

梁凯博, 孙立, 汪禹治, 等. 基于超轻量化卷积神经网络的番茄病虫害诊断[J]. 江苏农业学报, 2024, 40(3): 438-449.
doi:10.3969/j.issn.1000-4440.2024.03.006

基于超轻量化卷积神经网络的番茄病虫害诊断

梁凯博^{1,2}, 孙立¹, 汪禹治², 靳龙豪¹, 燕雪倩³, 曾旺¹

(1.北京物资学院信息学院, 北京 101149; 2.首都经济贸易大学管理工程学院, 北京 100070; 3.北京物资学院数据科学与统计学院, 北京 101149)

摘要: 针对番茄病虫害诊断中存在的传统卷积神经网络结构复杂、难以直接应用于便携终端, 以及现有轻量化卷积神经网络特征提取能力弱、识别准确率低、难以满足实际需要等问题, 本研究拟在原有轻量化卷积神经网络的基础上, 定义超轻量化卷积神经网络, 设计一种基于 SqueezeNet 网络改进的超轻量化卷积神经网络, 将其用于番茄病虫害诊断任务中。首先, 改进 SqueezeNet 网络中的 Fire 模块, 生成 2 种适用于不同特征维度的 Fire 模块, 并引入 ECA(高效通道注意力)模块以提高模型的特征提取能力; 其次, 结合扩展型指数线性单元函数(SELU)和 Mish 函数, 替代修正线性单元函数(ReLU)作为激活函数; 再次, 采用软池化(Softpool)替代原始的最大池化; 最后, 利用中心损失函数(Center loss)改进指数归一化损失函数(Softmax loss), 提高对近似病虫害的识别准确率。本研究选择了 8 种害虫和 9 种病害, 对害虫、病害、病虫害 3 类数据集进行数据增强, 并探讨了数据的小样本性、不平衡性对模型性能的影响。结果表明, 本研究提出的模型具有超轻量化的特点, 对害虫、病害、病虫害的识别准确率最高分别可达 98.83%、98.14% 和 97.71%, 能够很好地满足番茄病虫害诊断需求。

关键词: 图像识别; 番茄病虫害; 超轻量化卷积神经网络; 不平衡性

中图分类号: S641.2; TP391.41 **文献标识码:** A **文章编号:** 1000-4440(2024)03-0438-12

Diagnosis of tomato pests and diseases based on super lightweight convolutional neural network

LIANG Kai-bo^{1,2}, SUN Li¹, WANG Yu-zhi², JIN Long-hao¹, YAN Xue-qian³, ZENG Wang¹

(1. School of Information, Beijing Wuzi University, Beijing 101149, China; 2. School of Management Engineering, Capital University of Economics and Business, Beijing 100070, China; 3. School of Data Science and Statistics, Beijing Wuzi University, Beijing 101149, China)

Abstract: In the diagnosis of tomato diseases and pests, traditional convolutional neural network structures are complex and hard to be directly applied to portable terminals. Besides, existing lightweight convolutional neural networks exhibit weak feature extraction capabilities, low recognition accuracy, and are inadequate for practical applications. Aiming at the above problems, we intended to define a super lightweight convolutional neural network based on existing lightweight convolutional neural network, and to design an ultra-lightweight convolutional neural network by improving the SqueezeNet network for tomato disease and pest diagnosis tasks. Firstly,

we enhanced the Fire module in the SqueezeNet network, generated two Fire modules suitable for different feature dimensions. We introduced efficient channel attention (ECA) module to improve feature extraction capabilities of the model. Secondly, we incorporated scaled exponential linear unit (SELU) and Mish to replace rectified linear unit (ReLU) as activation function. Next, we employed Softpool instead of the original max pooling. Finally, we

收稿日期: 2023-03-31

基金项目: 广东省重点领域研发计划项目(2019B020214002); 北京市社会科学基金项目(20GLB026); 国家自然科学基金项目(71771028); 首都经济贸易大学研究生科技创新项目(2023KJCX062)

作者简介: 梁凯博(1997-), 男, 河北秦皇岛人, 博士研究生, 研究方向为深度学习、人工智能。(E-mail) liangkaiibo2955@gmail.com

通讯作者: 孙立, (E-mail) slsally@163.com

enhanced the exponential normalized loss (Softmax loss) by using Center loss function to improve the recognition accuracy of approximate diseases and pests. In this experiment, we selected eight types of pests and nine types of diseases to perform data augmentation on three datasets (pests, diseases, diseases and pests), and investigated the impact of small sample and data imbalance on model performance. Experimental results demonstrated that the network proposed in this study had super lightweight characteristics. The recognition accuracies for pests, diseases, and diseases and pests could reach up to 98.83%, 98.14% and 97.71%, respectively, which met the requirements for diagnosis effectively.

Key words: image recognition; tomato pests and diseases; super lightweight convolutional neural network; imbalance

精准农业是一种利用现代技术和数据分析方法来优化农业生产、对农田和作物进行精确管理的模式^[1]。随着数字智能化技术的不断完善,基于精准农业理念进行精细化生产和病虫害管理逐渐成为农业发展的主流。番茄是一种常见蔬菜,其产量在世界范围内仅次于马铃薯^[2]。番茄病虫害是威胁番茄产量和品质的主要因素之一^[3],因此番茄病虫害诊断对产业发展至关重要。

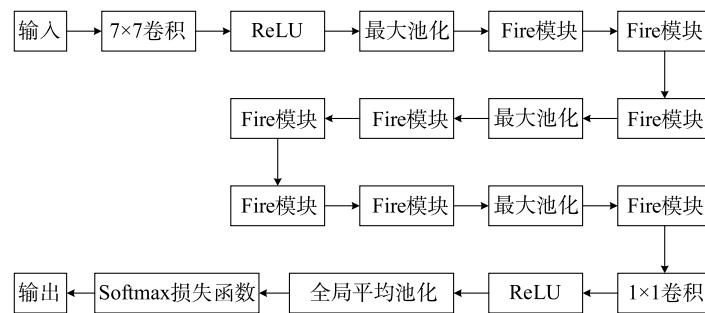
目前,病虫害快速诊断主要通过轮廓识别和深度学习实现^[4-5],但传统轮廓识别方式依赖于人工提取特征,难以满足自动化需要^[6]。在以往病虫害诊断与防治的大规模研究中^[7-8],由于病虫害种类繁多,形态复杂,传统轮廓识别方式难以满足精确诊断的要求,因此学者们一方面开发基于深度学习的病虫害诊断方法^[9-10],另一方面大规模收集数据集以满足深度学习对于特征的提取要求^[11]。然而,在农业领域,病虫害防治一般位于田间^[12-13],便携式终端难以运行复杂网络,现有轻量化模型的准确率难以满足实际需求。此外,受拍摄条件影响,所获数据集不能保证数据量充足以及各子数据集之间的平衡,成为病虫害精准防治的巨大阻碍。

