全面乡村振兴背景下基于 ResNeXt–CBAM 神经网络模型的番茄叶片病害识别研究

摘 要

乡村振兴战略是一项管全面、管长远的大战略，是新时代“三农”工作的总抓手，然而时至今日，乡村振兴之路仍有许多困难亟待解决。这其中，农业病虫害问题是制约乡村发展的一大桎梏。为此，本研究聚焦番茄病害智能识别难题，提出了一种基于 ResNeXt-CBAM 神经网络模型的创新解决方案。针对传统方法在复杂田间场景中存在的鲁棒性不足、模型可解释性低及边缘设备部署困难等问题，本研究通过多基数分组残差网络（ResNeXt）与双路径注意力机制（CBAM）的深度融合，构建轻量化高精度分类模型，并结合 YOLOv8-n 目标检测框架实现双阶段协同诊断。实验表明，模型在 11 类番茄病害数据集上平均识别准确率达 91.98%，Kappa 系数 0.911，较现有农业模型（如 MobileNetV3）提升 0.98%；研究通过算法创新与工程化落地，为破解“智慧农业最后一公里”难题提供了可复制的技术范式。

关键词：番茄病害识别；深度学习；卷积神经网络；乡村振兴；智慧农业

目 录

表格与插图清单

全面乡村振兴背景下基于 ResNeXt–CBAM 神经网络模型的番茄叶片病害识别研究

一、引言

（一）研究背景

在全球粮食安全格局深刻变革与农业数字化转型双重驱动之下，人工智能赋能的精准农业已成为重塑全球农业生产体系的关键所在。联合国粮农组织《2023 年粮食及农业状况》报告指出，气候变化引发的极端天气已导致全球农作物病害发生率年均增长 22%，构建智能病害监测体系被列为保障粮食安全的优先行动领域。在此背景下，我国《“十四五”数字经济发展规划》明确提出要“深化人工智能技术在农业生产经营全链条的融合应用”，2023 年中央一号文件更将“智慧农业”列为乡村振兴战略的五大核心工程之一。

番茄富含丰富的胡萝卜素和维生素 C，是全球第四大蔬菜作物，在我国的年产量现已突破 1.8 亿吨，占设施农业之比高达 32%，对于我国的经济发展具有重大的战略意义。然而在番茄生产过程中，病虫害影响对番茄的产量和质量，在损伤国家农业经济的同时也会给农民带来巨大的经济损失。而这其中，对番茄生产危害最大的疾病包括：细菌性斑点病（Bacterial spot）、早疫病（Early blight）、晚疫病（Late blight）、叶霉病（Leaf Mold）、白斑病（Septoria leaf spot）、蜘蛛螨（Spider mites & Two-spotted_ spider mite）、靶斑病（Target_spot）、番茄花叶病毒（Tomato mosaic virus）、番茄黄化曲叶病毒（Tomato yellow leaf curl virus）和白粉病（Powdery mildew）等。这些病害不仅严重影响番茄的产量和品质，还会增加农业生产的成本。传统的病害诊断方法依赖人工观察，既耗时又容易受到人为因素干扰，而单纯依赖化学防治又会影响果实质量，因此研究新型方法用于番茄疾病的诊断意义重大。

本研究立足国家粮食安全战略需求，聚焦农业产业链薄弱环节，通过构建基于卷积神经网络（CNN）的番茄病害智能诊断模型，探索人工智能技术在我国设施农业场景的落地路径。研究立足于轻量化神经网络模型设计，着力破解

农业场景中光照多变、样本稀缺等现实约束，为实现《数字乡村发展行动计划（2022-2025 年）》提出的“农业数字经济年增长率超过 8%”目标提供技术支撑。

（二）研究意义

本文研究可能的贡献包括：

1.推动智慧农业发展，提高番茄病害诊断的智能化水平，减少人工检测成本，提高农业生产效率。

2.筑牢粮食安全数字屏障，为《国家粮食安全中长期规划纲要》提出的"单位面积产能提升 10%"目标提供关键技术支撑。在耕地资源刚性约束下开辟"数字赋能、科技增粮"的新路径。

3.为后续农业病害检测系统的开发提供理论和实践依据，促进农业数字化转型。

（三）研究目标

1.构建高精度番茄病害分类模型。

2.构建可扩展的病害识别技术体系。

3.解决小样本数据下的过拟合问题。

二、文献综述

（一）相关研究

近年来，深度学习在农业病害识别领域取得了显著的进展。卷积神经网络（CNN）因其强大的特征提取能力，成为农业病害识别的核心技术。Mohanty 等（2016）在开创性研究中首次将 AlexNet 应用于植物病害分类，并在 PlantVillage 数据集上实现了“超过 90%的分类准确率”，证明了深度学习在农业领域的潜力。

农业病害图像数据稀缺则是农业病害识别领域的普遍问题。如何获得切实模拟真实田间环境的数据集是研究农业病虫害的重点。而 Shorten 和 Khoshgoftaar（2019）在综述中强调，“传统数据增强（如旋转、裁剪）通过简单的几何变换扩展数据集，但难以模拟真实田间环境的复杂性”。后续研究为突破此限

制做了很多的努力：Zhang 等（2021）提出基于 CycleGAN 的数据增强方法，生成不同光照条件下的病害样本，实验表明“CycleGAN 生成的数据使小样本场景下的模型准确率提升 12%”。

此外，超参数优化对模型性能至关重要。传统网格搜索效率低下，Bergstra 等（2011）提出的 TPE 算法通过贝叶斯优化框架，“在超参数搜索任务中，仅需传统方法 1/3 的迭代次数即可找到最优解”。近年来，元启发式算法（如麻雀搜索算法 SSA）因其全局搜索能力受到关注，Xue 和 Shen（2020）在 SSA 的原始论文中指出，“麻雀搜索算法在复杂优化问题中的收敛速度较粒子群算法（PSO）提升 35%”。

（二）研究不足

1.数据局限性

现有的公开数据集的样本多采集于实验室环境，缺乏真实田间场景的多样性，导致模型实际部署受限（Hughes & Salathé, 2015）。

2.模型泛化能力不足

Picon 等（2019）在跨温室测试中发现，“当测试环境与训练数据分布差异较大时，模型准确率下降超过 20%，尤其对小样本病害类别（如白粉病）的识别性能急剧恶化”。

