西北工业大学

数字图像处理–论文翻译

原论文标题：AI-Driven Computer Vision Detection of Cotton in Corn Fields
Using UAS Remote Sensing Data and Spot-Spray Application

涂佳恒

计算机学院

计算机科学与技术

2024 年 11 月

学号：2022302637

利用无人机遥感数据和点喷应用进行玉米地中棉花的 AI 驱动计算机视觉检测

摘要：为了有效防治棉田中棉铃象鼻虫（大棉铃象鼻虫）的再次侵扰，有必要解决在玉米（玉米属植物）和高粱（双色高粱属植物）等轮作作物中检测到志愿棉（棉属植物）植株的问题。目前的做法是在田边进行人工田间调查，这往往会导致遗漏在玉米和高粱中间生长的志愿棉植株。当这些志愿棉植株达到吐絮期（5-6片叶）时，它们可能会成为棉铃象鼻虫的寄主。因此，检测、定位并准确地对这些植株进行点喷处理至关重要。本文重点研究了YOLOv5m在玉米田吐絮期（VT）检测和定位志愿棉植株的应用。我们的研究结果表明，在大小为1207×923像素的图像上，志愿棉植株的平均精度（mAP）可达79%，交并率（IoU）为50%，分类准确率为78%。在 NVIDIA Tesla P100 GPU（16GB）上，平均检测推理速度为每秒 47 帧（FPS），在 NVIDIA Jetson TX2 GPU 上为每秒 0.4 帧，这突显了检测速度对实时应用可行性的相关性和影响。此外，我们展示了基于开发的计算机视觉（CV）算法，通过模拟定制的无人驾驶飞机系统（UAS）在点喷应用中的应用。这种基于 UAS 的方法能够在玉米地里近乎实时地检测和缓解杂草类植物，在 NVIDIA Tesla P100 GPU 上，“近乎实时”被定义为每帧约 0.02 秒，在 NVIDIA Jetson TX2 GPU 上为每帧约 2.5 秒，从而为控制棉铃象鼻虫提供了一种高效的管理解决方案。

关键词：棉铃象鼻虫；志愿棉（VC）；遥感；计算机视觉（CV）；YOLOv5；无人驾驶飞行器系统（UASs）；点喷

1. 引言

棉铃象鼻虫（Anthonomus *grandis* L.）是一种害虫，自 19 世纪 90 年代从墨西哥迁徙过来以来，已给美国棉花产业造成了超过 230 亿美元的损失[1]。尽管它已在美国大部分地区被根除，但它仍然是美国棉花产业关注的问题，尤其是在得克萨斯州。因此，根据日落顾问委员会的最新报告[2]，得克萨斯州棉铃象鼻虫根除基金会（TBWEF）的活动仍有持续的需求。该基金会已将该州划分为 16 个根除区，其中格兰德河谷下游（LRGV）仍在积极开展相关工作。

由于其热带气候条件[2]以及靠近墨西哥边境，该地区每年仍是棉铃象鼻虫重新侵扰最严重的区域。LRGV的脆弱性是由几个因素造成的，其中之一是其亚热带气候，夏季炎热潮湿，冬季温和凉爽，没有结冰条件，这使得棉花植株全年生长和结果，为棉铃象鼻虫提供了持续的食物来源。此外，来自墨西哥湾的飓风通常会将棉铃象鼻虫从墨西哥塔毛利帕斯州带到LRGV[3]。仅在2019年，该基金会就捕获了46,000只棉铃象鼻虫，表明问题的严重性以及TBWEF持续运作的必要性。如果没有TBWEF的灭虫努力，棉铃象鼻虫将给德克萨斯州的农民造成每年2亿美元的损失[4]。最近的报告表明，TBWEF的努力已显著减少了德克萨斯州的棉铃象鼻虫数量[5]。根据 TBWEF 发布的最新周报，从 2020 年到 2024 年，下里奥格兰德河谷捕获的棉铃象鼻虫的整个季节平均数量减少了约 99.91%[5]。

棉花（Gossypium *hirsutum* L.） 通常与玉米（Zea *mays* L.） 和高粱（Sorghum *bicolor* L.）等作物轮种。在像LRGV这样的气候地区，棉花种子可以全年存活，因此，在前一年收获期间可能掉落的棉花种子可以在今年的玉米和高粱植株中生长[3,6,7]。这种植物被称为自生棉花（VC）植物，本质上它们是杂草，可以在玉米、高粱、大豆和小麦等谷类作物的田地里出现。在州生产区域内，必须清除各种作物中的自生棉花，主要有两个原因：竞争导致的产量损失和棉铃象鼻虫防治的挑战。自生棉花植物与主要作物竞争，降低了产量，农民通常希望使用除草剂控制到80-90%。在南德克萨斯州和德克萨斯州东部，德克萨斯棉铃象鼻虫根除计划（TBWEP）正在进行中，自生棉花植物的存在显著增加了棉铃象鼻虫防治的复杂性和成本，因为棉铃象鼻虫以棉花植物为宿主。因此，德克萨斯州农业部对隔离区内可种植的非商业性棉花植株（6-8片叶子或更大）实行零容忍政策，以防止害虫侵袭。为了遵守这些规定并保护作物产量，农民必须通过及时和适当的除草剂应用来完全控制自生棉花。

为了最小化棉铃象鼻虫再次侵扰的可能性，TBWEF 使用信息素陷阱来检测棉铃象鼻虫，并使用农药将其消灭。作为 LRGV 地区棉铃象鼻虫缓解工作的一部分，每周都会检查有轮作作物的田地，看田边是否有 VC 植物。当检测到 VC 植物时，信息素陷阱的数量会增加。除了检查 VC 植物的存在外，还会检查信息素陷阱中是否有棉铃象鼻虫。如果在田边至少发现一只， 那么整个田地都会喷洒农药， 通常是马拉硫磷（C10H19O6PS2）超低容量（FYFANON®ULV AG），用量在 0.56 至 1.12 千克/公顷之间[8]。由于玉米和高粱田中间生长的 VC 植物未被检测到，因此无法单独喷洒。对整个田地进行喷洒会导致管理成本增加、环境问题以及许多有益昆虫的破坏。

如果能在玉米和高粱田间生长的薇甘菊植物被检测到并精确定位，从而能够部署具备点喷能力的无人驾驶飞机系统（UAS），就可以避免均匀喷雾。在薇甘菊植物到达小粒状阶段之前进行检测，并精确地用除草剂进行喷雾处理，可以消除这些植物，并最大限度地减少使用马拉硫磷的需求。然而，由于对除草剂的耐受性或检测不准确，一些薇甘菊植物可能会存活下来，在这种情况下，在季节后期可能需要应用马拉硫磷。为了在生长季的早期或晚期检测薇甘菊植物，可以使用由UAS收集的图像以及使用最先进的卷积神经网络（CNN）架构（如Mask R-CNN[9]、YOLOv3[3,10,11]、YOLOv5 [12]等）的计算机视觉（CV）算法进行遥感检测。自2021年发布以来，YOLOv5在CV应用中变得流行起来，并被用于检测

各种物体，如苹果[13]、口罩[14]、安全帽[15]等。由于与其他传统物体检测算法相比，YOLOv5具有更高的检测精度和更快的推理速度，因此在本研究中，YOLOv5被选为最可行的模型，用于近实时检测。最初，YOLOv5发布了四种不同的变体：YOLOv5s、YOLOv5m、YOLOv5l和YOLOv5x。然而，基于我们之前的研究[16]，YOLOv5m被选为理想的变体。

除了检测到VC植物外，还需要检测到植物的地理坐标，以便进行精确的喷雾应用。因此，本研究使用了带有地理标签的无人机图像。高质量的RGB（红色、绿色、蓝色）相机通常用于YOLOv3和YOLOv5的物体检测[13,14,17]。在大多数情况下，未进行辐射校正。未进行辐射校正的遥感图像容易受到各种环境条件的影响，包括光照、大气光散射、传感器噪声等[18]。未进行辐射校正的图像的数字编号（DN）无法代表实际的地表反射率[19-21]。因此，本研究进行了辐射校正。

YOLO 系列的目标检测算法会在图像中感兴趣的目标周围生成边界框（BBs）。这些边界框的位置是基于它们与图像左上角像素的距离[10,22,23]。基于边界框的坐标，可以确定每个边界框的中心坐标。像素坐标对于具有点喷能力的无人机系统的路径规划并无用处，因此有必要将像素坐标转换为 GPS 坐标，以便它们能够用于对检测到的杂草类植物进行点喷的路径规划。

无人机系统的飞行时间受电池容量的限制，因此最优飞行路径对于高效喷洒薇甘菊植物是必要的。最优路径规划可以基于旅行商问题（TSP）来进行，其目标是确定无人机系统对每个检测到的薇甘菊植物进行点喷的最短路径，然后返回起点[24,25]。针对此问题，已经测试了不同的算法，其中一些包括 Moon 等人[26]和 Shivgan 和 Dong[25]使用的遗传算法，Dorigo 等人[27]使用的蚁群优化（ACO）等。在本研究中，由于蚁群优化在实现上简单、可行，并且能更快地生成高质量的解决方案[27]，因此采用了蚁群优化。通过使用 DroneKit-Software（3.3.0）In The Loop（SITL）（3D Robotics，伯克利，CA，美国）、MAVProxy[28]和 Mission Planner（Ardupilot 开发团队和社区）对确定的最优飞行路径进行了测试。DroneKit-SITL提供了一个应用程序编程接口（API），用于在无人机的计算机上运行基于Python的应用程序，而MAVProxy和Mission Planner是地面控制站（GCS）软件程序，用于控制和模拟无人机的应用[29]。GCS和无人机之间的通信是通过一种名为MAVLINK的二进制串行遥测协议实现的[30]。

本研究的总体目标是开发一种计算机视觉（CV）算法，用于在玉米田中检测出草害植物，并利用检测到的位置进行最优的定点喷雾应用。四个具体目标是：（i）使用YOLOv5m开发一种计算机视觉（CV）算法，用于在玉米田中检测出草害植物，使用辐射（反射率校准）和伽柏校正的相对低分辨率（1.2兆像素）多光谱航拍图像；（ii）将检测到的草害植物的基于像素的边界框坐标转换为地理坐标；（iii）使用检测到的草害植物
的地理坐标，使用蚁群算法（ACO）生成最优飞行路径；（iv）基于生成的最优飞行路径，使用DroneKit-SITL和MAVProxy和Mission Planner GCS软件模拟定点喷雾无人机。

2. 材料与方法

2.1. 实验场地

本研究是在德克萨斯州大学农业生命研究中心位于德克萨斯州大学城附近的一个玉米地（图1；西经96∽25′45.9″，北纬30∽32′07.4″）进行的，该玉米地面积约为5.9公顷（14.6英亩）。实验区的大部分土壤为韦斯伍德粉质黏土壤土。

图 1 。实验场地位于得克萨斯州伯利森县大学城附近的得克萨斯 A&M 大学农场（西经 96◦25‘45.9 ‘ ‘，北纬 30◦ 32’ 07.4’ ‘ ），在那里，棉花植株被种植在玉米地的中间，以模拟存在志愿棉花植株的情况。

2.2. 图像数据采集

一台五波段（蓝色、绿色、红色、近红外、红边）多光谱相机——Red-Edge-MX（由AgEagle Aerial Systems Inc.提供，以MicaSense名义运营，位于堪萨斯州威奇托；图2）——被安装在一个定制的无人机系统上，用于在玉米植株达到吐丝期（VT）时收集带有棉花的玉米田的航拍图像（图3）。选择这款相机而不是传统的RGB相机的原因是，其图像带有地理标签，并且可以使用制造商提供的开源代码进行辐射校正等高级图像处理技术；此外，它还提供了在需要时使用其他波段（如近红外和红边）的灵活性。每个波段的中心波长和带宽如表1所示。在这个玉米植株的生长阶段获取图像是因为我们想要测试最坏的情况，即玉米冠层相对比VC植物更大更高。这在其他研究中也有体现，其中玉米田的最低检测精度出现在它们达到VT生长阶段时[7]。无人机系统（Hylio AG-110； ）美国德克萨斯州里士满的Hylio公司最初是为广播喷雾应用而设计的。数据于2021年5月14日中午11点至下午2点（中央夏令时）收集，地点位于海拔近4.6米（15英尺）处，使用自动曝光相机设置。这导致地面采样距离（GSD）约为0.34厘米/像素。选择这个高度有两个主要原因：一是提高图像分辨率和GSD，二是使从该高度进行高效的点喷应用时，不会导致喷雾液滴的过度扩散。由于定制的无人机系统不是为航空测量而设计的，因此没有可用的软件界面来根据重叠设置捕获图像。因此，图像是通过自动曝光相机设置捕获的。

在基于定时器的设置下，每秒钟捕获一张图像。由于缺乏重叠设置以及喷头无人机系统（通常，可喷雾无人机系统比测量无人机系统振动更剧烈）的设计方面引起的振动，这导致了许多未使用的和失真的图像。该无人机的有效载荷容量为 10 千克/升。该无人机以 2 米/秒的速度飞行。

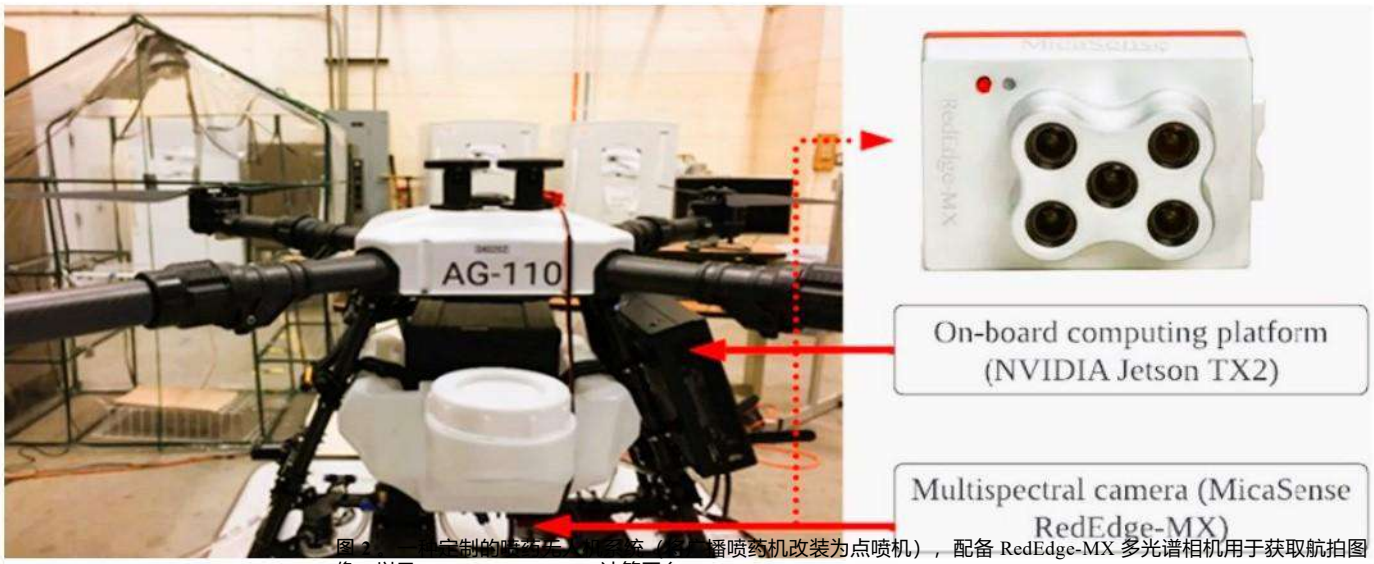

图 2。一架定制的喷雾无人机系统（大疆精灵播喷药机改装为点喷机），配备 RedEdge-MX 多光谱相机用于获取航拍图像，以及 NVIDIA Jetson TX2 计算平台[7]。

图 3。（A）定制的喷雾无人机系统在实验玉米地（包含一些为模拟志愿棉（VC）植株而种植的棉花植株）上空飞行，捕获五个波段的多光谱图像；（B）RGB（红、绿、蓝）合成图像显示实验区的一部分，可以看到处于营养生长阶段（VT）的玉米和一些模拟为 VC 植株的棉花植株。

表 1 。MicaSense RedEdge-MX 相机每个频谱波段中心波长和全宽半高（FWHM）带宽。

频谱波段	中心波长（纳米）	全宽半高带宽（纳米）
蓝色	475	20
绿色	560	20
红色	668	10
近红外	840	40
红边	717	10

一种机载计算平台，即由帕斯卡图形处理单元（GPU）组成的 Jetson TX2（英伟达公司，美国圣克拉拉）开发板，被安装在定制的无人机上，用于在玉米田中近乎实时地检测杂草类植物（图2）。帕斯卡GPU是一种低成本、快速且广泛使用的嵌入式人工智能计算设备。它由256个英伟达统一计算设备架构（CUDA；英伟达，美国圣克拉拉）核心组成，具有8GB的内存和32GB的存储空间[32]。

2.3. 制造商推荐的校正措施

通过 RedEdge-MX 相机采集的各个波段图像是根据制造商的建议[33]进行校正的，使用的是在数据采集当天飞行前不久拍摄的 RP 04 型反射率面板（图 4）的图像。

根据制造商的工厂校准（其中绝对反射率值沿着从 400 纳米到 850 纳米波长绘制），在研究中使用的面板对应的蓝色、绿色、红色、近红外和红边的反射率值分别为 0.60、0.61、0.61、0.60 和 0.56。按照制造商的流程，从反射率面板图像中提取兰伯特面板的一块区域，并将其辐照度值转换为反射率值的尺度，然后将其应用于整个图像以将其转换为反射率图像。辐射校正和伽马校正的源代码均来自 MicaSense 的 GitHub（GitHub, Inc., San Francisco， CA， USA）存储库[33]。根据所使用的面板的反射率值对原始源代码进行了修改，并生成与每个捕获图像的场数据收集对齐的 RGB 图像，如图 3 所示。生成 RGB 图像是因为 YOLOv5 网络架构接受具有三个通道（通常是 RGB 图像）的输入图像。在 Spyder 集成开发环境（IDE）5.2.2 版中使用了 Python 3.8.12 版（Python 软件基金会，美国特拉华州威尔明顿）。整个过程包括通过增益和曝光设置对图像进行归一化处理，然后进行转换

将它们转换为辐照度，然后是反射率。使用增益和曝光设置来标准化原始图像数据，校正曝光时间和传感器敏感度的差异，并允许准确转换为辐照度值。通过曝光和增益设置对图像进行标准化对于将原始图像转换为辐照度至关重要。曝光时间（表示相机传感器暴露于光线的持续时间）和增益（源自ISO设置）从图像元数据中获取。原始图像经过校正，以考虑暗像素偏移、渐晕效应和行梯度不准确性。在此之后，使用曝光时间和增益对校正后的图像进行标准化，确保不同图像的一致性。然后，通过将图像数据与增益-曝光乘积进行缩放，并应用针对图像位深调整的辐射定标系数，将其转换为辐照度单位（W/m2/nm/sr）。这个过程确保了可靠反射率映射所必需的准确和可比的辐照度值。在这之后，研究中应用了锐化掩码（一种图像锐化技术）作为增强技术来提高视觉清晰度，然后应用伽柏校正来使增强后的图像看起来更明亮，视觉上更接近我们的眼睛所看到的[33]。对于这些，使用了GitHub源代码中存在的原始值。锐化掩码技术使用线性滤波器 将高通滤波输入图像的一部分添加到原始图像中，通过过滤噪声来帮助锐化原始图像[34]。同样，伽柏校正是一种图像增强技术，用于最小化成像传感器的非线性影响，从而使图像看起来更明亮，对比度增强，视觉上更接近[35-37]。

2.4. YOLOv5

自2020年6月发布以来，YOLOv5已成为计算机视觉应用中流行的目标检测算法。YOLOv5最初以四种不同的变体YOLOv5s、YOLOv5m、YOLOv5l和YOLOv5x发布，其中下标表示网络深度和使用的参数数量。在这里，*s*、*m*、*l*和*x*分别表示YOLOv5网络的四个变体：小型、中型、大型和超大型。YOLOv5网络带有在Common Objects in Context（COCO）数据集上训练的预训练权重，该数据集包含80个不同的类别，因此YOLOv5被预训练以检测80个不同的类别。在本研究中，网络被定制以检测单个类别“vc”，用于VC植物。YOLOv5网络的架构包括三个主要组件：骨干网络、颈部/特征聚合网络（PANet）和头部/输出网络（图5）。骨干网络是一个卷积神经网络（CNN），负责聚合细粒度的图像。颈部由特征聚合层组成，这些层将特征组合起来构建特征金字塔网络。这些生成的特征图随后被传递到头部或输出网络。输出网络处理模型的最终检测阶段。它将锚点框应用于从上一层获得的特征图，并生成一个向量，其中包含目标对象的类别概率、对象得分以及包围目标对象的边界框的位置[40]。

