电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

专业学位硕士学位论文

MASTER THESIS FOR PROFESSIONAL DEGREE

论文题目 **基于深度学习的辣椒识别及采摘技术研究**

专业学位类别	电子信息
学　　号	**202222050706**
作者姓名	周志华
指导教师	高椿明　　教　授
学　　院	光电科学与工程学院

分类号 TP311.1 密级 公开

UDC [注1] 004.8

学 位 论 文

基于深度学习的辣椒识别及采摘技术研究

（题名和副题名）

周志华

（作者姓名）

指导教师 高椿明 教 授

电子科技大学 成 都

（姓名、职称、单位名称）

申请学位级别 硕士 专业学位类别 电子信息

提交论文日期 2025 年 3 月 20 日 论文答辩日期 2025 年 5 月 21 日

学位授予单位和日期 电子科技大学 2025 年 6 月

答辩委员会主席 邱琪

评阅人 YJ2506050211_1、YJ2506050211_2

注 1：注明《国际十进分类法 UDC》的类号。

Research On Chili Pepper Identification and Harvesting Technology Based on Deep Learning

A Master Thesis Submitted to
University of Electronic Science and Technology of China

Discipline **Electronic Information**

Student ID **202222050706**

Author **Zhihua Zhou**

Supervisor **Prof. Chunming Gao**

School **School of Optoelectronic Science and Engineering**

摘　要

辣椒作为我国种植面积最大的蔬菜作物之一，具有极高的经济效益，目前辣椒采摘仍然依赖传统人工采摘，人工采摘工作效率低，劳动强度大，并且随着城镇化进程的加快，农业劳动力持续减少，人工采摘的成本不断上升，亟需对辣椒的自动化采摘开展研究。

本文以田间辣椒为研究对象，基于深度学习和双目视觉定位对辣椒识别和采摘技术开展了深入研究，主要研究内容如下：

1. 提出一种改进 YOLOv8s 的辣椒识别算法。首先采集辣椒图像并进行标注，构建了高质量的辣椒数据，接着针对辣椒识别任务中的枝叶遮挡和密集重叠的问题，在 YOLOv8s 模型的 Neck 部位，加入 SEAM 注意力机制，提升模型对遮挡辣椒的检测能力；使用 Varifocal 损失函数实现更高效的样本加权机制，让模型更关注高质量的正样本；在后处理中使用 Soft-NMS 保留更多高质量的预测框，减少辣椒因密集重叠漏检而漏检的概率。实验结果表明，本文改进后的模型 mAP 相比原模型提高了 4.4%，在密集遮挡辣椒检测任务中具有较高的检测精度。

2. 基于 RealSense D415 深度相机建立了辣椒定位系统。首先通过张友正标定法对 D415 进行标定，计算相机内参。并将相机固定在机械臂一侧进行手在眼外标定，计算出手眼矩阵。进行定位实验，综合平均误差是 3.97 mm，满足采摘需求，最后计算辣椒偏转角度，辅助末端执行器更好的完成采摘。

3. 辣椒采摘系统搭建和采摘实验。搭建辣椒采摘系统和设计软件操作界面，在模拟辣椒采摘场景中进行采摘实验，最终的辣椒识别率达到 91.4%，采摘成功率达到 81.3%，本文提出的辣椒采摘系统可以准确识别和采摘辣椒。

关键词：辣椒，采摘，目标检测，双目视觉

ABSTRACT

Chili pepper, as one of the largest vegetable crops in China, has extremely high economic benefits. Currently, chili pepper harvesting still relies on traditional manual harvesting, which has low efficiency and high labor intensity. With the acceleration of urbanization, agricultural labor continues to decrease, and the cost of manual harvesting continues to rise. Therefore, there is an urgent need to conduct research on automated chili pepper harvesting.

This thesis takes chili peppers in the field as the research object, and conducts in-depth research on chili pepper recognition and harvesting technology based on deep learning and binocular vision positioning. The main research contents are as follows:

1. Propose an improved YOLOv8s chili pepper recognition algorithm. Firstly, chili pepper images were collected and annotated to construct high-quality pepper data. Then, aiming at the issues of branch and leaf occlusion and dense overlap in chili pepper recognition tasks, the YOLOv8s model was improved by incorporating the SEAM attention mechanism to enhance the model's ability to detect occluded chili pepper; Using Varifocal loss function to implement a more efficient sample weighting mechanism, allowing the model to focus more on high-quality positive samples; Use Soft NMS in post-processing to retain more high-quality prediction boxes and reduce the probability of missed detections of chili pepper due to dense overlap. The experimental results show that the improved model in this thesis has a 4.4% increase in mAP compared to the original model, and has high detection accuracy in dense occlusion chili pepper detection tasks.

2. A chili pepper localization system was established based on the RealSense D415 depth camera. Firstly, calibrate the camera using the Zhang Youzheng calibration method to obtain its internal parameters. Next, fix the camera on one side of the robotic arm for hand eye extrinsic calibration and calculate the hand eye matrix. Conduct positioning experiments with a comprehensive average error of 3.97 mm, which meets the harvesting requirements. Finally, calculate the chili pepper deflection angle to assist the end effector in better harvesting.

3. Construction and harvesting experiment of chili pepper harvesting system. Building a chili pepper harvesting system and designing software operation interface,

conducting harvesting experiments in simulated chili pepper harvesting scenarios, the final chili pepper recognition rate reached 91.4%, and the harvesting success rate reached 81.3%. The chili pepper harvesting system proposed in this thesis can accurately identify and harvest chili peppers.

Keywords: Chili Pepper, Harvesting, Object Detection, Binocular Vision

目　录

第一章 绪论

1.1 研究背景及意义

辣椒是一年生或有限多年生的茄科辣椒属植物，其营养价值丰富，包含多种维生素，辣椒素和膳食纤维等营养成分，其中维生素 C 的含量尤其高，维生素 C 可以有效增强免疫力，抗氧化以及保护心血管健康，辣椒素是辣椒辛辣的来源，具有抗炎，抗氧化，刺激唾液和胃液分泌，增强食欲，改善消化不良的功能，膳食纤维能促进肠道健康，帮助控制血糖和血脂[1]。辣椒的土壤适应性很强，耐旱力，耐瘠力较强。辣椒独特的辛辣口感，丰富的营养价值以及较低的生长环境需求，让辣椒在全国各地都有种植，其中在贵州，云南等多省的辣椒栽种面积超过 200 万亩。中国也是世界上最大的辣椒种植国，图 1-1 和图 1-2 是我国 2012 年至 2023 年辣椒种植面积和产量变化趋势图，近年来我国辣椒种植面积稳定在 1100 万亩以上，产量在 1600 万吨以上。辣椒除了直接食用之外，还可以制作成辣椒油，辣椒干，火锅底料等多种辣椒加工品，截止 2024 年 7 月，我国大约有 1.15 万家辣椒加工相关产业。2023 年整体辣椒产业市场规模达到 4000 亿元，具有极高的经济效益和社会效益[2]。

目前，辣椒采摘主要依靠人工采摘，而随着辣椒产业规模的不断扩大，同时农村的青壮年劳动力不断流失，走向城市，传统人工采摘人力严重不足，尤其是在辣椒的集中采摘期，劳动力成本更是急剧上升。而且人工采摘效率低，长时间的弯腰采摘也极易导致健康问题，这些问题使得辣椒采摘的成本占据了整个产业的 30%，也导致辣椒产业难以进一步发展，因此急需对自动化辣椒采摘开展研究，提高采摘效率，减少人工依赖。

在 20 世纪 80 年代，国外发达国家如美国，日本，荷兰等便开展对采摘机器人的研究，很多采摘机器人已经投入到实际使用当中。虽然我国是世界第一农业大国，但在采摘机器人领域的研究起步较晚，且目前大多处于实验室阶段，但随着国家发展战略的重视，《中国制造 2025》中将农机装备列入十大重大领域当中，提高农机装备信息收集、智能决策和精准作业能力[3]，近年来，我国的采摘机器人发展迅猛，与国外差距不断缩小。

辣椒自动化智能采摘存在的主要问题有：辣椒生长往往都堆叠到一起，极易受到其他辣椒及枝叶的遮挡，再加上光照，各种噪声的影响，使得准确识别和定位辣椒十分困难。同时遮挡重叠的辣椒也对机械臂和末端执行器的要求较高，本文对 YOLOv8s 模型进行改进和优化，并基于机械臂和深度相机，搭建辣椒采摘系统，按照辣椒图像采集、辣椒识别定位、机械臂运动采摘的步骤，完成辣椒的准确采摘，

实现辣椒采摘的智能化、机械化。这样一个高效准确的辣椒智能采摘系统对于推动我国辣椒产业的发展有重要的意义，之后更是可以由点到面，带动其他各类智能采摘系统和农业智能设备的发展，实现农机设备的精准化，智能化，信息化，推动我国农业的进一步发展。

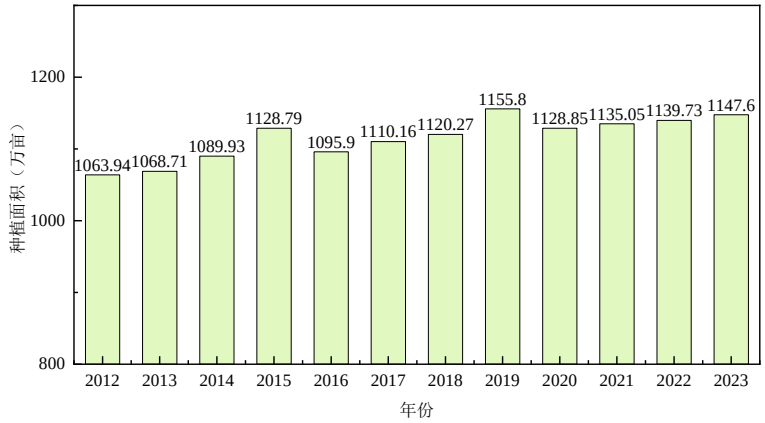


图 1-1 2012 至 2023 年中国辣椒种植面积变化趋势

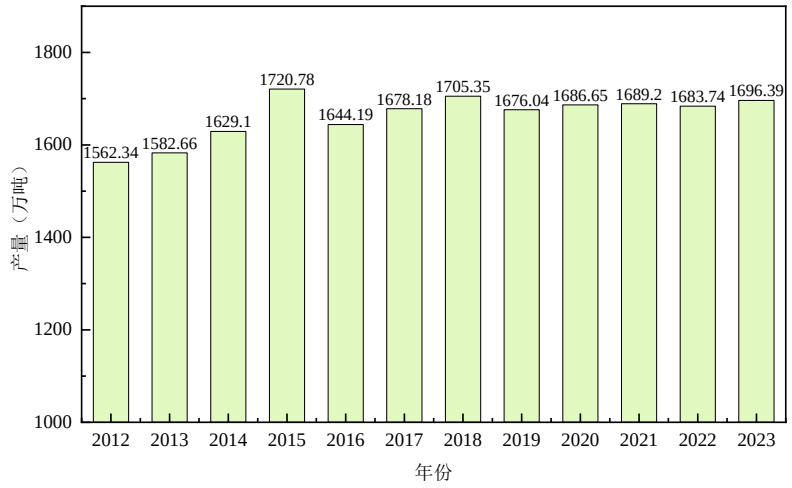


图 1-2 2012 至 2023 中国辣椒产量变化趋势

1.2 国内外研究现状

1.2.1 果实识别研究现状

果实识别是从图像采集系统拍摄的图片中，将目标果实与背景分离出来，是农业智能化的重要组成部分，果实能否精准识别，将直接影响后续果实的定位和采摘，目前用于果实识别的主要有两类方法，第一是基于传统图像处理的果实识别，第二是基于深度学习技术的果实识别。

1.2.1.1 基于传统图像处理的果实识别

早期的果实识别主要依赖基于形状，颜色等特征的传统图像处理。早在 1977 年，Parrish 等人[4]在相机镜头前安装红色滤波片，增强背景与苹果的对比度，同时减少照明和阴影的影响，再对拍摄的图像进行阈值处理，实现苹果的简单识别，该论文首次将图像处理用于果实识别。Hayashi 等人[5]根据茄子的形状和颜色特征进行茄子识别，首先根据茄子的颜色来区分茄子的果实和部分叶和茎，再依据茄子的长条形特征，将第一步分离的图像进行垂直分割操作，分离出茄子果实和部分叶，茎，最终实现茄子的识别。Kelman 等人[6]首先使用 Canny 边缘检测算法，提取图像中的所有边缘轮廓，然后通过凸性测试去除 Canny 滤波器最初识别的边缘，最后对剩余边缘进行分析即可检测出图像中的苹果。王晋[7]基于颜色特征识别苹果，先使用自动阈值处理苹果灰度图像，再还原分割后的图像颜色，并进行固定阈值分割，最终识别出苹果，算法平均识别率能达到 73%。Bairong Li[8]等人首先根据绿苹果与背景的相似性生成显著性图，然后采用高斯曲线拟合算法拟合 YUV 颜色空间中突出的 V 分量，再进行阈值分割检测绿苹果，最终检测率达到 91.84%。该论文解决绿果实与叶子，枝干等背景颜色相似的问题，但是在有阴影的图像中识别效果较差。赵德安[9]等人利用基于色差的 OTSU（大津法）技术先分割图像中的重叠苹果，再使用形态学方法对已分割的苹果进行孔洞填充和去噪，接着根据苹果的形状特征计算重叠苹果的直径和圆心，最后通过归一化匹配识别重叠苹果。

随着果实采摘需求的不断提高，基于阈值分割，边缘检测等传统图像处理难以满足果实识别的需要，有很多研究人员将机器学习中的一些算法用于果实识别领域中。

Wei Ji 等人[10]先通过矢量中值滤波器对苹果图像进行预处理，减少噪声，接着基于种子区域生长和颜色特征的阈值分割算法，提取彩色苹果图像的形状和颜色特征，最后再使用 SVM（支持向量机）对提取的特征进行分类，最终苹果检测的准确率达到 89%。C.W.Bac 等人[11]开发了一个多光谱系统，让甜椒不同的部位具有不同的波长，并将其作为特征送到分类器分类和回归树中学习，根据波长的不同可实现甜椒各部位的识别，但甜椒总体的识别率较低。张春龙等人[12]设计了一个混合分类器，以归一化的 g 分量和 HSV 颜色空间中 H、S 分量为特征参数的 SVM 分类器和以超绿算子(2G-R-B)为特征参数的阈值分类器，实现了近色背景中绿苹果的精准识别。王丹丹[13]先使用 K-means 聚类分割算法进行图像分割，再使用 Normalized Cut 谱聚类算法再次分割图像，接着提取苹果的外部轮廓，再使用 Spline 插值算法对因遮挡而轮廓不全的苹果重建。Chenglin Wang 等人[14]先使用小波变化

算法归一化处理荔枝图像中的表面照明，再使用 K-means 聚类算法实现荔枝的识别。

总的来说，基于形状，颜色等特征的传统图像处理能在相对简单的情况下，实现果实的精准识别，但果实图像中一旦存在大量重叠遮挡，明暗光照的情形，果实识别准确率就会大幅下降，虽然机器学习的引入一定程度上缓解了该问题，但始终需要复杂的图像预处理以及手动提取特征，无法满足采摘机器人在复杂环境情况下的果实识别。

1.2.1.2 基于深度学习的果实识别

随着计算机硬件的不断发展，深度学习也得到了飞速的发展，许多研究人员也将深度学习中的目标检测算法应用到果实识别领域中，深度学习通过多层神经网络从图像中自动提取特征，无需对图像进行复杂的预处理和手动提取特征，与传统图像处理的果实识别相比，具有更快的检测速度和更高的检测精度，目前已成为果实识别的主流算法。

目前用于果实识别的目标检测算法主要有两阶段的 R-CNN 系列算法，单阶段的 SSD，RetinaNet，YOLO 系列算法等等。

Wan 等人[15]通过优化 Faster R-CNN 模型中的卷积层和池化层，实现了更准确，更快的水果检测。Liu Jian 等人[16]在 R-FCN 网络中加入残差网络，有效提高了模型对苹果，橘子，香蕉三类水果的检测精度。Fangfang Gao 等人[17]使用 VGG16 网络作为 Faster-RCNN 的特征提取网络，实现遮挡场景下的苹果识别，检测的平均精度达到 87.9%。Parvathi 等人[18]使用 ResNet-50 网络替换 Faster R-CNN 中的特征提取网络，获得了较高的成熟椰子识别率。Yu 等人[19]同样采用 Resnet50 作为骨干网，同时结合特征金字塔网络架构进行特征提取，获得了更好的鲁棒性和检测精度。李佳俊等人[20]改进了 Faster R-CNN 模型的整体结构，并在 RPN 网络中增加 1×1，和 5×5 两种不同大小的滑动窗口，扩大模型的感受野，提高模型对不同大小草莓的识别性能，最终模型识别成熟草莓的平均精度达到 89.30%，未成熟草莓的平均精度达到 82.07%。

上述两阶段的 R-CNN 系列算法准确率较高，但其检测速度较慢，实时性差，而对果实采摘机器人而言，实时快速的果实识别是很重要的一环，因此两阶段的目标检测算法并不是果实识别最合适的选择。

单阶段目标检测算法检测速度很快，实时性好，并且检测精度较高，更适合用于采摘机器人果实识别。Xing Zhang 等人[21]在 SSD 目标检测模型中使用 MobileNet 轻量级特征提取网络替换 VGG16 网络，大大减小了 SSD 模型的大小，在没有降

低太多识别精度的同时，有效提高了实时性。Yutan Wang 等人[22]提出了改进轻量级 SSD 灵武长枣检测模型，该模型使用 peleenet 网络代替 VGG16 网络作为主干网络，增强了模型的特征提取能力，同时降低了模型的参数量，还在模型中加入 CA 和 GAM 注意力机制，并使用 Inceptionv2 模块替换 SSD 前三层结构，进一步提高模型的检测精度，最终优化后模型的 mAP 达到 97.32%，参数量为原模型的 30.37%。Hongxing Peng 等人[23]提出了 Retinanet-G2S 柑橘检测模型，使用 Res2Net-GF 算法作为骨干网络提升模型的特征提取能力，设计多尺度跨区域特征融合网络替代 FPN 网络，缓解 FPN 网络忽略低层特征定位信息的问题，最后优化了边界框回归定位算法，最终改进模型的 mAP 相比基础模型提高了 3.8%。刘天真等人[24]在 YOLOv3 模型中加入 SE 注意力机制，提高了模型对冬枣的敏感度，相比原 YOLOv3 模型，mAP 提高 4.78%，达到 82.01%。Yunchao Tang 等人[25]提出了改进 YOLOv4-tiny 油茶检测算法，使用 k-means++算法代替 k-means 优化锚框，解决 k-means 随机选择聚类中心，使得聚类容易陷入局部极小值，导致结果不稳定的问题，并在原模型的第二及第三个 CSPBlock 模块后各自加入一个 1×1 和 3×3 的卷积核，降低计算复杂度的同时，让模型更容易的学习到油茶的特征，最终的模型 AP 达到 92.07%。Xinfa Wang 等人[26]在 YOLOv5 中使用 MobileNetV4-Large 作为主干网络，降低了模型参数量，并加入 SE 注意力机制，增加模型对重要特征的注意，最后在三个检测层额外增加小目标检测层来检测较小的番茄，相比原模型的 mAP 增加 1.4%，达到 98.8%，大小仅为原来的 42.48%。Fenghua Wang 等人[27]提出了基于 YOLOv5s 的改进模型 YOLOv5s-CFL 进行青椒检测，使用轻量级的 Ghost 卷积替代 CSP 部分的卷积层，加快模型的检测速度，加入 CA 注意力机制提高 YOLOv5 的特征检测能力，使用 BiFPN 网络代替 Neck 部分的 PANet 网络，加强 YOLOv5 的特征融合能力，改进模型相比原始 YOLOv5s 模型，大小减小了 0.6MB，mAP 高出 1.1%。孙成宇[28]在 YOLOv7 模型的主干网络中加入 SimAM 注意力机制，提高模型对较小辣椒和遮挡辣椒的检测效果，在 Neck 部分使用 Slim-Neck 减小模型大小，最后使用 Shape-IOU 损失来评估预测框和真实框的重叠度，最终模型 mAP 提高了 1.3%，达到 88.7%。

总的来说，基于深度学习的果实识别算法相比传统图像处理方法，检测精度更高，检测速度更快，尤其是单阶段的目标检测算法在果实检测领域中表现出色，但是这些都是建立在一个优秀的大型数据集中。在枝叶遮挡，果实密集重叠的辣椒检测场景下，基于深度学习的辣椒检测算法仍然存在较多漏检和误检的情况，需要对现有的算法进行针对性的优化和改良，才能满足辣椒智能采摘的需求。

1.2.2 果实定位研究现状

果实定位是在识别模型检测出图片里目标果实的类别和位置后，计算目标果实的三维位置的过程，精准的果实定位能更好的帮助机器人完成采摘。目前采摘机器人中主要使用单目相机，双目相机，深度相机等来进行果实定位。

单目相机定位只用单个摄像头获取图像，估算场景中每个像素的深度信息，早期通过匹配图像中的边缘、纹理等特征，结合几何关系推算深度，但这种方法的精度较低，难以满足采摘机器人的定位需求。随着深度学习技术的发展，单目深度估计也得到了极大的发展。高研[29]用单目图像实现苹果定位，在单目图像深度估计模型 HRNet 的解码器网络中加入密集连接机制，卷积注意力模块提高模型的编码性能，还在解码器网络中加入条纹细化模块，突出图像中的边缘特征，最终定位深度误差在 1.6cm 以下。

双目相机定位类似人的双眼，使用两个单目相机同时拍摄，根据视差计算深度。早在 2000 年，Plebe 等人[30]就将双目视觉应用到果实的定位采摘中，先用两个相机拍摄同一柑橘的不同视角，再匹配两张柑橘图像，最后计算目标柑橘在机械臂中的三维坐标。张东航[31]设计了一种基于图像语义特征的双目匹配，减少了光照和阴影导致匹配失败的问题，最终草莓双目图像的匹配准确率达到 96,3%。郑雨睿[32]采用双目视觉实现樱桃番茄的精准定位，匹配算法使用了基于深度学习中的 CREstereo，实现左右视图的精准匹配，最终三维定位的平均误差是 5.99mm。

