

分类号： TP391.1

学校代码： 12715

UDC： 004.9

学 号： 2108540006026



西京学院  
XI JING UNIVERSITY

# 专业学位硕士学位论文 Thesis for Master's Degree

## 基于深度学习的番茄叶部病变检测方法研究

学 院：	计算机学院
学 位 类 别：	电子信息
领 域：	计算机技术
论 文 作 者：	梁会成
指 导 教 师：	王黎光 江巧永
日 期：	2023 年 12 月

西京学院研究生处

## 西京学院硕士学位论文

# 基于深度学习的番茄叶部病变检测方法研究

论文作者：	梁会成	学生类别：	全日制
专业学位类别：	电子信息	领域名称：	计算机技术
校内导师：	王黎光	职 称：	副教授
企业导师：	江巧永	职 称：	高级工程师

Dissertation Submitted to  
Xijing University  
for  
The Master Degree of  
Electronic and Information Engineering

**Research on Tomato Leaf Lesion Detection Method Based  
on Deep Learning**

by  
Liang Huicheng

Supervisor: Prof. Wang Liguang  
Supervisor: Senior Engineer. Jiang Qiaoyong

December 2023

## 摘 要

番茄是我国重要的经济作物之一，目前我国番茄产量约占世界番茄产量的三分之一。番茄生长过程中容易发生病害，对番茄生产和产业发展造成很大阻碍，给农户带来巨大经济损失。传统的番茄病害检测方法之一是农户通过观察番茄叶部特征人工判断病害种类，这种方式操作简便、实时性好，在番茄病害检测中发挥着重要作用。但采用这种方法需要检测者具有丰富的经验，而且由于主观因素，不同检测者对同一病害可能得到不同的检测结果，不利于及时确定病害并予以治疗。为解决这些问题，把深度学习技术应用到番茄病害检测，实地采集自然种植状态下七种番茄常见病害叶片及健康叶片构建数据集，研究基于深度学习的番茄叶部病害检测问题，分别以提高检测实时性及检测质量为研究重点，构建了两种基于改进 Yolov5 模型的番茄叶部病变检测算法，设计并实现了用于用户操作的番茄病害检测系统。本文主要创新点概括如下：

（1）构建了基于轻量级Yolov5 的番茄叶部病害检测模型。为了在保证模型精确度的前提下，提高检测速度，将传统Yolov5 网络模型的主干网络替换为Mobile Net-v3 的Bneck结构，使网络模型的参数量降低 64%；加入了CBAM注意力机制，将图像特征的空间信息与通道信息连接；将原模型中特征金字塔中的C3 模块改为基于堆叠思想的C2f模块，在加深网络深度的同时，在一定程度上降低了由于网络深度带来的梯度消失问题，保证了检测的精确度。仿真结果表明：提出的新模型在番茄病害检测的精确率与平均精度均值上分别比原模型提高 1.3%和 1.6%，检测速度提高 71%。

（2）构建了一种基于Swin-Yolov5 的番茄叶部病害检测模型。针对番茄病害实际检测中由于光线强度、拍摄角度及叶片重叠导致的检测效果不佳的问题，构建了一种Swin-Yolov5 模型，该模型引入Swin Transformer模块，先将输入图像分割成若干个小块，并分别提取每个块的特征，通过这种方法可提取到更多的番茄叶部病害信息；同时引入轻量级上采样算子Carafe，在上采样时能获取更大感受野，提取更多输入图像特征，并能根据输入特征进行优化重组，Carafe算子的轻量级特点，确保以非常小的额外参数及计算量，带来网络整体性能的显著提升。仿真结果表明：基于Swin-Yolov5 的番茄叶部病害检测模型的精确率及平均精度均值比原模型分别提高了 2.6%和 3.5%。

（3）针对用户应用需求，设计并实现了番茄病害检测系统，并对系统功能进行了测试。测试结果表明：该系统具备用户管理及登录、检测模型选择、病害图片检

测、检测结果可视化及病害防治建议等功能，用户基于该系统可以加载需要检测的病害图片，选择文中提出的两种检测模型并进行检测，检测完毕显示检测结果并给出病害防治建议。病害图片检测模块具有可扩展性，后期可开发并加载更多检测模型供用户选择。

**关键词：**番茄叶部病害，Yolov5，轻量化，Transformer，目标检测

## ABSTRACT

Tomatoes are one of the important economic crops in China, currently accounting for about one-third of the world's tomato production. Tomatoes are prone to diseases during their growth process, which greatly hinders tomato production and industrial development, and brings huge economic losses to farmers. One of the traditional methods for detecting tomato diseases is for farmers to manually determine the type of disease by observing the characteristics of tomato leaves. This method is easy to operate, has good real-time performance, and plays an important role in tomato disease detection. However, adopting this method requires the tester to have rich experience, and due to subjective factors, different testers may obtain different test results for the same disease, which is not conducive to timely identification and treatment of the disease. To address these issues, deep learning technology was applied to tomato disease detection. Seven common tomato disease leaves and healthy leaves under natural planting conditions were collected on site to construct a dataset. The problem of tomato leaf disease detection based on deep learning was studied, with a focus on improving detection timeliness and quality. Two tomato leaf disease detection algorithms based on the improved Yolov5 model were constructed, Designed and implemented a tomato disease detection system for user operation. The main innovation points of this article are summarized as follows:

(1) A tomato leaf disease detection model based on lightweight Yolov5 was constructed. In order to improve the detection speed while ensuring the accuracy of the model, the backbone network of the traditional Yolov5 network model was replaced with the Bneck structure of Mobile Net-v3, reducing the parameter count of the network model by 64%; Added CBAM attention mechanism to connect spatial information of image features with channel information; Changing the C3 module in the feature pyramid of the original model to a C2f module based on the stacking idea not only deepens the network depth, but also reduces the gradient vanishing problem caused by network depth to a certain extent, ensuring the accuracy of detection. The simulation results show that the proposed new model improves the accuracy and average accuracy of tomato disease detection by 1.3% and 1.6% respectively compared to the original model, and the detection speed is improved by 71%.

(2) A tomato leaf disease detection model based on Swin-Yolov5 was constructed. A

Swin-Yolov5 model was constructed to address the issue of poor detection performance in actual tomato disease detection due to light intensity, shooting angle, and leaf overlap. This model introduces the Swin Transformer module, which first segments the input image into several small blocks and extracts the features of each block separately. Through this method, more information about tomato leaf disease can be extracted; At the same time, the lightweight upsampling operator Carafe is introduced, which can obtain a larger receptive field during upsampling, extract more input image features, and optimize and recombine based on input features. The lightweight feature of the Carafe operator ensures a significant improvement in the overall performance of the network with very small additional parameters and computational complexity. The simulation results show that the accuracy and average accuracy of the tomato leaf disease detection model based on Swin-Yolov5 have been improved by 2.6% and 3.5%, respectively, compared to the original model.

(3) A tomato disease detection system was designed and implemented based on user application requirements, and its functionality was tested. The test results show that the system has functions such as user management and login, detection model selection, disease image detection, visualization of detection results, and disease prevention and control suggestions. Based on the system, users can load disease images that need to be detected, select the two detection models proposed in the article, and perform the detection. After the detection is completed, the detection results are displayed and disease prevention and control suggestions are given. The disease image detection module has scalability, and more detection models can be developed and loaded later for users to choose from.

**KEYWORDS:** Tomato leaf diseases, Yolov5, Lightweight, Transformer, Target detection

# 目 录

第一章 绪论.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状.....	2
1.2.1 传统目标检测.....	2
1.2.2 基于深度学习的目标检测.....	4
1.3 研究内容.....	8
1.4 论文结构.....	9
第二章 基础理论与相关技术介绍 .....	11
2.1 概述.....	11
2.2 卷积神经网络.....	11
2.2.1 卷积层.....	11
2.2.2 池化层.....	12
2.2.3 激活函数.....	13
2.2.4 全连接层.....	17
2.3 Transformer 系列算法.....	18
2.3.1 Transformer 算法.....	18
2.3.2 Vision-Transformer 算法.....	21
2.3.3 Swin-Transformer 算法.....	22
2.4 Mobile net 系列算法.....	23
2.4.1 Mobile Net-v1.....	23
2.4.2 Mobile Net-v2.....	25
2.4.3 Mobile Net-v3.....	26
2.5 Yolo 系列算法.....	27
2.5.1 YOLOv3 算法.....	27
2.5.2 YOLOv4 算法.....	28
2.5.3 YOLOv5 算法.....	30
2.6 本章小结.....	31
第三章 基于轻量级 YOLOv5 的番茄叶部病害检测模型.....	33
3.1 概述.....	33
3.2 番茄病害数据集构建.....	33
3.3 轻量级 YOLOv5 网络模型.....	35
3.3.1 Bneck 模块.....	36
3.3.2 C2f 模块.....	37
3.3.3 CBAM 注意力机制.....	37



3.3.4 损失函数.....	39
3.4 模型结果与分析.....	40
3.4.1 模型环境搭建.....	40
3.4.2 评价指标.....	41
3.4.3 消融实验.....	42
3.4.4 轻量级 Yolov5 模型性能分析.....	43
3.5 本章小结.....	47
第四章 基于 Swin-Yolov5 的番茄叶部病害检测模型.....	49
4.1 概述.....	49
4.2 Swin-Yolov5 网络模型.....	49
4.2.1 Swin-Yolov5 模型设计.....	49
4.2.2 轻量级上采样算子.....	50
4.3 模型性能评估.....	52
4.3.1 消融实验.....	52
4.3.2 模型对比.....	54
4.4 本章小结.....	57
第五章 基于深度学习的番茄叶部病害检测系统.....	59
5.1 概述.....	59
5.2 需求分析.....	59
5.3 系统设计.....	60
5.4 系统实现.....	61
5.5 系统测试.....	64
5.6 本章小结.....	66
第六章 总结与展望.....	67
6.1 全文总结.....	67
6.2 展望.....	68
参考文献.....	69

## 插图索引

图 1.1 传统目标检测流程.....	3
图 1.2 目标检测发展历程图.....	5
图 2.1 卷积操作.....	12
图 2.2 平均池化与最大池化.....	13
图 2.3 ReLu6 激活函数.....	14
图 2.4 LeakyReLu 激活函数.....	14
图 2.5 Sigmoid 和 H-Sigmoid 对比图.....	15
图 2.6 Swish 和 H-Swish 对比图.....	16
图 2.7 GELU 激活函数.....	16
图 2.8 Mish 激活函数.....	17
图 2.9 全连接层.....	17
图 2.10 Self-Attention 模块.....	19
图 2.11 Multi-Head Attention 模块.....	20
图 2.12 Vision Transformer 模块.....	21
图 2.13 Swin Transformer 模块.....	22
图 2.14 SW-MSA 操作流程.....	23
图 2.15 深度可分离卷积.....	24
图 2.16 残差网络.....	25
图 2.17 倒残差网络.....	26
图 2.18 SE 模块.....	26
图 2.19 Mobile Net-v3 最后一层对比图.....	267

图 2.20 目标边界框预测图.....	28
图 2.21 spp 结构图 .....	29
图 2.22 PAN 结构图 .....	29
图 2.23 sppf 结构图.....	30
图 3.1 番茄叶部病害样本.....	34
图 3.2 图片标注工具.....	35
图 3.3 Bneck 网络结构.....	36
图 3.4 轻量级网络模型结构.....	36
图 3.5 C2f 模块网络模型结构 .....	37
图 3.6 CBAM 注意力模块 .....	38
图 3.7 CAM 流程图.....	38
图 3.8 SAM 流程图 .....	39
图 3.9 轻量级网络混淆矩阵.....	45
图 3.10 轻量级网络 P-R 曲线.....	46
图 3.11 轻量级网络检测结果.....	46
图 4.1 Swin-Yolov5 网络模型.....	50
图 4.2 Carafe 算子操作流程 .....	51
图 4.3 上采样核预测模块.....	51
图 4.4 特征重组模块.....	52
图 4.5 Swin-Yolov5 混淆矩阵.....	55
图 4.6 Swin-Yolov5 P-R 曲线.....	56
图 4.7 Swin-Yolov5 检测结果.....	56
图 5.1 番茄病害检测系统功能结构图.....	60

---

---

图 5.2 番茄病害检测系统流程图.....	61
图 5.3 番茄病害检测系统注册界面.....	62
图 5.4 番茄病害检测系统登陆界面.....	62
图 5.5 选择模型.....	63
图 5.6 打开图片.....	63
图 5.7 番茄病害检测系统结构图.....	64

## 表格索引

表 3.1 轻量级网络消融实验.....	42
表 3.2 轻量级网络对比试验.....	44
表 3.3 番茄病害各类指标.....	44
表 3.3 番茄病害各类指标(续).....	45
表 4.1 Swin-Yolov5 网络消融实验.....	52
表 4.1 Swin-Yolov5 网络消融实验（续）.....	53
表 4.2 Swin-Yolov5 网络对比实验.....	54
表 4.3 番茄病害各类指标.....	55
表 5.1 登录功能测试表.....	64
表 5.2 注册功能测试表.....	65
表 5.3 模型选择测试表.....	65
表 5.4 模型初始化测试表.....	65
表 5.5 图片检测功能测试表.....	65

## 注释表

略写	英文全称	中文名称
HOG	Histogram of Oriented Gradient	定向梯度直方图
SIFT	Scale-Invariant Feature Transform	尺度不变特征变换
LBP	Local Binary Pattern	局部二值模式
SVM	Support Vector Machine	支持向量机
NMS	Non-Maximum Suppression	非极大值抑制
DPM	Deformable Parts Model	可变形部件模型
LRN	Local ResponSE Normalization	局部相应归一化
TPH	Transformer Prediction Heads	变压器预测头
DETR	Detection Transformer	检测变压器
LSTM	Long Short-Term Memory	长短时记忆网络
W-MSA	Windows Multi-head SElf-Attention	窗口多头注意力
MSA	Multi-head SElf-Attention	多头注意力机制
SW-MSA	shifted Windows Multi-head SElf-Attention	移动窗口多头注意力
PAN	Path Aggregation Network	路径聚合网络
FPN	Feature Pyramid Network	特征金字塔网络
SPP	Spatial Pyramid Pooling	空间金字塔池化
CAM	Channel Attention Module	通道注意力
SAM	Spatial Attention Module	空间注意力
CBAM	Convolutional Block Attention Module	卷积块注意模块

# 第一章 绪论

## 1.1 研究背景与意义

植物病害是农业生产中一种十分常见的现象，是影响农作物正常生长的重要因素之一，若没有及时发现采取相应措施，会严重影响农作物生产，因此早期发现并诊断这些植物病害至关重要，可提高作物产量及质量，减少经济损失<sup>[1]</sup>。番茄作为被全球广泛种植的农作物，在全球的蔬菜贸易中具有显著的地位，由于各个国家的生产能力与技术存在差异，番茄产量也大有不同，我国是全球番茄产量最大的国家<sup>[2]</sup>，根据官方数据 2022 年我国番茄的种植面积为 110.4 万公顷，番茄年产量达到 6515 万吨，接近全球番茄总产量的三分之一，番茄生产及其产品加工已经成为我国农业经济收入的重要来源之一，而且在我国蔬菜贸易中占比较大，所以番茄种植过程中的病虫害防治是确保其产量和质量的一个关键问题。

番茄病害的特征通常能够在其叶片上有所体现，不同病害种类在叶片上会表现出较大差异，比如番茄叶片的整体轮廓是否完整，叶片是否含有斑点，叶片的颜色是否发生改变等，这些特征有利于对病害种类进行识别。持续采集番茄叶片、提取其特征并判断是否正常，可有效监控番茄在生产过程中的健康情况，是检测番茄病害的前提<sup>[3]</sup>。我国对番茄病害传统的识别方法通常是人工识别，番茄种植人员会持续观察番茄的生长状况，在病害发生初期通常就会首先在其叶片上有所体现，番茄种植人员通过观察会根据自己以往的种植经验进行主观判断，或者根据书本上的病害图片资料进行对比，或根据自己观察到的特征通过网络进行查找考证。不管采取何种方式，传统的人工检测病害的方法都离不开种植人员的主观观察和判断，通常都具有如下不足。一是病害的判断高度依赖经验，经验不足者很难快速准确判断出病害种类。二是病害判断具有明显的主观特性，即使具有丰富经验，不同的人对同样的病症也可能得到不同的病害种类。因为有些病害的特征高度相似，有的病害初期特征并不典型，有的病害特征动态变化，因此不同的人可能得到不同的判断结果。三是人工方法对特征的某些细微差异的分辨能力不足。叶片上的特征会因为番茄品种、地形、光线、降水、气候等因素影响，即使是同种病害其特征也可能会略有差异，人工方法可能因为无法详细分辨这些细微的特征差异而导致对病害种类的判断出现误差。四是人工的方法费时、费力、工作效率低下。

本文以学校物联网与大数据省级实验平台的应用开发为需求牵引，以解决人工

判断番茄病害所存在的问题为具体的切入点，研究采用深度学习技术检测番茄叶部病害的方法，为推动乡村振兴、发展智慧农业开展积极研究探索。智慧农业是将传统农业与大数据技术、云计算技术、物联网技术、3S 技术、数据挖掘技术、机器学习技术、人工智能技术等现代科技进行深度融合的全新生产方式<sup>[4]</sup>，它可以对数据进行存储，通过对数据进行智能化分析，使农业发展智能化，集约化，环保化，精准化<sup>[5]</sup>。为全面推进我国乡村振兴，加快我国农业农村现代化<sup>[6]</sup>，党中央非常重视“智慧农业”的发展。从 2014 年开始，在党中央下发的政策文件中，多次提到“智慧农业”以及农业信息化的重要性，尤其是党的中央一号文件，其中多次强调了农业和农村地区的优先发展，尤其是智慧农业优先发展的战略方针<sup>[7]</sup>。

基于深度学习的番茄叶片病害检测技术首先采集各种番茄病害的图片，与正常图片一起构建训练集，然后采用深度学习技术构建深度学习网络，通过对训练集进行训练优化模型参数，得到优化的病害检测模型，最后通过测试集对检测模型性能进行评估。论文实地采集自然种植状态下七种番茄常见病害叶片及健康叶片构建数据集，研究基于深度学习的番茄叶部病害检测问题，分别以提高检测实时性及检测质量为研究重点，构建了两种基于改进 YOLOv5 模型的番茄叶部病变检测算法，设计并实现了用于用户操作的番茄病害检测系统。研究结果可以作为应用开发程序载入学校物联网与大数据实验平台，为平台提供番茄叶部病变的具体应用；还可作为番茄叶部病变辅助诊断系统，给用户决策提供参考。基于深度学习的番茄病害检测方法不需要专业的判断经验，病害检测客观快速，能检测出人工方法无法鉴别的某些特征上的细微差异，具有更高的检测效率，对解决传统病害诊断所存在的问题，具有重要的理论和参考价值。

## 1.2 国内外研究现状

### 1.2.1 传统目标检测

番茄叶部病变检测是一种应用于农业领域的目标检测方法，其核心技术是目标检测。传统的目标检测方法主要有 Viola-Jones 检测器<sup>[8]</sup>，HOG 算法<sup>[9]</sup>，DPM 模型<sup>[10]</sup>等，这三种算法的主要步骤如下：第一步先对输入的图片进行预处理操作；第二步区域选择也就是候选框，采用穷举策略通过设置大小不同的滑动窗口对图片进行遍历；第三步特征提取，包括 Viola-Jones 检测器中的小波特征（Haar-like features, Haar）<sup>[11]</sup>，HOG 算法和 DPM 模型中的定向梯度直方图（Histogram of Oriented Gradient, HOG）<sup>[12]</sup>，尺度不变特征变换（Scale-Invariant Feature Transform, SIFT）<sup>[13]</sup>局部二值模式



(Local Binary Pattern, LBP)<sup>[14]</sup>, Gabor滤波器<sup>[15]</sup>; 第四步分类器判定, 主要的分类器有支持向量机 (Support Vector Machine, SVM)<sup>[16]</sup>和Adaboost分类器<sup>[17]</sup>; 第五步过滤候选框, 主要技术为非极大值抑制 (Non-Maximum Suppression, NMS)<sup>[18]</sup>; 第六步输出。传统目标检测流程如图 1.1 所示。

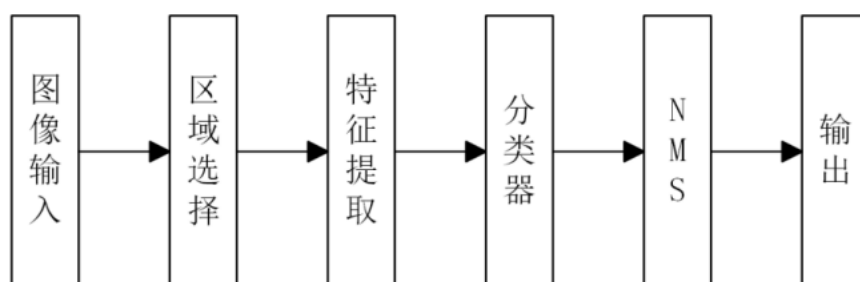


图 1.1 传统目标检测流程

Viola-Jones检测器是Viola和Jones在 2001 年针对人脸检测提出的一种算法, 该算法是在AdaBoost算法的基础上改良而来, 并第一次将小波特征技术应用到人脸检测上, 在人脸检测方面具有里程碑式的意义, 二人设计的这个算法称为Viola-Jones检测器<sup>[19]</sup>。Viola-Jones检测器主要是在四个方面进行改良:

- (1) 使用小波特征Haar: 对输入特征矩形区域的和或差进行阈值化。
- (2) 使用积分图像技术加速小波特征Haar的输入特征计算。
- (3) 使用Adaboost来创建二分类问题的分类器节点。Adaboost是一种迭代算法, 其核心思想是针对同一个训练集训练不同的弱分类器, 然后把把这些弱分类器集合起来, 构成一个强分类器。
- (4) 把强分类器节点组成筛选式级联。在人脸检测中只靠单一的强分类器很难保证准确率, 因此设计了一种类似于决策树的结构, 即第一次分类时允许通过的所有图像中包含一些不包括人脸的图片, 第二次分类时同样允许通过的所有图像中包含一些不包括人脸的图片, 以此类推下去, 只要有图像通过了整个级联, 则认为该图片里有人脸。这样的操作下来可以在前几次的分类器中就可以排除掉没有包含人脸的图片, 从而不需要走完整个筛选式级联, 提高了筛选式级联的运行效率。

HOG算法是Dalal在 2005 年的CVPR上提出的, 主要是用来解决行人检测问题, 主要操作是将图像分成一块块小的连通区域, 然后采集每个区域像素点的梯度或区域边缘的直方图, 最后将这些直方图组合起来并进行归一化处理, 就得到了特征向量。与其他的算法相比HOG有很多优点, HOG是在图像分割后的每一个块上进行操作, 因此对图像的几何和光学上可以有较好的不变性, 同时在粗略的空域抽样、精细的方向抽样以及较强的局部光学归一化等条件下, 只要行人大体上能够保持直立的姿势, 可以容许行人有一些细微的肢体动作, 这些细微的动作可以被忽略而不影

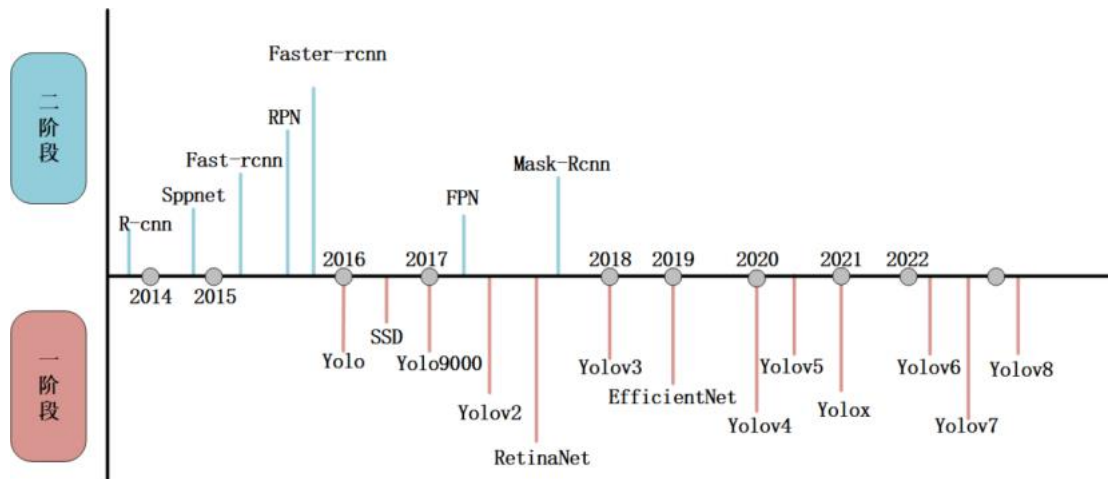
响检测效果。因此HOG特征是特别适合于做图像中的人体检测的。

DPM模型可变形的组件模型，是一种基于组件的检测算法，由Felzenszwalb在2008年提出，该模型在特征提取上取消了原HOG中的块，只保留了单元，在进行归一化时将该单元与其周围的四个单元组成一个区域进行归一化处理，效果与原HOG特征提取接近，同时DPM与原HOG的梯度方向计算方法也不同，DPM采用了有符号梯度和无符号梯度相结合的策略来计算梯度，但是DPM模型不具有普适性，因为用来检测时的激励模块不具有普适性，在每一次进行物体检测时都需要人工来进行设计激励模块，这样会花费大量的时间和工作量，且会因为物体旋转角度过大而无法准确识别，该模型的稳定性很差，但在该算法中提出的模型混合结构思想在深度学习中具有普适性，对后续的目标检测发展有深远的影响。

