

基于 Python 的农作物植株识别方法研究

朱可心

广东工商职业技术大学 广东 肇庆 526020

【摘要】：田间环境下采集的农作物图像通常存在较大的差异。不同拍摄时间和光照条件会影响植株的颜色和亮度，周围环境的变化也会使图像背景有所不同。与此同时，作物在不同生长阶段的形态并不相同，叶片发生遮挡后，一些原本较为明显的特征也可能被覆盖。针对这些问题，本文以 Python 开发环境为基础，对农作物植株识别过程中涉及的图像采集、预处理、目标检测及结果整理等环节进行分析。程序实现主要使用 OpenCV、NumPy 和 PyTorch，其中 OpenCV 用于图像读取和基本处理，NumPy 用于图像数据的运算，PyTorch 用于检测模型的加载与推理，各部分通过相应的数据格式进行连接。在图像预处理方面，结合田间图像的实际情况，对尺寸调整、滤波、光照和色彩处理以及数据增强等方法进行了讨论，重点说明不同方法的适用情况，未对具体参数进行深入比较。目标检测部分选取 YOLO 系列算法作为示例，介绍从植株类别判断、目标位置确定到检测结果输出的基本过程。由于目前尚未建立专用的作物图像数据集，也没有进行模型训练、参数调整及不同模型之间的实验比较，本文主要围绕识别流程及各技术环节之间的关系进行梳理，为后续开展具体作物识别实验提供参考。

【关键词】：Python；植株识别；机器视觉；图像处理；目标检测

DOI:10.12417/2811-0722.26.09.031

引言

农业生产中的许多田间信息其实可以通过人工观察获得，例如植株数量、作物分布和局部生长情况等。对于面积较小的地块，人工巡查和记录相对方便。但随着监测范围扩大，或者需要在不同时间进行多次观察，重复记录所花费的时间就会增加。不同人员在观察过程中也可能存在判断上的差异，尤其是在植株数量较多、目标之间相互遮挡或边界不明显时，最终得到的统计结果并不一定相同。采用图像采集后，现场情况可以以图像的形式保存下来。需要再次查看时，可以直接调用已有图像，也可以利用图像处理程序进行分析，这使田间观察得到的信息能够进一步保存和利用。

农田环境下获得的图像与实验室中采集的标准图像存在较大区别。天气、拍摄时间以及光照方向的变化都会影响图像的明暗和颜色，同一地块在不同时间拍摄时，植株呈现出的视觉效果也可能不同。另一方面，作物会随着生长阶段发生形态变化，幼苗期、开花期和成熟期的植株在大小、叶片数量及整体形态上均存在差异。种植密度较高时，叶片之间还会产生遮挡，土壤、杂草以及周围其他作物也会进入拍摄范围，使单株植株的完整轮廓难以获得。在较为简单的试验环境中，传统图像处理可以通过颜色、纹理和形状等特征，并结合一定的阈值条件完成目标提取。这类方法实现相对直接，但当拍摄环境发生变化时，原先设置的参数通常需要重新调整。Jiang 与 Li 从植物表型分析的角度对卷积神经网络的应用进行了总结，Yang 与 Xu 则对深度学习在园艺领域中的相关研究进行了梳理。这些研究表明，利用模型从图像中自动提取特征，已经成为农业视觉分析中较为常见的方法。

1 Python 开发环境与植株识别关键技术

在农作物植株识别系统中，Python 主要依托 OpenCV、

NumPy 和 PyTorch 等工具完成各项功能。OpenCV 提供图像读取、颜色空间转换、滤波去噪及目标标注等常用接口，可满足图像预处理和识别结果可视化的需求；NumPy 负责矩阵运算和数据处理，为图像计算提供基础支持；PyTorch 则用于构建和调用深度学习模型，完成模型训练、推理及参数优化等任务。各类开发工具之间兼容性较好，能够实现从数据处理到模型部署的完整开发流程。

图像预处理是植株识别的重要环节，其作用在于改善图像质量，减少复杂环境对识别结果的干扰。农业生产环境中常存在光照变化、背景复杂及植株遮挡等情况，容易影响目标特征的提取。因此，在模型识别之前，通常需要对图像进行尺寸统一、颜色转换、滤波去噪和图像增强等处理，以突出植株目标特征，提高后续识别的稳定性。

当前，深度学习已成为农作物植株识别的主要技术路线。卷积神经网络能够自动学习植株图像中的颜色、纹理和形态特征，减少了传统人工特征设计带来的局限。近年来，YOLO 等目标检测算法在检测速度和识别精度之间取得了较好的平衡，被广泛应用于果实检测、病虫害识别、杂草识别以及农业机器人视觉导航等场景。借助 Python 开发环境，可快速完成模型训练、调用与部署，提高植株识别系统的开发效率，同时也为后续功能扩展和系统优化提供了便利。