深度相机是一种能直接获取拍摄图像中每个像素点深度信息的相机，一般能同时提供彩色图像和深度图像，常见的深度相机主要要有结构光，双目视觉，飞行时光三种。麦春艳等人[33]使用 Kinect 深度相机采集果树的深度及彩色图像，将彩色图像映射到深度图像中构建点云数据，分割点云数据得到果实的三维坐标，平均定位误差 8.1mm。王明和[34]使用 Azure Kinect DK 深度相机实现苹果的三维定位，该相机基于飞行时间技术计算深度，处理速度快，受自然光影响小，适合室外作业。平均定位误差为 1.42mm。

综合来看，单目相机定位成本最低，但精度较低，难以满足采摘需求。双目相机定位精度较高，但却依赖复杂的左右视图匹配算法，计算耗时较长，并且比较容易受到光照影响。深度相机定位精度高，不易受到光照影响，选择合适的深度相机作为采摘机器人的视觉系统是最优选择。

1.2.3 采摘机器人研究现状

采摘机器人在大规模农田中能够有效提高生产效率，减少对人工的依赖，缓解劳动力短缺的问题，在现代农业中具有重要的意义。早期的农业机械化采摘主要有

两种方式，第一种是果农用液压剪来代替双手采摘果实，这种方法对果树的损伤很小，但效率较低，劳动强度依然较大。第二种方法是通过振摇果树枝干与果实产生共振，使得果实下落，这种方法效率较高，但对果树的损害很大，采摘的果实品质也会受到影响。现代农业采摘机器人主要依靠各种传感器感知目标果实，再将坐标信息发送给机械臂采摘，这种方法效率很高，工人劳动强度较小，对果树的伤害也很小。

国外如美国、日本、德国等发达国家在农业智能采摘机器人方面研究较早，1968 年，美国 ScbertzCE[35]等人首次提出了机械化采摘的概念，分析了机械化采摘中可能遇到的技术难题，对果树的损伤等问题，但受限于当时的工业水平，研发的机器人是基于振动的，仍然需要人工去操作，属于半自动的采摘机器，这为后续的果实采摘机器人奠定了基础。Kondo 等人[36]研发了一个番茄采摘机器人，这款机器人包括番茄识别定位系统和机械臂执行器。利用光电传感器来识别定位番茄，再将位置信息发送给机械臂，引导采摘。后续该团队[37]又研发了由 CPU 和彩色相机组成的番茄视觉识别系统，提高了番茄识别的准确率和定位精度，并在机械臂末端夹爪在中使用压力传感器，减小对番茄的损害。Williams 等人[38]研发了具有多组机械臂的猕猴桃采摘机器人，该机器人采用深度学习的卷积神经网络作为猕猴桃的识别模型，采用双目视觉作为定位系统，并可以动态的控制四个机械臂协调作业，互不干扰，有效提高了采摘效率。

国内对于果实智能采摘机器人的研究较晚，但随着众多研究人员的不断努力和国家的大力支持，我国的采摘机器人得到了飞速的发展，与国外的差距也在逐步缩小。

吕继东[39]对苹果采摘机器人开展了研究，基于颜色特征和 OTSU 阈值分割实现苹果识别，提出一种能够快速采摘振荡苹果的方法，最后还研发了识别定位系统，并在机械臂中加入传感器来实现自动避障。最终机器人苹果采摘成功率达到 84%。王丽丽[40]研发了一款可以移动导航的番茄采摘机器人，移动底盘基于模块化设计方法和嵌入式控制系统理论设计，自动导航系统基于激光传感技术开发，同时还采用图像处理和双目视觉实现番茄的识别和定位，最后研发了四自由度机械臂，实现番茄的采摘。最终的采摘机器人识别率达到 99%，定位误差小于 10mm，导航误差小于 8cm，采摘成功率达到 86.7%。李爽[41]使用 RealSense D435i 深度相机和 YOLOv5 目标检测模型实现青椒的精准识别和定位，并基于 ROS 系统开发了移动底盘，正常情况下青椒采摘成功率达到 90%，在夜间及枝叶遮挡较多的情况下，采摘成功率会有所降低。杨长辉等人[42]研发了一款柑橘采摘机器人，使用 YOLOv3 目标检测模型来识别番茄，使用 Kinect V2 深度相机获取番茄的三维位置信息，最

终机器人的采摘成功率达到 80.51%。肖旭[43]同样研发了一款柑橘采摘机器人，根据柑橘的生长特性，设计了一个三指末端执行器，可以采摘各种角度的柑橘，机器人视觉系统由 ZED 双目深度相机和改进 YOLOv3 模型组成，在柑橘采摘实验中，采摘机器人识别成功率达到 94%，采摘成功率达到 86%。

1.3 本文主要研究内容

本文主要研究的是基于深度学习的辣椒识别，三维定位和采摘技术，针对复杂环境下辣椒枝叶遮挡，密集重叠等问题，对 YOLOv8 深度学习模型进行改进和优化，并基于机械臂，末端执行器和深度相机，建立辣椒采摘系统，主要研究内容如下：

1.采集自然环境下辣椒图像，并使用 Labelimg 软件对图像进行标注，建立辣椒数据集，针对数据集中的辣椒枝叶遮挡，密集重叠等问题，提出了一个基于 YOLOv8s 的改进优化模型，在模型注意力机制，损失函数和非极大值抑制等方面进行优化，并设计实验，验证模型优化后的性能。

2.利用深度相机获取辣椒的三维数据，完成深度相机的标定和手眼标定，根据相应转换关系，计算目标辣椒在机械臂下的三维坐标，并对辣椒进行姿态分析，辅助机械臂调整姿态完成辣椒采摘。

3.设计辣椒采摘系统，实现视觉系统，机械臂和末端执行器与上位机的通信，完成硬件平台的搭建和软件界面设计。最后进行辣椒采摘实验，分析辣椒采摘系统的有效性。

1.4 本文结构安排

本文共六个章节，每章内容安排如下：

第一章：绪论，首先介绍了本文研究课题的背景和意义，接着介绍果实识别，定位技术及采摘机器人的国内外研究现状，最后介绍了本文的主要研究内容和结构安排。

第二章：辣椒识别相关理论技术，主要介绍了卷积神经网络的组成和目标检测算法中的相关理论基础，最后介绍了目标检测中单阶段和两阶段的经典代表模型。

第三章：辣椒采摘系统设计，主要介绍辣椒采摘系统的设计路线和各部分硬件的选型。

第四章：基于 YOLOv8 的辣椒识别算法，首先介绍了辣椒数据集的制作，接着对 YOLOv8 模型进行了原理分析，然后针对辣椒识别问题对 YOLOv8s 进行了改进和优化，最后设计实验，验证模型的性能。

第五章：辣椒定位与采摘实验。首先介绍相机成像模型和双目测距原理，完成深度相机的标定，再通过手眼标定，计算手眼矩阵，并进行定位实验。接着对辣椒进行姿态分析。最后通过采摘实验验证本文设计的采摘系统的有效性。

第六章：总结与展望，对本文主要工作进行总结，分析存在的不足以及对后续研究的展望。

第二章 辣椒识别相关理论技术

深度学习近年来成为人工智能领域的核心技术之一，在多个领域都取得了突破性进展，本文主要使用深度学习中的卷积神经网络来识别成熟辣椒。本章主要介绍了卷积神经网络，目标检测算法以及算法的评价指标，最后还介绍了几种常见的目标检测模型。

2.1 卷积神经网络基础

卷积神经网络(Convolutional Neural Network,CNN)是深度学习中的一种重要架构，它以卷积运算为核心，能够高效地捕捉数据中的局部模式和空间特征，因此在计算机视觉中的图像分类，目标检测，语义分割等任务中得到了广泛的应用[44]。CNN 由卷积层，池化层，全连接层，激活层等多种类型的层组成，基本结构如图 2-1 所示，每一层都执行特定的功能，最终实现从原始输入到任务输出的映射，下面将介绍 CNN 中的各层结构。

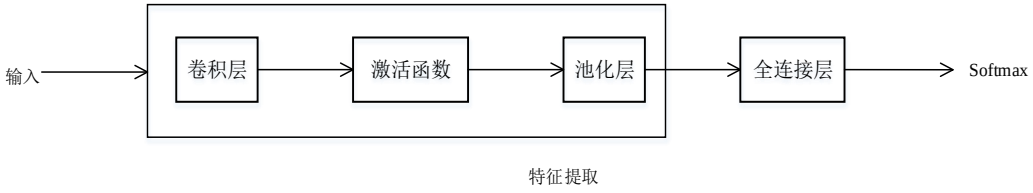


图 2-1 CNN 基本网络结构示意图

2.1.1 卷积层

卷积层是 CNN 中的核心模块，通过对输入的图像和卷积层中的卷积核进行卷积，提取输入图像的局部特征和空间信息。卷积是图像处理中的一种数学运算，通过一个较小的卷积核对输入图像进行加权求和，并不断滑动卷积核重复计算，最终输出一个新的特征图。二维图像卷积的计算公式如式(2-1)所示，计算示意图如图 2-2 所示：

$$(I*K)(i,j)=\sum\sum I(i+m,j+n)K(m,n) \tag{2-1}$$

上式中，K 是卷积核；I 是输入图像；m 和 n 是卷积核的长宽，一般情况下卷积核尺寸是较小的正方形，如 3×3，5×5，7×7 等等，卷积核会以步幅 s，从左到右，从上到下在输入图像的每个位置上进行卷积，得到最终的输出特征图。输出特征图尺寸一般比输入图像小，有时会对输入图像的边界添加额外的零像素，增加输出特

征图的尺寸，同时还可以提高边界特征的提取能力。输出特征图的尺寸与输入图像的关系如式(2-2)和式(2-3)所示，K 是卷积核大小，P 是填充的尺寸，S 是步幅。

$$W_{out}=\frac{W+2P+K}{S}+1 \tag{2-2}$$

$$H_{out}=\frac{H+2P+K}{S}+1 \tag{2-3}$$

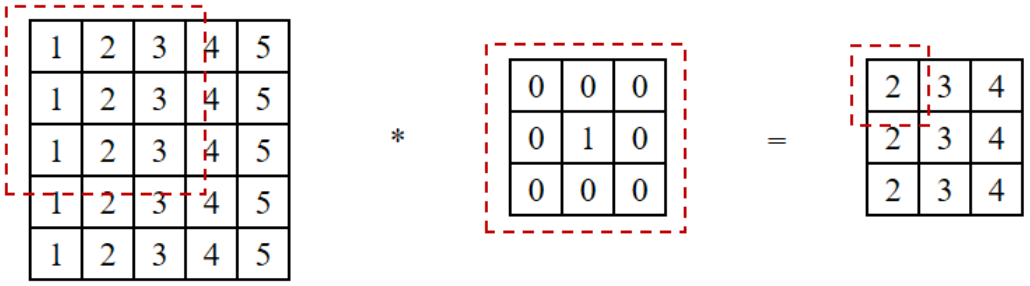


图 2-2 卷积计算示意图

卷积层的卷积核是 CNN 中最重要的训练参数，通过反向传播算法优化。在不同的卷积层中，可以设置不同数量、不同尺寸的卷积核，来提取不同大小和抽象程度的特征，完成图像分类，检测，分割等任务。

2.1.2 池化层

池化层是 CNN 中的一个基本模块，池化层一般位于卷积层后面，卷积层会提取到大量的特征信息，其中不可避免会存在大量的冗余特征，池化层此时就可以解决该问题，它的主要作用就是对卷积层提取到的特征进行降维，在保留关键特征的同时，减小输入特征图的尺寸，从而减小模型的参数量和计算量，防止过拟合，使模型的鲁棒性更好。

池化层的主要参数是池化窗口的尺寸和滑动的步长，池化窗口一般是 2×2，3×3 等，与卷积层相比，池化层没有需要训练的权重参数，因此不会引入额外的参数量，没有引起过拟合的风险。池化层的操作与卷积层相似，都是在窗口中对特征图进行处理，然后滑动窗口覆盖整个特征图，但不会重复处理特征图中的元素，根据池化窗口中操作的不同，可将池化分成最大池化和平均池化，平均池化如图 2-3(a)所示，在每个池化窗口内选择最大的值输出，最大池化如图 2-3(b)所示，在每个池化窗口内，计算区域内所有值的平均值作为输出，最大池化可以保留特征图中最显著的特征，平均池化则能保留区域内的整体信息，可以提高模型的平滑能力和泛化能力。

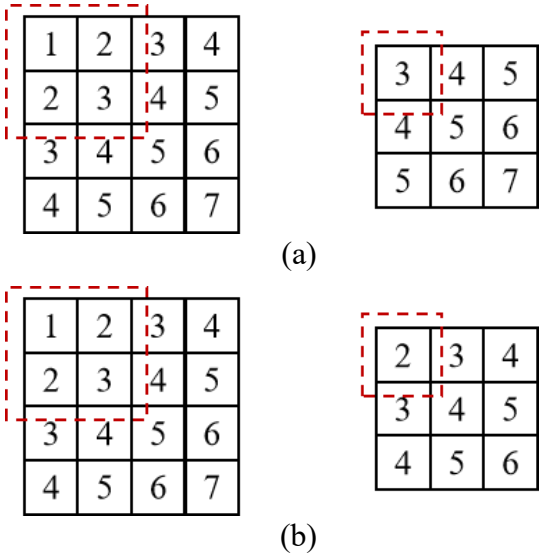


图 2-3 两种池化计算示意图。(a)平均池化；(b)最大池化

2.1.3 激活层

激活层是卷积神经网络中引进非线性的关键层，通过激活函数处理卷积层或全连接层的输出，让网络学习更为复杂的模式和关系，如果没有激活层，那神经网络只是一个线性的模型，简单的线性关系极大的限制了模型的表达能力，对复杂的问题如图像分类，检测，分割等都无法处理。因此为了让模型能够学习和逼近复杂的非线性关系，激活函数必不可少。卷积神经网络中经常用的激活函数有：ReLU 函数[46]，GeLU 函数[47]，Sigmoid 函数[48]，Tanh 函数[49]等。

Sigmoid 函数[48]可将输入映射到(0,1)范围内，表达式及其导数如下式(2-4)和(2-5)所示：

$$\mathrm{Sigmod}(x)=\frac{1}{1+e^{-x}} \tag{2-4}$$

$$\mathrm{Sigmod}'(x)=\mathrm{Sigmod}(x)\left(1-\mathrm{Sigmod}(x)\right) \tag{2-5}$$

式中的 e 是自然函数对数的底数。Sigmoid 函数的图像是一个平滑的 S 形曲线，如图 2-4 所示，在 x=0 时，输出为 0.5，输入 x 趋向负无穷，输出逼近 0，输入 x 趋向正无穷时，输出逼近 1，函数的输出限制在 0 到 1 之间，因此 Sigmoid 函数适合输出概率值，尤其是在二分类问题当中。Sigmoid 函数的导数可以由其自身来计算，如上式所示，这有助于其在反向传播中计算梯度。Sigmoid 函数在输入非常大或非常小时，输出接近 0 或 1，导数也接近 0，这会导致训练非常缓慢并且难以收敛，最终让梯度消失。另外由于 Sigmoid 函数的中心是 0.5，而不是 0，这会导致训练过程中梯度更新不对称，降低训练效率。

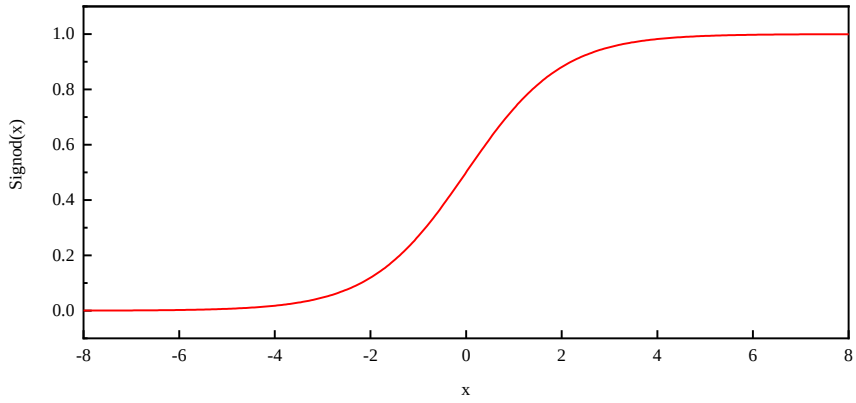


图 2-4 Sigmod 函数图

Tanh 激活函数[49]是双曲正切函数，其表达式及其导数如式(2-6)和(2-7)所示：

$$\mathrm{Tanh}(x)=\frac{e^{-x}-e^{-x}}{e^{-x}+e^{-x}} \tag{2-6}$$

$$\mathrm{Tanh}'(x)=1-\mathrm{Tanh}^{2}(x) \tag{2-7}$$

Tanh 函数图像与 Sigmod 类似，也是平滑的 S 形曲线，如图 2-5 所示，Tanh 函数是以(0,0)为中心，其输出不全是正值，能够有效提高网络的收敛速度，并让梯度的更新更加平衡。但是在极端输入的情况下，Tanh 函数仍然存在梯度消失的问题。

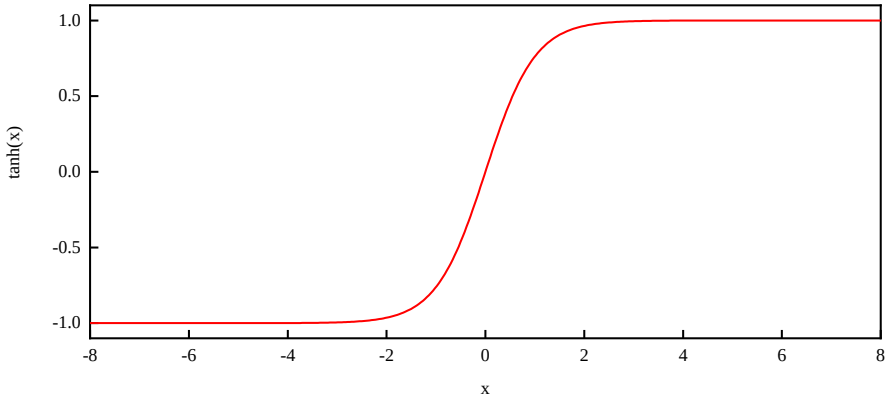


图 2-5 Tanh 函数图

ReLU 激活函数[46]是卷积神经网络中常用的激活函数，简单且高效，ReLU 公式如式(2-8)所示：

$$\mathrm{ReLU}(x)=\max(0,x) \tag{2-8}$$

ReLU 激活函数图像如图 2-6 所示，当输入 x 小于 0 时，输出为 0，导数也是 0，当输入 x 大于 0 时，输出等于输入 x，导数为 1。相比 Sigmoid 或者 Tanh 复杂

的指数计算，ReLU 的计算非常简单，只需与 0 比大小，在输入大于 0 时，ReLU 的导数恒为 1，在反向传播时梯度不会消失，一定程度上缓解了梯度消失的问题，尤其是在深层网络中。但 ReLU 也存在缺点，在输入小于等于 0 时，函数输出恒为 0，虽然一定程度上减少了网络参数，降低了过拟合的风险，但会导致神经元在训练过程中无法更新，也就是神经元死亡的问题。

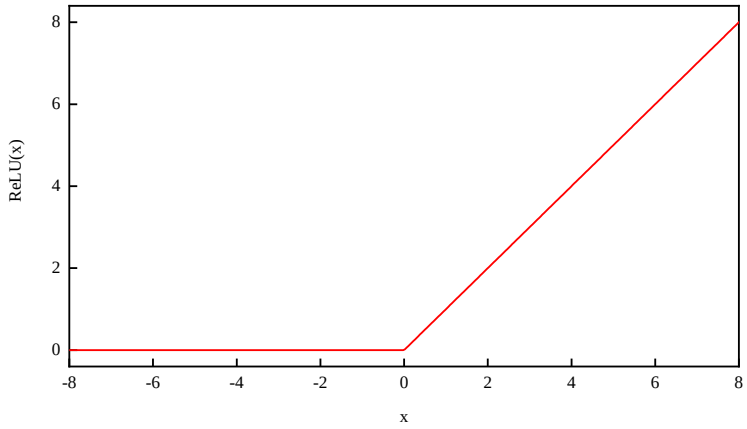


图 2-6 ReLU 函数图

GeLU 是广泛应用于大型预训练模型和复杂任务中的激活函数，近似的数学表达式如式(2-9)所示：

$$\mathrm{GeLU}(x)=0.5x\left(1+\tanh\left(\sqrt{\frac{2}{\pi}}\left(x+0.044715x^{3}\right)\right)\right) \tag{2-9}$$

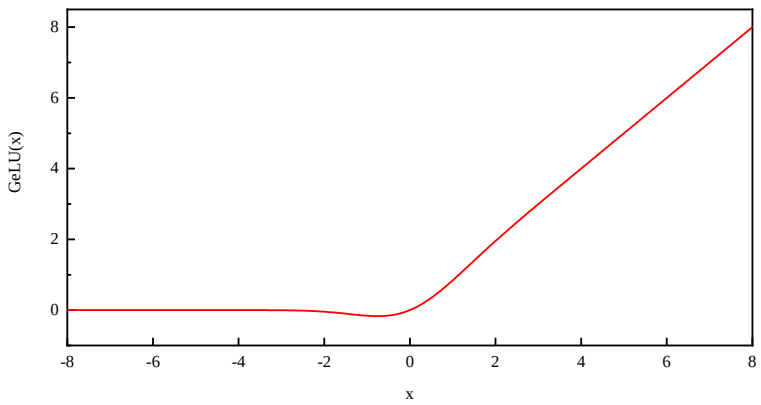


图 2-7 GeLU 函数图

GeLU 激活函数[47]的图像如图 2-7 所示，相比 ReLU，GeLU 更加平滑，在小于 0 的区域不会像 ReLU 那样直接截止为 0，而是通过一个平滑的变化，使得负值部分仍有梯度流动，不会出现 ReLU 神经元死亡的问题，有效提高了训练的稳定性