### 1.2.2 基于深度学习的目标检测

从2010年开始，随着互联网公司的发展，各种网上服务的增加，互联网公司获得大量的数据，使得网上数据可以变得轻易得到，并且伴随着高质量价格合理的传感器，价格低廉的数据存储和廉价计算性能（摩尔定律<sup>[20]</sup>）的普及，使得大规模的算力不再同以前一样稀缺，在得到大量数据以及硬件性能方面的支撑后，深度学习中的中流砥柱如感知机，卷积神经网络，长短期记忆网络又开始重新焕发生机，使得深度学习又重新回到大众视野。

2012年Hinton和他的学生Alex Krizhevsky提出并设计了Alexnet网络<sup>[21]</sup>，并以绝对的优势夺得了当年的ILSVRC 2012（ImageNet Large Scale Visual Recognition Challenge）竞赛的冠军，该网络是计算机视觉领域的重大突破，并首次证明了经过神经网络学习得到的特征可以超越人工手动设计提取到的特征，同时该网络是首次开始用GPU对神经网络进行训练，同时在Alexnet网络中用ReLU激活函数<sup>[22]</sup>替代传统的Sigmoid激活函数<sup>[23]</sup>以及Tanh激活函数，并使用LRN（Local Response Normalization）层对局部神经元的活动创建竞争机制，使其中响应比较大的神经元数值变得相对更大，同时对其他反馈较小的神经元进行抑制，增强了模型的泛化能力，在Alexnet网络中最后三层全连接层的前两层加入Dropout方法<sup>[24]</sup>，并按照一定比例随机失活部分神经元，以达到降低Alexnet神经网络过拟合的问题。Alexnet神经网络的提出使深度学习又一次迎来高潮，使陷入瓶颈期的深度学习迎来曙光，并且由于神经网络有较强的学习能力与特征提取能力，可以使模型通过自主学习提取到更详细的特征信息，摆脱了需要人工设计特征信息提取的传统方式，使得神经网络更加高效，Alexnet神经网络的提出使得在之后的时间涌现出跟多更好的算法，使得目标检



在国内Zhu在 2021 年提出了在Yolov5 的基础上增加一个Transformer预测头（Transformer Prediction Heads, TPH）来检测不同尺度的物体<sup>[25]</sup>，用Transformer预测头取代原来的预测头，利用该注意力机制来提高预测能力，该团队还集成了卷积块注意力模型CBAM，用来在密集对象的场景中找到网络模型的注意力区域，该团队提出的TPH-Yolov5 网络模型，提供了一些有用的策略，如数据增强，多尺度测试，多模型集成和利用额外的分类器。Qi在 2022 年提出了一种改进的SE-Yolov5 网络模型<sup>[26]</sup>，用于对番茄病毒病的识别，通过借鉴人类视觉注意机制，在Yolov5 模型中加入SE注意力机制模块，实现关键特征的提取。在番茄叶部病害的测试集上对训练好的网络模型进行了评价，准确率为 91.07%，平均精度（mAP@0.5）为 94.10%，但是该模型忽略了现实中因光照变化造成的影响，以及在实际检测时植物重叠部分的影响。Wu在 2022 年在Transformer（DS-DETR）番茄叶片病斑检测模型的基础上，有效地提出了几个改进的DETR网络模型<sup>[27]</sup>，该团队将无监督预训练方法引入到DETR网络模型中，以解决DETR网络模型训练周期过长、收敛速度较慢的问题，通过这样操作可以预先训练Transformer模型获得番茄叶片病害特征，在DS-DETR中加载预训练模型权重可以加快模型的收敛速度，空间调制协同注意机制（SMCA）用于分配高斯类空间权重到DS-DETR的查询框，使用具有不同权重的查询框来训练图像中的不同位置，以提高模型的准确性，在DS-DETR Transformer结构中增加了一种改进的相对位置码，相对位置编码促进了Transformer对输入标记的序列顺序的捕获，通过建立不同实例之间的位置关系，增强了空间位置特征，实验结果表明该团队提出的DS-DETR算法疾病检测的准确率达到 0.964，该团队改良后模型的优点是将目标检测的准确率提到

了 95%以上,但缺点是由于加入Transformer注意力机制使得改良后的网络模型参数量增加,计算复杂度增加。Gong H在 2022 年提出并改进了单阶段目标检测算法 Yolov5,以适应卫星图像来解决上述问题<sup>[28]</sup>,该团队从浅层网络开始添加新的特征融合层和预测头用于小目标检测,该方法可以最大限度地保留特征信息,用Swin Transformer Prediction Heads (SPH)替换原始网络模型中的预测头,SPH是一种先进的自注意机制,其移位窗口的设计可以降低神经网络的计算复杂度,将基于归一化的注意力模块(Normalization-based Attention Modules, NAMs)集成到Yolov5中,以归一化的方式提高注意力性能,经实验表明与原有的基础神经网络模型Yolov5相比,SPH-Yolov5在DOTA数据集上将平均平均精密密度(mAP)提高了0.071,该团队提出的网络模型是先进的,因卫星图片分辨率较低,可识度较差因此加入SPH注意力机制后可以对图片的特征提取有较好的表现,但是随之而来的是参数量的增加与计算复杂度的增加。Zheng在 2022 年提出了一种基于Yolox和Densenet的番茄检测方法<sup>[29]</sup>,该团队所提出的方法是能够降低因为光照变化,植物叶片的重叠,和遮挡带来的影响。同时该团队为了提高番茄的检测效率,采用Densenet代替Yolox的主干网络,实现了特征信息的高度融合和复用,在一定程度上改善了网络规模,使平均检测精度(mAP)达到94.80%,提高了检测准确率,该网络模型的优点是准确率较原模型大大提高,但是其缺点是计算复杂度与参数量增加,实际上是通过大量的计算来增加模型准确率。Zeng在 2023 年提出了一个轻量级的改进Yolov5为基础的算法,以实现实时定位和番茄果实的成熟度检测<sup>[30]</sup>,该团队使用下采样卷积层代替原始焦点层,使用Mobile Net-v3的Bneck模块重建Yolov5的骨干网络,为了进一步减小模型规模,对颈部层进行信道修剪,并采用遗传算法进行超参数优化,提高检测精度,使用自制的番茄数据集评估改进后的算法。实验结果表明,改进后的模型参数和每秒浮点运算(FLOPs)的数量分别压缩了78%和84.15%,该网络模型的优点是在确保准确率的同时大大的降低了模型复杂度以及模型参数量。Jing在 2023 年提出了一种BC-Yolov5 番茄病害识别方法<sup>[31]</sup>,该方法采用加权双向特征金字塔网络改变Neck结构,加入卷积块关注模块,改变检测层输入通道,改进Yolov5模型,实验表明,与已有的传统目标检测模型相比,BC-Yolov5对番茄病害的识别性能较好。

在国外RoyAM在 2021 年提出了一种基于改进的Yolov4算法,支持深度学习的多类植物病害目标检测模型<sup>[32]</sup>,该团队将其应用于苹果的各种植物病害检测,对所提出的模型进行了修改,以优化准确性,并通过在复杂的果园场景中检测疾病进行验证。虽然大多数现有模型仅限于大规模疾病检测,但该团队提出的模型解决了细粒度、多尺度等问题,对早期疾病检测的准确率有较好的提升。Ranjai Baidya在 2022

年提出了一种基于Yolov5 模型的新模型架构，其中在预测头上使用ConvMixers以及提供额外的预测头来处理较小的目标<sup>[33]</sup>。与Yolov5 和TPH-Yolov5 等相比，该团队改良后的模型架构可以更好地处理无人机检测图像问题，因为预测头的ConvMixer中的深度卷积和逐点卷积倾向于在空间上和通道方面混合学习的特征，因此可以获得每个特征之间一对一的关系。Seongho Jeong在 2022 年开发了两个DNN模型来诊断番茄叶潜蝇感染的番茄叶片<sup>[34]</sup>。DNNRN152 是开发的DNN模型之一，采用了众所周知的卷积神经网络结构ResNet152，DNNRN152 使用与ResNet152 相同的特征提取器和定制的分类器。DNNRN152 将番茄叶片图像作为输入，直接将图像分类为正常叶片或受番茄潜叶虫感染的叶片，Mask Rcn用于开发另一个DNN模型DNN-MRcnn，利用DNN-MRcnn对番茄叶片图像中的番茄潜叶虫感染区域进行检测，并将检测结果用于分类为正常叶片和番茄潜叶虫感染叶片，接着从现实世界的农业网站捕获相同病害的番茄叶片图像用来训练和评估DNN模型。作为初步研究，该团队比较了DNN模型的分类型模型DNNRN152 和检测模型DNN-MRcnn两种模型的性能，以确定哪种DNN模型能够更好地检测单一作物的单一植物病害，接着从农业网站上捕获图片作为测试集，将精确率(precision)，召回率(recall)，均衡平均数(F-score)作为评价两个开发的DNN模型的指标，通过对比所有指标，DNN-MRcnn优于DNNRN152 番茄潜叶虫的诊断性能，另外以IOU作为评价指标，IOU计算结果表明，在绝大多数测试数据集中，DNN-MRcnn模型优于DNNRN152 模型能够从输入图像中准确地检测出番茄潜叶螨感染的区域。Anil Bhujel在 2022 年提出通过结合不同的注意力模块来设计一个轻量级的卷积神经网络，以提高模型的性能<sup>[35]</sup>，从模型的性能和计算复杂度方面比较了植物病害分类中各种注意力模块的效率，使用标准检测准确性指标精确度、召回率和F1 评分评估模型的性能，结果表明，具有注意力机制的CNN提高了各个病害种类间查准率和召回率，从而提高了整体准确率提高了 1.1%，此外，与标准ResNet50 模型相比，轻量级模型显著降低了网络参数大约 16 倍，计算复杂度降低了 23 倍，然而，在所提出的轻量级模型中，与没有注意力模块的模型相比，具有注意力机制的模型名义上增加了网络复杂度和参数，从而产生更好的检测精度，虽然所有的注意力模块都增强了CNN的性能，但卷积块注意力模块(CBAM)是检测效果最好的，其次是自注意力(SA)机制，该团队通过对比不同的注意力机制，证明在目标检测中CBAM注意力机制在模型中具有较好的表现，该团队通过使用轻量级网络对网络进行改进，并加入注意力模块，使网络在保证准确率的情况下降低了网络的参数量，提升网络计算速度。Storey G在 2022 年提出了一项使用Mask R-CNN进行苹果园叶锈病检测任务的研究<sup>[36]</sup>，同时测试三种不同Mask R-CNN网络骨干网ResNet-50, Mobile

Netv3-Large和Mobile Netv3-Large-Mobile, 通过训练和评估, 用于对象检测, 分割和疾病检测的任务。该研究表明, 具有ResNet-50 主链的Mask R-CNN模型为该任务提供了良好的准确性, 特别是在检测叶子上非常小的锈病物体方面。Arifando R在 2023 年基于Yolov5 提出了一种轻量级、高精度的检测模型<sup>[37]</sup>, 该团队通过将GhostConv和C3Ghost模块集成Yolov5 网络中, 以减少参数数量和每秒浮点运算 (FLOPs), 在减少模型参数的同时保证检测精度, 该团队通过在Yolov5网络中添加SimSPPF模块来取代Yolov5 主干网络中的 SPPF模块, 以提高计算效率和准确的目标检测能力。最后, 该团队通过修改原有的Yolov5 结构, 开发了一个Slim Scale检测模型, 以使模型更高效、更快速, 这对于实时目标检测应用至关重要。根据实验结果, 改进后的Yolov5 模型在精度、召回率和mAP@5.0 方面优于原始Yolov5, 通过对模型复杂性的进一步分析, Improved-Yolov5 网络模型由于FLOPs更少、参数更少、内存使用量更少、推理时间更快, 因此效率更高。

综上, 基于深度学习的目标检测技术在农作物的病害识别上已经取得了一定的成果, 以深度学习为基础的训练模型可以大大提高病害识别准确度, 该技术可以用于番茄病害检测。

### 1.3 研究内容

使用深度学习中的目标检测算法对番茄的七种病害叶片以及健康叶片进行检测, 并基于 Yolov5 网络模型设计了两种网络模型, 通过合适的图像预处理方法, 充分利用两种网络模型对图像进行特征提取, 并比较两种网络模型的优缺点, 研究内容如下:

第一个模型是针对当前主流的目标检测模型虽然精确率较高, 但模型的参数量巨大, 计算时间较为耗时的问题, 构建了基于轻量级的 Yolov5 番茄叶部病害检测模型, 通过对原 Yolov5 网络模型进行改良, 使参数量降低, 提高模型检测速度, 同时保证检测的精度, 轻量级的 Yolov5 网络结构组成如下:

使用轻量级神经网络 Mobile Net-v3 的 Bneck 结构替换掉 Yolov5 中的主干网络部分, 在原网络模型中主干网络深度有 53 层, 经过使用 Bneck 模块后使原网络模型的主干网络下降到 11 层, 这样的处理方式可以使网络模型的参数量成几十倍的减少, 有效的提高了网络模型的检测速度。

在轻量级 Yolov5 网络模型中添加 CBAM 注意力机制, 这是因为在网络模型特征提取阶段, 随着网络模型深度的加深包含的信息增加, 特征提取将会变得困难, CBAM 注意力模块通过对输入特征中有效信息, 增加其权重值, 对于特征中不重要

的信息，降低其权重值，这样的操作可以使得网络对特征的提取更明显，使网络模型只关注有效信息，提高轻量级 Yolov5 网络模型的整体性能。

将原网络模型中的 C3 模块替换成 C2f 模块，C2f 模块主要是对上一层的输出进行一个分流，并结合残差结构和堆叠思想设计而来，这种结构可以大大加快网络的速度，同时经过如此多的堆叠操作，也对应了更密集的残差结构，这样的残差结构可以使网络模型更容易优化。

第二个模型的构建是针对在实际番茄病害检测过程中，由于光线强度，拍摄角度，部分叶片之间互相遮挡或者叶片缺失导致检测效果不显著的问题，构建了一种基于 Swin-Yolov5 的番茄叶部病害检测模型，该模型使用 Swin Transformer 网络结构替换掉原 Yolov5 网络模型的主干网络，该结构会将输入的图像先分割成一个个块，然后模型对每个块进行特征提取，这时块与块之间先不进行信息交互，在后续的特征提取中使用 W-MSA 模块和 SW-MSA 模块使块与块之间实现信息交互，这样做可以降低运算量，提高网络模型的检测精度，同时在模型中加入上采样算子 Carafe，这样在模型进行高语义向低语义进行融合时可以获取更大的感受野，提取更多的特征信息，并加入 CBAM 注意力机制和 C2f 模块，这样的改进方式可以使 Swin-Yolov5 网络模型在检测叶片缺失，遮挡等问题是有较好的准确率。

针对用户需求以本文训练好的模型为核心，设计并实现番茄叶部病害检测系统，首先对番茄病害检测系统进行需求分析说明系统的可行性，接着对系统的整体结构进行划分和设计，并对设计好的每个模块进行实现，最后对实现的系统进行测试，确保满足用户日常所需的正常使用。

## 1.4 论文结构

第一章绪论。主要包括对本文的研究背景和意义进行说明，阐述了我国现阶段大力开展智慧农业的大环境，提出了深度学习在智慧农业中的重要性，并结合国内外研究现状阐述了现阶段深度学习在目标检测中的优点与不足，并针对相应不足之处设计了两个模型以及本文主要工作内容。

第二章主要是对传统的深度学习进行详细的介绍，同时对目标检测相关理论也进行了详细的说明，并对本文后续研究的深度学习算法进行详细的阐述。

第三章是对原模型进行轻量化处理，该章节介绍了数据集的采集过程，设计了基于 Yolov5 网络的轻量级网络模型，对改良后的轻量级模型添加注意力机制，更改其中的 C3 模块，经过消融实验证明改良后的轻量级网络在参数量上有大幅度下降，计算速度上有较好的提升，并通过与主流模型的对比试验表明改良后的神经网络有

较好的准确率。

第四章是使用 Swin Transformer 网络模型对 Yolov5 的主干网络进行改良，并加入注意力机制，同时对 C3 模块进行改进，在模型中使用轻量级上采样算子，经消融实验证明改良后的模型比原模型准确率有所提升，并与其他主流模型相比在番茄病害检测上有较好的准确率。

第五章是以本文训练好的网络模型为核心，设计并实现一个针对番茄病害的检测系统，该系统可以准确的识别番茄病害种类，满足用户在使用该系统时的基本功能，并针对番茄病害类型给出专家建议。

第六章是对本文的主要研究内容进行总结，分析本文模型的不足之处与需要继续研究的部分，并对今后的研究内容进行展望。



## 第二章 基础理论与相关技术介绍

### 2.1 概述

深度学习作为机器学习的一个分支，近年来在目标检测领域取得了非常好的检测性能。本文主要研究基于深度学习技术的番茄叶片病害检测问题，分别以提高检测准确性及检测质量为研究重点，构建两个基于改进 YOLOv5 模型的番茄叶部病变检测模型。一是基于轻量级 YOLOv5 的番茄叶部病害检测模型，目的是在保证模型精确度的前提下提高模型检测速度，它将传统 YOLOv5 网络模型的主干网络替换为 MobileNet-v3 的 Bneck 结构，加入 CBAM 注意力机制，并将原模型中特征金字塔中的 C3 模块改为一种基于堆叠思想的 C2f 模块。二是构建了一种基于 Swin-YOLOv5 的番茄叶部病害检测模型，该模型在传统 YOLOv5 模型中引入 Swin Transformer 模块，把输入图像分割成若干小块以提取更多番茄叶部病害特征，同时引入轻量级上采样算子 Carafe，获取更大受视野。从上述研究路线可以看出，研究涉及的关键技术包括：卷积神经网络、YOLO 系列算法、Transformer 系列算法、MobileNet 系列算法等。

采用 YOLOv5 网络模型是因为相比于其他目标检测算法如 SSD<sup>[38]</sup>，Mask-RCnn<sup>[39]</sup>，RetinaNet<sup>[40]</sup>等，该算法相对而言具有小体积、低参数量、计算速度快等优点。MobileNet 也是轻量级网络模型，可以在保证准确率的前提下降低网络模型的参数量。Swin Transformer 模型可构建层次性的特征图，而且特征图的大小逐层减小，能确保良好检测性能并且不会产生大量网络参数。以 YOLOv5 网络模型为基础，通过上述技术改进，实现满足不同应用需求的番茄叶部病害检测模型。本章将分别分析这些相关技术及涉及到的基础理论。

### 2.2 卷积神经网络

#### 2.2.1 卷积层

卷积层是神经网络中独特的网络机制，卷积目的是对图像进行特征提取，具有局部感知机制和权值共享的两个特性，因为卷积实质可以理解为一个滑动窗口（卷积核）翻转之后在特征图上滑动并进行加乘计算，因此具有局部感知机特征提取功能。在滑动的过程中卷积核的值不会发生改变，因此具有权值共享的特性，这种特

性可以减少神经网络中的参数的数量，提升运行效率，降低计算复杂程度。在实际图像特征提取应用中，卷积核是否翻转对局部感知及特征提取不会产生不同结果，因此为了简化运算，通常不翻转卷积核，这样的运算称为互相关运算，在卷积层中特征矩阵和卷积核通过互相关运算得到输出特征矩阵。图像通常具有多个特征，进行卷积时，每个特征矩阵对应有一个卷积核，分别进行卷积计算，输出提取到的特征。如图 2.1 所示，假设输入一张彩色图像有红、绿、蓝三个色彩特征，由三个输入特征矩阵表示，所以卷积时每个卷积核也必须具有 3 个子卷积核，如下图中的卷积核 1 和卷积核 2 都有 3 个子卷积核，子卷积核分别与红、绿、蓝三个色彩特征矩阵进行卷积，输出卷积后提取到的特征矩阵。

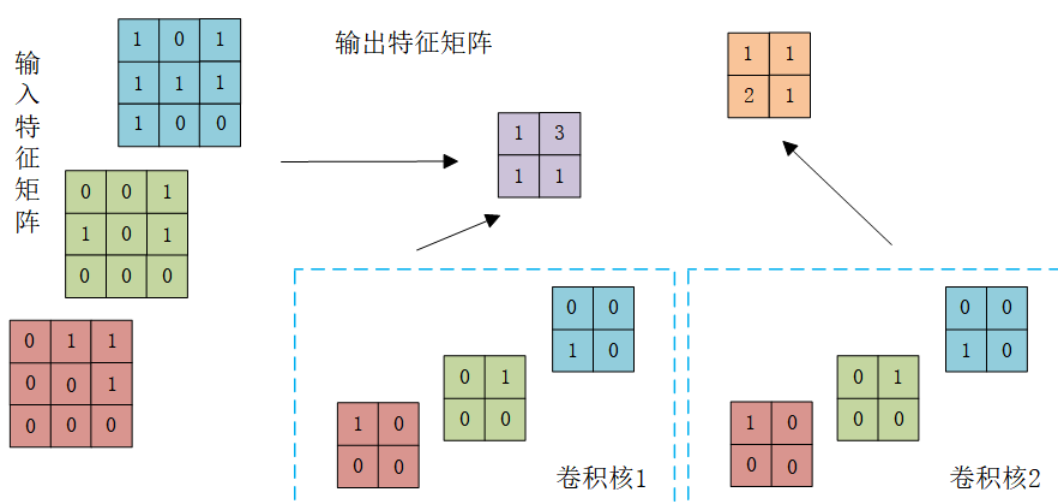


图 2.1 卷积操作

在进行卷积操作的过程中，矩阵经过卷积操作之后输出的特征矩阵的尺寸是由输入图片的大小和卷积核的大小和步长以及补零矩阵 padding 决定的，假设输入的图片大小为  $w$ ，卷积核大小为  $f$ ，步长为  $s$ ，padding 为  $p$ ，经过卷积后得到的特征矩阵大小为  $n$ ，公式(2.1)所示。

$$n = \frac{w - f + 2p}{s} + 1 \quad (2.1)$$

## 2.2.2 池化层

池化层<sup>[41]</sup>是卷积神经网络中一个重要的组成部分，通常在一层卷积计算之后进行一次池化。池化层具有特征选择、减少特征、减少训练参数等功能。池化层的功能是通过池化运算实现的，池化运算通常有最大池化和平均池化两种，池化核是一个  $n*m$  的矩阵，最大池化是以池化核对特征矩阵以一定步长进行扫描，取池化核每次扫描到的区域中的最大值作为池化结果，平均池化是取池化核每次扫描到的区域

中的平均值作为池化结果。如图 2.2 所示, 设池化核为  $2 \times 2$ , 步长为 2, 则池化核第一次扫描到特征矩阵最左上角的  $2 \times 2$  矩阵, 最大池化后输出的值是 30, 平均池化后输出的值是 16, 分别写在图 2.2 右侧池化层输出矩阵的左上角。然后池化层以步长 2 对输入特征矩阵向右扫描, 扫描到右上角的  $2 \times 2$  矩阵, 进行最大池化后得到 8, 平均池化后得到 6, 分别写在图 2.2 右侧池化层输出矩阵的右上角。按照这样的方法, 池化核扫描完输入矩阵, 分别得到最大池化和平均池化后的特征矩阵。

分析池化运算过程可以看出, 最大池化提取了池化核扫描到的区域中的最大值, 平均池化提取了池化核扫描到的区域中的均值, 这些计算方法都是特征选择的方法。如果池化核大于  $1 \times 1$ , 随着池化核的增大, 池化后提取的特征数量就会减少。池化核没有需要训练的参数, 不会额外增加参数, 因此随着特征的减少, 训练的参数也随之减少。在网络中加入池化层可以保证输入特征矩阵中局部特征的平移不变性, 而且可以抑制模型过拟合, 同时降低神经网络的计算复杂程度, 池化层的加入还可以降低卷积神经网络中卷积对位置的敏感程度。

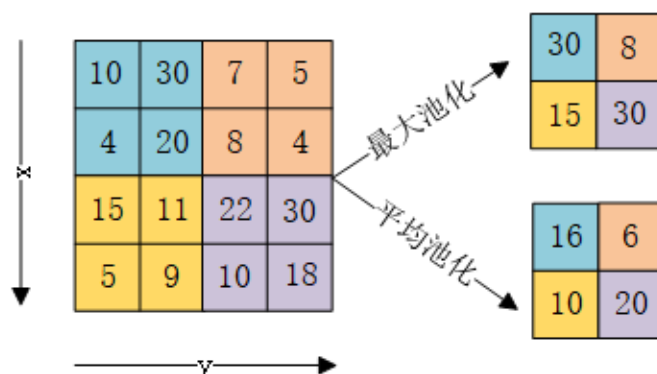


图 2.2 平均池化与最大池化

### 2.2.3 激活函数

输入神经元  $X$  通常通过线性计算  $Z=W^T X+b$  得到净输入信号  $Z$ , 其中  $W$  是输入神经元  $X$  的权值矩阵,  $b$  是对应的偏置矩阵, 净输入信号  $Z$  是一个连续值, 而类别是离散值, 因此通过激活函数判断输出神经元是否会被激活并输出相应的类的取值。激活函数是非线性函数, 它能够把输入神经元的线性运算值通过非线性运算转化为分类所需的离散值, 所以通常在全连接层之后、展平层之后或输出层加入激活函数, 发挥激活函数的功能。

设计深度神经网络时, 对激活函数通常有如下要求: 一是激活函数要连续并可导 (允许少数点上不可导), 因为通常通过梯度法优化网络参数, 可导的激活函数可以直接利用数值优化的方法来学习网络参数。二是激活函数及其导函数要尽可能简

单，有利于提高网络计算效率。三是激活函数的导数值域要在一个合适的区间内，不能太大也不能太小，尤其是对于深度神经网络，随着深度的增加，会设计多层激活函数，求梯度时各层激活函数的导数值会与其他导数值相乘，如果激活函数的导数值太大或太小，容易出现梯度爆炸或梯度消失，影响参数优化。

ReLU 函数能满足上述要求，是深度神经网络中最常用的激活函数之一。ReLU6 激活函数是在 ReLU 激活函数的基础上进行改进的，在 ReLU6 中当我们的输入值小于零时不进行处理，在 0 到 6 的区间不改变其值，在输入值大于 6 时将输入值变为 6，ReLU6 激活函数如图 2.3 所示。

