

硕士专业
学位论文

基于深度学习的小样本农作物害虫检测研究与应用

刘春磊

廣西大學

二〇二三 年 六 月

分类号 TP391.4

学校代码 10593

密级 公开

学号

硕士专业学位论文

基于深度学习的小样本农作物害虫检测研究与应用

RESEARCH AND APPLICATION OF FEW-SHOT CROP PEST DETECTION BASED ON DEEP LEARNING

作者姓名: 刘春磊

指导教师: 陈天恩研究员

合作导师:

专业名称: 计算机技术

研究方向: 目标检测

所在学院: 计算机与电子信息学院

论文答辩日期 2023 年 6 月 6 日 学位授予日期 2023 年 6 月 21 日

答辩委员会主席 蒙祖强 教授

基于深度学习的小样本农作物害虫检测研究与应用

摘 要

保障粮食安全是我国建设农业强国的头等大事。然而，农作物的生产经常遭受害虫的侵扰，这严重阻碍了农业的健康发展。因此，快速、准确地完成害虫识别，对害虫的有效治理和农业的安全生产尤为重要。传统的害虫检测方法依靠农业专家到害虫发生现场通过“眼观手查”进行害虫识别，存在时效性差、主观性强等问题。同时，害虫具有区域性和季节性的特点，且种类繁多，难以获得每一类害虫的大量图像，如何通过少量的害虫图像完成对新害虫类别的检测是一个亟待解决的问题。本文根据实际的害虫应用场景，提出了大样本条件下的玉米害虫检测算法和小样本条件下的农作物害虫检测算法，并实现了害虫检测微信小程序。本文的主要研究内容和结论如下：

（1）在大样本条件下，提出了基于 Faster R-CNN 改进的玉米害虫检测算法。首先，使用一系列数据增强组合提高数据输入的多样性，增加网络的鲁棒性。其次，针对极端宽高比的害虫，使用结合遗传算法的 K-means 聚类算法对网络锚框的生成进行优化。然后，针对自然环境下害虫图像背景复杂的问题，将特征金字塔 FPN 替换为能够获得更多害虫特征信息的递归特征金字塔网络 RFP。最后，使用改进的 Soft-NMS 算法用于减少玉米害虫的漏检率。实验结果表明，改进后的玉米害虫检测算法的 mAP 为 79.05%，相比于原始的 Faster R-CNN 算法提高了 3.49 个百分点。

(2) 提出了小样本条件下的农作物害虫检测算法。首先, 将全局特征模块与空间金字塔池化模块 SPP 融合在一起, 加强网络对不同尺度害虫的检测能力并且丰富了特性信息表示。然后, 在小样本害虫检测中引入基于度量学习的负边际损失函数, 避免将新类别错误地映射到相似的基类特征空间中。最后, 在输出端使用 CIoU Loss 进行边界框的回归预测, 提高检测框的定位精度。本文从公开的 IP102 数据集中选择了 68 个害虫种类, 考虑到同一类害虫在成虫和幼虫时外观的巨大差异, 将成虫和幼虫视为不同的类别进行检测, 得到了包含 64 类成虫和 16 类幼虫的新害虫数据集 PEST80。实验结果表明, 在 PEST80 害虫数据集上, 本文的方法在 1shot、3shot、5shot 和 10shot 时对 20 类新类害虫的检测 mAP 分别为 21.8%、32.3%、36.57%和 43.62%, 优于其他前沿的小样本目标检测算法。

(3) 设计并实现了基于深度学习的害虫检测微信小程序。害虫检测小程序可以帮助用户准确、快速地识别害虫种类。实现了害虫知识库功能, 用于害虫的检索和防护措施的查看, 对于害虫的防治具有非常重要的现实意义。

关键词: 深度学习 目标检测 害虫检测 小样本目标检测

RESEARCH AND APPLICATION OF FEW-SHOT CROP PEST DETECTION BASED ON DEEP LEARNING

ABSTRACT

Ensuring food security is a top priority for my country to build a strong agricultural country. However, crop production is often affected by pests, which seriously hinders the healthy development of agriculture. Therefore, rapid and accurate pest identification is important for effective pest management and safe agricultural production. The traditional method of pest detection relies on agricultural experts going to the pest site to identify pests by "eye and hand inspection", which has problems such as lack of timeliness and high subjectivity. Simultaneously, pests have regional and seasonal characteristics and there are many types. It is difficult to obtain a large number of images of each type of pest. How to complete the detection of new pest categories with a small number of pest images is an urgent problem to be solved. Based on the actual pest application scenarios, this paper proposes a corn pest detection algorithm under large sample conditions and a crop pest detection algorithm under small sample conditions, and implements a WeChat applet for pest detection. The main research contents and conclusions of this paper are as follows:

(1) Under the condition of large samples, an improved corn pest detection algorithm based on Faster R-CNN was proposed. First, a series of data augmentation combinations are used to increase the diversity of data input and increase the robustness of the network. Second, for pests with extreme aspect ratios, the generation of network anchor boxes is optimized using the K-means clustering algorithm combined with genetic algorithms. Then, for the problem of complex pest image background in natural environment, the feature pyramid FPN is replaced by the recursive feature pyramid network RFP which can obtain

more pest feature information. Finally, the improved Soft-NMS algorithm is used to reduce the missed detection rate of corn pests. The experimental results show that the mAP of the improved corn pest detection network is 79.05%, which is 3.49 percentage points higher than the original Faster R-CNN network.

(2) A pest detection algorithm under small sample condition is proposed. First, the global feature module is fused with the spatial pyramid pooling module SPP to strengthen the network's ability to detect pests of different scales and enrich the representation of feature information. Then, a negative margin loss based on metric learning is introduced in the few-shot pest detection to avoid mismapping new categories into similar base-class feature spaces. Finally, the CIoU loss is used at the output side to perform regression prediction of the bounding box to improve the positioning accuracy of the detection box. In this paper, 68 pest species were selected from the IP102 dataset. Considering the huge difference in the appearance of adults and larvae of the same pest species, adults and larvae were considered as different categories for detection, and 64 adult species and 16 pest species were obtained. A new pest dataset PEST80 for larvae. The experimental results show that on the selected pest dataset, the detection mAP of 20 new types of pests by the method in this paper is 21.8%, 32.3%, 36.57% and 43.62% at 1shot, 3shot, 5shot and 10shot respectively, which is better than other state-of-the-art few-shot object detection networks.

(3) Design and implement a WeChat applet for pest detection based on deep learning. The pest detection applet can help users accurately and quickly identify pest types, and realizes the knowledge base function of pests, which is used for pest retrieval and viewing of protective measures, which has very important practical significance for pest control.

KEY WORDS: Deep learning; Object detection; Pest detection; Few-shot object detection

目 录

| | |
|---------------------------------------|----|
| 第一章 绪论..... | 1 |
| 1.1 研究背景及意义..... | 1 |
| 1.2 国内外研究现状..... | 2 |
| 1.2.1 基于深度学习的目标检测算法..... | 2 |
| 1.2.2 基于深度学习的农作物害虫检测与识别算法 | 3 |
| 1.2.3 基于深度学习的小样本目标检测算法 | 4 |
| 1.3 本文研究内容..... | 7 |
| 1.4 本文章节安排..... | 8 |
| 第二章 相关技术基础 | 10 |
| 2.1 卷积神经网络..... | 10 |
| 2.1.1 卷积层..... | 10 |
| 2.1.2 池化层..... | 11 |
| 2.1.3 激活函数..... | 11 |
| 2.1.4 全连接层..... | 13 |
| 2.2 深度学习中的目标检测算法 | 13 |
| 2.2.1 两阶段目标检测算法..... | 14 |
| 2.2.2 单阶段目标检测算法..... | 15 |
| 2.3 小样本目标检测算法..... | 17 |
| 2.3.1 基于微调的小样本目标检测算法..... | 17 |
| 2.3.2 基于元学习的小样本目标检测算法..... | 18 |
| 2.4 评价指标..... | 18 |
| 2.5 本章小结..... | 20 |
| 第三章 基于改进 Faster R-CNN 的玉米害虫检测方法 | 21 |
| 3.1 Faster R-CNN 目标检测网络 | 21 |
| 3.1.1 特征提取网络..... | 22 |
| 3.1.2 区域生成模块..... | 23 |
| 3.1.3 RoI 模块..... | 23 |
| 3.1.4 分类和回归..... | 24 |
| 3.2 害虫数据集和数据处理..... | 25 |
| 3.2.1 数据集介绍..... | 25 |
| 3.2.2 数据增强..... | 25 |
| 3.3 改进的递归特征金字塔结构 | 26 |
| 3.4 优化网络锚框设置..... | 29 |
| 3.5 优化预测框的筛选算法..... | 30 |

| | |
|----------------------------|----|
| 3.6 实验结果与分析..... | 31 |
| 3.6.1 实验环境..... | 31 |
| 3.6.2 模型训练..... | 32 |
| 3.6.3 实验结果分析..... | 33 |
| 3.6.4 消融实验..... | 35 |
| 3.6.5 不同模型的对比..... | 36 |
| 3.7 本章小结..... | 37 |
| 第四章 小样本条件下的农作物害虫检测方法 | 38 |
| 4.1 小样本害虫数据集构建..... | 38 |
| 4.2 小样本害虫目标检测网络结构 | 39 |
| 4.3 多尺度特征融合模块..... | 40 |
| 4.4 负边距分类损失函数..... | 41 |
| 4.5 CIoU 边框回归损失函数..... | 43 |
| 4.6 实验结果与分析..... | 45 |
| 4.6.1 实验平台和参数设置..... | 45 |
| 4.6.2 实验结果分析..... | 45 |
| 4.6.3 消融实验..... | 47 |
| 4.6.4 不同模型的对比..... | 48 |
| 4.7 本章小结..... | 49 |
| 第五章 害虫检测系统 | 50 |
| 5.1 系统需求概述..... | 50 |
| 5.2 系统总体设计..... | 50 |
| 5.2.1 系统软硬件平台 | 50 |
| 5.2.2 框架设计..... | 51 |
| 5.2.3 系统功能设计..... | 52 |
| 5.2.4 系统检测流程..... | 53 |
| 5.3 系统展示..... | 53 |
| 5.4 本章小结..... | 56 |
| 第六章 总结与展望 | 57 |
| 6.1 工作总结..... | 57 |
| 6.2 研究展望..... | 58 |
| 参考文献..... | 59 |

第一章 绪论

1.1 研究背景及意义

农业在我国的经济体系中非常重要，是支撑国家经济发展的基础产业之一，也是满足人民物质生活需要的重要保障之一。中国既是一个农业大国，又是一个人口大国，从满足 14 亿人口的粮食需求来看，农作物的安全生产是重中之重。根据国家统计局^[1]的数据，2022 年全国粮食总产量达到 13731 亿斤，而据农业农村部办公厅预测^[2]，在 2023 年我国仅因病虫害造成的粮食损失预计为 3600 亿斤以上，“虫口夺粮”行动刻不容缓。农作物病虫害的种类众多，其可以在短时间内快速传播，危害程度巨大，甚至可以导致农作物死亡。对农作物病虫害的有效防治成为提高农作物质量和产量的关键，受到了越来越多的关注。

因为化学农药的易操作性和成本相对低廉，我国农作物的害虫治理主要还是依赖于化学农药的使用。然而，过量地使用化学农药，会对自然环境造成严重危害，比如污染土壤、水源，如果这些农药通过农作物等途径被人类食用，会对人体的身体健康产生严重危害，甚至造成神经系统、免疫系统的损伤。同时，化学农药的滥用不仅会杀死农作物害虫，也会伤害一些非目标害虫，危害了生态系统的物种多样性，严重影响了地球生态平衡。如果能够准确地检测识别害虫，就能够减少农药的使用，这不仅可以提高作物的质量和产量，还能够保护地球的生态环境。

目前，真实自然环境下的害虫检测主要依靠农业专家或农民的眼观手查，这高度依赖于人类自身的知识储备，检测结果主观性大、不可靠，且人工检测害虫的速度很慢。同时，真实自然环境下害虫的图像背景复杂，存在不同宽高比的害虫，比如幼虫通常为长条状，蛾类偏向正方形。这些因素严重制约着害虫检测的发展。因此，国内外专家对害虫检测展开了许多分析研究，渴望发现既满足害虫检测精度又可以达到较高检测速度的方法。如今，基于深度学习的目标检测方法在社会生产生活的方方面面得到了应用，害虫检测也成功应用了深度学习技术。基于深度学习的害虫检测与传统人工的方法相比在检测精度和速度上都有极大的提高。害虫的自动检测为我国农业智能化和信息化的发展提供了强有力的支撑和保障，使得农业生产更加高效、精准、便捷，促进了农业的可持续发展。

农业害虫的种类很多，根据文章^[3]提到，常见的农业害虫大概有 860 种。不同地区、不同季节的害虫差异很大，害虫的种类丰富，很难收集到大量高质量的害虫图像，而且害虫数据集的注释工作主要由相关农业专家完成，费时又费力。传统的目标检测网络需要大量的害虫图像用于模型训练，如著名的公开数据集 IP102^[4]就有 18k 张用于害

虫检测的图像数据。样本图像获取的高成本是深度学习应用于农作物害虫智能识别检测的一个严重阻碍。另一方面,即使是儿童也能通过很少的图像学习到新类别对象的识别与检测,借鉴于此,考虑传统的目标检测模型是否也能用少量的注释图像完成检测任务。此外,深度学习模型的扩容学习过程非常昂贵,在训练好的模型中增加新的检测类别需要重新训练所有的网络参数,在专业的 GPU 硬件上训练时仍需要很长时间,这使得模型的可用性和动态性大大降低。

为了解决上述存在的问题,本文首先提出了在大样本条件下的玉米害虫检测方法,该方法可以自动检测复杂自然环境下的害虫,节约了人力物力,为田间智能诊断奠定了基础。然后提出了小样本农作物害虫检测方法,当有些害虫的图像难以收集时,该方法可以通过很少的害虫图像完成对新类害虫的检测,大大提高了检测网络的泛化性,具有很好的应用前景。

1.2 国内外研究现状

1.2.1 基于深度学习的目标检测算法

目标检测是在图像识别的基础上进一步提出的任务,图像识别是将一整张图像识别为某个单一类别,但图像中可能不只有一种类别或者一个目标对象,目标检测通过外接矩阵将目标框选出来,并识别出该目标的类别,这样可以实现对图像中的多类别多数量进行检测识别,还可以用于后续的计数和目标跟踪等任务。

基于深度学习的目标检测能够分类为两种类型,一种是两阶段检测算法,另外一种是一阶段检测算法。两阶段检测算法由于有区域候选网络(Region Proposal Network, RPN)结构,在检测精度上有很大的优势,一阶段检测算法通过一个完整的卷积神经网络实现端到端的检测,检测速度相比两阶段算法来说快很多。2014年, Girshick 等^[5]提出了 R-CNN 网络,该网络首先通过选择性搜索算法得到可能包含感兴趣对象的候选框,接着把这些候选框送入卷积神经网络中进行分类和检测,R-CNN 开启了深度学习目标检测的新时代。但是 R-CNN 不能直接处理任意大小的输入图像,需要裁剪成统一大小尺寸之后再输入到支持向量机(Support Vector Machine, SVM)^[6]分类。SPPNet^[7]通过金字塔池化层网络对 R-CNN 进行改进,使得网络可以处理任意大小的输入图像,而不用裁剪图像比例,减少了网络因输入图像畸变而造成的精度损失。2015年, Girshick 等^[8]提出的 Fast R-CNN 融合 SPPNet 网络的优点进一步改进了 R-CNN,使得网络能够同时处理分类和检测任务。随后,最经典的 Faster R-CNN^[9]被提出,创造性地提出了 RPN 网络,不再需要通过外部的选择性搜索算法得到图像的候选区域,使得网络可以实现端到端的检测,并大大提高了网络的检测速度。Lin 等^[10]在 Faster R-CNN 的主干网络上,添加了特征金字塔网络(Feature Pyramid Networks, FPN),新的网络可以更好地检测大、中、小等不同尺度的目标,应对尺度变化很大的不同图像。He 等^[11]在 2017

年提出了 Mask R-CNN 网络, 将 Faster R-CNN 网络中的 RoI Pooling 替换为 RoI Align 结构, 解决了特征图和原始图像之间尺度对齐的问题, 更好的保留了像素级别的信息。Hei 等^[12]提出了 CornerNet 网络, 不再需要 Faster R-CNN 网络中繁琐的锚框机制, 该网络基于热力图估计图像中目标的中心点和宽高来预测检测对象。一阶段网络则没有 RPN 结构, 将检测任务中的分类和定位统一看作回归问题, 通过一个卷积神经网络实现端到端的检测, 可以在图像上直接生成最终的预测结果。主流的一阶段检测算法是 YOLO 系列。YOLO v1^[13]网络由 24 层卷积层和 2 层全连接层组成, 将整张图片看作一个 7x7 的网格, 每个网格只负责预测一个对象, 在 7x7 的网格最多可以预测 49 个对象, 因此, YOLO v1 网络的召回率很低。YOLO v2^[14]在 YOLO v1 的基础上, 去除了 YOLO v1 的全连接层, 使用了类似 Faster R-CNN 的锚框机制, 其可以预测多种比例的对象。YOLO v3^[15]通过多分支网络实现了多尺度对象的检测, 还将分类损失函数由 Softmax 函数改为 logistic 函数, 提高了网络对多类别预测的检测精度。SSD 网络^[16]为解决一阶段算法对多尺度目标难以检测而提出, 使用主干网络的多个特征层进行预测, 利用非极大值抑制 (Non-Maximum Supression, NMS) 对冗余的预测框进行筛选从而完成多尺度预测。He 等^[17]发现相比两阶段网络, 一阶段网络会受到训练中的大量负样本影响, 提出了 Focal Loss 函数, 该函数用于解决网络训练中数据集采用的正负样本不平衡的问题。

1.2.2 基于深度学习的农作物害虫检测与识别算法

传统基于机器学习的农作物害虫检测与识别算法往往需要先通过人工设计提取的特征信息, 然后再送入训练好的分类器中进行检测。这种方法只能检测背景干扰小、害虫特征明显、外部环境固定等情况, 而害虫存在生活环境复杂、容易出现聚集、尺度变化大等特点, 针对这些情况, 传统基于机器学习的害虫检测效果会大打折扣, 检测精度会大幅下降。面对这些困难, 深度学习可以利用大量的害虫数据集图像, 采用卷积神经网络提取到害虫的高质量特征表示, 极大地提高了害虫检测的精度和速度。根据害虫的检测环境不同, 一般将其分为两种, 一种是在吸虫灯、粘性板等诱捕环境中的, 一种是在真实自然环境下的。

针对诱捕环境的害虫检测, Wang 等^[18]重新设计了 RPN 网络的正负样本采样方法, 并在主干的残差网络中加入注意力机制, 在提出的 Agripest21 诱捕害虫数据集上取得了 78.7% 的检测准确率。Shen 等^[19]使用 Faster R-CNN 网络对瓦楞纸板成功诱捕的害虫图像进行检测任务, 通过压缩模型, 在大幅降低模型参数大小的情况下, 对六种储粮害虫的检测性能达到了 87.99% 的准确率。姚青^[20]等在 Yolo v4^[21]网络中添加双线性注意力机制, 对 19 种灯诱害虫的检测准确率达到 94.0%。Li 等^[22]基于 Faster R-CNN 网络提出了 Tpest-RCNN 网络, 针对黄色粘板上粉虱和蓟马等微小害虫使用迁移学习算法进行识别, 检测准确率达到 95.2%。梁勇等^[23]通过诱捕设备捕获稻纵卷叶螟和

二化螟成虫,通过多种数据增强输入到 YOLO v5 网络进行自动化检测和计数,最高的检测精度为 91.67%。Zhang 等^[24]提出 AgriPest-YOLO 模型,提出了协作和局部注意力机制获取害虫更丰富的特征以及减少噪声的干扰。还提出了分组空间金字塔池模块,融合多个尺度特征之间的感受野,在 24 类诱捕害虫数据集上得到了 71.3%的准确率。

针对真实自然环境的害虫检测,李子茂等^[25]对月季病虫害提出了一个新的两阶段检测模型,第一阶段将 Inception^[26]模块加入 YOLO v3 网络中,提高自然场景下模型的特征提取能力,同时得到月季病虫害的掩膜图像,第二阶段使用 K-means 聚类算法优化 Faster R-CNN 中的锚框,并将第一阶段得到的掩膜图像输入到改进的 YOLO v3 网络中,最终达到 82.26%的准确率。Liu 等^[27]在 Faster R-CNN 使用了全局激活特征金字塔技术,在自建的微小害虫数据集上取得了 71%的准确率。陈娟^[28]等对残差网络进行改进,在神经网络的深度残差块中加入高分辨率卷积层,同时提高卷积的通道数量,最终达到了 91.4%的识别准确率。为解决蚜虫聚集难以检测的问题,Li 等^[29]提出了一种由粗到细的两步检测网络,第一步利用卷积神经网络对害虫的聚集区域进行检测,然后对检测出的蚜虫聚集位置进行进一步的精细检测,显著提高了真实自然环境中蚜虫检测的精度。Liu 等^[30]使用混合和局部特征激活特征进行害虫的自动检测,首先利用特征图的全局信息构建全局激活特征金字塔,然后提出了局部激活区域候选网络增强定位功能,在自主构建的 16 种害虫数据集上达到了 75.03%的准确率。

通常诱捕环境需要提前布置相应的诱捕设备,并且只能捕获特定的害虫类别,环境适应能力差,而真实自然场景下的害虫检测则无需诱捕装置,只通过手机等手持设备就可以完成对害虫的自动检测,更加贴近田间智能诊断的使用。

1.2.3 基于深度学习的小样本目标检测算法

传统基于深度学习的目标检测算法往往需要大量的图像样本训练检测模型,高昂的图像获取成本严重制约着目标检测在很多领域的发展,比如农作物害虫种类繁多,因地域、季节等影响,会有各种各样的害虫,尤其是一些不常见的害虫,样本的获取更为困难,因此,小样本目标检测应运而生。小样本目标检测的目的是通过几个带注释的样本检测新的类别,其大致可分为两种范式,一类是基于元学习的,这种方法是通过学习一个元学习器泛化到新类的检测上。另一类是基于微调的,这种方法是先通过大量的基类数据训练一个基模型,然后冻结基模型的一部分参数,在小样本数据集上微调网络的某些模块参数达到较高的新类检测性能。两种方法都是为了小样本目标检测任务而提出的,评价指标也都相同,都希望在新类样本数量少的情况下提高模型的检测性能。但也有很多不同之处,当检测网络类别数目增加时,元学习的支持集需要每一个类别的样本图像参与训练。所以,网络训练所需显存需求会随着类别数目的增多而变大,而基于微调的方法则不会发生这种情况。另外,基于元学习的方法在数据输入分为支持集和查询集,每次都要通过计算无标注的查询集和有标注的支持集间

的距离从而对查询集图像进行检测训练,元学习利用这种方式来模拟模型在小样本情况下的状态。

上文介绍了小样本目标检测的两类范式,在小样本目标检测算法中有一些常用的改进策略,这些常用的改进策略可以归类为基于度量学习的方法、基于图卷积的方法、基于数据增强的方法和基于注意力机制的方法。以下是对这些改进策略的总结分析。