2.5. 图像数据准备

在经过辐射校正和伽马校正的RGB图像中，只选择那些至少包含一棵背景为玉米的VC植物的图像，然后使用Python的Augmenter库进行图像增强[41]。为此，使用了*rotate*、*flip_left_right*、*zoom_random*和flip_top_bottom操作，概率值分别为0.80、0.40、0.60和0.80。这些值的选取是为了确保每次图像通过增强管道时，80%的时间应用*rotate*操作；40%的时间应用*flip_left_right*操作；60%的时间应用*zoom_random*操作；80%的时间应用*flip_top_bottom*操作。通过这种方式，我们从总共34幅原始图像（至少包含一棵VC植物）中生成了总共521幅RGB图像，这些图像经过辐射校正和伽马校正，并且每幅图像的大小 都为原始大小，即1207×923像素。其中，417幅图像（80%）用于训练，77幅（15%）用于

用于验证，27 例（5%）用于测试。数据分割的比例是任意选择的。

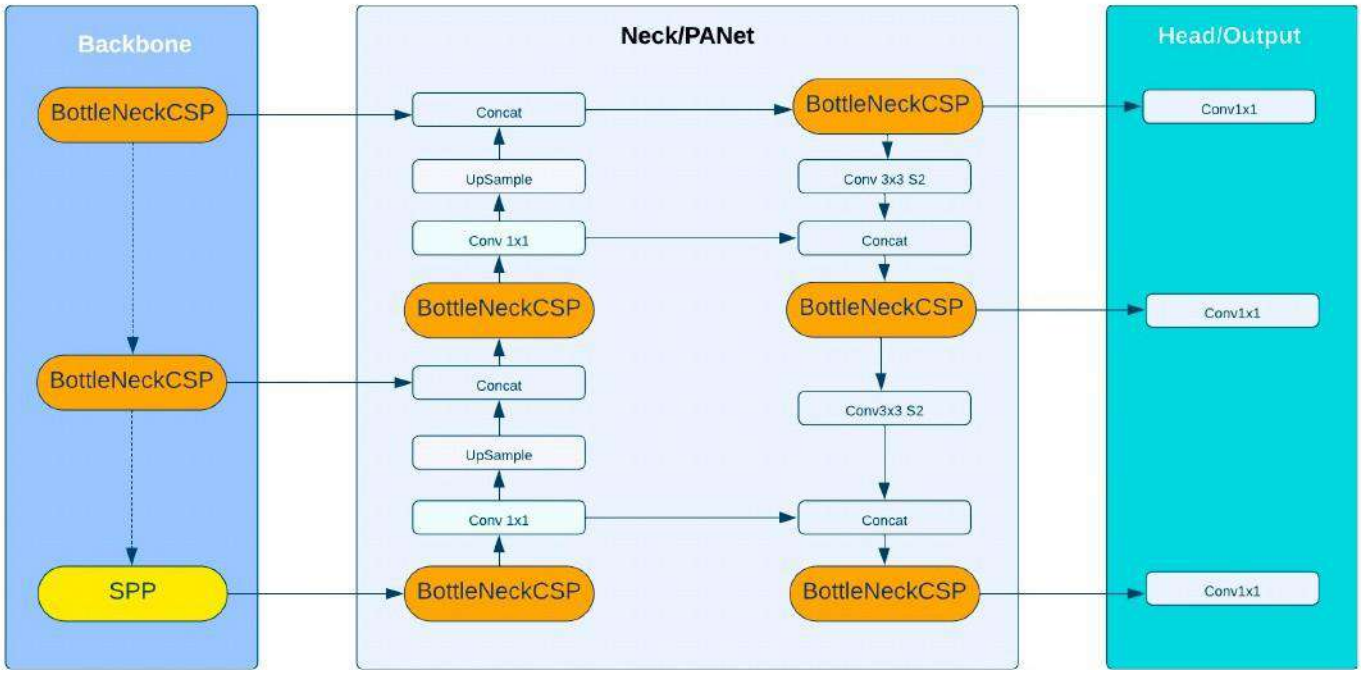

图 5. YOLOv5 网络架构的总体概述。

2.6. YOLOv5 训练

YOLOv5m 的源代码是从 Ultralytics 公司的 GitHub 存储库中获得的[12]。使用 PyTorch 框架（Facebook AI Research Lab，Melno Park，CA，USA），torch 版本为 1.10.0，计算统一设备架构（CUDA）版本为 11.1.0（NVIDIA，Santa Clara，CA，USA），在特斯拉 P100-PCIE-16GB（NVIDIA，Santa Clara，CA，USA）GPU 上使用谷歌 Colab（Google LLC，Melno Park，CA，USA）AI 平台对 YOLOv5m 模型进行训练。该模型使用原始超参数值进行训练，初始学习率为 0.01，最终学习率为 0.1，动量为 0.937，权重衰减为 0.0005，交并比（IoU）阈值为 0.50，总共迭代 621 次。这实质上意味着 YOLOv5m 在 621 次迭代中达到了收敛。

2.7. 边界框坐标转换

该过程的第一步是从 RedEdge-MX 相机的地理标记图像中提取左上角坐标，这些图像在辐射校正、伽马校正、图像增强和 RGB 波段对齐的过程中，通过修改从 MicaSense 的 GitHub 存储库获得的原始 Python 脚本[33]。每个地理标记图像的提取的 GPS 坐标被存储在一个逗号分隔变量（CSV）文件中。YOLOv5 detect.*py Python* 脚本被修改以提取每个检测到的边界框的中心坐标，这些坐标被存储在一个单独的 CSV 文件中。还开发了另一个 Python 脚本，用于同时利用这两个 CSV 文件，然后将检测到的 VC 植物的逐像素边界框转换为 GPS 坐标。在该 Python 脚本中，首先将 GSD 转换为米/像素格式，将小数坐标转换为通用横轴墨卡托（UTM）格式，从中提取北向和东向值，然后在转换过程中使用 UTM 区域 14 将基于像素的中心坐标转换为基于 UTM 的 GPS 坐标[42]。

该转换过程中潜在的误差来源包括图像初始地理标记的不准确性、GSD 的变化以及 UTM 转换的精度。检测到的边界框坐标的准确性也受到性能的影响。

YOLOv5 模型在物体检测中可能存在误差幅度。此外，图像处理步骤中的任何微小错位或失真都可能传播到最终的 GPS 坐标中。至关重要的是要考虑到这些因素，并应用适当的误差校正和验证步骤，以确保从逐像素的边界框推导出的 GPS 坐标的可靠性。

2.8. 蚁群算法的最优飞行路径

蚁群算法是解决诸如车辆路径问题、开放式车间调度问题和顺序排序问题等众多问题的最先进算法[27]。它基于这样一个事实：当蚂蚁沿着从蚁群到食物源的路径移动时，它们会留下信息素，信息素会随着时间的推移而蒸发。这意味着沿着较长路径行走的蚂蚁留下的信息素沉积较少，但沿着最短路径的信息素强度要高得多。在我们的案例应用中，VC 工厂位置是人工蚂蚁会通过尽可能短的路径找到的食物源。在过去的许多案例中，蚁群算法已被证明是广泛接受的确定无人机最优飞行路径的算法[43,44]。实现蚁群算法的源代码取自 *fabien-brulport* 的 GitHub 库[45]。原始源代码经过了修改，以生成一个 CSV 文件，其中包含按照蚁群算法生成的最优路径的节点顺序的 GPS 坐标（纬度、经度）作为输出。在我们的研究中，我们将信息素权重、启发权重和蒸发率的值分别设定为2.01、1和0.5，以实现不同组合值的尝试后最短距离。同样，代理数量（即人工蚂蚁的数量，即无人机的数量）和迭代次数的值分别为1和100。数学上，在任何给定的时间*t*，张等人[44]解释说，蚂蚁*k*选择从*i*到*j*路径的概率由以下公式给出：

$$P_{ij}^{(k)}(t) = \left\{ \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{S \subset allowed_k}[\tau_{is}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}} \text{if } j \in allowed_k \; ; \; 0 \text{ otherwise} \right\} \quad (1)$$

在这里，α和β分别是信息素和启发式权重。ηij（t）和τij（t）分别是点i和点j之间的信息素可见性和数量。S ⊂ allowedk 是蚂蚁可以选择的所有可能路径点的一个集合。信息素的量决定了可见性，进而决定了选择路径的概率。因此，信息素的量越高，点i和点j之间特定路径的可见性就越高，选择该路径的概率也就越大。

2.9. 点喷无人机系统仿真

一旦地理坐标以与通过蚁群优化算法获得的节点相对应的最优顺序保存下来，就使用 DroneKit-SITL（位于美国加利福尼亚州伯克利市的 3D Robotics 公司）Python 软件包，通过 Python 脚本模拟无人机的飞行路径，并在两个地面控制站（GCS）上进行监控：Mission Planner 版本 1.3.76 [46] 和 MAVProxy 版本 1.8.45 [28]。使用 MAVLink [30, 47] 协议在两个 GCS 和 DroneKit-SITL 之间进行通信。这种通信是通过使用传输控制协议（TCP）作为主端口，然后将输出端口转发到三个用户数据报协议（UDP）端口来实现的。实现整个过程的步骤在下面的算法 1 中有所展示。

一个 Python 脚本被用于读取一个 CSV 文件，该文件按顺序包含了由蚁群算法生成的所有节点的 GPS 坐标，这些坐标对应于检测到的需要点喷的 VC 植物的位置。在无人机系统模拟成功执行后，我们还能够在由 Hylio 开发的 AgroSol 2.105.0 版地面控制站上传带有最优飞行路径的 CSV 文件。AgroSol 软件提供了控制定制点喷无人机的接口。图 6 中的流程图展示了从数据收集到点喷无人机模拟的整个工作流程。

Algorithm 1 Spot Spray Simulation Algorithm

Require: CSV file with GPS coordinates of detected VC plants
Ensure: Simulated flight path based on ACO algorithm on Mission Planner GCS

```
Procedure SpotSpraySimulationAlgorithm:
Open two terminal windows from Anaconda environment;
while paths set to the directory where MAVLink was installed do
    if terminal window 1 then
    dronekit-sitl copter --home=30.534351,-96.431239,0,180
    --model=copter;
    end if
    mavproxy --master tcp:127.0.0.1:5760 --sitl 127.0.0.1:5501
    --out udp:127.0.0.1:14550 --out udp:127.0.0.1:14551 --out
    udp:127.0.0.1:14552;
end while
Open Mission Planner GCS;
while mode == guided do do
    if UDP port ==    14551 then
        Open Spyder IDE and execute python script "vc spot spray.py";
    else
        end
    end if
end while
end
```

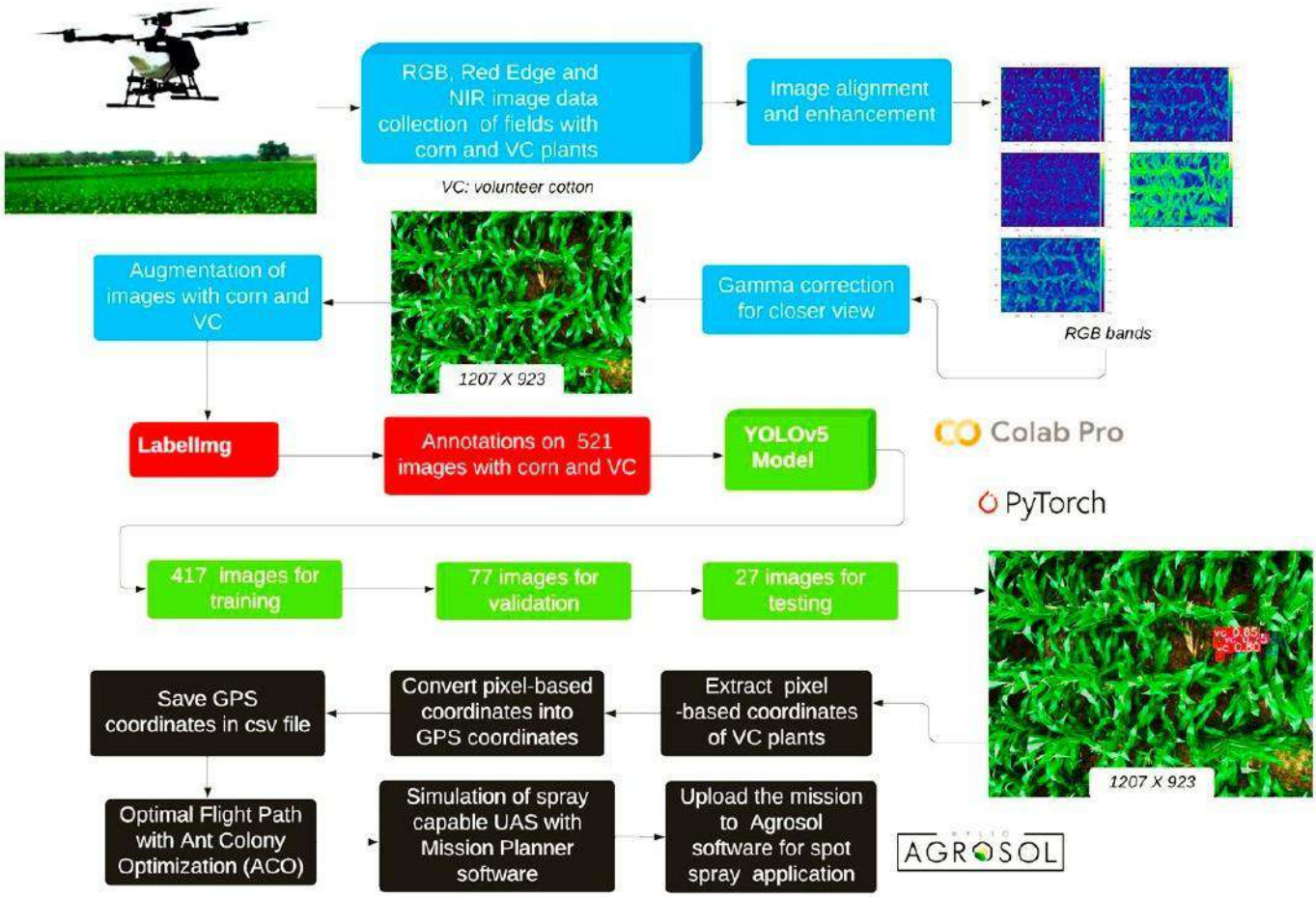

图 6. 一个流程图，展示了完整的工作流程，代表了本研究中使用的每个步骤。

3. 结果

3.1. 使用 *YOLOv5m* 的 *CV* 算法在辐射校正后的航拍图像上检测玉米地中的杂草类植物

图7展示了在训练和验证数据集上的框损失（IoU损失）和目标性损失的图形。在第600次迭代之前，两种损失都已收敛，之后没有看到进一步的改善。在训练数据集上，第476次迭代时框损失的最低值为0.028，而目标性损失的最低值为0.009，出现在第613次迭代。在验证数据集上，第394次迭代时目标性损失的最低值为0.0094，而框损失的最低值为0.0296，出现在第370次迭代。在图8中，可以看到不同性能指标的图形。在第320次迭代时，精度达到最大值约0.98，而在第525次迭代时，召回率达到最大值0.77。其中最重要的是mAP@0.50，其最大值为0.81，出现在第613次迭代。mAP@0.50值为0.81，表明模型在考虑IoU阈值为50%的检测时表现良好，这意味着预测的边界框与真实框重叠至少50%。这个相对较高的值表明，模型在这些条件下能够准确检测和定位志愿棉（VC）植物。在迭代次数为521时，mAP@0.50:0.95的最大值被获得，其值为0.33。mAP@0.50:0.95值为0.33，揭示了模型在不同IoU阈值（从50%到95%）下的性能表现更加细致。这个较低的值表明，虽然模型在识别和定位50%重叠的物体方面很有效，但随着正确检测所需的重叠变得更加困难，其性能会下降。这表明在实现精确定位方面存在潜在挑战，特别是在区分VC植物与背景和其他物体（如玉米植物、杂草和土壤）方面。精度-召回率曲线（PRC）显示总体准确率接近79%，F1分数的最大值在近40%的置信水平下为0.76（图9A、B）。图10中的混淆矩阵显示，YOLOv5m经过充分训练，能够以78%的准确率对VC植物进行分类，但由于玉米植物、杂草、土壤等背景类别的存在，损失率为22%。为了减少这些误分类，可以实施一些策略。首先，增加训练数据集的多样性和大小可以帮助模型更好地区分VC植物和其他类似物体。这可以通过在各种光照条件和生长阶段收集更多标注的图像来实现。其次，数据增强技术如旋转、缩放和颜色调整可以提高模型对田间环境变化的稳健性。

图10展示了经过训练的YOLOv5m模型的一些检测结果，该模型在NVIDIA Tesla P100 GPU-16GB上的平均推理速度为47帧每秒（FPS）。该模型后来被部署在安装在定制无人机系统（图2）上的NVIDIA Jetson TX2 GPU上，对于大小为640×640像素的图像，调整后的平均推理速度为2.535秒（约0.4 FPS）（图11）。YOLOv5m模型在NVIDIA Jetson TX2 GPU上的推理速度是评估实时应用可行性的关键因素。实时处理通常需要至少30 FPS的速度，以确保在动态环境中的平滑和响应性能。观察到的0.4 FPS的速度表明存在显著的延迟，使其不适合需要立即检测和响应的场景，例如实时监测和干预农业领域。这种低推理速度可以归因于多个原因。其中一个可能的原因是Jetson TX2板上的GPU硬件设置，无法充分利用其最大计算能力。