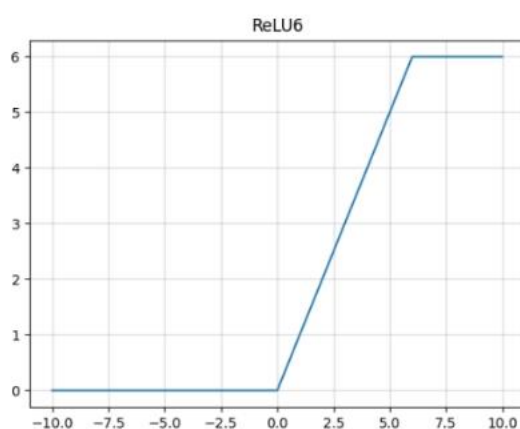


图 2.3 ReLU6 激活函数

ReLU6 激活函数表达式如 (2.2) 所示。

$$y = \text{Relu6}(x) = \min(\max(x, 0), 6) \quad (2.2)$$

如图 2.4 所示，Leaky ReLU 激活函数是另一种改进的 ReLU 函数，通过把非常小的线性分量给予负输入来调整负值的零梯度问题<sup>[42]</sup>。Leak 有助于扩大 ReLU 函数的范围，通常的  $\alpha$  值为 0.01 左右，Leaky ReLU 的函数范围是负无穷到正无穷。

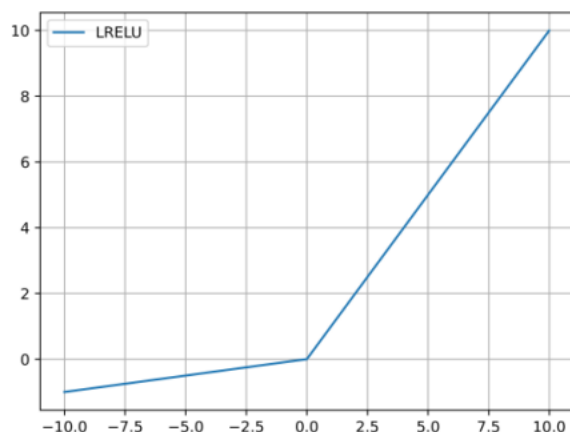


图 2.4 LeakyReLU 激活函数

Leaky ReLu激活函数表达式如(2.3)所示。

$$f(x) = \begin{cases} x & x > 0 \\ \alpha x & x \leq 0 \end{cases} \quad (2.3)$$

Swish激活函数是由谷歌团队在Sigmoid激活函数的基础上改进而来，它是一个自门控激活函数，用于替换Mobile Net-v2 中的部分ReLu6 激活函数。使用Swish激活函数可以提高神经网络的准确率，但是Swish的计算和求导因为有Sigmoid激活函数的加入过于复杂，不利于量化，因此使用H-Swish激活函数。通常情况下H-Sigmoid激活函数比较接近Sigmoid激活函数，因此通常使用H-Sigmoid激活函数来代替Sigmoid激活函数，Sigmoid和H-Sigmoid激活函数如式(2.4)和(2.5)所示，Sigmoid和H-Sigmoid激活函数表达式和对比如图 2.5 所示。

Sigmoid激活函数公式。

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (2.4)$$

H-Sigmoid激活函数公式。

$$H - Sigmoid(x) = \frac{ReLU6(x + 3)}{6} \quad (2.5)$$

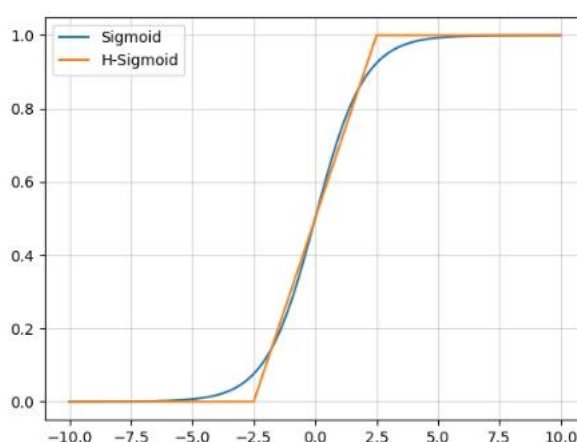


图 2.5 Sigmoid 和 H-Sigmoid 对比图

同理 Swish 激活函数可以用 H-Swish 激活函数来代替，Swish 和 H-Swish 激活函数公式如(2.6)与(2.7)所示，Swish 和 H-Swish 激活函数表达式和对比如图 2.6 所示。

Swish 激活函数公式。

$$Swish(x) = x * Sigmoid(x) \quad (2.6)$$

H-Swish激活函数公式。

$$H - Swish(x) = x \frac{ReLU6(x + 3)}{6} \quad (2.7)$$

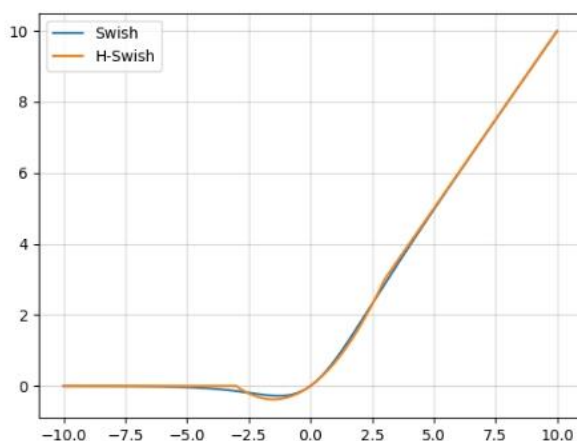


图 2.6 Swish 和 H-Swish 对比图

GELU激活函数是 2016 年提出的激活函数<sup>[43]</sup>, 如图 2.7 所示。GELU激活函数是一种基于高斯误差函数的激活函数, 相较于ReLU等激活函数, GELU激活函数更加平滑, 因为GELU的非线性变化是一种符合预期的随机正则变换方式, 有助于提高训练过程的收敛速度和性能, GELU激活函数的表达式如公式(2.8)所示。

$$GELU(x) = xP(X \leq x) = x\phi(x) \quad (2.8)$$

其中 $\phi(x)$ 指的是标准正态分布的累积分布函数, GELU函数表达式如公式(2.9)所示。

$$GELU(x) = x * \frac{1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)}{2} \quad (2.9)$$

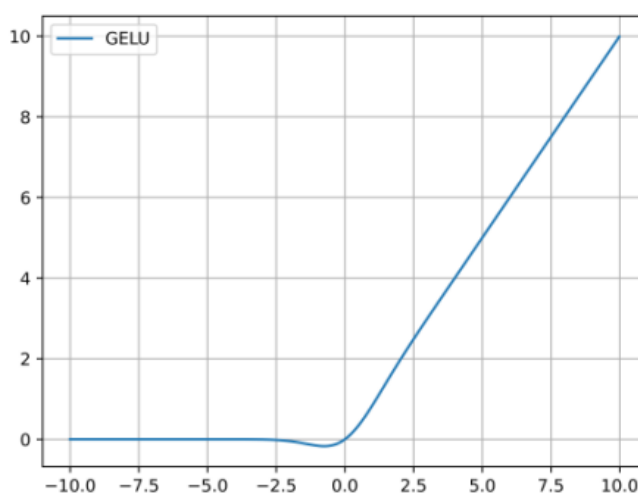


图 2.7 GELU 激活函数

如图 2.8 所示, Mish激活函数是 2019 年提出的<sup>[44]</sup>。Mish函数没有上边限, 这样可以避免由于封顶导致饱和, 在训练过程中可以保证不会有梯度消失的问题, 该函数存在下限可以确保有一定的正则化效应, Mish函数是非单调的这种特性可以在很

小的负输入情况下保持负输出，同时也提高了表现力和渐变流。Mish激活函数表达式如公式(2.10)所示。

$$Mish = x * \tanh(\ln(1 + e^x)) \quad (2.10)$$

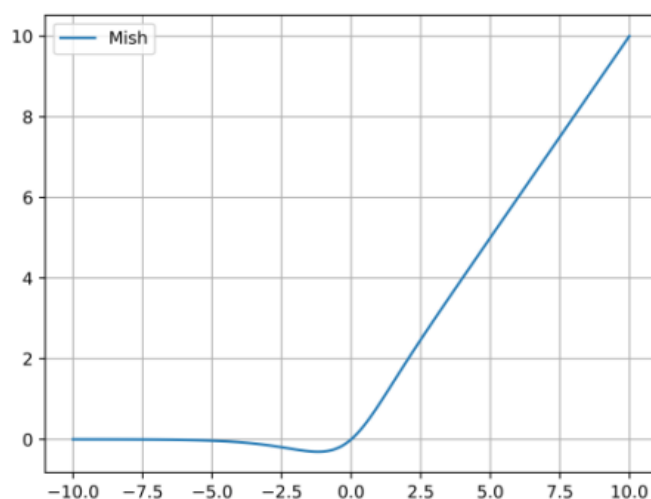


图 2.8 Mish 激活函数

## 2.2.4 全连接层

全连接层通常配置在神经网络的末端，将前面经过卷积、池化和激活函数得到的特征信息进行整合，并在样本空间映射出来。如图 2.9 所示。

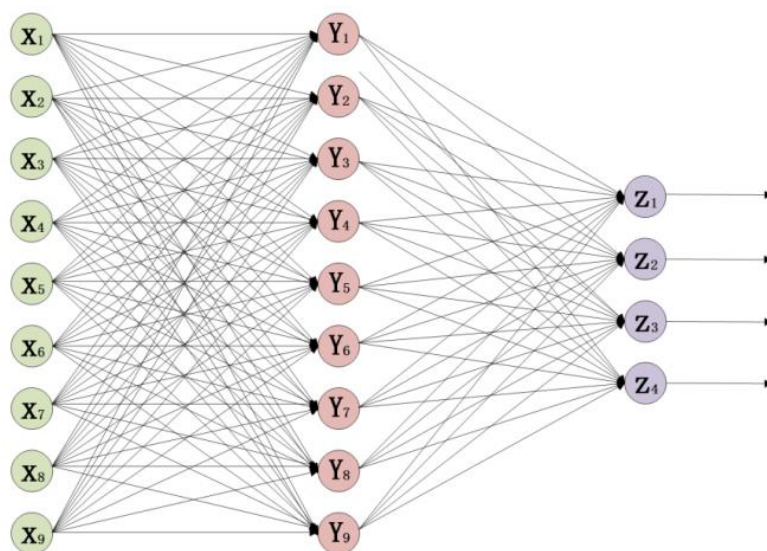


图 2.9 全连接层

全连接层的功能主要有以下几点：一是将高维张量展平为一维向量，适配输出。由于经过卷积、池化、激活函数后得到的特征是以高维张量的形式承载，而输出神经元是一维张量，所以需要通过全连接层把高维张量展平为一维向量，适配输出，

如把图中的张量展平为一维向量 $(x_1, x_2, \dots, x_9)^T$ 。二是综合所有特征，按照分类目标进行适配输出，如图中对特征 $(x_1, x_2, \dots, x_9)^T$ 进行线性计算后，得到综合后的特征 $(Y_1, Y_2, \dots, Y_9)^T$ ，再经过适配后得到净输入 $(z_1, z_2, \dots, z_9)^T$ 。

全连接层中的每个神经元都与前一层的每个神经元连接，每个连接都有权重和偏置，其计算公式如（2.11）所示，其中  $w$  为权重， $b$  为偏置。

$$Y_i = \sum_j w_{ij} X_j + b_i \quad (2.11)$$

## 2.3 Transformer 系列算法

### 2.3.1 Transformer 算法

Transformer是2017年Google在Computation and Language上发表的<sup>[45]</sup>，当时主要是针对自然语言处理领域提出，在此之前使用的模型大多是RNN和LSTM这类时序模型，这些模型记忆长度有限且无法并行化，特别是RNN模型记忆长度特别短，只有计算完 $t_i$ 时刻后的数据才能计算 $t_{i+1}$ 时刻的数据，因此效率很低。针对这些问题Google提出了Transformer模型来替代之前的时序网络，从理论上讲，Transformer如果不受硬件限制，它的记忆长度可以是无限的，并且可以并行化处理。在Transformer中有两个重要的模块Self-Attention模块，和在此基础上改进的Multi-Head Attention模块。

Self-Attention模块的公式如下公式(2.12)所示。

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.12)$$

Self-Attention模块如图2.10所示，假设输入的序列长度为2，输入有两个节点 $x_1, x_2$ ，然后通过  $f(x)$  将输入映射到 $a_1, a_2$ ，再将 $a_1, a_2$ 分别与矩阵 $W^q, W^k, W^v$ 运算，得到对应的 $q^i, k^i, v^i$ ，其中 $W^q, W^k, W^v$ 这三个参数是可训练的，在Self-Attention模块中是共享的，其中 $q$ 代表query，后续会去和每一个 $k$ 进行匹配。 $k$ 代表key，后续会被每个 $q$ 匹配。 $v$ 代表从 $a$ 中提取得到的信息，后续 $q$ 和 $k$ 匹配的过程可以理解成计算两者的相关性，相关性越大对应 $v$ 的权重也就越大。



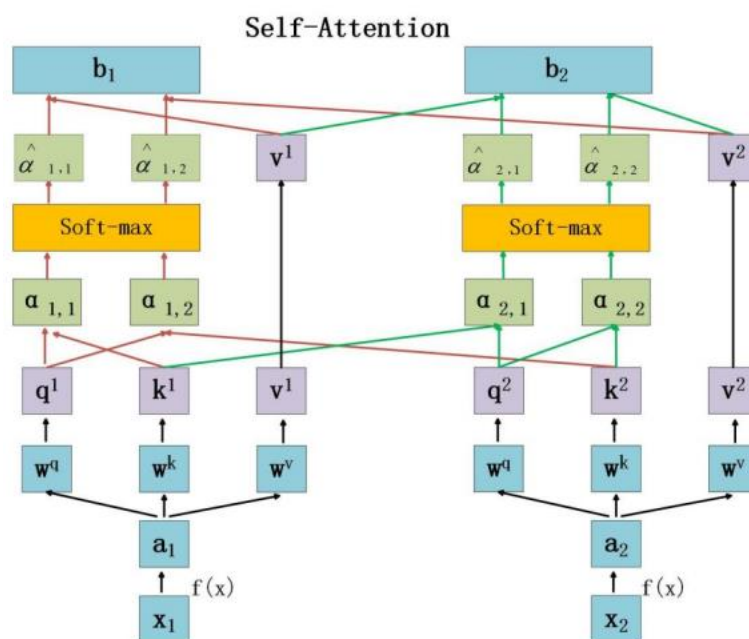


图 2.10 Self-Attention 模块

其中  $q^1 = a^1 * W^q$ ,  $q^2 = a^2 * W^q$ ,  $k^1 = a^1 * W^k$ ,  $k^2 = a^2 * W^k$ ,  $v^1 = a^1 * W^v$ ,  $v^2 = a^2 * W^v$ , 因为 Transformer 是可以并行的, 因此 Attention(Q,K,V) 公式中的 Q, K, V 如公式 (2.13) 所示。

$$\begin{aligned} Q &= \begin{pmatrix} q^1 \\ q^2 \end{pmatrix} = \begin{pmatrix} a^1 \\ a^2 \end{pmatrix} * W^q \\ K &= \begin{pmatrix} k^1 \\ k^2 \end{pmatrix} = \begin{pmatrix} a^1 \\ a^2 \end{pmatrix} * W^k \\ V &= \begin{pmatrix} v^1 \\ v^2 \end{pmatrix} = \begin{pmatrix} a^1 \\ a^2 \end{pmatrix} * W^v \end{aligned} \quad (2.13)$$

接着先拿  $q^1$  和每个  $k$  进行点乘操作, 接着除以  $\sqrt{d}$  得到对应的  $\alpha$ , 其中  $d$  代表向量  $k^i$  的长度, 接着除以  $\sqrt{d}$  的原因是因为进行点乘后的数值很大, 导致通过 softmax 后梯度变的很小, 所以通过除以  $\sqrt{d}$  来进行缩放,  $\alpha_{1,j}$  公式如下 (2.14) 所示。

$$\alpha_{1,1} = \frac{q^1 * k^1}{\sqrt{d}}, \quad \alpha_{1,2} = \frac{q^1 * k^2}{\sqrt{d}} \quad (2.14)$$

同理  $\alpha_{2,j}$  公式如下 (2.15) 所示。

$$\alpha_{2,1} = \frac{q^2 * k^1}{\sqrt{d}}, \quad \alpha_{2,2} = \frac{q^2 * k^2}{\sqrt{d}} \quad (2.15)$$

因为 Transformer 是可以并行的, 因此将  $\alpha_{1,j}$  与  $\alpha_{2,j}$  统一写成矩阵乘法形式, 公式如下 (2.16) 所示。

$$\begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} \\ \alpha_{2,1} & \alpha_{2,2} \end{pmatrix} = \frac{\begin{pmatrix} q^1 \\ q^2 \end{pmatrix} \begin{pmatrix} k^1 \\ k^2 \end{pmatrix}^T}{\sqrt{d}} \quad (2.16)$$

接着对每一行即 $(\alpha_{1,1}, \alpha_{1,2})$ 和 $(\alpha_{2,1}, \alpha_{2,2})$ 分别进行softmax处理得到 $(\hat{\alpha}_{1,1}, \hat{\alpha}_{1,2})$ 和 $(\hat{\alpha}_{2,1}, \hat{\alpha}_{2,2})$ ，这里的 $\hat{\alpha}$ 相当于计算得到针对每个v的权重，因此Attention(Q,K,V)公式中的 $\text{softmax}(\frac{QK^T}{\sqrt{d_k}})$ 部分就是上面所阐述的，接着针对每个v的权重进行加权得到最终结果 $b_i$ ，公式如(2.17)所示。

$$\begin{aligned} b_1 &= \hat{\alpha}_{1,1} * v^1 + \hat{\alpha}_{1,2} * v^2 \\ b_2 &= \hat{\alpha}_{2,1} * v^1 + \hat{\alpha}_{2,2} * v^2 \end{aligned} \quad (2.17)$$

因为Transformer是可以并行的，因此将 $b_1$ 与 $b_2$ 统一写成矩阵乘法形式，公式如下(2.18)所示。

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} \hat{\alpha}_{1,1} & \hat{\alpha}_{1,2} \\ \hat{\alpha}_{2,1} & \hat{\alpha}_{2,2} \end{pmatrix} \begin{pmatrix} v^1 \\ v^2 \end{pmatrix} \quad (2.18)$$

将上面公式整合到一起就是Self-Attention模块。

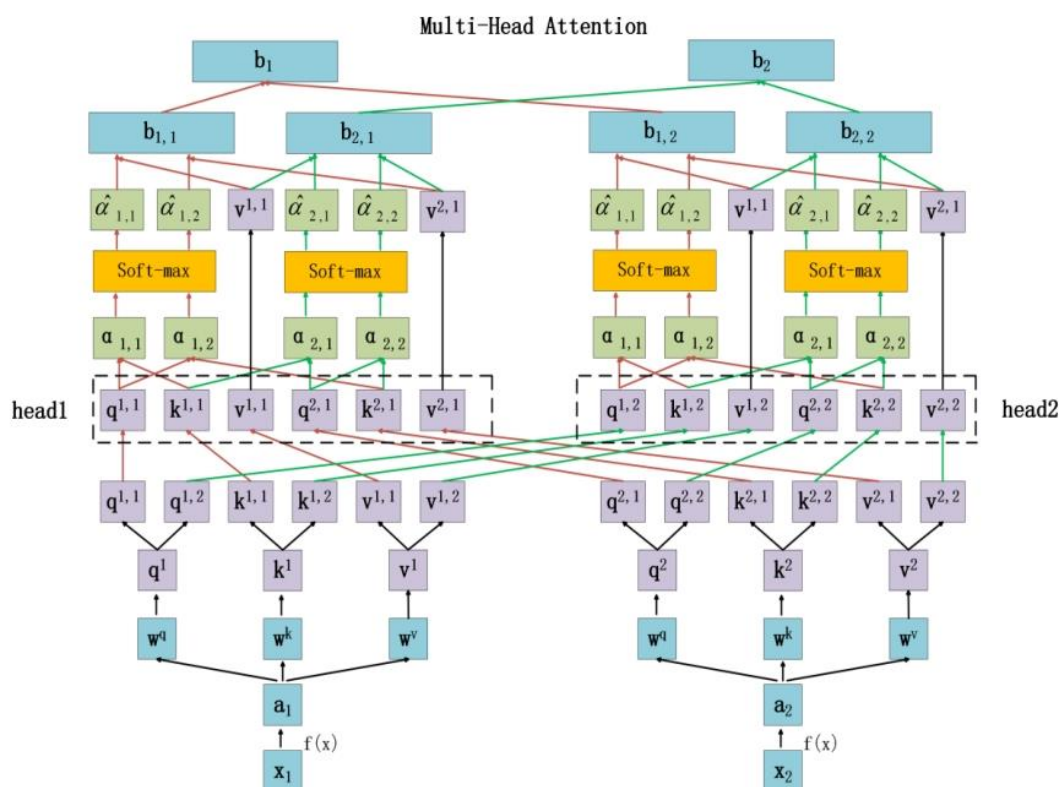


图 2.11 Multi-Head Attention 模块

Multi-Head Attention模块和Self-Attention模块一样将 $a_i$ 分别通过 $W^q$ ,  $W^k$ ,  $W^v$ ，得到对应的得到对应的 $q^i$ ,  $k^i$ ,  $v^i$ ，然后再根据使用的head的数目 $h$ 进一步把得到的 $q^i$ ,  $k^i$ ,

$v^i$ 均分成 $h$ 份。比如图 2.11 中假设 $h=2$ 然后将 $q^1$ 拆分成 $q^{1,1}$ 与 $q^{1,2}$ ，其中 $q^{1,1}$ 属于 $head_1$ ， $q^{1,2}$ 属于 $head_2$ 。Multi-Head Attention模块如图 2.11 所示。

其中 $head_i$ 的取值公式如公式(2.19)所示。

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (2.19)$$

通过上述公式就能得到每个 $head_i$ 对应的 $Q_i$ ， $K_i$ ， $V_i$ 参数，在针对每个 $head$ 使用和 Self-Attention中相同的方法的到相应的结果，公式如(2.20)所示。

$$Attention(Q_i, K_i, V_i) = Softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \quad (2.20)$$

将每一个 $head$ 得到的结果进行Concat拼接，将拼接后的结果通过 $W^0$ 进行融合，得到最终的结果，Multi-Head Attention模块公式如下(2.21)所示。

$$\begin{aligned} Multi-Head(Q, K, V) &= Concat(head_1, \dots, head_i)W^0 \\ where \ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \quad (2.21)$$

### 2.3.2 Vision-Transformer 算法

Vision-Transformer 模型<sup>[46]</sup>的提出推翻了自 2012 年 Alexnet 提出的卷积神经网络在计算机视觉的统治地位，如果在足够多的数据上去做预训练，在不添加卷积神经网络的情况下，用于自然语言处理的标准 Transformer 也可以应用到视觉处理，并得到良好性能。Vision-Transformer 首先将一张图分成若干个 patches，将这些 patches 变成一个序列，每个 patch 经过线性投射层操作，得到相应的特征，再加上相应的位置编码，这个向量序列就包含了图像块原有的图像信息及所在的图像块信息，将得到的向量序列传入 Transformer，因为所有的向量序列都跟其他的向量序列两两交互信息，因此只需输出第一个向量序列就即可。Vision-Transformer 模型图如 2.12 所示。

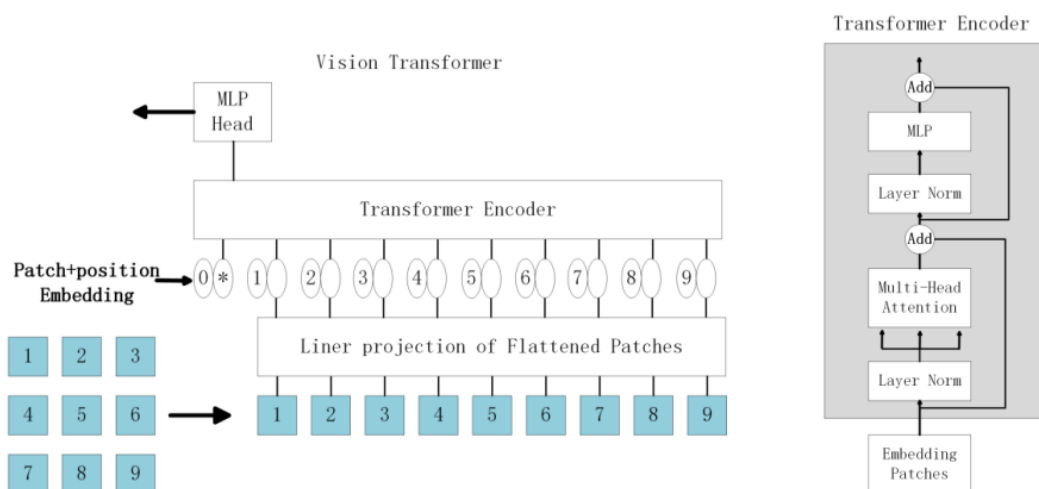


图 2.12 Vision Transformer 模块

### 2.3.3 Swin-Transformer 算法

Swin-Transformer 是微软研究院在 2021 年发表在 ICCV 上的一篇文章上提出的<sup>[47]</sup>。Swin-Transformer 网络模型把 Transformer 模型应用于视觉领域，在多项机器视觉任务中展现出良好性能，该模型在传统的 COCO 数据集上目标检测提升 2.7 个点，实例分割提升 2.6 个点。