(1) 基于度量学习的方法

Cao 等^[31]提出的 FADI 网络将基于两阶段微调方法(Two-stage Fine-tuning Approach, TFA)^[32]在微调阶段再细分为关联和鉴别两步,在关联步骤中,将新类投影到相似类别的基类空间中,通过伪标签将相似的基类看作新类。在鉴别步骤中,分离出新类的检测头,将基类和新类使用不同的检测头分开检测,最后再通过集合边缘最大值损失将包含基类和新类的所有类别分离开。Qiao 等^[33]基于 Faster R-CNN 网络提出了解耦的 DeFRCN 方法,设计出一个使用度量学习的离线原型计算方法,该方法计算候选区域同离线原型之间的相似度,使用得到的相似度微调模型更好地进行新类检测。Wu 等^[34]提出了 FSOD-UP 方法,该方法对所有的类别数据训练得到通用原型,而不再是单一类别的原型,通用原型包含着不同类别固有的恒定关系,通过度量类别间的共通之处可以挖掘出新类别的相关特征。Li 等^[35]提出了 CME 方法,发现在新类的类别识别与特征表示有着不可调和的矛盾。不同的基类之间相互疏远有利于新类的识别,基类的特征空间紧密邻近有利于更好地表示新类。CME 方法利用类边界损失给新类划分了足够的边界距离,同时尽量保证类别间的边界平衡。Li 等^[36]提出了 MM-FSOD,重构了元学习的分类训练方法,利用度量学习对不同类别进行聚类,得到这些类别的原型特征,之后通过原型特征对新类进行比对区分。Karlinsky 等^[37]提出了 RepMet 方法,创造性地利用一个距离度量模块替换 Faster R-CNN 网络的预测分类头。该度量模块创建了一个包含每个类若干个模型的特征嵌入空间。对候选区域利用度量模块得到属于不同类别、不同模型的概率,接着利用一个嵌入损失函数,缩小候选区域同概率值最大的嵌入向量间的距离,扩大同概率值最小的嵌入向量的距离。

通过度量大量样本的基类和小样本的新类之间的相似性,根据相似度结果的不同,不同类别之间距离增大,相同类别的数据更加聚拢,能够从众多的基类中区分出新类。

(2) 基于图卷积的方法

Kim 等^[38]在两阶段微调的基础上提出了 FSOD-SR 方法,任何一张图像中的不同类别之间都存在着某种关联,而不是独立存在的。例如在键盘和显示器附近的某个物体是鼠标的可能性远大于是球的可能。FSOD-SR 选择候选区域当作图节点,利用图卷积技术推导候选区域之间的关联,这样可以将全局信息融合起来进行分类预测。Zhu 等^[39]提出 SRR-FSD 方法,每个类别的名称对于分类也同样重要,利用图卷积对每个类的类别文本信息完成关系推理,将推理得到的文本信息同类别的视觉特征融合在一起,

进行后续的处理过程。Kim 等^[40]在元学习的基础上提出了 FSOD-KT 方法,将支持集得到的类别原型作为图的节点,支持集不同类别间的文本相似度作为图卷积的边。利用图卷积神经网络推理得到不同类别原型之间深层次的内在关联,完成对类别原型的特征增强,使用增强后的类别原型对查询集图像进行类别判断。Liu 等^[41]提出了 DRL-for-FSOD 方法,通过支持集数据得到的类别原型和查询集的图像共同组合为一个特征空间,将其作为图卷积网络的节点,通过皮尔逊相关系数计算类别原型和查询集图像之间的相似性当图的边,利用图卷积的推理,令不同类别间彼此远离,相同类别之间相互接近,降低了网络错误分类的情况。Han 等^[42]提出了 QA-FewDet 方法,在提出的异构图卷积网络中存在两种子图,一种是类间子图,用来推理基类和新类的类间关系,一种是类内子图,用来推理新类节点与所有候选区域的联系以及新类的候选区域之间的联系。

针对新类别训练样本数量少的问题,可以利用图卷积神经网络充分挖掘不同类别之间的内在关系。图卷积神经网络不同于普通卷积网络,图节点周围的结构不是完全相同的,不存在平移不变性,所以不同图节点之间存在各种各样的关联关系。

(3) 基于数据增强的方法

针对数据集中存在的尺度变化问题,Wu 等^[43]提出了多尺度正样本优化 MPSR 方法,通过图像金字塔生成多种尺度正样本,将这些正样本输入新构造的辅助分支处理为有偏移的多种尺度对象方形框,接着将不同尺度的对象方形框手动地与 FPN 的不同层次捆绑对应,因此模型可以更好地检测不同尺度的对象。Zhang 等^[44]等针对数据集缺乏足够的变化,通过幻觉网络(Hallucination)在原始样本上利用类原型和噪声共同作用生成额外变化的幻觉样本,然后将幻觉样本和原始样本混合起来,对分类器进一步优化,同时将在基类内存在的样本变化应用到新类中。Kaul 等^[45]提出了 Pseudo-Labeling 方法,首先生成新类样本的伪标签,接着使用自监督网络去除大量类别错误的新类伪标签。然后对剩下的伪标签逐步调整其边界框的质量,将处理后的新类伪标签加入到检测网络中,大大增加了新类的训练样本数量。Xiao 等^[46]提出了 FsDetView 方法,将支持集的类原型特征与查询图像的候选区域通过多种方式融合,利用特征自身、简单特征相减和逐通道相乘的特征融合方式,可以挖掘出查询集候选区域与类原型间的多种关联。Fan 等^[47]提出了 Attention-RPN 方法,在支持集的类别原型和查询集候选区域利用块状、局部和全局的联系进行特征融合,有效减少了背景信息对类别检测的影响。Li 等^[48]通过实验证明只通过简单的数据增强不能有效提高小样本新类的检测性能,提出了变换不变性(Transformation Invariant Principle, TIP)的数据扩充方法,对支持集和查询集的数据都通过 TIP 方法生成新的变化样本,在支持集的变化样本和原始样本上使用一致性损失,用查询集变化样本对原始样本作修正操作,然后将支持集和查询集的结果做归纳操作。Zhang 等^[49]提出了 CoRPNs 方法,针对新类样本量少

的问题,将单个的 RPN 网络扩展为多个 RPN 网络的组合,在正向进行网络训练时,不同的 RPN 结构会尽可能避免对新类候选框的遗漏。在反向梯度传播中,只会从候选框存在概率最大的 RPN 得到相应的梯度。

针对新类别样本数据量不足的问题,扩充数据样本可能是最直接有效的方式。数据增强可以通过直接增加训练的图像数量或者间接对特征进行增强的手段,增加检测网络的输入特征信息,可以得到更好的检测准确率,同时减少模型过拟合的可能。

(4) 基于注意力机制的方法

Wu 等^[50]提出了 FSOD-UP 方法,对候选区域和带条件的通用原型两部分使用注意力关注通道上的关系,候选区域的产生因此更加精准,从而改进了新类的检测结果。Yan 等^[51]提出了 Meta R-CNN 方法,由于每个候选区域都可能是单个对象或者背景,Meta R-CNN 解决了之前元学习方法无法处理单张图像存在多个对象的问题。其新增了预测头重塑网络分支,与 Faster R-CNN 共享主干网络,该分支用标注的支持图像获取每个类别的注意向量,将该向量应用到 Faster R-CNN 的候选区域特征,使用通道注意力机制检测查询图像中与这些向量所表示类别相同的对象。Fan 等^[47]提出了 Attention-RPN 方法,新增加深度互相关注意力区域候选网络模块,使用该注意力提高候选区域的生成质量。利用新类和基类之间的相似性来检测新类,通过在很少支持集图像的情况下,能够在查询图像中找到相应的前景部分,同时抑制背景的误检。Agarwal 等^[52]认为在小样本情况下使用元学习方法容易引起灾难性的遗忘和类别混淆,因此提出了 AGCM 方法。对不同的候选区域采用空间注意力机制寻找相似性,因此能够在分类器中得到更加紧密的类别特征簇。Zhang 等^[53]认为先前的方法在小样本情况下,很难获得位置良好的候选框,借鉴 DETR^[54]提出了 Meta-DETR 方法,去除了候选框预测,在图像上直接进行对象的分类和定位。利用 transformer 的多头注意力机制对查询集分支和支持集分支进行注意力的关联,最后通过 transformer 解码器的类别无关性对查询集进行检测。

关于小样本目标检测任务,想要从样本数很少的新类准确定位并识别出感兴趣的对象十分困难,而注意力机制能精准发现图像的感兴趣区域。注意力机制可以看作一个动态选择的过程,通过定位输入图像的重要区域从而对特征进行自适应加权。

1.3 本文研究内容

本文以自然环境下的农作物害虫数据集 IP102 作为实验数据集,以提高害虫检测的准确率为研究目标,在大样本条件下,提出了改进 Faster R-CNN 的玉米害虫检测算法,当新出现的害虫类别不足以满足传统目标检测训练要求时,提出了小样本条件下的农作物害虫检测算法。同时,设计并完成了一款害虫检测微信小程序。本文的主要研究内容包括以下几点:

(1) 基于改进 Faster R-CNN 的玉米害虫检测方法。采用递归特征金字塔网络加强网络的特征表示能力, 可以提高自然环境下害虫图像背景复杂的检测性能。针对玉米害虫数据集中存在极端宽高比的害虫, 采用结合遗传算法的 K-means 聚类算法重新计算出害虫数据集不同的锚框比例, 提高害虫的检测精度。使用 Soft-NMS 用于减少害虫被漏检的情况。

(2) 小样本条件下的农作物害虫检测方法。除了玉米, 还有许多其他的农作物, 考虑到这些作物上很多种类的害虫难以获得大量图像, 不能够满足传统的深度学习目标检测算法的训练需求。本研究在基于微调的小样本目标检测算法 TFA 上进行改进, 针对害虫图像具有类间相似、类内差异大的特点, 将害虫的成虫和幼虫阶段看作不同的类别进行检测, 从而实现细粒度的害虫检测。针对害虫图像少的问题, 通过多尺度特征融合获得更多的图像害虫特征信息。对于小样本条件下, 容易将新类别错误地识别为基类, 将带负边距的余弦损失函数加入到分类器中, 可以避免害虫的误检测。对于害虫检测框定位不精准的问题, 采用 CIoU Loss 函数用于进一步提高定位的精度。

(3) 开发了害虫识别与检测系统。现在智能手机在我国已经非常普及, 微信作为其中重要的通讯工具也有着广大的用户基础。本文在微信小程序上实现了害虫的检测和查询系统, 用以满足用户对害虫的自动检测需求, 同时用户还可以在系统上查询某类害虫的特性以及防治手段。

1.4 本文章节安排

论文一共由六个章节组成, 具体内容如下:

第一章: 绪论。本章首先阐述课题的相关背景及研究意义, 然后从目标检测算法、农作物害虫检测与识别算法、小样本目标检测算法三个方面总结了国内外研究现状。最后概述了本文的研究内容和具体的章节安排。

第二章: 相关技术基础。本章首先阐述了卷积神经网络的相关基础知识, 其次阐述了两阶段和一阶段两种经典的目标检测算法, 然后对小样本目标检测的两种范式进行了阐述。最后对本文的评价指标进行了介绍。

第三章: 基于改进 Faster R-CNN 的玉米害虫检测方法。本章首先介绍了使用的基线 Faster R-CNN 网络。接着介绍了使用到的玉米害虫数据集。接着从四个方面对基线方法进行了改进。在数据处理上, 针对数据量较少的问题, 使用了一系列的数据增强组合进行数据多样性处理。在模型结构上, 将特征金字塔 FPN 替换为递归特征金字塔结构 RFP, 可以获得害虫图像更丰富的特征信息。在锚框方面, 使用了结合遗传算法的 K-means 聚类算法, 改进了 Faster R-CNN 原始的固定锚框尺寸方法。在后处理方面, 原有的 NMS 算法会造成一定的漏检, 使用改进的 Soft-NMS 提高了检测精度和召回率。最后进行了详细的实验并就实验结果做出了一系列的分析。

第四章：小样本条件下的农作物害虫检测方法。本章首先构建了小样本害虫数据集，将害虫数据集分为基类和新类，同时由于同一类害虫的幼虫和成虫外观差异过大，本研究将成虫和幼虫视为不同的类别进行检测。然后提出了一种多尺度特征融合模块，采用多个 RoI Align 层和全局信息加强特征表示。针对新类容易误分类的情况，将负边距余弦损失加入到分类损失中，减少害虫的误检测。另外采用 CIoU Loss 提高害虫检测框的定位精度。

第五章：农作物害虫检测系统的设计与实现。首先对农作物害虫检测系统进行了相关的设计，然后对其中的功能进行了展示和说明。

第六章：总结和展望。总结了本文的主要研究内容，并对未来进一步的研究方向进行了展望。

第二章 相关技术基础

2.1 卷积神经网络

卷积神经网络作为深度学习的一个重要组成，是一个前馈神经网络，特别适合于处理图像这种三维数据，其一般由卷积层、池化层、激活函数以及全连接层组成。这些不同的模块都发挥着各自的作用，卷积层主要用于提取局部区域的特征，池化层通常用于特征的压缩，激活函数为不同特征之间添加非线性关系，全连接层一般用在网络的最后阶段，用于定位或者分类等。经过这些模块的处理，可以从卷积神经网络中得到想要的结果。以下对上述提到的这几部分进行具体的介绍。

2.1.1 卷积层

卷积层作为卷积神经网络的基本组成部分，通常用于对输入数据进行特征提取。当网络的输入是图像时，卷积层以 3 维数据为输入数据，通过卷积运算以 3 维数据为输出数据。卷积运算类似图像处理中的“滤波器操作”，主要参数有卷积核的大小、卷积核的个数、滑动步长的大小和填充边缘。卷积核通常的尺寸大小为 1x1、3x3、5x5、7x7 等，1x1 的卷积核通常被用来调整通道大小，达到通道对齐或者压缩模型的作用，其他尺寸的卷积核通常用于特征提取。利用这些卷积核在特征图从上到下、从左到右依次滑动固定步长，卷积核在滑动时和图像中相对应位置进行乘法求和操作，得到该位置上卷积的数值。如果要在进行卷积操作的过程中，保持固定的图像大小，则可以通过填充边缘来实现。如图 2-1 所示为两个卷积核运算过程的详细示意图。

卷积神经网络包括网络模型的构建和参数的学习。卷积层的位置、大小就是网络模型的组成部分，卷积核的参数就是卷积神经网络学习的一部分。卷积核具有权值共享的特性，即针对一张图像或特征图时，一个卷积核的参数是相同的，这种特性可以大大减少模型需要学习的参数量，避免传统图像处理算法的复杂过程。

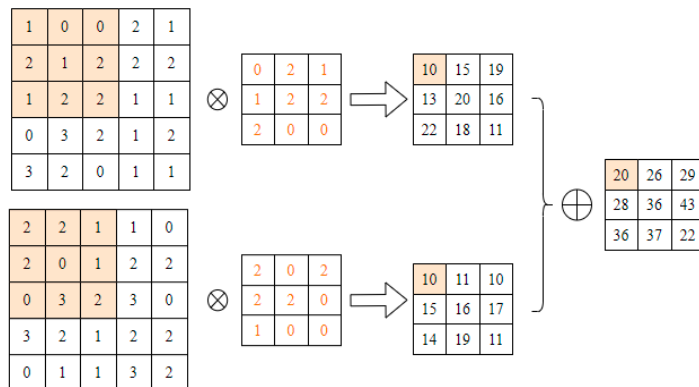


图 2-1 卷积过程

Fig. 2-1 The convolution process

2.1.2 池化层

卷积层运算过程中会产生大量的参数，这些参数通常并不都是网络所需的参数，容易造成参数冗余的现象。池化层可以对这些参数进行稀疏处理，减少网络中不必要的参数。另外，通过大量的训练样本，卷积神经网络最终会得到具有高度语义的处理结果，因此需要压缩特征图维度的操作，除了步长大于 1 的卷积核外，也能够用池化层进行维度的减少。相比卷积层来说，池化层一般没有需要学习的参数，可以大大减少网络的卷积运算，降低模型参数，加快整个模型的检测速度和减小模型的大小。常用的池化层一般有最大池化和平均池化两种，最大池化是选择特征矩阵中的最大值，平均池化是计算特征矩阵中的平均值。池化层的运算需要两个参数，一个是池化窗口的大小，一个是池化窗口的步长。如图 2-2 所示为池化层的示意图，上边为平均池化操作，下边为最大池化操作。图中池化窗口大小为 2×2 ，窗口步长为 2，即在 2×2 的特征矩阵中计算平均值或者最大值，然后移动 2 个像素单元大小，最大池化可以避免卷积操作中出现的均值误差，保留图像的纹理信息，平均池化则可以减少图像中的背景信息的干扰。

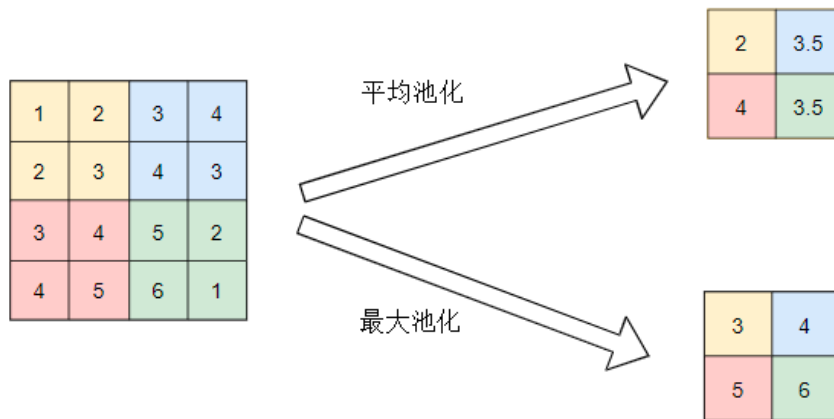


图 2-2 池化层过程

Fig 2-2 Pooling layer process

2.1.3 激活函数

卷积运算过程是一种线性操作，如果在卷积神经网络中只是对卷积层进行简单的堆砌，那么在网络的输入和输出之间只能得到线性关系，而在现实中很少有这种简单而直接的线性关系。激活函数可以在输入输出之间添加非线性关系，提高整个卷积神经网络的表达能力。在卷积神经网络使用中常见的激活函数有 Sigmoid 函数、Tanh 函数、ReLU 函数和 LeakyReLU 函数等，其函数和导函数图像如图 2-3 所示，以下将对这些激活函数逐一进行介绍。

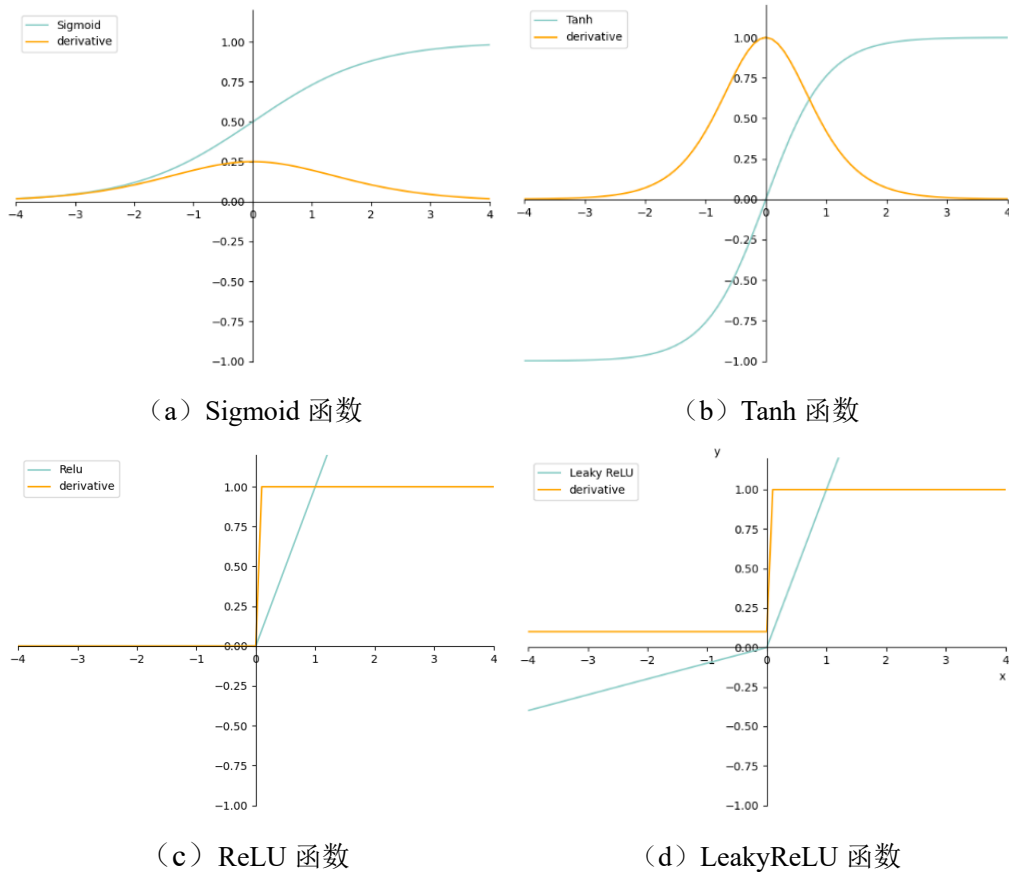


图 2-3 常用函数及其导函数

Fig. 2-3 Common functions and their derivatives

Sigmoid 函数的定义如公式 2-1 所示，从图 2-3 (a) 中可以观察到，当函数的输入在 0 附近时，函数的导数会限制在一个较小的线性范围内，有利于神经网络进行梯度传播。但当输入为正负无穷时，其导函数值在 0 附近，略微大于 0，导致梯度的反向更新变得很困难，发生梯度消失现象，使得网络模型难以收敛。

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (2-1)$$

Tanh 函数的定义如公式 2-2 所示，从图 2-3 (b) 中可以看出，Tanh 的函数曲线同 Sigmoid 函数很相似，都是“S”形，但是它的取值范围为[-1,1]，输出均值为零，相对 Sigmoid 函数来说，使用 Tanh 激活函数，卷积网络的收敛速度会更快一点，但是仍然存在梯度消失的情况。

$$\text{Tanh}(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2-2)$$

ReLU 函数为解决 Sigmoid 函数的梯度消失而提出，也是现在使用频率较高的激活函数，其定义如公式 2-3 所示，从图 2-3 (c) 中可见，在坐标的正半轴，导函数始终大于 0，不再存在梯度消失的现象，但是在负半轴，ReLU 函数的梯度则始终为 0，这会导致卷积网络的权重无法更新，造成神经网络坏死的现象，通过学习率的设置和

合理的参数初始化可以降低神经元坏死的概率。

$$\text{ReLU}(x) = \max(0, x) \quad (2-3)$$

LeakyReLU 激活函数是为了解决 ReLU 函数在曲线负半轴神经坏死而提出的，其定义如公式 2-4 所示，从图 2-3（d）中可见，LeakyReLU 激活函数在正半轴同 ReLU 函数的曲线完全一样，但在负半轴时，LeakyReLU 函数的值为负值，存在一点微小的梯度，可以解决神经网络训练中梯度消失的情况，有利于神经网络的梯度反向更新和网络的训练。

$$f(x) = \begin{cases} \alpha * x, & x \leq 0 \\ x, & x > 0 \end{cases} \quad (2-4)$$

2.1.4 全连接层

全连接层通常位于网络的最后阶段，如图 2-4 所示，全连接层的主要作用是得到最终的分类结果。卷积网络经过全连接层之前的多种卷积操作已经获取到具有高度语义信息的特征。全连接层可以将其中的特征信息投影到样本的标记空间中，寻找最符合某个类别的特征，得到该类别的分类结果。一般其函数表达式如式 2-5 所示，其中 X 和 Y 分别代表输入和输出特征， W 代表需要学习的全连接层权重， b 代表偏置向量。

$$Y = F(WX + b) \quad (2-5)$$

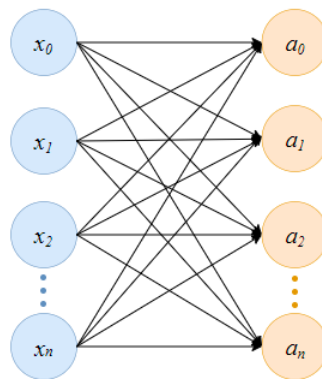


图 2-4 全连接层

Fig. 2-4 Fully connected layer

2.2 深度学习中的目标检测算法

目标检测是将给定图像中感兴趣的对象用外接矩形框选出来，并识别出该对象的类别。传统的目标检测方法需要手动地提取图像中的特征，不但费时又费力，而且不能保证足够的准确性，极大地限制了目标检测算法的发展。近些年来，基于深度学习的目标检测算法取得了极大的进步，被广泛地应用到各行各业中。基于深度学习的目标检测可以分为两阶段目标检测算法和一阶段目标检测算法两种主流方向。两阶段方