本研究拟基于轻量化模型 SqueezeNet,通过对其基本结构、激活函数、损失激活函数分别进行改进、优化,构建一种超轻量化网络模型 SSCNet (Squeeze-Smish-Centersoftmax Net),对 8 种番茄虫害和 9 种番茄病害进行识别研究,以期更好地实现番茄病虫害诊断的轻量化、高效化。

1 材料与方法

1.1 模型介绍

1.1.1 SqueezeNet 网络 随着深度学习研究的不断深入,卷积神经网络 (Convolutional neural network, CNN) 的应用朝着轻量化、高精度方向发展。通过精简可学习的参数数量、减少网络整体计算量等方式,能够有效提升训练效率、改善存储效率、节省占用内存,SqueezeNet^[14]是轻量化网络的代表,具体结构见图 1。通过将 3×3 卷积替换为 1×1 卷积,并进行降采样后置,使其参数量大幅度减少。此外,Fire 模块作为网络主干,由 Squeeze 和 Expand 2 部分构成,其中,Squeeze 部分由 1 组连续 1×1 卷积组成,Expand 部分由 1 组连续 1×1 卷积和 1 组连续 3×3 卷积融合而成,保证模型性能的同时大幅度提升模型效率。



ReLU:修正线性单元函数。

图 1 SqueezeNet 的结构

Fig.1 Structure of SqueezeNet

然而,由于现有版本的 SqueezeNet 存在结构简单、参数量少等特性,使其在处理大规模数据集或多

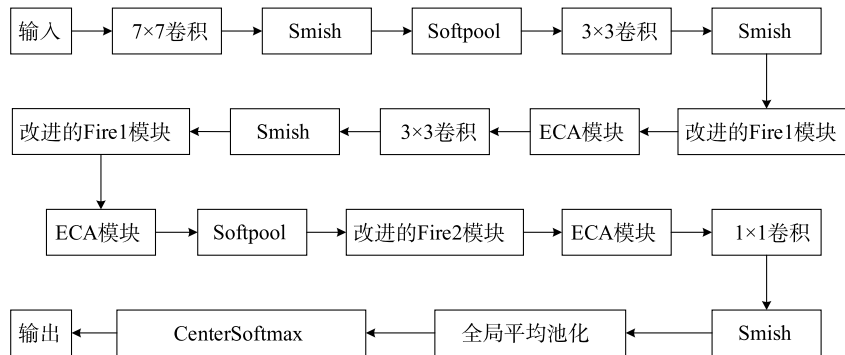
分类数据集时效果不佳^[15-18];此外,尽管 SqueezeNet 已基本实现轻量化,但仍有提升空间^[19]。因此,本

研究在现有轻量化网络的基础上,将超轻量化网络定义为:基于高效的网络计算方式,使参数量不超过 1×10^6 ,并在处理图像识别与分类任务中实现较高精度的一种网络类型。

在上述定义的基础上,本研究对 SqueezeNet 进行改进优化,试图达到网络性能与效率之间的平衡。

1.1.2 SSCNet 网络 本研究对 SqueezeNet 网络进行优化,提出了 SSCNet 网络(图 2)。具体优化情况

如下:第一,改进 Fire 模块,引入 ECA (Efficient channel attention) 模块;第二,结合扩展型指数线性单元函数 (SELU) 与 Mish 激活函数,提出一种平滑混合激活函数 (Smish),并将其应用到改进网络中;第三,用软池化 (Softpool) 替代最大池化;第四,基于中心损失函数 (Center loss) 改进指数归一化损失函数 (Softmax loss),提出一种中心指数归一化损失函数 (CenterSoftmax)。



CenterSoftmax: 中心指数归一化损失函数; Smish: 平滑混合激活函数; Softpool: 软池化; ECA: 高效通道注意力模块。

图 2 SSCNet 的结构

Fig.2 Structure of SSCNet

1.1.3 SSCNet 网络的特征提取 特征提取是卷积神经网络中的关键步骤,决定了网络对输入数据的理解能力^[20]。为了提升特征提取能力,SSCNet 网络引入了一系列改进措施。

1.1.3.1 改进的 Fire 模块 Fire 模块(图 3)能够保证模型特征提取能力,减少模型参数。本研究考虑到番茄病虫害图像周遭环境复杂,模型需要提取更高维度特征以进行归一化并分类,基于原 Fire 模块,本研究减少卷积层数量,调整其卷积核大小,并增加批归一化 (Batch normalization, BN)。图 3 显示,改进的 Fire1 模块采用 1×1 和 3×3 的卷积核进行特征初步提取;改进的 Fire2 模块采用 3×3 和 5×5 的卷积核进行高维特征提取。

1.1.3.2 ECA 模块 注意力机制^[21-23]能够赋予卷积神经网络“重点关注”的能力,让网络更加关注那些相对更重要的输入。由于进行番茄病虫害诊断时背景复杂、“噪音”较多,常规网络训练难以获得较好的结果,因此本研究引入 ECA 模块,实现输入特征图通道特征的增强(图 4)。由于病虫害数据集拍摄角度、距离及位置难以保持一致,图像中各类物体大小不一,将 ECA 模块集成到各个 Fire 模块之后,

通过动态调整其卷积核大小来适应不同通道数的特征图,进而实现不同程度的跨通道交互,提升识别病虫害的能力。其公式为:

$$k = \left\lceil \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right\rceil_{\text{odd}} \quad (1)$$

其中, k 为卷积核大小; C 为通道数; $\lceil \cdot \rceil_{\text{odd}}$ 表示 k 只能为奇数; γ 和 b 为控制 k 和 C 比例的超参数,使 k 能够动态地根据 C 选择其大小。此处 γ 设置为 2, b 设置为 1^[22]。