Swin-Transformer 所构建的网络模型特征图是有层次的，与传统的卷积神经网络十分类似，通过特征层的不断加深特征图的宽和高越来越小，正是因为这一特征 Swin-Transformer 在目标检测上有良好的表现，该模型通过窗口对特征图进行分割，然后在每一个窗口中进行特征提取，窗口与窗口之间不进行信息交互，这样的好处在于可以降低模型的运算量，Swin-Transformer 模型的结构如图 2.13 所示。

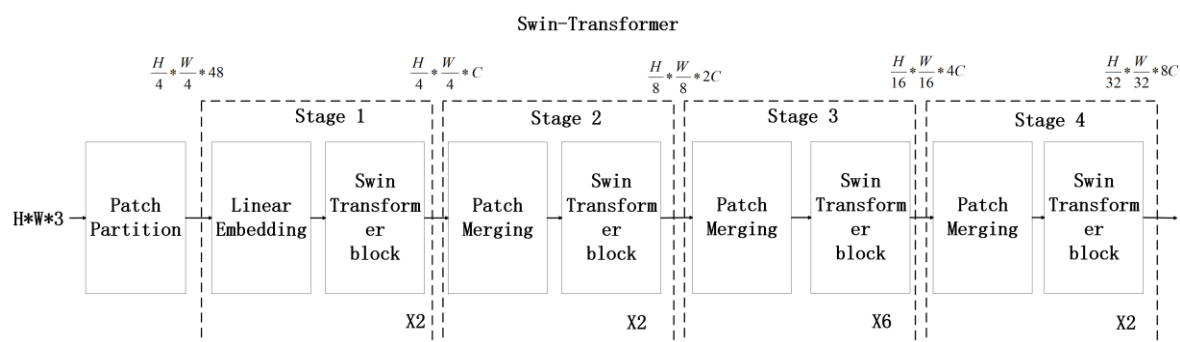


图 2.13 Swin Transformer 模块

Swin-Transformer 模型中的 patch merging 层本质是做一个下采样操作，对输入的特征图进行像素划分，然后对每一个 patch 中相同位置的元素进行 concat 拼接，将得到的新的特征图进行深度方向拼接，然后通过一个 LN 层，最后通过全连接将特征图的深度方向进行线性变换，得到的新特征图高宽减半，深度翻倍。

W-MSA(Windows Multi-head Self-Attention)模块的引入是为了减少模型的计算量，加快模型计算速度，在普通的 MSA(Multi-head Self-Attention)模块中，需要对特征图中的每个 patch 和其他的所有 patch 进行计算。在 W-MSA 中先对特征图按照一定大小划分成一个个窗口，然后每个窗口中的 patch 和其他 patch 进行计算，窗口与窗口之间不进行信息交互。两者计算量如公式(2.22)所示。

$$\begin{aligned}\Omega(MSA) &= 4hwc^2 + 2(hw)^2C \\ \Omega(W-MSA) &= 4hwc^2 + 2M^2hwC\end{aligned}\quad (2.22)$$

其中： $h, w, C$  分别是特征图的高度，宽度，深度， $M$  是每个窗口的大小。

SW-MSA(shifted Windows Multi-head Self-Attention)即偏移的 W-MSA，这个模型的提出主要是解决在 W-MSA 中窗口与窗口之间不进行信息传递的问题，即将特征图

在 W-MSA 的基础上向右侧和下侧偏移 $\frac{M}{2}$ 个像素。这样就可以解决窗口之间互补交互的问题。但是这样处理会使原本四个窗口变成九个窗口，然后对每个窗口计算又使本来简单的任务变得麻烦起来，因此对得到的新窗口加上标识然后拼接起来变为原来的四个窗口进行计算，经过计算之后再还原回原来的样子。具体流程如图 2.14 所示。

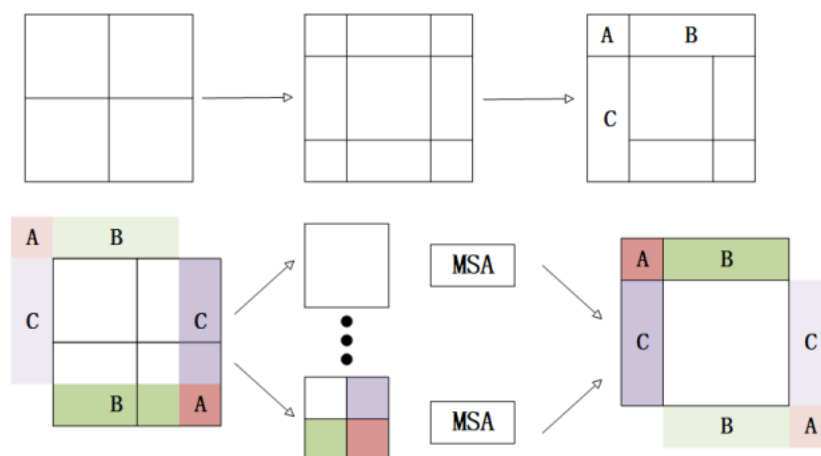


图 2.14 SW-MSA 操作流程

Layer Normalization是针对自然语言处理任务提出来的，在Swin-Transformer的模型中首先对输入的图像进行序列化处理，处理后的时序长度不是一个定值，LN会对每个序列进行标准化处理。而Batch Normalization是对一批数据进行标准化处理，在每个序列的batch不是统一长度时很难合理的计算训练数据的均值和方差，如果使用BN为了补齐长短不一的序列而添加元素，会使较长序列中的元素转为向量时规模相对变小，较短序列中的元素转换为向量时规模相对变大，增加了误差抖动。因此在Swin-Transformer模型中使用LN层网络。

## 2.4 Mobile net 系列算法

### 2.4.1 Mobile Net-v1

Mobile Net 网络是 google 团队在 2017 年提出的<sup>[48]</sup>，专注于嵌入式设备中的轻量级网络 cnn 网络。相比传统卷积神经网络，Mobile Net 网络能够在确保准确率的前提下大大减少模型的参数与运算量，因此本文选择该模型来改良 Yolov5 网络模型。

Mobile Net-v1 网络与传统的卷积神经网络 VGG16 相比，Mobile Net-v1 网络的分类准确率降低了 0.9%，但 Mobile Net-v1 网络参数的数量只有 VGG16 神经网络的 1/32，Mobile Net-v1 网络在同等资源的前提下可以发挥出更好的网络性能。Mobile Net-v1

网络是由 11 层的传统卷积层和 14 层的深度可分离卷积层和平均池化层加上全连接层与 softmax 层组合而成。

在 Mobile Net-v1 网络中最关键的技术就是深度可分离卷积 (Depthwise Separable Convolution), 深度可分离卷积主要由逐点卷积 (Pointwise Convolution) 和逐通道卷积 (Depthwise Convolution) 两个部分组成, 在 DW 卷积中卷积核的深度不再等于输入特征矩阵的深度而是等于 1, 输出特征矩阵的深度不再等于卷积核的个数而是等于输入特征矩阵的深度等于卷积核的个数, 这样可以确保每一个卷积核负责一个深度, 既通过 DW 卷积之后特征矩阵是不会发生深度上的改变。在 PW 卷积中卷积核的深度等于输入特征矩阵的深度, 输出特征矩阵的深度等于卷积核的个数。通常 DW 卷积与 PW 卷积一起使用, 与传统的卷积相比能节省很大一部分的计算量, 深度可分离卷积如图 2.15 所示。

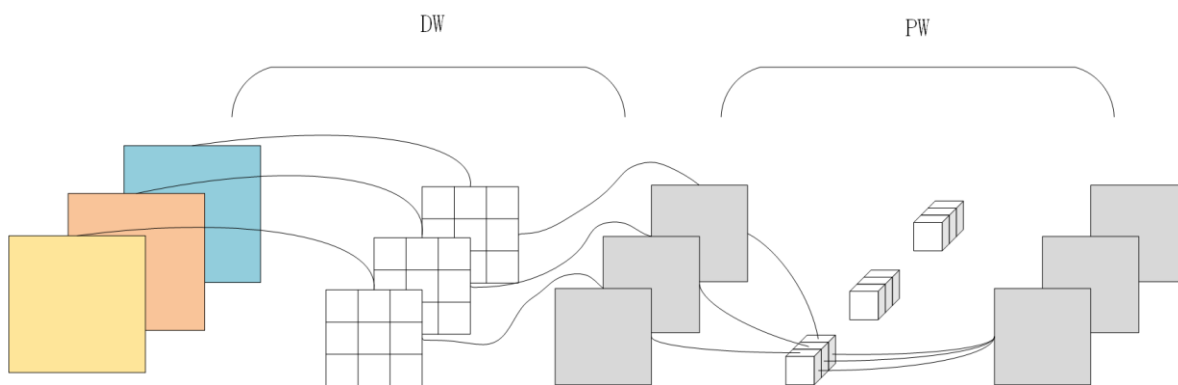


图 2.15 深度可分离卷积

下面对深度可分离卷积的计算量和普通卷积的计算量进行具体的分析, 普通卷积的计算量如公式 (2.23) 所示。

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \quad (2.23)$$

其中,  $D_K$  是卷积核的大小,  $D_F$  代表的是输入特征矩阵的宽和高, 假设这里的宽和高相等,  $N$  是输出特征矩阵的深度,  $M$  是输入特征矩阵的深度。

在深度可分离卷积中 DW 卷积的计算量如公式 (2.24) 所示。

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F \quad (2.24)$$

PW 卷积的计算量如公式 (2.25) 所示。

$$M \cdot N \cdot D_F \cdot D_F \quad (2.25)$$

则深度可分离卷积的计算量如公式 (2.26) 所示。

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \quad (2.26)$$



这里假设stride等于1，在Mobile Net网络中DW卷积使用的是3x3大小的卷积核。所以理论上普通卷积的计算量是Depthwise+Pointwise卷积计算量的9倍，公式如(2.27)所示。

$$\begin{aligned} & \frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} \\ &= \frac{1}{N} + \frac{1}{D_K^2} \\ &= \frac{1}{N} + \frac{1}{9} \end{aligned} \quad (2.27)$$

## 2.4.2 Mobile Net-v2

2018年Google公司在Mobile Net-v1网络的基础上对网络进行优化，提出了Mobile Net-v2网络<sup>[49]</sup>，Mobile Net-v1网络在训练后会出现Depthwise卷积中卷积核大部分为零，针对这一问题在Mobile Net-v2中提出倒残差结构，倒残差结构是在残差结构的基础上进行改良得到，当某些层学习效果非常差时，可通过残差块将这些层的权重参数设置为0，从而直接跳过这一部分<sup>[50]</sup>。

残差结构是通过1x1卷积对输入特征矩阵进行压缩减少输入特征矩阵的深度，再通过大小为3的卷积核进行卷积处理，最后使用大小为1的卷积核来进行扩充增加特征矩阵的深度，在倒残差结构中，与残差结构正好相反，先使用1x1的卷积进行升维操作使深度变得更深，接下来使用大小为3的卷积核进行深度可分离卷积，最后使用1x1的卷积进行降维处理，在残差结构中的激活函数使用ReLU激活函数，在倒残差结构中使用的是ReLU6激活函数，原因是因为ReLU激活函数对低维特征信息会造成大量的损失，残差网络如图2.16所示，倒残差网络如图2.17所示。

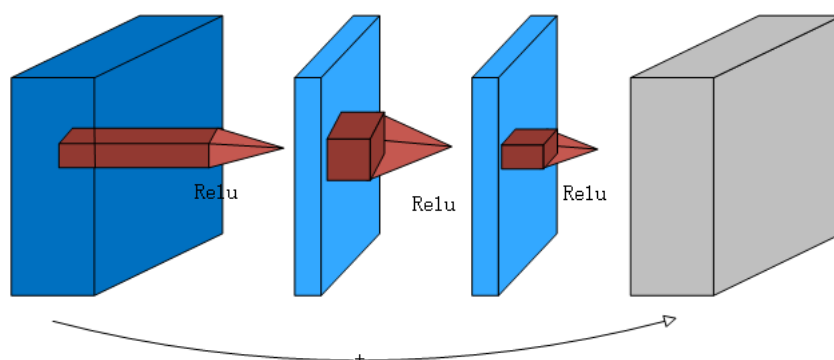


图 2.16 残差网络

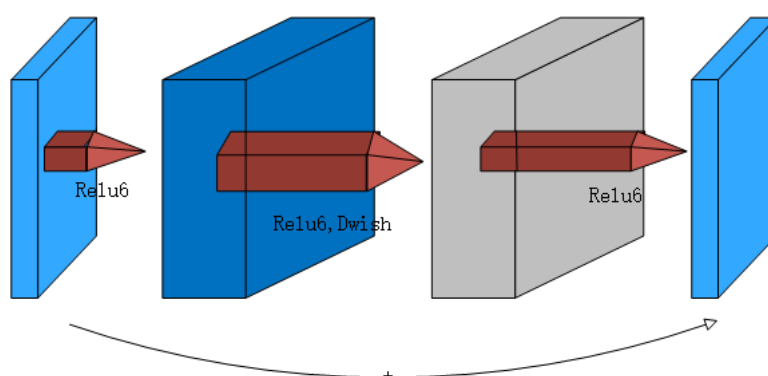


图 2.17 倒残差网络

### 2.4.3 Mobile Net-v3

Mobile Net-v3 神经网络是在 2019 年发表的<sup>[51]</sup>，它结合了 Mobile Net-v1 的深度可分离卷积和 Mobile Net-v2 网络的倒残差结构，并加入了 SE 模块，SE 模块是注意力机制(CBAM)的一种，它主要有两个操作，一是针对得到的特征矩阵的每一个通道进行全局平均池化处理，经过全局池化之后得到一个一维向量，其元素等于特征矩阵的通道数。二是对得到的一维向量进行两个全连接层操作输出向量。经过第一个全连接层操作后，节点的个数等于原特征矩阵通道数的 1/4，经过第二个全连接层操作后，节点个数等于特征矩阵的通道数，输出的向量会对特征矩阵的每一个通道计算出一个权重系数，对特征图中比较重要的通道信息赋予更大的权重，不重要的通道信息赋予较小的权重，SE 模块如图 2.18 所示。

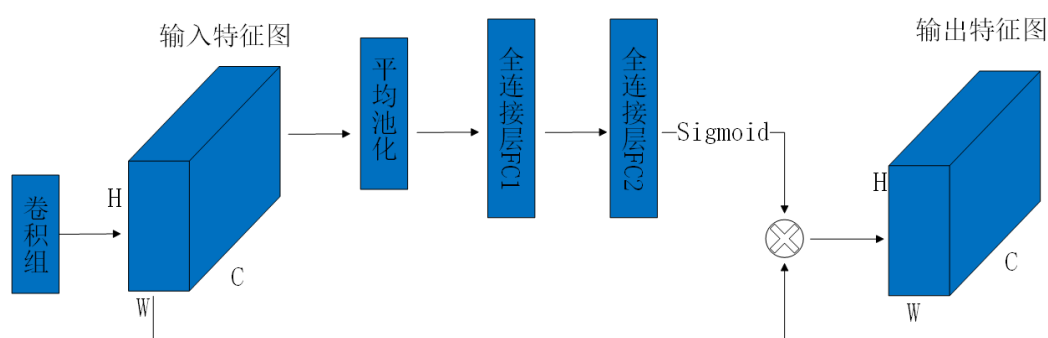


图 2.18 SE 模块

同时在 Mobile Net-v3 网络中重新设计并使用 H-Swish 激活函数，同时为了解决 Mobile Net-V2 网络模型中参数过多难以调优的问题，MobileNet-V3 网络模型通过使用 NAS 来搜索参数，优化神经网络每层的超参数，对模型进行优化。并重新设计了耗时层的结构，在 Mobile Net-v3 网络中将第一层的卷积核个数由原来的 32 降低到 16 但是准确率并不会下降，精简网络结构中的最后阶段，即将平均池化层前移到第一个卷积之后再跟两个卷积层输出，和原始的网络结构相比少了很多的层结构，调整



之后的准确率没有发生什么变化，但缩短了 11% 的推理时间，提升运行效率，原始网络最后一层与 Mobile Net-v3 网络中最后一层对比如图 2.19 所示。

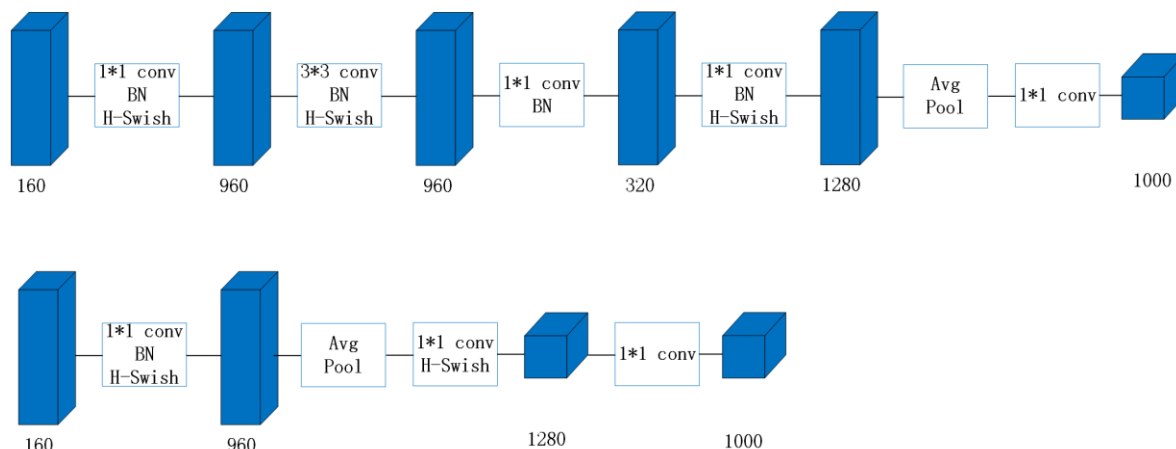


图 2.19 Mobile Net-v3 最后一层对比图

## 2.5 Yolo 系列算法

### 2.5.1 Yolov3 算法

Yolov3 算法<sup>[52]</sup>在 Yolov2 算法上进行了大量的优化，在 Yolov2 中使用的模型框架是 DarkNet-19，在 Yolov3 中使用了性能更强大的 DarkNet-53，由 52 层卷积层和最后一层的全连接层组成主体的特征提取网络结构，在 Yolov3 中没有使用最大池化下采样层，所有的下采样都是通过步距为 2 卷积层来实现，其准确率与 ResNet-152 接近。Yolov3 会在三个特征层上进行预测，每个特征层上会使用三个尺度，这三种尺度通过 k-means 得到共九组，分别为 10\*13, 16\*30, 33\*23, 30\*61, 62\*45, 59\*119, 116\*90, 156\*198, 373\*326。

在 Yolov3 中目标边界框的预测比 SSD 有一些改进，当 1\*1 的卷积层滑动到预测特征层的某一个窗口时，会对每一个先验框模板都预测四个回归参数  $t_x$ ,  $t_y$ ,  $t_w$ ,  $t_h$ ，一个 objectness 参数，和对应的每个类别的分数，其中虚线的矩形框所对应的就是先验框， $p_w$  和  $p_h$  分别为先验框的宽和高，蓝色的矩形框是网络预测的最终的位置 ( $b_x$ ,  $b_y$ ) 及宽  $b_w$  和高  $b_h$ ，公式如(2.28)所示。

$$\begin{aligned}\sigma(x) &= \text{Sigmoid}(x) \\ b_x &= \sigma(t_x) + c_x, b_y = \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w}, b_h = p_h e^{t_h}\end{aligned}\tag{2.28}$$

其中 $c_x$ 和 $c_y$ 分别为当前cell左上角点的坐标(红标), 通过 $c_x$ 和 $c_y$ 和Sigmoid函数可以计算出预测边界框的中心点坐标, 其中Sigmoid的作用就是将预测的边界框中心限制在当前的cell中。目标边界框预测图如图所示 2.20。

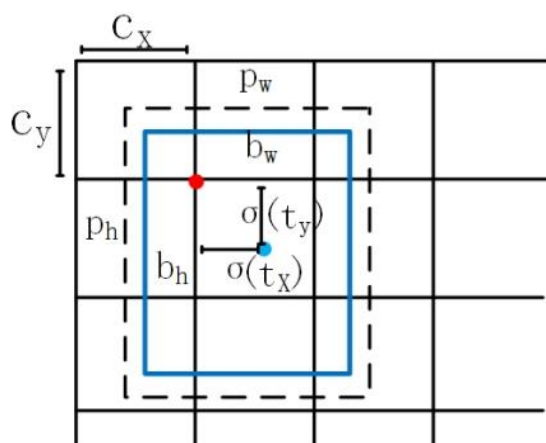


图 2.20 目标边界框预测图

在模型中对每一个真实值都会分配一个先验框, 一张图片有几个真实目标就会有几个正样本, 与真实值重合程度最大的称为正样本, 重合程度为超过设定阈值的为负样本, 重合值超过阈值但不是最大的, 既不是正样本也不是负样本, 会被丢弃, 这样的处理会使正样本减少。为解决这一问题, 后续版本中重合值超过阈值都会被分为正样本。

## 2.5.2 YOLOv4 算法

在 YOLOv4 中将 DarkNet-53 网络模型改良为 CSPDarkNet-53 网络结构<sup>[53]</sup>, 其中 CSP 结构可以增强 CNN 的学习能力, 消除计算瓶颈并降低对显存的使用, CSP 的加入可加快网络的推理速度, 提高网络的学习能力, CSP 网络结构是将输入平均分为两个分支, 第一个分支是步长为 1 的  $1 \times 1$  卷积核进行 ConvMish 卷积, 第二个分支是经过步长为 1 的  $1 \times 1$  卷积核进行 ConvBNMish 卷积, 经过一系列的 ResBlock, 最后再经过步长为 1 的  $1 \times 1$  卷积核进行 ConvBNMish 卷积并将输出与第一个分支进行 concat 拼接, 其中 ConvBNMish 是由卷积层加 BN 层加 Mish 激活函数构成, ResBlock 是  $1 \times 1$  的卷积层加上  $3 \times 3$  的卷积层和输入进行 Add 相加。

在 YOLOv4 中加入了 SPP 模块, 它的位置在 CSPDarkNet-53 与第一个预测特征层之间, SPP 模块在一定程度上借鉴的是空间金字塔模块 (FPN) 的思想, SPP 模块利用最大池化层融合不同感受野的特性, 假设它的输入是  $13 \times 13$  深度为 512, 第一个分支是输入直接接到输出, 第二个分支会经过一个卷积核为  $5 \times 5$  的最大池化下采样层,

第三个分支是经过一个卷积核为  $9 \times 9$  的最大池化下采样层，第四个分支是经过一个卷积核为  $13 \times 13$  的最大池化下采样层，其中步距都是 1 即输入和输出的高宽与深度不发生改变，最后对三次最大池化结果和未进行池化操作的数据进行拼接，经过拼接后的通道数变为输入的 4 倍，SPP 模块的加入实现不同尺度的特征融合，该模块融合了全局特征和局部特征，丰富了特征图的信息，在一定程度上解决目标多尺度的问题。Spp 结构如图 2.21 所示。

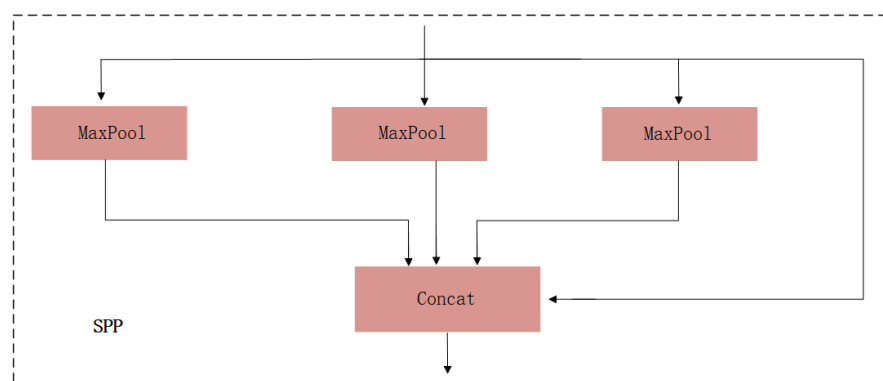


图 2.21 spp 结构图

PAN 结构的思想来源于 path aggregation network 这篇文章，CSPDarKnet-53 网络通过主干网络进行特征提取后，在提取的特征层上构建一个将高层的语义信息向低层进行融合的 FPN 模块，再构建一个将低层的语义信息向高层进行融合模块，将两个模块进行 Concat 连接就是 PAN 结构，PAN 结构图如图 2.22 所示。

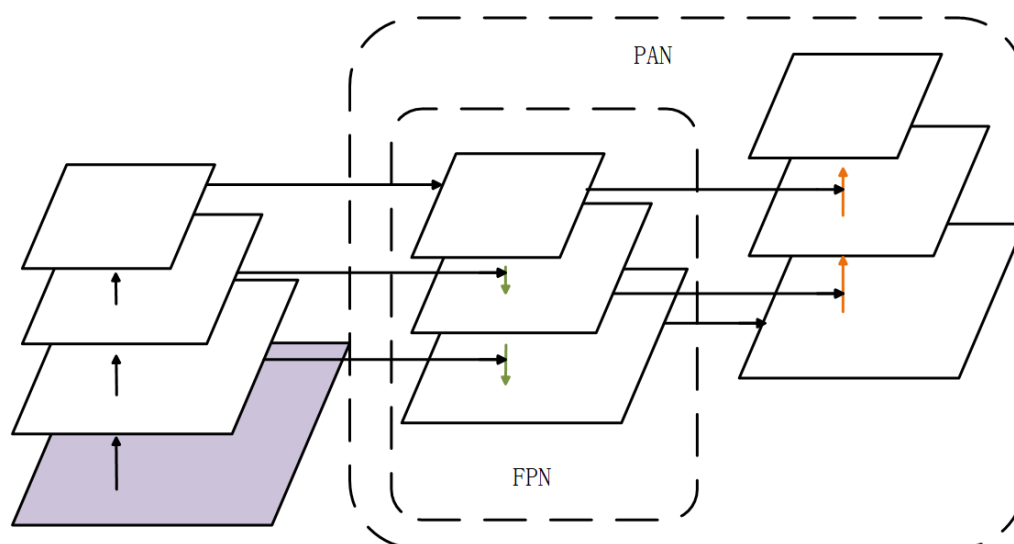


图 2.22 PAN 结构图

在 Yolo v3 中，当真实目标中心点非常靠近网格的左上角点或者右下角点时，网络的预测值需要负无穷或者正无穷时才能取到，而这种很极端的值在网络中一般无法达到。为了解决这个问题，在 YOLOv4 中引入一个大于 1 的缩放系数  $scalexy$ ，公式