法代表的算法有 Faster R-CNN 网络,一阶段方法代表的算法有 YOLO 网络系列和 SSD 网络。

2.2.1 两阶段目标检测算法

在 2014 年, Girshick 等人提出了划时代的两阶段目标检测算法 R-CNN, 其算法流程如图 2-5 所示, R-CNN 算法分为三个处理阶段, 分别是候选区域生成模块、区域特征提取模块以及区域特征分类。R-CNN 网络整体流程如下: 首先利用选择性搜索算法从输入图像中得到大约 2000 个候选框, 这些候选框中包含着想要检测的对象, 接着将这些候选框裁剪为统一尺寸的特征图, 利用深度卷积神经网络对这些候选框进行特征提取, 得到拥有高级语义信息的特征向量, 然后将这些特征向量送入支持向量机 SVM 中进行分类。最后, 将提取到的特征送入全连接层处理, 得到候选区域的预测位置。R-CNN 网络对比于传统的机器学习方法, 已经不再需要手动地提取图像特征, 通过卷积神经网络可以获得高质量的特征, 但是其仍然存在一些需要解决的问题。首先, 选择性搜索算法难以生成高质量的候选区域且耗时巨大。其次, 每一个候选区域都要单独送入卷积神经网络中进行检测, 存在大量冗余的计算, 检测网络的效率低下。最后, 分类使用独立的支持向量机完成, 与整个网络没有很强的关联性, 不能够实现端到端的图像检测。



图 2-5 R-CNN 网络图

Fig. 2-5 R-CNN network diagram

针对 R-CNN 存在的问题, He 等提出了 SPPNet 网络, SPPNet 首先不再裁剪候选区域到统一尺寸, 而是通过在卷积层后添加一个空间金字塔池化 (Spatial Pyramid Pooling, SPP) 层生成固定长度的特征向量, 随后再送到后续的全连接层处理, 避免了随机裁剪造成的图像变形问题。SPP 层的具体流程如下: 首先将任意输入大小的特征图分成 4×4 、 2×2 、 1×1 的小格子, 然后通过每个格子进行最大池化处理, 得到了 16、4、1 三个长度的特征向量, 将这三个特征向量拼接起来, 就可以得到固定的 $(16+4+1) \times 256$ 的大小输入到全连接层, 其中 256 为通道大小。SPPNet 相对于 R-CNN 来说, 只使用一次卷积网络就可以完成对整个图像的候选区域提取, 不再需要冗余的重复卷积, 还可以处理不同比例的图像, 不需要再将图像裁剪为特定大小, 最大程度地保留了原始图像的空间信息。但是 SPPNet 仍有需要解决的问题, 其需要额外的存储设备去保存特征信息, 另外, SPP 层的梯度没有反向传播到前边的卷积层, 卷积层的参数都处于冻结的状态, 因此网络不能充分地学习到足够的特征表示。

Fast R-CNN 进一步优化了 SPPNet 网络，提出了 RoI Pooling 模块，将由 RPN 模块得到的候选区域经过 RoI Pooling 池化操作生成 7x7 大小的特征图。同时 Fast R-CNN 也提出使用多任务损失进行模型训练，将总损失反向传播到卷积层，使用 Softmax 分类器和边界框回归得到分类和定位结果。虽然 Fast R-CNN 网络进一步改进在了 SPPNet，但是其候选区域仍然需要采用选择性搜索算法生成，该方法相比网络的其他部分相当耗时，而且得到的候选区域也不够准确，无法满足实际的使用需求。

因此，Girshick 等人进一步提出 Faster R-CNN 网络，主要改进是提出 RPN 模块代替了选择性搜索算法。Faster R-CNN 由主干卷积网络、RPN 网络、RoI Pooling 和检测头四部分组成。RPN 以主干网络生成的特征图为输入，在特征图上的每个像素位置生成相应的多个锚框，然后利用 NMS 等算法对这些锚框进行筛选，最终可以得到大约 2000 个区域候选框，接着将这些区域候选框送入 RoI 模块中，使用 RoI Pooling 方法统一为 7x7 大小的特征图，再经过多层全连接层得到预测的分类和回归值。Faster R-CNN 网络如图 2-6 所示，Faster R-CNN 真正意义上实现了端到端的检测，同时具有很高的网络检测精度和速度。

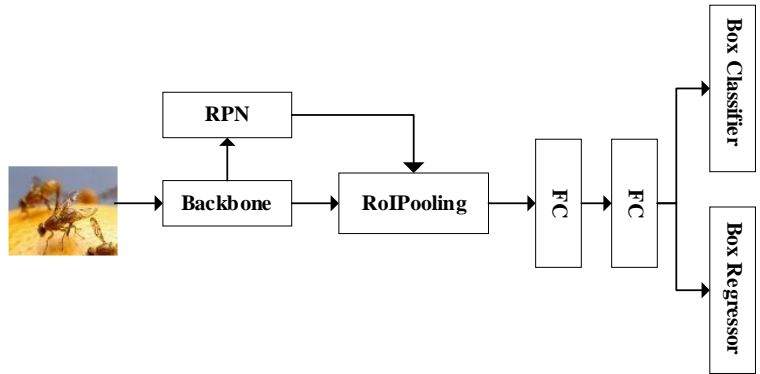


图 2-6 Faster R-CNN 网络图

Fig. 2-6 Faster R-CNN network diagram

2.2.2 单阶段目标检测算法

不同于两阶段目标检测算法，单阶段目标检测算法没有 RPN 模块，即没有单独用于候选区域生成的部分，而是将图像上的每个位置看作潜在对象，将分类和定位两个任务通过一个卷积网络完成整个操作。因此单阶段目标检测算法的检测速度相对占优，但检测精度和两阶段目标检测相比还有一定的距离。

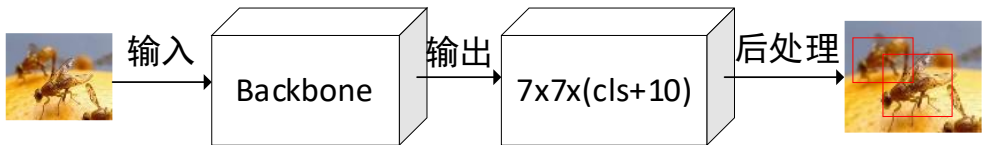


图 2-7 YOLO v1 网络图

Fig. 2-7 YOLO v1 network diagram

Redmon 等人提出了 YOLO v1 网络用于实时检测, 其结构如图 2-7 所示, YOLO v1 以图像为输入, 经过深度卷积神经网络提取特征, 得到一个 7×7 大小的特征图网格, 在这个特征图的每个网格上预测两个锚框, 通过设定阈值使用 NMS 去除冗余的预测框。YOLO v1 算法的核心思想在于将目标检测看作是一个回归问题, 经过一个神经网络就能够得到感兴趣对象的预测类别还有边界框的位置。YOLO v2 在 YOLO v1 的基础上做出进一步优化, 首先选用更强大的主干网络, 将该主干网络在 Imagenet 数据集上进行预训练, 使得网络模型学习到更多浅层的、通用的特征信息, 其次是使用 K-means 算法来聚类得到特定数据集的锚框大小和比例, 相对于固定的锚框大小和比例, 聚类出的锚框对数据集更加有针对性, 可以得到更精确的检测结果。2018 年 Redmon 等又提出了 YOLO v3 网络, 首先选用了更强的特征提取网络 Darknet53 作为特征提取的主干网络, 能够获取到更好的语义信息, 其次, 采用了感受野不同的三支检测头用来检测不同尺度大小的目标, 感受野大的检测头用来检测大目标, 感受野小的检测头用来检测小目标, 因此能够实现多尺度情况下的目标检测。此外, YOLO v3 采用 Sigmoid 函数代替了 Softmax 函数进行分类, 解决了不同类别之间相斥的问题, 网络对多类别的预测更加准确。最近的 YOLO v4 和 YOLO v5 算法在 YOLO v3 的基础上采用了许多新的优化策略, 如数据处理、损失函数、更好的主干网络和激活函数等, 进一步提高了单阶段目标检测算法的检测速度和精度。

SSD 算法结合了两阶段算法和一阶段算法的优点进行改进, 其结构如图 2-8 所示。首先通过 VGG16^[55] 主干网络进行特征提取, 得到具有高级语义信息的特征图, 然后通过六个不同特征图大小的卷积层进行边界框的分类和回归操作, 在每一个特征层, 借鉴 Faster R-CNN 的思路, 在六个卷积层处理后的特征图上的每个位置都设置不同比例和尺度的锚框, 最后通过非极大值抑制方法对六个特征图上的锚框进行筛选, 得到多尺度的预测结果。SSD 算法相对两阶段算法有更快的检测速度, 相对一阶段算法有更高的检测精度。

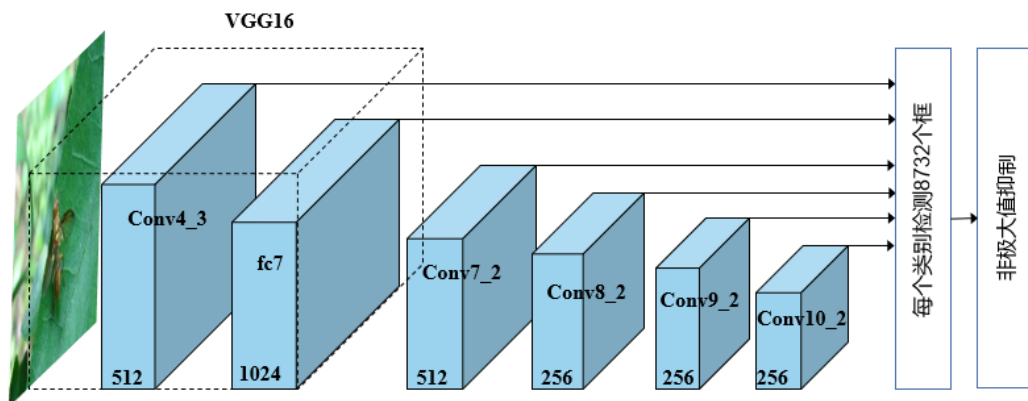


图 2-8 SSD 网络图

Fig. 2-8 SSD network diagram

2.3 小样本目标检测算法

小样本目标检测和传统的目标检测方法不同，其不再需要大量的图像训练模型才能检测新的类别。小样本目标检测将训练数据集分为有着大量数据的基类数据集和只有少量数据的新类数据集，新类的图像数据较少，一般不超过数十张。小样本目标检测首先在基类上训练，然后通过新类和基类进行模型微调。小样本目标检测分为两大类，一类是基于微调的方法，一类是基于元学习的方法，以下对这两种类型的经典网络模型进行详细的阐述。

2.3.1 基于微调的小样本目标检测算法

两阶段微调方法 TFA 是基于微调的基线方法，在网络模型上，TFA 将 Faster R-CNN 网络的主干网络、RPN 模块和 RoI 模块合起来称为类别不可知（class-agnostic）部分，将边界框分类器和回归器称为类别特定（class-specific）部分，如图 2-9 所示。在训练策略上，TFA 将网络模型的训练分为基础训练和微调两个阶段。在基础训练阶段，使用大量的基类数据集进行整个网络模型的训练，在微调阶段，构建了一个由少量的基类和新类数据组成的平衡数据集，然后固定类别不可知部分的参数，只用平衡数据集对类别特定部分进行微调。TFA 网络在基础训练阶段中可以从基类中学习到的与类别无关的特征信息，这些特征信息可以转移到新类的检测中而不需要进一步的参数更新。在微调阶段，只需要对类别特定部分进行参数的微调，就可以只通过少量的图像完成对新类别的检测。

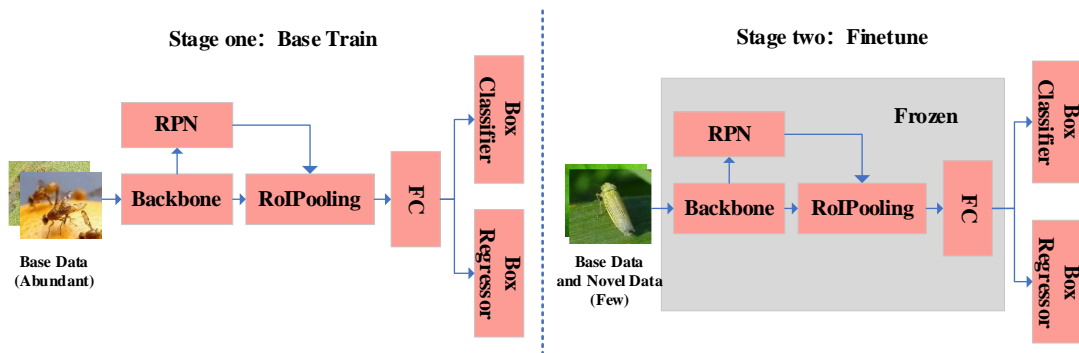


图 2-9 TFA 网络图

Fig. 2-9 TFA network diagram

同时，原始的 Faster R-CNN 网络在分类阶段使用全连接层进行预测框和真实边界框之间的相似性判断，TFA 在微调阶段将分类器替换为余弦相似性，如公式 2-6 所示， s_{ij} 是第 i 个预测框与第 j 个类别之间的相似性得分， α 为比例因子，默认值为 20。同全连接层的分类器相比，余弦相似性分类器对类内存在的方差可以起到抑制作用，使得相同类别数据更加聚集，在增加新类检测准确率的同时还可以减少基类检测精度的下降。

$$s_{i,j} = \frac{\alpha \mathcal{F}(x)_i^\top w_j}{\|\mathcal{F}(x)_i\| \|w_j\|} \quad (2-6)$$

2.3.2 基于元学习的小样本目标检测算法

基于元学习的小样本目标检测算法一般分为元训练和元测试两个阶段，在元训练阶段，通过不同的类别图像构成一个个元任务（episode），网络通过每个元任务学习具有泛化的特征，在元测试阶段，面对新的类别，可以快速进行模型训练完成对该类的检测。一个任务 T_j 可表示为 $T_j = S_j \cup Q_j = \{(I_1^j, M_1^j), \dots, (I_N^j, M_N^j)\} \cup \{(I_j^q, M_j^q)\}$ ，其中， S 表示支持集，存在着 N 个感兴趣的对象， Q 表示查询集图像， I 为具体的输入图像， M 为图像的标注信息。对支持集进行标注时，使用四通道方式，即在 RGB 三通道外，额外再使用一个掩膜通道将感兴趣的对象区域用数字 1 标出，图像的其他位置则设置为 0。一个元任务由 N 个不同类别图像的支持集和带注释的查询图像组成，而每个支持集类别包含 K 个感兴趣对象，即一个元任务的支持集有 $N \times K$ 个标注对象，这也被称为 N-Way K-shot 问题。

FSRW^[56]算法是元学习的基线方法，算法结构如图 2-10 所示，FSRW 整个网络模型是基于第一阶段网络 YOLO v2 进行的改进，在第一阶段网络中新增了重加权模块，该模块的输入为有着 $N \times K$ 个标注对象的支持集，使用卷积神经网络从支持集中得到 N 个类别向量，该向量代表每个类别的特征信息。元特征提取器以查询图像为输入，通过 YOLO v2 的主干网络提取查询集图像的特征，使用该特征与支持集的特征进行通道加权，之后在重加权后的特征上进行类别的回归预测。

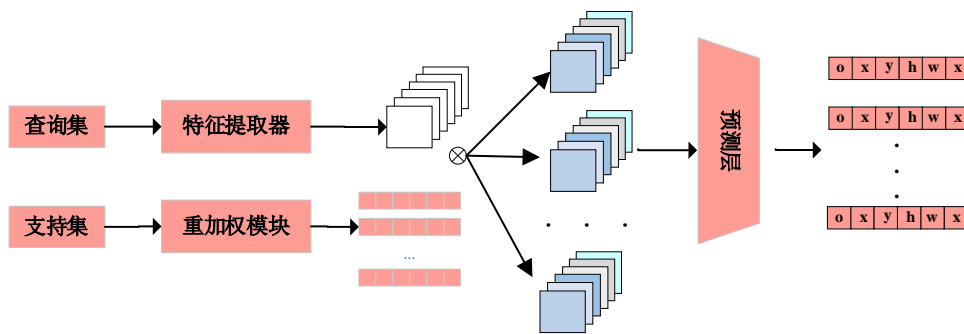


图 2-10 FSRW 网络图

Fig. 2-10 FSRW network diagram

2.4 评价指标

在目标检测中使用交并比(Intersection over Union, IoU)衡量预测框和真实框之间的相似性，IoU 值等于两个矩形框之间的交集除以两个框之间的并集，当交并比越大时，说明两个框越相似，最大值可为 1，即两框完全重合，最小值可为 0 时，说明两个框完全没有交集。通常，在网络中会设置相应的 IoU 阈值，当真实框与预测框之间的

交并比大于设置的阈值时，就认为成功检出了该对象，IoU 示意图如图 2-11 所示，

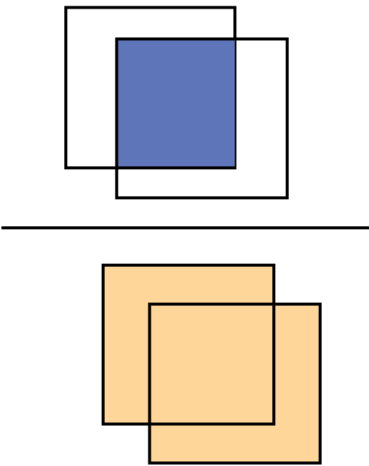


图 2-11 IoU 图

Fig. 2-11 IoU Diagram

为了对害虫检测网络的结果进行评估，本文使用一些常用的评价指标对检测结果进行评估，具体地，有精度（Precision，P），召回率（Recall，R），平均精度均值（Mean Average Precision，mAP）等指标。

关于样本预测与真实结果之间的关系，可以分为四种情况，如表 2-1 所示，其中，TP 表示正例被正确检出的数量，FP 表示负例被错误检出为正例的数量，TN 表示负例被正确归为负例的数量，FN 表示正例未被检出的数量。

表 2-1 混淆矩阵表

Table 2-1 Confusion matrix table

| 真实结果 | 预测结果 | |
|------|------|----|
| | 正例 | 负例 |
| 正例 | TP | FN |
| 负例 | FP | TN |

P 表示预测结果中所有正例数量占总的预测正例数量的比值，R 表示预测的正例结果占有所有真实正例的结果，R 也叫做查全率，P 和 R 的定义如公式 2-7 和 2-8 所示。AP 为不同召回率下的平均精度，根据每次得到的 P 和 R 值，绘制 PR 曲线，以召回率 R 为横坐标，准确率 P 为纵坐标，PR 曲线的面积即为 AP 值，其定义如式 2-9 所示。

$$P = \frac{TP}{TP + FP} \tag{2-7}$$

$$R = \frac{TP}{TP + FN} \tag{2-8}$$

$$AP = \int_0^1 P(R) dR \tag{2-9}$$

mAP 值为每个类别的平均精度值 AP 的均值, 定义如公式 2-10 所示, 其用于衡量网络模型在所有类别上的检测性能, 其中, n 为类别总数, AP_k 为第 k 个类别的 AP 值。mAP50 是指预测框与真实框之间的 IoU 值大于 0.5 时视为 TP, mAP@0.5:0.95 是 IoU 值从 0.5 到 0.95 每隔 0.05 为 IoU 值计算平均值得到。

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (2-10)$$

2.5 本章小结

本章首先介绍了卷积神经网络的基础知识, 包括卷积层、池化层、激活函数和全连接层。其次介绍了卷积神经网络中的目标检测算法, 包括两阶段算法和一阶段算法, 其中重点介绍了本文采用的两阶段 Faster R-CNN 算法。接着对两种经典的小样本目标检测方法进行了介绍, 包括基于微调的 TFA 方法和基于元学习的 FSRW 方法。最后, 对本文采用的评价指标进行了介绍。

第三章 基于改进 Faster R-CNN 的玉米害虫检测方法

针对复杂自然环境下玉米害虫的检测,本章基于改进 Faster R-CNN 网络对玉米害虫检测进行了研究。本章首先使用多种在线数据增强的组合增加网络输入数据的多样性,提高了模型的鲁棒性。其次,数据集中的害虫有不同的长宽比例,比如幼虫通常是细长条的,蛾类等害虫则偏向正方形,本章采用结合遗传算法的 K-means 聚类算法获得玉米害虫数据集特有的初始锚框比例。针对复杂自然环境下小目标害虫难以检测的问题,使用递归金字塔特征模块代替原有的 FPN 模块,可以获取输入图像数据更多的特征信息。然后,使用 Soft-NMS 后处理减少模型对害虫的漏检。本章为大样本条件下玉米害虫的准确检测提供了技术支持。

3.1 Faster R-CNN 目标检测网络

本章提出的方法是基于 Faster R-CNN 网络改进的,本节将首先对 Faster R-CNN 网络的组成和主要流程进行介绍,随后对网络的各个组成部分进行更为具体的说明。

Faster R-CNN 主要由特征提取网络、区域生成模块、RoI 模块和分类回归网络四部分组成,其实现流程如图 3-1 所示,首先使用特征提取网络提取图像的高级语义信息,得到特征图,然后将该特征图送到 RPN 模块中,生成大量的感兴趣区域,接着将感兴趣区域在特征图中的位置通过 RoI 模块统一为固定大小的候选区域,最后经过分类和回归的全连接层支路得到最终的检测结果。

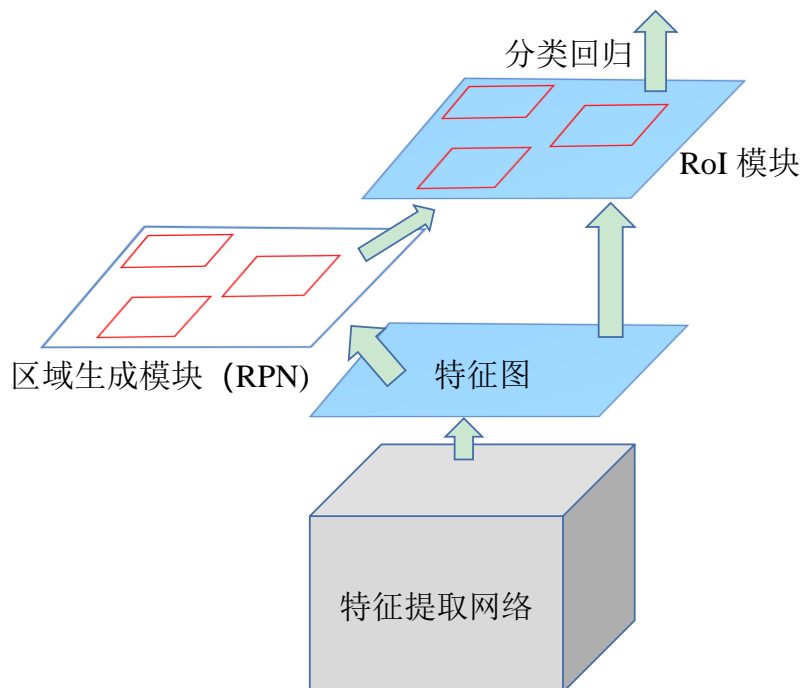


图 3-1 Faster R-CNN 网络结构图

Fig. 3-1 Faster R-CNN network structure diagram

3.1.1 特征提取网络

害虫检测的第一步是进行特征提取，特征提取网络作为整个检测网络的基础，决定着最终检测精度的优劣。为了获取足够强的特征图，通常需要大量的卷积层进行特征提取。原始的 Faster R-CNN 网络使用的特征提取网络为 VGG16 网络，但该网络随着深度的增加，在计算量增大的同时也会发生网络退化问题。因此近些年 Faster R-CNN 更多地选择 ResNet50^[57]作为特征提取网络。