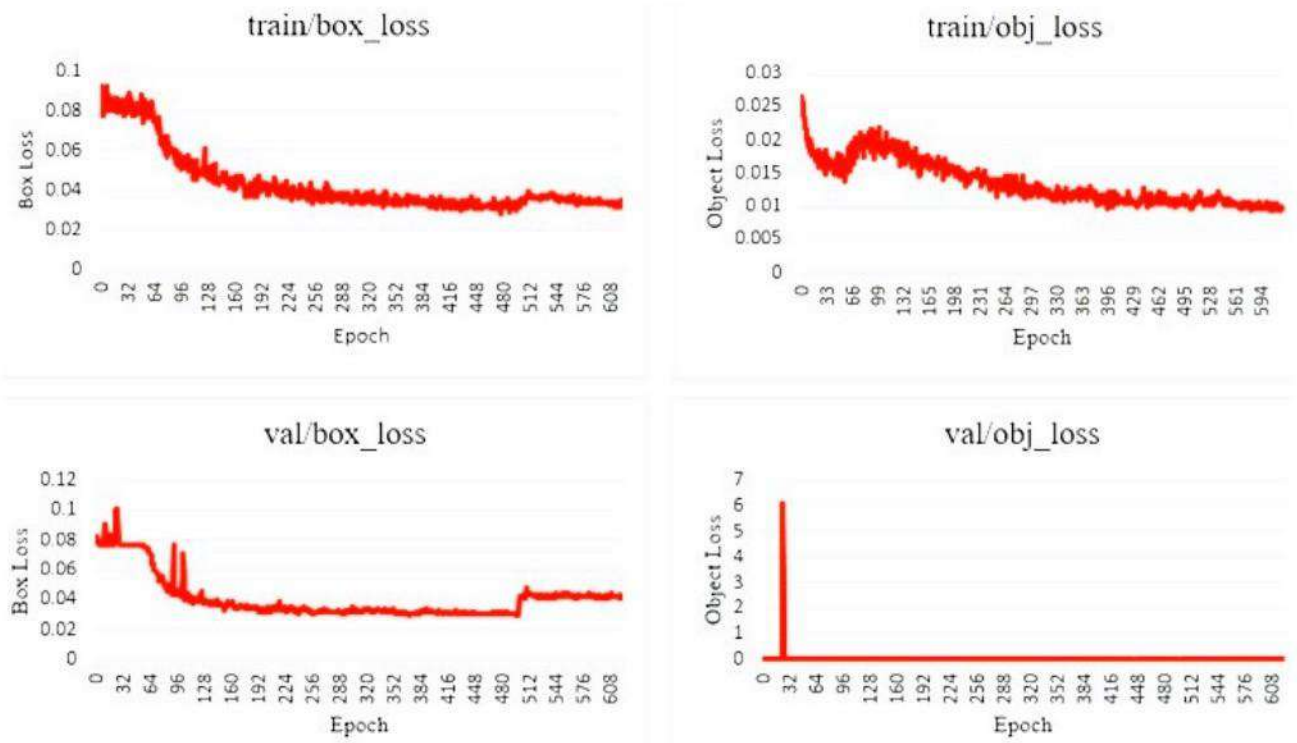

图 7. 在 YOLOv5m 对训练和验证数据集进行训练的过程中所获得的不同类型的损失。

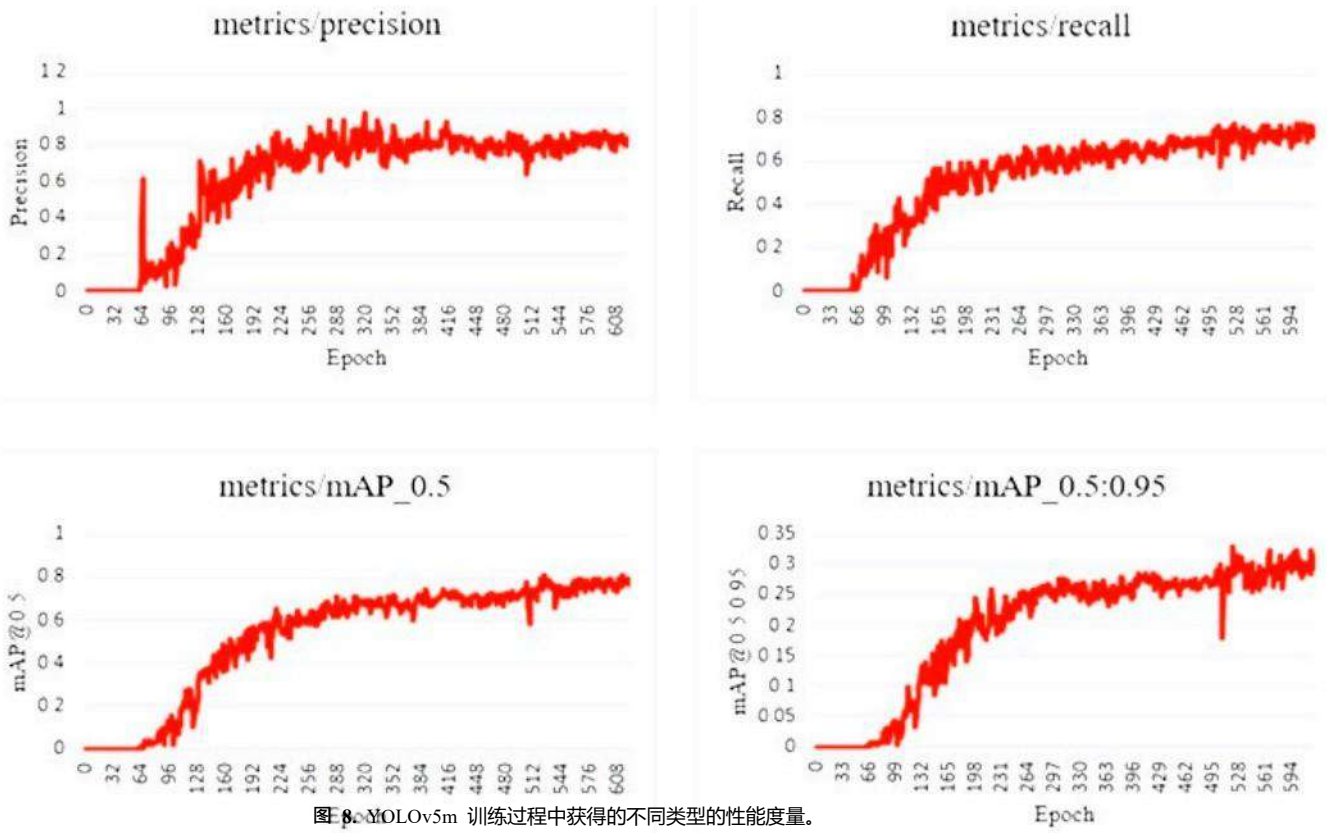

图 8. YOLOv5m 训练过程中获得的不同类型的性能度量。

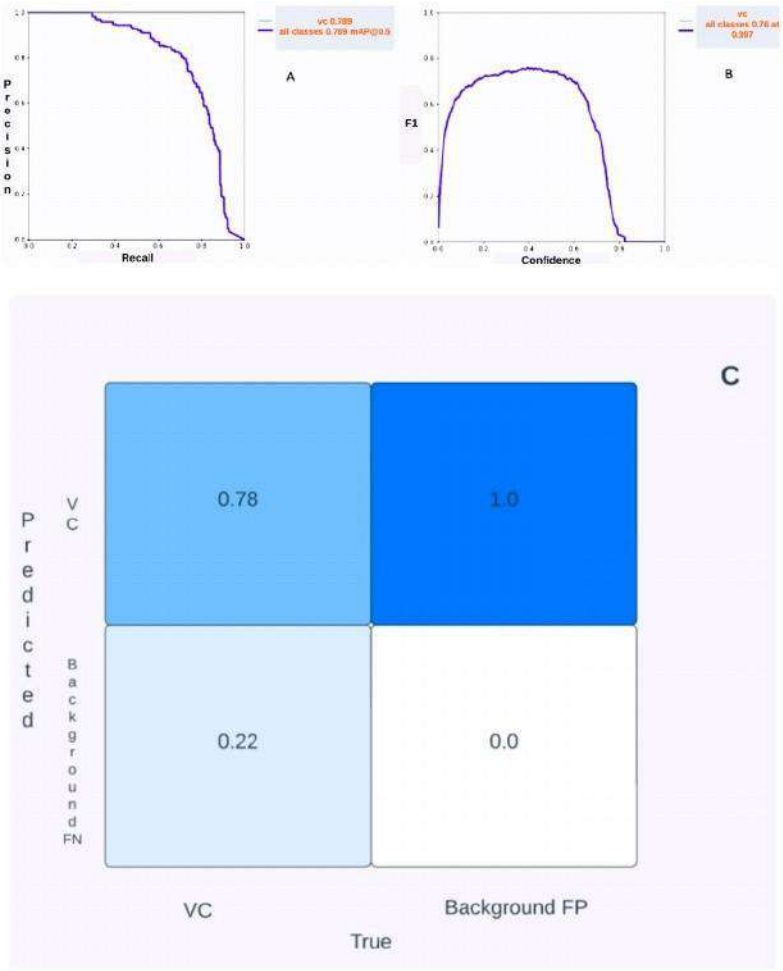

图 9. （A）精确率 - 召回率图，（B）F1 分数与置信度分数图，以及（C）训练 YOLOv5m 后得到的混淆矩阵。

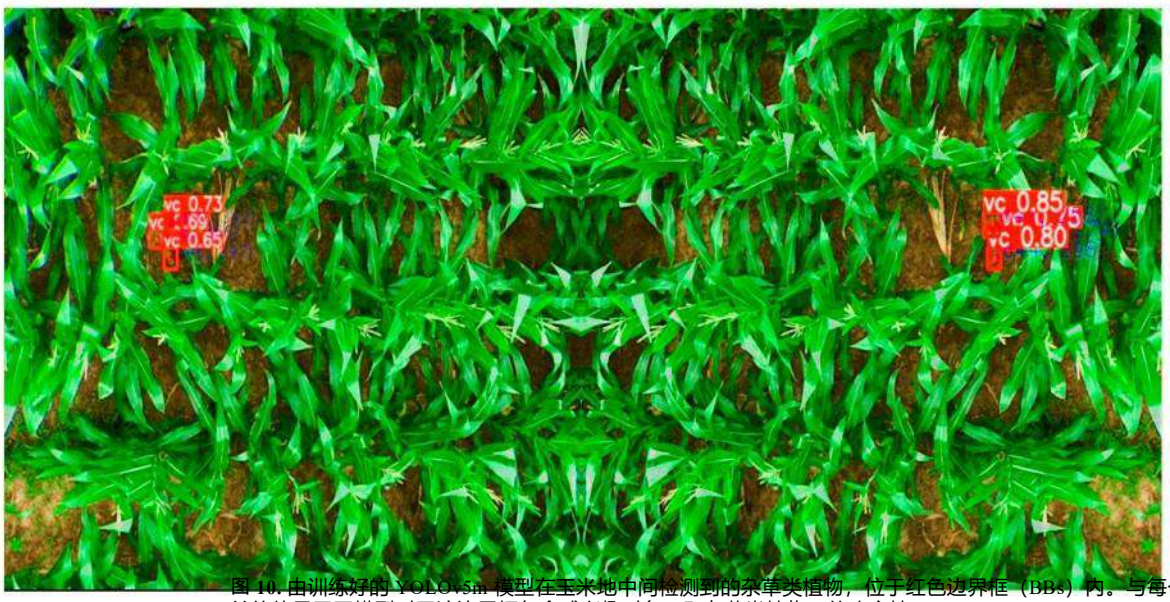

图 10. 由训练好的 YOLOv5m 模型在玉米地中间检测到的杂草类植物，位于红色边界框（BB5）内。与每个边界框相关的值显示了模型对于该边界框包含感兴趣对象（即杂草类植物）的确定性。

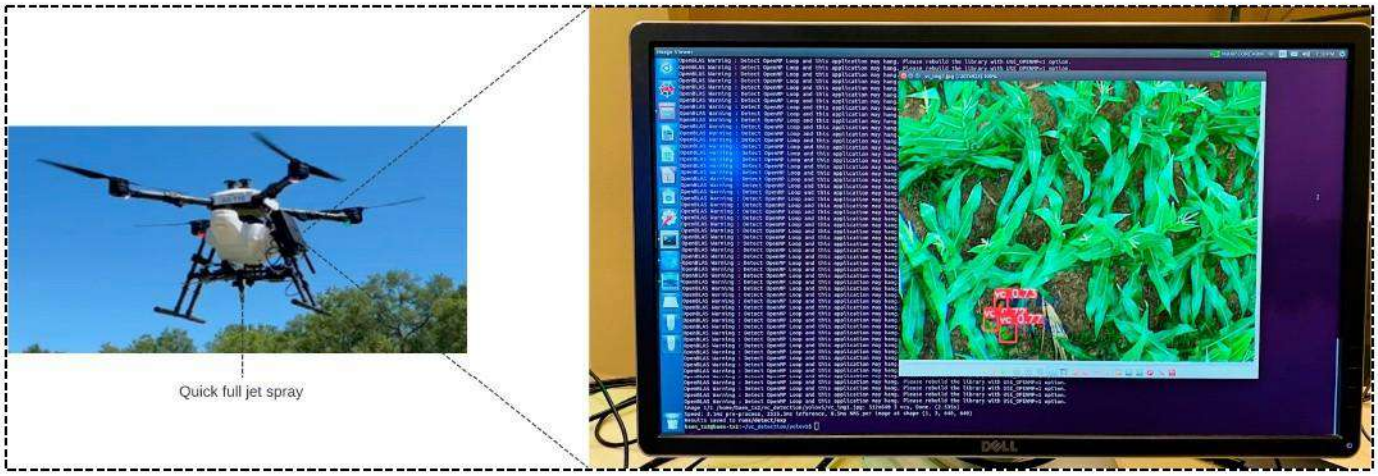

图 **11.** 通过将 YOLOv5m 部署在安装在定制的具备点喷功能的无人机上的 NVIDIA Jetson TX2 上，对玉米地中的杂草类植物进行检测。

3.2. 蚁群算法用于确定点喷应用的最优飞行路径

在实验区中，选择了十个随机的棉花植株位置，其相应的GPS坐标可以在表2的左侧看到。选择这十个随机位置是为了模拟真实的田间条件，同时确保无人机系统（UAS）在单次飞行中能够进行点喷的可行性。这些位置是随机选择的，以覆盖实验区的不同区域，确保无人机系统能够一次飞行有效地管理整个区域。选择这些位置的标准包括确保所有选择的位置都是可访问的，并且能够在无人机的操作限制（如飞行时间和电池寿命）内覆盖。每个位置都包含多个棉花植株（因为每个位置都种植了多个棉花种子）；然而，只考虑一个植株的GPS位置。这代表了模拟现实世界的情况，即棉花植株通常成组生长，通过考虑每组中一个植株的位置，就可以对包含该组棉花植株的整个区域进行喷雾。

表 **2** 。实验地块中十个 VC 植物随机选取位置的全球定位系统坐标（左）以及应用蚁群算法后的有序坐标/节点（右）。

风险投资	纬度	经度
1.	30.5343	−96.4312
2.	30.53431	−96.4301
3.	30.53492	−96.4303
4.	30.53386	−96.4299
5.	30.53531	−96.4289
6.	30.53484	−96.4299
7.	30.53687	−96.4284
8.	30.53537	−96.4289
9.	30.5359	−96.4283
10.	30.53321	−96.4298

蚁群算法

节点	纬度	经度
1.	30.53687	−96.4284
2.	30.5359	−96.4283
3.	30.53321	−96.4298
4.	30.53386	−96.4299
5.	30.53431	−96.4301
6.	30.53492	−96.4303
7.	30.5343	−96.4312
8.	30.53484	−96.4299
9.	30.53537	−96.4289
10.	30.53531	−96.4289
11.	30.53687	−96.4284

在这种情况下，第一个位置也被视为点喷无人机开始和结束飞行的起始位置。表2的右侧显示了按照蚁群算法确定的顺序生成的节点。节点1和节点11表示同一个位置，即飞行开始和结束的位置。点喷应用生成的最优路径可以在基于网页的输出（图12）上看到，该输出由*Streamlit* Python包（Streamlit Inc.，旧金山，CA，USA）生成。底部图表显示x轴上的迭代次数和y轴上的最优覆盖距离（以千米为单位）。点喷无人机沿着生成的路径覆盖的总距离为674.17米（0.67千米/0.42米）。

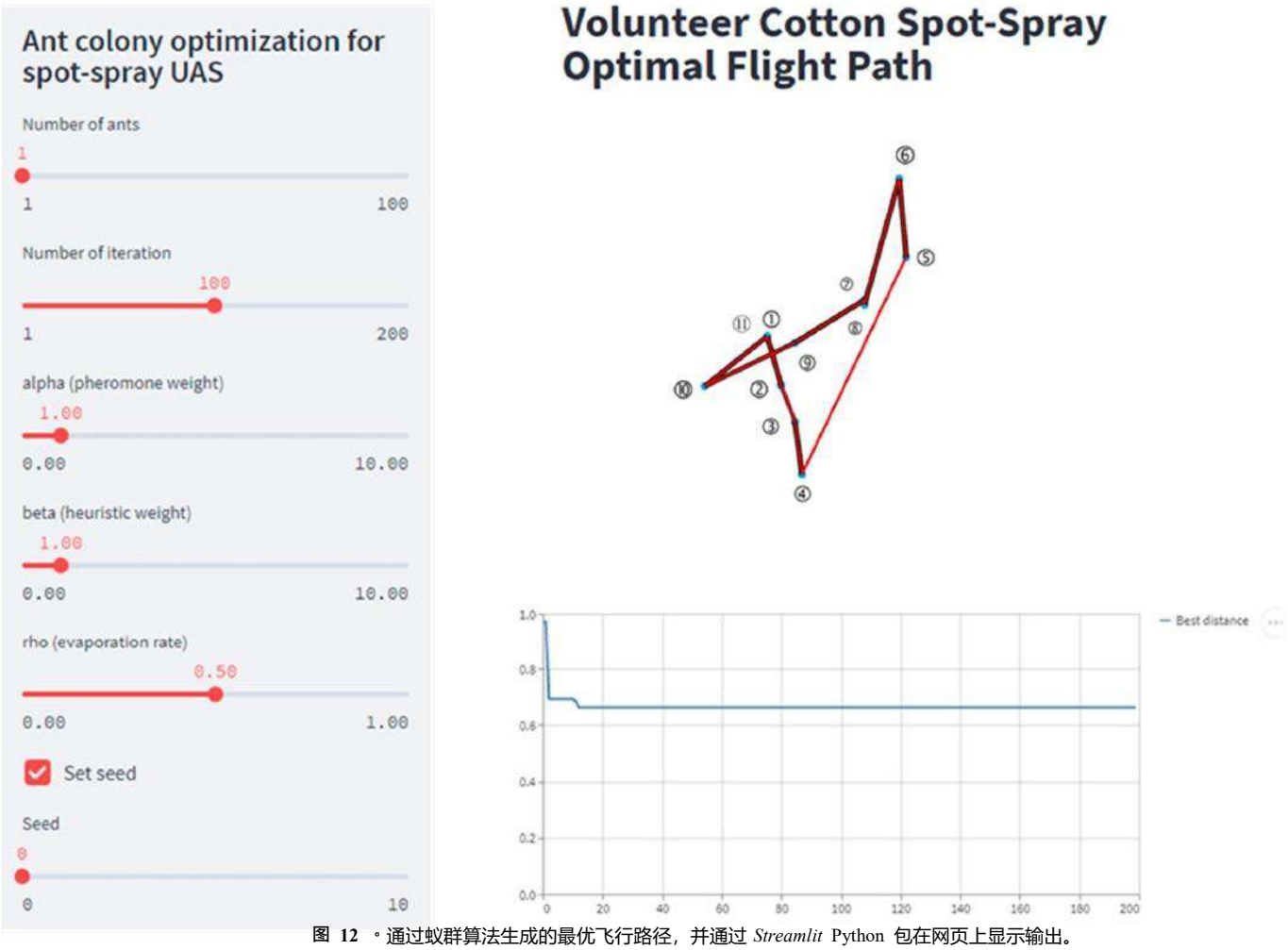

图 12 。通过蚁群算法生成的最优飞行路径，并通过 *Streamlit* Python 包在网页上显示输出。

3.3. 基于蚁群算法生成的最优飞行路径，在 *Mavproxy* 和 *Mission Planner* 上进行点喷无人机系统仿真

一旦通过蚁群算法获得最佳飞行路径（表 2），就使用 DroneKit-SITL 按照图 13A-C 所示在地面控制站 - *MAVProxy* 和任务规划器上关于点喷 *UAS* 模拟的部分中描述的步骤来模拟点喷 UAS。

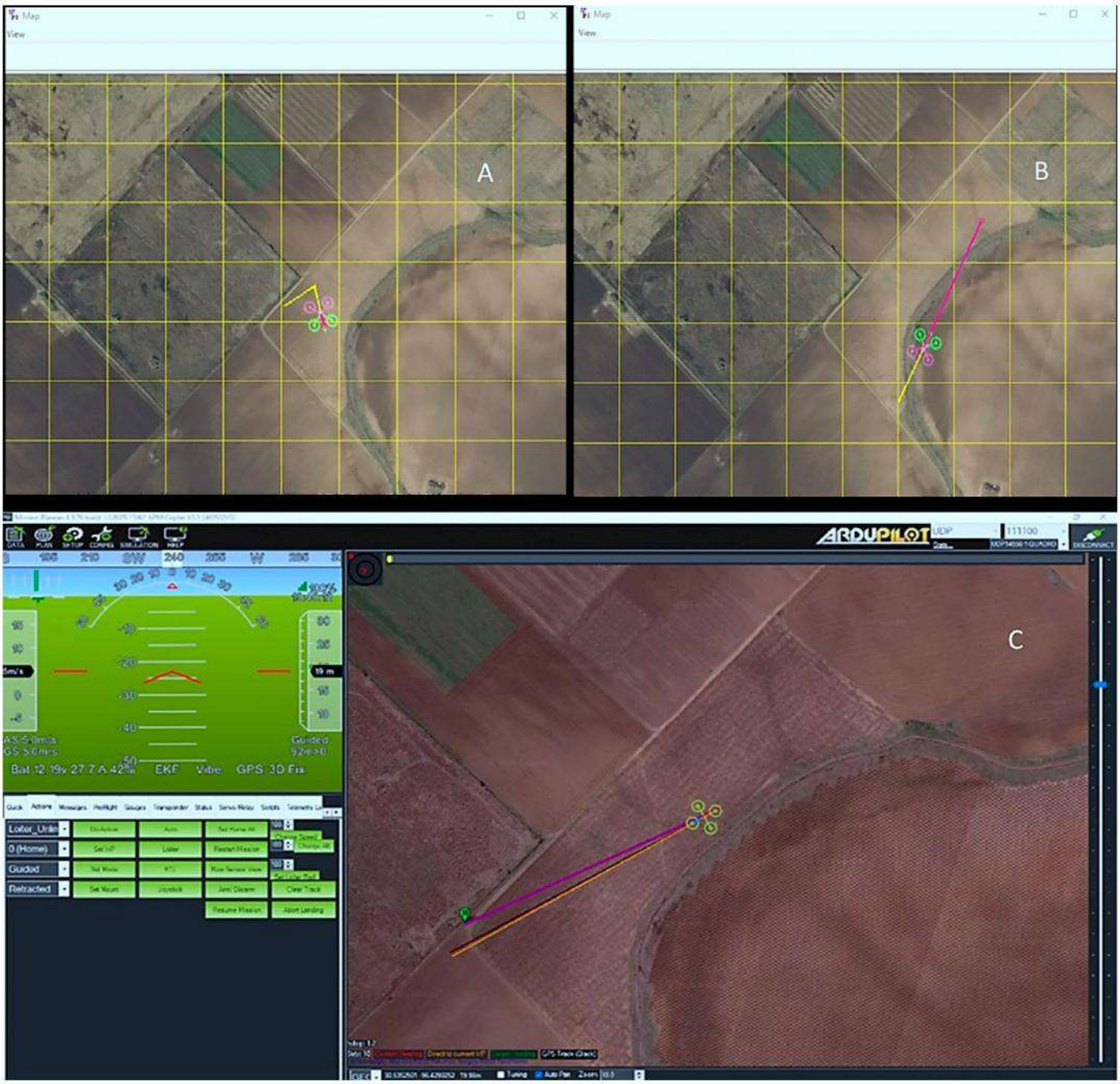

图 13. 在 MAVProxy（A、B）和 Mission Planner（C）地面控制站上的点喷式无人机系统模拟。图像 *A* 展示了模拟的无人机从节点 1 飞往节点 2，图像 *B* 展示了它从节点 4 飞往节点 5，图像 *C* 展示了模拟的无人机从节点 8 飞往节点 9。