和性能。GeLU 主要缺点是计算比较复杂，在一些简单的神经网络中，不如 ReLU 简单高效。

2.1.4 全连接层

全连接层一般位于CNN的最后，用于连接网络中的输入神经元和输出神经元，如图 2-8 所示，主要负责将前面卷积层和池化层提取到的特征映射到最终的输出，也就是对特征进行综合和决策，因为要与前一层所有的神经元相连，全连接层的表达能力和学习能力很强，能够捕捉到数据中的高层抽象特征，但也因为包含大量的神经元，全连接层的参数量非常庞大，导致计算量增多，训练时间增加，引起过拟合。因此在实际使用中，往往会采取一些方法对其优化，如 Droput[50]通过在训练过程中随机丢弃一部分神经元，让模型不过度依赖某些特定神经元，减少过拟合，全局池化[51]通过压缩整个特征图的空间维度，实现高效的特征融合和参数精简，避免过拟合。

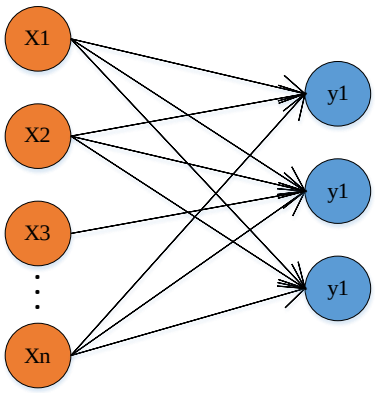


图 2-8 全连接层结构图

2.2 目标检测算法基础

目标检测是计算机视觉领域中最重要的任务之一，主要目标是识别图像中包含的目标类别，并使用方框框出目标在图像中的具体位置。目标检测的应用场景非常广泛，如智能驾驶[52]，人脸识别[53]，缺陷检测[54]]等。本节主要介绍目标检测算法中的一些基本概念，包括交并比，非极大值抑制，损失函数以及评价指标。

2.2.1 交并比

交并比(Intersection over Union,IoU)[55]是目标检测中评估预测框与实际物体位置重合度的重要参数，IoU 如图 2-9 所示，图中 B 是真实框，A 是预测框，IoU 计算公式如下式(2-10)所示：

$$\mathrm{IoU}=\frac{\mathrm{area}\left(\mathrm{A}\cap\mathrm{B}\right)}{\mathrm{area}\left(\mathrm{A}\cup\mathrm{B}\right)} \tag{2-10}$$

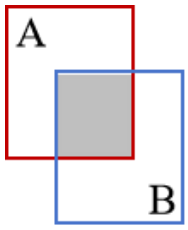


图 2-9 IoU 示意图

上式中 A∩B 是预测框与真实框的交集，如图 2.9 中灰色的部分所示，A∪B 为预测框与真实框的并集，也就是图 2.9 中所有的区域，IoU 的输出值在 0 到 1 之间，IoU 越大，就代表预测框与真实框越接近。一般会设定一个 0.5 的阈值，当 IoU 大于 0.5 时，认为目标被准确检测，否则，即为误检，如果对检测的准确性要求较高，可以使用更高的阈值，如 0.7，0.75 等等。IoU 也存在很多缺点，如无法完全反应预测框与真实框的相对位置，如图 2-10 所示，图中三组预测框与真实框的 IoU 相同，但预测框与真实框的相对位置差别很大，在预测框与真实框没有交集时，IoU 为 0，无法进行梯度运算，更新参数等等。针对上述问题，有很多 IoU 的改进版本被提出，如 GIoU[56]考虑了两个预测框和真实框没有交集的情况，DIoU[59]额外考虑了两个框之间的的距离，尺寸和重叠度，CIoU[59]在 DIoU 的基础上额外考虑了两个框的长宽比。根据检测任务的实际需要，选择合适的 IoU，可以有效提高模型的检测性能。

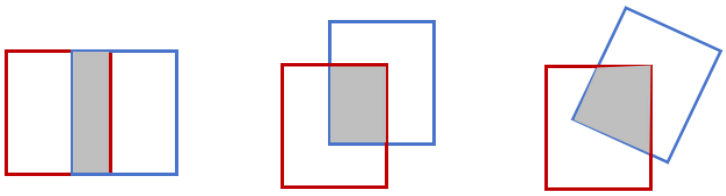
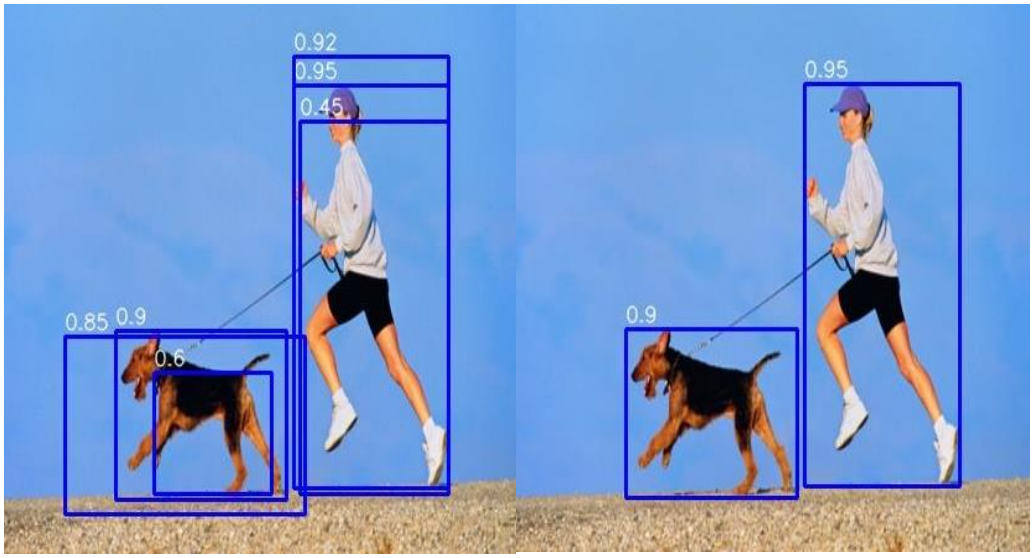


图 2-10 相同 IoU 的不同相对位置。(a)情形 1；(b)情形 2；(c)情形 3

2.2.2 非极大值抑制

非极大值抑制(Non-Maximum Suppression,NMS)[58]是目标检测后处理中常用的方法，其主要是在多个预测框中保留最接近真实框的那一个，去除其他多余的预测框，提升检测的准确性。目标检测任务中，通常会在目标周围产生多个候选边界框，如图 2-11(a)所示，图中存在两个目标，但却有多个预测框检测到了，NMS 的作用就是保留两个最准确的预测框。

NMS 算法的基本步骤如下，首先对所有的预测框根据其置信度分数进行排序，然后选择其中得分最高的框，并计算其他剩余的预测框与最高分框的 IoU，如果 IoU 高于设定的阈值，就认为该候选框与最高分框预测的是同一个目标，也就是这个候选框是冗余的，可将该候选框删除，之后再从剩余的框中选择得分最高的框，重复以上步骤，直至处理完所有的候选框。图 2-11(b)即为经过 NMS 处理过的示意图，最终保留了两个最准确的预测框。



(a) (b)

图 2-11 未使用 NMS 和使用 NMS 的比较。(a)未使用 NMS；(b)使用 NMS

2.2.3 损失函数

在目标检测任务中，设计合适的损失函数至关重要，它决定了模型的训练过程以及模型最终的检测结果[59]，目标检测任务主要包括两类损失：分类损失和回归损失，下面将介绍几种目标检测中常见的几个损失函数。

交叉熵损失函数是目标检测中最常用的分类损失函数，通过比较网络输出的概率和真实标签之间的差值来优化模型，在二分类问题中，常使用二元交叉熵损失(Binary Cross-Entopy Loss,BCE)，表达式如式(2-11)所示：

$$L_{BCE} = -\left[y\log\left(p\right) + \left(1-y\right)\log\left(1-p\right)\right] \tag{2-11}$$

上式中，y 是真实标签 0 或 1，p 是网络输出的类别为 1 预测概率值，1-p 是类别为 0 的预测概率值。预测概率值越接近真实标签值，BCE 损失函数输出越小。在多分类问题中常用多元交叉熵损失，计算公式如(2-12)所示：

$$L_{CCE} = -\sum_{i=1}^{N} y_i \log\left(p_i\right) \tag{2-12}$$

上式中，N 为类别，y_i 是第 i 个真实标签，若样本属于该类别，则 y_i=1，否则 y_i=0，p_i 是对该类的预测值，因此真实类别的预测概率才会影响损失函数。交叉熵损失能够计算出模型预测概率值和真实标签值的差值，为优化过程提供清晰的反馈，提高分类的准确性。

目标检测中常用的边界框回归损失函数有 L1 损失函数，L2 损失函数，Smooth L1 损失函数等，L1 和 L2 损失函数的公式如式(2-13)和(2-14)所示，图像如图 2-12 中的黑线红线所示：

$$L1=\frac{1}{N}\sum_{i=1}^{N}\left|y_i-y_i\right| \tag{2-13}$$

$$L2=\frac{1}{N}\sum_{i=1}^{N}\left(y_i-y_i\right)^2 \tag{2-14}$$

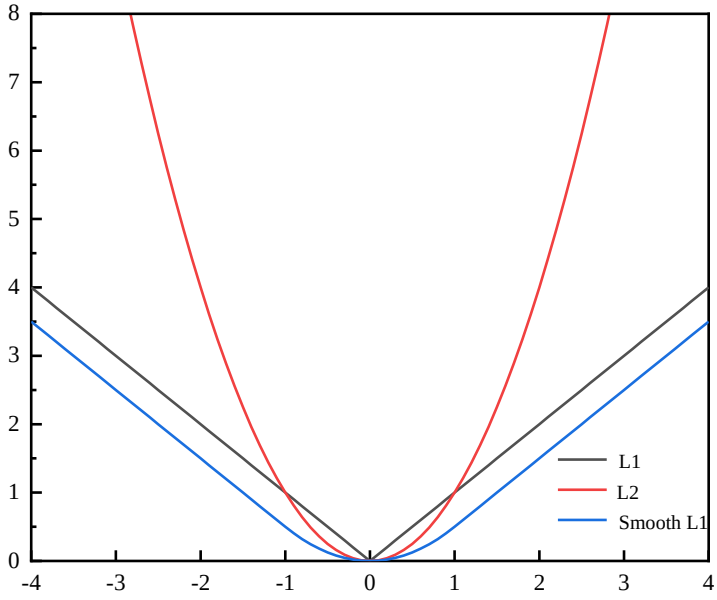


图 2-12 L1,L2 和 Smooth L1 损失函数图

L1 损失函数又称绝对误差损失，计算的是真实值和预测值差的绝对值的平均，L2 损失函数也加均方误差损失，它计算的则是真实值和预测值差的平方的平均，L1 损失函数是线性的，计算比较简单，对数据中的离群点不敏感，但是 L1 函数不是连续可导，并且梯度不会变化，可能会出现梯度消失，收敛较慢的问题。L2 函数则会因为差值的平方放大偏差，惩罚模型中较大误差的样本，因此对离群点十分敏感，但 L2 损失连续光滑，可导，不会出现梯度消失的问题，收敛较快。Smooth L1 是结合了 L1 和 L2 特性的损失函数，公式如(2-15)所示，图像如图 2-12 中蓝线所示：

$$L_{SmoothL1}=\begin{cases}0.5x^2 & |x|\le 1\\ |x|-0.5 & |x|>1\\ x=y-\hat{y}\end{cases} \tag{2-15}$$

当误差小于 1 时，使用 L2 损失，梯度平滑，有利于训练，当误差大于等于 1 时，使用 L1 损失，减小对离群点的惩罚。Smooth L1 损失函数结合 L1 和 L2 损失的优点，提供了一种更加平滑，和鲁棒性的损失，在目标检测的边界框回归计算中得到了广泛的应用。

2.2.4 评价指标

在模型训练完成后，需要评价指标衡量模型的性能，目前常用的评价指标有精确率(Precision)，召回率(Recall)，平均精度均值(mAP)，准确率(Accuracy)等，下面将介绍上述评价指标，首先介绍一下混淆矩阵，如表 2-1 所示

表 2-1 混淆矩阵

	错误	正确
阴性	False Negative(FN)	True Negative(TN)
阳性	False Positive(FP)	True Positive(TP)

上表中，FN 表示真实为正样本，预测为负样本，FP 表示真实为负样本，预测为正样本，TN 表示真实为负样本，预测为负样本，TP 表示真实为正样本，预测也为正样本，混淆矩阵展示了模型真实标签和预测标签之间的关系，基于混淆矩阵，可以计算上述提到的一些评价指标。

准确率表示模型正确分类所占的百分比，准确率越高，代表模型预测越准确，准确率公式如(2-16)所示：

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \tag{2-16}$$

精确率表示预测为正类的样本中，真实值为正样本所占的百分比，精确率越高，就代表模型错误分为正样本的情况越少。公式如(2-17)所示：

$$\text{Precision}=\frac{TP}{TP+FP} \tag{2-17}$$

召回率表示在所有正样本中，预测正确所占的百分比，召回率越高，代表模型能检测出更多的正样本，召回率公式如(2-18)所示：

$$\text{Recall}=\frac{TP}{TP+FN} \tag{2-18}$$

平均精度是精确率-召回率曲线下的面积，代表不同召回率下模型精度的平均值，反映了模型在不同检测阈值下的性能。AP 只表示单一类别，实际任务中往往存在多个类别，因此常用多个类别 AP 的平均值(mAP)衡量模型在多个类别上的整体精度，mAP 值越高，代表模型在所有类别上的检测性能越好，AP 及 mAP 公式如(2-19)和(2-20)所示：

$$\mathrm{AP}=\int_0^1 P\left(y\right)dy \tag{2-19}$$

$$\mathrm{mAP}=\frac{\sum_n^N \mathrm{AP}_n}{N} \tag{2-20}$$

总的来说，这些指标的选择需要根据实际任务和具体应用来确定，从而更好地指导模型的优化。

2.3 经典目标检测模型

目标检测模型主要可以分成两大类：单阶段和两阶段，单阶段检测模型直接从输入图像中预测目标的类别和位置，代表模型有 SSD(Single Shot Multibox Detector)[60]，YOLO(You Only Look Once)系列模型[66]等。两阶段模型首先生成一系列候选区域，然后再对候选区域进一步处理，更好地分类和定位物体。两阶段的代表模型有 R-CNN(Region with CNN features) [65]，Fast R-CNN[66]，Faster R-CNN[67]等，本节将简单的介绍下这些经典模型。

2.3.1 R-CNN

R-CNN 模型是 Ross Girshick 等人[65]在 2014 年提出的模型，首次将深度学习用于目标检测任务中，是目标检测从传统方法向深度学习转变的关键模型。

RCNN 模型的检测步骤如图 2-13 所示，R-CNN 是经典的二阶段检测模型，首先使用选择性搜素(Selective Search)根据输入图像的颜色，纹理，大小等信息将图像分割成多个小的区域，再通过合并相邻区域生成 2000 个左右的候选区域(Region Proposals)，覆盖几乎所有可能的物体，每个候选区域都包含一个可能的背景和目标。然后将所有的候选区域缩放至固定的大小输入到 CNN 中，提取候选区域的深层特征，最后使用 SVM 对提取的特征进行分类，用线性回归模型准确修改候选区域的边界框，完成目标检测。

R-CNN 首次使用 CNN 来提取特征，相比传统检测方法，精度有了显著的提高。但是每张图像生成的 2000 个候选区域需要大量的内存，并且需要经历 CNN，SVM 和回归模型多个阶段的训练，训练流程十分繁琐，计算量庞大，模型检测速

度很慢，实时性差，无法实时检测。但总的来说，R-CNN 仍是目标检测历史中里程碑式的模型，将 CNN 引入到目标检测任务中，为之后目标检测模型的优化指明了方向。

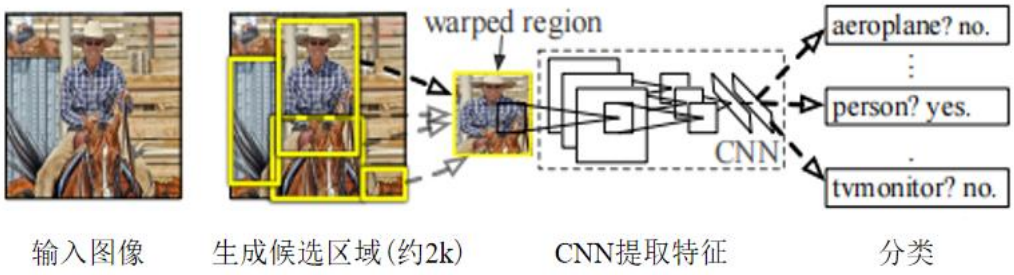


图 2-13 R-CNN 检测步骤[65]

2.3.2 Fast R-CNN

Fast R-CNN 模型是 Ross Girshick 等人[66]2015 年在 R-CNN 基础上提出的改进模型，其框架如图 2-14 所示。Fast R-CNN 工作流程如下，先使用选择性搜素算法生成候选区域，将原始图像输入到 CNN 中生成一个共享的特征图，候选区域映射到共享特征图上生成特征矩阵，每个特征矩阵通过感兴趣区域池化(Region of interest Pooling,RoI pooling)到固定 7×7 的大小，再展平为向量，并经过两个全连接层得到特征向量特，最后特征向量连接两个全连接层，用于边界框类比预测和边界框位置回归。相比 R-CNN，Fast R-CNN 将 CNN 特征提取，SVM 边界框分类和回归模型的边界框回归集成到一个 CNN 中，只需对原始图像提取一次特征，避免了候选区域单独提取特征的重复计算，训练流程简单高效，检测速度大幅提升。Fast R-CNN 的不足是仍然使用选择性搜素算法生成候选框，限制了整体模型的实时性。这也是后续二阶段目标检测模型的重点优化方向。

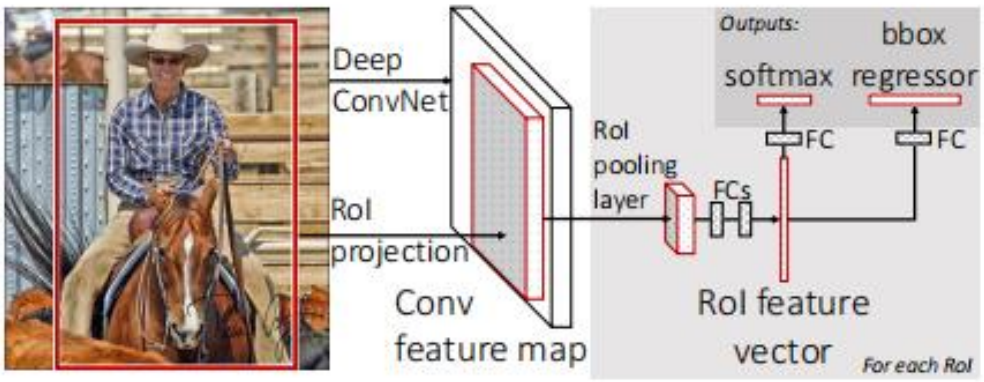


图 2-14 Fast R-CNN 模型框图[66]

2.3.3 Faster R-CNN

Faster R-CNN 是由何凯明等人[67]在 2015 年提出的目标检测模型，其解决了 Fast R-CNN 使用选择搜素算法而导致实时性差的问题，其使用 Region Proposal Network(RPN)网络生成候选区域，网络框图如图 2-15 所示。先使用 3×3 的滑动窗口在 CNN 中提取的共享特征图上进行特征提取，在每一点处生成一系列不同大小和长宽比的 9 个锚框(Anchor)，然后通过 Softmax 分类和边界框回归从大量的 anchor 中提取出 positive anchor 作为候选区域，后续候选区域的分类和回归与 Fast R-CNN 一样。Faster R-CNN 首次将特征提取，候选区域生成，候选区域的分类和边界框回归集成到一个统一的 CNN 网络中，有效提升了检测精度和速度，在之后的学术研究和工业检测中得到了广泛的应用。

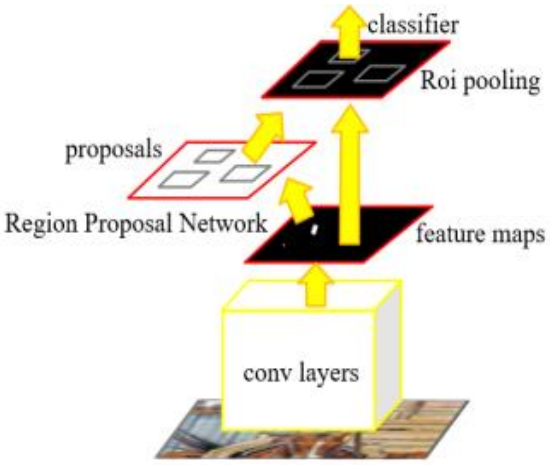


图 2-15 Faster R-CNN 模型框图

2.3.4 SSD

SSD(Single Shot Multibox Detecor)[60]是单阶段目标检测模型的代表模型，不同于上述两阶段模型需要先生成候选区域，SSD 可以直接对输入图像进行目标检测，速度快，检测精度较高，SSD 框架如图 2-16 所示，主要包含三部分：特征提取网络，多尺度检测和检测头，SSD 使用 VGG16 作为主干网络，提取输入图像的特征图，然后采用多尺度检测方法，有六个不同大小的特征图，用大尺度特征图的浅层信息检测小目标，小尺度特征图的深层信息检测大目标，同时还在每一层特征图中使用了 Faster R-CNN 中的锚框方法，匹配不同大小和长宽比的目标，最后通过检测头进行边界框回归和标签分类，再使用 NMS 方法来选出最优的预测框。在检测复杂场景中的小目标时，SSD 的检测精度要比 Faster R-CNN 要差一些，但 SSD 训练效率高，检测速度快，实时性好，使得其在实际任务中有更广泛的应用。