如(2.29)所示。

$$\begin{aligned} b_x &= \left( \sigma(t_x) \cdot scale_{xy} - \frac{scale_{xy} - 1}{2} \right) + c_x \\ b_y &= \left( \sigma(t_y) \cdot scale_{xy} - \frac{scale_{xy} - 1}{2} \right) + c_y \end{aligned} \quad (2.29)$$

### 2.5.3 YOLOv5 算法

在 YOLOv5 中将 SPP 结构优化为 SPPF，在 SPP 中是将输入的特征层并行的通过大小不同的 MaxPool2d，将他们的输出进行融合，但是在 SPPF 网络结构中是将输入的特征层串行的依次通过三个卷积核为 5\*5 的最大池化层，再将他们的输出进行 Concat 连接，这样做可以将前一个 5\*5 的最大池化层的输出作为下一个 5\*5 的最大池化层的输入，效果等同于 SPP 中卷积核为 9\*9 的最大池化层，但是可以提升网络的运行效率，SPPF 网络结构如图 2.23 所示。

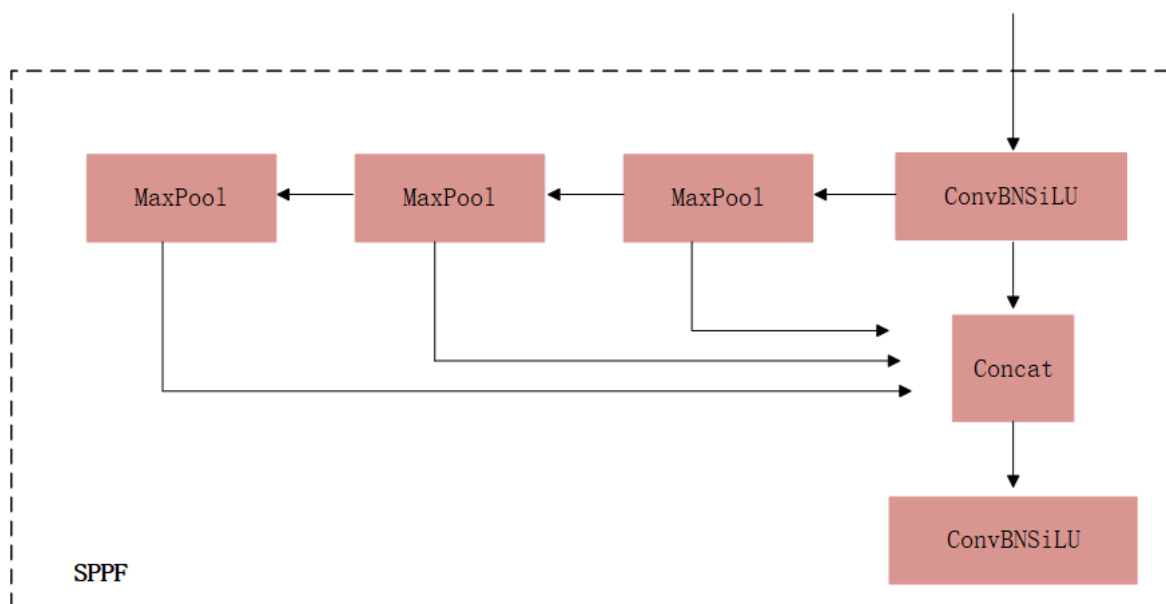


图 2.23 sppf 结构图

在 YOLOv5 中使用很多种方式对图像进行增强，第一种方式是 Mosaic 图像增强，他的第一个优点是通过对随机的四张图片拼接在一起进行组合，增加数据的多样性。第二个优点是若单独对某一张图片进行训练，其目标个数是很少的，但是将多张照片拼接在一起会增加训练目标的个数，第三个好处是 BN 层可以一次统计多张照片的参数。第二个图像增强的方式是对图像进行 Random affine 操作，即对图像进行随机的旋转，缩放，平移和剪切。第三个方式是对图像进行色度，饱和度以及明亮度的调整。

在实际的使用过程中由于设备内存资源和设备计算能力的有限性，将网络模型

布署到这些设备上存在一定的技术难度，而 Yolov5 网络模型在参数量上和计算量上比其他模型都有较好的优势，在检测速度和精确率上也有较好的表现，更适合嵌入到平台进行使用，因此本文选择以 Yolov5 为基础的网络模型作为番茄病害系统的基础模型。

## 2.6 本章小结

本章主要阐述了本文使用到的理论基础，首先，阐述了神经网络中卷积层、池化层、激活函数和全连接层的结构和相关理论，接着对本文检测算法中相关的Yolo系列算法、Transformer系列算法、Mobile Net系列算法的发展历程、网络结构及其特点进行了系统分析，为后续番茄叶部病害检测模型的构建提供了理论及技术支撑。



## 第三章 基于轻量级 YOLOv5 的番茄叶部病害检测模型

### 3.1 概述

数据集是模型研究的基础，由于研究需求的特殊性，采用自己采集、预处理、标注进而构建数据集的方法。在此基础上，重点研究了基于轻量级 YOLOv5 的番茄叶部病害检测模型，为降低网络模型的参数量，提高网络模型的计算速度，同时确保网络模型的准确率，使番茄病害检测可以快速准确运行并反馈给用户，将传统 YOLOv5 网络模型的主干网络替换为 Mobile Net-v3 的 Bneck 块来降低模型参数量，加入 CBAM 注意力机制，以加强空间信息与通道信息间的联系，并将传统 YOLOv5 模型中的 C3 结构替换为 C2f 模块，在加深网络模型深度的同时缓解梯度消失问题。本章首先介绍了番茄叶部病害图像的采集、预处理及人工标注标签并最终构建数据集的过程，然后详细分析了轻量级 YOLOv5 网络模型的改进原理，阐述了模型的设计方案及 Bneck 模块、C2f 模块、CBAM 注意力机制、损失函数的详细设计方法，最后对改进的网络模型进行消融实验和对比试验，实验结果证明所构建的轻量级 YOLOv5 网络模型降低了传统 YOLOv5 网络模型的参数量，提高了模型计算速度，番茄病害检测准确率也有所提升。

### 3.2 番茄病害数据集构建

轻量级 YOLOv5 网络模型是基于学校物联网与大数据平台的应用需求，开发基于平台的番茄叶部病害检测的应用算法，因此公开的数据集无法满足具体的应用需求，鉴于研究的特殊性，采用自己采集、预处理、标注进而构建数据集的方法。数据集的构建主要分为数据采集、图像增广、标签标注等步骤

#### (1) 数据采集

为了使数据集能够反映出各时期、各种病害类型的特征，使样本具有代表性，选择各类各个时期具有不同叶部病害特征的番茄植株进行图像采集，图 3.1 是番茄叶部病害样本。图像采集条件按照正常使用时的条件设定，主要是在自然光条件下进行拍摄，主要设备为手机，拍摄距离为 50cm 左右。为确保检测模型的性能，共采集了番茄细菌性斑点病、番茄早疫病、番茄花叶病毒病、番茄鱼目斑病、番茄褐斑病、二斑叶螨、番茄黄化曲叶病毒七类病害图片和健康的番茄叶部图片，每类番茄病害

都拍摄一千张图片，共计八千张番茄叶部病害图片作为数据集。



图 3.1 番茄叶部病害样本

## （2）图像增广

图像增广是通过对获取的数据集通过几何变换，图像模糊，噪声抑制，裁剪缩放，明亮度调整等技术来扩充数据集，使数据集中的图片特征变得更加多样性<sup>[54]</sup>。图像增广的作用主要体现在以下几个方面：一是增加图像数据的数量。通过对图像的随机几何变换，可以生成更多且多样化的图像数据，增加数据集的规模。通过几何变换后可增加更多的数据特征，使目标检测模型在训练时能更好地进行学习。二是提高目标检测模型的鲁棒性。鲁棒性是指一个系统或模型在面对不同的干扰、噪声或变化时能够保持稳定性和正确性的能力，在计算机视觉领域中，鲁性通常指的是模型对于图像中的各种变化和扰动的适应能力<sup>[55]</sup>。目标检测模型在图像在经过随机几何变换，比如旋转，裁剪，缩放等操作后进行训练，可以增强模型对图像特征的位置，角度，形变等学习能力，使模型更具有鲁棒性，有助于提升模型在未知数据上的表现能力。三是降低模型过拟合。过拟合是模型在验证数据集上表现良好准确率较高，但是在测试集上表现较差准确率较低的现象<sup>[56]</sup>。通过几何变换增加数据的多样性，减少该模型对数据集的依赖性，可以降低模型的过拟合问题。四是改善模型的泛化能力。泛化能力是指模型对未知数据的适应能力，即在测试集中准确率越高，说明模型的泛化能力越好<sup>[57]</sup>。图像增广可以使模型学习到训练集数据中更多不同的特征，使其在面对未知的数据时具有更好的泛化能力，通过随机几何变换操作可以增加模型对于不同角度和镜像变换的识别能力。

本文的数据增广没有采用传统方式，而是在开始训练模型时，先对输入的图像进行几何变换、裁剪缩放和明亮度调整处理，将处理后的图像输入到模型网络中，每训练一轮都进行一次数据增广操作，这样的操作可以扩充数据集、提高模型的鲁



棒性、降低模型的过拟合，并改善模型的泛化能力，从而提升模型的总体性能。

### (3) 标签制作

本文模型需要 coco 格式和 voc 格式两种标签，因此在打标签时没有选择传统的 Labellmg 或者 LabelMe 这两种工具，这两种工具在对图像进行标注时只能进行有规则的图形标注，同时由于数据量很大，在做数据标签时十分耗时，而且这两种打标签工具没有预加载功能，一旦开始打标签就要从第一张一直标注到最后一张，一旦在中间停止标注并关闭这两种标注工具，在下次开始打标签时就会造成标签混乱，需要从新开始制作标签，同时标签内的名字和图片名字也会产生误差，在训练时可能会造成找不到该图片的错误。

本文在进行打标签时使用的是一种半自动打标签工具 EISeg，该工具是由飞浆研发的一款标注软件，通过加载网络参数对需要标注的部分进行标注，标注后的 coco 格式包含了标签的名称和各个点的坐标，因为是对每一个图片进行单独标注工作，在对所有图片标注完成之后将产生的 json 文件 and 对应图片按照 1: 2: 7 的比例划分为验证集，测试集与训练集，再将每个数据集中的 json 文件合并为一个 json 文件。Voc 格式包含了标签的 id，番茄病害种类和各个点的坐标，同理按照比例进行划分，然后将 json 文件通过代码转换为 txt 格式，这样的操作都是为了方便模型后续进行训练与学习。标注工具如图 3.2 所示。

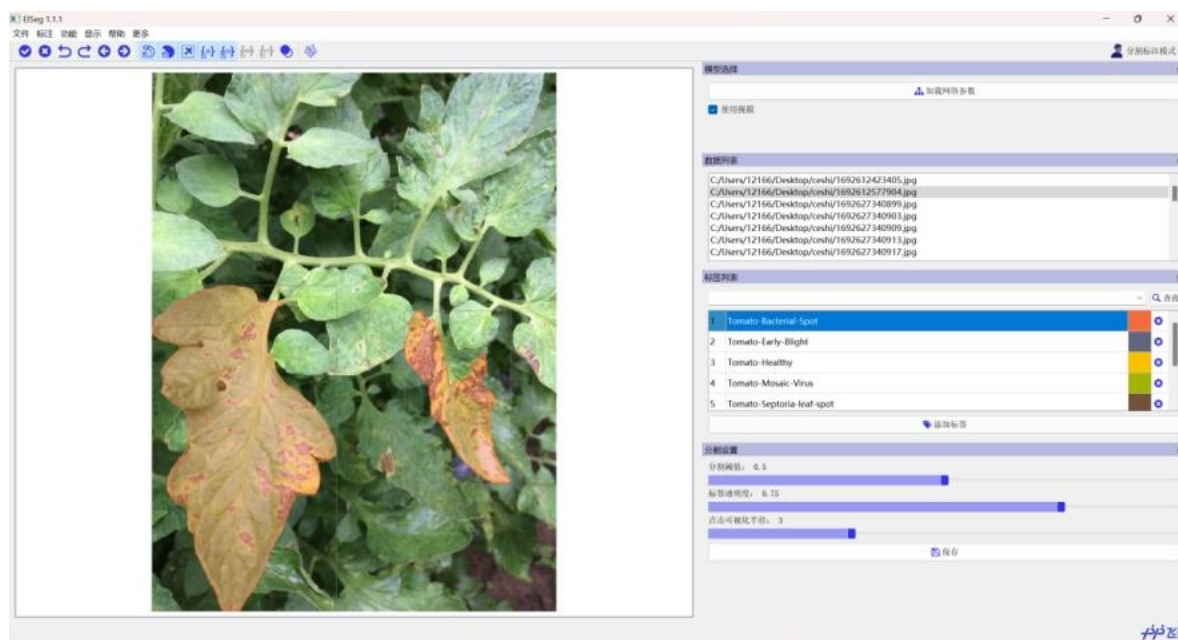


图 3.2 图片标注工具

## 3.3 轻量级 YOLOv5 网络模型

### 3.3.1 Bneck 模块

在用神经网络模型对图像进行特征提取时，随着卷积层的叠加，卷积模型的深度加深，参数量增多，网络模型的训练更为耗时。使用轻量级网络来对传统 Yolov5 模型进行改进，能使模型参数量和计算量减少，因此采用 Mobile Net-v3 中的 Bneck 块来代替 Yolov5 中的 backbone，Bneck 模型的参数量为 720150 个，每秒浮点运算次数（FLOPs）为 0.69G，而 Yolov5 中 backbone 的模型参数量为 4303040 个，每秒浮点运算次数（FLOPs）为 10.61G，经计算可知：改进后的模型参数量比原模型参数量减少了约百分之八十三，模型的复杂度大大降低；通过 FLOPs 的对比可知，改进后模型的计算速度明显提高。

Bneck 模块借用了残差网络模型的原理，并加入注意力机制模块，该模块可以对图像的特征信息进行较好的特征提取，其网络结构如图 3.3 所示。

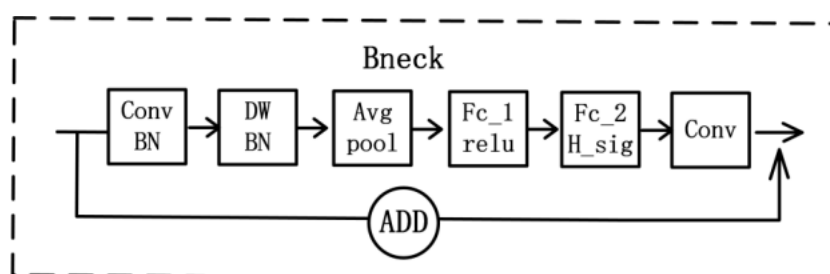


图 3.3 Bneck 网络结构

用上述的 Bneck 模块进行一系列特征提取之后，进行多尺度特征融合操作，该操作可以扩大感受野，方便不同的输入尺寸都能有较好的信息提取，并对不同尺度特征图的信息进行融合，最后接上由 FPN 与 PAN 结构组合的特征金字塔模型，模型整体框架如图 3.4 所示。

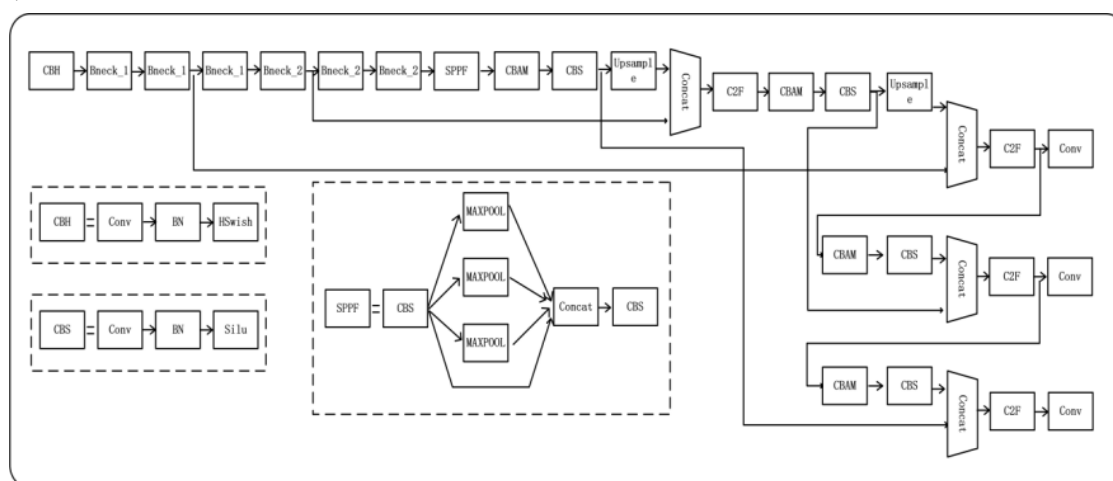


图 3.4 轻量级网络模型结构

### 3.3.2 C2f 模块

C2f 模块相比于 YOLOv5 中的 C3 模块更为轻量化，在 C3 模块中主要是采取对上一层的输出进行一个分流，并结合残差结构的思想设计而来，C2f 模块中采用多堆叠的思想可以减少模型整体的计算量。具体的实现步骤是将第一次卷积的通道数扩充为原来的两倍，然后将卷积结果在通道数上进行平均分割，其中第一部分不做任何卷积处理，第二部分采取堆叠思想进行三次叠加，将第二部分得到的输出与第一部分进行拼接操作。本文将所有的 C3 模块都替换成更为轻量化的 C2f 模块，这样可以大大加快网络的速度，同时经过如此多的堆叠其实也对应了更密集的残差结构，残差网络结构的特点是使神经网络模型更容易优化，同时 C2f 模块中的密集残差结构可以增加模型的深度，在一定程度上提高模型的准确率，同时 C2f 模块中的残差块之间使用了连续跳跃，在一定程度上缓解了由于网络模型中模型深度的增加而带来的梯度消失问题。C2f 模块网络模型结构如图 3.5 所示。

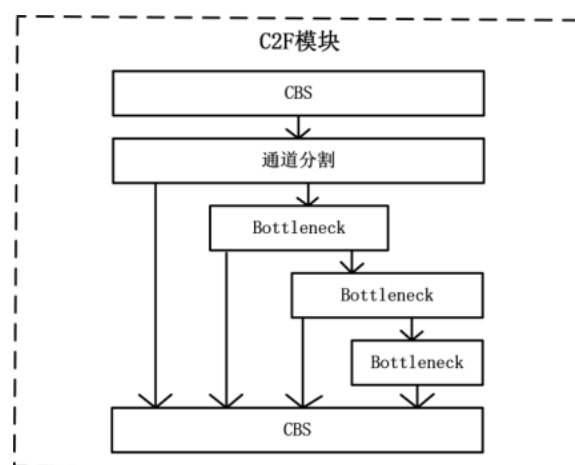


图 3.5 C2f 模块网络模型结构

### 3.3.3 CBAM 注意力机制

在神经网络中随着网络层数的叠加，深层网络的感受野会不断增加，远大于浅层网络的感受野。感受野越大使网络能够获取的图像特征信息越多，信息特征越全面，会影响处理速度，而且不重要的信息也会随着感受野的增加而相对增加。实际中番茄叶部特征通常是整片叶片的局部信息，感受野的增大不利于更加深入捕获特征，并且大量的正常信息在识别病害特征时容易造成模型过拟合或识别率降低，因为大的感受野中的大部分信息都是与病害无感的信息，病害特征变化对识别结果影响被正常信息稀释，不利于病害特征识别。借鉴人的注意力原理，即把人的注意力集中在感兴趣的信息上，能够提高兴趣点中信息的挖掘及处理能力。神经网络中的

注意力机制通过计算判别不同特征通道的重要程度并分配不同的权重，从而描述不同局部特征的重要性，筛选出重要的局部特征过滤掉不需要的特征，并把感受野集中在该特征上，挖掘局部和细节特征，提高神经网络的处理效能。

CBAM 是一种结合了空间和通道的注意力机制模块，它相比 SENet 只关注通道的注意力机制有更好的效果，CBAM 包含 CAM (Channel Attention Module) 和 SAM (Spatial Attention Module) 两个子模块，这两个子模块通过串联的方式连接在一起，对输入特征图分别实施了通道上的注意力机制和空间上的注意力机制。这样的网络结构可以减少网络模型的参数和计算力，同时保证了该网络模块能够作为即插即用的模块，可以随时集成到现有的网络架构中去，CBAM 注意力模块如图 3.6 所示。

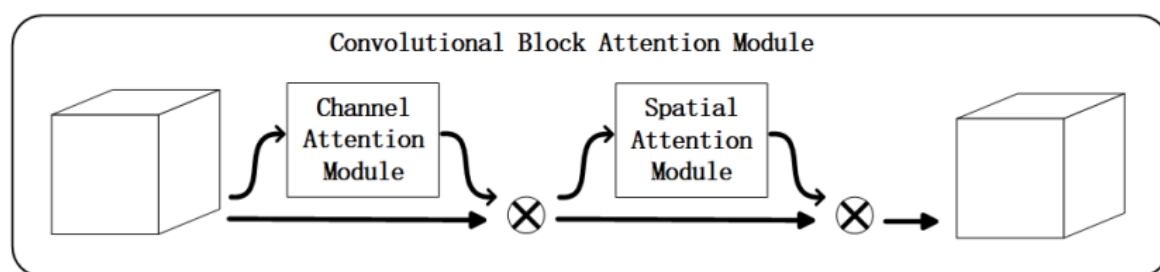


图 3.6 CBAM 注意力模块

CBAM 注意力模块的前半部分模块为通道注意力模块 (CAM)，它的作用是关注通道的特征是有意义的，通道注意力机制的实现可以分为三个部分，首先对输入的特征图分别进行全局平均池化和全局最大池化，然后将经过平均池化和最大池化层得到的结果输入到两层共享全连接层 (shared mlp)，这两层的共享全连接层的参数是相同的；第二步对共享全连接层输出的特征图进行逐元素相加，然后通过 Sigmoid 激活函数得到输入特征图每一个通道的权重值，该权重的取值范围在 0 到 1 之间，最后将得到的权重系数与原输入特征图相乘，得到最终的特征图。CAM 操作流程如图 3.7 所示。

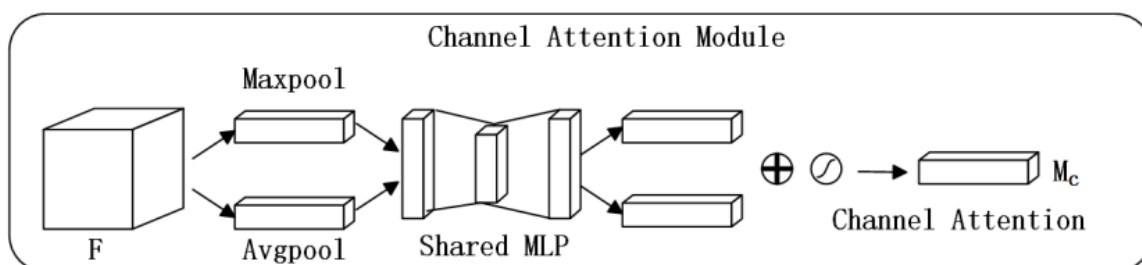


图 3.7 CAM 流程图

由上述可知通道注意力的公式如(3.1)所示，其中 $\sigma$ 为 Sigmoid 激活函数，F 为输入的特征图， $W_1$  和  $W_0$  为共享的权重值， $F_{avg}$  为输入的特征图 F 经过平均池化层操作

后得到的特征图， $F_{\max}$  为输入的特征图  $F$  经过最大池化层操作后得到的特征图， $M_c$  为最终得到的特征图。

$$M_c(F) = \sigma \left( MLP(AvgPool(F)) + MLP(MaxPool(F)) \right)$$

$$M_c(F) = \sigma \left( W_1 \left( W_0(F_{avg}) \right) + W_1 \left( W_0(F_{max}) \right) \right) \quad (3.1)$$

CBAM 注意力模块的后半部分为空间注意力模块 (SAM)，在 SAM 中会对输入的特征图在同一个通道维度上分别进行最大池化和平均池化操作，之后将这两个结果在通道维度上拼接起来，将拼接起来的特征图通过一个卷积核大小为  $7*7$  的卷积操作，将通道数降为 1，同时保持特征图大小不发生改变，然后再通过 Sigmoid 激活函数生成空间权重系数 (0-1 之间)，最后将得到的权重系数与原输入特征图相乘，得到最终的特征图。SAM 操作流程如图 3.8 所示。

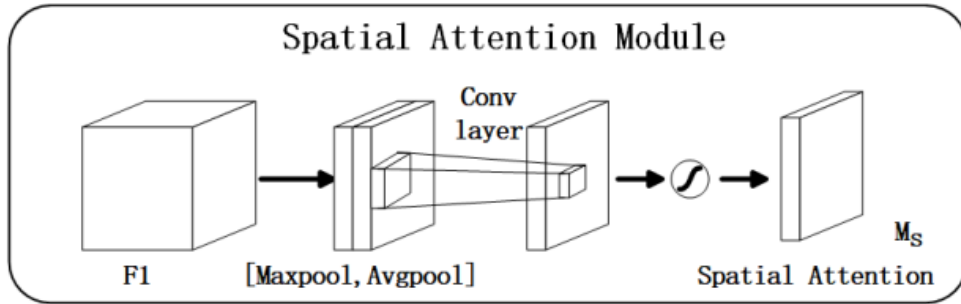


图 3.8 SAM 流程图

由上述可知空间注意力的公式如(3.2)所示。

$$M_s(F) = \sigma \left( F^{7 \times 7} ([AvgPool(F1); MaxPool(F1)]) \right)$$

$$M_s(F) = \sigma \left( F^{7 \times 7} ([F1_{avg}; F1_{max}]) \right) \quad (3.2)$$