ResNet50 相对于 VGG 网络最大的改进点是残差单元，如图 3-2 所示。左边为基本残差单元，右边为改进的瓶颈残差单元。基本残差单元加入了短路连接，将输入通过恒等变化映射到结果值，该短路连接可以保证在不增加网络参数的情况下，输出至少为输入本身，梯度的反向传播不会出现退化。右边的瓶颈残差单元则通过添加 1x1 的卷积层，可以将通道数先降低为较小的值做 3x3 卷积操作，之后再升高回原始的通道数，这样可以将网络变得更“宽”，获得更多的通道信息，但网络参数不会增加很多。ResNet50 网络主要采用右侧的瓶颈残差网络。

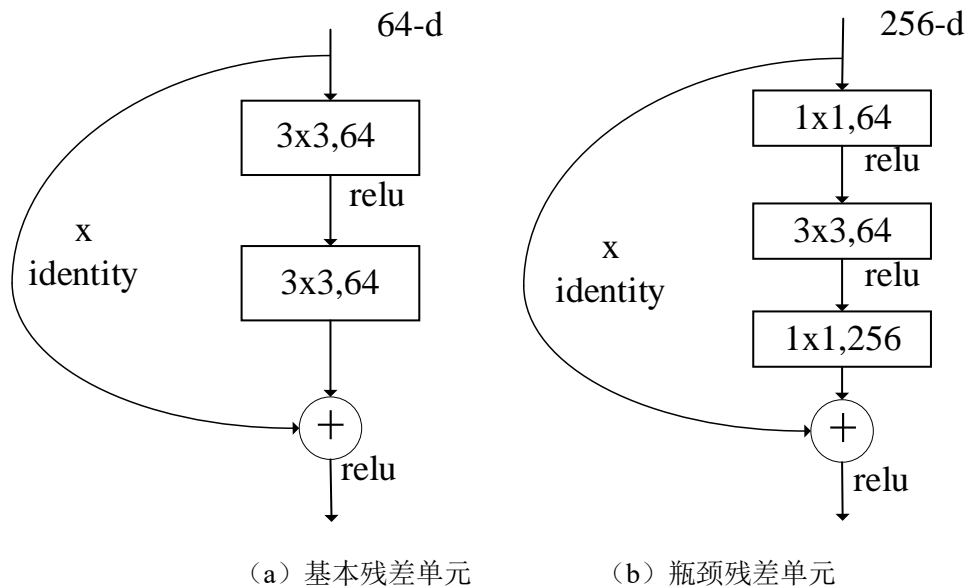


图 3-2 残差网络单元

Fig. 3-2 Residual network unit

Resnet50 网络的结构图如图 3-3 所示，Stage2 到 Stage5 的基本组成单元相同，都由瓶颈残差单元组成，Conv Block 在瓶颈残差网络的短路分支添加 1x1 的卷积层用于维度的变化，该模块会将特征图的尺度降低一半，同时通道数扩大 2 倍，Identity Block 模块在短路分支上就是输入本身，通道维度没有发生改变，因此可以串联多个在一起。网络的 Stage2 到 Stage5 分别包含 3+4+6+3=16 个瓶颈残差单元，每个单元包含有 3 层卷积层，再加上最开始的卷积层和结束的分类层，所以网络一共包含 50 层。Faster R-CNN 在特征提取网络使用 Resnet50，更深的网络层可以得到更丰富的特征信息，同时解决了网络退化的问题，使得网络的训练和测试误差都大幅降低。

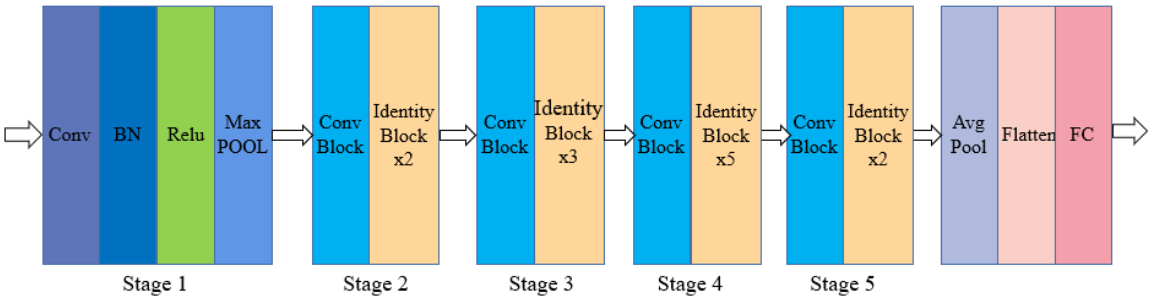


图 3-3 ResNet50 网络图

Fig. 3-3 ResNet50 network diagram

3.1.2 区域生成模块

RPN 模块是 Faster R-CNN 网络的主要创新点，在 Fast R-CNN 中使用的是选择性搜索算法进行候选区域的生成，该算法生成候选区域时间长，且不能集成到网络中完成端到端的检测。为解决这些问题提出了 RPN 模块，其结构如图 3-4 所示，RPN 的锚框机制为特征图上的每个网格生成 3 个尺度、3 种比例的锚框，一共形成 9 个预定义的锚框（图中 k 的取值则为 9）。使用两个 1x1 的卷积并行支路对每个锚框进行卷积计算，其中分类分支只区分是否为对象，不识别具体的对象类别，如果存在对象则视为前景，否则视作背景，所以生成 $2k$ 个分类得分。另外一个回归支路则需要预测锚框相对于真实框的偏移量，偏移量的计算包含中心点坐标值和宽高一共四个值。

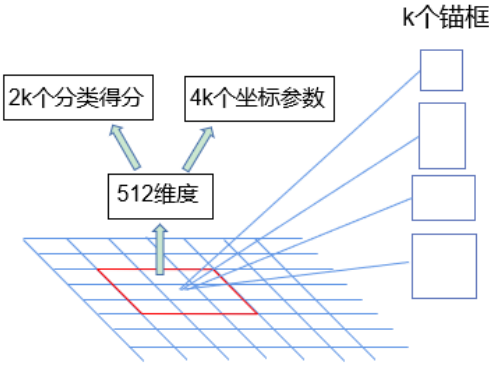


图 3-4 Region proposal network

Fig. 3-4 Region proposal network

3.1.3 RoI 模块

通过 RPN 模块可能得到包含目标的候选区域，为了随后的检测子网进行分类和回归处理，需要将候选区域统一成固定大小的维度。在候选区域映射回特征图的过程中，可能不会刚好是像素点的整数坐标。RoI Pooling 会将浮点数像素值直接取整。另外，在将候选区域统一为固定大小的过程中，RoI Pooling 也会直接将出现的浮点数进行取整操作，这两次的量化取整无法保留候选框在原图中的完整图像，造成信息的缺失，对于一些小目标对象来说，这是灾难性的打击。何凯明等提出了 RoIAlign 替换原有的

RoI Pooling 操作, 图 3-5 为 RoI Pooling 和 RoI Align 的过程处理图。RoI Align 首先在候选区域映射回特征图的过程中, 会保留像素的浮点数值, 其次在候选区域统一为固定大小的过程中, 会将候选区域使用 7×7 大小的网格进行平均分割, 在每个网格内均匀地采样 n 个点, 这些采样点是利用双线性插值计算得到的, 然后采用最大池化技术得到该网格的值。RoIAlign 算法通过双线性插值和浮点数计算, 将候选区域统一为固定大小的维度, 能够让对象的定位更加准确。

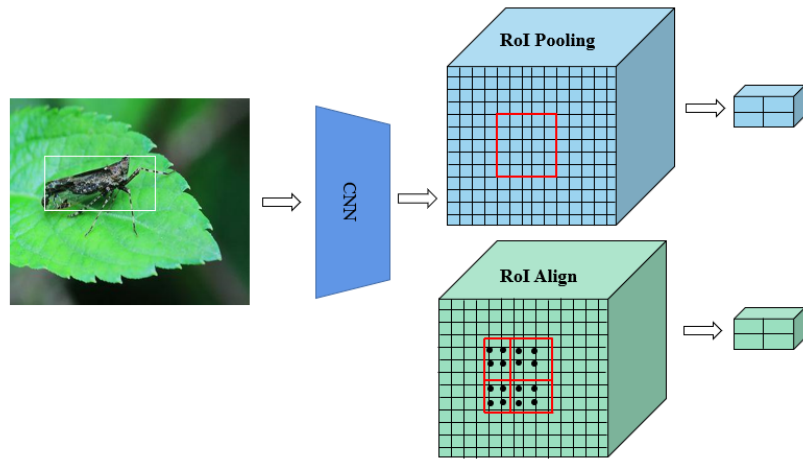


图 3-5 RoI Pooling 和 RoI Align

Fig. 3-5 RoI Pooling 和 RoI Align

3.1.4 分类和回归

对于 RoI Align 处理后的输出, 先通过两层共享的全连接层将候选区域的特征映射到样本特征空间中, 然后将结果分别送入到分类分支和回归分支。在分类分支上, 先使用独有的全连接层进一步处理特征映射, 接着使用 Softmax 函数计算属于每个类别的概率, 得到分类的结果值。在回归分支上, 同样先使用一个独有的全连接层进一步处理, 得到预测的回归结果值, 对输出的预测框进行微调, 获得更加精准的预测框, 分类和回归的模块如图 3-6 所示。

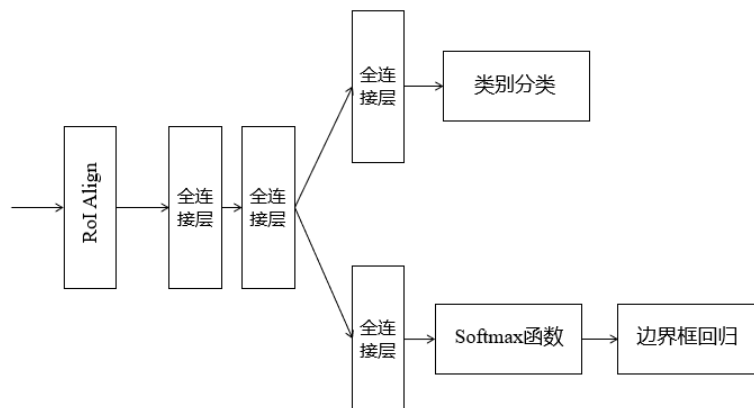


图 3-6 分类和回归网络图

Fig. 3-6 Classification and regression network diagram

3.2 害虫数据集和数据处理

3.2.1 数据集介绍

本章数据集来源于大型公开的 IP102 数据集，IP102 数据集包含大田和经济两种作物类型的害虫，具体的大田作物有水稻、玉米、小麦三种，经济作物有甜菜、苜蓿、葡萄、柑橘、芒果五种。该数据集包含超过 7 万多张害虫图像用于分类任务，包含 1 万 8 千多张图像用于害虫的检测任务。本章从 IP102 数据集中选取了玉米作物中全部的害虫，一共包含蛴螬、蝼蛄、金针虫、白边蛾等 13 种，共 3894 张图像，害虫的具体情况如下表 3-1 所示。将数据集按 8: 2 的比例划分为训练集和测试集，确保两个数据集中都包含着每类害虫的图像，得到训练集 3112 张和测试集 782 张。

表 3-1 玉米害虫数据集

Table 3-1 Corn pest dataset

| 害虫名称 | 害虫数量 | 害虫名称 | 害虫数量 |
|---------------------------|------|-----------------------------|------|
| 蛴螬 (grub) | 436 | 红蜘蛛 (red spider) | 160 |
| 蝼蛄 (mole cricket) | 868 | 玉米螟 (corn borer) | 225 |
| 金针虫 (wireworm) | 298 | 夜蛾 (army worm) | 206 |
| 白边蛾 (white margined moth) | 49 | 蚜虫 (aphids) | 879 |
| 小地老虎 (black cutworm) | 126 | 白星花金龟 (Potosiabre vitarsis) | 203 |
| 大地老虎 (large cutworm) | 153 | 桃蛀螟 (peach borer) | 97 |
| 黄地老虎 (yellow cutworm) | 194 | | |

3.2.2 数据增强

网络模型的输入数据越多，可以学习到的特征就越丰富，训练结果的鲁棒性就越好，玉米害虫数据集包含 13 种害虫 3894 张图像，对于深度模型训练这些图像数量还是偏少，可能导致模型出现过拟合的现象。深度学习的目标检测增加输入图像数据的方法有在线数据增强和离线数据增强。离线数据增强在网络训练前就通过数据增强得到扩充的数据，这种方式可视化好，但是需要占据很多的存储空间。在线数据增强则是在网络一边训练时一边使用数据增强，这样可以减少存储空间的占用且灵活性更好。本章采用 Albumentations 图像处理库进行图像的在线数据增强。该库相比 pytorch 自带的图像处理方法有更多的数据增强方法，有 70 多种图像预处理方法，且处理图像变化的速度很快，数据增强可以很灵活地进行组合。

单张图像样本的数据增强一般分为两类，一类是图像的几何变化，一类是图像的像素内容变化。几何变化不会对图像原本的信息进行变化，而是对部分像素位置重新排列组织，内容变化则对图像的像素值进行了变动。本章采用的单张图像几何变化有旋转、平移、缩放、转置等。采用的单张图像像素内容变化有颜色变化、模糊、加入

噪声等。本章对害虫图像进行在线数据增强，采用包括几何变化和-content变化在内的多种数据增强组合的形式，其中每个数据变化都是在一定的随机概率才发生，而不是一定要发生的。具体的数据增强组合如下所示。

1、以 0.15 的概率从随机旋转 90 度、平移缩放旋转、水平旋转、转置中选择一种变化。

2、以 0.15 的概率从 RGB 平移、色调饱和度值、随机伽马变换中随机选择一种。

3、以 0.1 的概率从随机亮度对比度、随机伽马变换中随机选择一种。

4、以 0.1 的概率从图像均值平滑滤波、中心模糊、高斯模糊中随机选择一种。

如图 3-7 为部分数据增强后的示意图，可以看出增强后的图像有较明显的变化，丰富了数据的多样性，使得网络训练更加鲁棒。



图 3-7 数据增强处理图

Fig. 3-7 Data augmentation processing diagram

3.3 改进的递归特征金字塔结构

基线的 Faster R-CNN 网络只使用主干网络最后一层的特征图进行分类和定位。一般认为在网络的主干网络进行特征提取的过程中，低层的特征图会保留更多细节信息，有利于检测目标的空间位置，在不断的卷积操作后，高层的特征图能够提炼出图像的高度语义信息，这些语义信息有助于目标的分类。很多研究都考虑将低层和高层的特

征图结合在一起,使得网络能够既保留低层的空间信息,又能够利用具有高级语义信息的高层特征图。

FPN 网络如图 3-8 所示,整个结构由三部分组成,一个是自底向上的过程,一个是自顶向下的过程,还有横向连接三种。在网络自底向上不断卷积进行特征提取的过程中,会逐渐降低特征图的宽高,逐渐加大特征的通道信息。例如图中输入图像大小为 1000×600 ,每经过一个 conv 卷积块之后,宽高降为一半,通道扩张为 2 倍,因为 conv1 包含的特征信息很初级,没有提取到对分类有用的信息,只使用之后的 conv2、conv3、conv4、conv5 参与后续的特征融合。为了使得不同层级的特征可以相互融合,需要通过横向连接将高层的特征图首先通过 1×1 的卷积将通道统一为相同的大小。高层的特征图想要与低层进行特征融合,要先上采样统一为相同大小的尺寸。上采样方法采用的是最近邻线性插值,最近邻插值可以保障在上采样的过程中最大程度地保留特征图的语义信息。在网络自顶向下的过程中,可以将高层丰富的语义信息传递给低层特征图,横向连接可以使得高层和低层的特征融合,融合后的每个特征层拥有不同的感受野,可以对图像中不同尺度的对象进行预测。最顶层的特征图预测大尺度对象,最底层的特征图预测小尺度对象。

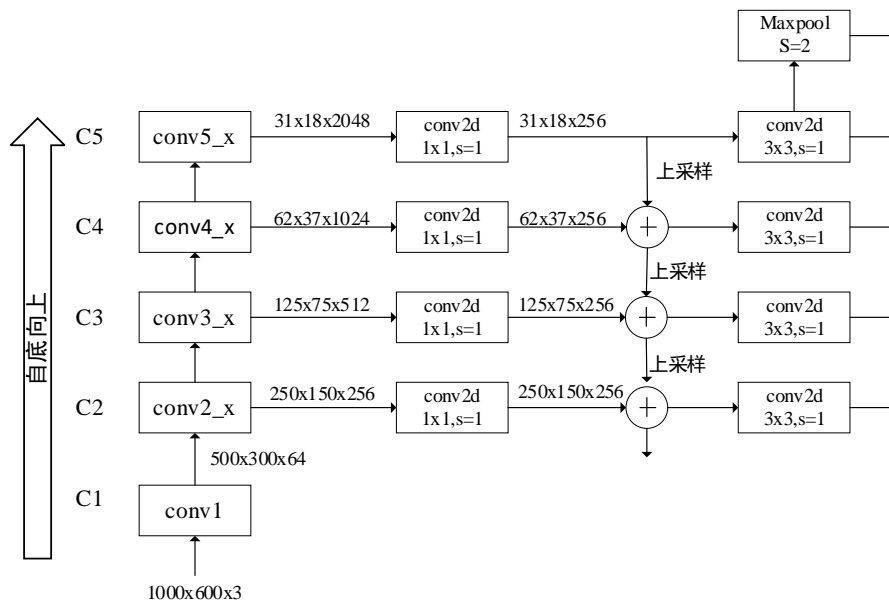


图 3-8 FPN 网络图

Fig. 3-8 FPN network diagram

针对自然环境下小目标害虫的检测,借鉴于人类通过两次查看和多次思考的优秀机制,为了获得更加丰富的特征信息,本章将 Faster R-CNN 的 FPN 模块替换为递归特征金字塔 (Recursive Feature Pyramid, RFP),该网络的结构图如 3-9 可见,将 FPN 融合后的特征输出再次输入到主干网络中,进行第二次的循环。该网络可以通过两次或者多次查看图像信息,对特征进行不断的提纯处理,形成更加强大的特征表示,通

过让递归后生成的一些错误信息回传到主干网络后进一步的修正和加强，可以更加准确地预测小目标害虫对象。

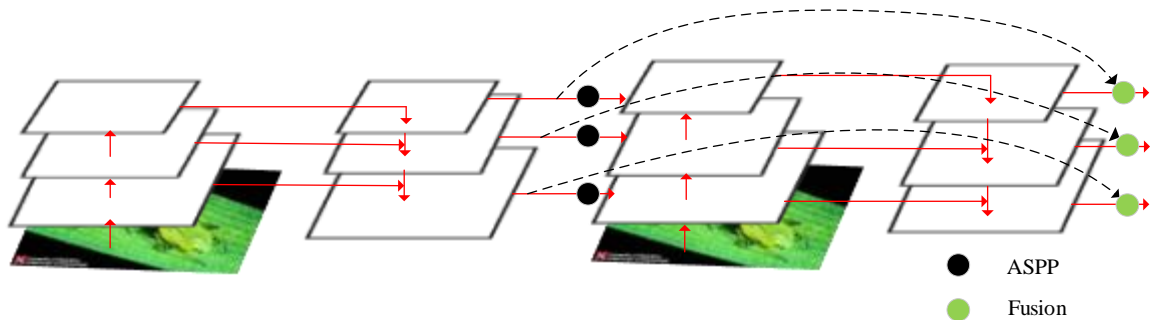


图 3-9 RFP 网络图

Fig. 3-9 RFP network diagram

RFP 网络中 ASPP 的具体结构如图 3-10 所示，该模块用于将 FPN 的输出转变为递归特征金字塔的输入之一，ASPP 模块的输入输出尺度相同，一共由四个分支组成，其中一个分支是利用全局池化层得到图像全局的特征压缩信息，其他的三个分支通过三种不同尺度的空洞卷积构成，空洞卷积在不增加模型参数数量的前提下可以提高特征图的感受野，这三个分支上的空洞卷积拥有大小不同的空洞率，融合这四个模块能够进一步地提高模型对小目标害虫的特征提取能力。将这些经过 ReLU 函数后按通道连接起来，形成最终的输出结果。

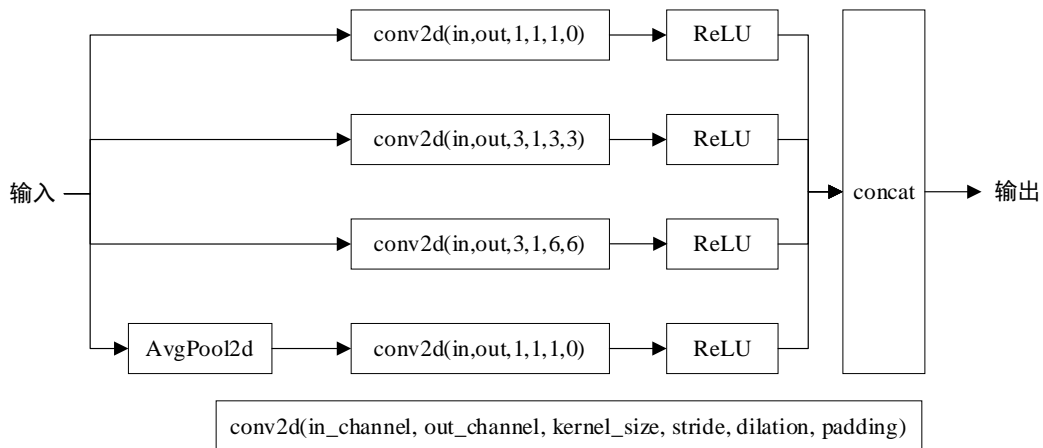


图 3-10 ASPP 网络图

Fig. 3-10 ASPP network diagram

RFP 网络中 Fusion 融合模块如图 3-11 所示，其 f_i^t 中表示第 t 阶段第 i 层的输出， f_i^{t+1} 表示第 $t+1$ 阶段第 i 层的输出，首先利用 1×1 卷积将第 $t+1$ 阶段和第 t 阶段的特征图对齐通道，接着对第 $t+1$ 阶段使用 Sigmoid 函数得到注意力分数 σ ，然后与上一阶段的特征图进行融合。

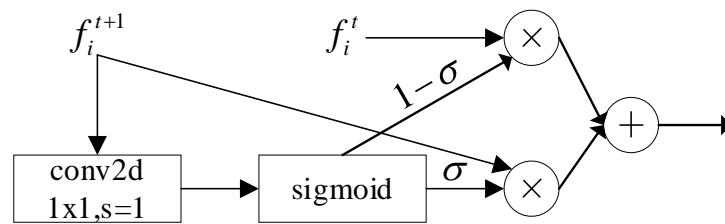


图 3-11 Fusion 网络图

Fig. 3-11 Fusion network diagram

3.4 优化网络锚框设置

RPN 模块以主干网络的特征图为输入，通过锚框生成算法，在特征图上生成多种对应的锚框，所生成锚框是固定大小和比例的，比例一般是 $\{1:2, 1:1, 2:1\}$ 这三种，大小一般是根据 FPN 的多个分支来确定，现在带有 FPN 特征金字塔的尺度大小一般是 32、64、128、256 和 512 这五种，其中每个尺度映射到 FPN 的每一层，这样特征图上的每个像素点都对应生成 15 个锚框，最初始的锚框尺度和比例对于网络的收敛速度和定位的精度有着非常重要的作用。尤其是不同玉米害虫的长宽比例差距很大，很多害虫的幼虫一般呈现出细长条的极端长宽比，因此推断出更加合适的初始锚框比例，可以得到更加准确的检测框。