3.4. 农业用太阳能无人机上的点喷任务

AgroSol（美国德克萨斯州里士满的Hylio公司）的GCS软件允许用户上传由蚁群算法生成的包含节点GPS坐标的CSV文件，然后，可以使用不同的设置进行点喷应用（图14）。在本研究中，一旦上传了包含点位的CSV文件，AgroSol就会生成一个点组，如图14左侧所示。然后，我们可以填写不同的参数值，如喷雾高度、喷雾量等。喷雾任务可以上传到无人机飞行控制器，然后，无人机可以前往每个点位，并在每个点位上精确点喷。

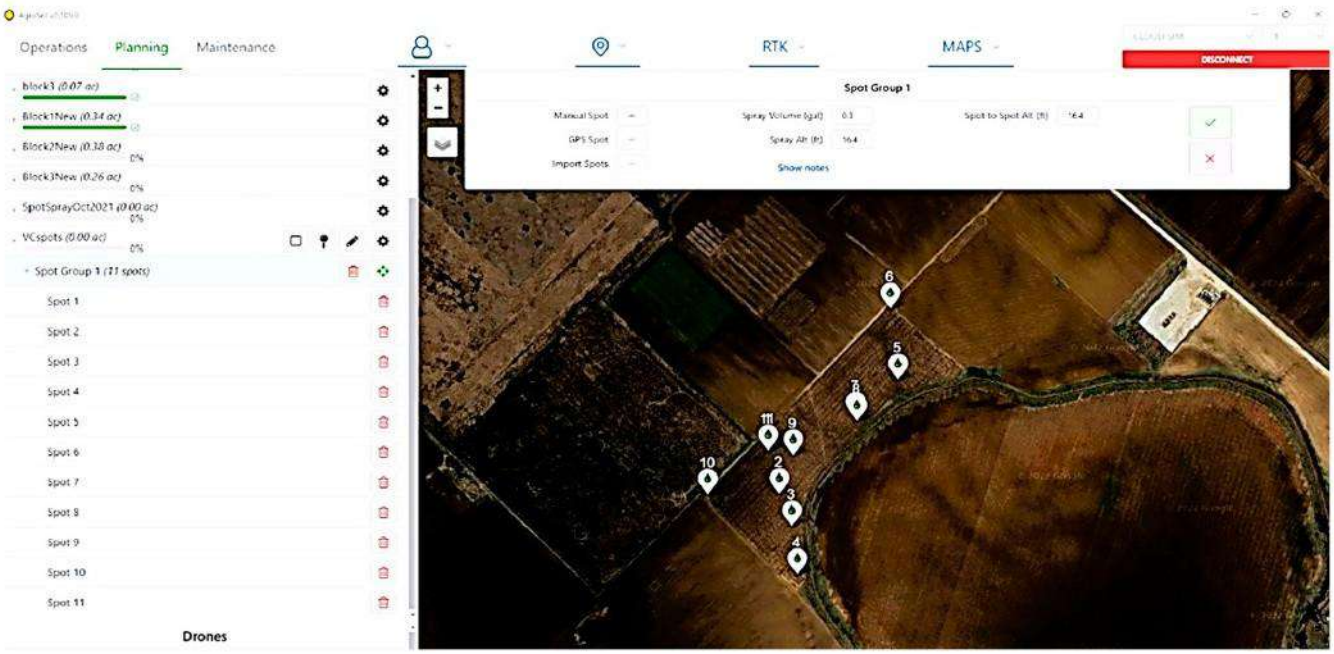

图 14. 上传包含通过蚁群算法生成的节点的 CSV 文件后，由 Agrosol 软件（2.87.5 版）生成的点喷节点。

4. 讨论

4.1. 使用 *YOLOv5m* 进行 *VC* 检测

张等人[48]在开发基于光谱反射特性区分棉花植株与其他作物的方法时，提到了多光谱遥感图像在检测其他作物中棉花的潜在应用。同样，在我们的前一项研究[49]中，我们能够证明基于无人机系统的多光谱遥感图像可以用来检测生长在玉米地里的棉株。在韦斯特布鲁克等人[50]的另一项研究中，作者能够利用航拍遥感RGB图像检测到早期生长阶段的棉株。本文所报道的研究借鉴了之前所有研究的动机和建议，因为它使用了由无人机系统收集的多光谱遥感图像。然而，不同之处在于使用了卷积神经网络算法，而不是传统的图像处理技术，如判别分析、主成分分析（PCA）、最大似然分类、线性光谱分解等。在张等人[48]进行的研究中，他们能够以100%的准确率区分棉花植株与其他作物，如玉米、大豆和高粱。然而，该方法使用手持式光谱辐射计或安装在拖拉机上的光谱辐射计，这些设备非常靠近作物冠层。这意味着他们的方法受到两个主要因素的限制：第一个因素是它不能应用于更大的农田区域；第二个因素是它是一个耗时的过程，不适合于对VC植物的近实时检测。我们之前的研究[49]使用了遥感多光谱图像和一些经典的机器学习技术，使VC检测过程半自动化；然而，它无法产生超过70%的分类准确率。

在本研究中，我们能够利用YOLOv5m开发一种CV算法，通过多光谱遥感图像检测在吐丝期（VT）生长阶段的玉米田中生长的VC植物（在穗原基生长阶段之前）。在另一项研究中[3]，我们能够利用未校准的RGB图像在早期玉米田（V3营养生长阶段）中检测到超过90%的VC植物。我们使用了YOLOv3，其分类准确率比之前的方法高出近29%[49]，但由于图像未进行辐射校正，训练模型对光照条件和环境因素的鲁棒性较差。此外，与本文所报道的研究相比，VC植物与玉米植物之间的相对高度差异较小。

在我们当前和最新的方法中，我们能够基于YOLOv5m开发一种CV算法，该算法对光照条件和环境因素更加稳健，因为它是在经过辐射校正的图像上进行训练的[51]。通过在经过辐射校正的图像上训练YOLOv5m模型，我们能够生成一个更可靠的模型，因为它是在更准确的数据上进行训练的[21]。使用当前的方法，我们能够以比之前的方法高出11.43%的准确率对VC植物进行分类，如图10所示。值得注意的是，在玉米田中，当植物处于VT阶段时，与之前研究中报道的V3阶段相比，获得了类似的检测准确率[3]。YOLOv5相对于其先驱和其他检测模型（如Faster R-CNN、EfficientDet和 SSD（单发多框检测器））具有几个优势，这些模型各有其优点和缺点。Faster R- CNN以其高精度而闻名，但它往往以牺牲推理速度变慢为代价，使其不太适合实时应用。EfficientDet 能有效地平衡准确性和速度，但它需要大量的超参数调整以及更大的计算资源。与 YOLOv5 相比，SSD 速度较快，但总体上在准确性方面有所滞后。

本研究的总体目标是加快德克萨斯州棉铃象鼻虫根除计划（TBWEP）的管理，同时降低化学成本。因此，除了开发一种近实时检测VC植物的系统外，还需要开发一种能将假阴性（即最大召回率）最小化的检测算法。这样，我们就可以最小化在玉米地中间遗漏可能成为棉铃象鼻虫宿主的VC植物的可能性。换句话说，对于我们的使用案例场景，我们的算法可以容忍假阳性，但不能容忍假阴性，因为棉铃象鼻虫造成的损害远远大于因假阳性而向一些不期望的位置喷洒所产生的额外成本。因此，可以使用第525次迭代获得的模型（图8）。然而，如果需要高精度（如在节省化学成本的情况下），则可以使用在第322次迭代训练的模型，其精度高达98%（图8）。为了定量分析精度和召回率之间的权衡，我们研究了YOLOv5m模型在不同迭代次数下的性能。在第525次迭代中，该模型的最大召回率为0.77，这对于最小化遗漏任何VC植物的可能性至关重要。高召回率表明模型能够检测到大多数VC植物，从而防止潜在的棉铃象鼻虫侵害。然而，这以降低精度为代价，即存在更多的假阳性。在我们的情境中，这种权衡是可以接受的，因为我们的主要目标是确保不遗漏任何VC植物。模型的整体性能由mAP@50表示，在第613次迭代中，其最大值为81%（图8）。通过调整Yan等人[40]提到的置信阈值，可以提高训练模型的性能。此外，我们还通过使用图像增强技术从有限数量的数据集中生成了521张图像，即34张图像。这样，本研究中报告的训练模型可以被视为相对更具泛化能力[52]。虽然图像增强技术可以提高训练集的多样性，但它们可能无法完全捕捉到现实场景中的变化，比如由于不同的光照条件、植物生长阶段和环境背景所导致的变化。为了提高模型的泛化能力，可以实施几种策略，其中之一是通过从各种田间条件和生长阶段收集更多标注的图像来增加训练数据集的大小和多样性，这有助于模型学习更稳健和可泛化的特征。

4.2. 蚁群算法在最佳飞行路径和点喷中的应用

无人机飞行参数，如飞行速度、飞行高度、偏航角等，被假设为固定值，并基于对信息素和启发式参数的等权重分配生成最优路径，如图12所示。这些假设与马等人[53]所做的假设类似。在我们的实现中，唯一使用的约束是两个GPS位置之间的距离。然而，在其他研究中，参数如

偏航角度被用作一些约束条件[44]。蚁群算法本质上是随机的，基于随机生成的变量生成足够好的解决方案，但没有全局最优解[54]。因此，图14所示的最优路径可能不是全局最优解，而只是基于我们研究中使用的随机生成变量的一个解决方案。在本研究中，蚁群算法使用的参数值可能不是最佳值，但经过合理组合的测试。从过去的研究中发现，蚁群算法对蒸发率高度敏感，蒸发率被认为相当于学习率；因此，选择正确的参数值至关重要[55,56]。Ojha等人[55]发现，当蒸发率超过0.5时，蚁群算法的性能会下降。因此，我们选择在研究中使用0.5作为蒸发率（图12）。从10个VC植物位置共生成了11个节点，如表2-左和表2-右所示。节点1和节点11表示相同的地点，无人机从这里开始并结束其穿越最优路径的旅程（图12-14）。

5. 结论与未来工作

5.1. 结论

本文介绍了我们在基于YOLOV5m的CV算法在玉米田间苗期检测玉米植株的成功应用。我们的方法与以往的方法相比，分类和检测精度有所提高。此外，我们还证明了低分辨率成像传感器，结合辐射和伽马校正等预处理算法，可以有效地检测玉米植株，同时对光照和环境条件的变化具有较强的稳健性。

此外，我们通过将训练好的 YOLOV5m 模型部署到安装在喷雾无人机上的计算平台上，展示了其在近实时检测方面的实用性。这种实现方式能够在全尺寸图像（1207×923 像素）中检测到 VC 植物，在 NVIDIA Jetson TX2 开发板的 Pascal GPU 上，调整后的平均推理速度约为 0.4 帧每秒。

此外，我们成功地将检测到的玉米植株的像素级边界框中心坐标转换为GPS坐标。这使得我们能够使用蚁群算法生成最优飞行路径，以模拟点喷应用。总的来说，我们的研究已经开发出一种计算机视觉算法，能够对玉米田中的玉米植株进行近实时检测，并利用定制的喷药无人机进行点喷应用。通过这项工作，我们创建了一个系统，有可能以更低的成本加快田间杂草防治工作。

5.2. 局限性与未来工作

在我们的研究中遇到了几个限制。首先，我们只有34张原始图像包含VC植物。为了解决这个问题，我们对整个原始图像集应用了图像增强技术。尽管这种方法可能导致数据泄露，但索尔·巴雷托[57]已经证明，在数据有限的情况下，它仍然可以有效地使用。另一个限制是在我们的研究中观察到的，即在Jetson TX2平台上部署训练好的YOLOv5m模型时，VC检测的推理速度很慢。通过适当配置硬件并部署YOLOv5的轻量级版本以实现实时应用，可以缓解这一限制。最后，我们的研究中没有在田间条件下验证点喷应用的模拟结果。因此，我们的未来工作将包括进行田间试验以验证模拟结果。

remote sensing

MDPI

Article

AI-Driven Computer Vision Detection of Cotton in Corn Fields Using UAS Remote Sensing Data and Spot-Spray Application

Pappu Kumar Yadav [1,*], J. Alex Thomasson [2], Robert Hardin [3], Stephen W. Searcy [3], Ulisses Braga-Neto [4], Sorin C. Popescu [5], Roberto Rodriguez III [6], Daniel E. Martin [7] and Juan Enciso [3]

1 Department of Agricultural & Biosystems Engineering, South Dakota State University, Brookings, SD 57007, USA
2 Department of Agricultural & Biological Engineering, Mississippi State University, Starkville, MS 39762, USA
3 Department of Biological & Agricultural Engineering, Texas A&M University, College Station, TX 77843, USA; s-searcy@tamu.edu (S.W.S.)
4 Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX 77843, USA
5 Department of Ecology & Conservation Biology, Texas A&M University, College Station, TX 77843, USA
6 Spatial Data Analysis and Visualization Laboratory, University of Hawaii at Hilo, Hilo, HI 96720, USA
7 Aerial Application Technology Research, U.S.D.A. Agriculture Research Service, College Station, TX 77845, USA
* Correspondence: pappu.yadav@sdstate.edu

Abstract: To effectively combat the re-infestation of boll weevils (*Anthonomus grandis* L.) in cotton fields, it is necessary to address the detection of volunteer cotton (VC) plants (Gossypium hirsutum L.) in rotation crops such as corn (*Zea mays* L.) and sorghum (*Sorghum bicolor* L.). The current practice involves manual field scouting at the field edges, which often leads to the oversight of VC plants growing in the middle of fields alongside corn and sorghum. As these VC plants reach the pinhead squaring stage (5–6 leaves), they can become hosts for boll weevil pests. Consequently, it becomes crucial to detect, locate, and accurately spot-spray these plants with appropriate chemicals. This paper focuses on the application of YOLOv5m to detect and locate VC plants during the tasseling (VT) growth stage of cornfields. Our results demonstrate that VC plants can be detected with a mean average precision (mAP) of 79% at an Intersection over Union (IoU) of 50% and a classification accuracy of 78% on images sized 1207 × 923 pixels. The average detection inference speed is 47 frames per second (FPS) on the NVIDIA Tesla P100 GPU-16 GB and 0.4 FPS on the NVIDIA Jetson TX2 GPU, which underscores the relevance and impact of detection speed on the feasibility of real-time applications. Additionally, we show the application of a customized unmanned aircraft system (UAS) for spot-spray applications through simulation based on the developed computer vision (CV) algorithm. This UAS-based approach enables the near-real-time detection and mitigation of VC plants in corn fields, with near-real-time defined as approximately 0.02 s per frame on the NVIDIA Tesla P100 GPU and 2.5 s per frame on the NVIDIA Jetson TX2 GPU, thereby offering an efficient management solution for controlling boll weevil pests.

Keywords: boll weevil; volunteer cotton (VC); remote sensing; computer vision (CV); YOLOv5; unmanned aircraft systems (UASs); spot-spray

Citation: Yadav, P.K.; Thomasson, J.A.; Hardin, R.; Searcy, S.W.; Braga-Neto, U.; Popescu, S.C.; Rodriguez, R., III; Martin, D.E.; Enciso, J. AI-Driven Computer Vision Detection of Cotton in Corn Fields Using UAS Remote Sensing Data and Spot-Spray Application. *Remote Sens.* **2024**, *16*, 2754. https://doi.org/10.3390/rs16152754

Academic Editor: Jochem Verrelst

Received: 1 May 2024
Revised: 28 June 2024
Accepted: 24 July 2024
Published: 27 July 2024

1. Introduction

The cotton boll weevil (*Anthonomus grandis* L.) is an insect pest that has caused more than USD 23 billion in losses to the U.S. cotton industry since migrating from Mexico in the 1890s [1]. It continues to be a matter of concern for the U.S. cotton industry, particularly in Texas, even though it has been eradicated from most of the U.S. Therefore, there is a continued need for the activities of the Texas Boll Weevil Eradication Foundation (TBWEF) as per the latest report of Sunset Advisory Commission [2]. The TBWEF has divided the state into 16 eradication zones in which the Lower Rio Grande Valley (LRGV) is still actively

functioning as it remains the region most prone to boll weevil re-infestation each year due to its tropical climatic conditions [2] and proximity to the Mexico border. The vulnerability of the LRGV is due to several factors, one of which includes its subtropical climate with hot and humid summers and mild to cool winters without freezing conditions, allowing cotton plants to grow and produce fruit year-round, providing a continuous food source for boll weevils. Additionally, tropical storms from the Gulf of Mexico commonly carry boll weevils from Tamaulipas State in Mexico to the LRGV [3]. In 2019 alone, 46,000 boll weevils were captured by the foundation, indicating the severity of the problem and a continued need for the functioning of the TBWEF. Without the eradication efforts by TBWEF, boll weevils would have caused an annual loss of USD 200 million to Texas farmers [4]. Recent reports indicate that the TBWEF's efforts have significantly reduced the number of boll weevils in Texas [5]. As per the latest weekly report published by the TBWEF, the season-long trap average of boll weevils captured has decreased by approximately 99.91% from the year 2020 to the year 2024 for the Lower Rio Grande Valley [5].

Cotton (*Gossypium hirsutum* L.) is commonly planted in rotation with crops like corn (*Zea mays* L.) and sorghum (*Sorghum bicolor* L.). In climatic areas like the LRGV, cotton seeds can survive year-round, and thus, seeds in cotton that might have fallen during the harvest in the previous year can grow among corn and sorghum plants in the present year [3,6,7]. Such plants are called volunteer cotton (VC) plants, which essentially act as weeds that can emerge in fields of grain crops like corn, sorghum, soybean, and wheat. The necessity of removing VC from various crops in the state's production regions stems from two main factors: yield loss due to competition and boll weevil control challenges. Volunteer cotton plants compete with primary crops, reducing yields, with farmers generally aiming for 80–90% control using herbicides. In South and Eastern Texas, where the Texas Boll Weevil Eradication Program (TBWEP) is active, the presence of volunteer cotton significantly complicates and increases the cost of boll weevil control as boll weevils use cotton plants as hosts. Consequently, the Texas Department of Agriculture enforces a zero-tolerance policy for hostable non-commercial cotton plants (6–8 leaves or larger) in quarantined zones to prevent infestations. To comply with these regulations and protect crop yields, farmers must achieve complete control of volunteer cotton through timely and appropriate herbicide applications.

To minimize the likelihood of boll weevil re-infestation, the TBWEF uses pheromone traps to detect boll weevils and pesticides to eliminate them. As part of boll weevil mitigation efforts in the LRGV region, fields with rotation crops are inspected for the presence of VC plants at the edges of fields on a weekly basis. When VC plants are detected, the number of pheromone traps increases. In addition to inspecting for the presence of VC plants, pheromone traps are also inspected for the presence of boll weevils. If at least one is found at the edge of a field, then the entire field is sprayed with a pesticide, commonly Malathion ($C_{10}H_{19}O_6PS_2$) ULV (FYFANON® ULV AG) at rates between 0.56 and 1.12 kg/ha [8]. Uniform spraying is utilized because VC plants growing in the middle of corn and sorghum fields remain undetected and so cannot be sprayed individually. Spraying entire fields results in both increased management costs and environmental concerns as well as destroying many beneficial insects.