其中 $\sigma$ 为 Sigmoid 激活函数， $F1$  为输入的特征图， $F7 \times 7$  代表卷积核大小为  $7*7$  的卷积操作， $F1_{avg}$  为输入的特征图  $F$  经过平均池化层操作后得到的特征图， $F1_{max}$  为输入的特征图  $F$  经过最大池化层操作后得到的特征图， $M_s$  为最终得到的特征图。

### 3.3.4 损失函数

本文设计的损失函数由三部分组成，即分类损失 (Classes loss)，Obj 损失以及定位损失 (Location loss)。

分类损失 (Classes loss) 采用的是二值交叉熵损失 (BCE loss)，分类损失函数公式如(3.3)所示。

$$L_{cls}(O, C) = \frac{\sum_{i \in pos} \sum_{j \in cls} (O_{ij} \ln(\hat{C}_{ij}) + (1 - O_{ij}) \ln(1 - \hat{C}_{ij}))}{N_{pos}}$$

$$\hat{C}_{ij} = Sigmoid(C_{ij}) \quad (3.3)$$

其中  $O_{ij} \in [0,1]$ ，即预测目标边界框  $i$  中是否存在第  $j$  类目标， $C_{ij}$  为预测值， $\hat{C}_{ij}$  为  $C_{ij}$  通过 Sigmoid 激活函数得到的目标概率， $N_{pos}$  为正样本的总个数。

Obj 损失 (Objectness loss) 指的是网络预测的目标边界框与真实边界框的 CIOU，采用二值交叉熵损失 (BCE loss)，Obj 损失函数公式如(3.4)所示

$$L_{obj}(O, C) = -\frac{\sum_{i \in pos} O_i \ln(\hat{C}_i) + (1 - O_i) \ln(1 - \hat{C}_i)}{N_{pos}}$$

$$\hat{C}_i = Sigmoid(C_i) \quad (3.4)$$

其中  $O_i \in [0,1]$ ，即预测目标边界框与真实目标边界框的 CIOU 值， $C$  为预测值， $\hat{C}_i$  是通过 Sigmoid 函数得到的预测置信度， $N_{pos}$  为正负样本的个数。

定位损失 (Location loss) 采用的是 CIOU loss，定位损失函数如(3.5)所示。

$$L_{loc} = 1 - CIOU$$

$$CIOU = IoU - \left( \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \right) \quad (3.5)$$

$$v = \frac{4}{\pi^2} \left( \arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2, \alpha = \frac{v}{(1 - IoU) + v}$$

其中  $\rho$  代表的是  $b$  与  $b^{gt}$  之间的欧氏距离，其中  $b$  是预测目标边界框的中心的坐标， $b^{gt}$  是真实目标边界框的中心点的坐标， $c$  是预测目标边界框与真实目标边界框最小外接矩形的对角线长度， $w^{gt}$  与  $h^{gt}$  是真实目标边界框的长与宽， $h$  与  $w$  是预测目标边界框的长与宽。

因此本文算法的总损失函数是由分类损失，Obj 损失以及定位损失相加而来，总损失函数如公式(3.6)所示。其中， $\lambda_1, \lambda_2, \lambda_3$  为平衡系数。

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc} \quad (3.6)$$

## 3.4 模型结果与分析

### 3.4.1 模型环境搭建

神经网络模型训练过程中需要大量的参数，激活函数的值，和一些缓冲区，其中每一个值都将决定下一层的值，每一次的迭代训练都会将这些值进行更新，这需要大量的缓冲区，如果采用传统的 CPU 为内核的训练方式所需的训练时间很长，甚至可能出现超出计算机的内存的情况。常用的解决方法是使用 GPU 对模型训练，

GPU 具有一定显存并可以进行并行操作, 提高训练速度。本文采用 GPU 对模型训练环境配置: GeForce RTX4060, 13th Gen Intel(R) Core(TM) i9-13900H, 主频为 2.60 GHz, 内存为 32G, 解释器为 python 3.8, 实验平台为 PyCharm, cuda 版本为 12.0。

超参数设置: 训练图片分辨率 640\*640\*3, 迭代次数 100, 批大小为 16, 优化器为 SGD, 初始学习率为 0.01, 周期学习率为 0.01, 权重衰减为 0.0005。

### 3.4.2 评价指标

为了对算法性能进行定量评估, 采用精确率(precision), 召回率(recall), 均衡平均数 (F-score) 和平均精度均值 (mAP) 来对模型进行衡量, IOU 是目标检测过程中预测框与真实框的交集部分与并集之间的比, 假设 A 为预测框, B 为真实框那么 IOU 的公式如(3.7)所示。

$$IOU = \frac{A \cap B}{A \cup B} \quad (3.7)$$

由公式可以看出 IOU 的值越大, 预测框与真实框的重合度越好, 模型的预测也就越加准确, 反之模型预测能力较差。

**精确率(precision):** 精确率又称查准率, 它是指正样本中经过预测后被判定为正的样本的总数量占正样本中经过预测被判定为正样本的总数量加上负样本中经过预测被判定为正样本的数量两者的总和的比值, 即预测正确的正样本数量占有所有预测结果为正样本数量的比值, 其中 TP 指实际为正样本经过预测后被判定为正样本的数量, FP 为实际为负样本但经过预测被判定为正样本的数量, 公式如(3.8)所示。

$$P = \frac{TP}{TP + FP} \quad (3.8)$$

**召回率(recall):** 召回率又称查全率, 它是针对原样本而言的, 它的含义是指正样本中经过预测被判定为正的样本总数量占正样本中经过预测被判定为正样本数量加上正样本中经过预测被判定为负样本数量两者的总和的比值, 即正确预测正样本数量占有所有正样本的比值, 其中FN为正样本中被预测为负样本的数量, 公式如(3.9)所示。

$$R = \frac{TP}{TP + FN} \quad (3.9)$$

**均衡平均数 (F-score):** 是指精确率与召回率的调和平均, 当准确率和召回率越高表示模型越好, 但是这两种情况同时表现很高的值的情况是矛盾的, 因此为了综合精确率和召回率, 引入均衡平均数, 公式如(3.10)所示。

$$F_{\beta} = (1 + \beta^2) \frac{P * R}{\beta^2 * P + R} \quad (3.10)$$

当 $\beta=1$  时, 即为F1-Score, F1 是精确率和召回率的调和均值, 公式如(3.11)所示。

$$F_1 = 2 \frac{P * R}{P + R} \quad (3.11)$$

均值平均精确率 (mAP): 是指对所有的类别的平均精确率 (AP) 求均值。AP 就是 PR 曲线下的面积, 是对不同召回率上的精确率进行平均, 当 PR 曲线的面积越接近 1, 即 AP 的值越高时表示该模型训练效果越好, 同时精确率与召回率在整体上也有较好的表现。AP 表征的是每个类别的精确率, 精确率与召回率是一对相此消彼长的性能指标, 当精确率的值较高时, 此时召回率的值可能并不高, 因此在 PR 曲线中右上角的凸出越好模型效果越好。mAP 反应整个模型的精确率, mAP 的值越大目标检测模型越好。

### 3.4.3 消融实验

在对整体模型评估前, 为了验证C2f模块, CBAM模块和Bneck结构这三种改良策略对原模型性能的增益, 本文设计了共计 5 组消融实验, 通过与原模型的对比来评估每个模块的性能, 每次增加或者减少模块后的实验环境保持一致, 实验中的所有消融实验都是在番茄病害数据集上进行, 输入图片尺寸为 640\*640, 本次的消融实验以参数量 (params), 每秒浮点运算次数 (FLOPs), 精确率(precision), 召回率(recall) 和平均精度均值 (mAP) 五个评价指标作为评估标准来对每次改进后的模型进行衡量, 实验结果如表 3.1 所示。

表 3.1 轻量级网络消融实验

Method				Params/M	FLOPs/G	P/%	R/%	mAP@.5/%
Yolov5	Bneck	CBAM	C2f					
√				7.2	16.6	93.8	85.7	93.3
√	√			1.9	3.2	88.4	83.5	92.8
√		√		7.7	17.5	90.5	86.8	93.6
√			√	7.4	17.2	92.6	88.2	94.3
√	√	√		2.4	4.1	92.1	91.5	93.4
√	√	√	√	2.6	4.7	95.1	94.1	94.9



由表 3.1 可知在传统 Yolov5 模型上将主干网络改进为 Bneck 网络之后, 参数量降低了 74%, 每秒浮点运算次数降低了 81%, mAP 下降了 0.5%, 和原模型相比准确率及模型训练效果变差, 特征提取阶段的效果不明显, 但计算量和参数量大大降低, 很大程度上加快模型计算速度; 在原模型中单独加入 CBAM 模块后参数量多了 0.5M, 每秒浮点运算次数增加了 0.9G, 同时 mAP 指上升了 0.3%, 相比于原网络该模块的增加是用计算量来换取较好的训练效果; 在原模型中单独加入 C2f 模块后参数量比原模型多了 0.2M, 每秒浮点运算次数增加了 0.6G, 同时 mAP 指标上升了 1.0%, 说明该模块的堆叠结构可以提高模型的性能; 在用 Bneck 替换原网络的主干网络并加入 CBAM 块之后, 和原模型相比参数量降低了 67%, 每秒浮点运算次数降低了 76%, mAP 提高了 0.1%, 可以看出在保证实验效果变化不大的情况下, 计算速度增快, 参数量降低明显; 当原模型的主干网络用 Bneck 模块替换并加入 CBAM 和 C2f 模块后可以看到, 与原模型相比参数量降低了 64%, 每秒浮点运算次数降低了 71%, mAP 上升了 1.6%。

在番茄病害数据集上进行的消融实验表明, 不同的改进方式组合不会使改进后的模型性能进行线性增加, 在单独将 Yolov5 中的主干网络替换成轻量级的网络后虽然 mAP 值会降低, 但参数量会降低 74%同时每秒浮点运算次数降低了 81%, 在加入 C2f 模块和 CBAM 注意力机制后, mAP 值提升 1.6%, 在降低计算量的同时提高模型的准确率, 说明 CBAM 注意力机制的加入可以使模型过滤掉不相干的信息, 提高网络模型的关注度, C2f 模块使用了堆叠结构可以加深网络模型的深度, 同时加快计算速度, 并在残差机构上使用连续跳跃的方法, 在一定程度上缓解了由于网络模型中深度的增加而带来的梯度消失问题。由此可见 Bneck 模块, C2f 模块和 CBAM 模块的加入有利于提高网络模型的目标检测能力, 说明提出的方法是有效的。

### 3.4.4 轻量级 Yolov5 模型性能分析

为了分析本文改进后的轻量级 Yolov5 模型的性能, 现提出的轻量级 Yolov5 模型与主流模型的性能进行对比, 主要对比模型为目标检测中的一阶段模型和二阶段模型, 其中二阶段模型为主干网络分别以 Res50 残差网络 Res101 残差网络为主干网络的 Faster Rcn 网络模型, 主干网络分别以 Res50 残差网络 Res101 残差网络为主干网络的 Mask-Rcn 网络模型, 一阶段模型包括以主干网络为 Vgg16 的 SSD 网络模型, 传统 Yolov5 网络模型, RetinaNet 网络模型和 MobileVit 网络模型, 各网络性能指标如表 3.2 所示。

表 3.2 轻量级网络对比试验

Method	Params/M	FLOPs/G	mAP@.5/%	F1
Faster Rcnm-Res50	41.14	123	82.3	81.70
Faster Rcnm-Res101	60.14	154	83.1	82.90
Mask-Rcnm-Res50	43.77	135	81.6	81.00
Mask-Rcnm-Res101	62.76	158	81.2	81.00
SSD-vgg16	24.98	50.5	83.4	82.74
Yolov5	7.2	16.6	93.3	89.57
RetinaNet	36.9	75.8	88.4	88.00
MobileVit	55.18	128.5	83.9	84.25
轻量级Yolov5	2.6	4.7	94.9	94.60

由表 3.2 可知本文构建的轻量级 Yolov5 网络在参数量上在所有模型中是最少的，比原 Yolov5 模型的参数量降低 64%，FLOPs 在所有的对比模型中是最少的，通过参数量和 FLOPs 两个值说明本文改良后的轻量级 Yolov5 网络在进行番茄叶部病害检测时，因为参数量的减少，模型计算复杂度的降低从而提高模型的检测效率与检测速度，经过对比实验可以分析出本文轻量级 Yolov5 网络 mAP 值相比于其他模型有较好的表现，比原模型提高了 1.6%，F1 值为 94.6，比其他模型的 F1 值都要好，说明本文设计的轻量级 Yolov5 网络模型在进行番茄病害检测时比其他模型进行病害检测有较好的检测性能提高，病害检测准确率要优于其他网络模型。

番茄细菌性斑点病、番茄早疫病、健康叶片的、番茄花叶病、番茄鱼目斑病、番茄褐斑病、二斑叶螨、番茄黄化曲叶病的精确率(precision)，召回率(recall)和准确率(Accuracy)如表 3.3 所示。在轻量级网络检测过程中，所有番茄病害类召回率最高的为番茄黄化曲叶病，番茄褐斑病和健康叶片这意味着这几个类的正确预测样本数量占有所有正样本的比值为 96%。精确率较高的为番茄花叶病和健康叶片，从准确率上来看番茄早疫病和番茄花叶病以及健康叶片的准确率达到 90%以上。

表 3.3 番茄病害各类指标

番茄病害种类	P/%	R/%	Acc/%
番茄细菌性斑点病	95.7	89	85.6
番茄早疫病	95.9	94	90.4
健康叶片	97.0	96	93.2

表 3.3 番茄病害各类指标(续)

番茄病害种类	P/%	R/%	Acc/%
番茄花叶病	96.0	95	91.3
番茄鱼目斑病	93.2	96	89.7
番茄褐斑病	91.8	89	82.4
二斑叶螨	91.3	94	86.2
番茄黄化曲叶病	91.4	96	88.1

轻量级 Yolov5 混淆矩阵的左侧代表了番茄病害预测的全部类别，矩阵中每一列的总数表示该类番茄病害的总数，每一行的值表示预测为该类病害的数量，由于数据量较大因此对结果进行归一化处理，使数值在 0 到 1 之间，轻量级 Yolov5 目标检测模型的混淆矩阵如图 3.9 所示。

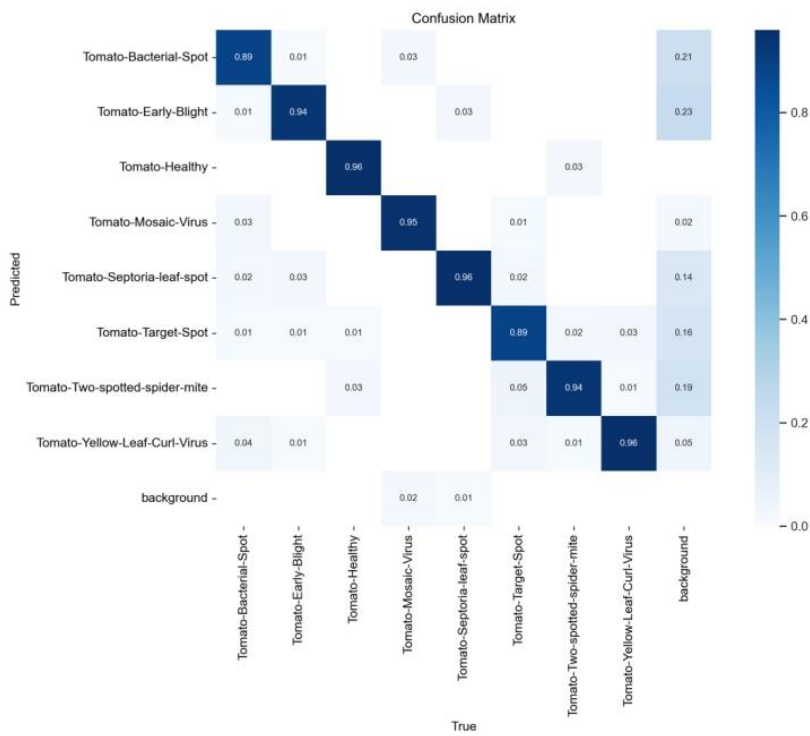


图 3.9 轻量级网络混淆矩阵

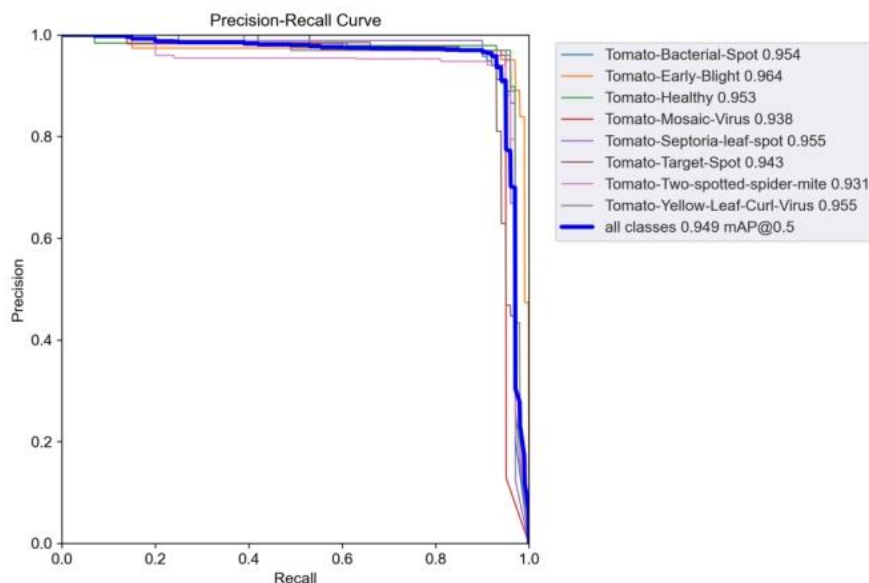


图 3.10 轻量级网络 P-R 曲线

P-R 曲线所围成的面积就是平均精确率（AP）的值，其中番茄细菌性斑点病的 AP 值为 98%，番茄早疫病病的 AP 值为 94.6%，番茄健康叶片的 AP 值为 95.6%，番茄花叶病毒病的 AP 值为 94.3%，番茄鱼目斑病的 AP 值为 92.9%，番茄褐斑病的 AP 值为 96.1%，二斑叶螨的 AP 值为 95.7%，番茄黄化曲叶病毒的 AP 值为 96.1%，经计算平均精度均（mAP）为 94.9%比原模型的平均精度均值（mAP）提高了 1.6%。轻量级目标检测模型的 P-R 曲线如图 3.10 所示。

用采集到的番茄叶部病害图片对训练好的轻量级 Yolov5 网络模型进行验证，结果如图 3.11 所示。验证结果表明，本文构建的轻量级 Yolov5 网络可以检测出番茄病害的种类和准确率。



图 3.11 轻量级网络检测结果

### 3.5 本章小结

本章首先阐述了数据集的采集、预处理及标签标注过程，然后详细分析了轻量级 Yolov5 网络模型的改进原理，以及采用 Bneck 模块、C2f 模块、CBAM 注意力机制、损失函数改进传统 Yolov5 网络模型的详细设计过程，最后对轻量级 Yolov5 网络模型的性能进行分析。分析结果表明：提出的基于 Yolov5 的轻量级网络模型能在保证番茄叶部病害检测准确率的情况下，将传统 Yolov5 模型的参数量降低为原模型的约 36%，每秒浮点运算次数下降到原模型的约 29%。为了加快模型的计算速度，将原模型的 C3 模块改为 C2f 模块，该模块采用堆叠思想将原模型的三次卷积改为三次堆叠结构，这种操作可以加深模型的深度，在一定程度上可以降低由于模型深度带来的梯度消失问题，同时加入 CBAM 注意力机制，该模块可以提高对输入图像特征的关注度，通过消融实验证明这两个模块对网络模型的性能有明显的提升，通过对比实验证明改良后的模型在番茄叶部病害识别上比原模型和其他模型都有较好的表现，并对改良后的轻量级网络实验结果进行分析。



## 第四章 基于 Swin-Yolov5 的番茄叶部病害检测模型

### 4.1 概述

实际检测过程中由于受到自然光线或者叶片之间互相遮挡,会导致拍摄的照片中番茄叶部病害特征不明显或者特征缺失,从而导致检测结果不理想甚至会漏检。为解决这一问题,本章研究了基于 Swin-Yolov5 的番茄叶部病害检测模型。该模型第一个改进是用 Swin-Transformer 模型替换掉 Yolov5 中的主干网络,并将输入图像分割成若干个小块,分别提取每个块的特征,通过这种方法可提取到更多番茄叶部病害信息,提高网络模型的检测精度;在分块提取特征时,块与块之间不进行交互,只通过位置编码对块进行标记,后续使用 W-MSA 模块和 SW-MSA 模块实现块与块之间的交互,以降低运算量。Swin-Yolov5 模型的第二个改进是将原模型中的上采样替换为轻量级上采样算子 Carafe,获取更大感受野,提取更多输入图像特征,并能根据输入特征进行优化重组,Carafe 算子的轻量级特点,确保以非常小的额外参数及计算量,带来网络整体性能的显著提升。本章首先阐述了 Swin-Transformer 模型的特点及用 Swin-Transformer 优化 Yolov5 的原理及设计过程,然后介绍了上采样算子 Carafe 的功能特点及设计方法,最后对模型性能进行分析评估。

### 4.2 Swin-Yolov5 网络模型

#### 4.2.1 Swin-Yolov5 模型设计

Swin-Yolov5 网络模型主要是用 Swin-Transformer 模型替换掉原 Yolov5 模型中的主干网络, Swin-Transformer 模型的优势在于将输入图片分割成一个个的小块,然后对分割后得到的图片分别进行特征提取,此时块与块之间不进行信息交互,在后续的特征提取中使用 W-MSA 模块和 SW-MSA 模块,实现块与块之间信息交互,以降低运算量,提高网络模型的检测精度。加入了注意力机制、C2f 模块以及轻量级算子的模型相比于原网络结构,可以获取更大的感受野,提取更多的病害特征信息,以增加少量参数量或计算量为代价,提高模型的整体性能。Swin-Yolov5 模型整体设计如图 4.1 所示。

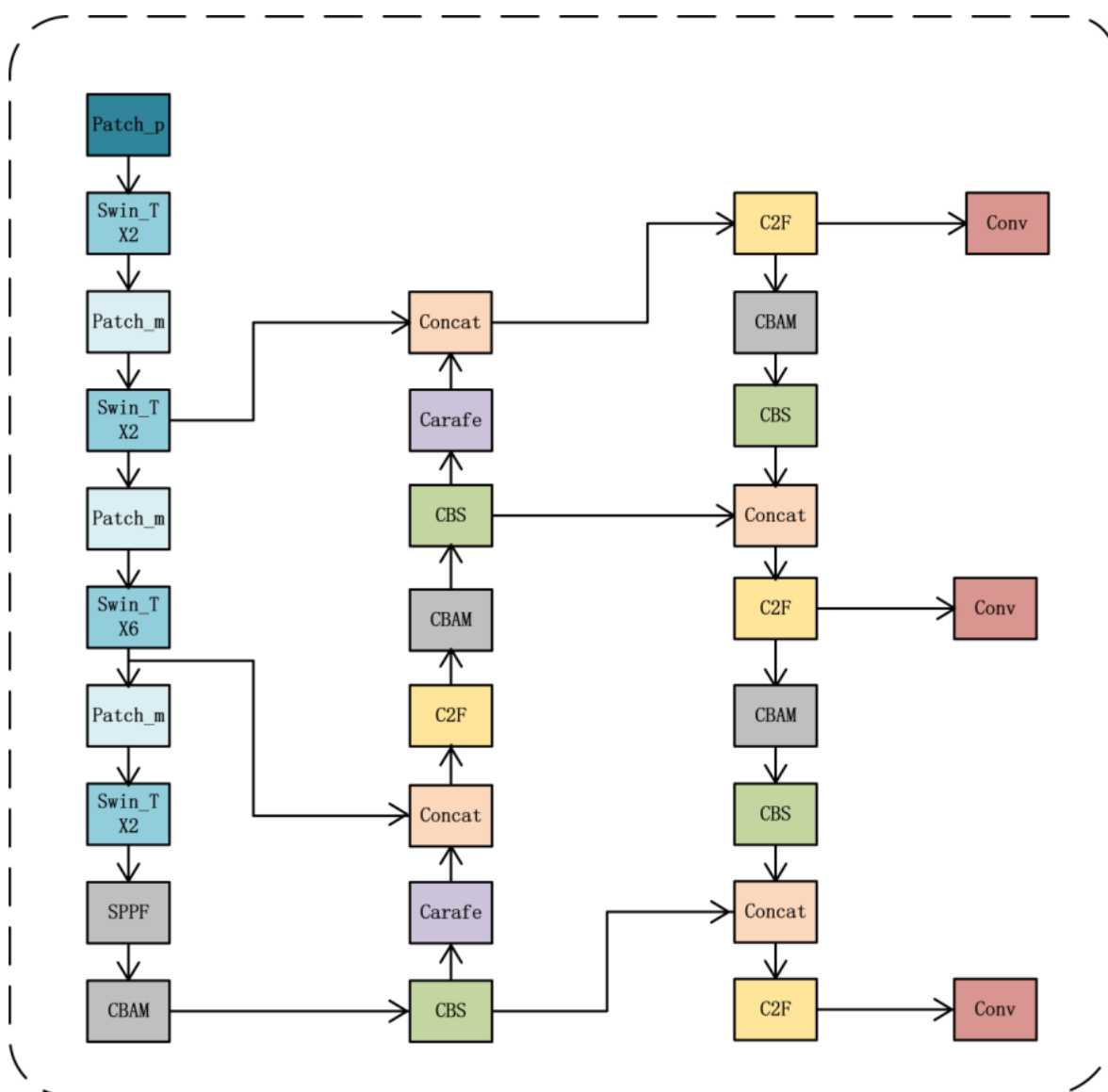


图 4.1 Swin-Yolov5 网络模型

### 4.2.2 轻量级上采样算子

上采样操作是对输入的特征图与上采样相对应的像素进行点积操作，与原模型中的最邻近上采样相比，本文中使用的轻量级上采样算子 **Carafe** 可以在进行上采样的时候获得较大的感受野，以对输入的特征图特征进行更好的提取，并且会根据输入的特征进行更好的重组；同时 **Carafe** 算子是轻量级的，可以通过很小的额外参数与计算量来给整个网络带来较好的提升。