3.计算资源依赖性强

Amara 等（2017）强调，“高精度模型（如 ResNet50）的参数量高达 23.5 M，难以满足边缘设备的实时性需求，导致田间实时诊断难以落地”。

（三）研究创新点

1.引入改进型 ResNeXt 架构，增强特征表达能力

在传统 ResNet 基础上，采用 ResNeXt 的多基数（Cardinality）分组卷积设计，通过增加网络宽度而非深度，提升特征多样性。

2.融合自注意力机制与 CBAM 模块，优化特征聚焦能力

通过全局建模，捕捉图像位置间依赖关系，解决复杂图片背景导致的识别干扰问题。融合自适应注意力机制，使用 CBAM 模块使模型聚焦于关键区域。

3.集成 YOLOv8 检测框架，构建两阶段协同诊断系统

使用 ResNeXt-YOLOv8 双层模型结构，实现目标检测、图像识别功能解耦，使得模型能够适应复杂背景、噪声下的识别环境，实现双阶段协同检测。

三、研究框架

本研究的框架如图所示：

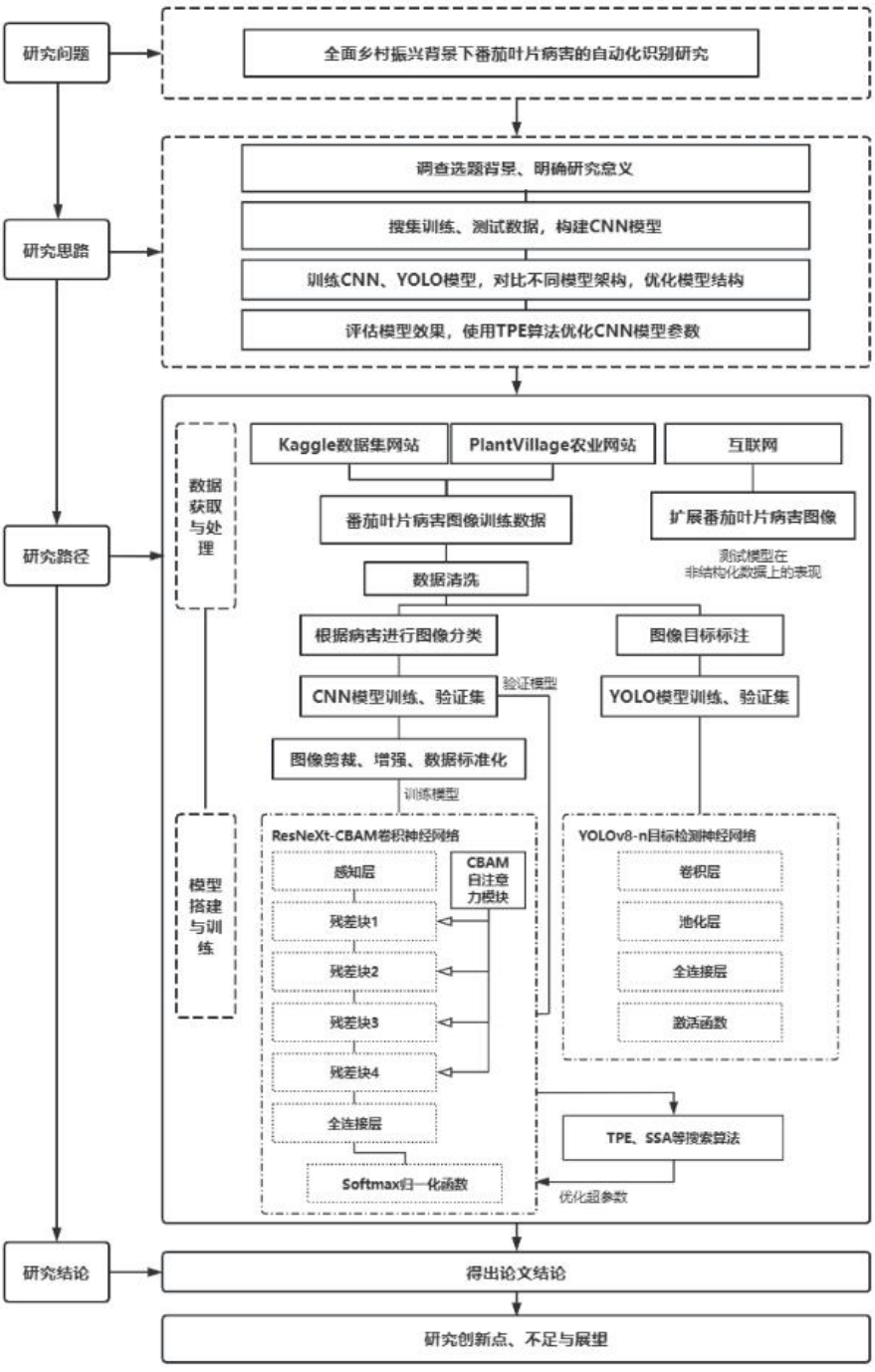

图 1 研究框架流程图

四、数据介绍

（一）数据来源

1.超参数搜索数据集

超参数搜索数据集来源于 Kaggle 网站的“Dataset of Tomato Leaves”数据集，地址为 https://www.kaggle.com/datasets/sunilgautam/dataset-of-tomato-leaves。该数据集包括 9 种染病番茄叶片图像和 1 组健康番茄叶片图像，共 11617 张图片，按 3:1 的比例划分为训练集和验证集。该数据集主要用于超参数搜索过程，以便于寻找到模型训练的最佳超参数，以提高后续训练的效果和效率。

2.番茄叶片数据集

为分析诊断番茄叶片相关疾病并建立相应模型，本文以 Kaggle 网站上的“Tomato Leaves Dataset”数据集为研究对象，地址为 https://www.kaggle.com/datasets/ashishmotwani/tomato/data?select=valid。研究将图像分类了十种番茄叶片疾病与健康，共 11 种类别。该数据集图片从实验室与野外环境中收集，共收录了 32534 张图片数据，每组样本容量均在 1000 张以上。

表 1 “Tomato Leaves Dataset”数据集各标签数据数量一览表

图像数据标签	训练集数据数量	验证集数据数量	总计
Bacterial_spot	2826	732	3558
Early_blight	2455	643	3098
Healthy	3113	792	3905
Late_blight	2754	739	3493
Leaf_Mold	2882	746	3628
Septoria_leaf_spot	1747	435	2182
Spider_mites Two-spotted_spider_mite	1827	457	2284
Target_spot	2039	498	2537
Tomato_mosaic_virus	2153	584	2737
Tomato_yellow_leaf_curl_virus	3051	805	3856
powdery_mildew	1004	252	1256

模型测试集由“Tomato Leaves Dataset"验证集中随机选取的 2200 张图片组成。

3.目标检测训练数据集

我们在互联网上通过爬虫搜集了 1136 张形态各异的番茄叶片图片，并使用 LabelImg 工具对图片进行人工目标标注，用于 YOLOv8-n 模型的训练。

（二）数据特征与划分

以下为超参数搜索时和模型训练时数据集的划分方式：

表 2 番茄叶片数据集划分

数据集类型	图像数量	用途
超参数搜索训练集	8715	寻找模型训练的最佳超参数
超参数搜索验证集	2902	寻找模型训练的最佳超参数
训练集	32025	模型训练
验证集	6683	评估模型性能

五、研究方法

（一）卷积神经网络（Convolution Neural Network）

在过去的几年内，深度学习逐渐成为机器学习领域中应用最广泛的计算方法。深度学习模型在众多复杂的认知任务中取得了很好的效果，在很多时候其表现甚至超越人类。卷积神经网络是计算机视觉领域最常用的深度学习模型，其在包括图像分类和分割、目标检测、语音识别等领域应用广泛[9]。本次研究主要使用用于图像分类的卷积神经网络，这种神经网络的基本结构[10]如图所示：

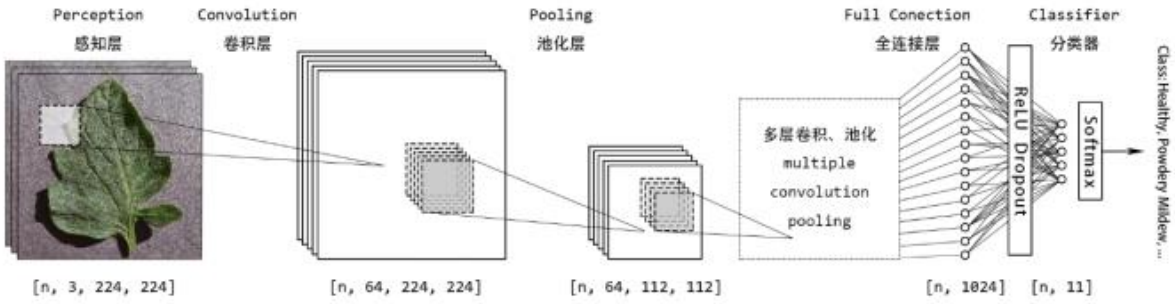

图 2 卷积神经网络基本结构图

1.感知层（Perception Layer）

在本模型中，感知层输入图像，使之可以用于模型训练的结构。这个结构通常会对图像进行随机旋转、平移、伸缩、翻转、对比度调整，并将像素数据按通道归一化，调整图像大小至目标尺寸，最终转化为张量形式。本模型图像输入尺寸是 224×224 像素，具有 RGB 三个通道。

2.卷积层（Convolution Layer）

卷积层是卷积神经网络的核心，也是整个网络学习图像形状、纹理的基础。卷积层的工作原理是二维数据的卷积操作，其过程可以用以下公式描述：

$$(I * K)_{c'}(x,y) = \sum_{c} \sum_{(i,j)=(1,1)}^{(k,k)} I_c\left(x - \left\lfloor\frac{k}{2}\right\rfloor + i, y - \left\lfloor\frac{k}{2}\right\rfloor + j\right) \cdot K_{c'}(i,j) \qquad \text{公式(1)}$$

上述公式中，I_c代表当前被卷积图像第c个通道的张量，$K_{c'}$是第c'个边长为k的卷积核，$(I * K)_{c'}$就是卷积所得图像第c'个通道的张量。对于生成的每个像素数据，其值相当于以其为中心的$k \times k$范围内原图像数据与卷积核数据的元素积之和。随着卷积核在被卷积图像上滑动，原图像的局部特征便会以卷积形式映射到新产生的图像上。

通常，卷积操作会扩展张量的通道数，使得模型能够学习到更多抽象的特征。对于正方形的输入图像，其空间尺寸满足以下公式：

$$O = \frac{W - k + 2P}{S} + 1 \qquad \text{公式(2)}$$

其中，O是输出图像的边长，W是输入边长，P是边缘填充大小，S是卷积核移动步长。下图展示了一个对通道数为 1 的图像做空间尺寸不变卷积的例子：

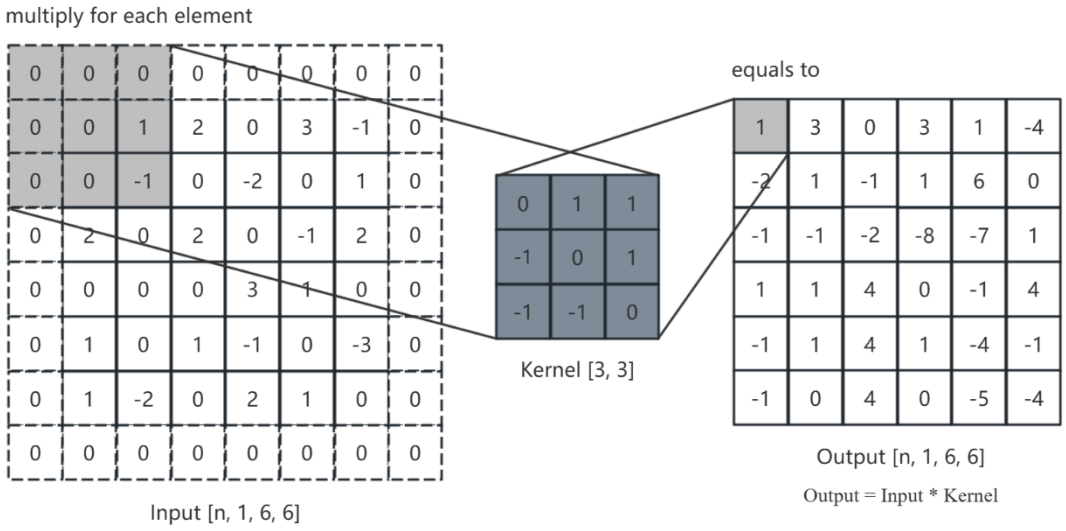

图 3 空间尺寸不变卷积示例图

其中卷积核边长为 3，移动步长为 1，边缘填充大小为 1（见 Input 四周虚线区域），填充数据为 0；图中 Output 中的灰色像素对应着 Input 中3×3大小