Uniform spray applications can be avoided if VC plants growing in the middle of corn and sorghum fields are detected and precisely located so that spot-spray-capable unmanned aircraft systems (UAS) can be deployed. Detecting VC plants before they reach the pinhead square stage and precisely spraying them with herbicides can eliminate the plants and minimize the need for applying Malathion. However, some VC plants may survive due to herbicide tolerance or inaccuracies in detection, in which case Malathion application later in the season could be needed. To detect VC plants either early or late in the growing season, remote sensing with images collected by UASs along with computer vision (CV) algorithms using state-of-the-art convolution neural network (CNN) architectures like Mask R-CNN [9], YOLOv3 [3,10,11], YOLOv5 [12], etc. can be used. Since its release in 2021, YOLOv5 has become popular in CV applications and has been used to detect

various objects like apples [13], face masks [14], safety helmets [15], etc. Due to its higher detection accuracy and faster inference speed compared to other traditional object detection algorithms, YOLOv5 was selected for this study as the most viable model for near-real-time detection. Originally YOLOv5 was released in four different variants: YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. However, YOLOv5m was chosen as the desirable variant based on our previous research [16].

In addition to detecting VC plants, the geographic coordinates of the detected plants are needed for precise spray application. Therefore, geotagged UAS-based imagery was used in this study. High-quality RGB (Red, Green, Blue) cameras have commonly been used for object detection with YOLOv3 and YOLOv5 [13,14,17]. In most cases, radiometric correction was not employed. Remote sensing imagery without radiometric correction is susceptible to varying environmental conditions including illumination, atmospheric light scattering, sensor noise, etc. [18]. Images that are not radiometrically corrected have digital numbers (DN) that do not represent actual surface reflectance [19–21]. Hence, radiometric correction was conducted in this study.

YOLO-series object detection algorithms generate bounding boxes (BBs) around the objects of interest present in the images. The locations of these BBs are based on their pixel-wise distances from the top-left corners of images [10,22,23]. Based on the BB coordinates, central coordinates of each BB can be determined. Pixel-wise coordinates are not useful for the path planning of spot-spray-capable UASs, so a method of converting pixel-wise BB coordinates into GPS coordinates is necessary so that they can be used for path planning for spot-spraying detected VC plants.

UAS flight times are limited by battery capacity, so an optimal flight path is a necessity to efficiently spray VC plants. Optimal path planning can be conducted based on the travelling salesman problem (TSP), in which the goal is to determine the shortest route for the UAS to spot-spray each of the detected VC plants and then return to the starting point [24,25]. Different algorithms have been tested for this, some of which include the genetic algorithm used by Moon et al. [26] and Shivgan and Dong [25], ant colony optimization (ACO) used by Dorigo et al. [27], etc. In this study, ACO was used because of its simplicity in implementation, feasibility, and faster speed to generate high-quality solutions [27]. The determined optimal flight path was tested by simulating the UAS with DroneKit-Software (3.3.0) In The Loop (SITL) (3D Robotics, Berkely, CA, USA), MAVProxy [28], and Mission Planner (Ardupilot Development Team and Community). DroneKit-SITL provides an application programming interface (API) to run Python-based applications on companion computers of UASs whereas MAVProxy and Mission Planner are ground control station (GCS) software programs to control and simulate UAS applications [29]. The communication between GCS and UASs is achieved through a binary serial telemetry protocol called MAVLink [30].

The overall goal of this research was to develop a CV algorithm for detecting VC plants in a corn field and use the detected locations for optimal spot-spray applications. The four specific objectives were (i) to develop a computer vision (CV) algorithm with YOLOv5m to detect VC plants in a corn field with radiometrically (reflectance calibrated) and gamma-corrected, relatively low-resolution (1.2 Megapixel), multispectral aerial imagery; (ii) to convert the pixel-based bounding box coordinates of detected VC plants into geographic coordinates; (iii) to use the detected geographic coordinates of VC plants to generate an optimal flight path with the ACO algorithm; and (iv) to simulate a spot-spray UAS with DroneKit-SITL and the MAVProxy and Mission Planner GCS software based on the generated optimal flight path.

2. Materials and Methods

2.1. Experiment Site

This study was conducted at a corn field (Figure 1; 96°25′45.9″W, 30°32′07.4″N) of roughly 5.9 hectares (14.6 acres) at the Texas A&M AgriLife Research farm near College Station, Texas. The majority of the soil in the experimental plot is Weswood silty clay loam,

followed by Yahola fine sandy loam and Belk clay [31]. The corn plants in the field were in the second leaf (V2) vegetative state when 90 cotton seeds each of two varieties (Phytogen 340 W3FE, CORTEVA Agriscience, Indianapolis, IN, USA and Deltapine 1646 B2XF, Bayer AG, Leverkusen, North Rhine-Westphalia, Germany) were planted at randomized locations among the corn plants to mimic the presence of VC plants in the field. Some were planted in line with corn plants while others were planted in the furrow middles.

Figure 1. Experiment field located at Texas A&M University farm near College Station, TX in Burleson County (96°25′45.9″W, 30°32′07.4″N) where cotton plants were planted in the middle of corn field to mimic the presence of volunteer cotton plants.

2.2. Image Data Acquisition

A five-band (Blue, Green, Red, Near-Infrared, RedEdge) multispectral camera—RedEdge-MX (AgEagle Aerial Systems Inc., d/b/a MicaSense, Wichita, Kansas; Figure 2)—was mounted on a customized UAS (Figure 2) for collecting aerial imagery of the corn field with cotton plants when the corn plants had reached tassel vegetation stage (VT) (Figure 3). The reason for choosing this camera instead of conventional RGB camera was because its images are geotagged and can be used for other advanced image processing techniques like radiometric correction using open-source code provided by the manufacturer; moreover, it gave the flexibility to use other bands like NIR and RedEdge when required. The central wavelength and bandwidth of each spectral band are shown in Table 1. The images were acquired at this growth stage of corn plants because we wanted to test the worst-case scenario in which corn canopies are relatively larger and taller than those of VC plants. This was shown by another study in which the lowest detection accuracy was for corn fields when they had reached the VT growth stage [7]. The UAS (Hylio AG-110; Hylio Inc., Richmond, TX, USA) was originally designed for broadcast spray applications. Data were collected on 14 May 2021 between 11:00 a.m. and 2:00 p.m. central daylight-saving time (CDT) at an altitude of nearly 4.6 m (15 feet) above ground level by using auto exposure camera setting. This resulted in an approximate ground sampling distance (GSD) of 0.34 cm/pixel. This altitude was chosen for two main reasons: one was for higher image resolution and GSD while the other was so that efficient spot-spray application could be performed from this altitude without causing too much of drift in the spray droplets. Since the customized UAS was not designed for aerial surveying, there was no software interface available to capture images based on overlap settings. Therefore, images were captured

on a timer-based setting enabling an image to be captured every second. This resulted in many unused and distorted images due to unavailability of overlap settings and vibrations caused due to the design aspect of the sprayer UAS (generally, spray-capable UASs have more vibrations as compared to surveying UASs). The UAS had a payload capacity of 10 kg/L. The UAS was flown at a speed of 2 m per second.

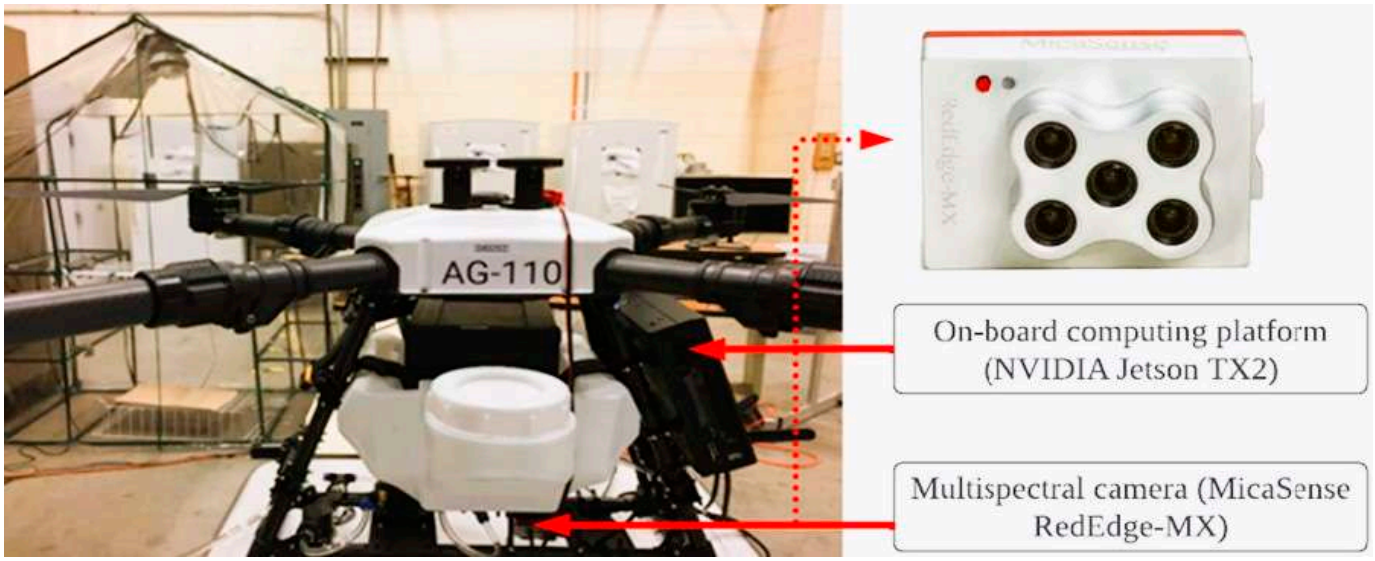

Figure 2. A customized sprayer UAS (broadcast sprayer converted to spot sprayer) with RedEdge-MX multispectral camera for capturing aerial imagery and NVIDIA Jetson TX2 computing platform [7].

Figure 3. (**A**) The customized spot-sprayer UAS flying over an experimental corn field (containing some cotton plants planted to mimic as volunteer cotton (VC) plants) capturing five band multispectral images; (**B**) RGB (Red, Green Blue) composite image showing a section of experimental plot where corn at vegetative tassel state (VT) and some cotton plants mimicking as VC plants can be seen.

Table 1. Center wavelength and full width at half-maximum (FWHM) bandwidth of each spectral band of MicaSense RedEdge-MX camera.

Spectral Band	Central Wavelength (nm)	FWHM Bandwidth (nm)
Blue	475	20
Green	560	20
Red	668	10
NIR	840	40
RedEdge	717	10

An onboard computing platform, the Jetson TX2 (NVIDIA Corporation, Santa Clara, CA, USA) development board that consists of a Pascal graphics processing unit (GPU), was mounted on the customized UAS with the intent of near-real-time detection of VC plants in corn fields (Figure 2). The Pascal GPU is low-cost, fast, and widely used as an embedded artificial intelligence computing device. It consists of 256 NVIDIA Compute Unified Device Architecture (CUDA; NVIDIA, Santa Clara, CA, USA) cores with 8 GB of RAM and 32 GB of storage capacity [32].

2.3. Manufacturer Recommended Corrections

The individual band images collected by RedEdge-MX camera were corrected based on manufacturer's recommendations [33] using the reflectance panel of type RP 04 (Figure 4) images taken on the day of data collection just before the flight.

Figure 4. Reflectance panel of type RP 04 image with blue band sensor of RedEdge-MX camera taken on the day of flight.

According to the manufacturer's factory calibration (in which absolute reflectance values were plotted along wavelengths ranging from 400 nm to 850 nm), the reflectance values corresponding to blue, green, red, NIR, and RedEdge were 0.60, 0.61, 0.61, 0.60, and 0.56, respectively, for the panel that was used in the study. As per the manufacturer's process, an area of the Lambertian panel from reflectance panel image was extracted and its radiance value was converted to the scale of reflectance value, which was then applied to the whole image to convert it into reflectance images. Source codes for both radiometric and gamma corrections were used from the GitHub (GitHub, Inc., San Francisco, CA, USA) repository of MicaSense [33]. The original source code was modified based on the reflectance values for the panel used and for generating RGB aligned images for each captured image from the field data collection as seen in Figure 3. RGB images were generated because YOLOv5 network architecture accepts input images with three channels (usually RGB images). Python version 3.8.12 (Python Software Foundation, Wilmington, DE, USA) was used in Spyder integrated development environment (IDE) version 5.2.2. The entire process involves normalizing images by gain and exposure settings and then converting

them into radiance followed by reflectance. The gain and exposure settings were used to normalize the raw image data, correcting for differences in exposure time and sensor sensitivity and allowing for accurate conversion to radiance values. The normalization of images by exposure and gain settings is crucial for converting raw images to radiance. Exposure time, representing the duration the camera sensor is exposed to light, and gain, derived from the ISO setting, are obtained from the image metadata. The raw image undergoes correction to account for dark pixel offset, vignette effects, and row gradient inaccuracies. Following this, the corrected image is normalized using the exposure time and gain, ensuring consistency across different images. This normalized image data is then converted to radiance units ($W/m^2/nm/sr$) by scaling it with the gain-exposure product and applying the radiometric calibration coefficient, adjusted for the image bit depth. This process ensures accurate and comparable radiance values essential for reliable reflectance mapping. After this, in the study, unsharp mask (an image sharpening technique) was applied as enhancement technique to improve visual sharpness, and then, gamma correction was applied to make the enhanced images appear brighter and visually closer to what our eyes see [33]. For these, the original values present in the GitHub source code were used. The unsharp mask technique uses linear filter to add a fraction of high-pass-filtered input image to the original image that helps sharpen the original image by filtering out the noise [34]. Similarly, gamma correction is an image enhancement technique that is used to minimize the effect of non-linearity of the imaging sensors, thereby making the images appear brighter with enhanced contrast and visually closer [35–37].

2.4. YOLOv5

Since its release in June of 2020, YOLOv5 [38] has become a popular algorithm for object detection in CV applications. YOLOv5 was originally released in four different variants, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, with the subscript based on the network depth and number of parameters used. Here, *s*, *m*, *l*, and *x* represent small, medium, large, and extra-large variants of the YOLOv5 network, respectively. The YOLOv5 network comes with pretrained weights from training on the Common Objects in Context (COCO) [39] dataset, which consists of 80 different classes, such that YOLOv5 is pretrained to detect 80 different classes. In this study, the network was customized to detect a single class, "*vc*", for VC plants. The architecture of YOLOv5 comprises three main components: the Backbone, the Neck/PANet, and the Head/Output (Figure 5). The Backbone is a convolutional neural network (CNN) responsible for aggregating fine-grained images. The Neck consists of feature aggregation layers that combine features to construct the feature pyramid network. These generated feature maps are then passed on to the Head or Output network. The Output network handles the final detection phase of the model. It applies anchor boxes to the feature map obtained from the previous layer and produces a vector containing the category probability of the target object, the object score, and the position of the bounding box that surrounds the object [40].

2.5. Image Data Preparation

Among the radiometrically and gamma-corrected RGB images, only the ones containing at least a VC plant with corn in the background were chosen and then image augmentation was applied using Augmenter Python library [41]. For this, *rotate*, *flip_left_right*, *zoom_random*, and *flip_top_bottom* were used with probability values of 0.80, 0.40, 0.60, and 0.80, respectively. These values were chosen so that each time an image was passed through the augmentation pipeline, 80% of the time, *rotate* operation was applied; 40% of the time, *flip_left_right* was applied; 60% of the time, *zoom_random* was applied; and 80% of the time, *flip_top_bottom* was applied. In this way, we were able to generate a total of 521 RGB images from a total of 34 original images (containing at least one VC plant) that were radiometrically and gamma-corrected, and each of them were of the original size, 1207 × 923 pixels. Out of these, 417 images (80%) were used for training, 77 (15%) were

used for validation, and 27 (5%) were used for testing. The percentages of data split were chosen arbitrarily.

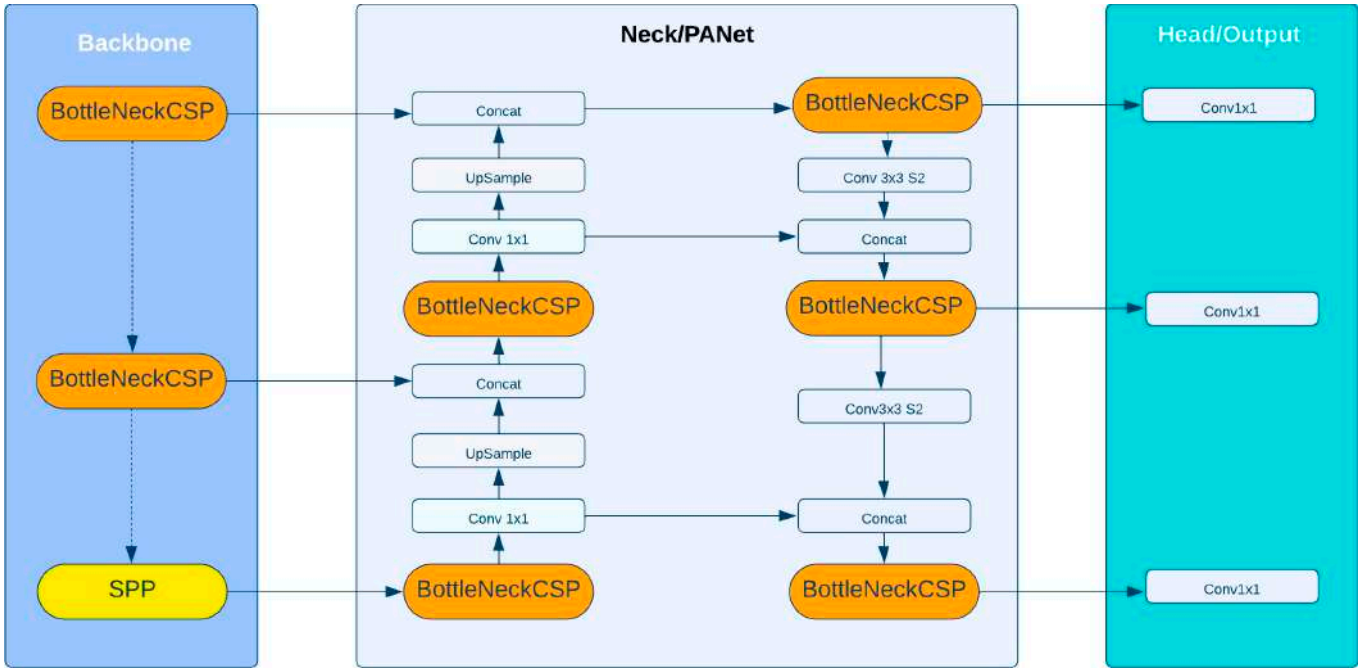

Figure 5. General overview of YOLOv5 network architecture.

2.6. YOLOv5 Training

The YOLOv5m source code was obtained from the GitHub repository of Ultralytics Inc. [12]. The PyTorch framework (Facebook AI Research Lab, Melno Park, CA, USA) with torch version 1.10.0 and Compute Unified Device Architecture (CUDA) version 11.1.0 (NVIDIA, Santa Clara, CA, USA) were used to train the YOLOv5m model on Tesla P100-PCIE-16 GB (NVIDIA, Santa Clara, CA, USA) GPU using the Google Colab (Google LLC, Melno Park, CA, USA) AI platform. The model was trained with the original hyperparameter values with initial learning rate of 0.01, final learning rate of 0.1, momentum of 0.937, weight decay of 0.0005, and intersection over union (IoU) threshold of 0.50 for a total of 621 iterations. This essentially implied that YOLOv5m had reached the convergence within the 621 iterations.

2.7. Bounding Box Coordinate Conversion

The first step involved in this process was to extract the top-left-corner coordinates from the geotagged images of RedEdge-MX camera during the processes of radiometric and gamma corrections, image enhancement, and RGB band alignments by modifying the original Python script obtained from the GitHub repository of MicaSense [33]. The extracted GPS coordinates of each geotagged image were stored in a comma-separated variable (CSV) file. The YOLOv5 *detect.py* Python script was modified to extract central coordinates of each detected bounding box, which were stored in a separate CSV file. Another Python script was developed to utilize both the CSV files and then convert the pixel-wise BB of the detected VC plants into GPS coordinates. In the Python script, GSD was first converted into meter/pixel format and the decimal coordinates were converted into Universal Transverse Mercator (UTM) format, from which northing and easting values were extracted, and then, pixel-based central coordinates were converted into UTM-based GPS coordinates using UTM zone of 14 during the conversion process [42].