K-means 算法可以根据训练集中对象的真实尺寸生成一组新的锚框，这些锚框是在本章的训练集基础上生成的，因为原始的欧式距离在大尺度锚框上的误差会远大于小尺寸锚框的误差，本研究采用 1-IoU 作为衡量样本点间距离的指标，如果锚框与簇中心的 IoU 值越小，则表示两者间距离越近。以下就是 K-means 算法的具体步骤：

- 1、初始化选择 k 个样本点作为聚类中心，
- 2、使用 1-IoU 值算出数据集中的样本点到每个聚类中心的距离值，把这些样本点分给距离最近的聚类中心，形成一个个簇，
- 3、根据每个簇的所有样本，重新生成每个簇的聚类中心，
- 4、重复步骤 2、3，直到聚类的结果不再发生变化或者达到预设的迭代步数，
- 5、最终 k 个聚类中心代表 k 个不同锚框，这些锚框可以用于后续的候选框生成。

遗传算法是进化算法中的一种，参考进化生物学中的遗传、变异、自然选择等现象发展而来。遗传算法用于求解最优化的搜索算法。遗传算法的求解过程如下所示：

第一步，初始化进化代数计数器 $t=0$ ， T 是最大进化代数，随机生成 M 个个体作为初始群体 $P(t)$ ；第二步，个体评价，计算 $P(t)$ 中各个个体的适应度值；第三步，选择运算，将选择算子作用于群体；第四步，交叉运算，将交叉算子作用于群体；第五步，变异运算，将变异算子作用于群体，并通过以上运算得到下一代群体 $P(t+1)$ ；第六步，终止条件判断，如果 t 不大于 T ： $t=t+1$ ，转到第二步；如果 $t>T$ ，则终止输出解。

本章首先使用 K-means 聚类算法得到锚框，然后使用遗传算法对锚框进行优化，具体的步骤如下所示：

第一步：使用 K-means 聚类算法得到 n 个锚框；

第二步：使用遗传算法随机对锚框的宽高进行变异，如果变异后效果变得更好（使用适应度进行评估）就将变异后的结果赋值给锚框，如果变异后效果变差就跳过，默认变异 1000 次。

第三步：将最终变异得到的锚框按照面积进行排序并返回。

本章考虑到玉米害虫尺度的多样性，最终选择聚类中心 n 的值为 9，结合遗传算法的 K-means 算法最终得到的 9 个锚框尺寸为[141, 135]、[251, 192]、[299, 296]、[491, 360]、[712, 277]、[392, 507]、[759, 395]、[603, 636]、[826, 556]，相对应的比例为 0.77、0.95、1.01、1.04、1.31、1.36、1.49、1.92、2.57。因为锚框尺度一般是成对出现的，为了尽可能地覆盖所有害虫比例。在原始的 0.5、1.0、2.0 的基础上增加了 3 组锚框比例，最终采用的锚框比例为[0.5, 1.0, 2.0, 1.5, 3.0, 0.3, 5.0, 0.2, 0.7]。

3.5 优化预测框的筛选算法

Faster R-CNN 网络在 RPN 模块和分类检测模块会生成大量的预测框，如果不加以限制，就会存在一个检测对象有多个预测框的情况，去除大量冗余的重叠预测框，找到最合适的预测框就成为了一个棘手的问题。

NMS 算法作为后处理过程，可以从众多的候选框中找到最想要的预测框，NMS 的公式如式 3-1 所示，NMS 的工作流程是：将所有候选框按置信度的大小依次排好顺序，从这些候选框中选择置信度最大的作为预测框，计算该候选框和其他所有的候选框间的 IoU 值，如果该 IoU 值大于设置的阈值则删除该候选框，保留其它的候选框。以上是一次遍历的过程，接着去除最大置信度的候选框，再从剩下的候选框中重复以上步骤，直到没有候选框为止。

$$s_i = \begin{cases} s_i, & iou(M, b_i) < N, \\ 0, & iou(M, b_i) \geq N \end{cases} \quad (3-1)$$

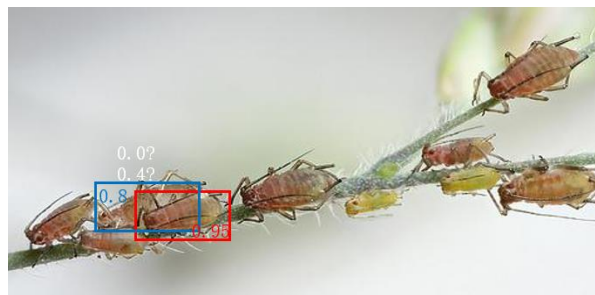


图 3-12 NMS 分析图

Fig 3-12 NMS analysis chart

当图像存在一些很靠近的对象时, NMS 算法会判定多个距离过近的候选框为同一个对象, 导致感兴趣对象的漏检。如图 3-12 所示, 蚜虫等玉米害虫通常聚集在一起, 如果使用 NMS 算法, 可能会漏掉很多的目标对象。

Soft-NMS 算法不再将与最大置信度边界框重叠的候选框去除, 而是将置信度的逐渐降低。具体步骤如图 3-13 所示:

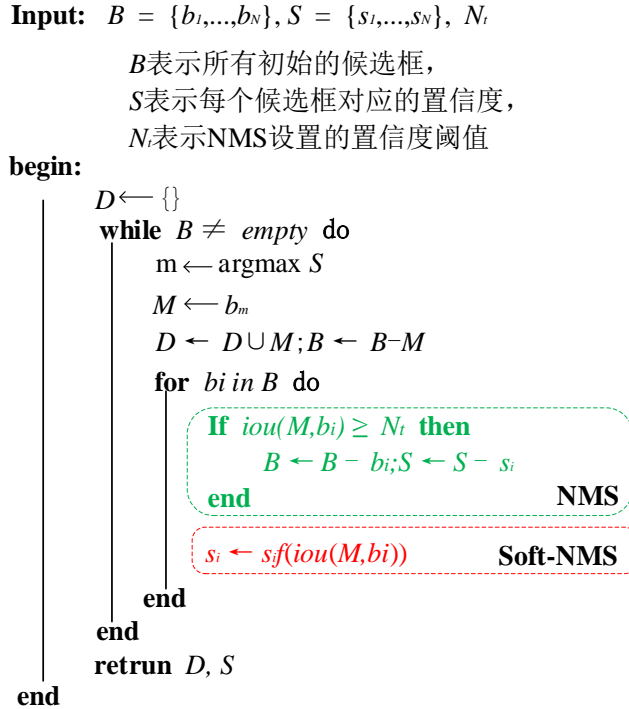


图 3-13 Soft-NMS 算法流程

Fig. 3-13 Soft-NMS algorithm process

使用 Soft-NMS 的简单表示如公式 3-2 所示, 其中, N_t 为设定的 IoU 阈值, M 是当前置信度最大的候选框, B_i 为与候选框 M 有交集的其他候选框, s_i 表示第 i 个候选框的置信度。交并比值越大的候选框的置信度减低的越大。但该函数存在一个不连续点, 将其进一步写为公式 3-3 的形式。

$$s_i = \begin{cases} s_i, & \text{iou}(M, b_i) < N_t, \\ s_i(1 - \text{iou}(M, b_i)), & \text{iou}(M, b_i) \geq N_t \end{cases} \quad (3-2)$$

$$S_i = S_i e^{-\frac{\text{iou}(M, b_i)^2}{\sigma}}, \quad \forall b_i \notin D \quad (3-3)$$

3.6 实验结果与分析

3.6.1 实验环境

本模型的实验在 ubuntu 操作系统上进行, 以 python 为开发语言, 基于 pytorch 深度学习框架实现, 版本为 1.10.0, 使用的显卡为 Tesla T4, 显存为 16GB。cuda 版本为 10.2, torchvision 版本为 0.11.0, cudnn 版本为 8.0.4。本章更加详细的实验环境配置如

表 3-2 所示。

表 3-2 实验环境表

| Table 3-2 Experimental environment table | |
|--|--|
| 实验环境 | 环境配置 |
| 操作系统 | ubuntu16.04.12 |
| Cpu | Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz |
| Gpu | Tesla T4 |
| 内存容量 | 252GB |
| 硬盘空间 | 1T |
| Pytorch | 1.10.0 |
| Torchvision | 0.11.0 |
| Cuda | 10.2 |
| Cudnn | 8.0.4 |

3.6.2 模型训练

学习率决定着能否在网络学习的过程中找到参数的最优解，是非常重要的参数之一，学习率设置过大，会造成网络的训练出现不收敛的现象，当学习率过小，可能会造成模型陷入局部最优或者收敛过慢，因此本模型选择初始学习率为 0.02，动量为 0.9，最大的迭代轮数为 24epoch。使用线性递减的学习率，在第 10 个和第 16 个 epoch 分别将学习率减低为上一轮的 0.01 倍，这样可以让网络在一开始的时候以一个较大的学习率快速地达到较高的检测精度，之后再慢慢的找到最优解。另外，使用 Imagenet 数据集在 ResNet50 上的预训练参数进行模型初始化，网络训练使用热身（warmup）策略，在模型训练的开始阶段使用设定初始学习率的百分之一，然后在 500 个迭代次数 iteration 内逐渐增长到初始的 0.02 学习率，这样设置的好处，可以使得网络避免掉入局部最优的情况，得到更高的检测精度，同时可以加快模型的训练速度。

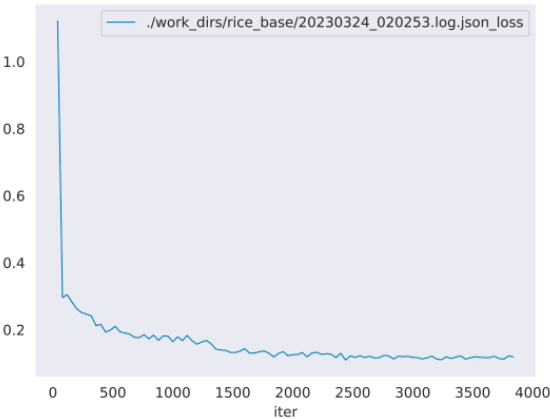


图 3-14 模型训练损失图

Fig. 3-14 Model training loss graph

图 3-14 是本模型的训练损失曲线，这个损失是分类损失和回归损失的总和，从图中可以看出，随着迭代次数的不同增加，模型的损失呈逐渐减少的趋势，在一开始的时候，因为设置的学习率较大，网络的损失下降很快，当迭代次数达到 2500 次后，这时一方面学习率变小，另一方面模型可能已经接近最优解了，模型开始趋于稳定，这说明本模型可以达到预期的训练效果。

3.6.3 实验结果分析

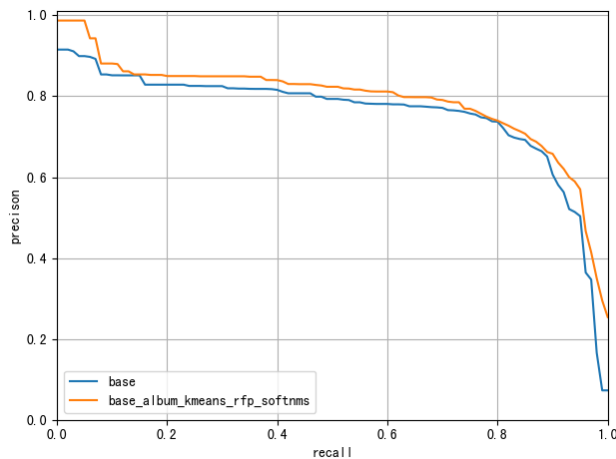


图 3-15 模型 PR 图

Fig. 3-15 Model PR graph

本章画出了原始的 Faster R-CNN 和改进后网络的 PR 曲线图，如图 3-15 所示，曲线与坐标轴围成的面积为网络的平均精度，从图中可以看出，在不同的召回率时，本章所提出的网络相比原始的 Faster R-CNN 在精度上有较大的提升，在整体的平均精度，本章提出的网络也优于原始的 Faster R-CNN 网络。

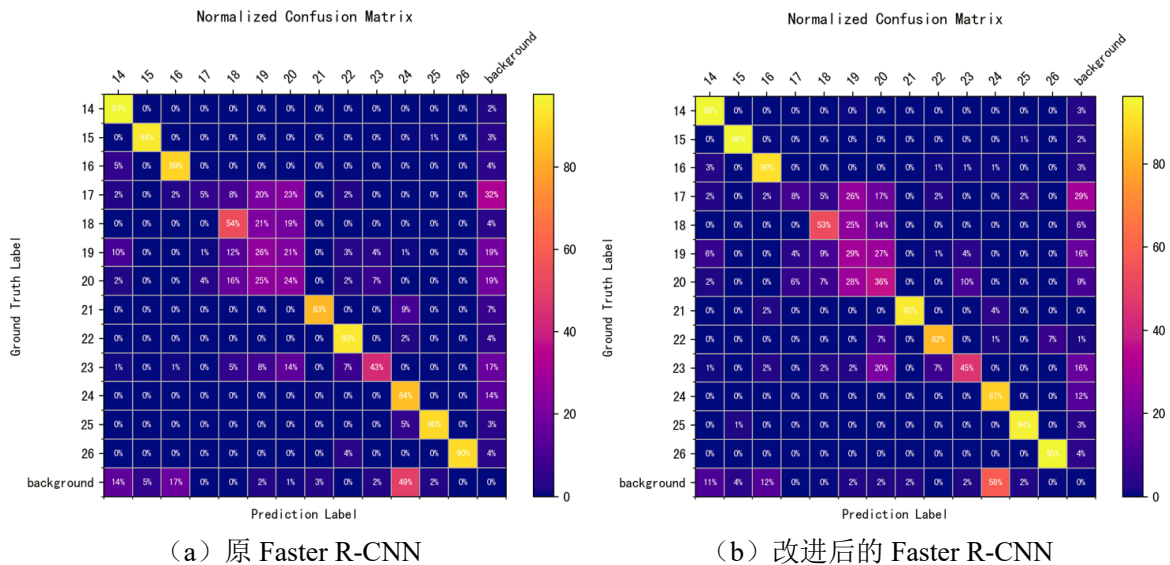


图 3-16 混淆矩阵图

Fig. 3-16 Confusion matrix diagram

为了说明本模型在类别分类上的优势，画出了原始 Faster R-CNN 和改进后的模型

在玉米害虫数据集上的混淆矩阵，如图 3-16 所示，混淆矩阵通常用来衡量分类的准确性，它的横向是预测的标签，纵向是真实的标签，对角线表示分类正确的百分比，其他则是错误分类的百分比，用颜色来表示分类的准确度，从紫色到黄色分类的准确率越来越高，黄色的方格表示分类正确率最高。从图中可以看出，本研究在分类准确率基本都大于原始的 Faster R-CNN 网络，尤其是第 20 类黄地老虎，本模型相比原始的 Faster R-CNN 网络提高了 12 个点的分类正确率。

表 3-3 各类别害虫的检测结果对比

Table 3-3 Comparison of detection results of various types of pests

| | Faster R-CNN | 改进的 Faster R-CNN |
|-------|--------------|------------------|
| 蛴螬 | 96.04% | 96.81% |
| 蝼蛄 | 97.20% | 97.85% |
| 金针虫 | 88.13% | 92.46% |
| 白边蛾 | 21.86% | 28.17% |
| 小地老虎 | 74.97% | 81.74% |
| 大地老虎 | 29.11% | 31.59% |
| 黄地老虎 | 35.50% | 44.61% |
| 红蜘蛛 | 95.02% | 98.58% |
| 玉米螟 | 94.32% | 97.8% |
| 夜蛾 | 73.22% | 78.02% |
| 蚜虫 | 84.11% | 86.56% |
| 白星花金龟 | 94.76% | 95.22% |
| 桃蛀螟 | 98.02% | 98.20% |
| mAP | 75.56% | 79.05% |

表 3-3 表示在测试数据集上原始 Faster R-CNN 和改进的 Faster R-CNN 的性能表现。从表中可以观察到，本章的改进网络对所有 13 种害虫在 mAP 指标上都有一定的增加。其中，在黄地老虎上的提升最明显，从 35.50%提升到 44.61%，增加了 9.11 个百分点，在小地老虎、大地老虎等也有较大的检测精度提升，分别提高了 6.77 个百分点和 2.48 个百分点。这表明本改进网络对于复杂自然环境中易混淆的害虫检测有一定的提升。另外，本网络对蚜虫的检测精度也从 84.11%提高到 86.56%，有 2.45 个百分点的提升，这说明本改进网络在小目标的检测性能也有良好的检测性能。另外，本改进网络对其它的害虫类别检测精度也均有一定的提升。

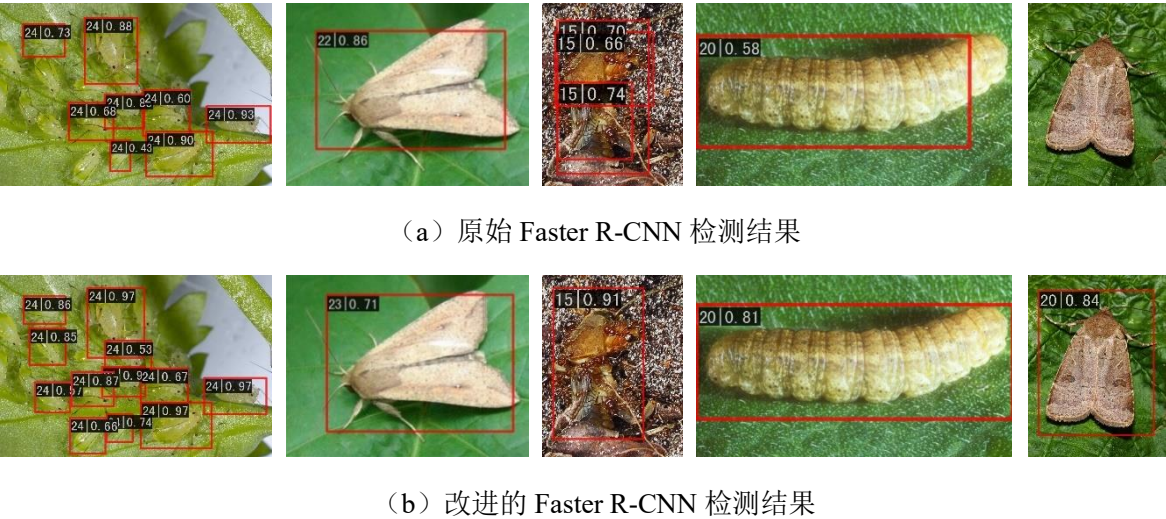


图 3-17 检测结果图

Fig.3-17 Detection result graph

上图 3-17 列举一些原始的 Faster R-CNN 网络和改进后的网络对害虫检测差异性比较大的一些图像，图（a）为原始的 Faster R-CNN 网络的检测结果，图（b）为改进后的 Faster R-CNN 网络的检测结果，从图中可以看出，在与背景相似的蚜虫小目标上，原始的 Faster R-CNN 漏掉了 many 蚜虫，而本章改进的模型则检测出更多的蚜虫数量。Faster R-CNN 将第二列图像的夜蛾错误识别为相似的玉米螟了，本网络不仅正确识别出夜蛾类别且定位更加精准。第三列该害虫是一只正被蚂蚁啃食的死掉蝼蛄害虫，其与背景的颜色很相近、不易辨认，原始的 Faster R-CNN 预测出多个检测框，且没有正确识别出害虫类别，改进后的网络可以得到正确的检测框和类别。本网络在第四列黄地老虎幼虫的定位更加精准，同时将第五列黄地老虎的成虫正确检出。

3.6.4 消融实验

消融实验用于探究网络的结构改进和训练策略等对网络精度的影响。为了验证本章所提出改进的有效性，基于选用的玉米害虫数据集，本章采用不同的改进策略分别进行了对比实验，结果如表 3-4 所示，选用 mAP 和 AR 两个指标对模型进行评价。

表 3-4 实验结果表

| Table 3-4 Experimental result table | | | |
|---|--------------|-------------|--|
| 改进策略 | AP50/% | AR/% | |
| FRCN | 75.56 | 64.7 | |
| FRCN +Album | 77.05 | 64.3 | |
| FRCN +Album+K-means | 78.14 | 65.3 | |
| FRCN +Album+K-means+RFP | 78.62 | 64.8 | |
| FRCN +Album+K-means+RFP+Soft-NMS | 79.05 | 74.9 | |

根据表 3-3 可见，改进后的模型相比原始的 Faster R-CNN 在 mAP 和 AR 指标上

分别提升了 3.49 个百分点和 10.2 个百分点。其中，数据增强改进提升较为显著，相比原始的 Faster R-CNN 网络在 mAP 值提升了 1.49 个百分点，主要原因是数据增强极大地丰富了输入图像的多样性，网络可以从不同类型的输入图像中学习到更加丰富的特征信息。在网络中使用了结合遗传算法的 K-means 算法后，网络的 mAP 和 AR 分别提高了 1.09 个百分点和 1 个百分点，这说明玉米害虫数据集中的真实边界框有较大的尺度变化，增加了这些尺度的锚框检测后，在精度和召回率上都有较明显的提升。将网络的颈部由 FPN 换为 RFP 后，网络的精度提升了 0.48 个百分点，这说明将 FPN 递归地查看两次，可以得到更加丰富的特征信息，有利于提高害虫检测的精度。最后，将网络的后处理由 NMS 升级为 Soft-NMS 函数，在精度有 0.43 个百分点的提升，但在召回率 AR 指标上提升巨大，由 64.8% 提高到 74.9%，提高了 10.1 个百分点，可以有效减少网络的漏检。

3.6.5 不同模型的对比

为了同其他不同网络模型进行对比，本章选取了两个一阶段检测网络 SSD 和带 SPP 结构的 YOLO v3 网络以及两阶段经典网络 Faster R-CNN 为对比模型。为保证对比的公平性，实验均在相同的硬件环境下进行实验，选用相同的输入图像大小，通过足够的迭代次数令这几个网络都收敛到最优结果，如表 3-5 可见最后的结果值。

表 3-5 与其他模型对比

Table 3-5 Compare with other models

| 模型 | 主干网络 | AP50/% | AR/% | FPS | FLOPs | 模型大小/MB |
|--------------|-----------|-------------|-------------|-------------|---------------|--------------|
| YOLO v3-SPP | DarkNet53 | 76.3 | 61.0 | 30.3 | 118.02 | 61.59 |
| SSD | VGGNet16 | 55.3 | 48.2 | 25.0 | 212.40 | 25.35 |
| Faster R-CNN | ResNet50 | 75.6 | 64.7 | 27.1 | 131.16 | 41.18 |
| 本模型 | ResNet50 | 79.1 | 74.9 | 14.1 | 122.69 | 65.72 |

由表可见，本模型的检测精度优于其他的一阶段检测算法和两阶段检测算法，达到了最优的 79.1%，相比 SSD 网络有 23.8 个百分点的提升，相比 YOLO v3-SPP 网络有 2.8 个百分点的提升，相比经典的两阶段检测网络 Faster R-CNN 有 3.5 个百分点的提升。本模型的召回率为 74.9%，也高于其他的三个模型。这是因为本章通过在数据集上应用了一系列的数据增强处理，加强了网络训练的数据多样性，使得网络可以学习到的特征更加丰富，其次，使用 K-means 算法得到适应于数据集的锚框比例，得到了检测更加准确的网络，然后，使用递归特征金字塔替换 FPN 模块，加强了网络的特征提取能力。最后，使用 Soft-NMS 减少了网络对于密集的害虫的漏检率。