的灰色区域，该区域称为这个像素的感受野。感受野的大小与卷积核空间尺寸匹配，决定了当前卷积层提取特征的局部性程度。

在卷积神经网络中，每一个卷积层输入张量的形状是$[N,C,H,W]$，分别表示该批次样本个数、单个样本通道数、高度、宽度。其中N值也被称为 batch size，是模型的重要超参数。在模型训练过程中，经常发生模型过于复杂导致的过拟合现象，以及模型深度过大导致的梯度消失现象，这些现象往往会使得模型在陌生数据上表现不佳，或在训练时阻碍模型参数的进一步优化[15]。为防止以上现象发生，我们在每一个卷积层后加入了批归一化（batch normalization）的正则化过程。

在一个训练批次中，对形状为$[n,1,h,w]$的张量$\boldsymbol{X}\in\mathbb{R}^{N\times C\times H\times W}$。批归一化的计算过程如下，其中：

$$\vec{\mu}_c=\frac{1}{N\cdot H\cdot W}\sum_{n=1}^{N}\sum_{h=1}^{H}\sum_{w=1}^{W}\boldsymbol{X}_{n,c,h,w} \qquad \text{公式(3)}$$

$\vec{\mu}$为单个通道内，所有样本的所有像素数据的均值，形状为$[1,C]$；

$$\overrightarrow{\sigma^2}_c=\frac{1}{N\cdot H\cdot W}\sum_{n=1}^{N}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(\boldsymbol{X}_{n,c,h,w}-\vec{\mu}_c\right)^2 \qquad \text{公式(4)}$$

$\overrightarrow{\sigma^2}$为单个通道内，所有样本的所有像素数据的方差，形状同样为$[1,C]$。

对于单个通道内的每一个像素数据，做以下线性变换：

$$\widehat{\boldsymbol{X}}_{n,c,h,w}=\frac{\boldsymbol{X}_{n,c,h,w}-\vec{\mu}_c}{\sqrt{\overrightarrow{\sigma^2}_c+\varepsilon}} \qquad \text{公式(5)}$$

$$\widehat{\boldsymbol{Y}}_{n,c,h,w}=\vec{\gamma}_c\cdot\widehat{\boldsymbol{X}}_{n,c,h,w}+\vec{\beta}_c \qquad \text{公式(6)}$$

其中$\vec{\gamma}_c$与$\vec{\beta}_c$为待学习参数，ε是一个小正数，防止分母为 0。归一化结果$\widehat{\boldsymbol{Y}}\in\mathbb{R}^{N\times C\times H\times W}$。

批归一化在通道层面使数据的分布接近标准正态分布，减少了样本内部的协变量偏移，使得卷积之后的模型更加稳定[15]。

3.池化层（Pooling Layer）

池化层又称汇聚层、下采样层，用于在卷积层之后对图像的每个通道提取主要特征的同时，减小图像空间尺寸。池化层的存在减少了整个模型的参数量，简化模型结构，在加速模型训练的同时减小了过拟合风险[10]。

本模型主要采用最大池化法，其原理如下：

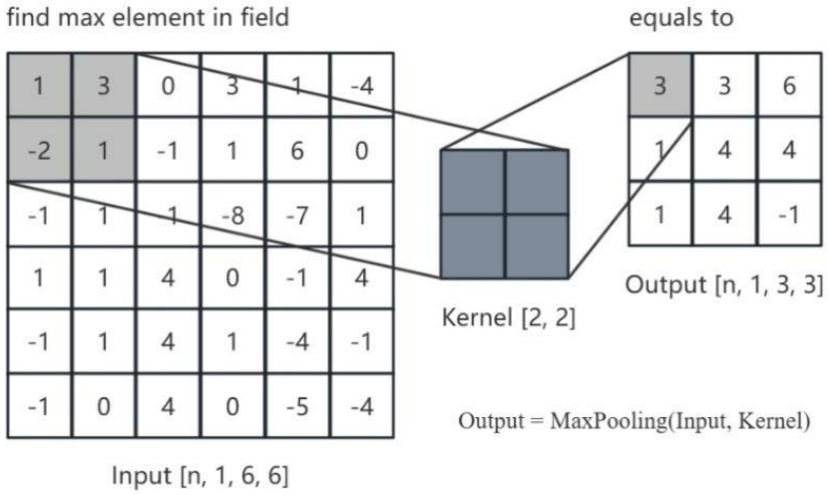

图 4 最大池化法示意图

其中下采样核边长为 2，移动步长为 2，无边缘填充。Output 中每一个像素数据，都等于其 Input 感受野中所有元素的最大值。由于下采样核的移动步长为 2，故图像线度会缩小为原来的一半。

现代卷积神经网络往往在池化层之后引入非线性的激活函数，以更好地拟合非线性数据、增强模型的特征提取能力[10]。本模型使用的激活函数为 ReLU 函数，其定义为：

$$ReLU(x) = max(x, 0) \qquad \text{公式(7)}$$

4.全连接层（Full Connection Layer）

全连接层用于收集前置卷积层、池化层传递过来的特征，并将这些数据进行处理、汇总、最终传递给分类器进行分类。在全连接层，我们需要将输入的张量展平为一个一维的向量并输入一个线性的神经元层。

需要指出的是，全连接层中，相邻线性神经元层的任意两个神经元之间都有权重连接，这无疑会导致模型复杂化，从而增加模型过拟合风险。因此，我们在全连接层引入了 Dropout 正则化机制。Dropout 机制可以将神经元按指定的几率关闭，从而减少了优化参数的数量，从而简化模型结构[16]。其效果示意见下图：

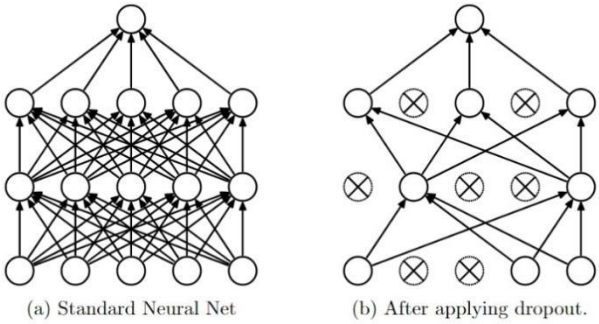

图 5 Dropout 正则化机制效果示意图

在引入了关闭概率为p的 Dropout 机制后，每一个全连接层的计算机制可表示为：

$$\vec{\boldsymbol{r}} \sim Bernoulli(p) \quad \text{公式(8)}$$

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{X_i} \times \vec{\boldsymbol{r}}_i \quad \text{公式(9)}$$

$$\boldsymbol{Y} = ReLU(\boldsymbol{W} \cdot \tilde{\boldsymbol{x}} + b) \quad \text{公式(10)}$$

其中$Bernoulli(p)$是一个元素为 0 或 1 的向量，对每一个元素而言，其为 0 的概率为p。$\boldsymbol{X}$为输入全连接层的特征向量，$\boldsymbol{Y}$则是输出向量。

5.分类器（Classifier）

分类器用于接受前置所有神经元的处理结果，并输出每个样本对应类别的概率。在本研究中，模型需要分辨源自 11 种不同类别的图片，因此分类器的输出形状为$[N, 11]$。为了满足概率归一原则，我们需要使用 Softmax 函数处理输出数据：

$$Softmax(X)_i = \frac{e^{X_i}}{\sum_{j=1}^{class} e^{X_j}} \quad \text{公式(11)}$$

其中$class$值是类别数。在经过 Softmax 函数归一化之后，输出的数据就是当前样本对应每个类别的概率。

（二）模型评价方法

1.交叉熵损失函数（Binary Cross Entropy Loss）

在神经网络中，损失函数用于量化表示当前模型对样本的预测与样本真实标签之间的差别大小。在多分类问题中，我们使用交叉熵损失函数以评估模型效果。其定义如下：

$$L(\boldsymbol{Y},\widehat{\boldsymbol{Y}})=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{Y}\,log_2(\widehat{\boldsymbol{Y}}) \qquad \text{公式(12)}$$

其中$\boldsymbol{Y}$是当前批中样本真实标签的独热编码向量，$\widehat{\boldsymbol{Y}}$是模型预测值向量，上述公式中对向量的运算都是按元素运算，而不是张量运算。对模型进行优化的过程，就是调整模型参数，使得模型损失函数值越来越小的迭代过程。

2.准确率（Accuracy）

对一批已知标签的数据，通过当前模型进行预测，其中预测类别正确的占比即模型在当前数据上的准确率。准确率是评估模型实际表现的最直观、最常用的数据。

3.混淆矩阵（Confusion Matrix）

在对一批数据进行预测时，混淆矩阵便可以很直观地反映各样本被模型预测为了什么类型这一信息。

对于总类别数为c的模型，其混淆矩阵的形状为$c\times c$，第i行第j列的数据表示实际标签为第i类、模型预测为第j类的样本数。当一个模型的表现越出色，其混淆矩阵的对角线上元素和应该越接近预测样本总数。通过分析混淆矩阵，我们可以很清晰地得知当前模型容易混淆哪些类型的数据，从而为在数据层面进一步优化模型提供思路。

4.Kappa 系数（Cohen's Kappa）

在混淆矩阵的基础上，我们引入了 Kappa 系数来进一步评估模型表现。Kappa 系数的计算基于混淆矩阵，并修正了随机猜测对准确率的影响，在多分类模型的评估上应用广泛。其计算方法由以下公式给出：

$$\kappa=\frac{p_o-p_e}{1-p_o} \qquad \text{公式(13)}$$

其中p_o被称为观测一致性，即分类器的总体准确率；p_e为期望一致性，即随机猜测时的一致性概率。这两个一致性系数的计算依靠混淆矩阵M：

$$p_o=\frac{\sum_{i=1}^{c}M_{ii}}{N} \qquad \text{公式(14)}$$

$$p_e=\sum_{i=1}^{c}\left(\frac{\sum_{j=1}^{c}M_{ij}}{N}\cdot\frac{\sum_{j=1}^{c}M_{ji}}{N}\right) \qquad \text{公式(15)}$$