**Carafe** 算子主要由两个模块构成，分别是特征重组模块和上采样核预测模块。假设上采样倍率为 $\sigma$ ，对于输入的特征图为  $H \times W \times C$ ，首先经过 **Carafe** 算子的上采样核预测模块对输入特征图的卷积核进行预测，对输入的特征图的内容进行感知处理，



并通过感知处理生成不同的卷积核，而不是像原模型中的上采样层对所有样本都使用固定的卷积核，然后使用特征重组模块完成最终的上采样操作，并输出大小为  $\sigma H * \sigma W * C$  的特征图，Carafe 算子操作流程如图 4.2 所示。

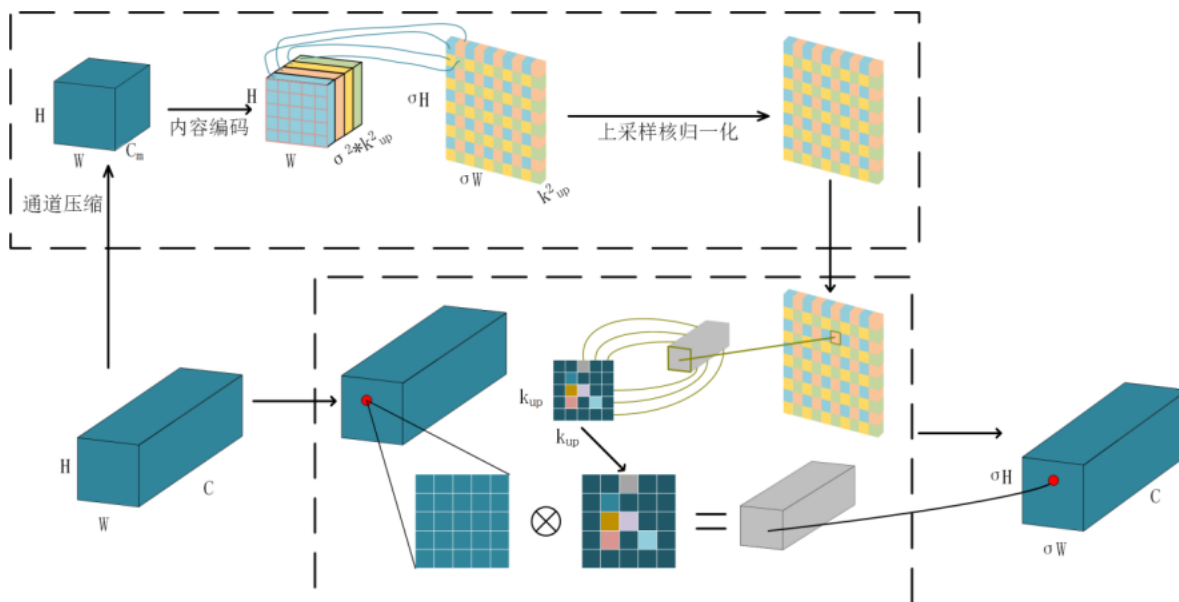


图 4.2 Carafe 算子操作流程

Carafe算子中的上采样核预测模块首先对输入大小为 $H*W*C$ 的特征图进行卷积核大小为 $1*1$ 的卷积操作，将输入特征图的通道进行压缩，压缩后的大小为 $H*W*C_m$ ，通过对输入的特征图进行压缩可以降低减少上采样核预测的计算量，假设要得到的上采样核预测的大小为 $\sigma H* \sigma W*k_{up}^2$ ，首先对压缩后的特征图进行encoder操作，即对压缩后的特征图进行编码操作，假设输入特征图的通道数为 $C_m$ ，则输出的特征图通道数为 $\sigma^2 k_{up}^2$ 。接着将输出的通道维度在空间维度展开，得到大小为 $\sigma H* \sigma W*k_{up}^2$ 的上采样核，最后对得到的上采样核用softmax函数进行归一化操作，使卷积核权重为1，上采样核预测模块操作如图 4.3 所示。

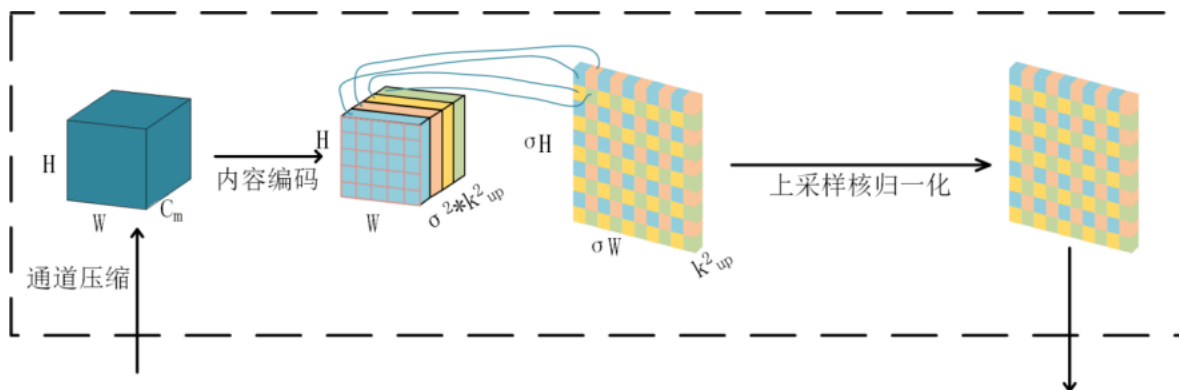


图 4.3 上采样核预测模块

Carafe算子中的特征重组模块，会对输出特征图中的每个位置，将其映射回输入特征图，然后取出大小为 $k_{up}^2 * k_{up}^2$ 的区域，将其与在上采样核预测模中预测出该区域部分的上采样核进行点积操作，相同位置不同通道使用同一个上采样核，然后将得到的所有特征模块进行重组操作，得到大小为 $\sigma H * \sigma W * C$ 的特征图。特征重组模块操作如图 4.4 所示。

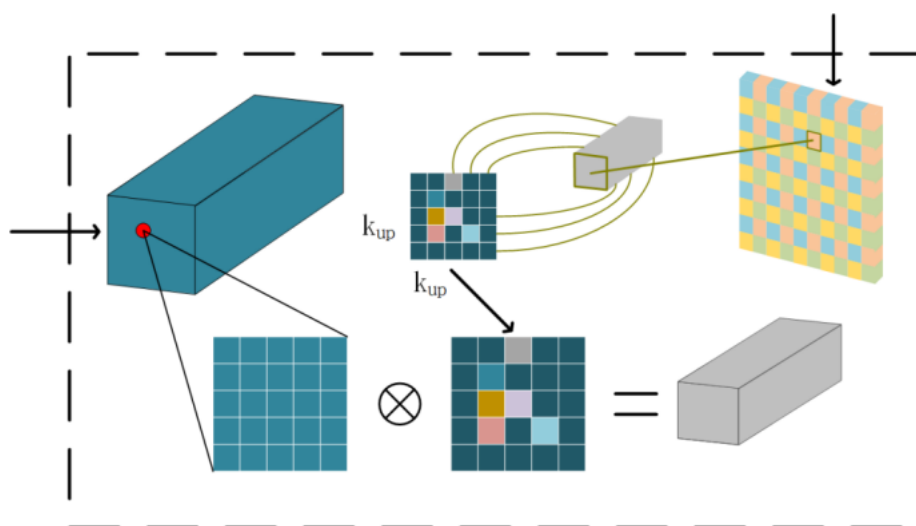


图 4.4 特征重组模块

## 4.3 模型性能评估

### 4.3.1 消融实验

在对整体模型评估前，为了验证C2f模块，CBAM模块，Swin结构和轻量级上采样算子这四种改良策略对原模型性能的提升，本文设计了共计 7 组消融实验，通过与原模型的对比来评估每个模块的性能，实验中的所有消融实验都是在番茄病害数据集上进行，实验环境一致，输入图片尺寸为 640\*640，通过参数量 (params)，每秒浮点运算次数 (FLOPs)，精确率 (precision)，召回率 (recall) 和平均精度均值 (mAP) 作为评估标准来对改进后的模型进行衡量，实验结果如表 4.1 所示。

表 4.1 Swin-Yolov5 网络消融实验

Method					Params/ M	FLOPs/G	P/%	R/%	mAP@.5/%
Yolov5	Swin	CBAM	C2f	Carafe					
√					7.2	16.6	93.8	85.7	93.3
√	√				29.1	75.9	94.4	93.5	95.8

表 4.1 Swin-Yolov5 网络消融实验（续）

Method					Params/ M	FLOPs/G	P/%	R/%	mAP@.5/%
Yolov5	Swin	CBAM	C2f	Carafe					
√		√			7.7	17.5	90.5	86.8	93.6
√			√		7.4	17.2	92.6	88.2	94.3
√				√	7.28	16.7	92.2	85.4	93.4
√	√	√			29.11	75.9	93.1	91.5	94.4
√	√	√	√		29.37	76.4	95.3	94.5	95.8
√	√	√	√	√	29.45	76.4	97.4	96.7	96.8

由表 4.1 可知在Yolov5 模型上将主干网络改进为Swin网络之后，参数量提高了 75.3%，每秒浮点运算次数提高了 78.2%，mAP提高了 2.5%，和原模型相比准确率及模型训练效果变强，特征提取阶段的效果十分明显，但计算量和参数量大大增加，很大程度上影响了模型计算速度；在原模型中单独加入CBAM模块后参数量多了 0.5M，每秒浮点运算次数增加了 0.9G，同时mAP指上升了 0.3%，相比于原网络该模块的增加是用计算量来换取较好的训练效果；在原模型中单独加入C2f模块后参数量比原模型多了 0.2M，每秒浮点运算次数增加了 0.6G，同时mAP指上升了 1.0%，说明该模块的堆叠结构可以提高模型的性能；加入轻量级上采样算子后参数量提高了 0.08m，每秒浮点运算次数增加了 0.1G，总体来看没有发生较大改变；在用Swin替换原网络的主干网络并加入CBAM块之后，和原模型相比参数量提高了 75.3%，每秒浮点运算次数提高了 78.2%，mAP提高了 1.1%，可以看出计算速度变慢，参数量增加明显；当原模型的主干网络用Swin模块替换并加入CBAM和C2f模块后可以看出与原模型相比参数量增加了 75.5%，每秒浮点运算次数提高了 78.3%，mAP上升了 2.5%，当加入所有的模块并用Swin替代原网络的主干网络后，与原模型相比参数量增加了 75.5%，每秒浮点运算次数提高了 78.3%，mAP上升了 3.5%。

在番茄病害数据集上进行的消融实验表明，不同的改进方式组合不会使改进后的模型性能进行线性增加，在单独将 Yolov5 中的主干网络替换 Swin 网络后上涨了 2.5%，但是参数量会增加了 75.3%同时计算速度增加了 78.2%，在单独加入 CBAM 注意力机制或 C2f 模块或轻量级算子时对模型整体会有提高，由此可见 Swin 模块，C2f 模块，轻量级上采样算子和 CBAM 模块的加入有利于提高网络模型的目标检测能力，说明提出的方法是有效的。

### 4.3.2 模型对比

为了验证本文改进后模型的有效性，现将本文改进后的模型与主流模型的性能进行对比实验，主要对比模型为目标检测中的一阶段模型和二阶段模型，其中二阶段模型为主干网络分别以res50 残差网络Res101 残差网络为主干网络的Faster Rcn网络模型，主干网络分别以res50 残差网络Res101 残差网络为主干网络的Mask-Rcn网络模型，一阶段模型包括以主干网络为vgg16 的SSD网络模型，原Yolov5 网络模型，RetinaNet网络模型和MobileVit网络模型，各网络性能指标如表 4.2 所示

表 4.2 Swin-Yolov5 网络对比实验

Method	Params/M	FLOPs	mAP@.5/%	F1
Faster Rcn-Res50	41.14	123	82.3	81.70
Faster Rcn-Res101	60.14	154	83.1	82.90
Mask-Rcn-Res50	43.77	135	81.6	81.00
Mask-Rcn-Res101	62.76	158	81.2	81.00
SSD-vgg16	24.98	50.5	83.4	82.74
Yolov5	7.2	16.6	93.6	89.57
RetinaNet	36.9	75.8	88.4	88.00
MobileVit	55.18	128.5	83.9	84.25
轻量级Yolov5	2.6	4.7	94.9	94.60
Swin-Yolov5	29.45	76.4	96.8	95.85

由表 4.2 可知，Swin-Yolov5 网络模型的参数量比原模型提高了 75.3%，但是除了原模型和轻量级Yolov5 模型之外，与其他模型相比Swin-Yolov5 模型的参数量相对还是比较少的，FLOPs在对比的网络模型中也有优势，通过对比实验证明本文设计的Swin-Yolov5 网络的mAP值比原网络提高 3.5%，比轻量级Yolov5 网络提高 1.9%，说明设计的Swin-Yolov5 网络在叶片遮挡和叶片缺失以及环境影响的因素下也可以对番茄病害进行准确的识别，并且Swin-Yolov5 网络模型的F1 值也优于其他网络模型，说明该网络模型设计的较为合理。

由表 4.3 可知在Swin-Yolov5 网络训练过程中，所有番茄病害类召回率最高的为番茄黄化曲叶病、番茄花叶病和健康叶片，这几个类的正确预测样本数量占有所有正样本的比值为 97%，精确率较高的为番茄细菌性斑点病、番茄花叶病和健康叶片，预测正确的正样本数量占有所有预测结果为正样本数量的比值较高，从准确率上来看

番茄早疫病和番茄花叶病以及健康叶片的准确率达到 95%左右。

表 4.3 番茄病害各类指标

番茄病害种类	P/%	R/%	Acc/%
番茄细菌性斑点病	97.8	90	88.2
番茄早疫病	97.9	95	93.1
健康叶片	98.0	97	95.1
番茄花叶病	97.0	97	94.2
番茄鱼目斑病	94.2	98	92.5
番茄褐斑病	92.8	90	84.1
二斑叶螨	92.3	95	88.0
番茄黄化曲叶病	92.4	97	89.8

Swin-Yolov5 的混淆矩阵左侧代表了番茄病害预测的全部类别，矩阵中每一列的总数表示该类番茄病害的总数，每一行的值表示预测为该类病害的数量，由于数据量较大因此对结果进行归一化处理，使数值在 0 到 1 之间，Swin-Yolov5 目标检测模型的混淆矩阵如图 4.5 所示。

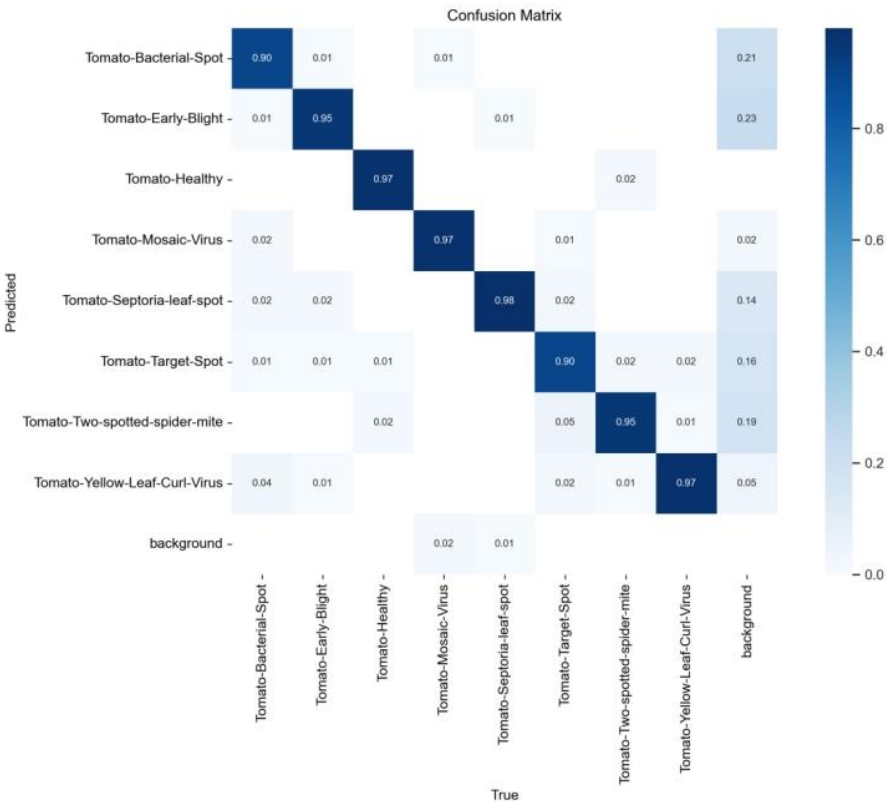


图 4.5 Swin-Yolov5 混淆矩阵

Swin-Yolov5 目标检测模型的P-R曲线如图 4.6 所示。

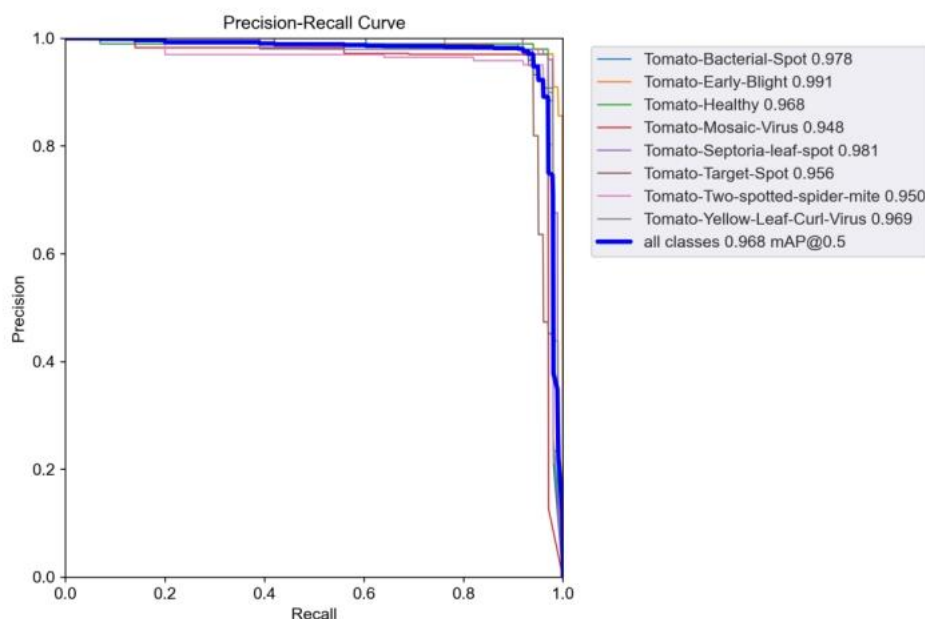


图 4.6 Swin-Yolov5 P-R 曲线

由图可知P-R曲线所围成的面积就是平均精确率（AP）的值，其中番茄细菌性斑点病的AP值为 97.8%，番茄早疫病病的AP值为 99.1%，番茄健康叶片的AP值为 96.8%，番茄花叶病毒病的AP值为 94.8%，番茄鱼目斑病的AP值为 98.1%，番茄褐斑病的AP值为 95.6%，二斑叶螨的AP值为 95.0%，番茄黄化曲叶病毒的AP值为 96.9%，经计算平均精度均值（mAP）为 96.8%比原模型的平均精度均值（mAP）提高了 3.5%，比本文提出的轻量级网络提高了 1.9%。

Swin-Yolov5 网络模型检测番茄叶部病害结果如图 4.7 所示。



图 4.7 Swin-Yolov5 检测结果

## 4.4 本章小结

本章首先阐述了 Swin-Transformer 模型的特点及用 Swin-Transformer 优化 Yolov5 的原理及设计过程，然后介绍了上采样算子 Carafe 的功能特点及设计方法，最后对模型性能进行分析评估。

针对自然光线或者叶片之间互相遮挡的问题，研究了基于 Swin-Yolov5 的番茄叶部病害检测模型。该模型第一个改进是用 Swin-Transformer 模型替换掉 Yolov5 中的主干网络，第二个改进是将原模型中的上采样替换为轻量级上采样算子 Carafe，获取了更大感受野，提取更多输入图像特征，并能根据输入特征进行优化重组。Carafe 算子的轻量级特点，确保以非常小的额外参数及计算量，带来网络整体性能的显著提升。研究结果表明：改良后的 Swin-Yolov5 网络模型在参数量上与计算复杂程度上表现较好，在 mAP 值上高于其他网络模型，比原模型提高了 3.5%，比轻量级 Yolov5 网络提高了 1.9%，说明 Swin-Yolov5 网络模型在叶片之间互相遮挡、叶片缺失或拍摄环境复杂的情况下，对番茄叶部病害的检测也有较好的检测准确率。





## 第五章 基于深度学习的番茄叶部病害检测系统

### 5.1 概述

本课题研究是基于学校物联网与大数据平台的应用需求，开发基于平台的番茄叶部病害检测的应用算法，最终将把算法作为智慧农业的一个具体应用嵌入平台，为用户提供番茄病害在线识别服务。因平台处于实际运行状态，担负其他实际业务，因此新的应用模块的加载和测试需要各项审批手续，且对技术成熟度的要求更高，还不具备在实际平台加载测试的条件，因此设计并构建了基于深度学习的番茄叶部病害检测系统，实现用户登录、图像上传、病害识别、结果可视化、病害防治建议等功能。

本章首先进行了系统的需求分析，然后设计了系统的整体方案，接着阐述了系统的实现过程，最后对系统进行了测试。

### 5.2 需求分析

用户在实际生产的过程中，发现番茄叶部出现异常，需要能尽快借助算法平台对病害进行检测，得到病害种类及治疗建议。用户首先登录检测系统，系统提供基于轻量级的 Yolov5 和 Swin-Yolov5 两种网络模型供用户选择，轻量级 Yolov5 检测速度更快，适合对检测速度要求高的应用；Swin-Yolov5 侧重检测质量，检测时间相对较长，但检测质量更高，用户可根据实际需求选择需要的检测模型。用户通过手机拍摄番茄叶部图像作为测试对象并上传病变检测系统，系统运行相应的检测模型后，可视化地展示检测出的病害种类，并给出该病害的治疗建议。

使用 python 的中较为方便的语言框架 PyQt5 和 OpenCV 进行集成<sup>[58]</sup>，通过搭建简洁的番茄病害检测系统，用户可以通过简单的操作对番茄病害进行检测。作为用户与算法系统交互的平台，该番茄病害检测系统设计了如下功能和性能需求：

(1) 番茄病害检测系统具有用户注册和登录模块。用户注册和登录功能是系统的基础的功能，当用户注册成功并登录系统后才能使用番茄病害检测服务。本文的番茄病害检测系统注册只需要输入用户名和密码就可以完成注册，当用户完成注册并登录后，可以使用该系统进行番茄病害类型的识别与检测。

(2) 番茄病害检测系统应具备更换模型的模块，番茄病害检测系统的模型可随着后续功能的需求进行更新换代，随着研究的深入训练出更好的模型，能够把训练好的新模型良好地移植到该系统上，因此该系统应具有较好的模型移植功能。

(3) 番茄病害检测系统应具备番茄病害检测与识别功能，用户在将待检测的番茄病害图像上传后，通过图像检测模块选择要检测的图像，番茄病害检测系统会调用相应模型对农户输入的危害图像进行检测，并在系统检测信息中将结果反馈给农户，同时给出专家建议，方便农户后续对番茄病害进行有效科学的治理。

(4) 番茄病害检测系统应具有准确的识别度，为了实现对番茄病害准确的识别与检测，该系统使用的病害检测模型对番茄病害的各类准确度均达到 90%以上，满足用户在日常使用中的基本所需，可以给用户反馈正确的番茄病害类型，能科学有效的指导用户对番茄病害的防治。

### 5.3 系统设计

目前设计的番茄病害检测系统为第一代版本，实现了农户使用的基础功能，一共有四个功能模块，系统整体功能结构如图 5.1 所示。

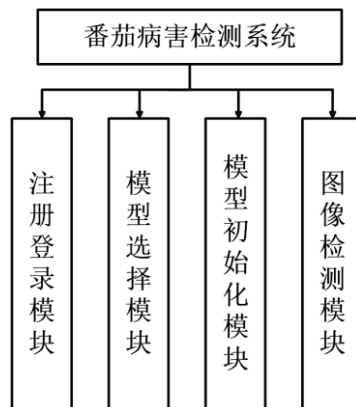


图 5.1 番茄病害检测系统功能结构图

第一个模块是注册登录模块，该模块是系统的基础模块，用户在经过简单的注册后就可以登录并使用该系统。第二个模块为模型选择模块，用户可以根据自己的应用需求选择相应的检测模型。本系统是版本 1，仅仅提供了文中提出的两种检测模型，模块预留了程序扩展接口，随着后续研究的深入，能够实现现有模型的更新及新模型的加载。第三个模块为模型初始化模块，这是系统运行必不可缺的步骤，主要是对用户选择的模型进行初始化操作，因为不同的模型具有各自独特的参数设置，当选择某个模型后，通过初始化加载该模型运行所需的初始参数，确保模型能够正确运行。第四个模块是图像检测模块，该模块是检测模型的运行模块，把用户上传