从表 3-5 中可以看出，本章的改进模型在 FPS、浮点运算数(floating point operations, FLOPs)和模型大小三个评价指标上并不是最优的。原始 Fast-RCNN、YOLOv3-SPP 和

SSD 网络模型的检测速度要优于本文的改进模型。从 FLOPs 方面分析网络的计算复杂度,本章提出的改进网络比 SSD 和原始的 Faster R-CNN 的浮点数运算量要小,比 YOLO v3-SPP 网络要大一点,但差距不大。从网络模型大小分析,本章提出的改进网络要大于表中其他目标检测的网络模型,但是对于现在的计算机,占用的空间并不是很大,所以产生的影响并不是最关键的。

3.7 本章小结

本章采用了具有更高精度的两阶段目标检测算法 Faster R-CNN 作为基础模型,针对模型对蚜虫等小尺寸的精度不足,以及害虫数据集存在蚜虫和玉米螟等尺度差距很大的问题,对特征金字塔模块进行改进,提出了基于 Faster R-CNN 改进的玉米害虫检测网络,该网络在 FPN 的基础上可以对特征进行两次融合,大大提高了特征的代表能力,从总体上提高了模型的类别检测和定位能力。本章使用带遗传算法的聚类算法对玉米害虫数据集的对象尺寸进行统计分析,得到了更加合适的锚框比例,提高了网络对幼虫这种长宽比值较大的检测性能。针对 NMS 会因为图像中密集对象的存在而造成网络漏检的情况,本章使用了 Soft-NMS 算法。所以,改进后的模型在玉米害虫检测上达到了较高的准确率。

第四章 小样本条件下的农作物害虫检测方法

在使用基于传统深度学习的相关方法进行害虫检测时,需要采集大量的害虫图像,但是害虫具有区域性和季节性,且种类众多,难以获得每一类害虫的大量图像,为了只通过少量害虫图像就可以实现检测功能,本章在 TFA 的基础上,提出了一种基于微调的小样本害虫检测方法。利用多尺度特征融合模块将全局信息模块与空间金字塔池化模块融合在一起,加强了网络对不同尺度的处理能力并提高了特征间的关联性,丰富了网络的特性信息表示。采用度量学习改进分类损失函数,使得网络中不同类别之间彼此远离,相同类别之间更加靠近,减少了害虫的错误分类,提高了新类别害虫的类别检测精度。将网络的回归损失函数替换为 CIoU Loss^[58],加快了损失收敛的速度,同时提高了害虫检测的预测框定位精度。由于害虫不同生命周期形态差异过大,将数据集中害虫的成虫和幼虫当作不同类别进行检测,完成了细粒度的害虫检测任务。

4.1 小样本害虫数据集构建

本章数据来源于 IP102 数据集,该数据集图像均来自自然真实环境,包括大田作物(水稻、玉米、小麦、甜菜和紫花苜蓿)和经济作物(葡萄、柑橘和芒果)害虫共计 102 类,包含 18983 张图像。这些害虫处在不同生命周期的不同阶段,有卵、幼虫、蛹和成虫,同一个类别在不同的生命周期差异过大,如图 4-1 所示为草地贪夜蛾的成虫和幼虫,不同类别在同一生命周期内又极为相似。本章选取其中的幼虫和成虫两个阶段作为研究对象,将其作为不同的类别重新编码,组成了新的数据集 PEST80,为了对模型更好地进行验证,新数据集去除了图像数量小于 50 张的类别,且从数量过多的类别中随机选取了 500 张,一共获得 64 个类别,共 13489 张图像。将 64 类害虫的成虫和幼虫分别编码,得到 80 个类别,其中成虫 64 类,包含 10778 张图像,幼虫 16 类,包含 2711 张图像。PEST80 的类别数量在 50-500,按 8:2 的比例将数据集划分为训练集和验证集,训练集图像包含 10791 张,测试集图像 2698 张。本研究从 80 个类别中选取 60 类作为基类,包含 49 类成虫和 11 类幼虫,20 类作为新类,包含 15 类成虫和 5 类幼虫,害虫类别和数量的详细信息见表 4-1 所示,其中标粗的为新类害虫。



图 4-1 草地贪夜蛾的幼虫和成虫

Fig. 4-1 Larvae of early and adults

表 4-1 PEST80 数据集

Table 4-1 PEST80 dataset

| 害虫名称 | 成虫 /幼虫 | 害虫名称 | 成虫 /幼虫 | 害虫名称 | 成虫 /幼虫 |
|-----------------------|-----------|----------------------------------|-----------|------------------------------|-----------|
| rice leaf roller | 90/84 | bird cherry-oataphid | 67/— | Ampelophaga | 279/— |
| Rice leaf caterpillar | 55/58 | pentahaleus major | 71/— | Lycorma delicatula | 338/— |
| asiatic rice borer | 65/101 | longlegged spider mite | 59/— | Xylotrechus | 157/— |
| yellow rice borer | 59/— | wheat phloeothrips | 92/— | Cicadella viridis | 206/— |
| rice gall midge | 93/— | wheat sawfly | 105/— | Miridae | 500/— |
| brown plant hopper | 111/— | flea beetle | 345/— | Panonychus citri McGregor | 99/— |
| white backed plant | 78/— | cabbage army worm | 77/169 | Icerya purchasi Maskell | 158/— |
| small brown plant | 51/— | beet army worm | 122/291 | Ceroplastes rubens | 65/— |
| rice water weevil | 155/— | Beet spot flies | 69/— | Chrysomphalus aonidum | 50/— |
| rice leafhopper | 117/— | meadow moth | 94/— | Aleurocanthus spiniferus | —/60 |
| grub | —/436 | beet weevil | 146/— | Aleurocanthus spiniferus | 64/— |
| mole cricket | 500/— | sericaorient alismots chulsky | 81/— | Dacus dorsalis(Hendel) | 115/— |
| wireworm | 127/298 | alfalfa weevil | 62/— | Prodenia litura | 186/228 |
| black cutworm | 174/126 | flax budworm | 239/171 | Adristyrannus | 121/— |
| large cutworm | 77/76 | alfalfa plant bug | 227/— | Scirtothrips dorsalis Hood | 75/— |
| yellow cutworm | 100/94 | tarnished plant bug | 349/— | Lawana imitata Melichar | 128/— |
| red spider | 160/— | Locustoidea | 500/— | Salurnis marginella Guerr | 64/— |
| corn borer | 225/199 | lytta polita | 244/— | Deporaus marginatus Pascoe | 172/— |
| army worm | —/176 | legume blister beetle | 435/— | Rhytidodera bowrinii white | 138/— |
| aphids | 500/— | blister beetle | 500/— | Sternochetus frigidus | 62/— |
| Potosiabre vitarsis | 203/— | odontothrips loti | 65/— | Cicadellidae | 500/— |
| peach borer | 97/144 | oides decempunctata | 134/— | | |
| english grain aphid | 130/— | Pseudococcus comstocki Kuwana | 51/— | | |

4.2 小样本害虫目标检测网络结构

在本文的 2.3.1 小节已经对两阶段迁移网络 TFA 进行了较为详细的介绍，本章基

于 TFA 进行改进, 提出新的网络模型, 如图 4-2 所示。主要是以下三点改进: 首先, 使用多尺度特征融合增强小样本条件下的特征信息表示, 同时缓解不同尺度的害虫检测问题。其次, 在网络的分类头引入负边距余弦交叉熵损失函数, 该损失函数可以提高类内紧凑性和类间相离性, 减少新类别害虫的误检测。最后, 在网络的边框预测方面, 将 Smooth L1 损失函数替换为 CIoU Loss 函数, 可以进一步提高害虫检测的预测框定位精度。

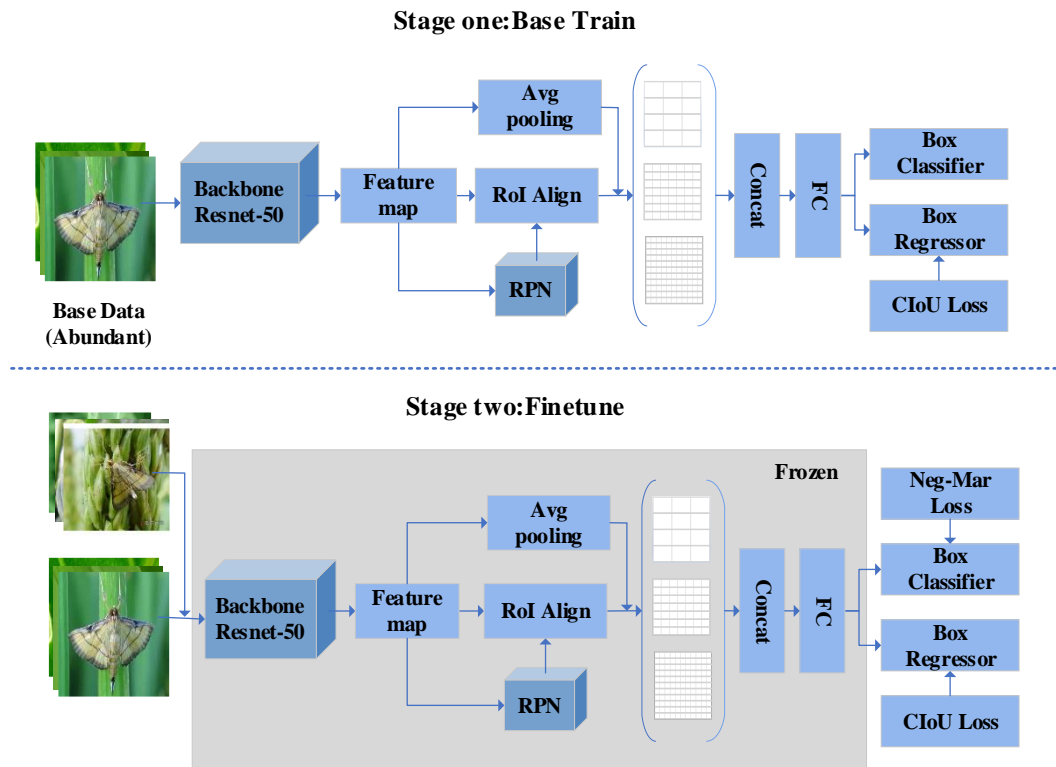


图 4-2 所提模型框架

Fig. 4-2 Proposed model framework

4.3 多尺度特征融合模块

在 Faster R-CNN 网络中, 将主干网络得到的特征图和 RPN 模块得到的候选区域联合输入到 ROIAlign 模块, 生成 7×7 固定尺度的池化特征图, 使用该池化特征图进行随后的分类和回归。在网络训练过程中, 只使用 7×7 尺度的池化特征层可能会导致特征信息的丢失, 在通用的目标检测网络中, 能够通过大量的样本图像减少这种特征信息丢失的现象。而小样本条件下, 新类别参与网络训练的样本图像很少, 可以获得的类别特征信息有限, 这种信息丢失的情况可能会造成害虫的误检测。其次, 由于新类别的样本数量很少, 不同尺度的害虫对网络预测的影响被放大, 新类中存在着小目标害虫 (比如蚜虫) 和大目标害虫 (比如蛾类) 等不同尺度的类别, 难以通过很少的新类图像实现对不同尺度害虫的检测, 这大大降低了网络对不同尺度类别的泛化能力。为了解决上述存在的问题, 本章通过 ROIAlign 生成多个尺度的池化特征图, 这些特征

可以实现相互之间的数据互补,通过融合不同尺度的特征信息,丰富最终的表达能力,缓解输入新类图像不足的情况,同时多个不同尺度的特征图融合还能够缓解尺度变化问题。具体结构如图 4-3 所示。

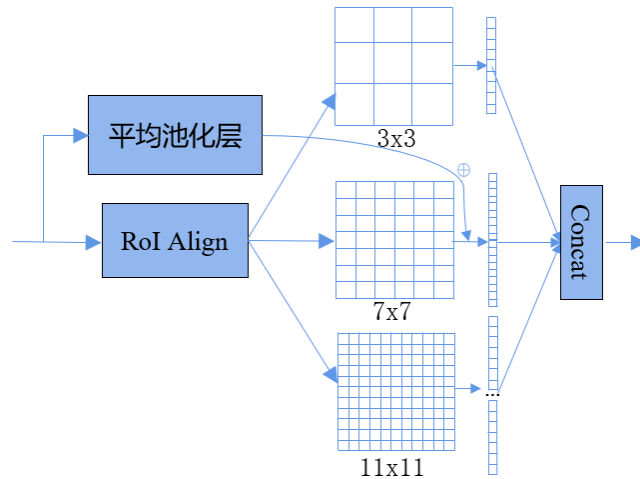


图 4-3 多尺度特征融合图

Fig. 4-3 Multi-scale feature fusion diagram

原始的 Faster R-CNN 生成固定 7x7 大小的池化特征图,本章选取了 3x3、7x7、11x11 三种尺度,将通过 RPN 模块生成的候选区域调整为不同尺度的池化特征图,通过这种并行的池化技术可以获得更加全面丰富的特征信息,11x11 这种大尺度池化特征图可以关注到局部上下文信息,有利于蚜虫等小目标害虫的检测,3x3 这种小尺度池化特征图则可以关注到全局信息,有利于蛾类等大目标害虫的检测。每个池化特征图都包含不同级别的语义信息,为了尽可能避免特征信息的丢失,将三种尺度的池化特征图分别展平为一维特征向量,然后通过 concat 函数拼接在一起,拼接操作可以直接增加特征的数量,实现局部特征信息和全局特征信息的融合,丰富了池化特征图的表达信息。另外,将主干网络产生的特征图作为一个全局语义信息整体地加入到池化特征图中,具体是先通过一个平均池化层,将主干网生成的特征图处理为 7x7 的大小,然后和 7x7 的池化特征图作 add 函数相加操作,平均池化可以保留图像整体的数据特征,add 操作则在 7x7 的池化特征中额外增加了图像的全局特征信息。在上述多重特征信息的叠加处理下,能够很好的应对样本量较少的新类别检测任务。

4.4 负边距分类损失函数

TFA 的基训练阶段和传统的目标检测差别不大,本章仍然选择 Faster R-CNN 的损失函数作为分类损失函数,在网络的微调阶段,虽然多尺度特征融合在样本量少的情況下能够形成更丰富的特征表示,解决语义信息不足的问题,但是新类容易受到一些相似的基类样本影响,会错误地将新类样本映射到相似的基类特征空间,造成基类和新类的类别混淆。本章引入了负边距分类损失函数,该函数可以有效地区分不同类别,

同时避免在检测过程中新类投影到相似基类特征空间中，减少新类害虫的误检。

在 Faster R-CNN 网络中，使用 Softmax 分类器的交叉熵损失函数计算每个类别的概率进行类别预测，Softmax 损失函数如公式 4-1 所示：

$$L_s = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{f_{y_i}}}{\sum_{j=1}^C e^{f_j}} \quad (4-1)$$

其中， C 表示训练类别数量， N 表示训练样本数量， f_j 表示类别 j 对应得全连接层输出结果， f_j 进一步可表示为类别权重向量 W 和特征向量 x 之间的内积，即 $f_j = W_j^T x = \|W_j\| \|x\| \cos \theta_j$ ， θ 为两个向量之间的夹角，因此损失 L_s 可表示为：

$$L_s = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{\|W_{y_i}\| \|x_i\| \cos(\theta_{y_i})}}{\sum_{j=1}^C e^{\|W_j\| \|x_i\| \cos(\theta_j)}} \quad (4-2)$$

但是 Softmax 函数只对预测框进行类别概率预测，不能让同一个类别彼此聚集，不同类别之间彼此远离，形成很强的类别区分性。文献[59]将正的大的边界加入到 Softmax 函数中，提出了 L-Softmax 函数，如公式 4-3 所示。

$$L_l = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{\|W_{y_i}\| \|x_i\| \psi(\theta_{y_i})}}{e^{\|W_{y_i}\| \|x_i\| \psi(\theta_{y_i})} + \sum_{j=1, j \neq y_i}^C e^{\|W_j\| \|x_i\| \cos(\theta_j)}} \quad (4-3)$$

其中， $\psi(\theta) = \begin{cases} \cos(m\theta), & 0 \leq \theta \leq \pi/m \\ D(\theta), & \pi/m \leq \theta \leq \pi \end{cases}$ ， $D(\theta)$ 也是单调递减函数，等同于 $\cos(m\theta)$ 。

m 表示为边距参数。因为 \cos 函数在 $[0, \pi]$ 的范围内是单调递减函数，当 m 越大，相比于原始的 $\cos(\theta)$ ， $\cos(m\theta)$ 就越小，因此不同类别的预测值更加靠近，不同类间的决策边界需要为原先的 m 倍大小才能完成类别间的区分，模型学习更加困难，从而分类效果更好。根据文献[60]，将边距参数带入公式 4-3 中，可进一步表示为公式 4-4：

$$L_l = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{\alpha(s(z_i, W_{y_i}) - m)}}{e^{\alpha(s(z_i, W_{y_i}) - m)} + \sum_{j=1, j \neq y_i}^C e^{\alpha(s(z_i, W_j))}} \quad (4-4)$$

其中边距参数 $m \geq 0$ ， α 为温度系数， s 为相似性函数，但是 L_l 损失不适用于小样本这种开放场景（open-set）中，正的大的边界在小样本情况下加剧了类别混淆，从而错误地将新类样本映射到相似的基类特征空间中，降低模型在新类别的分类性能。文献[61]通过理论分析和一系列的实验证明，使用适当的负边距，即 $m \leq 0$ 时，可以解决新类中不同的数据投影到多个不同的基类特征空间的问题，降低了新类的类内方差，有利于新类检测性能的提高。

在 TFA 预测模块的分类损失函数中，在第一阶段仍然使用全连接层的分类器，在

微调阶段,使用余弦相似度的分类器替换了原始 Faster R-CNN 的基于全连接层的分类器,该定义如式 4-5 可见,用于衡量输入特征 $F(x)$ 和不同类别权重向量 w 之间相似度。本章使用该余弦函数作为相似度函数,将公式 4-5 带入公式 4-4 中,可以得到最终的负边距余弦 Softmax 函数,如公式 4-6 所示。

$$s_{i,j} = \frac{\alpha \mathcal{F}(x)_i^\top w_j}{\|\mathcal{F}(x)_i\| \|w_j\|} \quad (4-5)$$

$$L = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{\alpha(\cos(z_i, W_{y_i}) - m)}}{e^{\alpha(\cos(z_i, W_{y_i}) - m)} + \sum_{j=1, j \neq y_i}^C e^{\alpha(\cos(z_i, W_j))}} \quad (4-6)$$

4.5 CIoU 边框回归损失函数

TFA 网络的损失函数包括分类损失和回归损失两部分,上文对分类损失进行了介绍并做出相应改进。原始的 TFA 网络所用的回归损失为 Smooth L1 Loss 函数,如公式 4-7 所示,该损失函数对于检测网络中的离群点和异常点,梯度变化相对较小,网络训练时具有很强的鲁棒性。

$$\text{Smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1, \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (4-7)$$

在 TFA 网络中实际使用 Smooth L1 Loss 回归损失的形式如公式 4-8 所示:

$$L_{reg}(t_i, t_i^*) = \sum_{i \in (x, y, w, h)} \text{Smooth}_{L_1}(t_i - t_i^*) \quad (4-8)$$

其中, $t_i^* = (t_x^*, t_y^*, t_w^*, t_h^*)$ 为真实框的中心点和长宽的坐标值, $t_i = (t_x, t_y, t_w, t_h)$ 为预测框的对应的坐标值,回归损失由四个点的损失函数直接相加得到,用于衡量预测框和真实框之间的差异,Smooth L1 Loss 函数假设预测框的四个点坐标是彼此不相关的,分别对独立的四个点进行损失计算,然而实际上检测网络对于预测框的评价标准是与真实框之间的交并比值,该交并比需要考虑到四个点之间的关系。

当不同的预测框与真实框的 Smooth L1 损失值相等时,但 IoU 指标值有可能差别巨大,无法准确的反映出边界框的重合程度。如图 4-4 所示,图 4-4 (a)、(b) 中两组框有相同的 Smooth L1 损失值,但是两组框的重叠面积不同, IoU 值差别很大。因此 Smooth L1 Loss 作为回归损失函数时和预测框的评价标准不一致,不能够准确反映网络对预测框的评估。

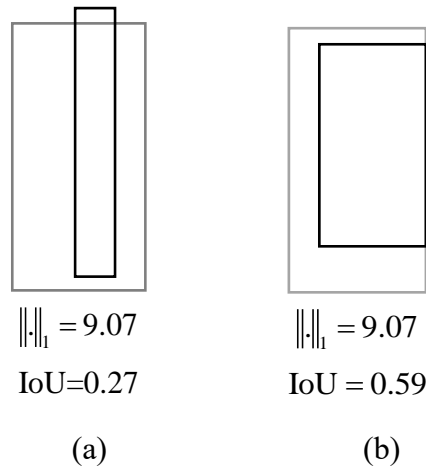


图 4-4 Smooth L1 损失和 IoU 损失对比

Fig. 4-4 Comparison of Smooth L1 loss and IoU loss

为了解决以上问题,同时进一步优化边界框回归损失函数,本章将 Smooth L1 Loss 函数替换为 CIoU Loss 函数。CIoU Loss 的计算公式如式 4-9 所示。它解决了预测框和真实框重叠的问题,另外还考虑了预测框与真实框中心点的距离问题以及两个框之间的长宽比。

$$L_{CIoU} = 1 - IoU + \frac{\rho(b, b^{gt})}{c^2} + \alpha v \quad (4-9)$$

其中,公式中 IoU 表示预测框与真实框之间 IoU 值,即用于衡量预测框和真实框之间重叠面积。

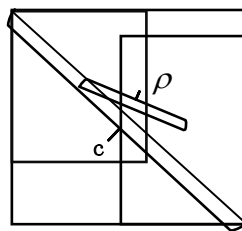


图 4-5 CIoU 结构图

Fig. 4-5 CIoU structure diagram

公式 4-9 的第二项用于计算真实框与预测框之间中心点的距离。如图 4-5 所示,其中, b 是预测框的中心点坐标, b^{gt} 是目标框的中心点坐标, ρ 用于计算两个中心点间的欧式距离, c 是预测框和真实框组成的最小外接矩形的对角线距离值。

αv 的定义如式 4-10 和 4-11 所示,用于衡量预测框和真实框之间的长宽比。其中, α 为平衡系数, v 表示预测框与真实框间长宽比的相似性, w 、 h 为预测框的宽度和高度, w^{gt} 、 h^{gt} 为目标框的宽度和高度。

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (4-10)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (4-11)$$

相比 Smooth L1 损失函数, CIoU Loss 函数同时考虑了预测框和真实框间的重叠面积、中心点间的距离和长宽比例这三种因素。两个边界框之间的重叠面积与检测网络对预测框的评价标准一致, 两个边界框中心点之间的欧式距离可以加速网络的训练速度, 还解决了当 IoU 等于 0 时损失计算失灵等问题。考虑预测框和真实框之间的长宽比, 能够进一步提高了预测框的精度。

4.6 实验结果与分析

4.6.1 实验平台和参数设置

本章模型的训练和测试都在相同的环境下进行。操作系统使用 Ubuntu 16.04 LTS, 图形处理器为 NVIDIA Tesla T4, 显存为双卡 16GB。使用 pytorch 1.10.0 作为深度学习框架训练。