Kappa 系数越接近 1，表示模型预测效果越好。

（三）模型训练方法

本模型使用 Adam 梯度下降优化器，对模型参数进行优化。Adam 算法源于随机梯度下降算法（Stochastic Gradient Descent），同时融合了 Momentum、RMS Prop 等改进算法[18]，在深度学习神经网络中应用广泛。其基本迭代方法如下：

$$\begin{cases} v_t = \beta_1 v_{t-1} + (1-\beta_1)\dfrac{\partial L}{\partial w} \\ s_t = \beta_2 s_{t-1} + (1-\beta_2)\left(\dfrac{\partial L}{\partial w}\right)^2 \\ w_t = w_{t-1} - \dfrac{\alpha v_t}{\sqrt{s_t + \varepsilon}} \end{cases} \quad \text{公式(16)}$$

其中α、β_1、β_2分别为学习率、动量衰减系数、二阶矩衰减系数，是整个模型训练过程中的重要超参数，其取值直接影响到最终模型效果。

Adam 梯度下降优化器可以根据梯度的变化自适应地调节学习率，以避免模型训练过程中的震荡与不稳定，显著减少了模型的训练数据。

（四）模型超参数搜索方法

本研究使用了麻雀搜索算法（Sparrow Search Algorithm）、TPE 搜索算法（Tree-structured Parzen Estimator）两种搜索算法进行对比实验论证，并优化模型超参数，使得模型能在训练过程中更快收敛，并取得更高的准确率。

1.麻雀搜索算法

作为一种新兴的元启发搜索算法，麻雀搜索算法于 2020 年被提出，其核心思想是模拟麻雀在觅食过程中的分工行为，通过动态调整发现者、跟随者、警戒者三种角色的位置，有效平衡了全局搜索与局部优化的性能。

该算法将一个麻雀种群分为三种角色：发现者、跟随者、警戒者。算法开始时，种群中的个体被随即放置在搜索空间中。在每一次迭代时，随机生成预警值$R_2 \in [0,1]$，该预警值将决定发现者的行为；种群中位置最优的前 20%的个体为发现者，当预警值小于安全阈值SE时，发现者扩大搜索范围，即趋向全局搜索，否则向附近安全区域移动，即趋向局部优化；剩余 80%为追随者，对于每一个追随者个体，若其当前位置较优，则会向全局最优个体的方向靠近，否则会移动至更远的区域搜索；警戒者是种群中随机挑选的 20%个体，当警戒者

距离当前全局最优位置过远时，表示其被天敌捕食的概率很大，故其向最优位置靠近。重复上述迭代过程，直到达到最大迭代数，或当前已满足收敛条件[7]。

麻雀搜索算法很好地平衡了全局搜索和局部优化，并创新性地引入了随机扰动和警戒者的身份，避免过早陷入局部最优。相较粒子群算法、模拟退火算法等传统元启发搜索算法，麻雀搜索算法的收敛速度更快，寻优能力更强，因此也在近些年得到广泛应用。

2.TPE 搜索算法

作为贝叶斯优化算法的变体，TPE 搜索算法通过智能采样，用尽可能少的采样次数搜寻最优的参数组合。区别于传统贝叶斯优化方法，TPE 算法采用了 Parzen 窗估计器对参数的分布进行建模，并使用树结构处理各参数之间的依赖关系。

算法开始时，在搜索空间中随机采样若干个点并进行评估，并保存评估结果。对于每一次迭代，将历史采样观测集，按评估结果排序，并按照一定的比例分为性能优良和较差的两个集合G和B，并按照以下公式对这两个集合建立核密度估计模型：

$$p(x|G) = \frac{1}{|G|}\sum_{x_i \in G} K(x, x_i) \qquad \text{公式(17)}$$

$$p(x|B) = \frac{1}{|B|}\sum_{x_i \in B} K(x, x_i) \qquad \text{公式(18)}$$

其中K是核函数；接着在$p(x|G)$中选择新的采样点，使得 $p(x \mid G)/p(x \mid B)$ 的值尽可能大，以保证新采样的观测在下一次迭代时被划到性能优良的集合之中。而在计算$p(x|G)$、$p(x|B)$的值时，树状结构能够较好地处理参数之间的层级依赖，并保证了高效地在高维解空间中进行搜索[17]。

TPE 搜索算法能够通过较少的采样次数获得更优的参数组合，而且在高维搜索之中表现优异，而树状结构又保证其可以处理离散/连续混合参数空间，因此非常适合深度学习模型的超参数调优任务。

六、模型构建与实验

（一）搭建实验环境

本实验代码使用 Python 语言编写，调用了当下主流的深度学习库 PyTorch 以搭建卷积神经网络模型，并使用 Pandas、OpenCV 等库对图像数据进行读取、变换。为加速模型训练过程，实验依赖 NVIDIA GPU、CUDA 框架以及 Cudnn 神经网络训练加速模块。

（二）数据的读取与处理

使用 Torchvision 中的 ImageFolder 类，可以方便地读取已经完成分类且存放在树状文件夹中的图片数据。加载之后的数据集已经分好训练集与验证集，我们随机抽出了验证集中 2200 张图片作为测试集。以下是训练集与验证集中各个类别图片的数量统计：

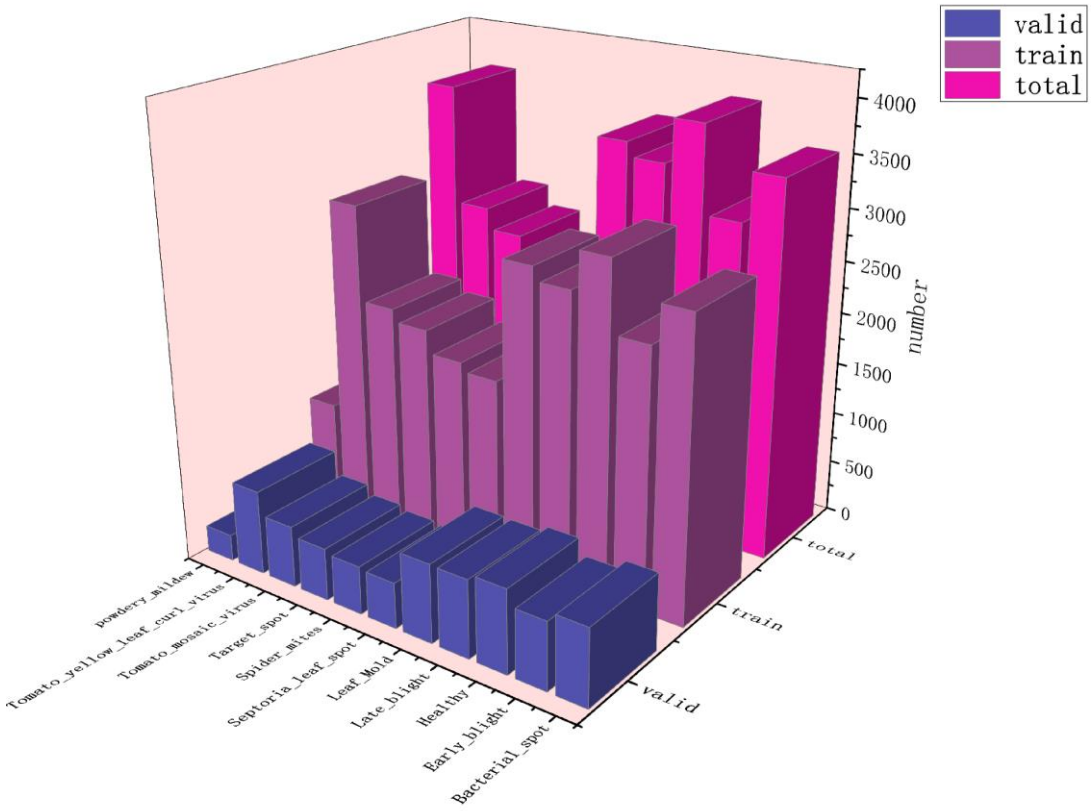

图 6 训练集与验证集中各类别图片数量统计图

由于训练集中各个类别的数据量并没有出现极端不平衡的现象，因此我们没有使用相关算法均衡数据集。

在读取数据之后，我们选择对图片做数据增强处理，包括对图片进行平移、旋转、翻转、缩放、对比度调整等操作，以增加数据的多样性，从而防止模型过拟合。下图展示了对一张训练集进行数据增强的过程：

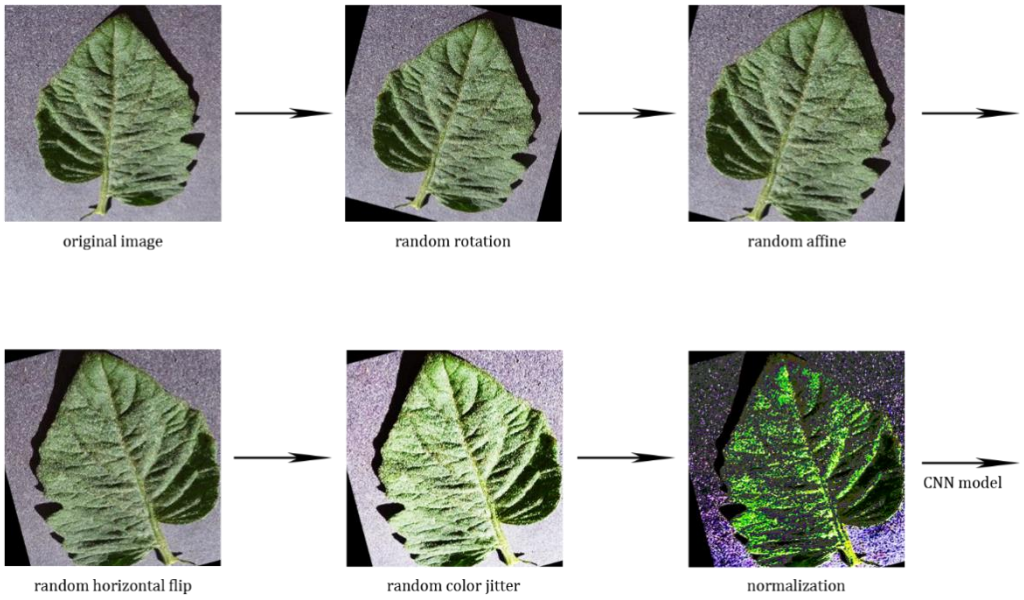

图 7 数据增强过程图

经过数据增强的图像被进一步转化为张量形式，并按通道进行归一化处理。归一化后 RGB 三通道数据的分布近似是均值为$[0.485, 0.456, 0.406]$，方差为$[0.229, 0.224, 0.225]$的正态分布。