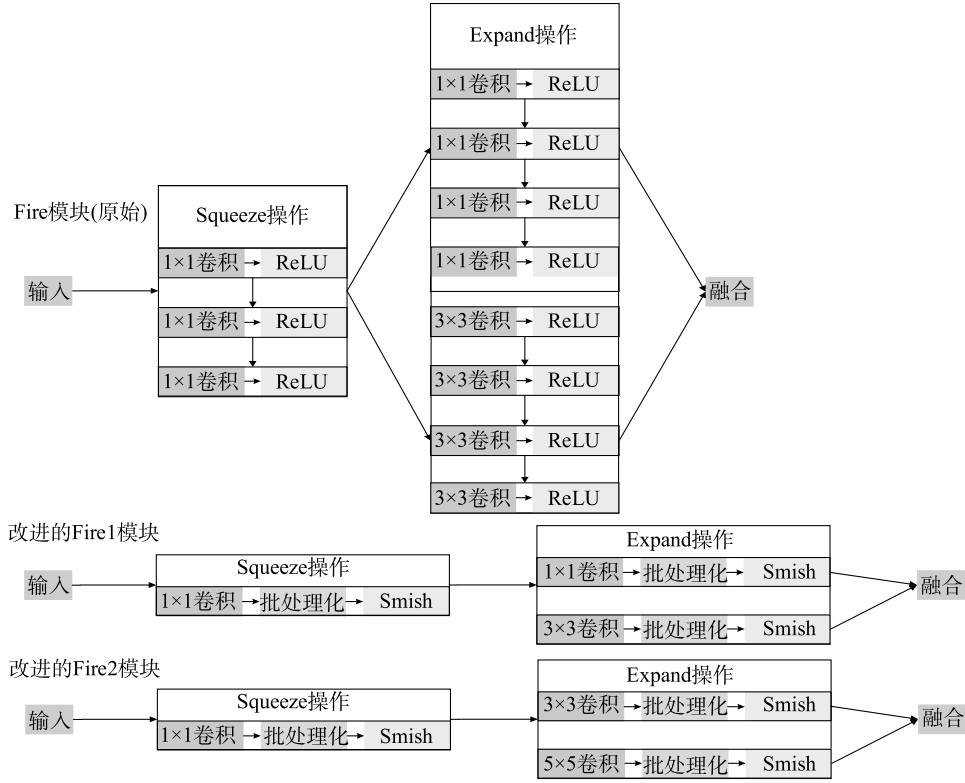
1.1.3.3 平滑混合激活函数 在当前研究中,因结构简单、收敛速度快,修正线性单元函数 (ReLU) 得到了广泛应用,但也存在神经元容易死亡、参数无法更新等问题。目前,大量改进的 ReLU 被相继提出^[24-27],其中扩展型指数线性单元函数能够实现输出内部归一化,及对网络的自归一化,避免出现梯度消失和爆炸等问题,如公式(2)所示:

$$SELU(x) = \lambda \begin{cases} \alpha(e^x - 1), & (x \leq 0) \\ x, & (x > 0) \end{cases} \quad (2)$$

其中, λ 表示输入值。为了使神经网络权重在连续反向传播迭代中稳定,需要使每个层的激活输出的均值为 0,此时就需要 λ 充当平衡因子,由公式

(3) 求解;此外,为了控制函数在负半轴上的指数衰减速度,就需要用 α 来进行控制,由公式(4)求解;互补误差函数 $\text{erfc}(x)$ 则是用来计算 SELU 函数中 λ

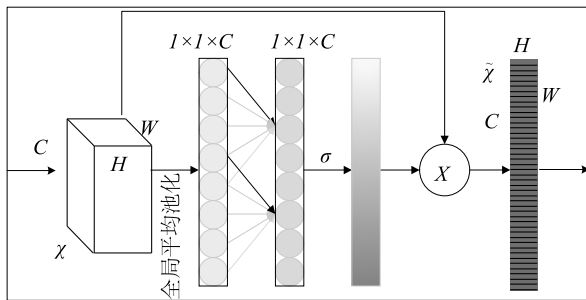
参数的优化值,避免其训练过程中出现梯度消失或爆炸等问题,计算方法见公式(5)。



ReLU:修正线性单元函数;Smish:平滑混合激活函数。

图3 Fire 模块结构对比

Fig.3 Structure comparison of Fire modules



C :通道数; H :高度; W :宽度; X :输入特征; σ :激活函数; \tilde{x} :输出特征; X :2 处特征输入后融合的结果。

图4 高效通道注意力(ECA)模块结构

Fig.4 Structure of efficient channel attention (ECA) module

$$\lambda = [2\text{erfc}(\sqrt{2})e^2 + \pi\text{erfc}(\frac{1}{\sqrt{2}})^2e - 2(2 + \pi)]$$

$$\text{erfc}(\frac{1}{\sqrt{2}})\sqrt{e+\pi+2}]^{-\frac{1}{2}} \approx 1.0507 \quad (3)$$

$$\alpha = -\frac{\sqrt{\frac{2}{\pi}}}{\text{erfc}(\frac{1}{\sqrt{2}})\exp(\frac{1}{2})-1} \approx 1.67326 \quad (4)$$

$$\text{erfc}(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (5)$$

式中, x 为函数的自变量; t 为用于积分运算的虚拟变量,表示函数中某一点的值。

尽管 SELU 避免了 ReLU 中的部分问题,但由于 SELU 依然存在强制性等缺陷,难以进一步提升网络收敛速度,因此本研究引入 Mish 函数与之进行结合。

Mish 函数具有无上界有下界、无穷阶连续性和光滑性等优势。相较于 ReLU 系列函数,Mish 函数的泛化性能更好。如公式(6)~公式(8)所示:

$$\text{Mish}(x) = x \text{Tanh}[\text{Softplus}(x)] \quad (6)$$

$$\text{Tanh}(x) = (e^x - e^{-x}) / (e^x + e^{-x}) \quad (7)$$

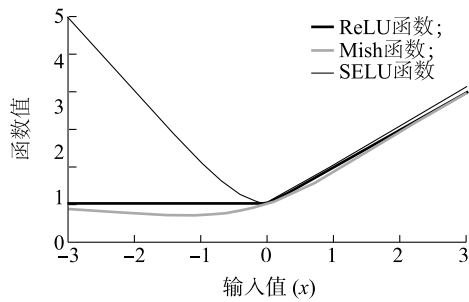
$$\text{Softplus}(x) = \ln(1 + e^x) \quad (8)$$

但 Mish 函数计算复杂,时间成本高,直接应用则与超轻量化理念不符。故本研究在 ReLU 函数的基础上,结合 SELU、Mish 2 种函数特点,设计一种新激活函数,即 Smish 函数(图 5),计算见公式(9):

$$\text{Smish}(x) = \lambda \begin{cases} x \text{Tanh}[\text{Softplus}(x)], & (x \leq 0) \\ x, & (x > 0) \end{cases} \quad (9)$$

对公式(9)求导,如公式(10)所示:

$$\text{Smish}'(x) = \lambda \begin{cases} \frac{e^x \omega}{\delta^2}, & (x \leq 0) \\ 1, & (x > 0) \end{cases} \quad (10)$$