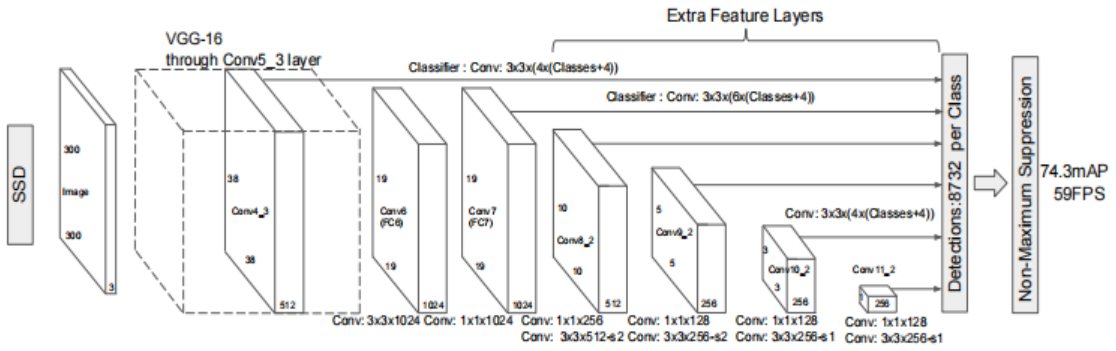


图 2-16 SSD 模型结构图[60]

2.3.5 YOLO 系列

YOLO 系列模型同样是单阶段目标检测模型的代表模型，不需要先生成候选区域，而是将目标检测视作回归问题，直接从输入图像中预测物体的类别和位置。本节将介绍一下 YOLO 系列的经典模型，本文使用的基线模型是 YOLOv8 算法，将在第四章中详细介绍。

YOLOv1[61]是 Joseph Redmon 等人在 2016 年提出的第一个 YOLO 模型，其网络框架如图 2-17 通过 GoogLeNet 分类网络将输入图像分成 S×S 的网格，每个网格预测一组边界框的位置和对应的类别，与 R-CNN 系列算法相比，YOLOv1 的检测速度很快，实时性好，但检测精度不够高。

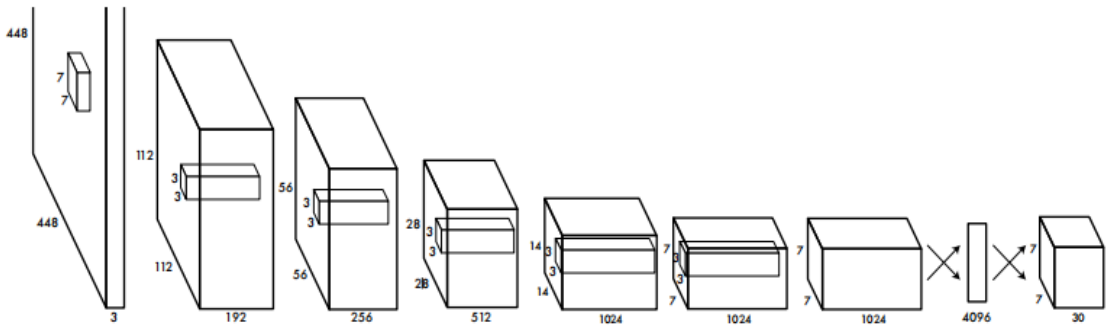


图 2-17 YOLOv1 模型结构图[65]

YOLOv2[62]在 YOLOv1 基础上将特征提取网络换成 Darknet-19，Darknet-19 网络相比 GooLeNet 网络，更加高效和轻量，并且在每个卷积层后使用了批量归一化方法(Batch Normalization,BN)加速模型的训练，提高模型的泛化能力，减小过拟合的影响，同时 YOLOv2 还采用了多尺度训练和 Faster R-CNN 中的 anchor 机制，让模型更好地检测不同尺度和形状的目标，尤其是提升了模型对较小目标的检测精度。

YOLOv3[63]使用了一个更深的特征提取网络 Darknet-53，Darknet-53 使用了残差连接和更深的网络连接，相比 Darknet-19 网络，特征提取和表达能力更强，此外，YOLOv3 还借鉴了特征金字塔网络，在不同尺度的特征图中提取特征，提高了模型对多尺度目标的检测精度。

YOLOv5 由 Ultarlytics 团队在 2020 年开发，在前几个 YOLO 系列模型的基础上，提出了许多创新和改进，进一步提高了检测精度和速度，YOLOv5 模型框架如图 2-18 所示，主要由三个部分组成：特征提取，特征融合和检测头，特征提取网络在 Darknet-53 的基础上，加入了 Cross-Stage PArtial (CSP)网络，在保留较高特征提取和表达能力的同时，减少了计算量，提高了推理速度。在特征融合部分，使用了 FPN＋PAN(Path Aggregation Network)的结构，将深层语义信息和浅层的细节信息融合，增强了不同特征层之间的信息流动，进一步提高了多尺度检测能力。检测头部分引入了了 CIoU，GIoU 等改进 IoU 损失函数，减小了边界框之间的误差，提高了预测框的定位精度。

YOLOv5 除了在模型结构上的创新外，它是基于 PyTorch 框架实现，训练效率高，便在不同平台部署，并且提供了五种不同尺寸的模型，使用者可根据实际应用和计算资源的情况选择合适的版本训练。灵活，高效，容易部署等诸多优点让 YOLOv5 在学术界和工业界得到了广泛的应用。

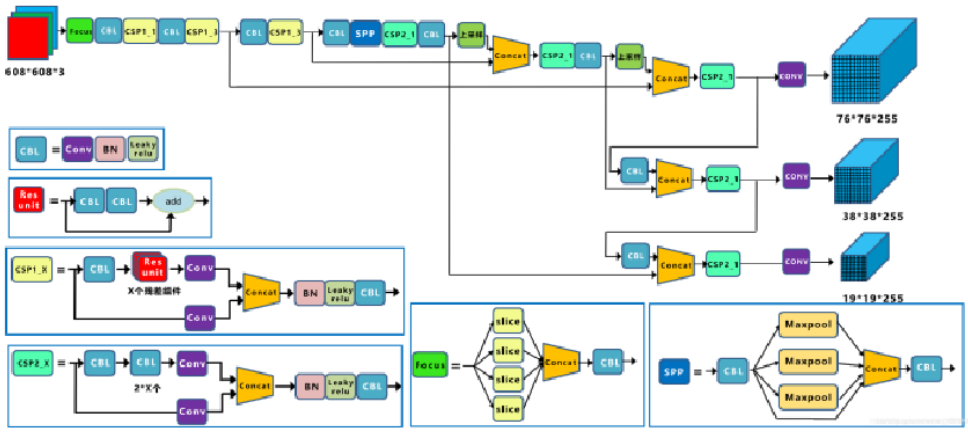


图 2-18 YOLOv5 网络结构图

2.4 本章小结

本章首先介绍了 CNN 的主要组成部分，然后介绍了目标检测中的关键理论，包括交并比，非极大值抑制，损失函数以及评价指标，最后介绍目标检测中的一些经典模型，包括两阶段的 R-CNN 系列模型和单阶段的 SSD，YOLO 系列模型。

第三章 辣椒采摘系统设计

辣椒采摘目前主要依赖人工采摘，人工采摘工作效率低，劳动强度大，人工成本高，对于辣椒人工采摘的诸多弊端，本文提出了基于深度学习的辣椒识别及采摘系统，该系统可实现对辣椒的精准识别和采摘。本章主要介绍该系统的设计路线和硬件选型，硬件系统主要包括机械臂、末端执行器、相机、上位机等。

3.1 辣椒采摘系统设计

本文提出的辣椒采摘系统主要包括辣椒图像处理，坐标转换和机械臂采摘三个模块，设计路线如图 3-1 所示。

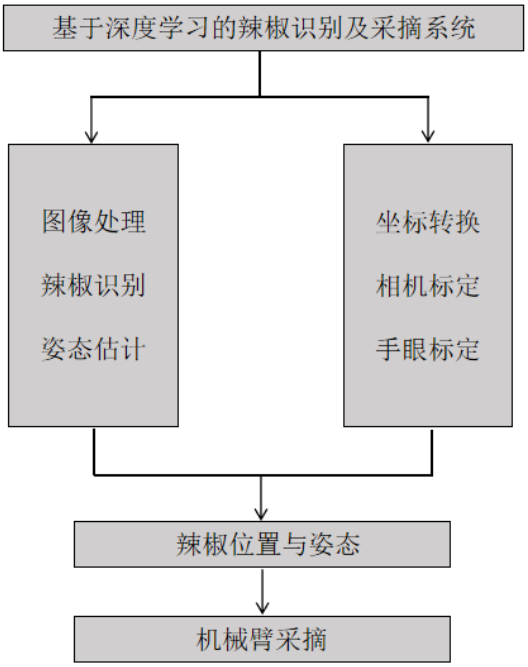


图 3-1 辣椒采摘系统路线图

辣椒图像处理是辣椒采摘系统中最重要的部分，该部分主要功能首先是使用相机拍摄辣椒图片，然后通过本文提出的基于 YOLOv8s 的改进辣椒识别模型对图片检测，预测图片中辣椒的位置，对于偏转较大的辣椒还要计算其偏转角度，方便机械臂采摘，这部分内容将在本文第四章节中详细介绍。

YOLOv8s 预测的辣椒位置是辣椒在图片中的位置，此时机械臂还无法对其采摘，还需要将目标辣椒在图像中的二维坐标转换到机械臂基坐标系下的坐标，也就

需要完成相机标定以及相机与机械臂的联合手眼标定，这部分内容将在第五章中详细介绍。

本文的辣椒采摘系统实物图如图 3-2 所示，整体由上位机，RealSense D415 深度相机，ABB IRB1200 六自由度机械臂，机械臂控制柜，机械臂示教器，末端执行器等组成。本文使用 PC 作为上位机，其余部分通过 USB 接口线和网线与上位机连接，由上位机对整个采摘系统进行控制。

图 3-2 辣椒采摘系统实物图

3.2 辣椒采摘系统硬件选型

辣椒采摘系统各个部分都对最终系统的采摘性能有较大的影响，因此为采摘系统选择合适的硬件至关重要，下面将介绍辣椒采摘系统中的机械臂，末端执行器和深度相机。

3.2.1 ABB IRB1200 机械臂

机械臂是辣椒采摘系统中重要组成部分，机械臂的选型应当考虑到一下几点：第一：为了实现辣椒的精准采摘，机械臂的定位精度和重复定位精度应当够高；第二：辣椒枝叶遮挡较多，机械臂应当具有较高的灵活性；第三，为了实现视野中辣椒的一次性采摘，机械臂的工作半径应当较大。

本文使用 ABB IRB 1200 机械臂来搭建采摘系统，其结构组成如图 3-3 所示，图中 A 至 F 分别是机械臂的六个转动轴，具体性能参数如表 3-1 所示，该机械臂的重复定位精度是±0.01mm，绝对定位精度平均值是 0.14mm，精度很高可以实现辣椒的精准采摘。并且机械臂具有六个转动轴，灵活性够高，可以很好的应对辣椒

果实和枝干的遮挡问题。同时机械臂工作半径达到 900mm，可以一次采摘视野中的所有辣椒。机械臂通过套接字(Socket)通讯与上位机通信，机械臂可以实时发送当前采摘状态至上位机，上位机可以实时发送目标辣椒位置和姿态给机械臂执行抓取。综上所述，该机械臂满足辣椒采摘的需求。

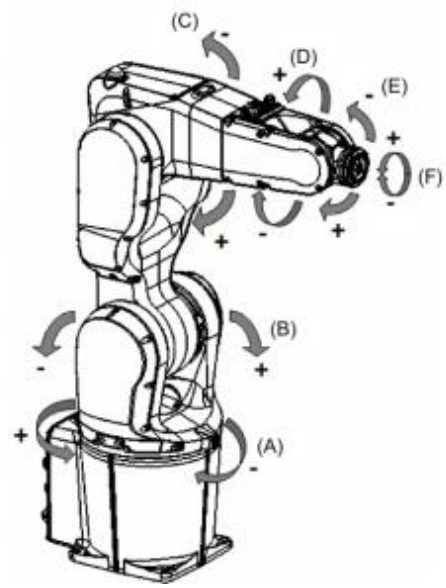


图 3-3 ABB IRB 1200 机械臂结构图

表 3-1 ABB IRB 1200 机械臂参数表

配置	参数
轴数	6
臂展	900mm
最大负载	5kg
关节范围	轴 1：-170°～+170° 轴 2：-100°～+130° 轴 3：-200°～+70° 轴 4：-270°～+270° 轴 5：-130°～+130° 轴 6：-400°～+400°
重复定位精度	±0.01mm
绝对定位精度	0.14mm

3.2.2 末端执行器

辣椒采摘的末端执行器需要考虑到一下两点：第一辣椒生长密集，果实重叠，枝叶遮挡情况较多，因此末端执行器应当小巧灵活，第二辣椒采摘时，需保证辣椒

果实完整，不能损伤辣椒表面。根据上述需求，本文选用的末端执行器是二指电动型夹爪，其结构组成如图 3-4 所示，具体的性能参数如表 3-2 所示，该夹爪结构简单，小巧灵活，精度较高，可以满足重叠遮挡的辣椒采摘需要。此外夹爪部分是由柔性材料所制，在采摘时不会损伤辣椒表面。夹爪通过串口通讯与上位机通信，上位机可发送指令控制夹爪开关状态。

图 3-4 二指电动夹爪结构图

表 3-2 二指电动夹爪参数表

配置	参数
尺寸	58×142×156mm
重量	370g
抓取范围	10mm-140mm
抓取重量	0.6kg
供电电压	12-24V

末端执行器通过一个连接器与机械臂六轴末端相连，机械臂原本的工具中心点(Tool Coordinate System,TCP)在第六轴的末端，在加装了末端执行器后，需将 TCP 修改到末端执行器的夹取位置处，否则机械臂无法将末端执行器运动到指定辣椒处采摘。本文采用机械臂自带的四点标定法标定 TCP，即不改变末端执行器的夹取点，只改变机械臂不同轴的姿态，记录四组不同的姿态信息，计算求解得到最终末端执行器的 TCP 点相对原 TCP 的位置是(-0.98mm，0.99mm，226.02mm)，姿态保持不变。

3.2.3 深度相机

本文使用深度相机获取辣椒在空间中的三维坐标信息，目前主流的深度相机主要有三类：双目视觉，结构光和飞行时间(TOF)。

结构光深度相机是将经过编码的光斑照射到物体表面，再由专门的红外摄像头进行采集，根据编码光斑的畸变即可计算相应的深度，结构光深度相机精度较高，但受环境光干扰严重。

双目视觉深度相机使用两个相同摄像头同时拍摄同一物体，通过算法计算视差获取物体的深度信息，双目深度相机成本低，但匹配算法复杂，计算量较大。

TOF 深度相机通过计算发射和接收的光脉冲飞行时间计算深度，抗干扰能力较强，但功耗较大且无法检测多棱角物体。

本文研究对象是自然环境下的辣椒，综合考虑三种深度相机的优缺点，本文最终选择 Inter RealSense D415 深度相机作为辣椒的视觉定位系统，D415 深度相机的相关参数如表 3-3 所示，基本结构如图 3-5 所示。

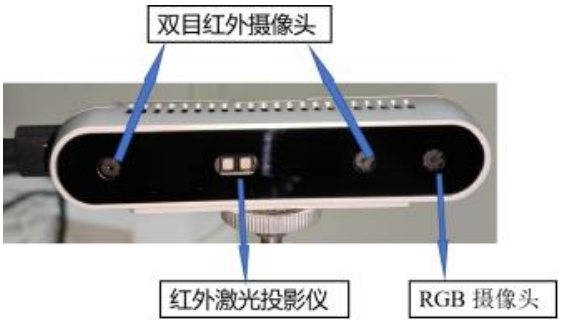


图 3-5 RealSense D415 深度相机结构图

表 3-3 RealSense D415 深度相机参数

名称	参数
Depth Resolution	Up to 1280×720
Depth Frame Rate	Up to 90 fps
Dimensions	99mm×20mm×23mm
Minimum Depth Distance	0.16mm
Baseline	55mm

D415 深度相机包括两个双目红外摄像头，一个红外激光投影仪和 RGB 摄像图，两个双目红外摄像头拍摄同一场景中的图片，对左右图片进行匹配后，通过视差和相机内参计算物体的深度，红外投影仪可以投射一系列不可见的红外条纹到场景中，在低纹理环境中提供可识别的特征点，避免双目因缺失纹理而导致匹配识别，RGB 摄像头用来捕捉彩色图像，与深度图对齐。RealSense D415 深度相机使

用双目视觉和辅助结构光的组合实现了深度的精确测量，满足辣椒采摘系统的定位需要。

3.3 本章小结

本章首先介绍了本文提出的基于深度学习的辣椒识别及采摘系统的设计路线，接着对辣椒采摘系统的机械臂，末端执行器以及深度相机等硬件选型进行了详细介绍。

第四章 基于 YOLOv8 的辣椒识别

辣椒在生长时，往往会存在大量相互重叠，枝叶遮挡的情况，为了实现对辣椒的精准识别和定位，本章基于深度学习进行了深入的研究，首先建立了高质量的辣椒数据集，然后针对辣椒的识别难点对基线模型 YOLOv8s 进行了针对性的改进和优化，最后通过实验比较本文提出的模型和原模型的检测效果，验证了本文提出的模型的优越性。

4.1 辣椒图像数据集制作

4.1.1 辣椒图像采集

本文识别的辣椒品种主要是长条形的绿色辣椒，辣椒图像数据集主要通过实地采集和网络爬取得到，网络上的有效辣椒图像较少，小部分辣椒数据从网络获得，大部分是在实地采集，辣椒图像的采集地点在四川省宜宾市附近的辣椒田地，拍摄时间在 2024 年的 6 月到 7 月，6 月到 7 月正是辣椒成熟的时间，拍摄时间从早到晚，覆盖全天，同时也在各种天气条件和阳光照射角度进行了拍摄，保证了辣椒数据集的多样性。

最终的辣椒数据集中一共 2488 张图片，其中现场拍摄的有 1751 张，网络爬取的有 737 张，图 4-1 展示了数据集中的部分辣椒图像，图片中大多存在枝叶遮挡，密集重叠的问题，对辣椒检测模型的要求很高。

图 4-1 辣椒数据集部分图像展示

4.1.2 辣椒图像标注

本文使用 labelimg 标注工具将辣椒数据集标注为 YOLO 格式，labelimg 软件标注界面如图 4-2 所示，使用长方形框将原始图像中的辣椒目标框出，再选择标签“chili”(本文的辣椒标签统一使用“chili”来表示)，每张辣椒图像的标注文件包括方框的坐标和相应类别，标注完成后，会以 txt 文本的格式保存到预先设置的文件夹中。

图 4-2 labelimg 标注辣椒数据集

4.2 YOLOv8 模型分析

4.2.1 YOLOv8s 网络结构

YOLOv8(You Only Look Once version 8)是 YOLO 系列中兼顾精度和速度的版本，它建立在过去几代 YOLO 模型的基础上，同时结合了当下最新的亮点和改进，具有较好的检测性能。

YOLOv8 的框架如图 4-3 所示，主要包括三个部分，第一部分是主干网络(Backbone)，Backbone 部分将 YOLOv5 中的 C3 模块替换为 C2f 模块。C2f 模块通过更多的分支连接，有更丰富的梯度流，在轻量化的同时增强了模型的特征提取能力；第二部分是颈部网络(Neck)，Neck 部分继续沿用了 FPN+PAN 的策略，融合了不同大小的特征图中的信息，让模型能够更好的检测不同大小的目标；第三部分是头部网络 Head，Head 部分采用了解耦头结构，分离了检测头和分类头，还将 anchor-based 换成了 anchor-free，直接预测目标中心点偏移和宽高，简化模型减少超参数

依赖。此外，YOLOv8 的正负样本分配方式不再采用静态分配策略，而是使用了 Task-Aligned Assigner 动态分配策略，对所有像素点的分类和回归的加权分数进行排序，再选择前 k 个作为正样本，提升正负样本分配质量。YOLOv8 的分类损失用的是 BCE Loss，回归损失用的是 Distribution Focal Loss +CIOU Loss，更准确的优化预测框的位置，这些改进和优化使得 YOLOv8 性能处于当前目标检测算法中的前列。