（三）模型的构建与调整

在本研究中，我们使用了三种不同的 CNN 模型——VGG16 模型、ResNet50 模型、ResNeXt-CBAM 模型进行对比试验，以深入探究不同架构的神经网络在训练过程中与最终表现上的差别。

1.VGG16 模型

VGG（**Visual Geometry Group Network**）是一种经典的卷积神经网络模型，由 Visual Geometry Group 提出，并在 ImageNet 数据集上效果优秀。VGG 相较之前的卷积神经网络，拥有更小的卷积核与下采样核尺寸，从而减少了模型的参数量，并增加了模型的深度，以便提取更多有效特征[19]。以下是 VGG16 模型的结构示意图：

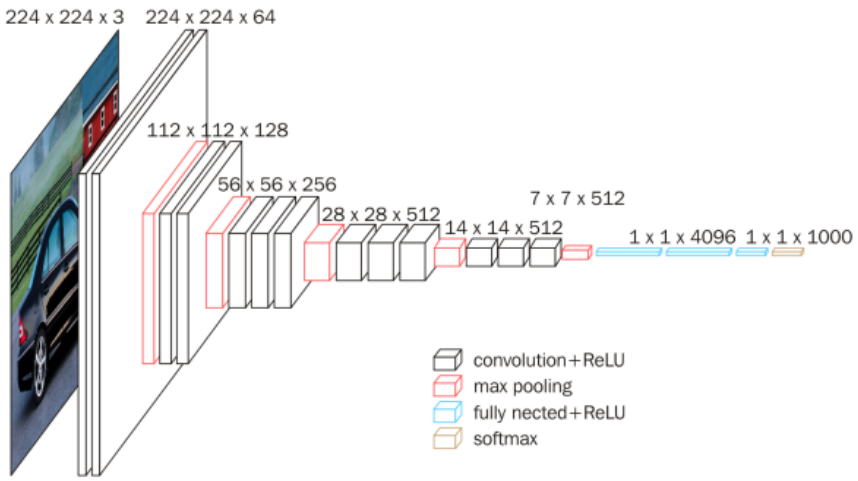

图 8 VGG 模型结构示意图

2.ResNet50 模型

随着卷积神经网络模型深度的增加，学者发现，模型在训练集上的准确率会趋于饱和甚至下降，而这并非过拟合所为。这种现象被称为退化现象，其原因是在模型训练时，深模型中梯度的反向传播受阻，模型参数难以得到有效优化。

为解决退化现象对模型深度的问题，Kaiming He 等人于 2017 年提出了残差神经网络（Residual Network）。该网络在瓶颈卷积模块（Bottleneck）中引入了桥接（Shortcut）的概念，这个结构直接将卷积模块的输入与输出相加，为提供了梯度绕开卷积层继续反向传播的途径，从而有效缓解了深层网络难以优化的问题[11]：

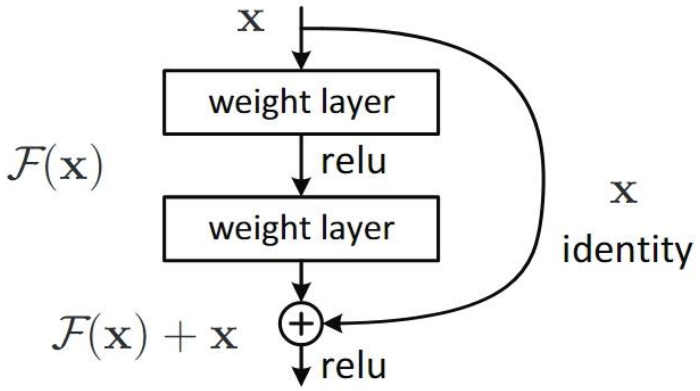

图 9 ResNet 网络原理图

ResNet50 模型作为一种常用的残差神经网络模型，拥有 4 个残差块共计 50 个卷积层，其使用卷积核边长为 7 的卷积层作为感知层，并简化了全连接层与分类器的结构。

在搭建 ResNet50 模型时，我们注意到，该模型的感知层卷积核边长为 7，移动步长为 2，输出通道数直接从 3 膨胀至 64；经过分析我们认为，感受野过大可能会造成感知层无法有效提取图像特征，通道膨胀过陡会导致特征丢失，因此我们将感知层替换为 3 层卷积核尺寸为 3 的卷积层，通过多级卷积缓解通道膨胀陡度，从而有效提取图像特征。

3.ResNeXt-CBAM 模型

ResNeXt 神经网络模型是 ResNet 模型的重要改进，其参数量与普通 ResNet 模型相当，但模型效果却优于 ResNet，并在训练时更为高效。其在瓶颈卷积模块（Bottleneck）的中间卷积层。受 Inception 模型中多路径处理的启发，其引入基数（Cardinality）的概念，将该卷积层按基数分解为多个通道组，每个组独立进行卷积操作，且各组的结构完全一致[20]：

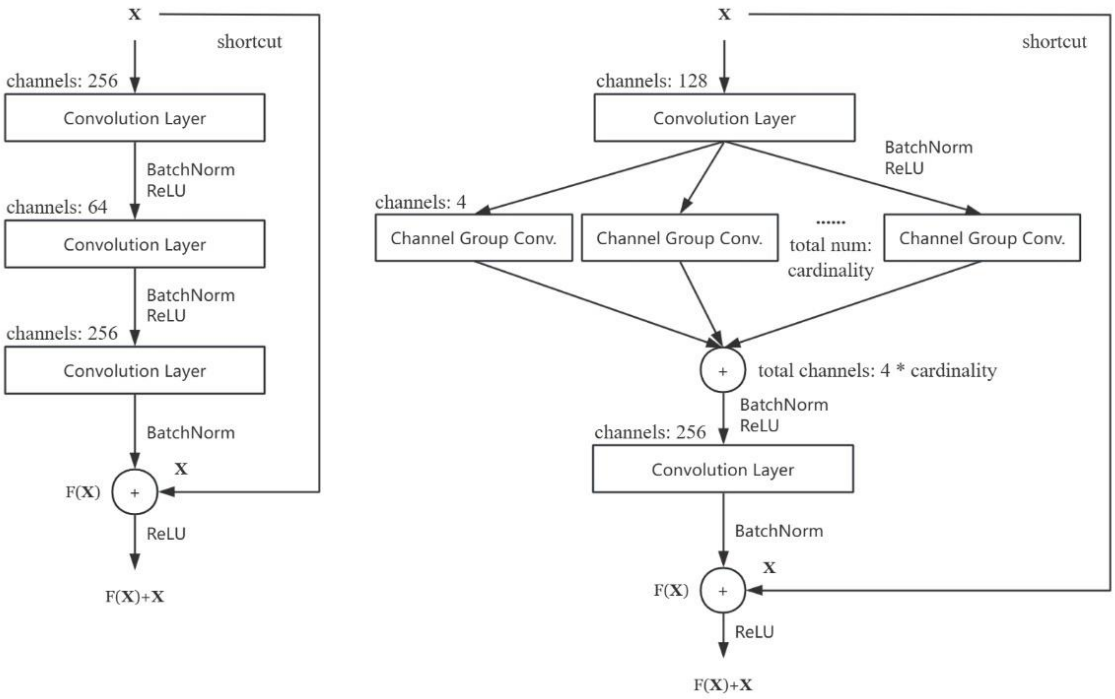

图 10 ResNeXt 神经网络结构图

在 ResNeXt50 的基础上，我们创新性地引入了自注意力机制（Self-Attention），即 CBAM（Convolutional Block Attention Module）模块[21]。自注意力

机制旨在探寻输入信息内部各部分之间的相关性，并为各部分数据赋予对应的权重，并对信息进行加权求和之后送至下一层进行处理，使得模型将注意力放在主要信息上，忽略次要特征和噪声。

2017 年 Google Brain 提出的 Transformer 架构，使得自注意力机制被广泛应用于自然语言处理（NLP）相关领域[12]，而自注意力机制在计算机视觉任务中也有非凡的效果。在图像识别上，在 CBAM 模块中，自注意力机制主要用于明确图像不同通道之间的权重关系，以及不同空间位置之间的权重关系，由 SEBlock（Squeeze & Excitation Block）和 SABlock（Spatial Attention Block）两个子模块实现。

（1）SEBlock

SEBlock 的主要目的是为每一个通道赋予一个注意力权值，并将通道内每个数据与对应权值的积作为输出结果。注意力权值通过一个线性全连接的挤压-激励（Squeeze & Excitation）单元进行学习，缩减比例（reduction ratio）决定了这个单元对信息的挤压程度。