ReLU 函数:修正线性单元函数;Smish 函数:平滑混合激活函数;SELU 函数:扩展型指数线性单元函数。

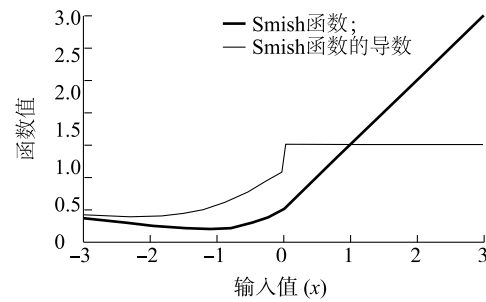
图 5 ReLU、Mish、SELU 函数与 Smish 函数、Smish 函数的导数曲线

Fig.5 Curves of functions of ReLU, Mish, SELU, Smish and derivative of Smish

$$\omega = 4(x+1) + 4e^{2x} + e^{3x} + e^x(4x+6) \quad (11)$$

$$\delta = 2e^x + e^{2x} + 2 \quad (12)$$

当 $x > 0$ 时,Smish 函数取值与 SELU 函数相同,很好地继承 ReLU 系列函数轻量的特点,提高了计算效率;当 $x \leq 0$ 时,Smish 函数取值为 λ 与 Mish 函数的乘积。其中 Mish 函数部分一方面避免了 x 接近负无穷时的“饱和区”,另一方面稳定了网络梯度,能够让神经元得到有效更新。此外,Smish 函数计算公式中保留 SELU 中 λ 的取值,能够确保函数斜率大于 1(图 5),放大病虫害数据间的细微差距。



1.1.3.4 Softpool 常见池化操作有最大池化(Max-pool)和平均池化(Averagepool),但有研究表明上述 2 种池化会丢失大量图像信息,降低网络整体性能^[28]。为保证超轻量化网络性能,本研究引入 Softpool 替代原 Maxpool。根据特征值计算区域(R)的特征值权重,如公式(13)所示:

$$\omega_i = \frac{e^{a_i}}{\sum_{j \in R} e^{a_j}} \quad (13)$$

其中,权重(ω_i)能够保证传递重要特征,区域(R)中特征值在反向传递时都将预设最小梯度, i 和 j 均指池化窗口内像素的索引,而 a_i 和 a_j 分别指池化窗口中第 i 个和第 j 个元素的激活值。得到权重(ω_i)后,通过加权区域(R)内特征值得到的输出为:

$$\tilde{a} = \sum_{j \in R} \omega_j \times a_j \quad (14)$$

1.1.3.5 改进指数归一化损失函数 在多分类识别中,常用的分类损失函数为 Softmax loss^[29],通过对模型输出进行 Softmax 变换,将其转换为概率分布,然后计算 Softmax 输出与实际标签之间的交叉熵损失,从而量化预测输出与实际标签之间的差异,如公式(15)所示:

$$l_{\text{Softmax loss}} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \ln \left(\frac{e^{Z_{i,j}}}{\sum_{k=1}^M e^{Z_{i,k}}} \right) \quad (15)$$

式中, N 是数据集中样本总量; M 是病虫害类别数量; $y_{i,j}$ 是第 i 个样本的第 j 类别的实际标签; $Z_{i,j}$ 、 $Z_{i,k}$ 分别是第 i 个样本分别属于第 j 个、第 k 个类别的模型预测输出; $\sum_{k=1}^M e^{Z_{i,k}}$ 是指第 i 个样本的所有预测输出进行归一化后转换为概率分布; $\frac{e^{Z_{i,j}}}{\sum_{k=1}^M e^{Z_{i,k}}}$ 则是预测第 i 个样本属于类别 j 的概率。

对于病虫害数据集来说,Softmax loss 计算了所有预测输出与实际病虫害标签之间的交叉熵,并将所有样本的交叉熵取平均值作为整个数据集的损失值。然而,尽管相似病虫害之间存在一定的类内变化,但由于 Softmax loss 对这些深层特征不具备足够的辨别能力,因此难以进行识别。

近年来以 Center loss^[30]为代表的解决样本特征类间距过小的函数相继被提出,如公式(16)所示:

$$l_{\text{Center loss}} = \frac{1}{2} \sum_{i=1}^m \|x_i - c_{y_i}\|^2 \quad (16)$$

式中, x_i 指第 i 个样本的特征向量; y_i 则是与 x_i 所对应的类别标签; c_{y_i} 是指 y_i 的类别中心, 即网络训练过程中对于每一个批次数据中某一类病虫害特征的中心, 为了对该类病虫害进行判别, 需要该类病虫害中所有特征与特征中心的距离尽可能小。但由于 c_{y_i} 在网络训练过程中随着深层特征的改变不断更新, 因此在网络迭代中平均每一类病虫害的特征并不现实。

本研究基于 Center loss, 设计一种改进的损失函数, 此处将其命名为中心指数归一化损失函数, 具体计算见公式(17)~公式(19)。公式(17)是计算特征向量(x_i)相对于 Center loss 的梯度, 可用于说明特征向量与其类别中心的偏离方程对于损失函数的影响。其中 x_i 代表神经网络的特征向量, c_{y_i} 为特征向量所对应的类别中心。在公式(18)中, ΔC_j 表示类中心(c_j)的更新量, $\delta(y_i=j)$ 是一个指示函数, 当样本 i 的真实标签(y_i)等于类别 j 时取值为 1, 否则取值为 0。公式

(18) 基于所有属于同一类 j 的样本计算类中心(c_j)的更新量, 通过特征向量(x_i)与其类别中心(c_j)之间的差异来增大病害与病害、害虫与害虫之间的类间距, 从而使得区分 2 种相似的病虫害时准确率更高。对每类病虫害来说, 将其样本的类别中心与其类别特征中心相减后取均值, 获得不同样本类别中心的步进值。在此基础上, 类别随着网络模型训练的不断迭代, 损失值不断缩小, 相似病虫害间隔逐步增大, 使得识别结果更加准确^[31]。

$$\frac{\partial l_c}{\partial x_i} = x_i - c_{y_i} \quad (17)$$

$$\Delta C_j = \frac{\sum_{i=1}^m \delta(y_i=j) \times (c_j - x_i)}{1 + \sum_{i=1}^m \delta(y_i=j)} \quad (18)$$