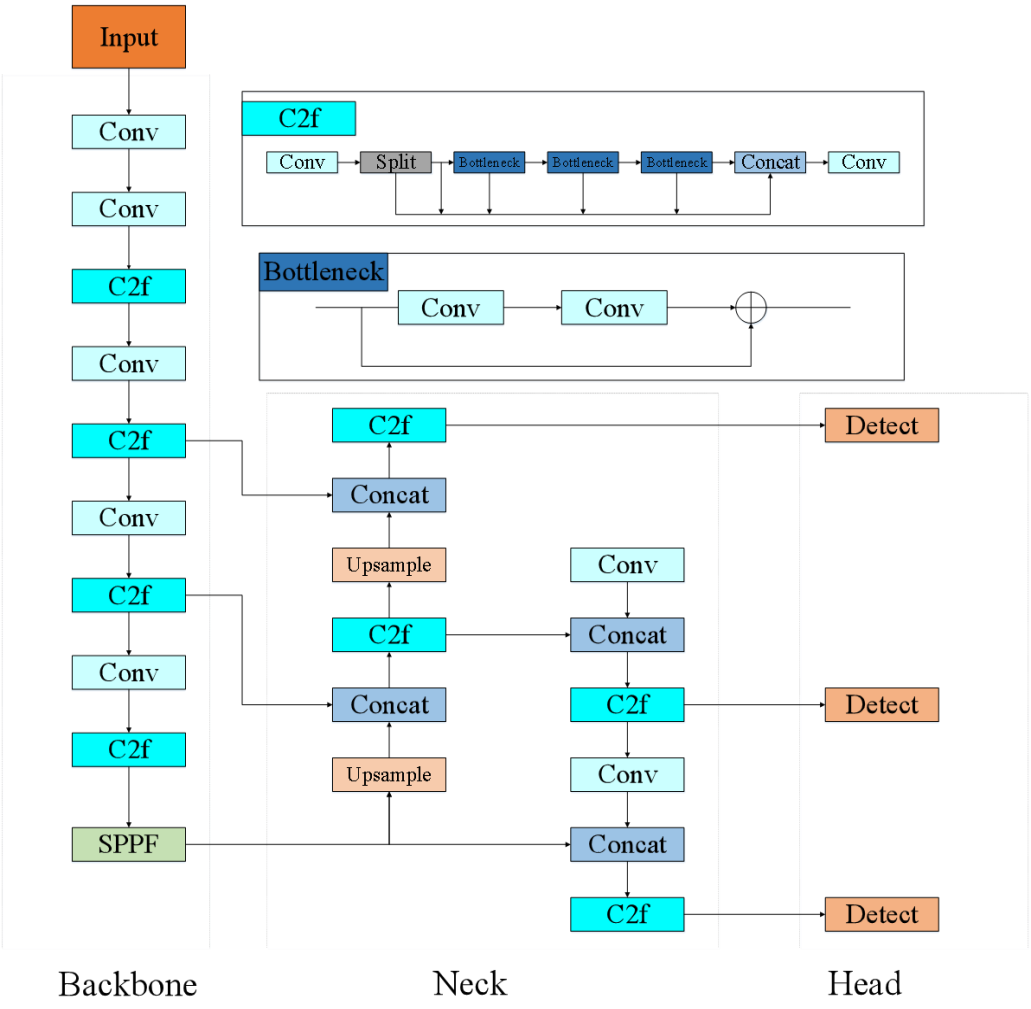


图 4-3 YOLOv8 网络结构图

4.2.2 YOLOv8s 检测结果

本文使用 YOLOv8s 作为辣椒检测的基线模型，其在本文辣椒数据集上的检测结果如表 4-1 所示：

表 4-1 YOLOv8s 辣椒检测结果

模型	P(%)	R(%)	mAP(%)	FPS	Parameters(M)
YOLOv8s	81.2	77.4	84.1	62.9	11.14

从表中可知，YOLOv8s 的检测精度和检测速度都还不错，其准确率达到了81.2%，召回率达到了 77.4%，mAP 达到了 84.1%，FPS 达到了 62.9，但作为辣椒采摘机器人的检测模型，检测精度还是不够高，还需要对模型进行针对性的改进和优化，提高检测性能。

4.3 改进 YOLOv8s 辣椒检测模型

尽管YOLOv8s的精度和速度已经足够优秀，但本文数据集中的辣椒枝叶遮挡，密集重叠的情况较多，这使得YOLOv8s无法完全发挥出他的性能，因此需要对YOLOv8s进行针对性改进和优化，提高模型的检测性能，本文将从以下几个方面来改进YOLOv8s。

4.3.1 注意力机制

注意力机制的灵感来自于人类大脑的工作方式。人类大脑在分析信息时，并非同时关注全部信息，而是会有选择的关注其中最重要的部分，忽略一些不相干或不重要的部分，这个生物学原理启发了研究人员设计算法去模拟类似的“聚焦”能力，计算机视觉中的注意力机制就是模拟人眼的视觉聚焦机制，使得神经网络能够选择性地关注输入图像中的关键部分，忽略其他不重要的部分，从而提高模型的检测性能。目前，在目标检测领域中常见的注意力机制有 SE[68]，CBAM[69]等。

压缩和激励注意力机制(Squeeze and Excitation,SE)[68]通过压缩和激励这两步操作动态的调整不同通道的重要性，结构如图 4-4 所示，第一步操作压缩主要通过对每个通道的特征图进行全局平均池化来获取通道的全局信息，例如图 4-4 中 $H\times W\times C$ 的特征图被压缩成 $1\times1\times C$ 的张量，每个通道的特征图被压缩成了一个值 Z，这个值也就包含了每个通道的上下文信息，公式如(4-1)所示：

$$Z_c=\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}x_{ijc} \tag{4-1}$$

第二步操作激励首先会通过一个全连接层，将原来的 C 个通道压缩到原来的 $1/r$，再通过一个 ReLU 激活函数进行非线性变换，之后再通过全连接层恢复通道数到 C，最后使用 Sigmoid 激活函数将输出映射到[0,1]区间，生成一个通道权重系数 s，公式如(4-2)所示：

$$S_c=\mathrm{Sigmod}\left(FC_2*\mathrm{ReLU}\left(FC_1*Z_c\right)\right) \tag{4-2}$$

最后再重新加权，将每个通道的权重系数与原始特征图中的每个通道进行逐通道相乘，将注意力权重加权到每个通道的特征上，如式(4-3)所示，这样网络可以

自适应的调整每个通道的重要性，从而增强模型对重要特征的关注，提高模型的检测能力。

$$x_c = F\left(u_c, s_c\right) = s_c \cdot u_c \tag{4-3}$$

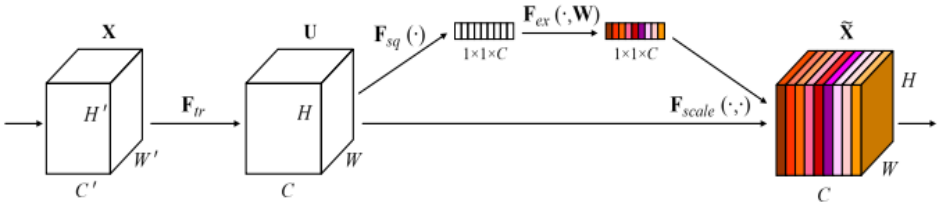


图 4-4 SE 框图[68]

卷积注意力模块(Convolution Block Attention Module,CBAM)[69]是一种集成了通道注意力和空间注意力机制的模块，通过对空间和通道维度的特征进行加权，增强模型对重要特征的关注，提升模型的感知能力。CBAM 模块结构如图 4-5 所示，主要由通道注意力模块(Channel Attention Module,CAM)和空间注意力模块(Spatial Attention module,SAM)两部分组成。

CAM 如图 4-6 所示，首先对输入特征的每个通道进行全局平均池化和最大池化，获取每个通道的全局信息，然后通过一个共享全连接层生成每个通道的注意力权重，再用 Sigmoid 激活函数将权重映射到(0,1)区间内，最后将得到的权重与初始特征图相乘，得到经过注意力权重加权的通道特征图。

SAM 如图 4-7 所示，先对输入特征图沿通道方向分别进行最大池化和平均池化，然后将池化后的特征沿着通道维度拼接在一起，再通过一个卷积层生成空间注意力权重，同样对权重应用 Sigmoid 激活函数，将权重限制到 0 到 1 之间，最后将得到的权重应用到原始特征图上。

通道注意力模块可以学习到每个通道的重要性，可以增强重要特征通道的响应，空间注意力模块可以学习到特征图中每个空间位置的重要性，从而突出重要区域，CBAM 将通道和空间两个注意力模块串联到一起，动态调整特征图的通道权重和空间权重，帮助模型聚焦重要信息。

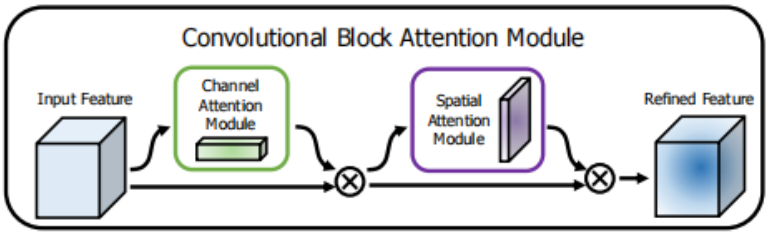


图 4-5 CBAM 框图[69]

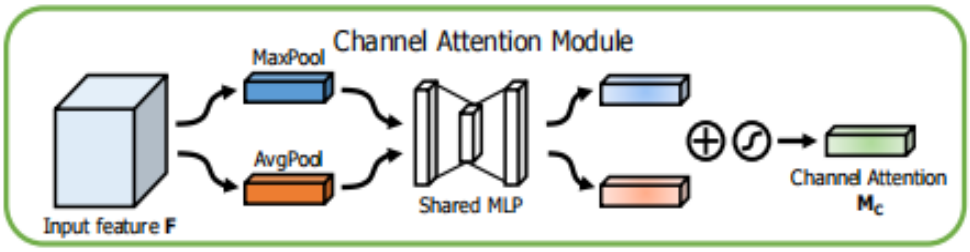


图 4-6 CAM 框图[69]

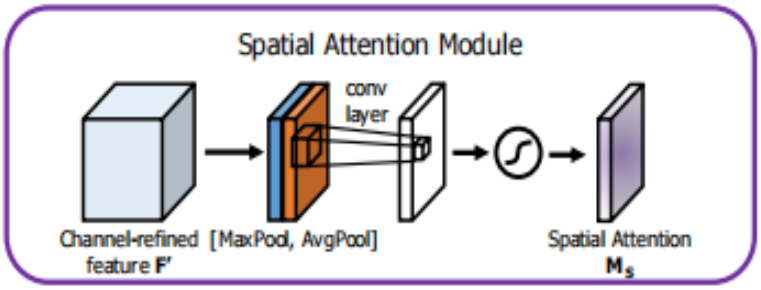


图 4-7 SAM 框图[69]

分离和增强注意力模块(Separated and Enhancement Attention Module,SEAM)首次在人脸检测器 YOLO-FaceV2[70]中提出，YOLO-FaceV2 是在 YOLOv5 框架基础之上进行优化和改进，专门针对密集人脸和遮挡人脸的检测器，该检测器提出了 SEAM 模块补偿脸部因被遮挡降低的响应损失，同时增强没有被遮挡脸部的响应来改进遮挡问题。本文的辣椒识别与人脸检测类似，也存在着大量密集遮挡的情况，同样可以将 SEAM 注意力机制用到 YOLOv8s 中，提高模型的辣椒检测性能。

SEAM 模块结构如图 4-8 所示，图中左侧是 SEAM 模块的整体架构，输入首先经过三个不同尺寸的通道空间混合模块(Channel and Spatial Mixing Module,CSMM)，CSMM 模块结构如图 4-8 右侧所示，先对输入的块进行嵌入，然后通过 GeLU 激活函数进行非线性变换，GeLU 是基于高斯误差函数的激活函数，相比常见的 ReLU，GeLU 更加平滑，有更好的泛化能力和收敛性。之后再经过批量归一化，加快训练速度并提高性能，接着再通过深度可分离卷积学习空间维度和通道之间的相关性，同时减少计算量和参数量。不同尺度的 CSMM 模块输出经过平均池化后，再通过两层全连接网络融合每个通道的特征，让模型加强各个通道间的相关性。接着通过指数函数将权从重范围从[0,1]映射到[1,e]，减小位置误差的影响。最后将 SEAM 输出的注意力权重与原始的输入特征图相乘。上述步骤能让模型学习到遮挡部位和未被遮挡部位之间的关系，从而补偿由于被遮挡而引起的信息损失，让模型有效的处理遮挡问题。

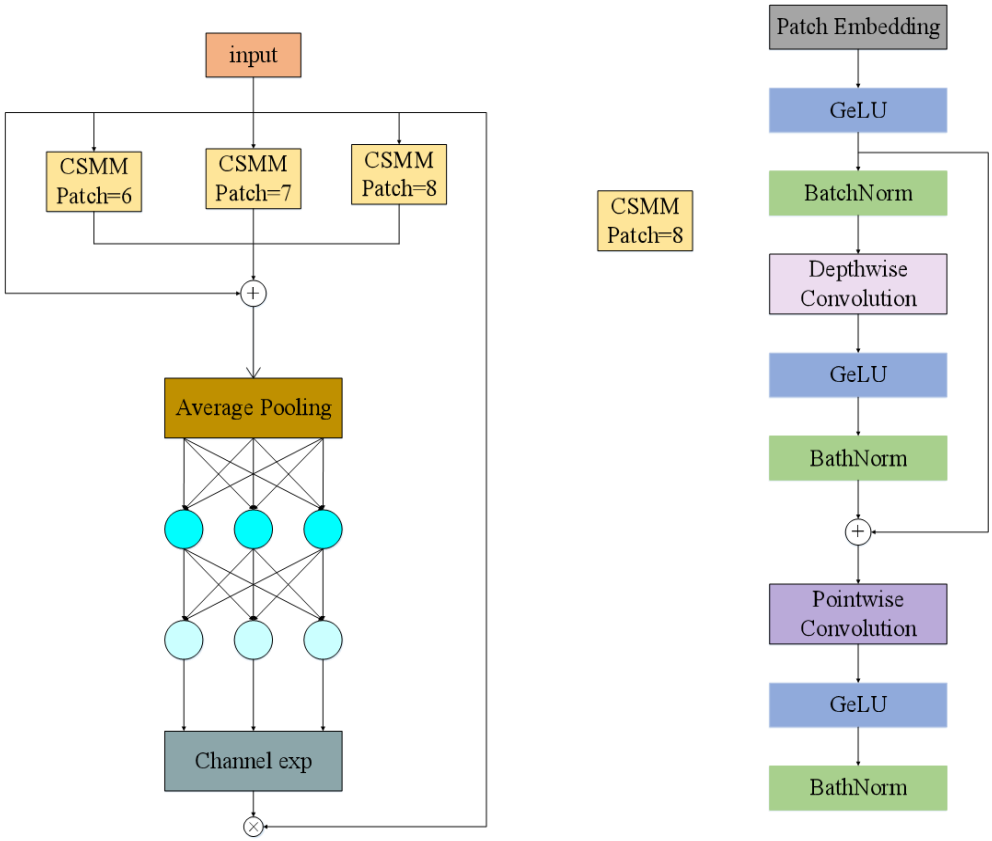


图 4-8 SEAM 和 CSMM 框图

4.3.2 损失函数优化

YOLOv8s 中使用的分类损失函数是 BCE 损失函数，BCE 简单高效，在许多任务中表现良好，但是在类别不均衡的情况下，BCE 容易对简单样本过拟合，忽略了对困难样本分类的学习。本文的辣椒数据集存在者大量密集重叠的目标，这类样本是模型往往难以准确分类和预测的困难样本，对模型的训练影响较大。因此本文使用 Varifocal Loss 函数[72]替代 BCE，Varifocal Loss 是在 Focal Loss[71]改进而来，能将分类任务和回归任务结合起来，让模型更关注高质量的正样本，能有效提高模型的检测性能。

Focal Loss是在交叉熵损失函数的基础上修改得到的，其数学表达式如式(4-4)所示：

$$FL(p_t)=\begin{cases}-\alpha(1-p_t)^{\gamma}\,log(p_t) & y=1\\ -(1-\alpha)\,p_t^{\gamma}log(1-p_t) & y=0\end{cases} \tag{4-4}$$

式中 p_t 是模型对样本的预测概率；α 是权重因子，用来平衡样本类别不平衡的影响，在计算正样本时，权重因子为 α，在计算负样本时，权重因子为 1-α。对于易分类样本，正样本模型预测概率值 p_t 很大，调制因子$(1-p_t)^{\gamma}$的值接近 0，样本损失

很小，负样本模型预测概率值 p_t 很小，调制因子 p_t^γ 的值同样接近 0，样本损失也很小，而对于困难样本，正样本模型预测概率 p_t 值较小，调制因子$(1-p_t)^\gamma$ 的值较大，样本损失也较大，负样本模型预测概率 p_t 值较大，调制因子 p_t^γ 也较大，样本损失也较大。

Focal Loss 能够有效提高困难样本损失贡献，抑制易分类样本的损失贡献。γ 是聚焦参数，该参数平滑地调整了易分类样本权重降低的速率，在 $\gamma=0$ 时，Focal Loss 退化为交叉损失函数，随着 γ 值的不断增大，调节因子的影响也会增加。

Varifocal Loss 的基本公式如下式所示：

$$\mathrm{VFL}(p,q)=\begin{cases}-q\left(q\log(p)+(1-q)\log(1-p)\right) & q>0\\ -\alpha p^{\gamma}log(1-p) & q=0\end{cases} \tag{4-5}$$

式中 p 为模型预测的 IoU 感知分类评分；q 是预测的边界框与真是框的 IoU 值，对于正样本 q>0，负样本 q=0，Varifocal Loss 将分类置信度分数与定位任务中的 IoU 值结合到一起，根据 IoU 的大小动态调整正负样本的权重，提高分类和回归的精度。Varifocal Loss 与 Focal loss 相比，用 p^γ 因子只抑制了负样本(q=0)对损失的贡献，而没有抑制正样本的损失，缓解了密集检测中负样本过多的问题，并且正样本的 q 值越大，它在损失中的贡献也就越大。Varifocal Loss 实现了更高效的样本加权机制，在密集重叠的辣椒检测任务中，能有效提高模型性能。

4.3.3 NMS 优化

在第二章中介绍过目标检测中的 NMS 算法，YOLOv8 的后处理中也是使用的该算法，其能在诸多预测框中，去除多余的预测框，保留每个目标的最优预测框。

但在辣椒数据集中存在很多大量重叠较大的目标，如图 4-9 所示。图中红色和黄色是第一个辣椒的预测框，蓝色是被遮挡辣椒的预测框，在 NMS 算法的处理过程中，首先会选择置信度为 0.95 的红色预测框作为最高分框，然后计算剩下的框与它的 IoU 值，如果不存在重叠较大的目标，那其余目标的预测框与最高分框的 IoU 值会很小，不会超过阈值，也就保留了其余目标的预测框，而当如图中所示辣椒重叠很大时，被遮挡辣椒的蓝色预测框与最高分框的 IoU 值同样会远远超出阈值，该框会和黄框一样被当作最高分框的冗余框删除，那被遮挡的辣椒也就无法被检测到。

为了缓解 NMS 在密集重叠场景中漏检情况，本文使用 Soft-NMS 算法[73]替代 NMS，Soft-NMS 是对 NMS 算法的改进，在传统的 NMS 算法中，会直接删除所有与最高分框 IoU 超过阈值的预测框，而 Soft-NMS 的核心思想是，并不直接将超过

阈值的预测框删除，而是通过衰减函数减少其置信度分数，IoU 值越高，减少的置信度分数越多，常用的衰减函数有两种，一个是高斯函数，另一个是线性函数，公式如下两式所示，本文所使用的衰减函数是高斯函数。

$$s = se^{-\frac{\mathrm{IoU}^2}{\sigma}} \tag{4-6}$$

$$s = \begin{cases} s & \mathrm{IoU} < N_t \\ s(1-\mathrm{IoU}) & \mathrm{IoU} \geq N_t \end{cases} \tag{4-7}$$

上式中 s 是置信度分数；σ 是控制衰减率的参数。之后再重复上述步骤，最终将置信度分数小于阈值的框删除。Soft-NMS 能在密集重叠的场景中，保留更多的检测框，有效减少漏检的情况。

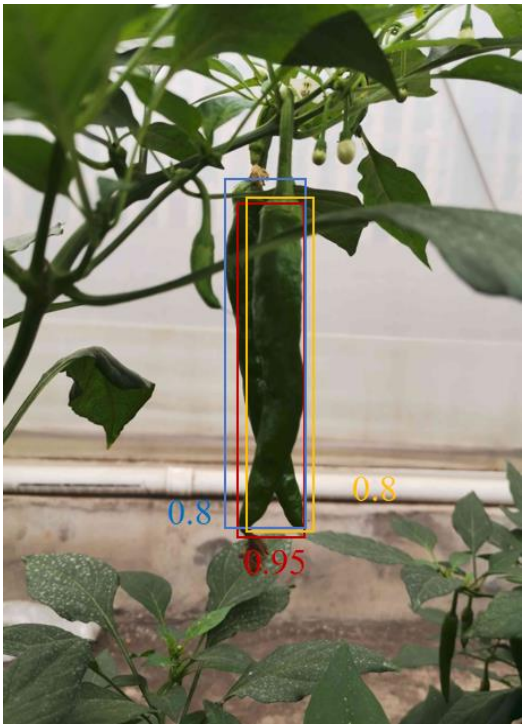


图 4-9 重叠较大的辣椒示意图

综上所述，本文主要对 YOLOv8s 模型做了如下优化，在 NECK 部分加入 SEAM 注意力机制，如图 4-10 中的红色部分所示，增强未被遮挡辣椒的响应，同时补偿辣椒因遮挡降低的响应，从而提升模型对遮挡辣椒的检测能力。其次，用 Varifocal 损失函数代替交叉熵损失函数，实现了更高效的样本加权机制。最后在后处理过程中，使用 Soft-NMS 算法替换 NMS 算法，让模型能在密集重叠的辣椒场景中，保留更多的高质量检测框，减少漏检的情形。最终改进后的 YOLOv8s 模型如图 4-10 所示。

