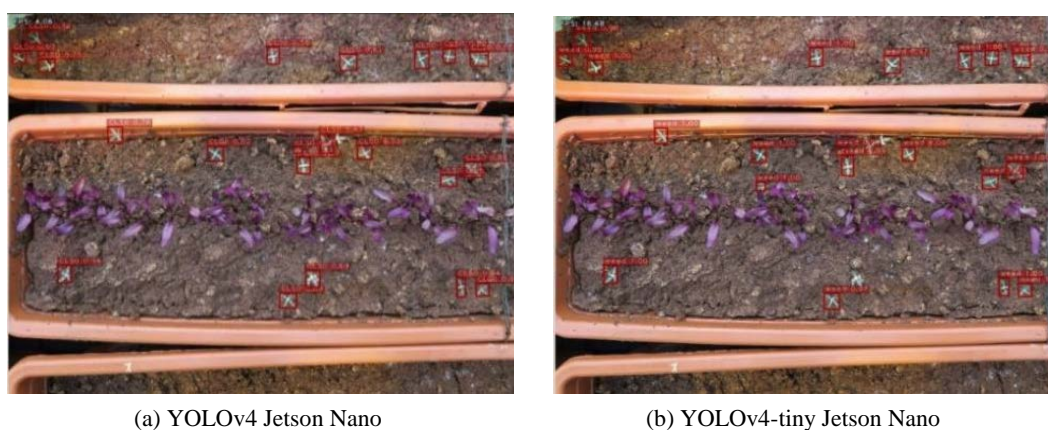


Figure 9. Comparison of indoor object detection results in Jetson Nano end of two models

图 9.2 种模型 Jetson Nano 端室内图像目标检测结果对比

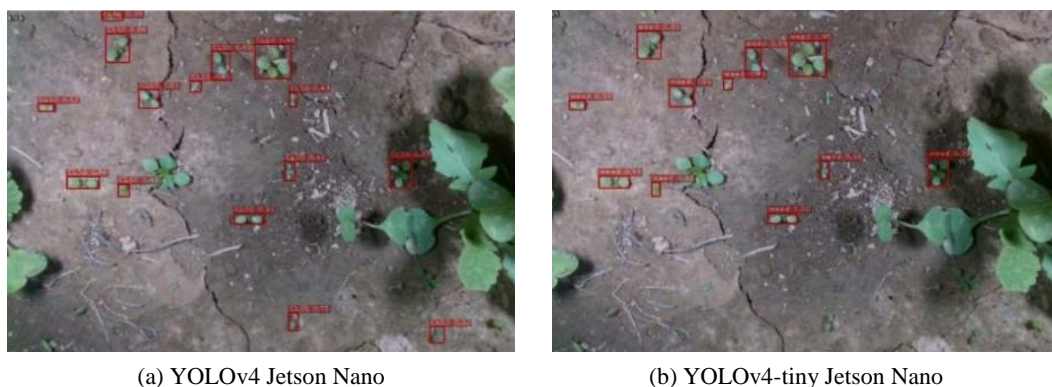


Figure 10. Comparison of outdoor image target detection results of Jetson Nano end between two models

图 10.2 种模型 Jetson Nano 端室外图像目标检测结果对比

部署在 Jetson Nano 上后，模型的整体识别效果表现没有 PC 端好，部分置信度降低，出现漏检情况。

其他指标在部署过程中不会有变化且已在上文对比，因此仅对比单张处理时间。PC 端与 Jetson Nano 速度指标对比如表 5 所示。

Table 5. Comparison of velocity indexes of two models
表 5.2 种模型速度指标对比

| 网络 | 单张处理时间(s) (PC) | 单张处理时间(s) (Nano) |
|-------------|----------------|------------------|
| YOLOv4 | 0.0360 | 0.2439 |
| YOLOv4-tiny | 0.0068 | 0.0453 |

由表 5 可知，YOLOv4 模型单张处理时间为 0.2439 秒，FPS 值为 4.1；YOLOv4-tiny 模型单张处理时间为 0.0453 秒，转换 FPS 值为 16.5，仍然领先 YOLOv4 模型，大约为其 4.0 倍，基本达到实时检测速度。该倍数与上文已计算的 PC 端 PTH 模型运算时的倍数相近。由此数据可知，TensorRT 作为加速度库能够从一定程度上提高模型速度，但嵌入式设备受硬件要求限制，GPU 配置受限又会降低速度，但相对速度保持稳定。

总体而言，在本次研究中，我们深入比较了三种目标检测算法(YOLOv4、YOLOv4-tiny 和 YOLOx)在不同环境及设备上的性能。从实验结果分析来看，各算法在速度和准确度方面展现出一定的优劣。

在速度方面，YOLOv4-tiny 在多个场景中表现出色，其单张处理时间最短，远优于其他两种算法。其轻量化结构使其在嵌入式设备上表现出相对较高的实时性能。其次是 YOLOx 算法，其基于 YOLOv5 改进的结构也带来了明显的速度提升，虽不及 YOLOv4-tiny，但仍在一定程度上实现了速度与准确度的平衡。然而，YOLOv4 在速度方面稍显不足，其相对较长的单张处理时间限制了其在实时应用中的应用。