的图像作为检测对象，对病害类型进行检测，将检测结果反馈给用户，并提供专家指导建议，提供相应番茄病害治理的意见，给用户科学有效的解决办法。

番茄病害检测系统的流程如图 5.2 所示。

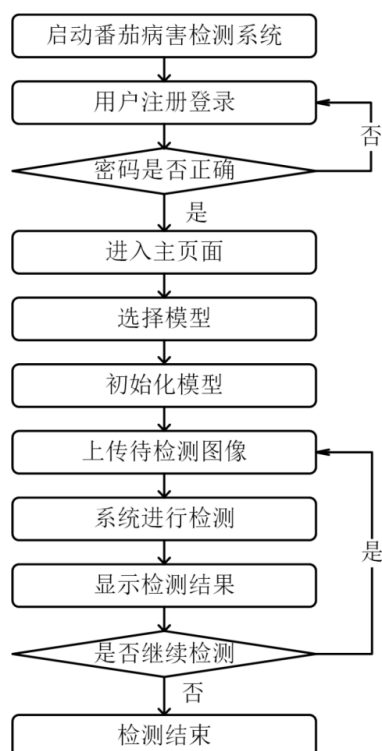


图 5.2 番茄病害检测系统流程图

首先启动番茄病害检测系统，接着用户在注册登录页面输入用户名和密码，系统会对输入的用户名和密码进行判断，假如不存在会提示密码或账号输入错误，正确输入账户密码后会进入主页面，然后农户选择相应的网络模型并对模型进行初始化操作，接着用户选择需要进行检测的病害图片传入到系统中，点击开始系统会对加载进模型的图像进行自动检测，并给出相应病害类别的识别结果，同时在检测结果中给出专家建议，用户可以根据专家意见对病害的植株进行科学的治疗，在进行番茄病害图像检测后，用户可以继续检测也可以选择结束检测并退出系统。

## 5.4 系统实现

在搭建系统时，首先将训练后的模型进行保存之后，通过 PyQt5 设计三个界面，分别为注册界面，登陆界面和番茄病害检测系统主页面，注册登录界面如图 5.3 所示，用户在填写用户名和密码之后系统会进行判断，数据库没有该账号则创建成功，若存在则提示账户已存在。



图 5.3 番茄病害检测系统注册界面

用户在输入账号密码之后系统会进行判断，若是账号密码不匹配则提示输入错误，若账号密码匹配则登陆成功，进入番茄病害检测系统主页面，登陆界面如图 5.4 所示。



图 5.4 番茄病害检测系统登陆界面

番茄病害检测系统主页面包含三个模块，分别为模型选择模块，模型初始化模块和图像检测模块。在进入番茄病害检测系统后首先点击选择模型，农户可以根据实际情况选择训练好的模型，若需要快速对植物病害进行检测时选择轻量级的 YOLOv5 模型；若上传的番茄病害叶片图像复杂时，可选择 Swin-YOLOv5 模型，选择模型如图 5.5 所示。由于该系统为初代版本，只提供了本文提出的两种网络模型，在后续研究中会根据用户的进一步需求增加其他模型供用户选择，系统设计考虑了后续其他检测模块的移植问题，预留了相应的接口以方便后续的开发与管理。考虑到用户的使用体验，系统的功能和操作尽量简洁明确，便于用户操作使用，年龄较大或没有太多计算机知识的用户也能方便使用。

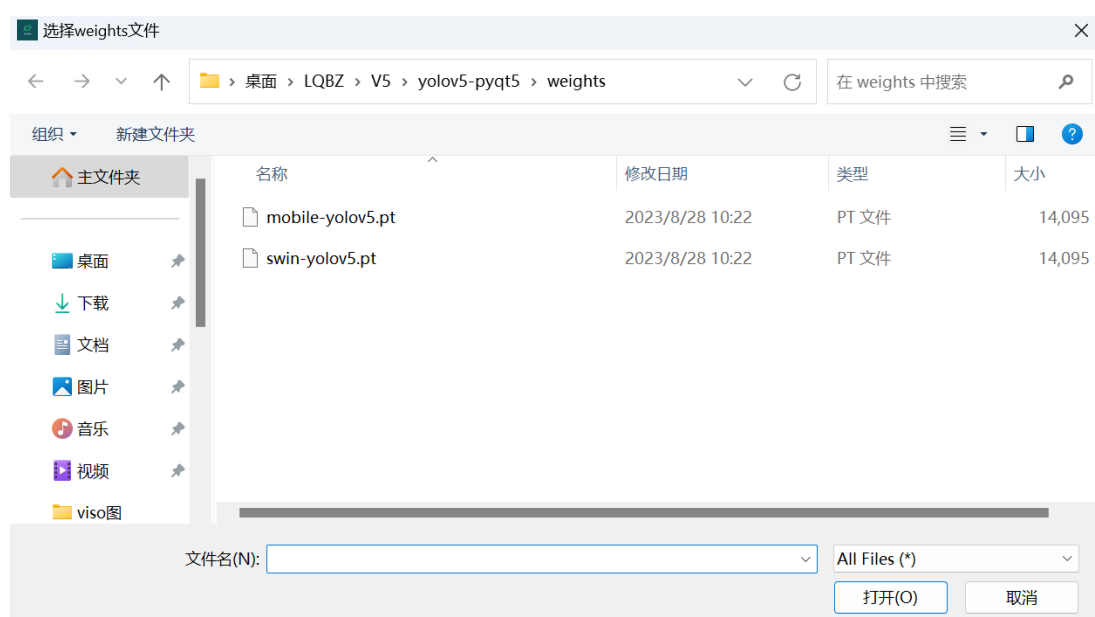


图 5.5 选择模型

图像检测模块为本项目的核心模块，首先农户在番茄病害检测系统主页面选择模型并对模型进行初始化，接着用户将需要检测的图像上传到系统，图像检测模块中包含图像上传功能，用户在该模块选择需要检测的图片并将图像上传到番茄病害检测系统中，系统会将上传的图像在主页面显示，上传检测图片如图 5.6 所示。

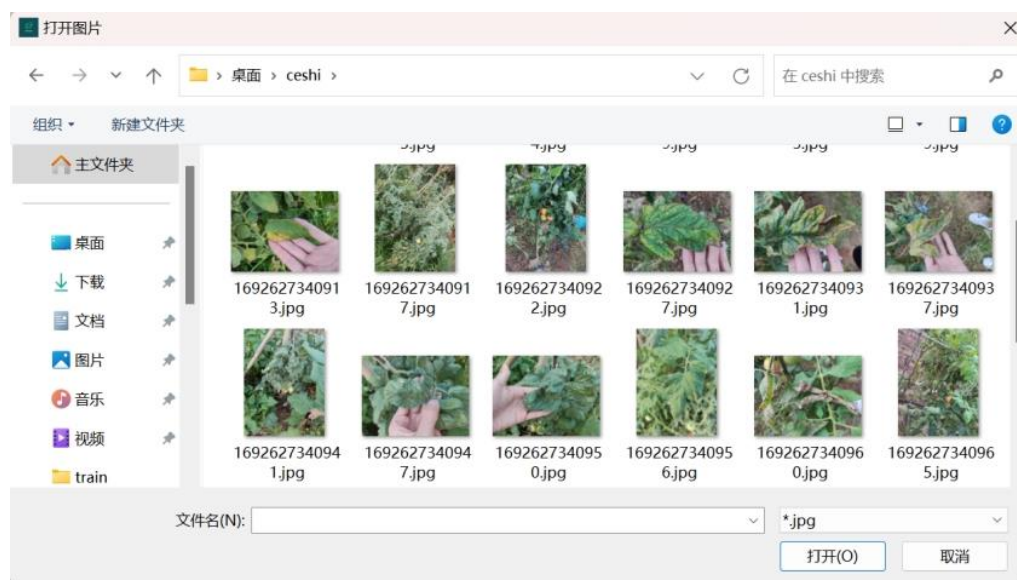


图 5.6 打开图片

用户点击开始按钮，系统会调用用户所选择的已经初始化好的网络模型对上传的图像进行识别，检测结果将在系统主页面进行展示，并在右侧检测信息中显示专家意见，农户根据专家给出的建议对病害植株进行治疗，番茄病害检测系统界面如图 5.7 所示。

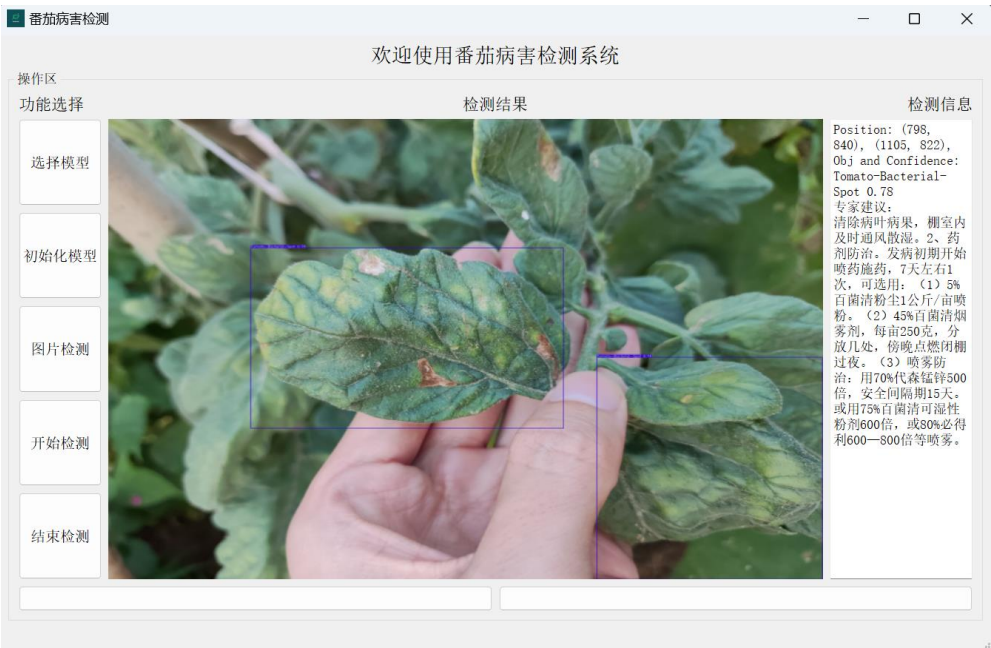


图 5.7 番茄病害检测系统结构图

5.5 系统测试

为了确保番茄病害检测系统的性能，采用了黑盒测试方法<sup>[59]</sup>，在测试过程中不仅输入正确的数据，同时也输入错误的数据，目的是在测试中尽可能地发现系统在使用时的错误和隐患，并依据发现的错误和隐患提出解决方案，使番茄病害检测系统功能完善，确保用户在使用番茄病害检测系统时可以准确识别各类番茄病害。

(1) 登录功能测试

分别以输入正确账号错误密码，输入错误账号错误密码，不输入账号密码，输入正确账号正确密码四种情况进行登录功能测试，登录功能测试如表 5.1 所示。

表 5.1 登录功能测试表

测试方法	预期	结果
输入正确账号错误密码	提示“用户名或密码错误”	与预期一致
输入错误账号错误密码	提示“用户名或密码错误”	与预期一致
不输入账号密码	提示“输入账号密码”	与预期一致
输入正确账号正确密码	登录系统	与预期一致

(2) 注册功能测试

分别以输入重复账号，不输入账号，输入账号不输入密码，输入账号与密码四种情况进行注册功能测试，注册功能测试如表 5.2 所示。

表 5.2 注册功能测试表

测试方法	预期	结果
输入重复账号	提示“用户名已存在”	与预期一致
不输入账号密码	提示“请输入账号”	与预期一致
输入账号不输入密码	提示“请输入密码”	与预期一致
输入正确账号正确密码	注册成功	与预期一致

### (3) 模型选择功能测试

分别以选择模型和不选择模型两种情况进行测试，选择模型的情况下有选择轻量级 YOLOv5 模型和 Swin-YOLOv5 模型两种情况，模型选择功能测试如表 5.3 所示。

表 5.3 模型选择测试表

测试方法	预期	结果
不选择模型	提示“请选择模型”	与预期一致
选择轻量级 YOLOv5 模型	模型选择成功	与预期一致
选择 Swin-YOLOv5 模型	模型选择成功	与预期一致

### (4) 模型初始化功能测试

分别以轻量级 YOLOv5 模型初始化和 Swin-YOLOv5 模型初始化两种情况进行测试，模型初始化测试如表 5.4 所示。

表 5.4 模型初始化测试表

测试方法	预期	结果
不模型初始化	提示“请模型初始化”	与预期一致
轻量级 YOLOv5 模型初始化	模型检测成功	与预期一致
Swin-YOLOv5 模型初始化	模型检测成功	与预期一致

### (5) 图像检测功能测试

分别以输入不带有番茄叶部病害的图片和带有番茄叶部病害的图片两种情况进行测试，图像检测功能测试如表 5.5 所示。

表 5.5 图片检测功能测试表

测试方法	预期	结果
轻量级 YOLOv5 上传不带番茄叶部病害图片	无检测结果	与预期一致
轻量级 YOLOv5 上传番茄叶部病害图片	输出检测的番茄病害种类	与预期一致
Swin-YOLOv5 上传不带番茄叶部病害图片	无检测结果	与预期一致
Swin-YOLOv5 上传番茄叶部病害图片	输出检测的番茄病害种类	与预期一致

经过上述测试本文设计的番茄病害检测系统在性能上没有缺陷，同时对番茄病害的检测也十分准确。

## 5.6 本章小结

本章设计了一个用于番茄叶部病害检测的测试系统，采用 PyQt5 设计并实现番茄病害检测系统界面，以该界面为用户交互平台，实现了用户管理、模型初始化、模型选择及病害图像检测等功能。在分析系统的应用需求的基础上，以简洁、便于使用、模型可扩展的要求设计了系统的整体方案，详细阐述了系统各功能模块的作用、使用方法及实现过程，并对照功能设计对系统进行了全面测试，测试结果表明该系统达到了预期设计目标，能够满足对番茄叶部病害的检测需求。



## 第六章 总结与展望

### 6.1 全文总结

深度学习技术的发展给农业病虫害检测提供了新的思路,运用深度学习技术检测作物病虫害在智慧农业领域应用越来越广泛。本课题基于学校物联网与大数据平台在智慧农业领域的应用需求,以基于深度学习的番茄叶部病害检测为切入点,开发基于平台的番茄叶部病害检测的应用算法,最终把算法作为智慧农业的一个具体应用嵌入平台,为用户提供番茄病害在线识别服务。由于番茄病害的种类繁多且病害与病害之间具有相似性,因此传统的人工检测番茄病害时,需要非常丰富的工作经验,这给用户使用带来很大困难。即使是经验丰富的用户在检测病害种类时,也会存在非常大的主观性,对同一种病害可能会出现不一致的判断。因此研究采用深度学习技术检测番茄病害具有重要的理论及实际价值。本文在分析国内外基于深度学习的目标检测、尤其是对农作物病害的检测研究现状的基础上,分别以提高检测实时性及检测质量为研究重点,构建了两种基于改进 Yolov5 模型的番茄叶部病变检测算法,设计并实现了用于用户操作的番茄病害检测系统。主要工作总结如下:

(1) 以提高检测实时性为重点,提出基于Yolov5 网络模型的轻量级网络模型,该模型通过将Mobile Net-v3 中的Bneck结构替换掉Yolov5 中的主干网络来降低模型的参数量,使模型加快计算速度,并加入CBAM注意力机制,该注意力机制可以将图像特征的空间信息与通道信息相联系,将原模型中的C3 模块改良为C2f模块,该模块采用堆叠思想,并参考残差网络重新设计网络结构,在一定程度上可以降低由网络深度带来的梯度消失问题,改良后的轻量级网络比原模型参数量下降 64%,每秒浮点运算次数下降 71%,mAP值上升了 1.6%。试验结果表明改良后的轻量级网络能在保证准确率的情况下提升模型计算速度。

(2) 以提高复杂识别条件下的检测质量为研究重点,提出了基于Swin Transformer的改进Yolov5 网络模型。该模型主要解决由于环境的复杂导致的采集番茄叶部病害图像过程中受光线和叶片遮挡的原因,致使对番茄叶部病害特征提取不明显,进而影响病害识别效果的问题,用Swin Transformer替换传统Yolov5 网络模型中的backbone部分,并加入轻量级上采样算子Carafe,在上采样时能获取更大感受野,提取更多输入图像特征,并能根据输入特征进行优化重组,Carafe算子的轻量级特点,确保以非常小的额外参数及计算量,带来网络整体性能的显著提升。仿真结果表明:

基于Swin-Yolov5的番茄叶部病害检测模型比传统Yolov5模型的mAP值提高3.5%，比轻量级网络提高1.9%，说明该模型在复杂环境中较好的表现。

(3) 设计了用于番茄叶部病害检测的测试系统。采用PyQt5设计并实现番茄病害检测系统界面，以该界面为用户交互平台，实现了用户管理、模型初始化、模型选择及病害图像检测等功能。在分析系统的应用需求的基础上，以简洁、便于使用、模型可扩展的要求设计了系统的整体方案，详细阐述了系统各功能模块的作用、使用方法及实现过程，并对照功能设计对系统进行了全面测试，测试结果表明该系统达到了预期设计目标，能够满足对番茄叶部病害的检测需求。

## 6.2 展望

本文对番茄叶部病害的研究是以Yolov5为原型提出了两种改进算法，在番茄叶部病害检测中取得了相对较好的检测性能，但是仍然存在一些问题需要进一步改进，改进方向如下：

(1) 本文研究的算法在番茄叶部病害上展现了较好的准确度，但是只涉及了番茄叶部病害的七个类别具有一定的局限性，因此在后续的研究中将扩充数据集，完善番茄病害的智能识别检测。

(2) 本文的轻量级网络在叶片遮挡下很难有较好的识别率，虽然Swin-Yolov5模型在该问题上有较好的表现，但是该模型参数量很大，计算十分耗时，因此在后续的研究中将综合复杂环境的检测精度和模型轻量化上对算法进行改进，以实现模型轻量化并能在复杂环境有较好识别率的模型。

(3) 目前番茄病害检测系统比较简单，界面还需要进一步完善优化，在后续的研究中还丰富系统功能，美化系统界面，提高技术成熟度，把该系统作为一个应用程序嵌入学校的物联网与大数据平台。

## 参考文献

- [1] Kaya Y, GÜrsoy E. A novel multi-head CNN design to identify plant diseases using the fusion of RGB images[J]. *Ecological Informatics*, 2023, 75: 101998.
- [2] 孙永珍,贺靖,魏芳等. “十三五”我国番茄产业发展及其国际竞争力评价[J]. *中国瓜菜*, 2023, 36(01): 112-116.
- [3] 赵佳悦. 基于卷积神经网络的番茄病害检测研究[D]. 山东师范大学, 2020.
- [4] 文枚,张连刚,陈天庆. 乡村旅游发展顶层设计: 政策演变与展望——基于 2004—2020 年“中央一号文件”的政策回顾[J]. *中南林业科技大学学报(社会科学版)*, 2021, 15(06): 101-107.
- [5] 韩二锋,王智海. 大数据在智慧农业发展中的应用探析[J]. *南方农机*, 2023, 54(15): 43-45.
- [6] 朱香玉. 乡村振兴视域下农业现代化现实困境和突破路径[J]. *辽宁农业职业技术学院学报*, 2023, 25(03): 13-17.
- [7] 商量. 智慧农业可视化监测系统应用研究[J]. *智慧农业导刊*, 2023, 3(14): 17-20.
- [8] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]//*Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. Ieee*, 2001, 1: I-I.
- [9] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//*2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). Ieee*, 2005, 1: 886-893.
- [10] Felzenszwalb P, McAllester D, Ramanan D. A discriminatively trained, multiscale, deformable part model[C]//*2008 IEEE conference on computer vision and pattern recognition. Ieee*, 2008: 1-8.
- [11] Papageorgiou C P, Oren M, Poggio T. A general framework for object detection[C]//*Sixth international conference on computer vision (IEEE Cat. No. 98CH36271). IEEE*, 1998: 555-562.
- [12] Deniz O, Bueno G, Salido J, et al. Face recognition using histograms of oriented gradients[J]. *Pattern recognition letters*, 2011, 32(12): 1598-1603.
- [13] Lowe D G. Object recognition from local scale-invariant features[C]//*Proceedings of the seventh IEEE international conference on computer vision. Ieee*, 1999, 2: 1150-1157.

- [14] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns[J]. IEEE Transactions on pattern analysis and machine intelligence, 2002, 24(7): 971-987.
- [15] Manjunath B S, Ma W Y. Texture features for browsing and retrieval of image data[J]. IEEE Transactions on pattern analysis and machine intelligence, 1996, 18(8): 837-842.
- [16] Noble W S. What is a support vector machine?[J]. Nature biotechnology, 2006, 24(12): 1565-1567.
- [17] Freund Y, Schapire R E. Experiments with a new boosting algorithm[C]//icml. 1996, 96: 148-156.
- [18] Neubeck A, Van Gool L. Efficient non-maximum suppression[C]//18th international conference on pattern recognition (ICPR'06). IEEE, 2006, 3: 850-855.
- [19] Viola P, Jones M J. Robust real-time face detection[J]. International journal of computer vision, 2004, 57: 137-154.
- [20] Moore G E. Cramming more components onto integrated circuits[J]. Proceedings of the IEEE, 1998, 86(1): 82-85.
- [21] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.
- [22] 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.
- [23] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [24] 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.
- [25] Zhu X, Lyu S, Wang X, et al. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 2778-2788.
- [26] Qi J, Liu X, Liu K, et al. An improved Yolov5 model based on visual attention mechanism: Application to recognition of tomato virus disease[J]. Computers and electronics in agriculture, 2022, 194: 106780.
- [27] Wu J, Wen C, Chen H, et al. DS-DETR: A Model for Tomato Leaf Disease segmentation and Damage Evaluation[J]. Agronomy, 2022, 12(9): 2023.

- [28]Gong H, Mu T, Li Q, et al. Swin-transformer-enabled YOLOv5 with attention mechanism for small object detection on satellite images[J]. Remote Sensing, 2022, 14(12): 2861.
- [29]Zheng H, Wang G, Li X. YOLOX-Dense-CT: a detection algorithm for cherry tomatoes based on YOLOX and DenseNet[J]. Journal of Food Measurement and Characterization, 2022, 16(6): 4788-4799.
- [30]Zeng T, Li S, Song Q, et al. Lightweight tomato real-time detection method based on improved YOLO and mobile deployment[J]. Computers and Electronics in Agriculture, 2023, 205: 107625.
- [31]Zhang Y, Song C, Zhang D. Deep learning-based object detection improvement for tomato disease[J]. IEEE access, 2020, 8: 56607-56614.
- [32]Roy A M, Bhaduri J. A deep learning enabled multi-class plant disease detection model based on computer vision[J]. Ai, 2021, 2(3): 413-428.
- [33]Baidya R, Jeong H. Yolov5 with convMixer prediction heads for precise object detection in drone imagery[J]. sensors, 2022, 22(21): 8424.
- [34]Jeong S, Jeong S, Bong J. Detection of tomato leaf miner using deep neural network[J]. Sensors, 2022, 22(24): 9959.
- [35]Bhujel A, Kim N E, Arulmozhi E, et al. A lightweight Attention-based convolutional neural networks for tomato leaf disease classification[J]. Agriculture, 2022, 12(2): 228.
- [36]Storey G, Meng Q, Li B. Leaf disease segmentation and detection in apple orchards for precise smart spraying in sustainable agriculture[J]. Sustainability, 2022, 14(3): 1458.
- [37]Arifando R, Eto S, Wada C. Improved Yolov5-Based Lightweight Object Detection Algorithm for People with Visual Impairment to Detect Buses[J]. Applied Sciences, 2023, 13(9): 5802.
- [38]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.
- [39]He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
- [40]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.

- [41] Lin M, Chen Q, Yan S. Network in network[J]. arXiv preprint arXiv:1312.4400, 2013.
- [42] Maas A L, Hannun A Y, Ng A Y. Rectifier nonlinearities improve neural network acoustic models[C]//Proc. icml. 2013, 30(1): 3.
- [43] Hendrycks D, Gimpel K. Gaussian error linear units (gelus)[J]. arXiv preprint arXiv:1606.08415, 2016.
- [44] Mish M D. A self regularized non-monotonic activation function 2019[J]. arXiv preprint arXiv:1908.08681, 1908.
- [45] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [46] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.
- [47] Liu Z, Lin Y, Cao Y, et al. Swin Transformer: Hierarchical vision Transformer using shifted windows[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022.
- [48] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.
- [49] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.
- [50] 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.
- [51] Howard A, Sandler M, Chu G, et al. searching for mobilenetv3[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1314-1324.
- [52] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.
- [53] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.
- [54] 杨琼. 基于局部特征的图像分类方法研究与应用[D].西安工业大学,2023.
- [55] 梁小玉. 深度神经网络的鲁棒性与准确性之间的权衡研究[D].浙江科技学院,2023.
- [56] 任义丽,罗路.卷积神经网络过拟合问题研究[J].信息系统工程,2019(05):140-142.
- [57] Huang W, Zhao X, Huang W, et al. A training strategy to improve the generalization capability of deep learning-based significant wave height prediction models in offshore

- China[J]. Ocean Engineering, 2023, 283: 114938.
- [58] 桑晓丹,郭锐.基于 PyQt5 的数字图像处理实验平台设计[J].电子技术与软件工程,2021(18):129-130.
- [59] 陶慧杰. 面向 Web 应用的智能化测试平台的设计与实现[D].北京交通大学,2022.