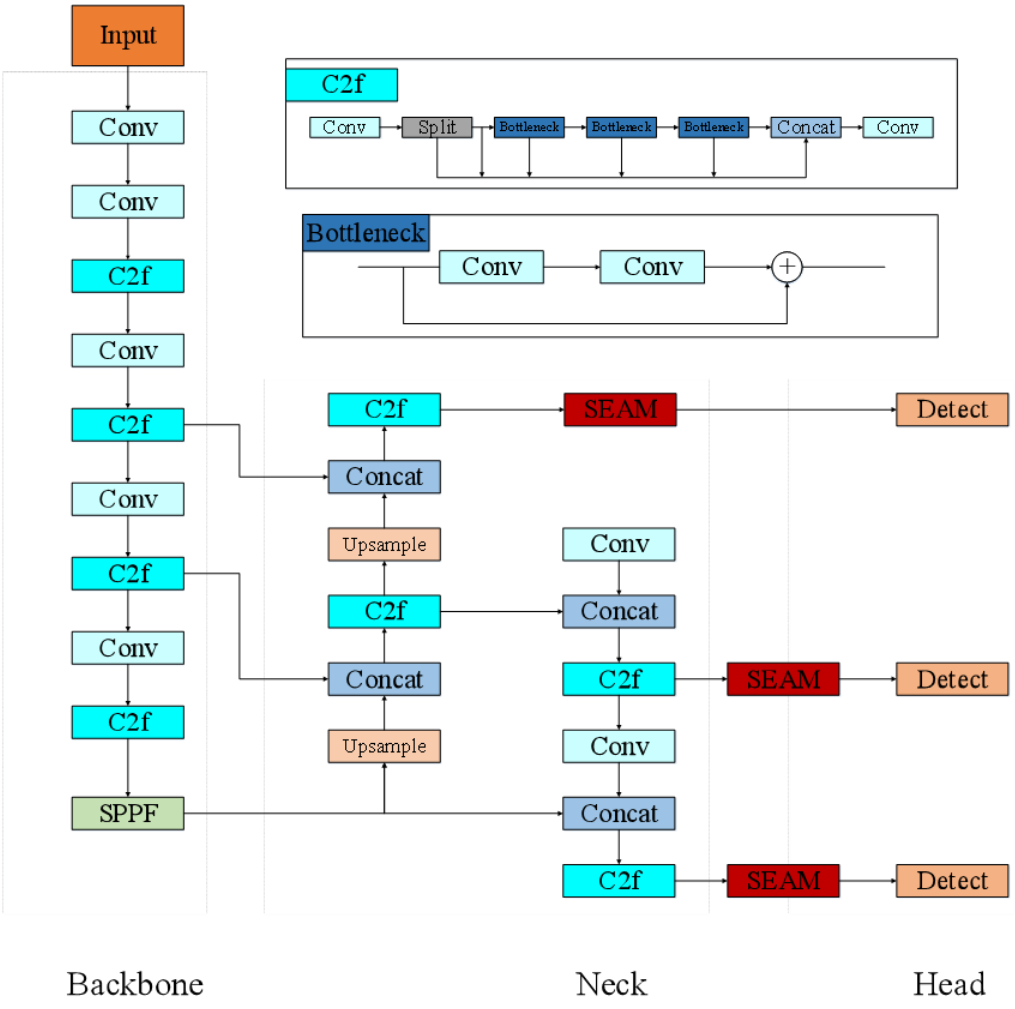


图 4-10 改进后的的 YOLOv8s 网络结构图

4.4 实验结果和分析

4.4.1 实验环境与参数配置

深度学习对 GPU 性能要求较高，GPU 性能越强，模型推理的速度越快，本文之后所有的实验都是在同一计算机平台下完成，具体的计算机硬件和软件参数配置如所表 4-2 所示。

表 4-2 软硬件参数配置

名称	配置
GPU	NVidia GeForce GTX 4090 24GB
CPU	lntel(R) Xeon(R) Gold 6133 CPU @2.50GHz2.50 GHz
操作系统	Windows 11

表 4-2 软硬件参数配置（续）

名称	配置
语言	Python3.8
深度学习框架	Pytorch1.7
CUDA	11.6
编辑器	Pycharm

本文的数据集按照训练集，验证集及测试集 8：1：1 的比例随机分配，本文使用的相关训练参数如表 4-3 所示。

表 4-3 训练参数配置表

训练参数	
Batchsize	32
训练轮数	200
初始学习率	0.01
优化器	SGD
初始动量	0.937

4.4.2 消融实验

4.4.2.1 注意力机制

本节实验旨在验证网络融合注意力机制后对模型检测性能的提升，实验以 YOLOv8s 基础网络为基准，在模型相同位置处分别引入 SE，CBAM，以及本文使用的 SEAM 注意力机制进行实验，最终实验结果如表 4-4 所示：

表 4-4 不同注意力机制的实验结果

模型	P(%)	R(%)	mAP(%)	参数量(M)
YOLOv8s	81.2	77.4	84.1	11.14
YOLOv8s+SE	81.8	77.5	85.4	11.14
YOLOv8s+CBAM	82.7	78.2	86.2	11.48
YOLOv8s+SEAM	85.8	77.7	87.2	11.54

从表中可以看出，在原模型YOLOv8s中加入注意力机制后，模型的检测性能都得到了有效的提升。在准确率方面，未加入任何注意力机制的YOLOv8s是 81.2%，引入SE注意力机制后是 81.8%，引入CBAM后是 82.7%，引入SEAM后更是达到了 85.8%，相比初始模型提高了 4.6%。在召回率方面，引入注意力机制后，也都有所提升，SE、CBAM、SEAM，分别提升了 0.1%，0.8%，0.3%，在mAP方面，模型加入SEAM注意力机制提升最高，达到 87.2%，提高了 3.1%，SE达到 85.4%，提高

了 1.3%，CBAM达到 86.2%，提高了 2.1%。并且SEAM相比原模型，并没有引入太多的额外参数量。

综合来看，YOLOv8s 加入三种注意力机制后，性能都得到了提升，说明 SE，CBAM 注意力机制都可以让模型更加关注重要特征，加强重要区域的影响，提高模型的检测能力，而本文辣椒数据集中存在大量遮挡重叠的情况，因此 SE，CBAM 注意力机制对模型检测性能的提升较小。SEAM 注意力机制则可以让模型学习到遮挡辣椒和未被遮挡辣椒之间的关系，补偿了辣椒因补遮挡降低的响应损失，同时增强未被遮挡辣椒的响应。加入 SEAM 后模型的检测性能提升最大，证明了 SEAM 在遮挡目标检测中的有效性。

4.4.2.2 损失函数

本节实验旨在验证 Varifaocal 损失函数对于模型辣椒检测性能的提升，使用 Focal 损失函数和 Varifocal 损失函数代替 BCE 损失函数，最终的实验结果如表 4-5 所示：

表 4-5 不同损失函数的实验结果

损失函数	P(%)	R(%)	mAP(%)
BCE Loss	81.2	77.4	84.1
Focal Loss	82.3	76.4	85.0
Varifocal Loss	82.9	77.5	85.7

由表中可知，使用 Focal 损失函数代替 BCE 损失函数，模型的准确率提高了 1.1%，mAP 提升了 0.9%，使用 Varifocal 损失函数代替 BCE 损失函数，模型的准确率提高了 1.7%，召回率提高了 0.1%，mAP 从 84.1%提升到 85.7%，提高了 1.6%，模型的性能得到了进一步的提升。Varifocal 损失函数在遮挡辣椒中检测精度最高，说明其能让 YOLOv8s 在训练时关注到更多高质量的正样本，并通过 IoU 加权，让模型同时优化分类得分和定位精度，提高了预测框的质量，最终提高模型检测密集遮挡辣椒的能力。

4.4.2.3 更换 NMS

本节实验是为了验证 Soft-NMS 对密集遮挡辣椒检测性能是否有提升，分别使用 NMS 和 Soft-NMS 算法进行实验，NMS 和 Soft-NMS 的实验结果对比如表 4-6 所示，从表中可知，在模型后处理中使用 Soft-NMS 代替 NMS，模型准确率从 81.2%提升到 82.0%，提高了 0.8%，召回率从 77.4%提升到 78.4%，提高了 1%，mAP 从

84.1%提升到 86.0%，提高了 1.9%，表明 Soft-NMS 能在密集重叠的辣椒中，保留更多的检测框，有效减少漏检的概率。图 4-11 是 NMS 和 Soft-NMS 的实验结果可视化比较图，图 4-11(a)是 NMS 的检测结果，对于遮挡重叠的辣椒，漏检了两个，图 4-11(b)是 Soft-NMS 的检测结果，没有出现漏检的情况，也进一步证明了 Soft-NMS 可以提升模型在密集遮挡场景下的检测性能

表 4-6 NMS 和 Soft-NMS 的实验结果

模型	P(%)	R(%)	mAP(%)
NMS	81.2	77.4	84.1
Soft-NMS	82.0	78.4	86.0

(a) (b)

图 4-11 NMS 和 Soft-NMS 的检测结果。(a)NMS；(b)Soft-NMS

为了验证本文提出的模型在密集遮挡任务中的检测性能，将测试集中重叠遮挡较多的图片挑选出来组成遮挡测试集，用原 YOLOv8s 模型和本文提出的改进模型分别进行检测，结果如表 4-7 所示。

本文提出的改进模型相比原 YOLOv8s 准确率从 75.3%提升到 83.1%,，提高了 7.8%，召回率从 70.2%提升到 74.2%，提高了 4%，mAP 从 78.6%提升到 83.8%，提高了 5.2%，本文提出的模型在密集遮挡的辣椒检测中有更好的表现，说明本文对 YOLOv8s 做出的改进是有效的。

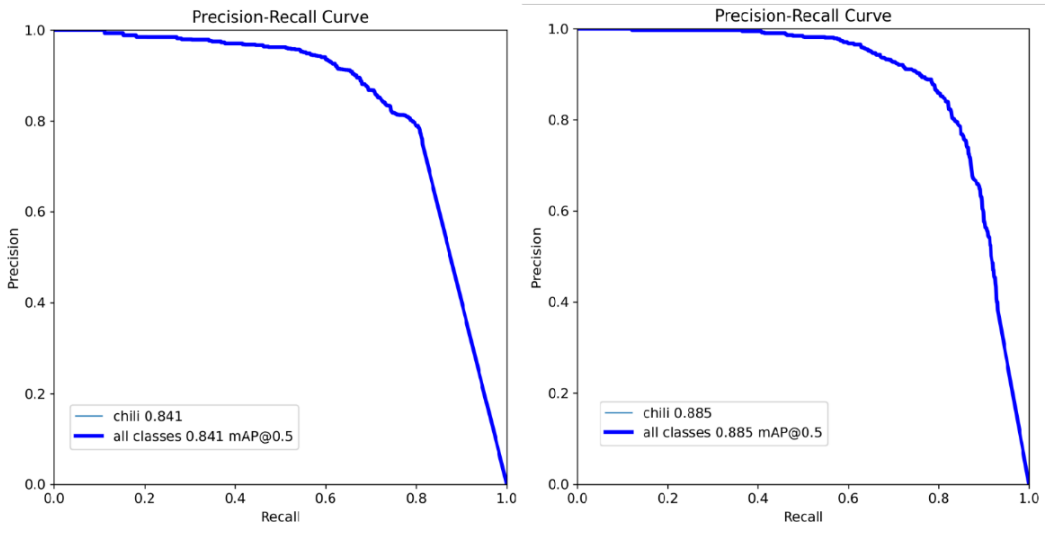
表 4-7 遮挡测试集中的检测结果比较

模型	P(%)	R(%)	mAP(%)
YOLOv8s	75.3	70.2	78.6
Ours	83.1	74.2	83.8

在完整测试集上的检测结果比较如表 4-8 和所示。完整测试集上本文提出的改进模型相比原 YOLOv8s 准确率从 81.2%提升到 87.2%,召回率从 77.4%提升到 79.1%，提高了 1.7%，mAP 从 84.1%提升到 88.5%，提高了 4.4%，模型参数量仅增加了 0.4M，FPS 虽然从 62.9 下降到 57.1，但是对于辣椒采摘机器人来说，这个 FPS 完全可以满足采摘要求。

表 4-8 完整测试集中的检测结果比较

模型	P(%)	R(%)	mAP(%)	FPS	Parameters(M)
YOLOv8s	81.2	77.4	84.1	62.9	11.14
Ours	87.2	79.1	88.5	57.1	11.54



(a) (b)

图 4-12 PR 曲线对比结果图。(a)原模型；(a)本文提出的模型

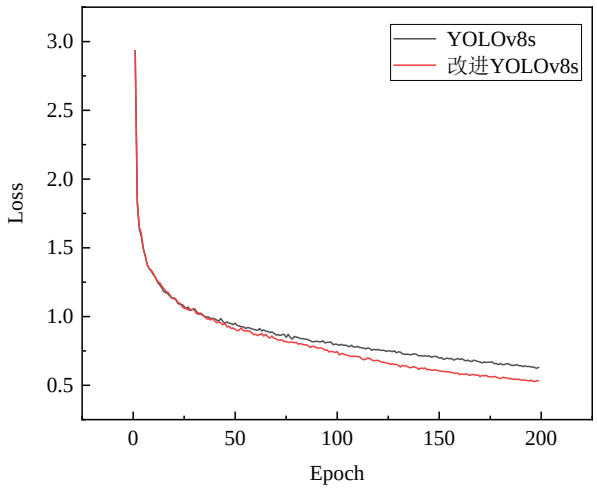


图 4-13 边界框回归损失迭代曲线图

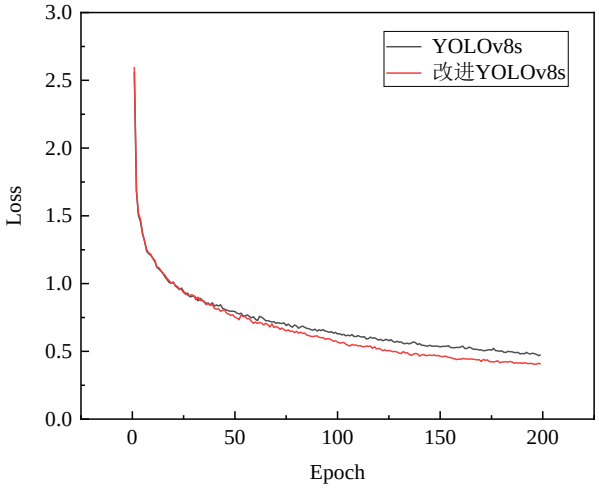


图 4-14 分类损失迭代曲线图

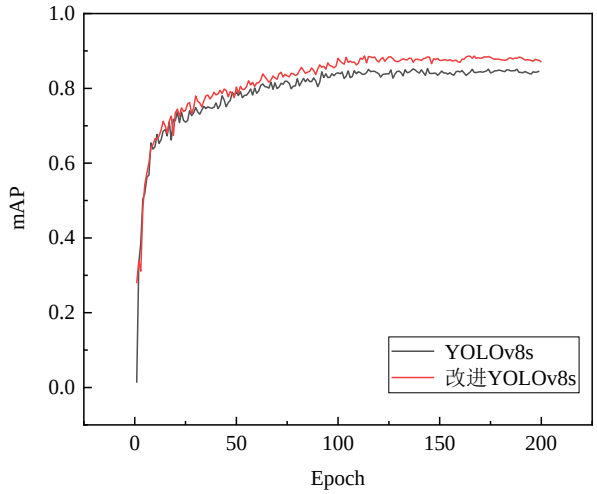


图 4-15 mAP 迭代曲线图

图 4-12 是原模型和本文优化后的模型在完整测试集中的精度召回(PR)曲线比较图，PR 曲线与 xy 轴围成的面积越大，代表模型的效果越好，从图中可看出本文提出模型的 PR 曲线与坐标轴围成的面积明显大于原模型，进一步验证了本文提出模型的优越性。图 4-13 和图 4-14 分别是原 YOLOv8s 和改进后模型的边界框回归损失和分类损失对比图，黑色是曲线是 YOLOv8s 原模型，红色曲线是改进后的 YOLOv8s 模型，从图中可以看出，两个模型的边界框回归损失和分类损失都随着 Epoch 的增加而下降，并最终趋于稳定，本文改进后的模型相比原模型的损失更低，说明本文改进后的模型特征提取能力，拟合能力更强。图 4-15 是原模型和本文优化后模型的 mAP 随训练轮次的变化，从图中可清晰的看到，两个模型的 mAP 都随着训练轮次的增加而增加，并在 125 轮之后趋于平稳，本文优化后的模型 mAP 更高，表明它的检测性能更好，对于密集遮挡辣椒的识别更加准确。

综上所述，本文对YOLOv8s模型进行的各项改进都是有效的，在没有引入过多参数量和牺牲太多检测速度的情况下，有效提高了YOLOv8s对密集遮挡辣椒的检测精度。

4.4.3 对比实验

为了体现本文提出的模型在密集重叠辣椒检测中的优越性，在同样的数据集和实验环境下，将本文提出的模型与近几年的目标检测模型进行对比，对比结果如表 4-9 所示：

表 4-9 不同目标检测模型在辣椒数据集上的检测结果

模型	P(%)	R(%)	mAP(%)	FPS	参数量(M)
YOLOv5s	76.2	77.0	83.0	58.8	7.02
YOLOv7-tiny	82.4	71.9	83.5	69.9	6.01
YOLOv8s	81.2	77.4	84.1	62.9	11.14
YOLOv9	85.6	74.4	87.5	63.5	9.74
YOLOv10s	84.3	78.9	87.3	91.6	8.06
Ours	87.2	79.1	88.5	57.1	11.54

从表中可知，本文提出的模型相较 YOLOv8 及其之前的 YOLOv5 和 YOLOv7 模型，mAP 分别提高了 5.5%，5%，4.4%，检测精度有较大的提升，与 YOLOv9 和 YOLOv10 模型来说，mAP 也提高了 1%和 0.8%，检测精度也有不小的提升，YOLOv9 和 YOLOv10 作为 2024 年推出的 YOLO 最新系列的模型，也进一步验证了本文模型的优越性。虽然 FPS 相比较其他的模型略低一些，但在辣椒采摘系统中，足以满足实时采摘的需求。

4.4.4 辣椒检测结果分析

本节用本文提出的辣椒检测模型与原 YOLOv8s 模型检测不同情况下的辣椒，进一步验证本文所提出的改进模型的优越性。辣椒检测结果如图 4-16 所示，图 4-16(a)是所检测辣椒的标签，图 4-16(b)是原 YOLOv8s 的检测结果，图 4-16(c)是本文提出的优化模型的检测结果。

图 4-16(a)第一行中的辣椒数量较少，不存在太多枝叶遮挡和辣椒互相遮挡的情况，虽然辣椒与背景比较类似，但总体特征还是比较明显，YOLOv8s 模型和本文提出的模型都可以很准确的检测出所有的辣椒，如图 4-16(b)和(c)第一行所示。图 4-16(a)中的第二行辣椒重叠遮挡较多，第三行中的辣椒数量较多，且大多被枝叶遮挡。YOLOv8s 模型第二张辣椒的检测结果中出现了较多的漏检，第三张辣椒

中出现了较多的误检。而本文提出的改进模型的检测结果虽然也存在一些漏检和误检的情况，但数量远远少于 YOLOv8s 模型，进一步证明了本文提出的改进模型在辣椒密集重叠，枝叶遮挡场景中的检测能力。

(a) (b) (c)

图 4-16 辣椒检测效果对比图。(a)标签；(b)YOLOv8s；改进 YOLOv8s

4.5 本章小结

本章主要内容是提出了一个在密集遮挡的辣椒检测场景中具有高精度的检测模型。首先介绍了辣椒数据集的制作，接着介绍了基线模型 YOLOv8s，为了提升其在密集遮挡辣椒中的检测性能，在其基础上作了以下改进：（1）在 YOLOv8 模型中引入 SEAM 注意力机制，补偿辣椒因遮挡降低的响应损失，增强未被遮挡辣椒的响应来提升模型对遮挡辣椒的检测性能，使用 SEAM 注意力机制后，模型 mAP 提高了 3.1%；（2）在 YOLOv8 中使用 Varifocal 损失函数，动态融合定位质量信息到分类损失中，实现了更高效的样本加权机制，mAP 提高了 1.6%；（3）在 YOLOv8 的后处理过程中使用 Soft-NMS，减少辣椒因过度重叠而漏检的概率，mAP 提高了 1.9%。最后对各个改进模块进行消融实验，并将最终改进的模型与其他目标检测模型进行比较，本文提出的模型具有最高的检测精度。

第五章 辣椒定位与采摘实验

使用上一章中改进的 YOLOv8s 模型能够成功检测到成熟辣椒，可以获取辣椒在图像中的二维坐标位置，机械臂完成采摘还需要获取辣椒在机械臂坐标系下的坐标信息。本章首先对相机进行标定，获取相机的内参矩阵，再进行手眼标定得到手眼矩阵，将辣椒在相机坐标系下的坐标转换到机械臂下的坐标，并对偏转较大的辣椒进行简单的姿态分析，最后通过采摘实验验证本文提出的辣椒采摘系统的有效性。

5.1 深度相机标定

Realsense d415 深度相机在出厂时都会对相机内参进行标定，但在长期的使用和搬动过程中，难免发生碰撞导致内参变化，为了提高辣椒定位的精度，因此有必要重新标定相机的内参。

5.1.1 相机的成像模型

相机的参数标定主要提高通过已知物理尺寸的标定板推导相机的内参和外参[74]。相机的成像模型涉及四个坐标系：世界，相机，图像和像素坐标系，相机成像模型如图 5-1 所示。

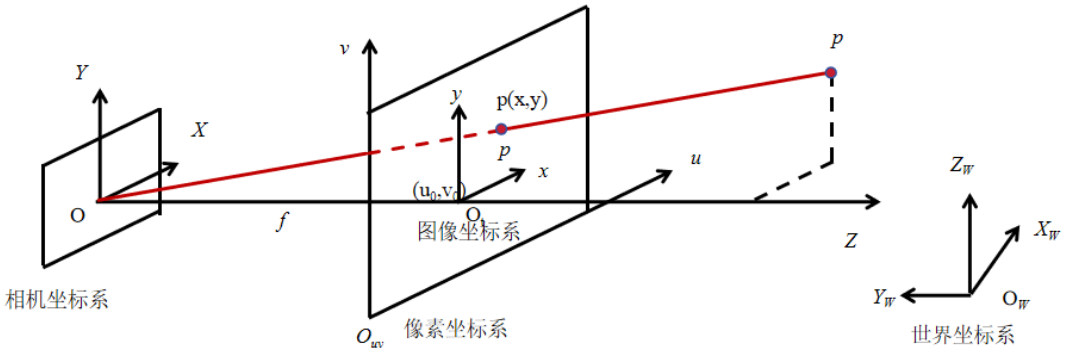


图 5-1 相机成像模型

相机坐标系是以相机的光心为原点，沿着相机的光轴方向定义的坐标系，如图中的 *XOY* 坐标系所示，*f* 是相机的焦距像素坐标系是图像中实际像素的坐标系，如图中的 *uOv* 坐标系所示，(*u*,*v*)表示图像中某个像素的位置。图像坐标系是相机成像后，得到的二维图像坐标系，也就是图中的 *xOy* 坐标系。图像坐标系 *x*,*y* 两个坐

标轴与像素坐标系两个坐标轴 u,v 平行，图像坐标系原点在像素坐标系下的坐标是(u_0,v_0)，世界坐标系是整个场景的坐标系，描述场景中物体的位置和方向，如图中的 $X_WO_WY_W$ 坐标系所示。相机的外参描述了世界坐标系和相机坐标系之间的关系，相机的内参描述了相机坐标系和像素坐标系的关系，内参主要是相机的焦距，畸变参数和主点，确定了相机的外参和内参就可以实现像素坐标系到世界坐标系的转换。