在准确度方面，YOLOx 算法在 F1 值、mAP 值和计数准确度等多个指标上表现出色，稳定地领先于其他两种算法。YOLOv4-tiny 在计数准确度上稍有不足，而 YOLOv4 则在各项准确度指标上相对较低。此外，YOLOx 算法在室内和室外图像的各项检测任务中均展现出良好的综合性能。

综上所述，YOLOx 算法在速度与准确度的平衡方面表现优异，尤其适用于嵌入式设备。YOLOv4-tiny 虽速度最快，但在一些准确度指标上稍逊。YOLOv4 在准确度方面有提升空间，但速度限制了其实时应用。不同算法的优劣需要根据具体应用场景来选择，综合考虑速度和准确度的权衡。

5. 总结与展望

本文主要研究对象为 YOLO 系列算法，设计了一种基于嵌入式神经网络的杂草识别模型，通过模型训练获得有效模型进行目标检测，将模型部署在两个终端设备上能够更好的进行参数分析。本文经过实验，获得目标结果，并对准确度和速度进行横向和纵向评估。此外，本研究方法在其他目标检测领域的研究中也能够为算法选择提供更客观的评估标准。综上所述，本研究对于促进杂草识别技术发展以及在实际应用中推广目标检测算法具有重要意义和价值。

1) 提出将神经网络模型部署在使用率较高的嵌入式设备 Jetson Nano 上，促进人工智能和边缘计算结合，提供新思路，具有良好研究前景。

2) 提出将 YOLOv4、YOLOv4-tiny 和 YOLOx 算法分别部署在 PC 端和嵌入式设备，并进行算法指标对比，了解层级结构改进或删减对算法性能的影响，为后续算法选择提供理论支撑和数据参考。

通过实验数据可以看出，将神经网络部署在嵌入式设备时基本达到实时识别水平，但由于嵌入式设备受硬件要求限制，部署在其上的模型准确度及速度仍有较大提升空间，基于此，本文在此提

出对未来工作的展望：① 增加数据集照片数量，扩大样本数，提高系统鲁棒性；② 采用算力更强、GPU 配置更好的嵌入式设备，优化模型运行环境；③ 采用准确度更高、速度更快、结构更好的深度神经网络模型。

参考文献

- [1] Sze, V., Chen, Y.H., Yang, T.J. and Emer, J.S. (2017) Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE*, **105**, 2295-2329. <https://doi.org/10.1109/JPROC.2017.2761740>
- [2] Zhang, Y., Chan, W. and Jaitly, N. (2017) Very Deep Convolutional Networks for End-to-End Speech Recognition. 2017 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, 5-9 March 2017, 4845-4849. <https://doi.org/10.1109/ICASSP.2017.7953077>
- [3] 王欣, 雷珺, 李小欢, 汤成. 基于深度神经网络的智能交互式学习系统[J]. 电子设计工程, 2022, 30(22): 73-77.
- [4] 郭圣明, 贺筱媛, 胡晓峰, 吴琳, 欧微. 军用信息系统智能化的挑战与趋势[J]. 控制理论与应用, 2016, 33(12): 1562-1571.
- [5] 田娜, 杨晓文, 单东林, 吴继成. 我国数字农业现状与展望[J]. 中国农机化学报, 2019, 40(4): 210-213.
- [6] Lin, G., Tang, Y., Zou, X., Cheng, J. and Xiong, J. (2020) Fruit Detection in Natural Environment Using Partial Shape Matching and Probabilistic Hough Transform. *Precision Agriculture*, **21**, 160-177. <https://doi.org/10.1007/s11119-019-09662-w>
- [7] Gao, J., French, A.P., Pound, M.P., He, Y., Pridmore, T.P. and Pieters, J.G. (2020) Deep Convolutional Neural Networks for Image-Based Convolvulus Sepium Detection in Sugar Beet Fields. *Plant Methods*, **16**, 29. <https://doi.org/10.1186/s13007-020-00570-z>
- [8] Liu, J., and Wang, X. (2020) Early Recognition of Tomato Gray Leaf Spot Disease Based on MobileNetv2-YOLOv3 Mode. *Plant Methods*, **16**, 83. <https://doi.org/10.1186/s13007-020-00624-2>
- [9] Mazzia, V., Khaliq, A., Salvetti, F. and Chiaberge, M. (2020) Real-Time Apple Detection System Using Embedded Systems with Hardware Accelerators: An Edge AI Application. *IEEE Access*, **8**, 9102-9114. <https://doi.org/10.1109/ACCESS.2020.2964608>
- [10] 郭吉政. 免疫卷积神经网络并行优化及其嵌入式系统应用[D]: [硕士学位论文]. 上海: 东华大学, 2015.
- [11] Cass, S. (2020) Nvidia Makes It Easy to Embed AI: The Jetson Nano Packs a Lot of Machine-Learning Power into DIY Projects-[Hands on]. *IEEE Spectrum*, **57**, 14-16. <https://doi.org/10.1109/MSPEC.2020.9126102>
- [12] 李涛, 温广月, 钱振官, 田志慧, 沈国辉, 梁帝允. 不同类型杂草危害对小麦产量的影响[J]. 中国植保导刊, 2013, 33(4): 28-30.