Potential sources of error in this conversion process included inaccuracies in the initial geotagging of images, variations in GSD, and the precision of the UTM conversion. The accuracy of the detected bounding box coordinates was also influenced by the performance

of the YOLOv5 model, which may have had a margin of error in object detection. Additionally, any slight misalignments or distortions in the image processing steps could have propagated through to the final GPS coordinates. It was crucial to consider these factors and apply appropriate error correction and validation steps to ensure the reliability of the GPS coordinates derived from pixel-wise bounding boxes.

2.8. Optimal Flight Path with ACO Algorithm

ACO algorithm is state-of-the-art algorithm for many problems like vehicle routing, open-shop scheduling, and sequential ordering problems [27]. It is based on the fact that when ants move along a path from their colony to a food source, they deposit pheromones that evaporate over time. This means ants travelling through longer paths have less intense pheromone deposits, but the pheromone intensity is much higher along the shortest path. In our used case application, VC plant locations were the food source that the artificial ants would find through the shortest possible route. In many past cases, ACO has proven to be widely accepted algorithm for determining optimal flight paths for UASs [43,44]. Source code to implement ACO was used from the GitHub repository of *fabien-brulport* [45]. The original source code was modified to generate a CSV file containing GPS coordinates (latitude, longitude) in the order of nodes generated for the optimal route as the output from the ACO algorithm. In our study, we set the values of pheromone weight, heuristic weight, and evaporation rate as 2.01, 1, and 0.5, respectively, to achieve the shortest distance after trying with different combinations of values. Similarly, the values for number of agents, i.e., number of artificial ants (i.e., equivalence of number of UASs), and number of iterations were 1 and 100, respectively. Mathematically, at any given time *t*, the probability that ant *k* choses a path from *i* to *j*, as explained by Zhang et al. [44], is given by the following formula:

$$P_{ij}^{(k)}(t) = \left\{ \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{S \subset allowed_k}[\tau_{is}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}} \text{if } j \in allowed_k \; ; \; 0 \text{ otherwise} \right\} \quad (1)$$

Here, α and β are pheromone and heuristic weights, respectively. η_{ij} (t) and τ_{ij} (t) are visibility and quantity of pheromone between points i and j. $S \subset allowed_k$ is a set of all the possible points in a path that the ant can choose from. The pheromone amount determines the visibility, which, in turn, determines the probability of choosing a path. Therefore, higher pheromone quantity increases the visibility of a particular path between points i and j and, hence, the probability of choosing the path increases.

2.9. Spot-Spray UAS Simulation

Once the geographic coordinates were saved in optimal order corresponding to nodes obtained from ACO, the DroneKit-SITL (3D Robotics, Berkeley, CA, USA) Python package was used to simulate flight paths of UAS with a Python script and monitored on two ground control stations (GCSs): Mission Planner version 1.3.76 [46] and MAVProxy version 1.8.45 [28]. The MAVLink [30,47] protocol was used to communicate between the two GCSs and DroneKit-SITL. This communication was accomplished by using transmission control protocol (TCP) for the master port and then port-forwarding the output to three user datagram protocol (UDP) ports. The steps used to implement the entire process are represented in Algorithm 1 as shown in below.

A Python script was used to read a CSV file that contained GPS coordinates of all the nodes in order, generated by the ACO algorithm, which corresponded to the locations of the detected VC plants for spot-spraying. After the UAS simulation was performed successfully, we were also able to upload the CSV file with optimal flight paths on AgroSol version 2.105.0 GCS, developed by Hylio (Hylio, Inc., Richmond, TX, USA). The AgroSol software provided interface to control the spot-spray UAS that was customized. The flowchart in Figure 6 shows the entire workflow starting from data collection to spot-spray UAS simulation.

Algorithm 1 Spot Spray Simulation Algorithm

Require: CSV file with GPS coordinates of detected VC plants
Ensure: Simulated flight path based on ACO algorithm on Mission Planner GCS

```
Procedure SpotSpraySimulationAlgorithm:
Open two terminal windows from Anaconda environment;
while paths set to the directory where MAVLink was installed do
    if terminal window 1 then
    dronekit-sitl copter --home=30.534351,-96.431239,0,180
    --model=copter;
    end if
    mavproxy --master tcp:127.0.0.1:5760 --sitl 127.0.0.1:5501
    --out udp:127.0.0.1:14550 --out udp:127.0.0.1:14551 --out
   udp:127.0.0.1:14552;
end while
Open Mission Planner GCS;
while mode == guided do do
   if UDP port ==   14551 then
      Open Spyder IDE and execute python script "vc spot spray.py";
   else
      end
   end if
end while
end
```

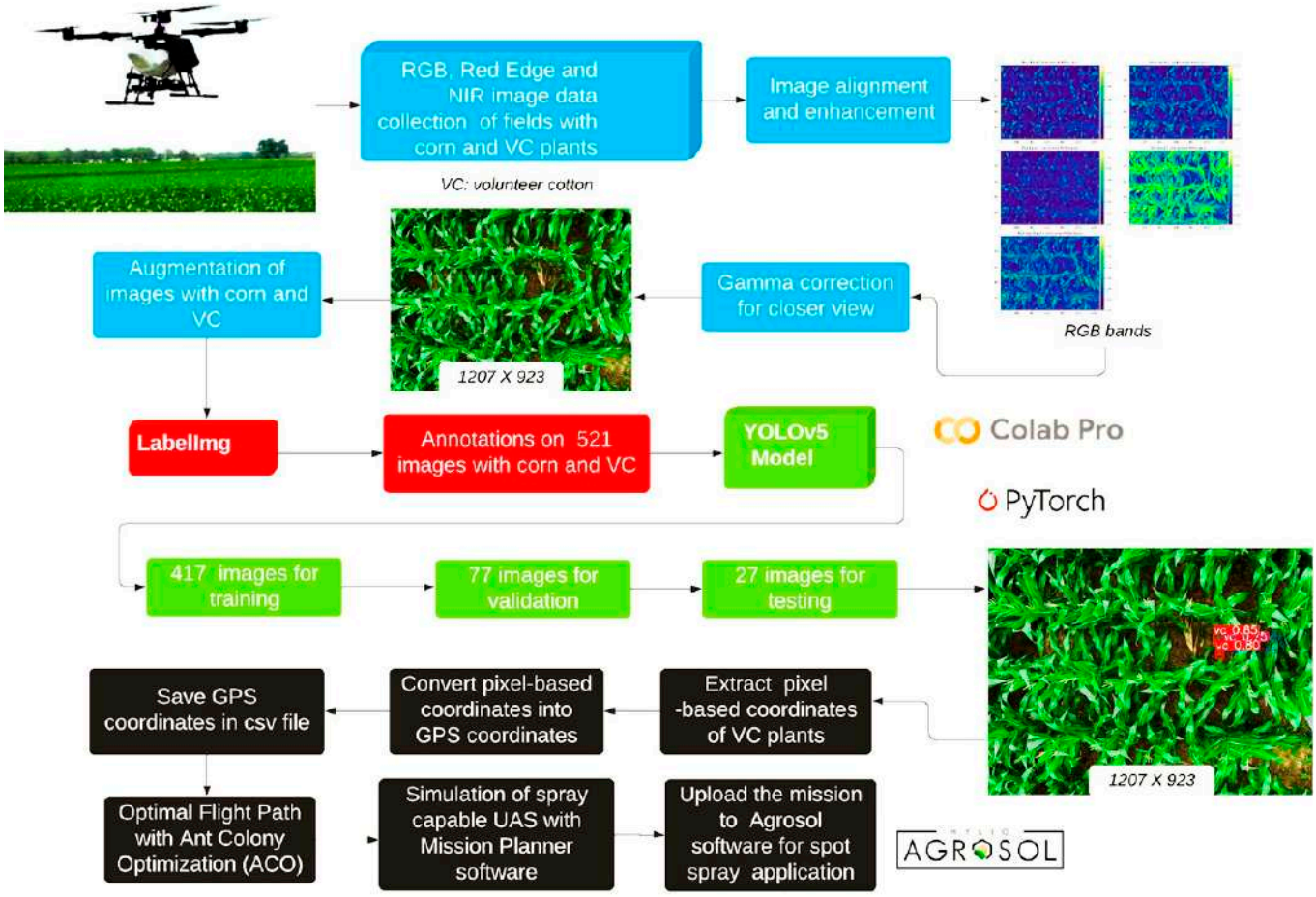

Figure 6. A flowchart that shows complete workflow representing each step used in this study.

3. Results

3.1. CV Algorithm with YOLOv5m for Detecting VC Plants in a Corn Field on Radiometrically Corrected Aerial Imagery

Figure 7 shows graphs for box loss (IoU loss) and objectness loss on both training and validation datasets. Before the 600th iteration, both the losses had converged, and no improvement was seen beyond. The lowest value of box loss was found to be 0.028 at the 476th iteration while the lowest value for objectness loss, i.e., 0.009, was found at iteration number 613 on the training dataset. On the validation dataset, the lowest value of objectness loss was found to be 0.0094 at the 394th iteration. Similarly, the lowest value of box loss on the validation dataset was found to be 0.0296 at iteration number 370. In Figure 8, graphs of different performance metrices can be seen. It was found that the maximum value of precision reached around 0.98 at the 320th iteration while the maximum value for recall was found to be 0.77 at the 525th iteration. The most important metric out of all of these was mAP@0.50, whose maximum value, i.e., 0.81, was reached at iteration number 613. The mAP@0.50 value of 0.81 indicates that the model performs well when considering detections with an Intersection over Union (IoU) threshold of 50%, meaning the predicted bounding boxes overlap with the ground truth by at least 50%. This relatively high value suggests that the model can accurately detect and localize volunteer cotton (VC) plants under these conditions. At iteration number 521, the maximum value for mAP@0.50:0.95 was obtained, and this was found to be 0.33. The mAP@0.50:0.95 value of 0.33 highlights a more nuanced picture of the model's performance across a range of IoU thresholds (from 50% to 95%). This lower value suggests that while the model is effective at identifying and localizing objects at a 50% overlap, its performance decreases as the required overlap for a correct detection becomes more challenging. This indicates potential challenges in achieving precise localization, particularly in distinguishing VC plants from the background and other objects like corn plants, weeds, and soil. The precision-recall curve (PRC) resulted in an overall accuracy of nearly 79% and the maximum value of F1-score was found to be 0.76 at nearly the 40% confidence level (Figure 9A,B). The confusion matrix in Figure 10 shows that YOLOv5m was trained enough to classify VC plants with an accuracy of 78% and loss of 22% owing to the background class of corn plants, weeds, soil, etc. To mitigate these misclassifications, several strategies can be implemented. First, increasing the diversity and size of the training dataset can help the model better distinguish between VC plants and other similar objects. This can be achieved by collecting more annotated images under various lighting conditions and growth stages. Second, data augmentation techniques such as rotation, scaling, and color adjustments can improve the model's robustness to variations in the field environment.

Figure 10 shows some detection results from the trained YOLOv5m model that resulted in an average inference speed of 47 frames per second (FPS) on NVIDIA Tesla P100 GPU-16GB. This model was later deployed on the NVIDIA Jetson TX2 GPU that was mounted on the custom UAS (Figure 2) and resulted in an adjusted average inference speed of 2.535 s (~0.4 FPS) for images of size 640 × 640 pixels (Figure 11). The inference speed of the YOLOv5m model on the NVIDIA Jetson TX2 GPU is a critical factor for assessing the feasibility of real-time applications. Real-time processing typically requires a minimum speed of 30 FPS to ensure smooth and responsive performance in dynamic environments. The observed speed of 0.4 FPS indicates a significant lag, making it unsuitable for scenarios where immediate detection and response are necessary such as in real-time monitoring and intervention in agricultural fields. This low inference speed can be attributed to multiple reasons. One possible reason could be the settings of the GPU hardware on the Jetson TX2 board, which could not utilize its maximum computing power.

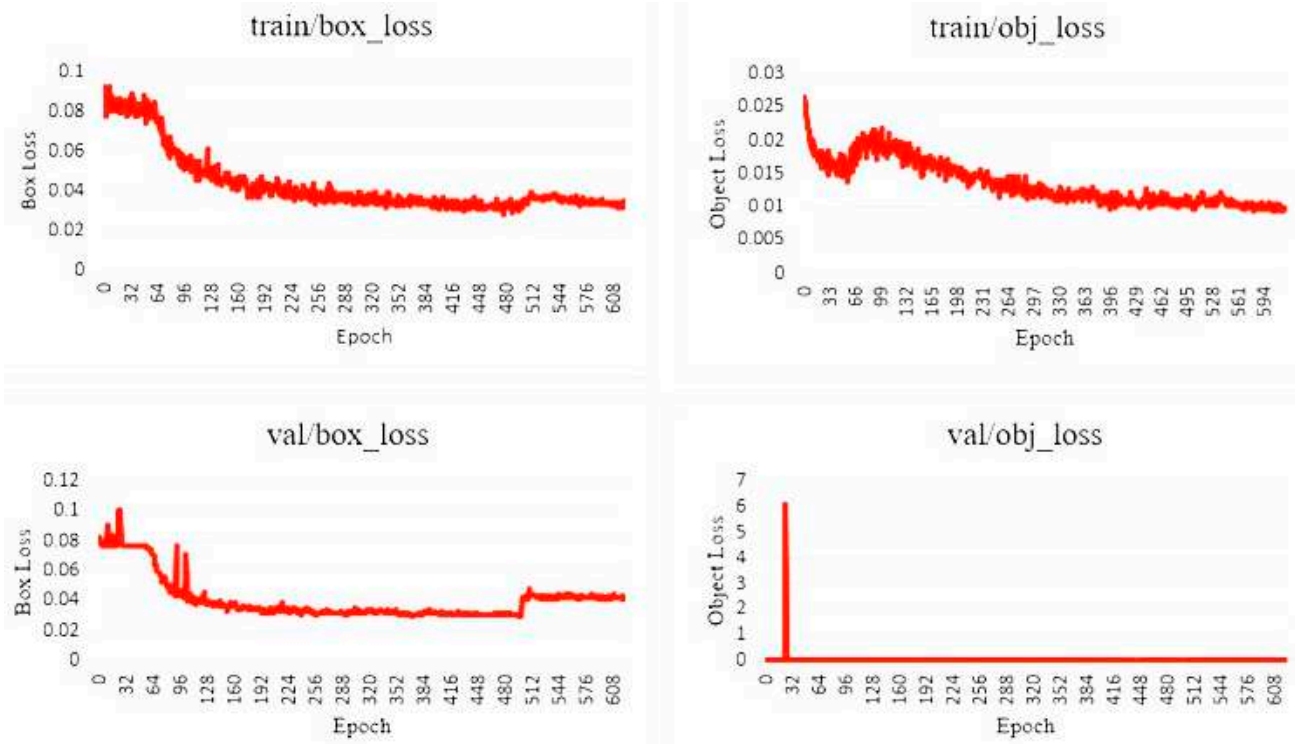

Figure 7. Different types of losses that were obtained during the training process of YOLOv5m on training and validation datasets.

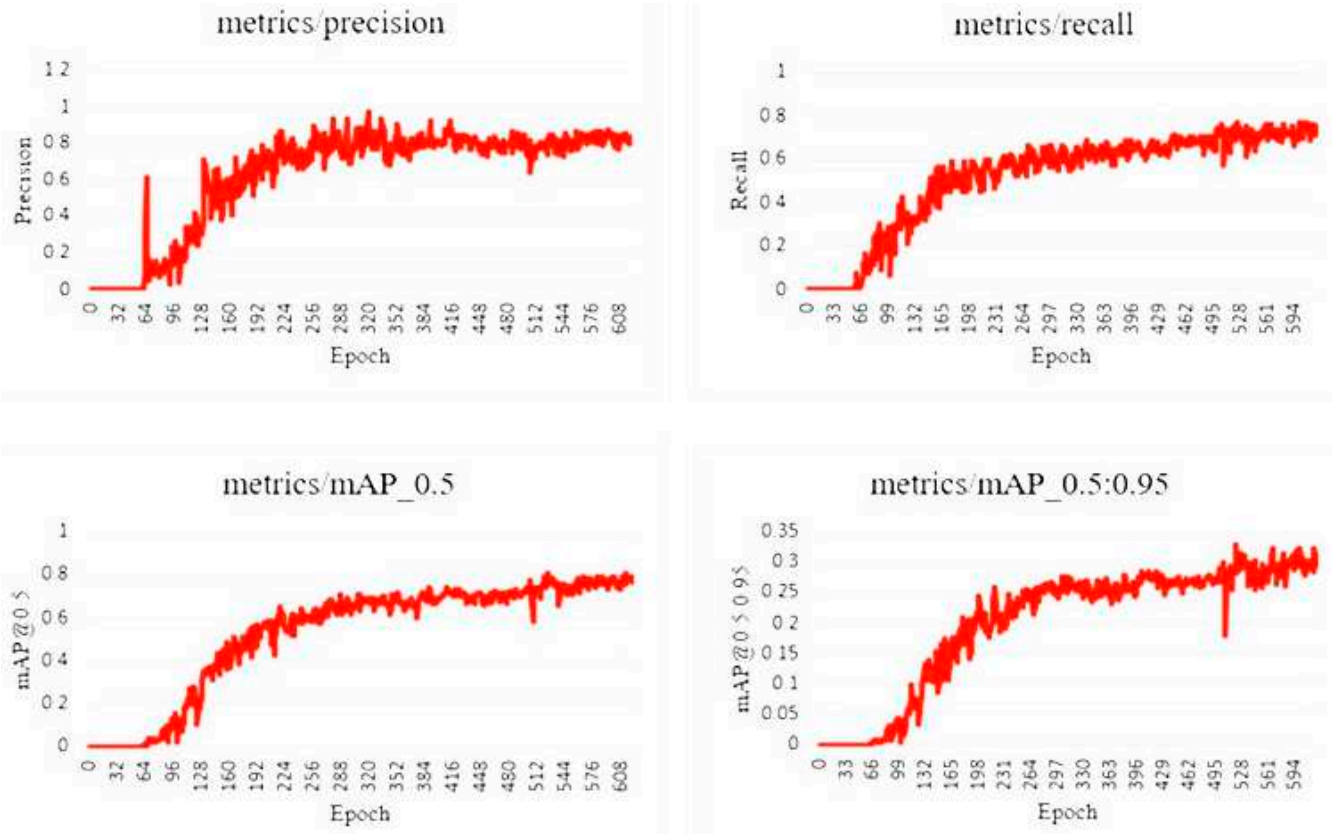

Figure 8. Different types of performance metrices that were obtained during the training process of YOLOv5m.

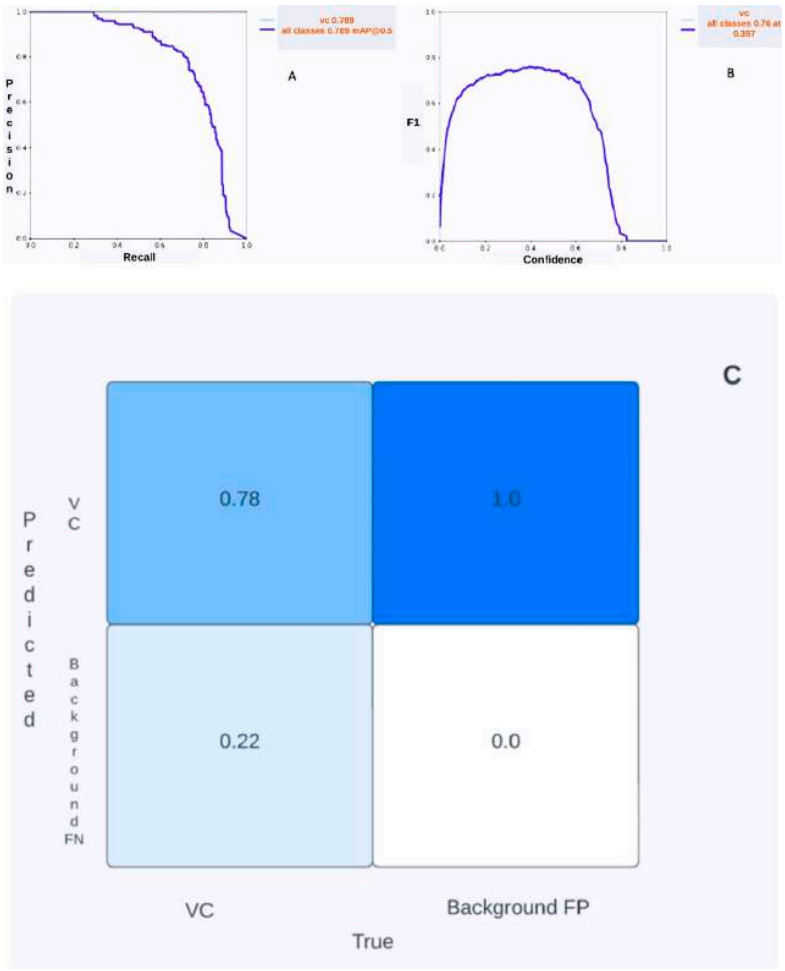

Figure 9. (**A**) Precision-recall plot, (**B**) F1-score vs confidence score plot, and (**C**) confusion matrix obtained after training YOLOv5m.