假设存在一点 P，在世界坐标系中的坐标是(X_W,Y_W,Z_W)，在相机坐标系中的坐标是(X,Y,Z)，P 在图像中的成像点是 p，p 在图像坐标系中的坐标是(x,y)，在像素坐标系下的坐标是(u,v)。根据像素坐标系和图像坐标系的关系，(x,y)和(u,v)有如下关系：

$$u=\frac{x}{dx}+u_0 \tag{5-1}$$

$$v=\frac{y}{dy}+v_0 \tag{5-2}$$

式(5-1)和(5-2)中的dx，dy分别表示每个像素点的实际大小，(u_0,v_0)是图像坐标系的原点在像素坐标系下的坐标。将上面两式改写成齐次坐标，如式(5-3)所示

$$\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}\frac{1}{dx} & 0 & u_0\\ 0 & \frac{1}{dy} & v_0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix} \tag{5-3}$$

(x,y)和(X,Y,Z)可以根据图像坐标系和相机坐标系的相似关系得到，如式(5-4)和(5-5)所示：

$$x=f\frac{X}{Z} \tag{5-4}$$

$$y=f\frac{Y}{Z} \tag{5-5}$$

将上两式改写成齐次形式，如式(5-6)所示：

$$Z\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}f_x & 0 & 0 & 0\\ 0 & f_y & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix} \tag{5-6}$$

世界坐标系和相机坐标系都是三维坐标系，(X,Y,Z)和(X_W,Y_W,Z_W)的关系可由刚体变换得到，如式(5-7)所示

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{5-7}$$

上式中 R 是旋转矩阵，t 是平移矩阵，将式(5-7)和式(5-6)代入到式(5-3)可得式(5-8):

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{5-8}$$

用 $M1$ 和 $M2$ 表示式(5-3)和(5-6)的变换矩阵：

$$M1M2 = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{f_x}{dx} & 0 & u_0 \\ 0 & \frac{f_y}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5-9}$$

上式就是相机的内参矩阵，只和相机自身相关器件的参数有关，相机标定就是计算相机的内参矩阵。

5.1.2 双目测距原理

本文选择的RealSense D415 深度相机的深度计算方法主要是双目测距原理[75]，双目测距模型如图 5-2 所示。图中P是待测物体。$O1$ 和$O2$ 代表两个相机的光心，两相机间的距离是基线B，P到基线的距离也就是要求的深度Z，f是相机的焦距，$P1$和$P2$ 是待测物体P在两个相机中所成的像，$P1P2$ 之间的距离D可由下式计算得出：

$$D = B - (x1 - x2) \tag{5-10}$$

再由三角行相似原理可得：

$$\frac{B - (x1 - x2)}{B} = \frac{Z - f}{Z} \tag{5-11}$$

移项可得：

$$Z = \frac{fB}{x1 - x2} \tag{5-12}$$

$x1$-$x2$ 是 $P1$ 和 $P2$ 在水平方向上的像素差，也就是视差。焦距 f 和基线 B 可标定相机得到，然后就可以根据式(5-12)计算出深度。

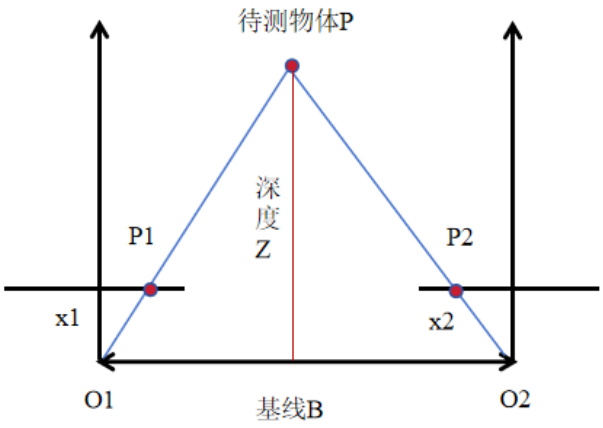


图 5-2 双目测距原理图

5.1.3 RealSense D415 相机标定

本文使用的标定方法是张正友标定法[76]，使用 Matlab 中的 Stereo Camera Calibrator 工具标定相机，标定板选用的是 12×9 黑白棋盘格亚克力板，如图 5-3 所示，每个方格的大小是 6 mm×6 mm，在相机不同角度和距离下拍摄标定板图片，一共采集 26 组左右相机图片。

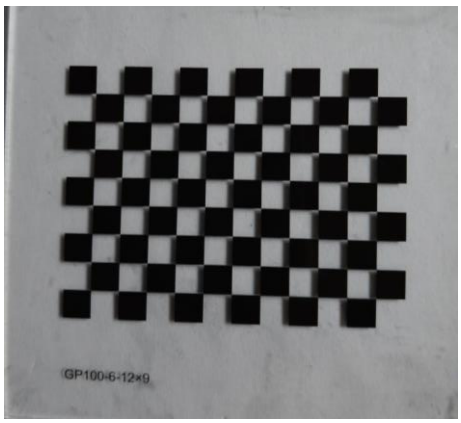


图 5-3 12×9 标定板图片

将图片导入到 Stereo Camera Calibrator 模块中，最终求解得到的 d415 相机参数如表 5-1 所示，主要包括左右相机的内参矩阵，切向畸变和径向畸变参数，左右

相机的相对位置以及基线等。相机的标定误差如图 5-4 示，从图中可看出，大致在 0.05 到 0.25 个像素之间，平均标定误差是 0.15 像素，一般认为，平均标定误差在 0.5 像素以下，标定数据即可使用，本文误差只有 0.15 像素，标定结果比较精准。图 5-5 是相机拍摄时，标定板和相机的相对位置关系。

表 5-1 D415 相机标定参数结果

参数	左相机	右相机
内参矩阵	$\begin{bmatrix} 920.42 & 0.74 & 631.31 \\ 0 & 919.78 & 365.79 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 923.40 & -0.04 & 623.73 \\ 0 & 923.37 & 367.27 \\ 0 & 0 & 1 \end{bmatrix}$
径向畸变	$[-0.0055 \quad 0.1683 \quad -0.2935]$	$[-0.1110 \quad 1.0994 \quad -2.9747]$
切向畸变	$[-0.0011 \quad -0.0018]$	$[-0.0006 \quad -0.0012]$
旋转矩阵	$\begin{bmatrix} 1 & -0.0004 & -0.0087 \\ 0.0004 & 1 & 0.0016 \\ 0.0087 & -0.0016 & 1 \end{bmatrix}$	
平移矩阵	$[-55.05 \quad 0.02 \quad -0.30]$	
基线	55.05	

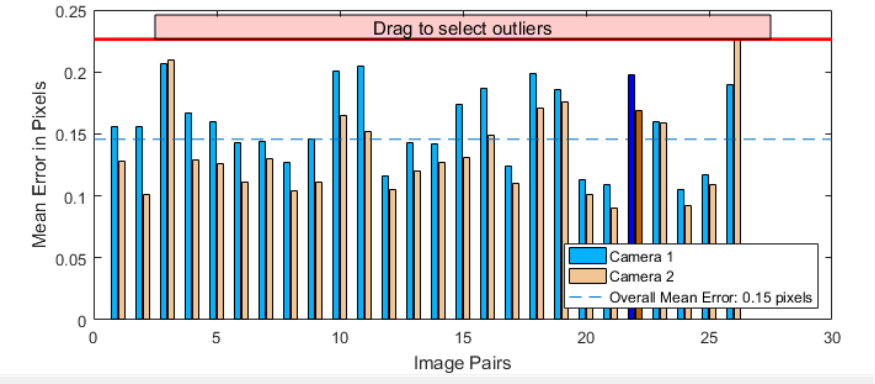


图 5-4 相机标定误差

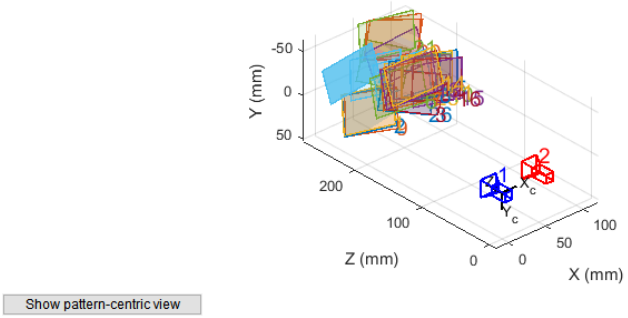


图 5-5 标定板与相机的相对位置关系

5.2 手眼标定

在上一小节中已经求得了 D415 相机的相关参数矩阵，根据上述矩阵可以求得拍摄的辣椒在相机坐标系下的三维坐标，为了让机械臂能运动到辣椒位置完成采摘，还需要进行手眼标定[77]，求得机械臂坐标系与相机坐标系之间的关系，这样就可以计算出辣椒在机械臂坐标系下的坐标。手眼标定主要有两种方式，第一种是手在眼上，如图 5-6(a)所示，也就是将相机固定到机械臂末端上，相机不会跟随机械臂移动，这种方式视野范围限制较大，很容易出现遗漏的情况，第二种是手在眼外，如图 5-6(b)所示，也就是将相机固定在机械臂外部，相机不会随着机械臂移动而移动，这种方法视野广泛，但机械臂可能会遮挡相机。综合考虑辣椒采摘的实际应用场景，本文选择了眼在手外的安装和标定方法。将深度相机固定在机械臂一侧，如图 5-7 所示。

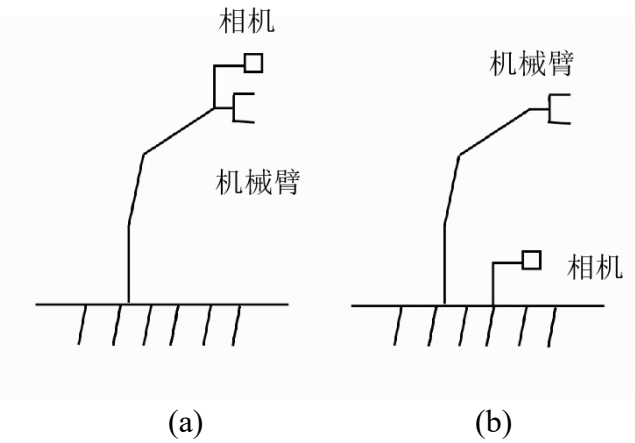


图 5-6 手眼标定类型。(a)眼在手上；(b)眼在手外

图 5-7 眼在手外安装实物图

眼在手外标定主要是获取相机坐标系和机械臂基座系之间的位置关系，标定原理如图 5-8 所示，在机械臂的末端固定标定板，图中 base 是机械臂基座，tool 是机械臂末端，cal 是标定板，camera 是相机，在某一姿态下，对于标定板上的一点 P_1，经过标定板在相机坐标系下的变换矩阵 T_1，相机在机械臂基坐标系下的变换

矩阵 X，和机械臂基坐标系在机械臂末端坐标系下的变换矩阵 T_2，可以得到 P_1 在机械臂末端坐标系下的坐标 P_2，也就是式(5-13)：

$$T_2 X T_1 P_1 = P_2 \tag{5-13}$$

改变机械臂的位姿，由于标点板固定在机械臂末端上，因此对与点 P_1 同样可以得到一组上述公式：

$$T_2' X T_1' P_1 = P_2 \tag{5-14}$$

联立式(5-13)和(5-14)并移项可得：

$$T_2'^{-1} T_2 X = X T_1 T_1'^{-1} \tag{5-15}$$

式(5-15)可简化为：

$$AX = XB \tag{5-16}$$

式中 $A=T_2'^{-1}T_2$，$B=T_1'T_1^{-1}$，A 可由机械臂的示教器中读取，B 可由相机标定的外参获取，多次改变机械臂的位姿，即可建立方程组，最终可以解出 X，也就是深度相机在机械臂基坐标系下的位置和姿态。

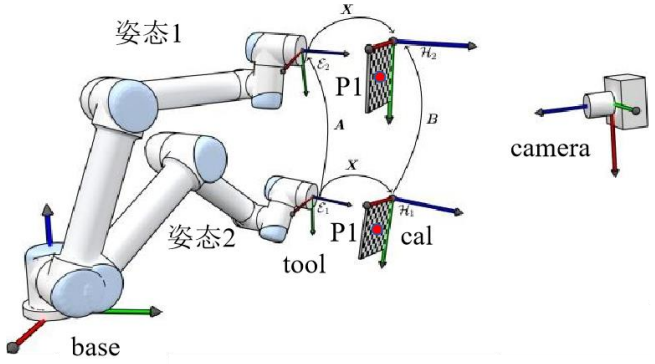


图 5-8 眼在手外标定原理图

本文最终求得的手眼矩阵如式(5-17)所示：

$$X = \begin{bmatrix} -0.0609 & -0.0078 & -0.9981 & 7.9977 \\ -0.9979 & 0.0202 & -0.0608 & -249.7257 \\ -0.0197 & -0.9998 & -0.0090 & 134.4114 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5-17}$$

5.3 定位实验

在前两个小节中，对相机进行了双目标定和手眼标定，由此可以求得相机拍摄图像中一点在机械臂基坐标系下的位置，为了验证定位系统的定位效果，本节进行

了定位实验。由于很难测量出空间中一点在机械臂基坐标系下的三维坐标位置，本文选择机械臂六轴 TCP 点，TCP 点在基坐标系下的三维坐标可由机械臂的示教器读出，查阅机械臂的产品规格手册，机械臂的绝对精度平均值是 0.14 mm，最大值是 0.45 mm，也就是示教器中 TCP 点的三维坐标的平均误差大概是 0.14 mm，这个误差相比相机和标定的误差已经足够小，因此可以将示教器中读出的三维坐标作为真实值，同时使用相机拍摄机械臂六轴的 TCP 点，再通过手眼矩阵即可计算出 TCP 点在基坐标系下的位置。改变机械臂的位置，空间不同点的计算值和真实值的比较如表 5-2 所示

表 5-2 真实值和定位计算值的比较表

序号	真实值(mm)			计算值(mm)			综合误差
	Z	X	Y	Z	X	Y	
1	459.98	-236.43	190.22	457.10	-237.74	193.16	4.32
2	501.41	-240.26	185.69	498.85	-242.33	187.76	3.89
3	520.13	-206.85	81.93	518.23	-204.67	82.41	2.93
4	552.00	-278.48	168.77	549.39	-281.66	169.41	4.16
5	574.50	-216.19	194.68	573.54	-218.33	197.42	3.61
6	604.65	-160.79	178.69	604.01	-159.97	182.06	3.53
7	626.94	-198.56	203.54	625.83	-200.02	206.83	3.77
8	645.01	-205.10	188.87	647.31	-206.96	191.02	3.66
9	674.00	-166.72	177.81	675.69	-167.06	180.23	2.97
10	693.05	-145.77	140.81	694.72	-145.31	142.87	2.69
11	716.46	-150.23	172.75	717.71	-150.93	174.76	2.47
12	741.35	-154.97	201.44	742.61	-157.59	204.10	3.94
13	760.72	-141.72	265.75	760.99	-145.26	270.01	5.55
14	791.33	-138.30	319.50	791.72	-142.03	324.18	6.00
15	809.86	-122.91	289.09	808.40	-126.91	293.46	6.10

从上表中可以得知，计算值和真实值在 XYZ 三个方向上的平均误差分别是：2.07 mm，2.68 mm，1.53 mm，综合误差的平均值是 3.97 mm。考虑到本文采摘的成熟辣椒长度在 80 mm-200 mm 之间，直径在 15 mm-30 mm 之间，使用的二指电动夹爪张开的口径达到 140 mm，辣椒采摘的容错空间远远大于系统的综合误差。实验结果表明本文建立的视觉定位系统具有较高的定位精度，远远满足成熟辣椒采摘的需求。

5.4 辣椒姿态分析

辣椒在经过第四章中的辣椒识别模型检测后，可以获得已识别辣椒在图像中的二位坐标，在再通过深度相机，即可获取辣椒的深度，最后经过手眼变换矩阵，就能计算出辣椒在机械臂坐标系下相应的坐标，末端执行器也可以运动到识别辣椒处开始采摘。大部分辣椒由于重力的影响，向下生长，整体呈细长的圆柱形，因此本文将二指末端执行器水平放置作为初始状态，如第三章中的图 3-2 所示，就可以采摘到大部分向下生长的辣椒。但是还有部分辣椒由于各种原因，会向左右，前后偏转，辣椒的前后偏转，不会影响到二指末端执行器的采摘，如果辣椒的左右偏转较大，二指末端执行器仍是在初始状态采摘，就很有可能无法采摘，强行夹取则会损伤辣椒，机械臂六轴需要根据辣椒偏转的角度旋转相应的角度，来带动夹爪旋转，才能更好的完成采摘，因此有必要对这部分辣椒进行简单的姿态分析，计算辣椒与竖直方向的夹角。

本文在第四章辣椒识别模型检测结果的基础上，基于传统图像处理使用 python 中的 OpenCV 库计算辣椒相对竖直直线的偏转角度。为了更方便的检测辣椒的姿态，可以根据边界框的坐标，并适当向外扩充一些，将识别辣椒裁剪出来，如图 5-9(a)所示。辣椒图像都是彩色图像，具有 RGB 三个颜色分量，为了减少计算量，节约计算资源，本文只使用蓝色通道的分量进行计算，图 5-9(b)是只有蓝色分量的辣椒图像。

图像拍摄时，往往都存在较多噪声，为了减小噪声对检测结果的干扰，需要先对图像进行预处理，本文需要的是辣椒图像的轮廓，因此在减少噪声的同时，要尽量保留辣椒的边缘轮廓，本文选择使用双边滤波[79]，双边滤波是非线性滤波的一种，其结合像素值相似性和空间邻近性进行加权平均，同时考虑了灰度相似性和空域信息，在降低噪声的同时，保留了图像较多的边缘轮廓。图 5-9 (b)使用双边滤波器去噪后的结果如图 5-9(c)所示。

双边滤波后需要对辣椒图像进行边缘检测，将辣椒的轮廓检测出来，本文使用的是 Canny 边缘检测[80]，Canny 边缘检测是经典的边缘检测算子之一，其首先对图像进行高斯滤波，进一步减少噪声的影响，然后计算每一个像素的梯度，接着对所有像素的梯度进行非极大值抑制，来细化边缘并消除边缘响应，然后再使用双阈值将所有的像素分成弱边缘像素，强边缘像素和非边缘像素，最后将强边缘像素和弱边缘像素中与强边缘像素相邻的像素连接起来，组成完整的边缘。Canny 边缘检测的结果如图 5-9(d)所示。

边缘检测的结果中往往存在很多轮廓，需要根据轮廓的特征，找出待检测辣椒的轮廓，本文根据轮廓所围的面积大小筛选轮廓，面积最大的就是待检测辣椒的轮

廓，轮廓筛选结果如图 5-9(e)所示。最后对检测出的辣椒轮廓进行矩形拟合，计算出能包围所有轮廓的最小外接矩形，再计算矩形的中心对称线与垂直线的角度，矩形拟合结果及角度计算如图图 5-9(f)所示。该角度就是辣椒左右偏转的角度，也是采摘该辣椒夹爪需要旋转的角度。对于偏转不太大的辣椒，末端执行器不需要进行相应的旋转，也能很好的完成采摘，因此对偏转角度在 20 度以下的辣椒，发送给机械臂的偏转信息设置为 0 度，对超过 20 度的辣椒，正常发送偏转角度，帮助机械臂更好的完成辣椒采摘。

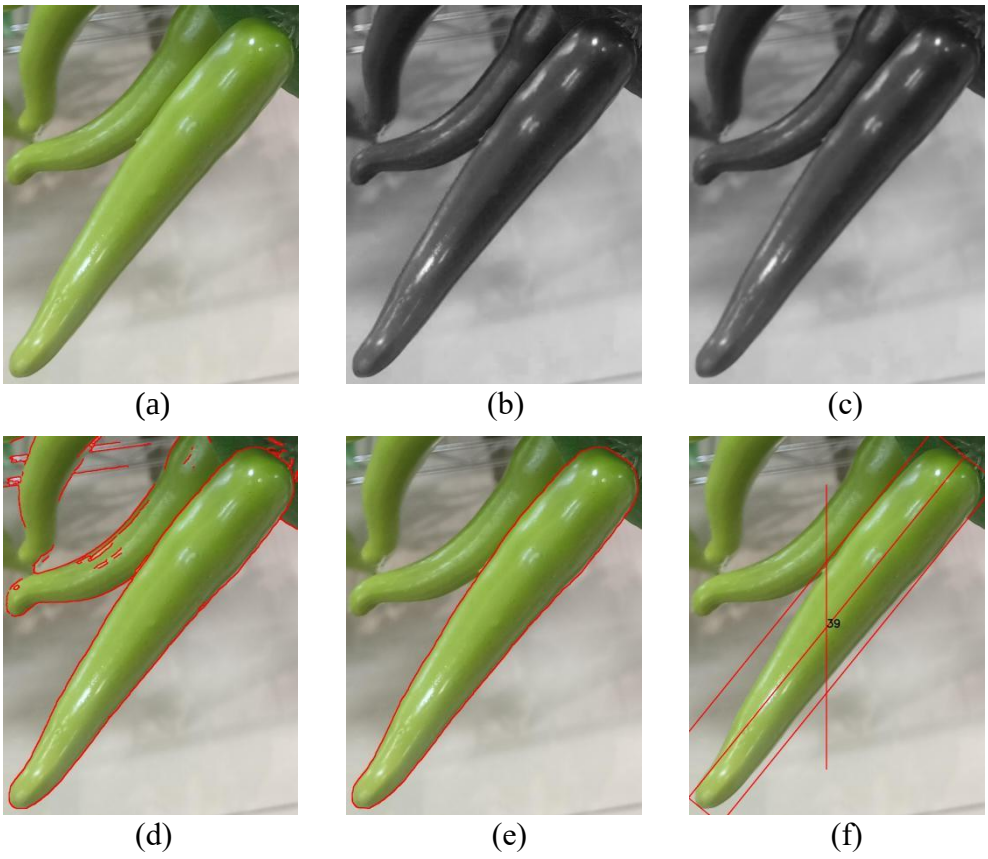


图 5-9 辣椒偏转角度计算过程。(a)辣椒原始图像；(b)辣椒蓝色通道图像；(c)双边滤波后的结果；(d)Canny 边缘检测的结果；(e)轮廓筛选的结果；(f)矩形框拟合的结果