$$l = l_{\text{Softmax loss}} + \lambda l_{\text{Center loss}} \quad (19)$$

1.2 试验预备

本试验的整体流程见图 6。

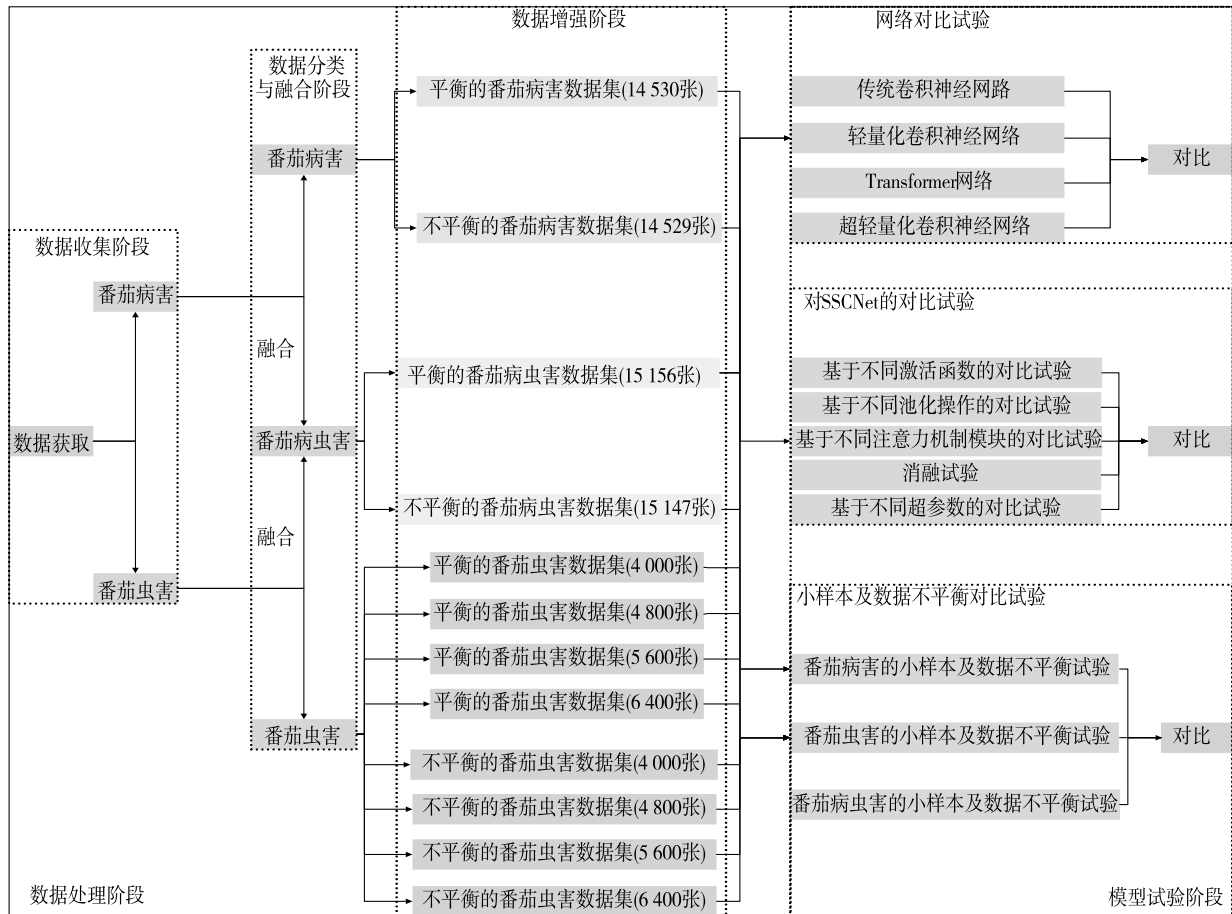
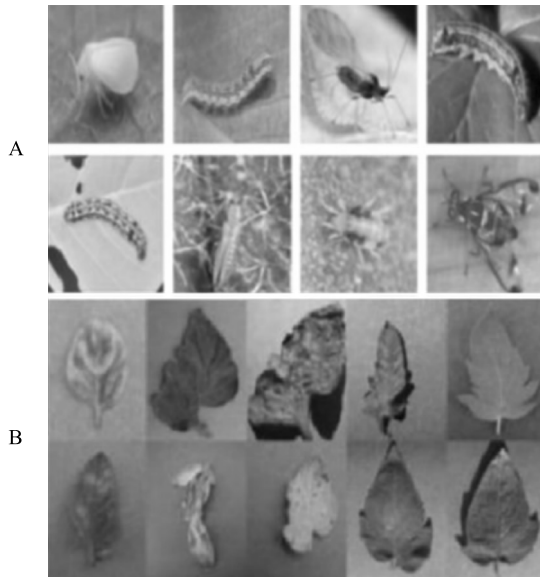


图 6 多阶段番茄叶病虫害试验流程

Fig.6 Flow chart for multi-stage tomato leaf pest and disease experiments

1.2.1 数据集描述 本研究数据包括 2 部分,分别为番茄叶害虫数据集和番茄叶病害数据集。其中害虫数据由 Huang 等^[32]自行收集(图 7A);病害数据集源于 Plant Village 数据集中的番茄叶病害数据集^[33](图 7B)。上述 2 个数据集的具体情况分别见表 1、表 2。



A:害虫;B:病害。

图 7 数据集示意

Fig.7 Schematic diagram of the data set

表 1 番茄叶害虫数据集分布

Table 1 Distribution of tomato leaf pest dataset

害虫种类	数据量(张)	占比(%)
棕榈蓟马	25	4.045
瓜实蝇	43	6.958
银叶粉虱	62	10.032
二斑叶螨	75	12.136
菜夜蛾	76	12.298
斜纹夜蛾	97	15.696
茄夜蛾	109	17.638
桃蚜	131	21.197

1.2.2 试验环境 编程语言为 Python3.6;深度学习框架为 Tensorflow2.0。图像处理单元(GPU)为 Tesla V100-PCIE-16 GB,运行内存为 12 G。

1.2.3 数据预处理 将所有图像的分辨率统一为 224×224×3,JPG 格式。所有数据集均按照 8:2 的比例分成训练集与测试集。

表 2 番茄叶病害数据集分布

Table 2 Distribution of tomato leaf disease dataset

病害种类	数据量(张)	占比(%)
花叶病	299	2.06
叶霉病	761	5.24
早疫病	800	5.51
褐斑病	1 123	7.73
无(健康叶片)	1 273	8.76
二斑叶螨病	1 341	9.23
壳针孢叶斑病	1 417	9.75
晚疫病	1 527	10.51
疮痂病	1 702	11.71
黄化曲叶病	4 286	29.50

1.2.4 数据增强 数据增强能较好解决小样本和数据不平衡的情况,避免过拟合和性能下降等问题。由表 1、表 2 可知,番茄叶害虫数据集数据量少且不平衡,番茄叶病害数据集数据量虽充足,但也存在不平衡的问题。本研究通过旋转、平移、色度变换及随机视角等方式对数据进行增强与整理,得到数据集有:

(1)4 个数据平衡的番茄害虫数据集。通过增强,将原番茄虫害数据集扩充为数据总量分别为 4 000、4 800、5 600 及 6 400 的 4 个数据集,每个数据集中各子数据集的数据量相等。

(2)4 个数据不平衡的番茄害虫数据集。通过增强,将原番茄害虫数据集按照各子数据集分布比例进行扩充,4 个数据集的数据总量分别为 4 000、4 800、5 600 及 6 400。

(3)2 个番茄病害数据集。番茄病害的原数据集即为不平衡的番茄病害数据集。通过增强,对数据量不足 1 453 的子数据集进行增强,数据量超过 1 453 的子数据集通过随机删除使其数据量减至 1 453,在原有不平衡番茄病害数据集的基础上,形成数据总量为 14 530 的数据平衡的番茄病害数据集。

(4)1 个数据不平衡的番茄病虫害数据集。将原番茄害虫数据集和原番茄病害数据集融合,得到包含 18 个类别、数据总量为 15 147 的数据集。

(5)1 个数据平衡的番茄病虫害数据集。通过增强,分别对数据量不足 842 的子数据集进行增强,对于数据量超过 842 的子数据集通过随机删除使数据量减至 842,最后形成数据量为 15 156、包含 18 个类别、各子数据集数据量均为 842 的数据集。

1.2.5 超参数设置 为保证试验规格一致,将预设

超参数设置为:学习率为0.000 1,迭代次数为 50,优化器为 Adam,批量大小为 32,训练方式为 5 折交叉验证。

1.2.6 模型训练评价标准 本研究重点关注的指标有 4 个,分别是准确率、参数量、运行时间及方差。其中准确率^[12]和运行时间作为最直观的评判指标。准确率的计算参照公式(20)。

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (20)$$

其中, TP 为真阳率; TN 为假阴率; FP 为假阳率; FN 为假阴率。

2 结果与分析

2.1 SSCNet 模型的特征提取效果

特征图是卷积神经网络中的重要输出,代表了不同层级抽象特征的结果。图 8 显示,通过引入注意力机制模块、调整池化和激活函数等,SSCNet 模型能够生成更具代表性的特征图,从而能够更好地反映输入的病虫害图像中的关键信息。

表 3 模型性能对比

Table 3 Performance comparison of models

模型类型	模型名称	准确率(%)	参数量(个)	训练时间(s)	准确率的方差(%)
传统卷积神经网络模型	VGG19	76.54	139 643 986	5 246	1.33
	VGG16	77.28	134 334 290	4 893	1.47
	ResNet101	73.63	43 674 568	6 277	1.08
	ResNet50	81.07	25 636 712	4 269	1.27
	InceptionV3	87.11	24 734 048	5 820	1.14
	Convext	80.00	87 584 914	14 588	0.81
轻量化卷积神经网络模型	SqueezeNet	65.26	1 256 432	1 576	3.42
	MobileNetv3-small	80.03	1 361 097	1 653	2.57
	MobileNetv3-large	83.87	3 130 086	1 825	1.98
	EfficientNetv2	70.90	5 942 370	2 144	1.93
	DenseNet121	86.79	7 055 954	2 747	1.33
	MnasNet	71.25	3 237 064	1 711	1.99
	ShuffleNetv2	79.77	2 268 232	1 826	1.74
	Vit	94.10	85 812 498	16 342	0.64
超轻量化卷积神经网络模型	SSCNet	86.22	404 402	1 244	1.47

从整体来看,除 Vit 模型外,其他模型的识别精度均未达到能够开展真实诊断工作的要求(但考虑到 Vit 模型的训练时间长、成本过高,并不符合本研

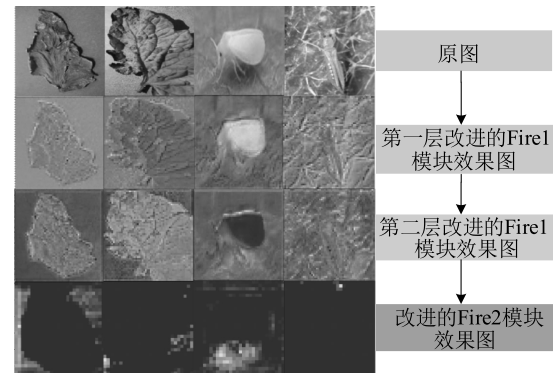


图 8 SSCNet 模型的特征提取效果

Fig.8 Feature extraction effect of SSCNet model

2.2 网络对比

为验证 SSCNet 模型的性能,本研究选取主流模型及轻量化 CNN 模型进行对比试验,试验选用 1.2.4 节中的数据不平衡的番茄病虫害数据集。表 3 显示,相较于传统 CNN 模型和常规轻量化 CNN 模型,SSCNet 在参数量和训练时间上存在较大优势,在识别准确率和方差上也存在一定优势。

究提出的超轻量化网络定义,故不进行进一步的讨论)。为了探究造成这一现象的原因,本研究从数据平衡性角度出发,讨论不同样本量及数据平衡性

对于模型性能的影响。

2.3 数据集的小样本性和数据不平衡性对模型性能的影响

2.3.1 不同虫害数据集对模型性能的影响 表 4 显示,原始数据集的数据量较少,导致难以进行有效的网络特征提取,进而出现过拟合现象,无法满足需求。在比较平衡数据集和不平衡数据集时,发现使用平衡数据集时模型准确率更高,说明数据不平衡性确实会对模型性能产生影响。

表 4 不同虫害数据集结果比对

Table 4 Comparison of results from different datasets (pests)

数据集类型	数据总量 (张)	准确率 (%)	训练时间 (s)	准确率的方差 (%)
平衡数据集	4 000	94.71	595	1.81
	4 800	95.97	660	0.92
	5 600	96.25	782	0.72
	6 400	96.17	813	0.54
不平衡数据集	4 000	86.00	602	2.04
	4 800	91.85	683	1.85
	5 600	93.42	781	1.57
	6 400	95.92	824	1.23
原始数据集	618	50.25	216	1.99

2.3.2 不同病害数据集对模型性能的影响 病害数据集数据量较大,因此不考虑进一步扩充。表 5 显示,平衡数据集与不平衡数据集间的模型性能差异较小,但是使用平衡数据集时模型整体性能更优。

2.3.3 不同病虫害数据集对模型性能的影响 表 6 显示,病害数据与害虫数据融合后,导致数据集的不平衡性进一步加剧。将数据集进行平衡处理后,模型识别准确率有了明显提升。因此,影响模型识别精度的主要因素是各子数据集之间的平衡性。

表 5 不同病害数据集结果比对

Table 5 Comparison of results from different datasets (diseases)