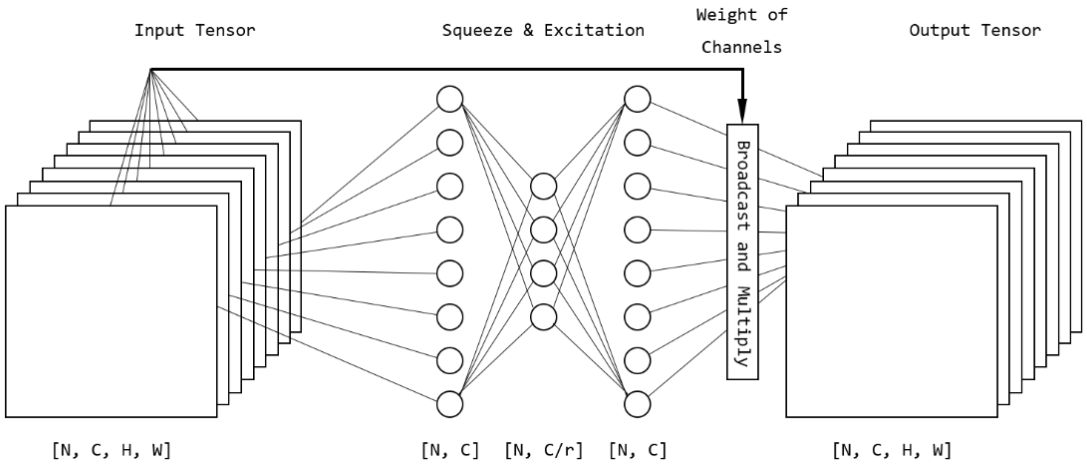

图 11 SEBlock 模块结构示意图

（2）SABlock

相比于 SEBlock 对通道注意力权重的学习，SABlock 偏向于学习图像的空间注意力权重。这个模块对输入数据进行跨通道最大下采样、平均下采样，将通

道数降为 2。对于结果张量进行卷积操作，生成的单通道张量便是每个像素的注意力权值。将该权值与输入张量的每个通道按元素相乘即可得到输出。

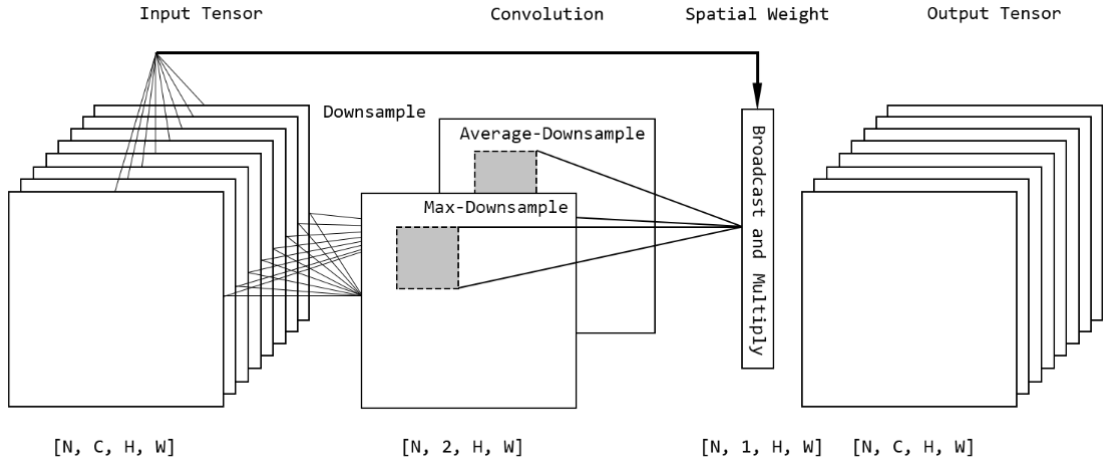

图 12 SABlock 模块结构示意图

以上是 CBAM 模块的基本构造。在 CBAM-ResNeXt 模型中，CBAM 模块位于每一个瓶颈卷积模块的卷积层之后，整个模型一共有 16 个 CBAM 模块，有效提升了模型对主要特征的注意力，引导模型忽略次要特征和背景噪声。

（四）模型的训练与优化

本研究对 CNN 模型的实验分为三组。第一组为模型探究实验，用于测试 VGG16、ResNet50、ResNeXt-CBAM 三种模型的性能，并选取其中表现最好的模型做进一步优化；探究实验的训练轮数为 25。第二组为模型超参数搜索实验，用于搜索能使模型损失快速收敛的超参数组合，每一次搜寻尝试会对模型进行 10 轮训练。第三组实验为模型优化实验，该实验用于在确定最优超参数组合后，对模型进行深度训练，该实验的训练轮数为 50。灵活调整不同实验中的训练轮数，可以保证高效获得实验结果。

1.训练时学习率动态调整策略

在深度学习模型的训练过程中，若学习率始终保持不变，模型效果难以进一步提高。而动态调整学习率，可使模型参数趋于收敛后仍旧能进一步做局部优化。在本实验中，我们使用了学习率阶梯式衰减策略，即每训练 10 轮，学习率衰减为原来的 0.1 倍。

2.模型超参数初始值

表 3 模型超参数初始值一览表

超参数名称	变量名	初始值
初始学习率	learning_rate	0.0001
每批次样本数	batch_size	16
Adam动量衰减系数	beta1	0.9
Adam二阶矩衰减系数	beta2	0.999
L2正则化系数	l2_lambda	0.00001

3.不同模型在探究实验中的表现

在对不同模型的训练中我们发现，对模型的参数按一定方式进行初始化，可以有效加速模型收敛，并避免梯度消失或梯度爆炸。研究表明，Kaiming 初始化方案对于使用 ReLU 激活函数的模型较为有效。Kaiming 初始化使模型参数服从正态分布$N(0,\frac{2}{n_{in}})$，其中n_{in}是输入特征数，以保证特征信息X在经过 ReLU 激活函数前后方差基本一致[14]。以下是三种模型在应用 Kaiming 初始化方案前后的训练曲线：