2 农作物植株识别流程设计

识别从图像采集开始。不同设备得到的画面具有不同特点。固定摄像机位置基本不变，适合长时间观察某个区域；手机等移动终端使用灵活，可以随时改变拍摄位置，但距离和角度很难完全一致。农业机器人更接近作物，比较容易拍到叶片等细节，同时也可能因为距离较近而出现大面积遮挡。无人机则适合迅速覆盖较大的农田，得到的是较大范围的俯视影像，

不过单株植株在图像中的面积相对有限。实际选择设备时,关注点并不只是分辨率或者设备性能,还包括监测范围、目标尺度以及后面的使用方式。

图像采集完成以后,最先面对的是数据规格问题。同一批数据如果来自不同设备,分辨率、颜色通道甚至文件格式都可能存在差异。模型并不会自动消除这些区别,输入以前必须先完成统一。对于画面本身已经比较清晰的图像,没有必要再增加很多处理步骤;如果存在明显噪声,可以针对噪声进行处理;亮度偏低或局部反差不足时,再考虑亮度、对比度等调整。这里更重要的是先判断图像到底存在什么问题,而不是按照固定顺序把所有预处理全部执行一遍。

不同应用对结果的处理方式并不一样。单纯展示检测效果时,把检测框和类别名称画回原图即可;植株计数则要排除重复框和明显错误目标,否则简单累加检测框数量很容易产生偏差;连续田间监测还涉及采集时间、设备来源等信息,单张图片的识别结果需要与对应的时间或位置建立关系。Meraj等的研究也反映出,农业视觉任务会随着研究对象和应用条件变化而产生差异,程序输入并不适合完全固定。采集时记录设备型号、拍摄参数等信息虽然属于基础工作,却能在后续数据整理和问题回查时发挥作用。

程序结构也会影响后面的修改。把图像读取、模型推理和结果输出全部写进一个较大的程序,最初编写起来可能比较直接,后期替换模型或者修改输入来源时就会牵一发而动全身。将这些部分分别处理,数据在哪个环节发生变化也更容易看清。这样的划分并不是为了增加程序复杂度,而是让不同功能之间少一些相互牵连。

3 农作物植株图像预处理

图像预处理首先面对的是尺寸问题。不同相机或者不同采集平台产生的图像分辨率差别较大,输入模型时通常要重新调整。尺寸过小,叶片边缘和细小纹理可能被压缩掉;尺寸过大,计算量又会上升,实际运行速度受到影响。对于植株识别而言,保留下来的究竟是多大尺度的目标信息,并没有脱离任务的统一答案。另外,颜色和通道处理同样不能简单理解成“处理得越充分越好”。有些模型要求特定的颜色空间或者通道排列,数据格式不正确时,即使图像本身看起来没有异常,模型接收到的内容也可能已经发生变化。光照校正和色彩调整更适用于存在明显画面问题的图像。田间拍摄过程中,部分叶片处于阴影,部分被阳光直射,适当调整亮度与对比度有助于改善画面,但如果改变幅度过大,原本具有区分作用的颜色关系也可能被削弱。

其次,滤波处理主要针对噪声。均值、中值和高斯滤波在处理效果上各有差别,使用哪一种与噪声类型和图像特点有关。植株叶片的边缘、叶脉和局部纹理本身就是识别时可能利

用到的信息,如果滤波过强,噪声虽然降低了,细节也一起变得模糊。实际处理时,与其连续叠加多个滤波步骤,不如先确认噪声是否真的对识别造成影响。

当原始样本数量有限时,翻转、旋转、缩放等操作可以让训练图像产生更多变化,模型接触到的样本形式也会增加,但这些图像仍然建立在原始数据之上,并没有真正增加新的田间环境。比如不同生长阶段的植株、严重遮挡情况下的叶片以及特殊天气条件下的画面,都无法仅靠简单几何变换得到。因此,数据增强更适合看作真实数据的补充,而不是替代。

训练数据和测试数据的处理过程也不能随意变化。训练时如果采用一种尺寸、颜色或者归一化方式,测试阶段却换成另一套规则,最后得到的差异就可能来自输入处理,而不是模型自身。对于以后建立专用数据集而言,原始图像、处理方式以及处理后的数据最好分别保留,实验过程中出现异常时才有条件重新追溯。

4 农作物植株识别方法设计

田间图像往往同时包含多个植株目标,仅进行图像分类只能回答“图像中是什么”,却难以直接说明目标位于什么位置。因此,植株识别通常更适合采用目标检测方法。YOLO系列算法将类别判断和位置预测放在同一检测框架中,可以一次处理图像中的多个目标,比较符合田间多目标识别的需求。Badgujar等^[3]的综述也对YOLO在农业目标检测中的应用进行了归纳。本文以YOLO系列作为方法说明的基础,重点讨论其识别流程,不进一步比较不同版本之间的性能差异。