本章模型采用具有特征金字塔和 ResNet-50 做主干网络的 Faster R-CNN 作为基础模型。优化器使用随机梯度下降 (Stochastic Gradient Descent, SGD) 算法, 设置网络的动量恒定为 0.9, 权重衰减稀疏设置为 0.0001, 因为样本数量的不同, 因此在基础训练和微调阶段选择不同的学习率进行网络模型的训练, 基训练时学习率为 0.02, 小样本微调阶段不同 shot 条件下统一学习率设置为 0.002。

4.6.2 实验结果分析

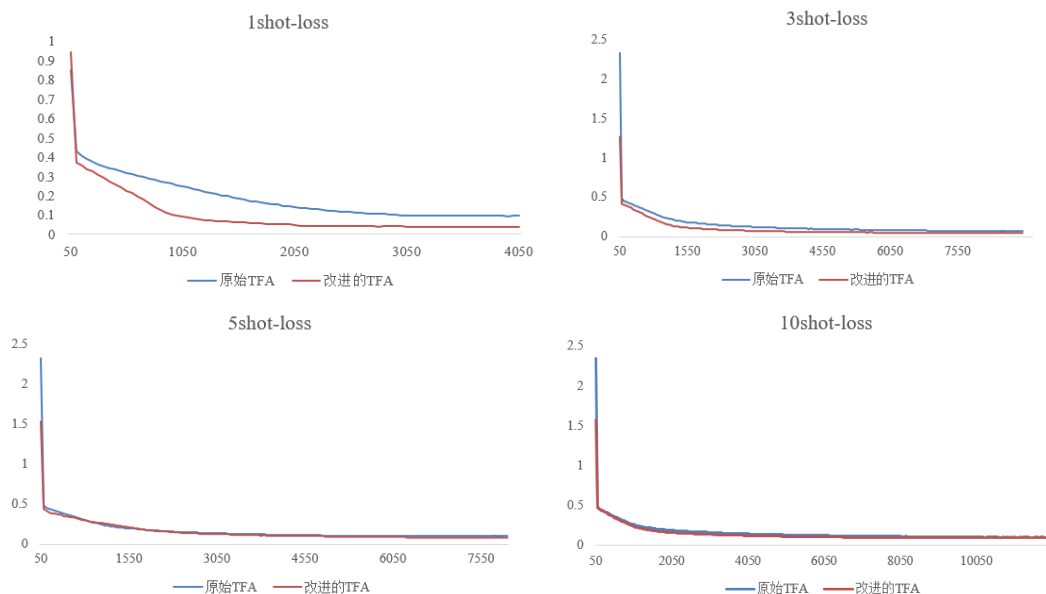


图 4-6 训练损失对比图

Fig. 4-6 Training loss comparison chart

以上为原始的 TFA 和改进后 TFA 的训练损失图, 从图中可以看出, 各个网络模型均已经达到收敛, 且在不同 shot 条件下, 改进后的 TFA 网络在训练过程中的损失值

始终要小于原始的 TFA，同时，相比于原始的 TFA 网络，改进后的 TFA 网络的也更快地达到网络模型收敛。

表 4-2 害虫检测结果
Table 4-2 Pest detection results

| 害虫 | 1shot | | 3shot | | 5shot | | 10shot | |
|--------------------------|--------|---------------|--------|---------------|--------|---------------|--------|---------------|
| | 原始 | 改进 | 原始 | 改进 | 原始 | 改进 | 原始 | 改进 |
| | TFA | TFA | TFA | TFA | TFA | TFA | TFA | TFA |
| rice leaf roller | 0.202 | 0.146 | 0.254 | 0.452 | 0.365 | 0.488 | 0.523 | 0.628 |
| asiatic rice borer | 0.16 | 0.201 | 0.17 | 0.252 | 0.278 | 0.318 | 0.502 | 0.541 |
| yellow rice borer | 0.38 | 0.368 | 0.481 | 0.524 | 0.51 | 0.574 | 0.577 | 0.638 |
| brown plant hopper | 0.032 | 0.115 | 0.04 | 0.089 | 0.021 | 0.142 | 0.318 | 0.515 |
| white backed plant | 0.499 | 0.387 | 0.305 | 0.318 | 0.244 | 0.392 | 0.407 | 0.379 |
| rice water weevil | 0.162 | 0.169 | 0.301 | 0.332 | 0.332 | 0.389 | 0.416 | 0.324 |
| rice leafhopper | 0.144 | 0.118 | 0.397 | 0.422 | 0.413 | 0.489 | 0.486 | 0.474 |
| wireworm | 0.233 | 0.337 | 0.356 | 0.265 | 0.306 | 0.281 | 0.45 | 0.539 |
| black cutworm | 0.277 | 0.404 | 0.297 | 0.401 | 0.346 | 0.439 | 0.297 | 0.239 |
| corn borer | 0.166 | 0.169 | 0.172 | 0.236 | 0.284 | 0.247 | 0.288 | 0.255 |
| peach borer | 0.021 | 0.03 | 0.24 | 0.26 | 0.377 | 0.295 | 0.322 | 0.34 |
| penthaleus major | 0.285 | 0.466 | 0.722 | 0.828 | 0.764 | 0.82 | 0.739 | 0.755 |
| wheat phloeothrips | 0.124 | 0.161 | 0.317 | 0.404 | 0.361 | 0.318 | 0.48 | 0.36 |
| meadow moth | 0.173 | 0.201 | 0.138 | 0.17 | 0.202 | 0.252 | 0.6 | 0.503 |
| beet weevil | 0.36 | 0.317 | 0.221 | 0.299 | 0.255 | 0.278 | 0.344 | 0.313 |
| alfalfa plant bug | 0.165 | 0.287 | 0.196 | 0.286 | 0.295 | 0.21 | 0.257 | 0.252 |
| odonthrips loti | 0.409 | 0.326 | 0.608 | 0.558 | 0.563 | 0.56 | 0.621 | 0.605 |
| ceroplastes rubens | 0.002 | 0.116 | 0.004 | 0.111 | 0.045 | 0.132 | 0.051 | 0.158 |
| aleurocanthus spiniferus | 0.013 | 0.042 | 0.017 | 0.129 | 0.164 | 0.31 | 0.202 | 0.499 |
| deporausmarginatusPascoe | 0.007 | 0.01 | 0.233 | 0.26 | 0.26 | 0.379 | 0.267 | 0.408 |
| mAP | 0.1907 | 0.2184 | 0.2734 | 0.3298 | 0.3192 | 0.3657 | 0.4073 | 0.4362 |

表 4-2 是不同 shot 情况下，各个害虫类别在原始的 TFA 和改进后 TFA 的检测结果。从表中可以看出，改进后的 TFA 网络在不同的 shot 情况下 mAP 均有提升。具体在 1shot、3shot、5shot、10shot 时分别在 14 个、18 个、14 个、10 个害虫类别上检测性能有所提升。其中，1shot 条件下提升最大的害虫类别为 penthaleus major，提升了 18.1 个百分点，3shot 条件下提升最大的害虫类别为 rice leaf roller，提升了 19.8 个百分点，5shot 条件下提升最大的害虫类别为 white backed plant，提升了 14.8 个百分点，

10shot 条件下提升最大的害虫类别为 *Aleurocanthus spiniferus*, 提升了 29.7 个百分点。



(a) 原图



(b) TFA 预测框



(c) 本模型预测

图 4-7 检测结果图

Fig.4-7 Detection result graph

从害虫图像的特点可见, 它和自然图像明显不同的是类内具有多样性和类间具有相似性。这个特点会干扰到害虫检测头的正确分类, 从而产生误检的情况。图 4-7 展示了 TFA 网络和本章网络模型对新类害虫的检测情况, 其中, 由前 4 列图像可见, 本网络模型相对于 TFA 网络减少了新类害虫误检情况的发生, 后两列图像表明本网络模型相比于 TFA 网络降低了新类害虫漏检的情况。第一列害虫是金针虫成虫的检测情况, 该图像中害虫与背景颜色相近, 较难检测, TFA 网络错误地将该虫识别为苜蓿叶象甲, 而本网络模型则成功检测。TFA 网络将第二列的跳甲害虫错误识别为相似的东方绢金龟, 而本网络模型也能较好的检测出跳甲害虫。第三列是甜菜象虫的检测情况, TFA 网络在置信度大于 0.3 的情况下, 识别出置信度为 0.42 的东方绢金龟和置信度为 0.37 的甜菜象虫两个害虫类别, 而本网络则成功检测出甜菜象虫这一类别。第四列为牧草盲蝽的识别结果, TFA 网络错误将器识别为蝗虫, 本网络在成功检测的同时也更加精准地检测出害虫的位置。第五列和第六列为苜蓿夜蛾和金针虫的检测情况, TFA 网络对这种与背景相似的害虫发生了漏检, 未能将其识别出来, 而本网络模型能够避免这种害虫的漏检。

4.6.3 消融实验

为了验证本章提出改进方案的有效性。以 TFA 为基线程序(Baseline), 依次加入多尺度特征融合方法、负边距余弦损失函数和 CIOU Loss 函数改进到 TFA 基线方法中, 比较不同改进与基线方法的 $mAP@0.5$ 指标值, 实验结果如表 4-2 所示, 其中 \checkmark 表示使用该方法。

表 4-3 实验结果表

Table 4-3 Experimental result table

| | | | | | |
|--------|----------------------------|-------|-------|-------|--------------|
| Method | Baseline(TFA) | √ | √ | √ | √ |
| | Multi-scale Feature Fusion | | √ | √ | √ |
| | Neg-Mar Cos-Softmax Loss | | | √ | √ |
| | CIoU Loss | | | | √ |
| Shot | 1 | 19.07 | 19.81 | 21.65 | 21.84 |
| | 3 | 27.34 | 29.71 | 31.75 | 32.98 |
| | 5 | 31.93 | 35.25 | 36.36 | 36.57 |
| | 10 | 40.75 | 43.00 | 43.17 | 43.62 |

由表 4-3 可知，本章测试集在原始的 TFA 基线方法上 1shot、3shot、5shot、10shot 的 mAP 值分别为 19.07%、27.34%、31.92%、40.75%。在模型中引入多尺度特征融合后，模型在 1shot、3shot、5shot、10shot 上的检测精度都有提高，在 5shot 上提升最明显，有 3.32 个百分点，即使在 1shot 上，也有 0.74 个百分点的提升。说明多尺度特征融合能够充分地利用少量样本图像信息，有效提高小样本害虫的检测性能，但是在较多害虫图像时表现更佳。在使用负边距余弦损失后，模型在 1shot、3shot、5shot、10shot 分别提高了 1.84%、2.04%、1.11%、0.17%，可见负边距余弦损失在样本量足够小时，能够将新类从基类的特征空间中分离出来，可以提高新类害虫检测性能，该改进点在 1shot 和 3shot 上表现最优，说明样本量较少的时候该改进点提升更明显。在本模型使用了 CIoU Loss 后，使得害虫预测框的检测更加精准，网络模型的精度也有一定提升，在 3shot 的情况下表现最好，相对于 TFA 基线网络有 1.23 个百分点的提升。

4.6.4 不同模型的对比

为了更好地验证本章的算法性能，在 TFA w/fc、TFA w/cos、FSCE 和 MPSR 网络模型上测试了本章的数据集，得到结果如表 4-4 所见。

表 4-4 不同模型检测结果对比

Table 4-4 Comparison of detection results of different models

| | | | | | |
|-----------|----------|--------------|--------------|--------------|--------------|
| model | Backbone | 1 | 3 | 5 | 10 |
| MPSR | FRCN-101 | 10.92 | 18.32 | 23.20 | 31.69 |
| FSCE | FRCN-50 | 11.12 | 22.70 | 26.28 | 33.29 |
| TFA w/fc | FRCN-50 | 16.14 | 23.09 | 24.14 | 30.00 |
| TFA w/cos | FRCN-50 | 19.07 | 27.34 | 31.93 | 40.75 |
| 本模型 | FRCN-50 | 21.84 | 32.98 | 36.57 | 43.62 |

由表 4-3 可以得出, 本章模型在 1shot 情况下检测 20 类新类的平均精度比 MPSR 提高了 10.92 个百分点, 比 FSCE 提高 10.72 个百分点, 比 TFA w/fc 提高 5.7 个百分点, 比 TFA w/cos 提高 2.77 个百分点; 在 3shot 情况下检测 20 类新类的平均精度比 MPSR 提高 14.16 个百分点, 比 FSCE 提高 10.28 个百分点, 比 TFA w/fc 提高 9.89 个百分点, 比 TFA w/cos 提高 5.64 个百分点; 在 5shot 情况下检测 20 类新类的平均精度比 MPSR 提高 13.37 个百分点, 比 FSCE 提高 10.29 个百分点, 比 TFA w/fc 提高 12.43 个百分点, 比 TFA w/cos 提高 4.64 个百分点; 在 10shot 情况下检测 20 类新类的平均精度比 MPSR 提高 11.93 个百分点, 比 FSCE 提高 10.33 个百分点, 比 TFA w/fc 提高 13.62 个百分点, 比 TFA w/cos 提高 2.87 个百分点; 综上所述, 对比于其他的网络模型, 本章改进网络模型在 1shot、3shot、5shot、10shot 的情况下对包含幼虫和成虫的 20 类新类上的平均精度表现最优。

4.7 本章小结

针对基于深度学习的害虫检测需要大量图像的问题, 本章提出一种基于微调和多尺度特征融合的小样本害虫检测方法, 通过在 TFA 网络中添加多尺度特征融合模块用于充分挖掘少量样本图像中的特征信息, 针对害虫特有的类间相似, 类内不同的性质, 引入带负边距的余弦损失用于减少新类害虫与基类害虫之间的类别混淆, 同时优选了 CIoU Loss 作为预测模块的边界框损失函数, 这些共同组成了本章提出的新网络模型。在本章构建的包含幼虫和成虫在内的 80 类害虫数据集上进行实验, 当 IoU 阈值为 0.5 时, 模型在 1shot、3shot、5shot、10shot 上达到了 21.84%、32.98%、36.57%、43.62%。同改进前的 TFA 模型对比, 分别提高了 2.77%、5.64%、4.64%、2.87%。考虑到实际的农业应用场景, 大量的高质量害虫图像难以获得, 模型可以通过少量的新类图像完成对新类别害虫的检测。

第五章 害虫检测系统

我国土地资源丰富,农作物种类众多,其常遭受害虫的侵扰从而影响产量和质量。农民对于害虫的认识不够,不能准确识别害虫,而专业的农技工作者又比较少,当某地发生虫害后,农业专家不能迅速到达虫害发生的地点进行检测,不利于农作物的安全生产。为了能够帮助农民快速且准确的识别害虫,本章设计了一个害虫检测系统,通过广泛应用的微信小程序为入口,简单方便的为农民提供害虫检测和防治功能,对农作物的安全生产具有重大意义。

5.1 系统需求概述

在开发害虫检测系统之前,需要对该系统实现的功能进行需求分析,只有明确了系统的需求,开发出来的系统才能真正令用户满意。本系统的用户设定为农民,用户可以通过注册登录系统后进行使用。用户可以通过直接拍照或者手机文件上传害虫图像,调用第四章训练好的害虫检测模型对害虫图像进行检测,然后将检测结果以图像和文字的形式显示到用户页面上,用户可以对该次检测结果进行评价,在检测过程中,系统会同时将原始图像和检测结果图像保存到本地文件夹中。其次,用户还可以通过害虫知识库模块,查看某种农作物下的害虫信息,以及该害虫相关的防治措施。另外,本系统面向的是非专业人士,系统的响应时间需要尽可能的短,满足实时性的需要。最后,本系统界面应当简单美观,操作应当易于使用。

5.2 系统总体设计

5.2.1 系统软硬件平台

本系统实验配置环境如表 5-1 所示,分为服务器端和客户端,服务器端的操作环境为 Windows 10,客户端为移动端的微信小程序。

表 5-1 系统配置环境

| Table 5-1 System configuration environment | |
|--|---------------------|
| 实验环境 | 配置参数 |
| 微信版本 | V8.0.33 |
| 操作系统 | Microft Windows 10 |
| 处理器 | Intel core i5-4210M |
| 显卡 | GeForce 840M |
| 编程语言 | Python |

客户端开发使用微信开发者工具，通过 AppID 新建一个害虫检测项目，如图 5-1 为微信开发者工具页面，左侧为预览窗口，中间是项目目录，右侧为代码编辑区域，开启模拟器选项，可以在左侧的预览窗口看到实时的项目页面。

本系统采用 Flask 框架开发，Flask 是使用 python 为基础语言创造出来的 web 开发框架。该框架没有对数据库的约束层和任何种类的验证和安全，开发者可以完成按照自己的需求充分灵活的实现功能的开发，可以在短时间内搭建一个网络应用。本系统在检测程序上将 Flask 和 Pytorch 框架结合，进行核心检测功能的实现，对于其他的功能需求，采用 MySQL 数据库进行数据的读取和输入，使用 Flask web 开发技术进行实现。

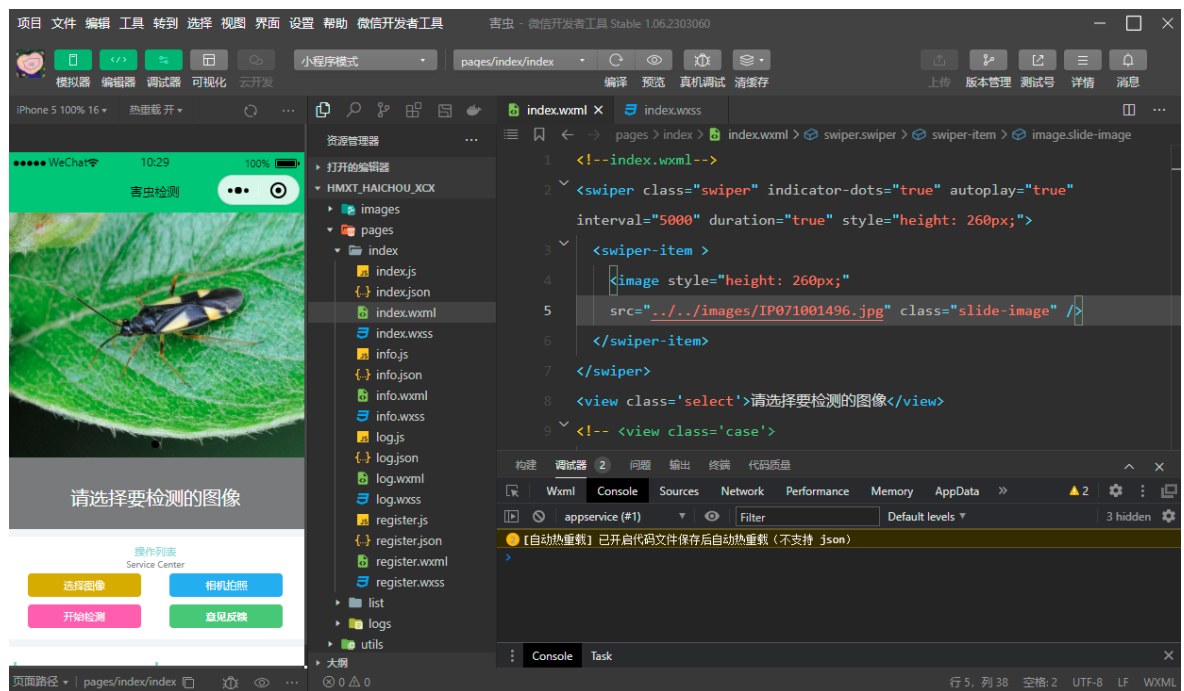


图 5-1 微信开发工具界面

Fig. 5-1 The interface of wechat development tool

5.2.2 框架设计

系统框架如图 5-2 所示，整个系统分为后台服务器和前台微信小程序两部分。WXML 用于构建微信小程序的页面结构，WXSS 用于微信小程序页面的显示样式，JSON 为小程序的配置文件。系统使用 JS 文件进行前台的逻辑处理和前后台的数据交互，小程序页面间的跳转使用 JS 函数完成，然后向后台发送 POST 或者 GET 请求，后台服务端接收到不同的前台请求，通过选择小样本害虫检测网络进行害虫检测，或者选择其他与数据库进行交互的功能。后台逻辑处理完成之后，以 JSON 格式对处理结果值进行封装并返回给前台。在前台页面进行加载的过程中，通过提前编写的 JS 文件对数据进行解析并显示。

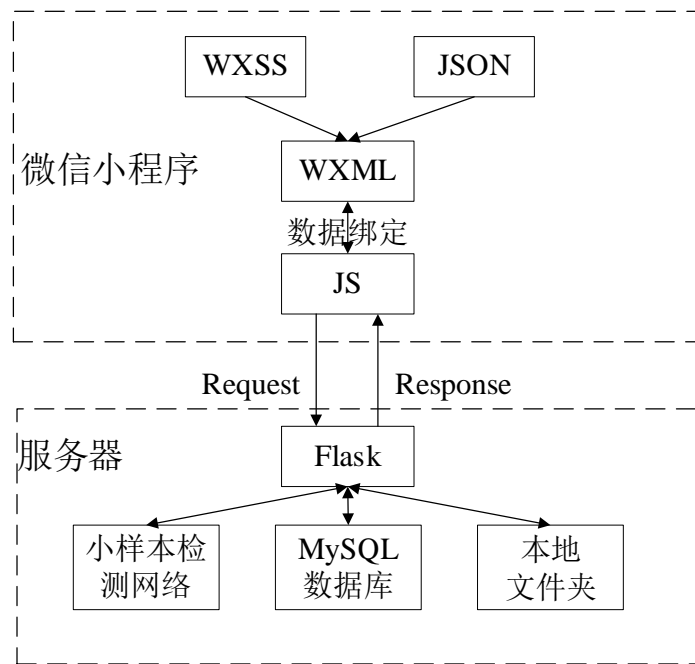


图 5-2 系统框架图

Fig. 5-2 System frame diagram

5.2.3 系统功能设计

当前设计的害虫检测系统为初始系统版本，功能还不够完善，仅仅完成了基本功能。具体功能如图 5-3 所示，主要包含三大功能模块，分别是个人信息管理模块、害虫检测功能、害虫知识库。

(1) 个人信息管理模块

注册和登录是一个软件系统所应有的基本功能，只有当用户完成系统注册和登录功能后，才能使用本系统进行害虫的检测、害虫信息的查询和害虫的防治措施等。另外，用户还可以对检测结果进行评价，将结果存储到数据库。

(2) 害虫检测功能

害虫检测是本系统的核心功能，用户可以通过拍照上传或者本地图片上传的方式选择一张想要检测的害虫图像，然后点击检测害虫，该张图像就会被发送到后台的检测程序，后台的检测程序处理完毕后，检测出害虫的位置和类别，检测结果会显示到用户输入图像的页面上。用户输入的原始图像和检测后的害虫图像都将保存到本地文件夹中。

(3) 害虫知识库

该模块有两个功能，分别是害虫检索和害虫防治知识。害虫检索可以通过农作物名称或者害虫名称检索害虫，进而了解到该类农作物可能会遭受到危害的害虫种类，害虫防治知识是有关该害虫的农业、物理、化学防治手段的介绍。