Figure 10. VC plants detected in the middle of a corn field within the red bounding boxes (BBs) by trained YOLOv5m model. The values associated with each BB show model's certainty that the bounding box contains an object of interest, i.e., VC plant.

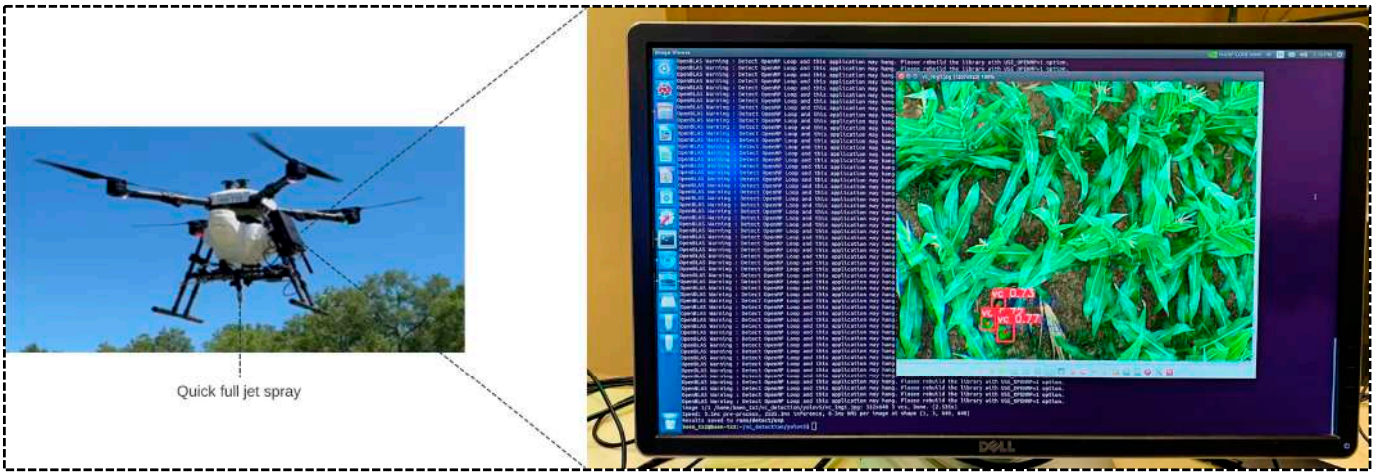

Figure 11. YOLOv5m detection of VC plants in a corn field by being deployed on NVIDIA Jetson TX2 mounted on a custom spot-spray-capable UAS.

3.2. ACO Algorithm to Determine Optimal Flight Path for Spot-Spray Applications

Ten random locations of VC plants in the experimental plot were chosen whose corresponding GPS coordinates can be seen on the left side of Table 2. The selection of ten random locations for volunteer cotton (VC) plants in the experimental plot was guided by the need to simulate realistic field conditions while ensuring the feasibility of spot-spraying in a single flight of the unmanned aerial system (UAS). These locations were chosen arbitrarily to cover a diverse range of the plot, ensuring that the UAS could efficiently manage the entire area in one flight. The criterion for selecting these locations involved ensuring that all chosen spots were accessible and could be covered within the operational constraints of the UAS such as its flight time and battery life. Each location consisted of multiple VC plants (since, at each location, multiple cotton seeds were planted); however, only one plant's GPS location was considered. This represented mimicking a real-world scenario in which VC plants usually grow in groups and where, by considering the location of one plant from each group, the entire spot consisting of the group of VC plants can be sprayed.

Table 2. GPS coordinates of randomly chosen locations of ten VC plants in experimental plot (left) and the ordered coordinates/nodes after the application of ACO algorithm (right).

VC	Latitudes	Longitudes		Nodes	Latitudes	Longitudes
1.	30.5343	−96.4312		1.	30.53687	−96.4284
2.	30.53431	−96.4301		2.	30.5359	−96.4283
3.	30.53492	−96.4303		3.	30.53321	−96.4298
4.	30.53386	−96.4299		4.	30.53386	−96.4299
5.	30.53531	−96.4289		5.	30.53431	−96.4301
6.	30.53484	−96.4299		6.	30.53492	−96.4303
7.	30.53687	−96.4284	ACO	7.	30.5343	−96.4312
8.	30.53537	−96.4289		8.	30.53484	−96.4299
9.	30.5359	−96.4283		9.	30.53537	−96.4289
10.	30.53321	−96.4298		10.	30.53531	−96.4289
				11.	30.53687	−96.4284

In this case, the first location was also considered to be the home location of the spot-spray UAS at which the flight began and ended. The right side of Table 2 shows the generated nodes in the order determined by the ACO algorithm. Nodes 1 and 11 represented the same location from which the flight began and at which it ended. The optimal path generated for spot-spray application can be seen on webpage-based output (Figure 12) that was generated by the *Streamlit* Python package (Streamlit Inc., San Francisco, CA, USA). The bottom graph shows the number of iterations on the x-axis and the optimal distance covered on the y-axis (in kilometers). The total distance covered by the spot-spray-capable UAS along the generated path was found to be 674.17 m (0.67 km/0.42 m).

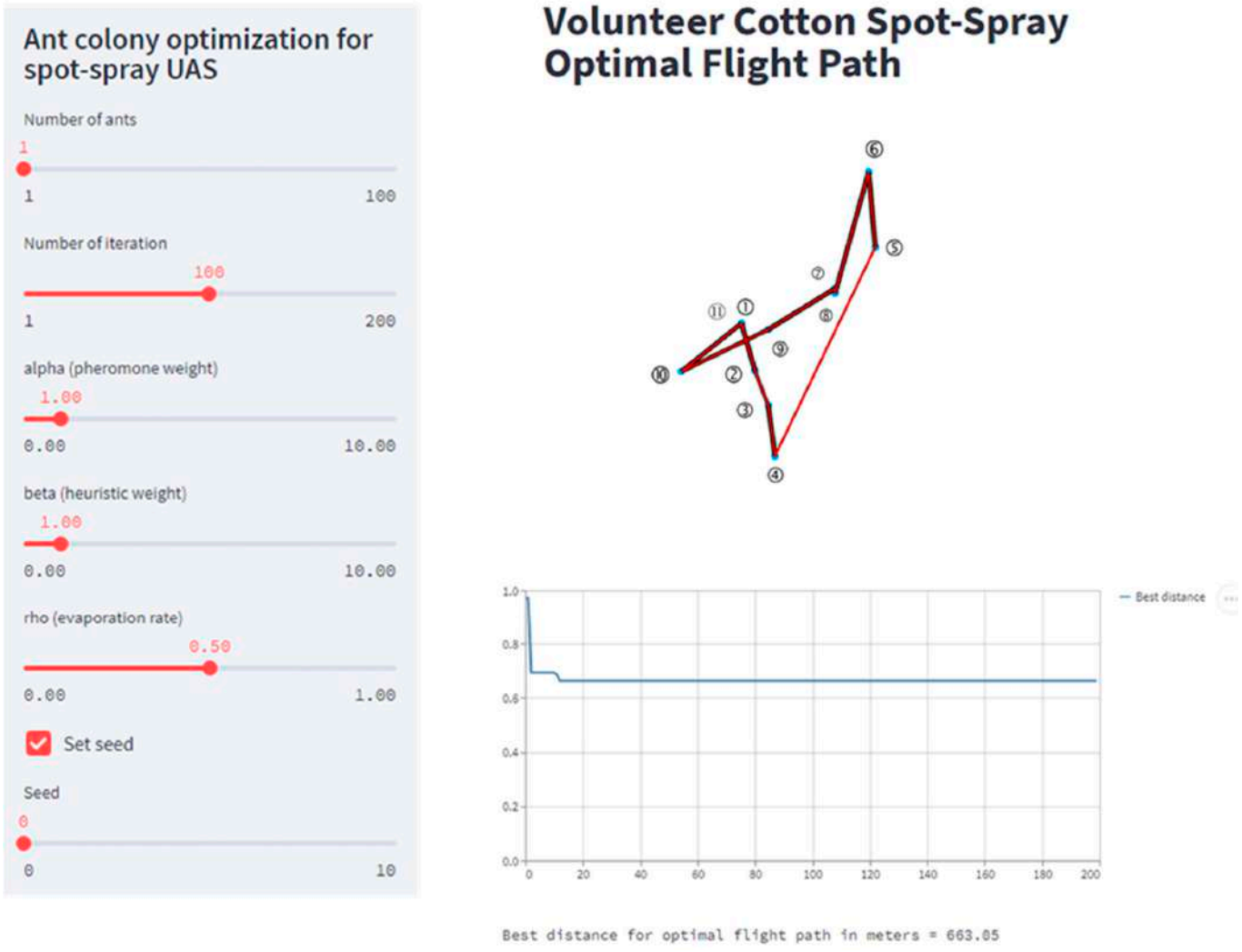

Figure 12. Optimal flight path generated by ACO algorithms and output shown by *Streamlit* Python package on a webpage.

3.3. Spot-Spray UAS Simulation on Mavproxy and Mission Planner Based on the Optimal Flight Path Generated by ACO Algorithm

Once the optimal flight path was obtained (Table 2) by the ACO algorithm, the DroneKit-SITL was used to simulate the spot-spray UAS by following the steps described under the section on *spot-spray UAS simulation* on both GCS-MAVProxy and Mission Planner as shown in Figure 13A–C.

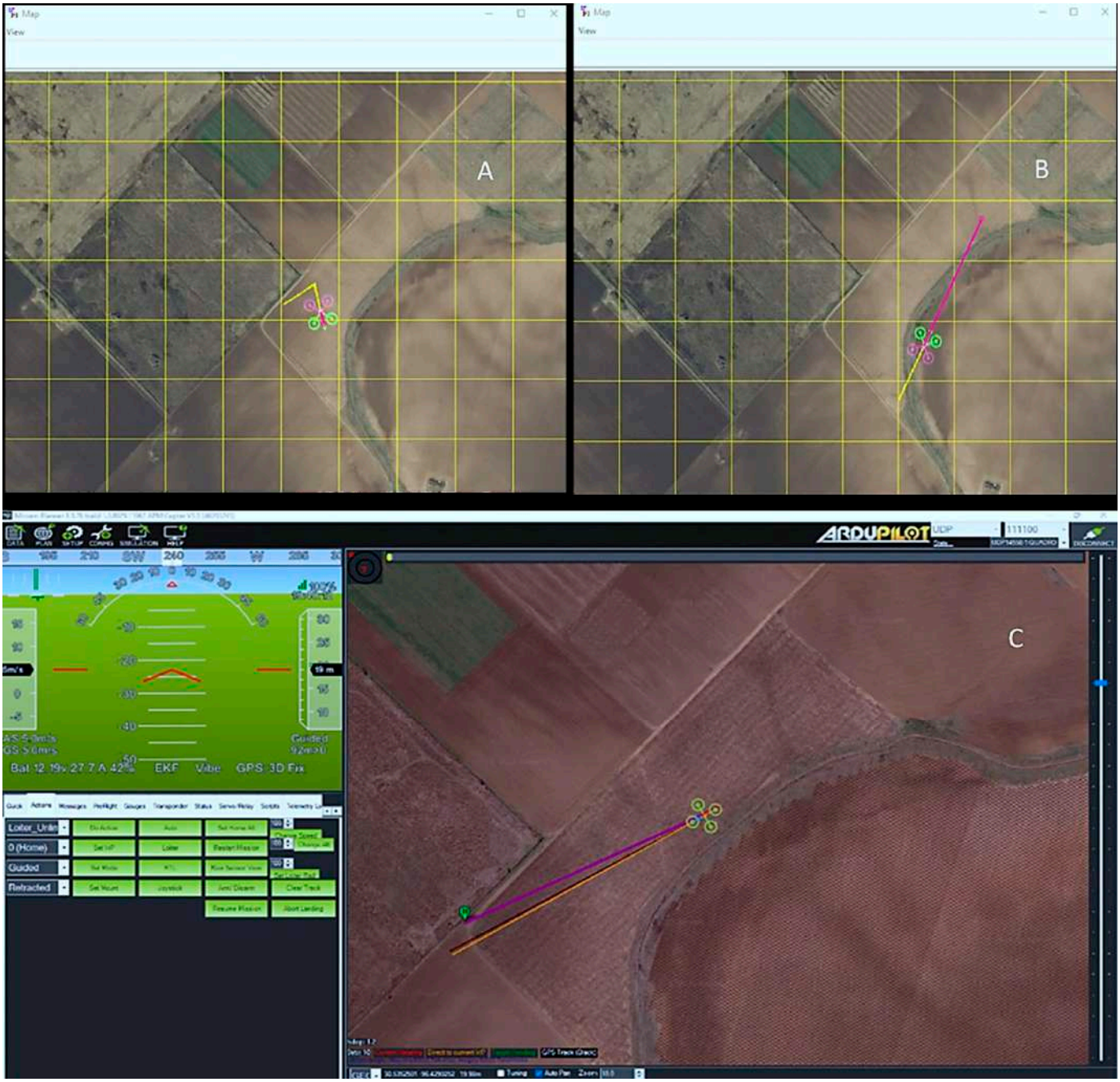

Figure 13. Spot-spray UAS simulation on MAVProxy (**A**,**B**) and Mission Planner (**C**) GCS. Image *A* shows the simulated UAS flying from node 1 to 2 while image *B* shows it flying from node 4 to 5. Image *C* shows the simulated UAS flying from node 8 to 9.

3.4. Spot-Spray Mission on Agrosol GCS

The AgroSol (Hylio Inc., Richmond, TX, USA) GCS software allows one to upload the CSV file containing GPS coordinates of nodes generated by the ACO algorithm, and then, different settings for spot-spray applications can be used for real-life spray applications (Figure 14). Once the CSV file containing the spot locations was uploaded in this study, AgroSol generated a spot group as seen on the left side of Figure 14. It then allowed us to fill in the values for different parameters like the spray altitude, spray volume, etc. The spray mission could be uploaded to the UAS flight controller, and then, the UAS could go to each of the spot locations and precisely spot-spray at each of them.

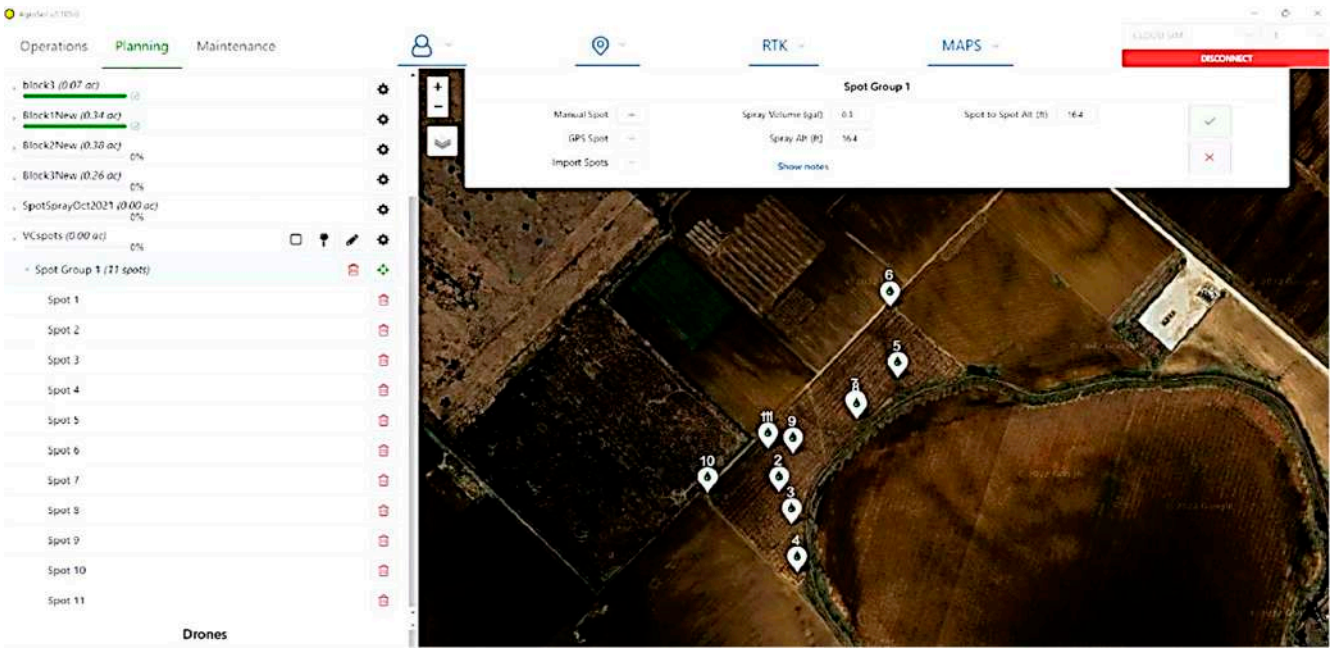

Figure 14. Spot-spray nodes generated by Agrosol software (2.87.5) after uploading the CSV file containing nodes generated by ACO algorithm.

4. Discussion

4.1. VC Detection with YOLOv5m

The potential application of multispectral remote sensing imagery for VC detection in other crops was mentioned by Zhang et al. [48] when they were developing methods of discriminating cotton plants from other crops based on spectral reflectance properties. Similarly, in our previous study [49], we were able to show that UAS-based multispectral remote sensing imagery can be used to detect VC plants growing in corn fields. In another study by Westbrook et al. [50], the authors were able to detect early-growth-stage VC plants with aerial remote sensing RGB imagery. The study reported in this paper took motivation and recommendations from all the previous studies as it made use of multispectral remote sensing imagery collected by a UAS. However, the difference lay in the use of a CV algorithm as opposed to conventional image processing techniques like discriminant and principal component analysis (PCA), maximum likelihood classification, linear spectral unmixing, etc. In the study conducted by Zhang et al. [48], they were able to discriminate cotton plants from other crops like corn, soybean, and sorghum with 100% accuracy; however, the method used a handheld spectroradiometer or the ones mounted on a tractor very close to crop canopies. This means that their method was limited by two major factors: the first one was that it could not be applied to larger areas of field and the second one was that it was a time-consuming process that was not suitable for the near-real-time detection of VC plants. Our previous study [49] used remote sensing multispectral imagery and some classical machine learning techniques to make the process of VC detection semi-automatic; it, however, could not produce a classification accuracy greater than 70%.

In this study, we were able to develop a CV algorithm with YOLOv5m to detect VC plants (before the pinhead square growth stage) growing in a corn field at the tassel (VT) growth stage using multispectral remote sensing imagery. In another study [3], we were able to detect VC plants with more than 90% accuracy in an early-growth corn field (V3 vegetative growth stage) with uncalibrated RGB imagery. YOLOv3 was used, which resulted in a nearly 29% higher classification accuracy than the previous approach [49], but the trained model was less robust to illumination conditions and environmental factors as the images were not radiometrically corrected. Moreover, the relative height difference between VC and corn plants was less compared to that in the study reported in this paper.

In our current and latest approach, reported herein, we were able to develop a CV algorithm based on YOLOv5m, which was more robust to illumination conditions and environmental factors as it was trained on radiometrically corrected imagery [51]. By training the YOLOv5m model on radiometrically corrected imagery, we were able to generate a more reliable model because it was trained on more accurate data [21]. With the current approach, we were able to classify VC plants at 11.43% higher accuracy than the previous method [49] as seen in Figure 10. It is also noteworthy that similar detection accuracy was obtained in a corn field when the plants were at the VT stage unlike the V3 stage reported in the other study [3]. YOLOv5 offers several advantages over its predecessors and other detection models such as Faster R-CNN, EfficientDet, and SSD (Single Shot MultiBox Detector) that have their own strengths and weaknesses. Faster R-CNN is known for its high accuracy, but it often comes at the cost of slower inference speeds, making it less suitable for real-time applications. EfficientDet balances accuracy and speed effectively, but it requires extensive hyperparameter tuning and larger computational resources. SSD offers good speed but generally lags in accuracy compared to YOLOv5.