5.5 辣椒采摘实验

5.5.1 软件界面设计

为了更方便的对辣椒进行识定位采摘，本文基于 python 中的 GUI 库 Tkinter 开发了一个简易的辣椒采摘软件界面，将辣椒的识别定位，定位，采摘功能集成到

一起，具备显示深度相机彩色图像，检测辣椒，计算目标辣椒的三维位置和偏转角度，将相关信息发送给机械臂和末端执行器完成采摘等功能。设计出的辣椒采摘界面如图 5-10 所示，图中左侧黑色部分是 RealSense D415 深度相机的画面，在其下方的文本框中会输出已识别的辣椒在机械臂坐标系中的坐标信息和偏转角度。右侧是连接深度相机，机械臂和末端执行器的按钮可以控制他们与上位机的连接状态，机械臂复位按钮可以让机械臂回到初始位置，末端执行器复位按钮可以让夹爪回到初始的张开状态。辣椒采摘按钮是用改进后的 YOLOv8s 模型对深度相机中的画面进行辣椒识别，根据辣椒已检测到辣椒的三维坐标和偏转角度至下方文本框，同时将相关信息发送给机械臂执行采摘。右下角是辣椒采摘的相关数据统计。工作中的辣椒采摘软件界面如图 5-11 所示。整体的辣椒采摘流程如图 5-12 所示。

图 5-10 辣椒采摘系统软件界面

图 5-11 辣椒采摘软件界面运行效果图

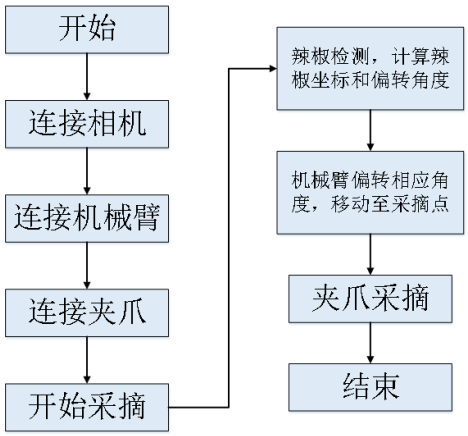


图 5-12 辣椒采摘流程图

5.5.2 辣椒采摘实验结果分析

为了验证辣椒采摘系统的可行性，在完成辣椒采摘系统的硬件搭建和软件界面设计后,需要进行采摘实验，本文搭建了模拟的仿真辣椒环境来进行实验，根据辣椒遮挡重叠的程度，搭建了轻度遮挡和重度遮挡两种采摘场景，如图 5-13 所示，左侧的辣椒比较少，遮挡重叠的情况也较少，右侧的辣椒比较密集，遮挡重叠的情况较多。

图 5-13 模拟辣椒采摘场景

在两组辣椒采摘场景中进行多组采摘实验，辣椒识别和采摘结果如表 5-3 所示。分别进行了多组采摘实验，遮挡较少的辣椒一共有 38 个，成功识别 36 个，成功采摘 33 个，识别成功率达到 94.7%，采摘成功率达到 86.8%，遮挡较多的辣椒一共有 90 个，成功识别 81 个，成功采摘 71 个，识别成功率达到 90.0%，采摘成功率达到 78.8%。一共有 128 个辣椒，成功识别 117 个辣椒，成功采摘 104 个辣椒，总体识别成功率达到 91.4%，采摘成功率达到 81.3%，单个辣椒的采摘时间在

10-15s 之间。其中辣椒没有识别出的原因主要是被遮挡的过多，采摘识别的原因有：辣椒被枝叶遮挡过多，末端执行器难以靠近辣椒完成采摘；辣椒之间靠的太近，末端执行器张开较大，一次夹取了两个辣椒，导致其中一个滑落而采摘失败。

表 5-3 辣椒识别和采摘结果表

遮挡程度	编号	辣椒数	识别数	采摘数
轻度	1	8	8	8
	2	10	10	9
	3	11	10	9
	4	9	8	7
重度	1	20	18	16
	2	22	21	19
	3	25	22	19
	4	23	20	17
总计		128	117	104

总的来说，本文提出的辣椒识别模型和采摘系统能够较好的完成识别和采摘任务，验证了辣椒采摘系统的可行性。其中一次辣椒采摘的机械臂中间过程如图 5-14(a)-(d)所示。

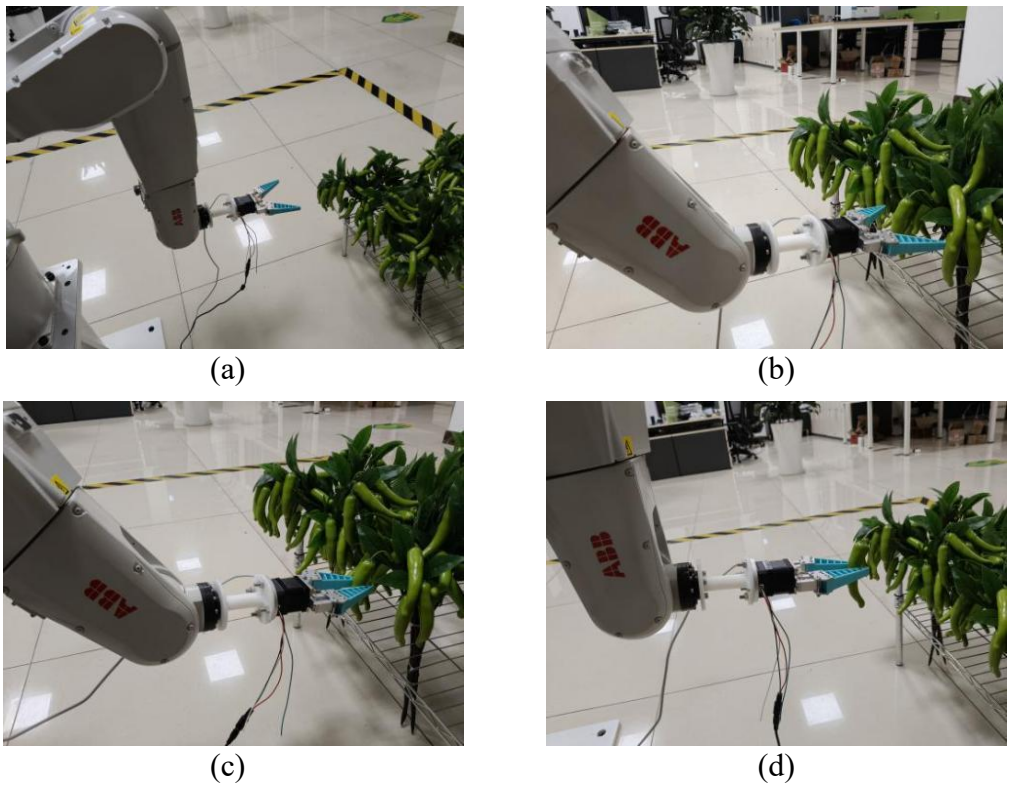


图 5-14 辣椒采摘过程。(a)机械臂初始状态；(b)机械臂到达目标辣椒位置；(c)末端执行器夹取辣椒；(d)机械臂完成辣椒采摘

5.6 本章小结

本章首先接着介绍了相机的成像模型和双目相机的测距原理，然后对深度相机进行标定和机械臂的联合手眼标定，并进行了定位实验，综合定位误差是 3.97 mm，误差很小，满足辣椒采摘的需求。接着对辣椒的姿态进行了分析，计算辣椒的偏转角度，最后介绍了辣椒采摘系统的软件操作界面和采摘流程，并在模拟辣椒采摘环境中进行了采摘实验，总体的辣椒识别成功率达到 91.4%，采摘成功率达到 81.3%，取得了较好的实验结果，基本满足采摘需求。

第六章 总结和展望

6.1 论文总结

为了推动我国农机装备的自动化、信息化和精准化，本文以田间辣椒为研究对象，利用深度学习，双目视觉定位等技术，进行辣椒采摘系统的研发，论文主要研究了以下内容：

（1）建立了辣椒数据集，通过实地拍摄和网络爬取获取了 2488 张辣椒图像，并进行标注，构建了高质量的辣椒数据集。

（2）提出一种改进 YOLOv8s 的辣椒识别算法，针对辣椒数据集中的枝叶遮挡和密集重叠的问题，在 YOLOv8s 模型的 Neck 中，加入 SEAM 注意力机制，提升模型对遮挡辣椒的检测能力，使用 Varifocal 损失函数实现更高效的样本加权机制，在后处理中使用 Soft-NMS 保留更多的高质量预测框，减少漏检。改进后的模型相比原模型 mAP 提高了 4.4%，实验结果表明，本文提出的模型在辣椒检测中具有更高的检测精度。

（3）基于 RealSense D415 深度相机建立辣椒定位系统，首先依据张友正标定法对相机进行标定，获取相机的内参。分析手眼标定的类型和优缺点，选择手在眼外，将相机固定在机械臂外，并求出手眼矩阵，进行定位实验，综合平均误差是 3.97 mm，满足采摘需求，最后计算辣椒偏转角度，辅助夹爪更好的完成采摘。

（4）完成辣椒采摘系统搭建和软件界面设计，在模拟辣椒采摘场景中进行采摘实验，最终的辣椒识别率达到 91.4%，采摘成功率达到 81.3%，本文提出的辣椒采摘系统可以准确的识别和采摘辣椒。

6.2 后续工作展望

本文针对密集遮挡的辣椒设计了基于YOLOv8s的目标检测算法，并设计了辣椒采摘，取得一定的研究成果，但受限于时间和实验条件，也存在一些不足之处，在以下几个方面仍需进一步研究：

（1）目前本文基于YOLOv8s的改进模型检测性能还不错，但是目标检测算法发展迅速，后续考虑选择检测性能更好的基线模型进行改进优化，进一步提高模型复杂场景下辣椒的检测准确率。

（2）目前二指电动夹爪还是不够灵活，难以采摘被枝叶层层遮挡的辣椒，后续需设计更加小巧灵活的末端执行器。同时，本文用传统图像处理对辣椒进行了简单

的姿态分析，后续考虑在目标检测算法中融入辣椒的位姿识别，以便末端执行器更好的完成辣椒采摘。

（3）辣椒采摘系统目前只进行了实验室模拟采摘实验，没有设计移动底盘进行田间实验，后续应考虑设计移动底盘，搭载辣椒采摘系统在田间开展进一步的实验研究。

参考文献

[1] 中国科学院中国植物志编辑委员会. 中国植物志[M]. 北京: 科学出版社, 1997.

[2] 中华人民共和国统计局. 中国统计年鉴[M]. 北京：中国统计出版社, 2022.

[3] 刘成良, 林洪振, 李彦明, 等. 农业装备智能控制技术研究现状与发展趋势分析[J]. 农业机械学报, 2020, 51(1): 1-18.

[4] Parrish E A, Goksel A K. Pictorial pattern recognition applied to fruit harvesting[J]. Transactions of the ASAE, 1977, 20(5): 822-0827.

[5] Hayashi S, Ganno K, Ishii Y, et al. Robotic harvesting system for eggplants[J]. Japan Agricultural Research Quarterly: JARQ, 2002, 36(3): 164-168.

[6] Kelman E E, Linker R. Vision-based localisation of mature apples in tree images using convexity[J]. Biosystems Engineering, 2014, 118: 174-185.

[7] 王晋. 自然环境下苹果采摘机器人视觉系统的关键技术研究[D]. 燕山: 燕山大学, 2014.

[8] Li B, Long Y, Song H. Detection of green apples in natural scenes based on saliency theory and Gaussian curve fitting[J]. International Journal of Agricultural and Biological Engineering, 2018, 11(1): 192-198.

[9] 赵德安, 沈甜, 陈玉, 等. 苹果采摘机器人快速跟踪识别重叠果实[J]. 农业工程学报, 2015, 31(2): 22-28.

[10] Ji W, Zhao D, Cheng F, et al. Automatic recognition vision system guided for apple harvesting robot[J]. Computers & Electrical Engineering, 2012, 38(5): 1186-1195.

[11] Bac C W, Hemming J, Van Henten E J. Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper[J]. Computers and Electronics in Agriculture, 2013, 96: 148-162.

[12] 张春龙, 张楫, 张俊雄, 等. 近色背景中树上绿色苹果识别方法[J]. 农业机械学报, 2014, 45(10): 277-281.

[13] 王丹丹. 重叠及遮挡影响下的苹果目标识别与定位方法研究[D]. 杨凌: 西北农林科技大学, 2016.

[14] Wang C, Zou X, Tang Y, et al. Localisation of litchi in an unstructured environment using binocular stereo vision[J]. Biosystems Engineering, 2016, 145: 39-51.

[15] Wan S, Goudos S. Faster R-CNN for multi-class fruit detection using a robotic vision system[J]. Computer Networks, 2020, 168: 107036.

[16] Jian L, Mingrui Z, Xifeng G. A fruit detection algorithm based on r-fcn in natural scene[C]. Chinese Control And Decision Conference, Hefei, China, 2020: 487-492.

[17] Gao F, Fu L, Zhang X, et al. Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN[J]. Computers and Electronics in Agriculture, 2020, 176: 105634.

[18] Parvathi S, Selvi S T. Detection of maturity stages of coconuts in complex background using Faster R-CNN model[J]. Biosystems Engineering, 2021, 202: 119-132.

[19] Yu Y, Zhang K, Yang L, et al. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN[J]. Computers and electronics in agriculture, 2019, 163: 104846.

[20] 李佳俊, 朱子峰, 刘洪鑫, 等. 基于改进 Faster R-CNN 模型的草莓果实识别算法[J]. 湖北农业科学, 2023, 62(11): 183.

[21] Zhang X, Gao Q, Pan D, et al. Research on spatial positioning system of fruits to be picked in field based on binocular vision and SSD model[C]. Journal of physics: Conference Series, Xiamen, China, 2021, 1748(4): 042011.

[22] Wang Y, Xing Z, Ma L, et al. Object detection algorithm for lingwu long jujubes based on the improved SSD[J]. Agriculture, 2022, 12(9): 1456.

[23] Peng H, Chen H, Zhang X, et al. Retinanet_G2S: A multi-scale feature fusion-based network for fruit detection of punna navel oranges in complex field environments[J]. Precision Agriculture, 2024, 25(2): 889-913.

[24] 刘天真, 滕桂法, 苑迎春, 等. 基于改进 YOLO v3 的自然场景下冬枣果实识别方法[J]. 农业机械学报, 2021, 52(5): 17-25.

[25] Tang Y, Zhou H, Wang H, et al. Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision[J]. Expert Systems with Applications, 2023, 211: 118573.

[26] Wang X, Wu Z, Jia M, et al. Lightweight SM-YOLOv5 tomato fruit detection algorithm for plant factory[J]. Sensors, 2023, 23(6): 3336.

[27] Wang F, Sun Z, Chen Y, et al. Xiaomila green pepper target detection method under complex environment based on improved YOLOv5s[J]. Agronomy, 2022, 12(6): 1477.

[28] 孙成宇. 基于深度学习的辣椒细粒度识别及采摘定位研究[D]. 贵阳: 贵州大学, 2024. 55-62.

[29] 高研. 基于单目图像的苹果目标定位及采摘装置研发[D]. 杨凌: 西北农林科技大学, 2023. 17-26.

[30] Plebe A, Grasso G. Localization of spherical fruits for robotic harvesting[J]. Machine Vision and Applications, 2001, 13: 70-79.

[31] 张东航. 基于双目定位的草莓采摘机器人设计[D]. 北京: 北方工业大学, 2024. 9-22.

[32] 郑雨睿. 基于双目视觉的樱桃番茄识别定位技术研究[D]. 杭州: 杭州电子科技大学, 2024. 41-56.

[33] 麦春艳, 郑立华, 孙红, 等. 基于 RGB-D 相机的果树三维重构与果实识别定位[J]. 农业机械学报, 2015 (S1): 35-40.

[34] 王明和. 融合深度学习目标检测的苹果采摘系统研发[D]. 杨凌: 西北农林科技大学, 2023. 24-32.

[35] Schertz C E, Brown G K. Basic considerations in mechanizing citrus harvest[J]. Transactions of the ASAE, 1968, 11(3): 344-0346.

[36] Kondo N, Monta M, Fujiura T. Fruit harvesting robots in Japan[J]. Advances in Space Research, 1996, 18(1-2): 181-184.

[37] Kondo N, Yata K, Iida M, et al. Development of an end-effector for a tomato cluster harvesting robot[J]. Engineering in Agriculture, Environment and Food, 2010, 3(1): 20-24.

[38] Williams H, Ting C, Nejati M, et al. Improvements to and large-scale evaluation of a robotic kiwifruit harvester[J]. Journal of Field Robotics, 2020, 37(2): 187-201.

[39] 吕继东. 苹果采摘机器人视觉测量与避障控制研究[J]. 镇江: 江苏大学, 2012.

[40] 王丽丽. 番茄采摘机器人关键技术研究[D]. 北京: 北京工业大学, 2017.

[41] 李爽. 基于 ROS 和 YOLOv5 的辣椒自主移动采摘平台关键技术研究[D]. 扬州: 扬州大学, 2022.

[42] 杨长辉, 刘艳平, 王毅, 等. 自然环境下柑橘采摘机器人识别定位系统研究[J]. 农业机械学报, 2019, 50(12): 14-22.

[43] 肖旭. 柑橘采摘机器人的采摘关键技术研究与试验[D]. 长沙: 湖南农业大学, 2022.

[44] Wang S C. Interdisciplinary computing in Java programming[M]. Springer Science & Business Media, 2003.

[45] Qin Z, Yu F, Liu C, et al. How convolutional neural network see the world-A survey of convolutional neural network visualization methods[J]. arXiv preprint arXiv:1804.11191, 2018.

[46] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks[C]. Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011: 315-323.

[47] Hendrycks D, Gimpel K. Gaussian error linear units (gelus)[J]. arXiv preprint arXiv:1606.08415, 2016.

[48] DAI Z, WANG P, WEI H. Signal detection based on Sigmoid function in Non-Gaussian noise[J]. 电子与信息学报, 2019, 41(12): 2945-2950.

[49] Ramachandran P, Zoph B, Le Q V. Searching for activation functions[J]. arXiv preprint arXiv:1710.05941, 2017.

[50] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The journal of machine learning research, 2014, 15(1): 1929-1958.

[51] Lin M, Chen Q, Yan S. Network in network[J]. arXiv preprint arXiv:1312.4400, 2013.

[52] Yang Y, Wu Z, Xu Q, et al. Deep learning technique-based steering of autonomous car[J]. International Journal of Computational Intelligence and Applications, 2018, 17(02): 1850006.

[53] Contardo P, Sernani P, Tomassini S, et al. FRMDB: Face recognition using multiple points of view[J]. Sensors, 2023, 23(4): 1939.

[54] 陶显, 侯伟, 徐德. 基于深度学习的表面缺陷检测方法综述[J]. 自动化学报, 2021, 47(5): 1017-1034.

[55] Yu J ,Jiang Y ,Wang Z , et al. UnitBox: An Advanced Object Detection Network. [J]. CoRR, 2016, abs/1608.01471.

[56] Rezatofighi H, Tsoi N, Gwak J Y, et al. Generalized intersection over union: A metric and a loss for bounding box regression[C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Long Beach, USA, 2019: 658-666.

[57] 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, New York, USA, 2020, 34(07): 12994-13000.

[58] Neubeck A, Van Gool L. Efficient non-maximum suppression[C]. 18th international conference on pattern recognition, Beijing, China, 2006, 3: 850-855.

[59] Baraud Y. Tests and estimation strategies associated to some loss functions[J]. Probability Theory and Related Fields, 2021, 180(3): 799-846.

[60] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]. Proceedings of the European Conference on Computer Vision, Amsterdam, Netherlands, 2016: 21-37.

[61] 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 Rcognition, Las Vegas, USA, 2016: 779-788.

[62] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, USA, 2017: 7264-7271.

[63] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[64] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

[65] 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, Columbus, USA, 2014: 580-587.

[66] Girshick R. Fast r-cnn[C]. Proceedings of the IEEE International Conference on Computer Vision, Santiago, Republic of Chile, 2015: 1440-1448.

[67] 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.

[68] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, 2018: 7132-7141.

[69] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]. Proceedings of the European Conference on Computer Vision, Munich, Germany, 2018: 4-19.

[70] Yu Z, Huang H, Chen W, et al. Yolo-facev2: A scale and occlusion aware face detector[J]. Pattern Recognition, 2024, 155: 110714.

[71] 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, Venice, Italy, 2017: 2980-2988.

[72] Zhang H, Wang Y, Dayoub F, et al. Varifocalnet: An iou-aware dense object detector[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Kuala Lumpur, Malaysia, 2021: 8514-8523.

[73] Bodla N, Singh B, Chellappa R, et al. Soft-NMS--improving object detection with one line of code[C]. Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 2017: 5561-5569.

[74] 姜大志, 孙闵, 刘淼, 等. 数码相机标定方法研究[J]. 南京航空航天大学学报, 2001, 33(1): 55-59.

[75] 沈彤, 刘文波, 王京. 基于双目立体视觉的目标测距系统[J]. 电子测量技术, 2015 (4): 52-54.

[76] Zhang Z. Flexible camera calibration by viewing a plane from unknown orientations[C]. Proceedings of the IEE International Conference on Computer Vision, Kerkyra, Greece, 1999, 1: 666-673.

[77] 陈铁牛. 机器人运动学标定的研究[D]. 哈尔滨: 哈尔滨工业大学, 2010:50-54.

[78] 胡为, 刘冲, 傅莉, 等. 一种高精度的机器人手眼标定算法[J]. 火力与指挥控制, 2018, 43(9): 19-24.

[79] Tomasi C, Manduchi R. Bilateral filtering for gray and color images[C]. Sixth International Conference on Computer Vision, Bombay, India, 1998: 839-846.

[80] Canny J. A computational approach to edge detection[J]. IEEE Transactions on pattern analysis and machine intelligence, 1986 (6): 679-698.