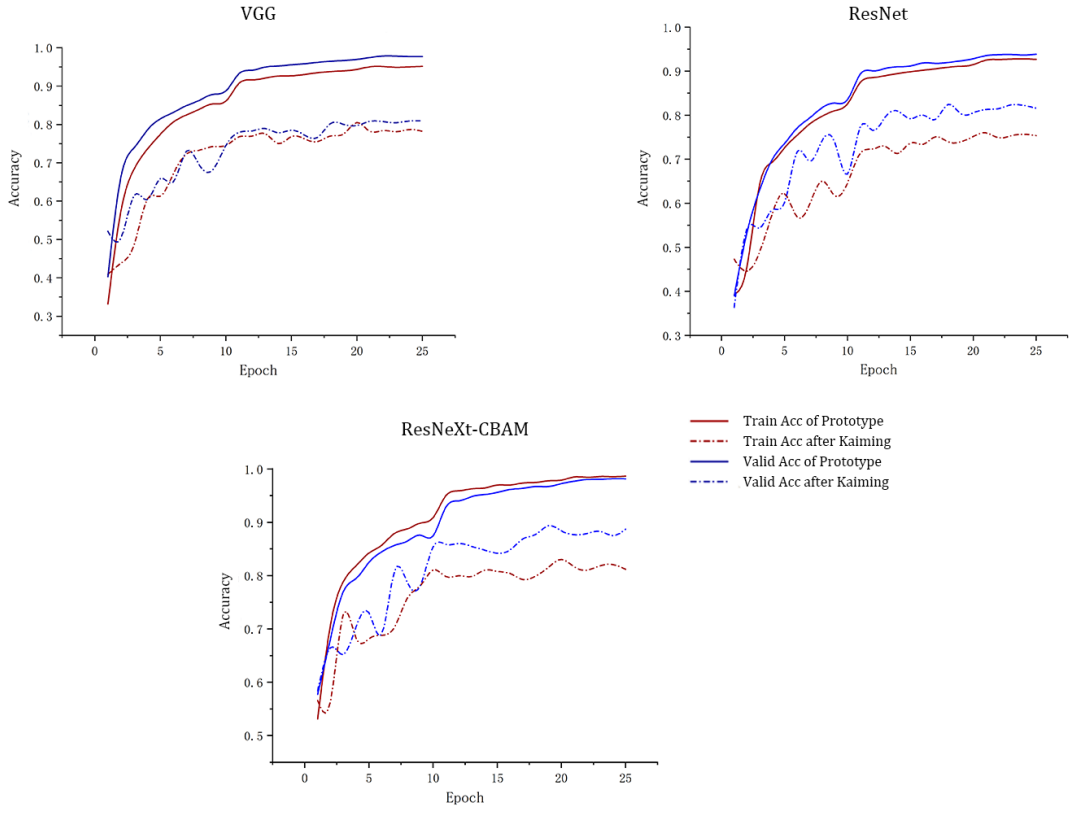

图 13 模型 Kaiming 初始化方案前后的训练曲线

对三种不同的 CNN 模型进行对比试验后，我们得到了这些模型在测试集上的准确率和 Kappa 系数：

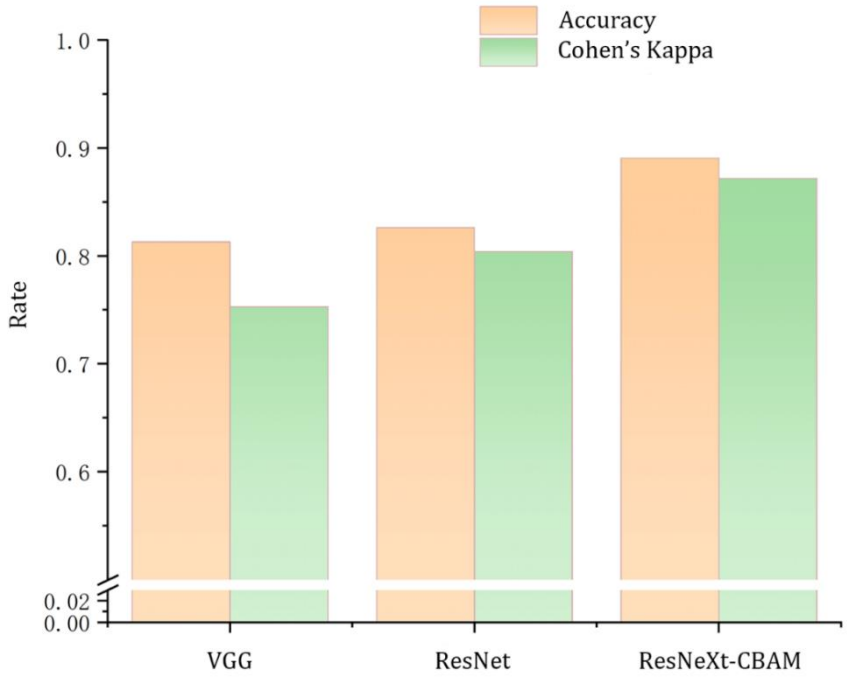

图 14 模型在测试集上的准确率、Kappa 系数对比图

对上述图表数据进行分析，我们可以得出结论：ResNeXt-CBAM 模型在当前任务上表现最好。VGG16 模型的参数量过大，模型复杂度高，有很大的过拟合风险。ResNet50 模型相较于 VGG16 在效果上改善明显，但应对噪声和背景干扰的能力逊于 ResNeXt-CBAM。由于引入了自注意力机制，ResNeXt-CBAM 在当前任务中可以将注意力更多地放在叶片的形状与纹理上，对不同类别图像的特征提取能力更强，因此在当前任务上表现更优。基于上述论证，我们选择 ResNeXt-CBAM 模型做后续实验。

4. ResNeXt-CBAM 模型超参数搜索实验

在模型超参数搜索实验中，我们对比使用了 SSA 搜索算法、TPE 搜索算法。下表为模型的待搜索超参数以及其取值范围：

表 4 模型的待搜索超参数信息一览表

超参数名称	变量名	类型	取值范围
初始学习率	learning_rate	连续型	$[10^{-5}, 10^{-3}]$
Adam动量衰减系数	beta1	连续型	[0.8,0.99]
Adam二阶矩衰减系数	beta2	连续型	[0.98,0.9999]

L2正则化系数	wd	连续型	$[10^{-6},10^{-4}]$
dropout概率	dropout_rate	连续型	[0.3,0.6]
基数	cardinality	离散型	[16,32,64]
瓶颈宽度	base	离散型	[2,4,8]
SEBlock缩减比例	SE_reduction_ratio	离散型	[8,16,32]
SABlock卷积核尺寸	SA_kernel_size	离散型	[3,5,7,9]

通过两种搜索算法对比，我们选择 TPE 搜索算法对超参数进行优化。其原因在于 TPE 算法对于混合型搜索空间更为友好，且 SSA 算法在高维搜索空间表现一般。

在超参数搜索实验中，我们评估当前超参数优劣的方法是，以当前超参数在小数据集上对模型进行 10 次训练，并评估模型在测试集上的准确率。准确率越高，当前超参数的效果越好。以下是使用 TPE 搜索算法对最优超参数组合进行搜索的过程：

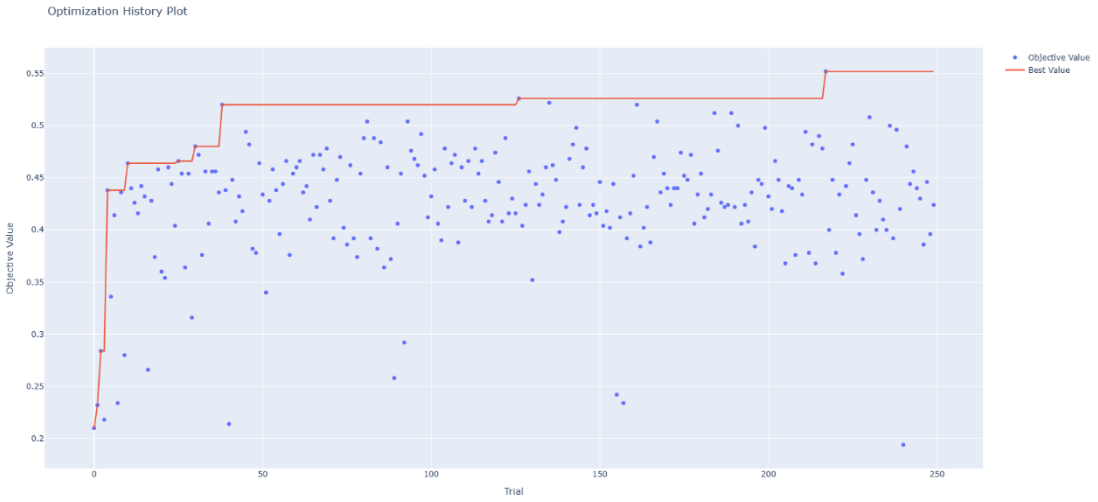

图 14 TPE 搜索算法搜索最优超参数组合过程图

根据结果，我们同样能得到不同超参数对结果的影响程度：

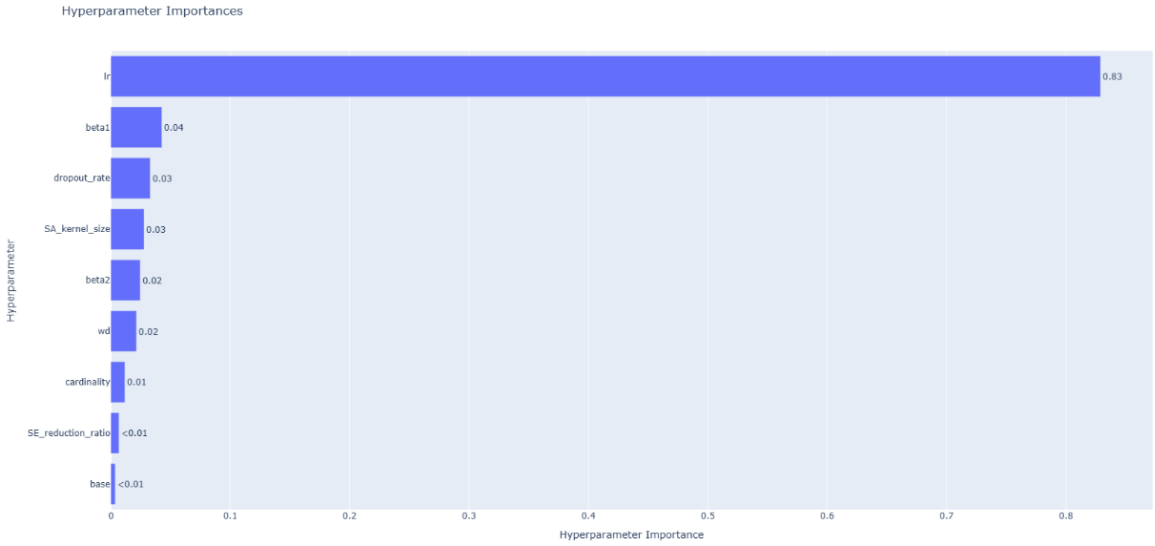

图 15 不同超参数对结果的影响程度对比图

可见，在所有超参数之中，学习率（lr）对模型训练效果的影响非常大。实际上，无论对哪一种深度学习模型，训练时初始学习率都需要谨慎选择。

下表展示了 TPE 算法搜寻到的最优超参数组合：

表 5 TPE 算法搜寻到的最优超参数组合信息表

超参数名称	变量名	类型	最佳取值
初始学习率	learning_rate	连续型	7.88×10^{-4}
Adam动量衰减系数	beta1	连续型	0.86857
Adam二阶矩衰减系数	beta2	连续型	0.99009
L2正则化系数	wd	连续型	2.80×10^{-5}
dropout概率	dropout_rate	连续型	0.46643
基数	cardinality	离散型	16
瓶颈宽度	base	离散型	4
SEBlock缩减比例	SE_reduction_ratio	离散型	16
SABlock卷积核尺寸	SA_kernel_size	离散型	3