数据集类型	准确率 (%)	训练时间 (s)	准确率的方差 (%)
平衡数据集	95.62	1 053	0.64
不平衡数据集	95.25	1 075	0.73

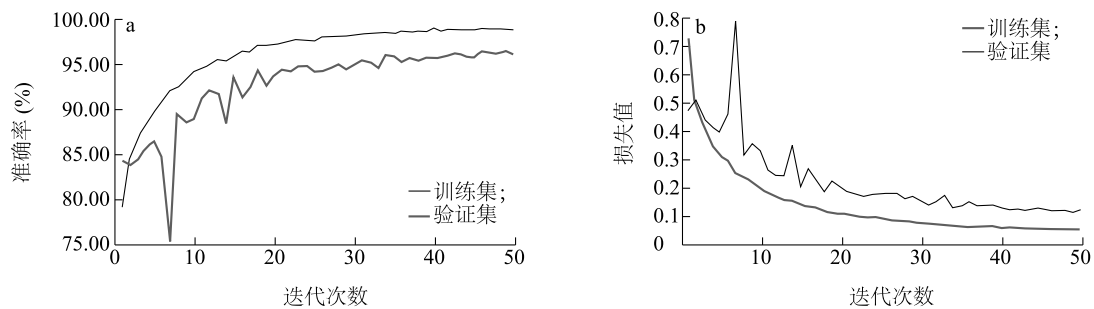
表 6 不同病虫害数据集结果比对

Table 6 Comparison of results from different datasets (diseases and pests)

数据集类型	准确率 (%)	训练时间 (s)	准确率的方差 (%)
平衡数据集	96.27	1 223	0.27
不平衡数据集	86.22	1 244	1.47

2.4 基于 SSCNet 的试验结果对比

本研究在模型主体、激活函数、池化操作及损失函数处均进行了改进。为验证改动的必要性,对比改动优势,选用数据平衡的番茄病虫害数据集对 SSCNet 模型进行试验,结果(图 9)表明,SSCNet 模型在训练过程中训练集准确率提升稳定;验证集准确率在迭代前期经历了较大波动,随着迭代次数的不断增加,变化也逐步趋于稳定。



a: 准确率; b: 损失值。

图 9 SSCNet 试验结果

Fig.9 Results of SSCNet experiment

2.4.1 基于不同激活函数的试验结果对比 表 7 显示,激活函数类型对模型性能有一定影响。其中

使用 Smish 的 SSCNet 模型识别准确率为 96.27%,其模型训练时间与 ReLU 等函数差距不大,但更加

稳定。因此综合来看,使用 Smish 的 SSCNet 模型更能满足现实中病虫害诊断的要求。

表 7 不同激活函数的试验结果对比

Table 7 Comparison of test results based on different activation functions

激活函数类型	准确率 (%)	训练时间 (s)	准确率的方差 (%)
ReLU	91.76	1 166	0.53
ELU	78.78	1 217	1.12
SELU	75.32	1 271	0.77
GeLU	90.16	1 268	0.69
Swish	91.55	1 320	0.74
Mish	93.04	1 197	0.19
Smish	96.27	1 223	0.27

2.4.2 基于不同池化操作的试验结果对比 对比 Softpool、Maxpool 和 Averagepool 对 SSCNet 模型识别性能的影响,表 8 显示,使用 Softpool 的 SSCNet 模型识别准确率达 96.27%,相比于使用其他 2 种常见池化操作更能满足番茄病虫害的诊断要求。

2.4.3 基于不同注意力模块的试验结果对比 对比不同注意力模块在 SSCNet 模型中的表现,表 9 显示,不同注意力模块对模型性能影响不同,使用

ECA 模块的 SSCNet 模型识别准确率达 96.27%,结果优于使用其他 2 种注意力模块。

表 8 不同池化层结果对比

Table 8 Comparison of results from different pooling layers

池化层类型	准确率 (%)	训练时间 (s)	准确率的方差 (%)
Maxpool	95.88	1 228	0.48
Averagepool	96.01	1 220	0.31
Softpool	96.27	1 223	0.27

表 9 不同注意力模块结果对比

Table 9 Comparison of results from different attention modules

注意力模块类型	准确率 (%)	训练时间 (s)	准确率的方差 (%)
SE	94.86	1 289	0.65
CBAM	95.73	1 241	0.29
ECA	96.27	1 223	0.27

2.4.4 消融试验 为了验证对 SSCNet 模型中各个结构改动的可靠性,本研究进行消融试验。表 10 显示,尽管去掉一些模块后模型效率更高,但模型整体性能也会不同程度地下降。综合来看,SSCNet 模型表现更优。

表 10 消融试验结果

Table 10 Results of ablation experiment

模型类型	改进的 Fire1	改进的 Fire2	ECA	Softpool	Smish	CenterSoftmax	训练时间 (s)	准确率 (%)	准确率的方差 (%)
1		✓	✓	✓	✓	✓	1 367	94.76	0.84
2	✓		✓	✓	✓	✓	1 409	90.51	0.83
3	✓	✓		✓	✓	✓	1 207	82.43	2.41
4	✓	✓	✓		✓	✓	1 217	96.01	1.13
5	✓	✓	✓	✓		✓	1 166	91.76	1.27
6	✓	✓	✓	✓	✓		1 189	94.27	0.83
7	✓	✓			✓	✓	1 213	83.79	3.35
8	✓	✓	✓			✓	1 132	90.94	2.06
9	✓	✓	✓	✓			1 141	91.17	3.41
10	✓	✓				✓	1 119	94.37	2.02
11	✓	✓	✓				1 087	93.55	1.77
SSCNet	✓	✓	✓	✓	✓	✓	1 223	96.27	0.27

2.4.5 不同超参数对比 除了数据量及网络结构之外,超参数也是影响模型性能的指标之一。为了

探究超参数对模型性能的影响,本研究对比了不同优化器及学习率(优化器为 Adam、SGD 和 RMSProp;学习率设置为 0.010 0、0.001 0 及 0.000 1)下的结果。表 11 显示,当优化器与学习率的组合为 Adam+0.001、RMSProp+0.001、Adam+0.001 时,SSCNet 模型的表现分别在害虫、病害及病虫害诊断任务中最好,准确率分别为 98.83%、98.14% 和 97.71%,能较好地完成任务。

表 11 基于不同超参数的 SSCNet 模型训练准确率

Table 11 Accuracy of SSCNet model under different hyper-parameters

数据集类型	优化器	准确率(%)		
		学习率 0.010 0	学习率 0.001 0	学习率 0.000 1
虫害	Adam	6.04	98.83	96.17
	SGD	91.44	53.78	37.39
	RMSProp	93.70	98.65	95.86
病害	Adam	11.70	96.13	95.62
	SGD	86.44	81.03	59.63
	RMSProp	95.49	98.14	95.77
病虫害	Adam	1.67	97.71	96.27
	SGD	93.97	61.92	38.83
	RMSProp	93.11	96.39	94.22