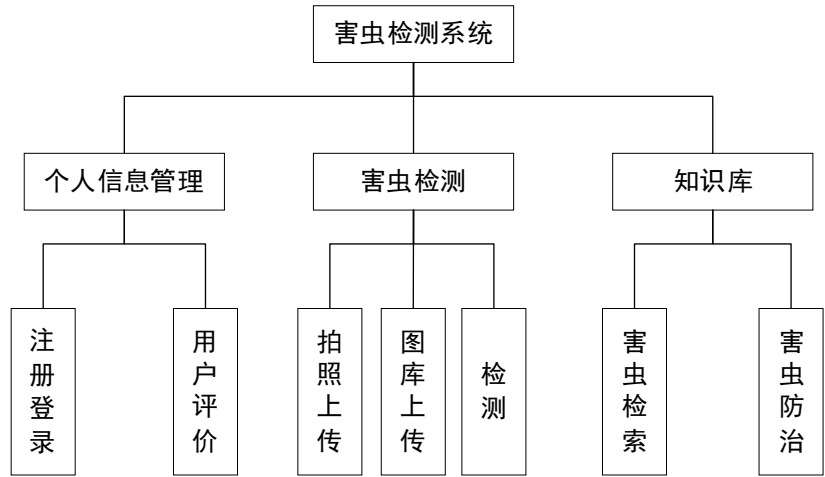


图 5-3 系统功能图

Fig. 5-3 System operation graph

5.2.4 系统检测流程

害虫检测流程为用户通过手机拍照或者文件系统上传待检测的图像，选择好的图像会显示在页面上。用户点击开始检测的按钮后，害虫图像将传输到后台服务端，后台检测模型将对图像进行检测，将检测结果数据返回到微信小程序端进行展示，检测图像将覆盖原有图像，同时在下方显示出检测文字。检测流程如图 5-4 所示。

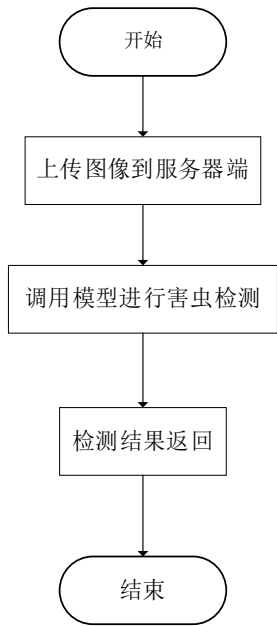


图 5-4 系统检测流程图

Fig. 5-4 System detection flow chart

5.3 系统展示

(1) 用户注册和登录功能

本系统的注册和登录页面如图 5-5 所示，用户在注册本系统时输入账号名和密码

即可，这里密码只设置一次输入，后期可以很方便地修改密码，用户不需要在密码注册时进行两次校验。用户在注册账号成功后，即可以通过账号和密码登录本系统。

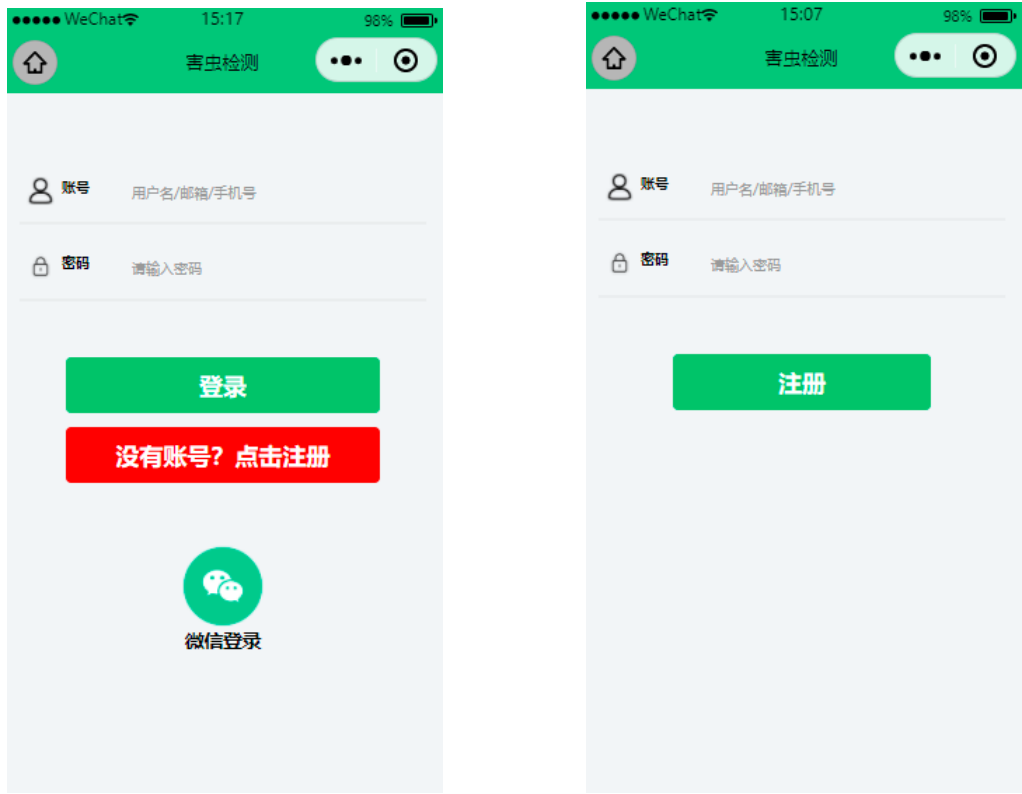


图 5-5 注册登录页面

Fig. 5-5 Registration login page

(2) 当用户登录系统后，进入的首页如图 5-6 所示，在页面的上方展示了用户要检测的图像或者是检测完成的结果。当没有害虫图像的检测结果时，图像下方会显示“害虫检测”。当检测结果从服务器返回数据后，该区域会显示检测的结果。再下方有四个按钮，分别是“选择图像”、“相机拍照”、“开始检测”、“意见反馈”，用户可以通过点击“选择图像”从用户手机本地文件夹中选择要检测的图像，也可以选择“相机拍照”直接拍摄一张现场的害虫图像进行检测，当点击“开始检测”后就进行到检测流程中，通过服务器端的检测模型对害虫进行检测。在检测过程中，系统会同时将原始图像和检测结果图像保存到本地文件夹中。如果用户对检测结果有异议时，可以进行相应的意见反馈。

为了对本系统的总检测流程用时进行估算，在本地 GPU 设备（GeForce 840M）条件下，本系统在 IP102 测试集中随机选取 10 张图像进行测试，得到了从拍照上传到识别结果下发整个流程的用时，分别如下：7726ms、4172ms、5354ms、3805ms、7973ms、4521ms、4334ms、3698ms、3695ms、4083ms，得到的时间平均值为 4936.1ms。在本地设备条件下，整个流程所用的时间已经基本满足日常所需，随着硬件的升级，整个检测流程所用时间会更加少。

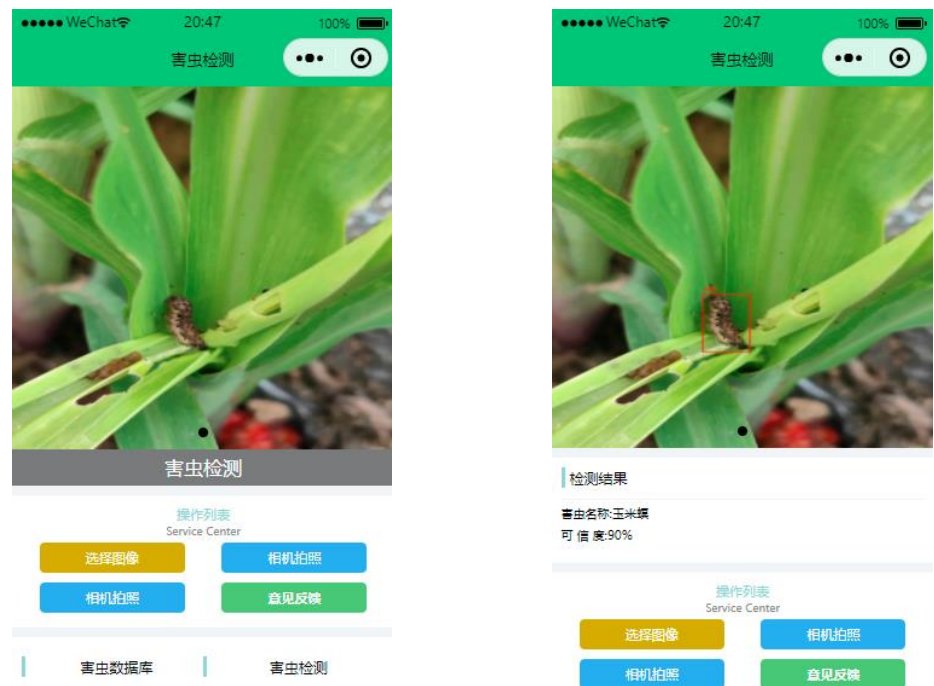


图 5-6 害虫检测页面

Fig. 5-6 Pest detection page

(3) 害虫知识库

害虫知识库用于查看害虫的特性和害虫的防治手段。当用户选择害虫知识库后，可以查看不同农作物的害虫信息。当用户点击该类害虫的信息时会跳转到害虫的防治页面，这个页面会显示该种害虫的基本信息和相应的农业防治、化学防治和生物防治等防治手段。

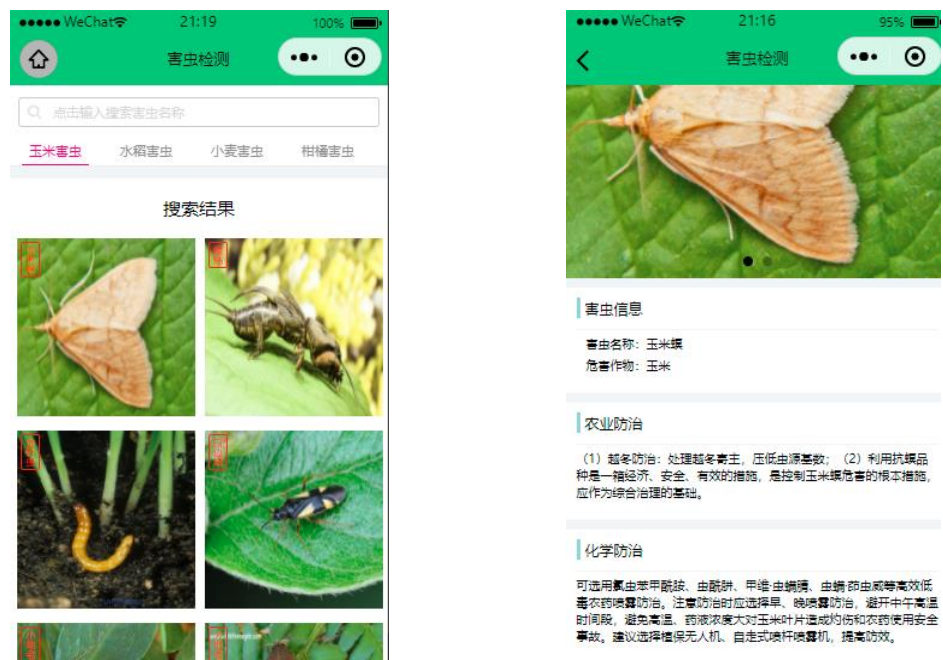


图 5-7 害虫知识库页面

Fig. 5-7 Pest knowledge base page

5.4 本章小结

本章在前文害虫检测理论的基础上，为农民设计并实现了一款害虫检测微信小程序，成功的将理论运用到实际生产中。在本章，首先进行了需求分析，接着根据需求分析对系统进行了总体设计，然后对本系统的功能进行了展示。当前害虫检测系统已经能够成功完成害虫的检测功能，对农业的安全生产具有非常重要的意义。

第六章 总结与展望

6.1 工作总结

农业作为我国重要的基础产业，不仅可以解决粮食短缺问题，还能提高农民的收入水平。但是，农作物生产经常受到大量害虫的侵袭，这无疑阻碍了农作物质量和产量的高速发展。目前，主要采用化学农药来治理害虫，虽然在一定程度上减少了害虫的危害，但不可避免地会对地下水、土壤等生态造成破坏。农民滥用农药的主要原因是他们不能准确地识别害虫种类，对害虫的防治措施缺乏足够的认识。因此，准确的害虫检测对农业的可持续发展具有非常重要的意义。同时，害虫具有区域性和季节性的特点，种类繁多，难以获得每一类害虫的大量图像。在小样本条件下的害虫检测是一个亟待解决的问题。本文在大样本条件和小样本条件分别提出了基于深度学习的害虫检测方法，具体研究内容如下：

(1) 使用一系列的改进策略用于提高玉米害虫的检测精度。首先，通过几何变化和内容变化的在线数据增强提高输入数据的多样性。其次，针对玉米害虫数据集，使用结合遗传算法的 K-means 聚类算法得到了适应玉米害虫数据集的锚框比例。然后，将 Faster R-CNN 网络的颈部替换为递归特征金字塔网络，用于更好地检测自然环境中的大尺度害虫。最后，采用改进的 Soft-NMS 算法减少玉米害虫的漏检率。本文基于 Faster R-CNN 改进的网络在玉米害虫数据集的 mAP0 达到了 79.05%。对比原始的 Faster R-CNN 提高了 3.49 个百分点。

(2) 提出一种基于微调和多尺度特征融合的小样本害虫检测方法。针对害虫具有不同的生命阶段，每个阶段的害虫外观差异很大，将害虫的幼虫和成虫阶段作为不同的类别分别处理。使用三个尺度不同的 RoI 模块间的融合来提高模型在小样本情况下对不同尺度害虫的检测，并且利用整张图像的特征信息来增强害虫的类别信息。针对小样本害虫检测中容易产生的误检测，引入基于度量学习的负边际损失，解决了基类和新类容易混淆的问题。针对原始 Faster R-CNN 使用的 Smooth L1 损失函数在边界框的回归预测中不够精准，使用 CIoU Loss 进一步提高了预测框的检测精度。在采用的害虫数据集上，提出的方法在 1shot、3shot、5shot 和 10shot 时对新类害虫的检测 mAP 分别为 21.8%、32.3%、36.57%和 43.62%，相比其他前沿的小样本目标检测网络，检测精度都是最高的。

(3) 设计并实现了害虫检测的微信小程序。利用害虫检测小程序，用户只需要上传害虫的照片，就可以获得害虫的预测结果，大大提高了害虫检测的准确率和效率。同时，还将害虫知识库整合到小程序中，用户可以根据害虫名称或某种农作物对应的害虫查询相关信息，并了解相应的防护措施。

本文基于深度学习提出了两种害虫检测网络，一种是大样本条件下的玉米害虫检测，一种是小样本条件下的多种农作物害虫检测。在大样本玉米害虫检测下，检测精度较高，能够满足实际的生产使用需求。在小样本害虫检测条件下，也可以达到不错的检测性能，能够通过少量的图像实现对新害虫类别的检测，为农作物上众多种类的害虫检测提供了新的思路，探索了新的方法。

6.2 研究展望

本文对大样本条件下的害虫检测和小样本条件下的害虫检测都提出了新的检测改进网络，但是仍然存在一些问题，可以从以下几个方面进一步完善和发展：

本文已经提出一种小样本条件下的害虫检测网络，但是该网络在害虫的检测精度上仍然需要进一步提升，在后续的研究中，可以进一步提出新的网络模型或者使用更加有效的优化策略，提高网络在新类害虫的检测准确率，同时也要保证原有基类害虫的检测精度，保证小样本害虫检测也能用于实际的生产中。

由于小样本条件下只能依靠少量几张图像进行训练，所以图像数据集的质量至关重要。IP102 害虫数据集主要来源于网络下载，虽然经过了农业专家的筛选，但质量仍然参差不齐，建立一个大的、高质量的自然环境下害虫数据集是很有必要的一件事，这样研究得到鲁棒性更强的小样本目标检测模型。

近年来，增量小样本目标检测也被提出，现有小样本目标检测方法更加注重新类的检测性能，而增量小样本目标检测则同时关注基类和新类上的检测性能。同时，在复杂的现实环境中，当发现新的类别需要检测时，可能存在不能够获取原有基类数据的情况，增量小样本目标检测在训练新类害虫时，不再需要基类的图像，能够将基类的害虫知识保持住并能够转移到新类别。

参考文献

- [1] 国家统计局. 国家统计局关于 2022 年粮食产量数据的公告 [EB/OL]. (2022-12-12). http://www.lswz.gov.cn/html/xinwen/2022-12/12/content_273203.shtml.
- [2] 农业农村部办公厅. 2023 年“虫口夺粮”保丰收行动方案 [EB/OL]. (2023-03-03). http://www.moa.gov.cn/govpublic/ZZYGLS/202303/t20230307_6422499.htm.
- [3] 陆宴辉, 赵紫华, 蔡晓明, 等. 我国农业害虫综合防治研究进展[J]. 应用昆虫学报, 2017, 54(3): 349-363.
- [4] Wu X, Zhan C, Lai Y K, et al. Ip102: A large-scale benchmark dataset for insect pest recognition[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 8787-8796.
- [5] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.
- [6] Platt J. Sequential minimal optimization: A fast algorithm for training support vector machines [J]. Microsoft Research Technical Report, 1998, 98.
- [7] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(9): 1904-1916.
- [8] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
- [9] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
- [10] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.
- [11] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
- [12] Law H, Deng J. Cornernet: Detecting objects as paired keypoints[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 734-750.
- [13] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
- [14] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.

- [15] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.
- [16] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.
- [17] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.
- [18] Wang R, Jiao L, Xie C, et al. S-RPN: Sampling-balanced region proposal network for small crop pest detection[J]. Computers and Electronics in Agriculture, 2021, 187: 106290.
- [19] Shen Y, Zhou H, Li J, et al. Detection of stored-grain insects using deep learning[J]. Computers and Electronics in Agriculture, 2018, 145: 319-325.
- [20] 姚青, 姚波, 吕军, 等. 基于双线性注意力网络的农业灯诱害虫细粒度图像识别研究[J]. 中国农业科学, 2021, 54(21): 4562-4572.
- [21] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.
- [22] Li W, Wang D, Li M, et al. Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse[J]. Computers and Electronics in Agriculture, 2021, 183: 106048.
- [23] 梁勇, 邱荣洲, 李志鹏, 等. 基于深度学习的水稻主要害虫识别方法[J/OL]. 农业机械学报: 1-13[2023-0330].<http://kns.cnki.net/kcms/detail/11.1964.S.20220519.0919.002.htm>.
- [24] Zhang W, Huang H, Sun Y, et al. AgriPest-YOLO: A rapid light-trap agricultural pest detection method based on deep learning[J]. Frontiers in Plant Science, 2022, 13.
- [25] 李子茂, 刘恋冬, 夏梦, 等. 基于深度学习的月季多叶片病虫害检测研究[J]. 中国农机化学报, 2021, 42(8): 169-176.
- [26] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.
- [27] Liu L, Wang R, Xie C, et al. A global activated feature pyramid network for tiny pest detection in the wild[J]. Machine Vision and Applications, 2022, 33(5): 76.
- [28] 陈娟, 陈良勇, 王生生, 等. 基于改进残差网络的园林害虫图像识别[J]. 农业机械学报, 2019, 50(5): 187-195.
- [29] Li R, Wang R, Xie C, et al. A coarse-to-fine network for aphid recognition and detection in the field[J]. Biosystems Engineering, 2019, 187: 39-52.
- [30] Liu L, Xie C, Wang R, et al. Deep learning based automatic multiclass wild pest monitoring approach using hybrid global and local activated features[J]. IEEE Transactions on Industrial Informatics, 2020, 17(11): 7589-7598.

- [31] Cao Y, Wang J, Jin Y, et al. Few-Shot Object Detection via Association and DIscrimination[J]. Advances in Neural Information Processing Systems, 2021, 34.
- [32] Wang X, Huang T E, Darrell T, et al. Frustratingly simple few-shot object detection[J]. arXiv:2003.06957, 2020.
- [33] Qiao L, Zhao Y, Li Z, et al. DeFRCN: Decoupled Faster R-CNN for Few-Shot Object Detection[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, Oct 11-17,2021. Piscataway: IEEE, 2021: 8681-8690.
- [34] Wu A, Han Y, Zhu L, et al. Universal-prototype augmentation for few-shot object detection[J]. arXiv:2103.01077, 2021.
- [35] Li B, Yang B, Liu C, et al. Beyond max-margin: Class margin equilibrium for few-shot object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, Jun 19-25,2021. Piscataway: IEEE, 2021: 7363-7372.
- [36] Li Y, Feng W, Lyu S, et al. MM-FSOD: Meta and metric integrated few-shot object detection[J]. arXiv:2012.15159, 2020.
- [37] Karlinsky L, Shtok J, Harary S, et al. Repmet: Representative-based metric learning for classification and few-shot object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach,CA, Jun 16-20,2019. Piscataway: IEEE, 2019: 5197-5206.
- [38] Kim G, Jung H G, Lee S W. Spatial reasoning for few-shot object detection[J]. Pattern Recognition, 2021, 120: 108118.
- [39] Zhu C, Chen F, Ahmed U, et al. Semantic relation reasoning for shot-stable few-shot object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2021: 8782-8791.
- [40] KIM G, JUNG H-G, LEE S-W. Few-shot object detection via knowledge transfer[C]//2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Piscataway: IEEE, 2020: 3564-3569.
- [41] LIU W, LI H, YU S, et al. Dynamic Relevance Learning for Few-Shot Object Detection[J]. arXiv:2108.02235, 2021.
- [42] HAN G, HE Y, HUANG S, et al. Query adaptive few-shot object detection with heterogeneous graph convolutional networks[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, Oct 11-17,2021. Piscataway: IEEE, 2021: 3263-3272.
- [43] WU J, LIU S, HUANG D, et al. Multi-scale positive sample refinement for few-shot object detection[C]// European conference on computer vision. Cham: Springer, 2020: 456-472.
- [44] Zhang W, Wang Y X. Hallucination improves few-shot object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Tennessee, Jun 19-25, 2021.

- Piscataway: IEEE, 2021: 13008-13017.
- [45] Kaul P, Xie W, Zisserman A. Label, verify, correct: a simple few shot object detection method[C]//Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, Jun 19- 24, 2022. Piscataway: IEEE, 2022: 14237-14247.
- [46] Xiao Y, Marlet R. Few-shot object detection and viewpoint estimation for objects in the wild[C]//European conference on computer vision. Cham: Springer, 2020: 192-210.
- [47] Fan Q, Zhuo W, Tang C K, et al. Few-shot object detection with attention-RPN and multi-relation detector[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, Jun 13-19. Piscataway: IEEE, 2020: 4013-4022.
- [48] Li A, Li Z. Transformation invariant few-shot object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Tennessee, Jun 19-25, 2021. Piscataway: IEEE, 2021: 3094-3102.
- [49] Zhang W, Wang Y X, Forsyth D A. Cooperating RPN's Improve Few-Shot Object Detection[J]. arXiv:2011.10142, 2020.
- [50] Wu A, Han Y, Zhu L, et al. Universal-prototype augmentation for few-shot object detection[J]. arXiv:2103.01077, 2021.
- [51] Yan X, Chen Z, Xu A, et al. Meta r-cnn: Towards general solver for instance-level low-shot learning[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Oct 27-Nov 2. Piscataway: IEEE, 2019: 9577-9586.
- [52] Agarwal A, Majee A, Subramanian A, et al. Attention Guided Cosine Margin To Overcome Class-Imbalance in Few-Shot Road Object Detection[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Hawaii, Jan 4-8, 2022. Piscataway: IEEE, 2022: 221-230.
- [53] Hu H, Bai S, Li A, et al. Dense relation distillation with context-aware aggregation for few-shot object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2021: 10185-10194.
- [54] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [55] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
- [56] Kang B, Liu Z, Wang X, et al. Few-shot object detection via feature reweighting[C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Oct 27-Nov 2, 2019. Piscataway: IEEE, 2019: 8420-8429.
- [57] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

- [58] Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(7): 12993-13000.
- [59] Liu W, Wen Y, Yu Z, et al. Large-margin softmax loss for convolutional neural networks[J]. arXiv preprint arXiv:1612.02295, 2016.
- [60] Wang H, Wang Y, Zhou Z, et al. Cosface: Large margin cosine loss for deep face recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 5265-5274.
- [61] Liu B, Cao Y, Lin Y, et al. Negative margin matters: Understanding margin in few-shot classification[C]//Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16. Springer International Publishing, 2020: 438-455.