5.ResNeXt-CBAM 模型优化实验与有效性验证

以下是 ResNeXt-CBAM 模型在 50 轮深度训练中的训练曲线与训练结果：

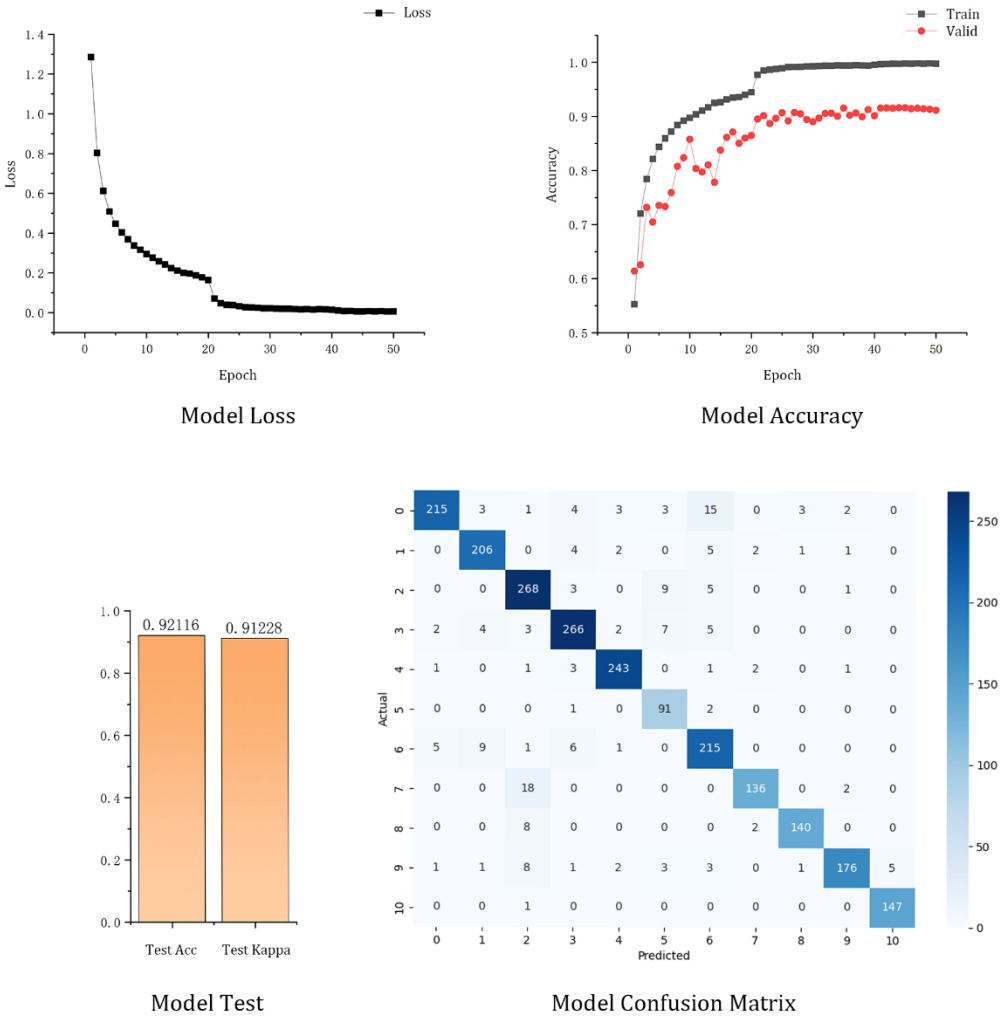

图 16 ResNeXt-CBAM 模型在 50 轮深度训练中的训练结果

（五）模型的扩展与应用

为了使训练后的 ResNeXt-CBAM 能够在更为复杂的图像上同样表现优异，我们同时训练了一个 YOLOv8-n 模型，用于对图像中的番茄叶片目标进行检测。

YOLO 是一种常用的图像目标检测模型架构[13]，其基本工作原理是判断输入图像中是否存在目标对象，并使用矩形框紧确地框选出可能是目标的物体。在本研究中，我们选择了 1136 张形态各异的番茄叶片图像，并手动框选出图像中的目标，并将图像与目标位置作为训练数据送入 YOLOv8-n 模型进行训练。在进行 100 轮训练之后，YOLOv8-n 模型已经可以较为精确地框选出验证图像中的目标。下图展示了训练后的 YOLO 模型对一些验证图像中目标的检测情况：

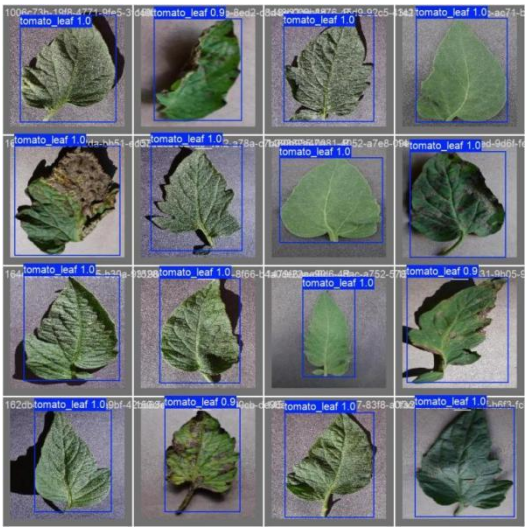

图 17 训练后 YOLO 模型对验证图像的检测情况

通过 ResNeXt-CBAM 和 YOLO 模型的组合使用，研究得到的模型得以适用于更大的应用场景。以下是组合模型的基本工作原理图：

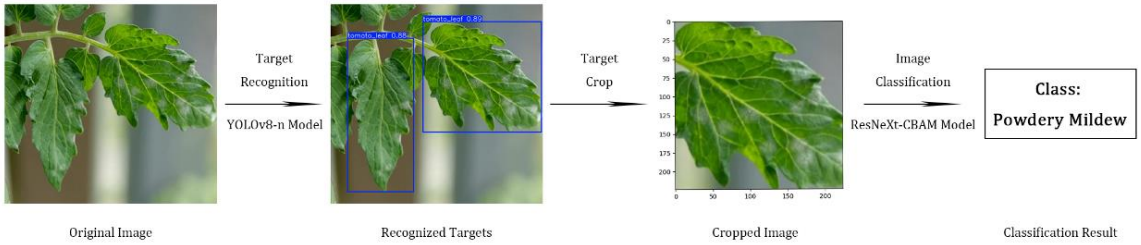

图 18 组合模型的基本工作原理图

使用 YOLOv8-n 识别待预测图像中的目标并将其裁剪，将裁剪后的对象再送进 ResNeXt-CBAM 模型进行预测，可以有效排除 ResNeXt-CBAM 模型因背景图像干扰造成的错误，提高了整个模型预测的鲁棒性，使得模型可以在更广泛的应用场景工作。

考虑到模型功能的可拓展性，我们对目标检测、图像识别进行单独建模，使目标检测与图像识别功能解耦。由于训练性能优秀的 YOLO 模型，需要大量

带有目标选框的、类型齐全的训练数据，这些数据往往只能由人工标注，这无疑会提高整个模型训练的成本。而若将两个功能独立建模，对于 YOLO 模型而言，就无需关注训练数据的类型是否齐全，而用于分类的 ResNeXt-CBAM 模型也无需关注目标选框。当模型功能需要扩展时，我们仅需搜集新类型的图片并送入 ResNeXt-CBAM 训练，无需再对其进行人工目标选框。综上所述，使用目标检测模型、图像识别模型组合模型再扩展性上更胜一筹。

七、结论与展望

（一）研究结论

1.ResNeXt-CBAM 模型能够较好地完成不同番茄叶片病害的识别任务。根据实验数据可知，本模型在 11 类番茄病害数据集上平均识别准确率达 91.98%，Kappa 系数 0.911，较现有农业模型（如 MobileNetV3）提升 0.98%，能够较好的完成病害检测任务。

2.Kaiming 模型参数初始化方案可以有效加速模型收敛。采用 Kaiming 参数初始化后，模型在 25 个 epoch 内达到较高验证准确率，较未初始化时收敛速度大幅提升。实验表明，该方案使训练初期损失值下降曲线平滑度大幅提升，有效避免梯度消失现象。

3.TPE 搜索算法可以有效搜索模型较优超参数。对比网格搜索与随机搜索，TPE 算法在相同计算资源下将超参数搜索时间大大缩短，获得的超参数组合使模型测试集准确率与 Kappa 系数得以提升。

4.目标检测模型与图像识别模型组合可以更好地解决复杂背景图像识别问题。通过 YOLOv8 目标检测模型预筛选叶片区域，结合 ResNeXt-CBAM 分类模型，使得模型更好地处理背景复杂的图像。

（二）研究不足

1.深度学习模型学习过程难以解释。本研究构建的 ResNeXt-CBAM 模型虽然实现了高精度分类，但其决策过程仍呈现典型的“黑箱”特性，容易导致农民信任度低、模型优化盲区等问题。

2.模型性能过于依赖数据集质量与多样性。现有数据集以实验室环境为主，真实田间场景样本占比不足，可能会影响模型的使用性能。

3.未能解释超参数对模型性能的影响机制。尽管为轻量化模型，但在无 NPU 加速的设备上推理延迟仍然较高，难以满足部分实际场景的需要。

（三）展望

1.引入可视化工具

为本研究所用模型建立可视化操作，降低模型的使用难度，提高模型使用的受众，挺高模型的实用性，助力农民快捷学习使用。

2.跨作物研究

将本研究所用的模型方法迁移至其他作物，从而增大模型的覆盖面，进一步提升模型在农业生产的帮助。

3.计算优化和系统集成

进一步提升模型的推理能力和速度，使其能够更快的对病害做出诊断。

参考文献

[1] Amara, J., et al. (2017). A deep learning-based approach for banana leaf diseases classification. International Journal of Advanced Computer Science and Applications.

[2] Bergstra, J., et al. (2011). Algorithms for hyper-parameter optimization. NeurIPS.

[3] Hughes, D., & Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. Plant Phenomics.

[4] Mohanty, S. P., et al. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science.

[5] Picon, A., et al. (2019). Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. Computers and Electronics in Agriculture.

[6] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data.

[7] Xue, J., & Shen, B. (2020). A novel swarm intelligence optimization approach: Sparrow search algorithm. Systems Science & Control Engineering.

[8] Zhang, S., et al. (2021). CycleGAN-based data augmentation for improving plant disease recognition under limited data. Computers and Electronics in A

[9] Alzubaidi, L., Zhang, J., Humaidi, et al. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*.

[10] Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A Survey of the Recent Architectures of Deep Convolutional Neural Networks. Artificial Intelligence Review.

[11] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[12] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is All you Need. Neural Information Processing Systems.

[13] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[14] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. 2015 IEEE International Conference on Computer Vision (ICCV).

[15] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv: Learning.

[16] Srivastava, N., Hinton, Geoffrey E., et al. (2014). Dropout: a simple way

to prevent neural networks from overfitting. Journal of Machine Learning Research.
[17]Ozaki, Y., Tanigaki, Y., Watanabe, S., & Onishi, M. (2020). Multi-objective tree-structured parzen estimator for computationally expensive optimization problems. Proceedings of the 2020 Genetic and Evolutionary Computation Conference.
[18]Kingma, Diederik P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv: Learning.
[19]Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations.
[20]Xie, S., Girshick, R., Dollar, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[21]Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. In Computer Vision – ECCV 2018, Lecture Notes in Computer Science (pp. 3–19).