模型推理时,原始图像先转换成对应的数据格式,随后经过计算得到检测框坐标、类别和置信度等结果。置信度反映模型对当前预测的判断程度,并不能单独作为识别正确与否的依据。阈值设置也会改变最终保留下来的检测目标:阈值较低时,保留的目标增多,同时可能出现更多误检;阈值提高以后,一些受到遮挡或特征不完整的植株又可能被过滤掉。具体阈值需要结合数据和实验结果确定,不能仅依赖经验给出一个固定值。

模型得到检测结果后,还需要根据任务进行筛选和整理。对于可视化任务,检测框和标签可以直接绘制在原图上;计数任务则关注重复目标和无效目标;监测任务还需要进一步保存时间、来源等信息。同一个检测模型经过不同的结果处理,可以服务于不同的农业应用。

模型运行还受到硬件条件影响。实验室计算机通常能够提供较充足的计算资源,而无人机、农业机器人等边缘设备在算力、存储和功耗方面存在限制。较大的模型虽然可能具有更强的特征表达能力,但推理时间和运行资源也会随之增加;模型过度压缩则可能牺牲部分识别效果,因此轻量化网络和边缘计算在实际部署中具有一定研究空间。现阶段未进行具体部署实

验,相关讨论主要停留在模型与设备之间的适配关系上。

5 系统应用分析

植株识别真正进入农业场景以后,模型输出的检测框只是一个中间结果。对于生长监测来说,更有价值的是不同时间得到的数据能否对应起来;用于植株计数时,检测框数量还不能直接等同于真实株数;如果与无人机或农机系统结合,目标位置可能还要转换成更适合后续作业的数据形式。由此来看,识别模型只是整个系统的一部分,前面的图像采集和后面的数据应用同样影响最终效果。

田间环境变化是模型应用中比较现实的问题。不同品种的叶片大小、颜色和形态存在差异,同一品种在幼苗期和成熟期也不会保持完全相同的外观。植株长得越密,遮挡情况往往越明显;杂草进入画面以后,模型还要区分目标植株和背景植物。光照变化则可能让同一目标在不同图像中呈现不同亮度。不同农业任务对结果的要求也不一样,植株计数更关注漏检和重复检测,长期监测则更重视时间上的稳定性,实际作业还可能进一步关注目标位置。没有统一数据集和明确任务时,简单拿一个精度数字无法代表全部应用情况。

目前的研究尚未建立专属农作物数据集,也没有继续开展模型训练、参数比较及设备部署实验,因此重点仍然放在识别方法和流程的整理上。真正进入具体作物研究以后,可以从田

间图像采集入手,对不同生长阶段、天气条件和遮挡情况进行覆盖,再建立训练集与测试集。在这个基础上,预处理方式、模型参数以及硬件条件之间的差异才有机会通过实验表现出来。相较于单纯比较一个数字,这种过程更能说明算法为什么会产生差别,也更接近实际应用。

农作物植株识别是由多个环节共同组成的过程,检测模型只是其中的一部分。图片进入程序以后,涉及图像格式、尺寸、颜色通道以及模型输入等数据处理,检测完成后需根据具体任务整理输出结果。Python及相关开发工具主要用于环节之间的衔接,识别效果与训练数据、算法本身以及实际应用环境有关。

6 结语

本文针对基于Python的农作物植株识别方法进行了研究,对Python开发环境及关键技术、植株图像预处理、识别方法设计和系统应用进行了较为系统的分析。Python凭借丰富的开源生态和良好的开发环境,为农作物植株识别系统的构建提供了便利,在图像处理、模型部署及功能扩展等方面具有较好的应用价值。受研究内容所限,本文未结合具体实验数据对识别模型进行验证。后续可结合典型农作物数据集开展模型训练与性能测试,并融合边缘计算、轻量化网络等技术,不断提升植株识别系统的识别精度和环境适应能力,为智慧农业相关应用提供参考。

参考文献:

- [1] Jiang Y,Li C.Convolutional Neural Networks for Image-Based High-Throughput Plant Phenotyping:A Review[J].Plant Phenomics,2020,2020:4152816.
- [2] Yang B,Xu Y.Applications of deep-learning approaches in horticultural research:a review[J].Horticulture Research,2021,8:123.
- [3] Badgujar C M,Poulose A,Gan H.Agricultural Object Detection with You Only Look Once(YOLO)Algorithm:A Bibliometric and Systematic Literature Review[J].Computers and Electronics in Agriculture,2024.
- [4] Meraj T,Sharif M I,Raza M,et al.Computer vision-based plants phenotyping:A comprehensive survey[J].iScience,2024,27(1): 108709.
- [5] Zhao C,Zhang Y,Du J,et al.Crop Phenomics:Current Status and Perspectives.Frontiers in Plant Science,2019,10:714.