The overarching goal of this study was to speed up the management aspect of the Texas Boll Weevil Eradication Program (TBWEP) by simultaneously reducing the chemical costs. Therefore, apart from developing a system for the near-real-time detection of VC plants, it is also required to develop a detection algorithm that results in minimal false negatives (i.e., maximum recall). This way, we can minimize the possibility of missing VC plants in the middle of corn fields that can potentially act as hosts for boll weevil pests. In other words, for our used case scenario, our algorithm can be tolerant to false positives but not to false negatives as the damages incurred by boll weevil pests are much more than the extra costs incurred due to spraying some undesired locations because of false positives. For this reason, the model obtained at the 525th iteration can be used (Figure 8). However, if high precision is required (like in the case of saving chemical costs), then the model trained at iteration 322 can be used, resulting in a precision as high as 98% (Figure 8). To quantitatively analyze the trade-offs between precision and recall, we examined the performance of the YOLOv5m model at different iterations. The model at the 525th iteration demonstrated a maximum recall of 0.77, which was critical for minimizing the possibility of missing any VC plants. This high recall value indicates that the model could detect most of the VC plants, thus preventing potential boll weevil infestations. However, this came at the cost of a lower precision, meaning there were more false positives. In our context, this trade-off is acceptable as the primary objective is to ensure no VC plants are missed. The overall performance of the model is represented by mAP@50, whose maximum value was found to be 81% at the 613th iteration (Figure 8). The performance of the trained model can be improved by adjusting the confidence threshold value mentioned by Yan et al. [40]. Apart from these, we had generated 521 images from a limited number of datasets, i.e., 34 images, by using image augmentation techniques. In this way, the trained model reported in this study can be regarded as relatively more generalizable [52]. While image augmentation techniques can enhance the diversity of a training set, they may not fully capture the variability present in real-world scenarios such as that due to different lighting conditions, plant growth stages, and environmental backgrounds. To improve the generalizability of the model, several strategies can be implemented, one of which is increasing the size and diversity of the training dataset by collecting more annotated images from various field conditions and growth stages, which can help the model learn more robust and generalizable features.

4.2. ACO Algorithm for Optimal Flight Path and Spot-Spray

UAS flight parameters such as aerial speed, flight altitude, yaw angle, etc. were assumed to be fixed and the optimal path generated, shown in Figure 12, was based on equal weights assigned to both pheromones and heuristic parameters. These assumptions were like the ones made by Ma et al. [53]. The only constraint used in our implementation was the distance between two GPS locations. However, in other studies, parameters like the

yaw angle were used as some of the constraints [44]. The ACO algorithm being stochastic in nature generates sufficiently good solutions based on randomly generated variables but no globally optimal solution [54]. Therefore, the optimal path shown in Figure 14 may not be the global optimal solution but simply one based on the randomly generated variables used in our study. The parameter values used for the ACO in this study may not be the best ones but were tested with reasonable combinations. From past studies, it was found that ACO is highly sensitive to the evaporation rate, which is considered to be the equivalent of the learning rate; therefore choosing the right value of this parameter is crucial [55,56]. Ojha et al. [55] found that the performance of ACO decreased beyond the 0.5 value for the evaporation rate. Hence, we chose to use 0.5 as the evaporation rate in our case (Figure 12). A total of 11 nodes were generated from the 10 VC plant locations, as seen in Table 2—left and 2—right. Nodes 1 and 11 represent the same location from which the UAS began and ended its journey traversing through the optimal path generated (Figures 12–14).

5. Conclusions and Future Work

5.1. Conclusions

This paper has presented our findings on the successful application of a CV algorithm based on YOLOV5m for the detection of VC plants in a corn field during the tassel growth stage using UAS remote sensing multispectral imagery. Our approach showcased improved classification and detection accuracies compared to previous methods. Additionally, we demonstrated that a low-resolution imaging sensor, coupled with preprocessing algorithms such as radiometric and gamma corrections, can effectively detect VC plants while maintaining robustness against variations in illumination and environmental conditions.

Furthermore, we showcased the practicality of the trained YOLOV5m model for near-real-time detection by deploying it on a computing platform mounted on a sprayer UAS. This implementation enabled the detection of VC plants in full-scale images (1207 $\times$ 923 pixels) at an adjusted average inference speed of approximately 0.4 FPS on a Pascal GPU of the Jetson TX2 development board from NVIDIA.

Moreover, we successfully converted the pixel-wise bounding box central coordinates of the detected VC plants into GPS coordinates. This allowed us to generate optimal flight paths using the ACO algorithm for simulating spot-spray applications. Overall, our research has led to the development of a CV algorithm capable of near-real-time VC plant detection in corn fields along with spot-spray applications utilizing a customized spray-capable UAS. Through this work, we have created a system with the potential to expedite TBWEP mitigation efforts possibly at a reduced management cost.

5.2. Limitations and Future Work

We encountered several limitations in our study. Firstly, we only had 34 original images that contained VC plants. To address this issue, we applied image augmentation techniques to the entire set of original images. Although this approach could potentially lead to data leakage, Saulo Barreto [57] has shown that it can still be used effectively in cases with limited data. Another limitation was observed in our study regarding the slow inference speed of VC detection when deploying the trained YOLOv5m model on the Jetson TX2 platform. This limitation can be mitigated by configuring the hardware appropriately and deploying a lighter version of YOLOv5 for real-time applications. Lastly, the simulation results of spot-spray applications were not validated under field conditions. Therefore, our future work will involve conducting field tests to validate the simulated results.

Author Contributions: Conceptualization, P.K.Y. and J.A.T.; methodology, P.K.Y. and J.A.T.; software, P.K.Y.; validation, P.K.Y. and J.A.T.; formal analysis, P.K.Y. and J.A.T.; investigation, P.K.Y., J.A.T., R.H., S.W.S., U.B.-N., S.C.P., R.R.III and D.E.M.; resources, J.A.T., R.H. and J.E.; data curation, P.K.Y. and J.A.T.; writing—original draft preparation, P.K.Y.; writing—review and editing, P.K.Y., J.A.T., R.H., S.W.S., U.B.-N., S.C.P., R.R.III, D.E.M. and J.E.; visualization, P.K.Y. and J.A.T.; supervision, J.A.T. and R.H.; project administration, J.A.T. and R.H.; funding acquisition, J.A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded and supported by the United States Department of Agriculture's Animal and Plant Health Inspection Service (APHIS) Cooperative Agreement# AP20PPQS&T00C046.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: All data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: Sincere thanks are extended to the farm manager (Stephen P. Labar), the student assistants (Roy Graves, Madison Hodges, Sam Pyka, Reese Rusk, Raul Sebastian, JT Womack, John Marshall, Katelyn Meszaros, Lane Fisher, and Reagan Smith) involved during the field work, and all the unanimous reviewers whose feedback helped improve the quality of this paper. We would also like to thank USDA-APHIS for supporting this project.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Harden, G.H. *Texas Boll Weevil Eradication Foundation Cooperative Agreement*; United States Department of Agriculture: Washington, DC, USA, 2018.
2. Roming, R.; Leonard, A.; Seagraves, A.; Miguel, S.S.; Jones, E.; Ogle, S. Sunset Staff Reports with Final Results. 2021. Available online: www.sunset.texas.gov (accessed on 23 July 2024).
3. Yadav, P.K.; Thomasson, J.A.; Hardin, R.; Searcy, S.W.; Braga-Neto, U.; Popescu, S.C.; Martin, D.E.; Rodriguez, R.; Meza, K.; Enciso, J.; et al. Detecting volunteer cotton plants in a corn field with deep learning on UAV remote-sensing imagery. *Comput. Electron. Agric.* **2023**, *204*, 107551. [CrossRef]
4. Wayne, R. Texas Department of Agriculture Commissioner Sid Miller. Texas Department of Agriculture. Available online: https://www.texasagriculture.gov/News-Events/Article/3021/Commissioner-Miller-Announces-Successes-for-Boll-Weevil-Eradication-in-Texas (accessed on 18 June 2024).
5. Texas Boll Weevil Eradication Foundation, Inc. Weekly Report. 2024. Available online: https://www.txbollweevil.org/Zones/WeeklyMaster.pdf (accessed on 18 June 2024).
6. Wang, T.; Mei, X.; Thomasson, J.A.; Han, X.; Yadav, P.K. GIS-based volunteer cotton habitat prediction and plant-level detection with UAV remote sensing. *Comput. Electron. Agric.* **2022**, *193*, 106629. [CrossRef]
7. Yadav, P.K.; Thomasson, J.A.; Searcy, S.W.; Hardin, R.G.; Braga-Neto, U.; Popescu, S.C.; Martin, D.E.; Rodriguez, R.; Meza, K.; Enciso, J.; et al. Assessing the performance of YOLOv5 algorithm for detecting volunteer cotton plants in corn fields at three different growth stages. *Artif. Intell. Agric.* **2022**, *6*, 292–303. [CrossRef]
8. FMC Corporation. "FYFANON ULV AG," *FYFANON ULV AG*. Philadelphia. 2001. Available online: https://www.sciencedirect.com/science/article/abs/pii/B9780815513810500075?via=ihub (accessed on 18 June 2024).
9. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
10. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
11. Yadav, P.K.; Thomasson, J.A.; Hardin, R.G.; Searcy, S.W.; Braga-Neto, U.M.; Popescu, S.C.; Martin, D.E.; Rodriguez, R.; Meza, K.; Enciso, J.; et al. Volunteer cotton plant detection in corn field with deep learning. In *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping VII*; SPIE: Orlando, Fl, USA, 2022; p. 3. [CrossRef]
12. Jocher, G.; Changyu, L.; Hogan, A.; Yu, L.; Rai, P.; Sullivan, T. YOLOv5, ultralytics/yolov5: Initial Release. Available online: https://github.com/ultralytics/yolov5/tree/v1.0 (accessed on 20 February 2023).
13. Kuznetsova, A.; Maleva, T.; Soloviev, V. YOLOv5 versus YOLOv3 for Apple Detection. In *Cyber-Physical Systems: Modelling and Intelligent Control*, 338th ed.; Kravets, A.G., Bolshakov, A.A., Shcherbakov, M., Eds.; Springer Nature: Warsaw, Poland, 2021.
14. Sharma, V. *Face Mask Detection using YOLOv5 for COVID-19*; California State University-San Marcos: San Marcos, CA, USA, 2020.
15. Zhou, F.; Zhao, H.; Nie, Z. Safety Helmet Detection Based on YOLOv5. In Proceedings of the 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), Shenyang, China, 22–24 January 2021; pp. 6–11. [CrossRef]
16. Yadav, P.K.; Thomasson, J.A.; Hardin, R.; Searcy, S.W.; Braga-Neto, U.; Popescu, S.C.; Rodriguez, R.; Martin, D.E.; Enciso, J.; Meza, K.; et al. Plastic Contaminant Detection in Aerial Imagery of Cotton Fields Using Deep Learning. *Agriculture* **2023**, *13*, 1365. [CrossRef]
17. Li, G.; Suo, R.; Zhao, G.; Gao, C.; Fu, L.; Shi, F.; Dhupia, J.; Li, R.; Cui, Y. Real-time detection of kiwifruit flower and bud simultaneously in orchard using YOLOv4 for robotic pollination. *Comput. Electron. Agric.* **2022**, *193*, 106641. [CrossRef]
18. Hausamann, D.; Zirnig, W.; Schreier, G.; Strobl, P. Monitoring of gas pipelines—A civil UAV application. *Aircr. Eng. Aerosp. Technol.* **2005**, *77*, 352–360. [CrossRef]
19. Minařík, R.; Langhammer, J.; Hanuš, J. Radiometric and atmospheric corrections of multispectral μMCA Camera for UAV spectroscopy. *Remote Sens.* **2019**, *11*, 2428. [CrossRef]
20. Biday, S.G.; Bhosle, U. Relative Radiometric Correction of Multitemporal Satellite Imagery Using Fourier and Wavelet Transform. *J. Indian Soc. Remote Sens.* **2012**, *40*, 201–213. [CrossRef]

21. Mamaghani, B.; Salvaggio, C. Multispectral sensor calibration and characterization for sUAS remote sensing. *Sensors* **2019**, *19*, 4453. [CrossRef]
22. Redmon, J.; Sinigardi, S.; Hager, T.; Maaz, M.; Zhang, V.; Alasuutari, J.; Kahn, P.; Ovodov, I.; Veitch-Michaelis, J.; Dujardin, A.; et al. AlexeyAB/darknet: YOLOv3. Available online: https://zenodo.org/record/5622675 (accessed on 6 June 2022).
23. Jocher, G.; Stoken, A.; Borovec, J.; Christopher, S.T.; Laughing, L.C. Ultralytics/yolov5: v4.0-nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration. *Zenodo* **2021**. [CrossRef]
24. Sorma, R.; Wadud, A.; Karim, S.M.R.; Ahamed, F.A.S. Solving Traveling Salesman Problem by Using Genetic Algorithm. *Electr. Electron. Eng.* **2020**, *10*, 27–31. [CrossRef]
25. Shivgan, R.; Dong, Z. Energy-Efficient Drone Coverage Path Planning using Genetic Algorithm. In Proceedings of the 2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR), Newark, NJ, USA, 11–14 May 2020. [CrossRef]
26. Moon, C.; Kim, J.; Choi, G.; Seo, Y. An efficient genetic algorithm for the traveling salesman problem with precedence constraints. *Eur. J. Oper. Res.* **2002**, *140*, 606–617. [CrossRef]
27. Dorigo, M.; Birattari, M.; Stutzle, T. Ant Colony Optimization. *Stud. Comput. Intell.* **2021**, *947*, 3–8. [CrossRef]
28. Tridgell, A.; Barker, P. ArduPilot MAVProxy. Available online: https://ardupilot.org/mavproxy/index.html (accessed on 2 March 2022).
29. Qays, H.M.; Jumaa, B.A.; Salman, A.D. Design and Implementation of Autonomous Quadcopter using SITL Simulator. *Iraqi J. Comput. Commun. Control. Syst. Eng.* **2020**, 1–16. [CrossRef]
30. Meier, L. MAVLink Developer Guide. Available online: https://mavlink.io/en/ (accessed on 19 May 2021).
31. USDA-Natural Resources Conservation Service, "Web Soil Survey". Available online: https://websoilsurvey.sc.egov.usda.gov/App/HomePage.htm (accessed on 19 July 2021).
32. NVIDIA. Jetson TX2 Module. Available online: https://developer.nvidia.com/embedded/jetson-tx2 (accessed on 19 September 2021).
33. MicaSense Incorporated. MicaSense RedEdge and Altum Image Processing Tutorials. Available online: https://github.com/micasense/imageprocessing (accessed on 14 October 2021).
34. Allebach, J.P. Optimal unsharp mask for image sharpening and noise removal. *J. Electron. Imaging* **2005**, *14*, 023005. [CrossRef]
35. Guo, H.; He, H.; Chen, M. Gamma correction for digital fringe projection profilometry. *Appl. Opt.* **2004**, *43*, 2906–2914. [CrossRef] [PubMed]
36. Ju, M.; Ding, C.; Zhang, D.; Guo, Y.J. Gamma-correction-based visibility restoration for single hazy images. *IEEE Signal Process. Lett.* **2018**, *25*, 1084–1088. [CrossRef]
37. Xu, G.; Su, J.; Pan, H.; Zhang, Z.; Gong, H. An image enhancement method based on gamma correction. In Proceedings of the 2009 Second International Symposium on Computational Intelligence and Design, Changsha, China, 12–14 December 2009; pp. 60–63. [CrossRef]
38. Jocher, G.; Stoken, A.; Borovec, J.; Chaurasia, A.; Liu, C.; Hajek, J.; Diaconu, L.; Defretin, Y.; Lohia, A.; Milanko, B.; et al. ultralytics/yolov5: v5.0-YOLOv5-P6 1280 Models, AWS, Supervise.ly and YouTube Integrations. Available online: https://zenodo.org/record/4679653 (accessed on 21 July 2021).
39. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 740–755. [CrossRef]
40. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A real-time apple targets detection method for picking robot based on improved YOLOv5. *Remote. Sens.* **2021**, *13*, 1619. [CrossRef]
41. Bloice, M.D. Augmentor: Image Augmentation Library in Python for Machine Learning. August 2017. Available online: https://zenodo.org/records/1041946 (accessed on 5 August 2022).
42. "State of Texas UTM Zones," Texas Parks & Wildlife. Available online: https://tpwd.texas.gov/publications/pwdpubs/media/pwd_mp_e0100_1070ah_08.pdf (accessed on 5 November 2021).
43. Sun, Y.; Chen, J.; Du, C. Path planning of UAVs based on improved ant colony system. In Proceedings of the 2020 IEEE International Conference on Progress in Informatics and Computing (PIC), Shanghai, China, 18–20 December 2020; pp. 396–400. [CrossRef]
44. Zhang, C.; Zhen, Z.; Wang, D.; Li, M. UAV path planning method based on ant colony optimization. In Proceedings of the 2010 Chinese Control and Decision Conference, CCDC 2010, Xuzhou, China, 26–28 May 2010; pp. 3790–3792. [CrossRef]
45. Fabien-brulport. Ant Colony Optimisation. Available online: https://github.com/fabien-brulport/ant-colony (accessed on 23 July 2024).
46. Oborne, M. ArduPilot Mission Planner. Available online: https://github.com/ArduPilot/MissionPlanner (accessed on 23 July 2024).
47. Koubaa, A.; Allouch, A.; Alajlan, M.; Javed, Y.; Belghith, A.; Khalgui, M. Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey. *IEEE Access* **2019**, *7*, 87658–87680. [CrossRef]
48. Zhang, H.; Lan, Y.; Suh, C.P.; Westbrook, J.K.; Lacey, R.; Hoffmann, W.C. Differentiation of Cotton From Other Crops at Different Growth Stages Using Spectral Properties and Discriminant Analysis. *Trans. ASAB* **2012**, *55*, 1623–1630. [CrossRef]

49. Yadav, P.; Thomasson, J.A.; Enciso, J.; Samanta, S.; Shrestha, A. Assessment of different image enhancement and classification techniques in detection of volunteer cotton using UAV remote sensing. In Proceedings of the Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV, Baltimore, MD, USA, 15–16 April 2019; Volume 11008, pp. 152–165. [CrossRef]
50. Westbrook, J.K.; Eyster, R.S.; Yang, C.; Suh, C.P.C. Airborne multispectral identification of individual cotton plants using consumer-grade cameras. *Remote. Sens. Appl.* **2016**, *4*, 37–43. [CrossRef]
51. Rumora, L.; Miler, M.; Medak, D. Impact of various atmospheric corrections on sentinel-2 land cover classification accuracy using machine learning classifiers. *ISPRS Int. J. Geo-Inform.* **2020**, *9*, 227. [CrossRef]
52. Gan, T.; Zha, Z.; Hu, C.; Jin, Z. Detection of Polyps During Colonoscopy Procedure Using YOLOv5 Network. In Proceedings of the 3rd International Workshop and Challenge on Computer Vision in Endoscopy (EndoCV2021), Nice, France, 13 April 2021.
53. Ma, G.; Haibin, D.; Liu, S. Improved Ant Colony Algorithm for Global Optimal Trajectory Planning of UAV under Complex Environment. *Int. J. Comput. Sci. Appl.* **2007**, *4*, 57–68.
54. Bianchi, L.; Dorigo, M.; Gambardella, L.M.; Gutjahr, W.J. A survey on metaheuristics for stochastic combinatorial optimization. *Nat. Comput.* **2009**, *8*, 239–287. [CrossRef]
55. Ojha, V.K.; Abraham, A.; Snášel, V. ACO for continuous function optimization: A performance analysis. In Proceedings of the 2014 14th International Conference on Intelligent Systems Design and Applications, Okinawa, Japan, 28–30 November 2014; pp. 145–150. [CrossRef]
56. Ebadinezhad, S. DEACO: Adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem. *Eng. Appl. Artif. Intell.* **2020**, *92*, 103649. [CrossRef]
57. Barreto, S. Data Augmentation. Baeldung. Available online: https://www.baeldung.com/cs/ml-data-augmentation#:~:text=Data%20Augmentation%20on%20Test,%20Validation,also%20use%20it%20during%20testing (accessed on 9 January 2022).