虫害数据集:数据总量为6 400的平衡数据集;病害数据集:数据总量为14 530的平衡数据集;病虫害数据集:数据总量为15 156的平衡数据集。

3 讨论

针对当前卷积神经网络在进行番茄病虫害诊断时因结构过于复杂导致难以应用于手持终端设备,以及现有轻量化网络特征提取能力弱、识别准确率低、难以满足实际需求等问题^[34-37],本研究在轻量化卷积神经网络的基础上,定义了超轻量化卷积神经网络,并依照该定义设计了一种超轻量化卷积神经网络 SSCNet,比常规轻量化卷积神经网络效率更高、性能更强。此外,通过讨论数据小样本性、不平衡性以及模型内不同结构对模型性能的影响,可得出以下结论:第一,由 2.3 节中 3 组试验结果可知,数据不平衡对模型性能影响较大;第二,由 2.4 节中 5 组试验结果可知,模型中各个结构的细微变化均会对模型识别性能产生较大影响。因此在后续工作中为了保证模型在番茄病虫害诊断任务中的准确率和稳定性,应注意以下 2 个方面:第一,要避免子数据集之间的数据量差距过大、出现不平衡现象;第

二,在模型训练过程中,针对特定识别任务对于模型结构、超参数进行适当调整,从而获得最优效果。

参考文献:

- [1] PIERCE F J, NOWAK P. Aspects of precision agriculture[J]. *Advances in Agronomy*, 1999, 67: 1-85.
- [2] AGARWAL M, GUPTA S K, BISWAS K K. Development of efficient CNN model for tomato crop disease identification[J]. *Sustainable Computing: Informatics and Systems*, 2020, 28: 100407.
- [3] REN F, LIU W, WU G. Feature reuse residual networks for insect pest recognition[J]. *IEEE Access*, 2019, 7: 122758-122768.
- [4] 彭红星,何慧君,高宗梅,等. 基于改进 ShuffleNetV2 模型的荔枝病虫害识别方法[J]. *农业机械学报*, 2022, 53(12): 290-300.
- [5] 曹乐平. 基于机器视觉的植物病虫害实时识别方法[J]. *中国农学通报*, 2015, 31(20): 244-249.
- [6] ALVES A N, SOUZA W S R, BORGES D L. Cotton pests classification in field-based images using deep residual networks[J]. *Computers and Electronics in Agriculture*, 2020, 174: 105488.
- [7] BOLLIS E, MAIA H, PEDRINI H, et al. Weakly supervised attention-based models using activation maps for citrus mite and insect pest classification[J]. *Computers and Electronics in Agriculture*, 2022, 195: 106839.
- [8] HUANG M L, CHUANG T C, LIAO Y C. Application of transfer learning and image augmentation technology for tomato pest identification[J]. *Sustainable Computing: Informatics and Systems*, 2022, 33: 100646.
- [9] 周维,牛永真,王亚伟,等. 基于改进的 YOLOv4-GhostNet 水稻病虫害识别方法[J]. *江苏农业学报*, 2022, 38(3): 685-695.
- [10] 温长吉,王启锐,陈洪锐,等. 面向大规模多类别的病虫害识别模型[J]. *农业工程学报*, 2022, 38(8): 169-177.
- [11] 罗昱晟,冉力争,张小荣,等. 基于大数据与图像识别的柑橘病虫害防治系统的设计与实现[J]. *智能计算机与应用*, 2020, 10(12): 170-173.
- [12] 骆润玖,殷惠莉,刘伟康,等. 基于 YOLOv5-C 的广佛手病虫害识别[J]. *华南农业大学学报*, 2023, 44(1): 151-160.
- [13] 杨英茹,吴华瑞,张燕,等. 基于复杂环境的番茄叶部图像病虫害识别[J]. *中国农机化学报*, 2021, 42(9): 177-186.
- [14] IANDOLA F N, HAN S, MOSKEWICZ M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size[J]. *arXiv*, 2016. DOI:10.48550/arXiv.1602.07360.
- [15] ERTAM F, YAKUT O F, TUNCER T. Pattern lock screen detection method based on lightweight deep feature extraction[J]. *Neural Computing and Applications*, 2023, 35(2): 1549-1567.
- [16] ZHONG H, LYU Y, YUAN R, et al. Bearing fault diagnosis using transfer learning and self-attention ensemble lightweight convolutional neural network[J]. *Neurocomputing*, 2022, 501: 765-777.
- [17] ZHONG Y, HAO J, LIU Q, et al. Novel diagnosis method for GIS mechanical defects based on an improved lightweight CNN model

- with load adaptive matching[J]. IEEE Transactions on Industrial Informatics, 2023, 19(11): 11041-11051.
- [18] MINU M, CANESSANE R, RAMESH S. Optimal squeeze net with deep neural network-based arial image classification model in unmanned aerial vehicles[J]. Traitement Du Signal, 2022, 39(1): 275-281.
- [19] PANG Y, ZHANG Y, WANG Y, et al. SOCNet: a lightweight and fine-grained object recognition network for satellite on-orbit computing[J]. IEEE Transactions on Geoscience and Remote Sensing, 2022, 60. DOI: 10.1109/TGRS.2022.3216215.
- [20] LI Y, NIE J, CHAO X. Do we really need deep CNN for plant diseases identification? [J]. Computers and Electronics in Agriculture, 2020, 178: 105803.
- [21] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]. Salt Lake City, UT, USA: IEEE, 2018.
- [22] WANG Q, WU B, ZHU P, et al. ECA-Net: efficient channel attention for deep convolutional neural networks[C]. Seattle, WA, USA: IEEE, 2020.
- [23] WOO S, PARK J, LEE J Y, et al. CBAM: convolutional block attention module[C]. Munich, Germany: Springer, 2018.
- [24] KLAMBAUER G, UNTERTHINER T, MAYR A, et al. Self-normalizing neural networks[J]. 2017. DOI: 10.48550/arXiv.1706.02515.
- [25] HENDRYCKS D, GIMPEL K. Gaussian error linear units (gelus) [J]. arXiv, 2016. DOI: 10.48550/arXiv.1606.08415.
- [26] RAMACHANDRAM P, ZOPH B, LE Q V. Searching for activation functions[J]. arXiv, 2017. DOI: 10.48550/arXiv.1710.05941.
- [27] MISRA D. Mish: a self regularized non-monotonic neural activation function[J]. arXiv, 2019. DOI: 10.48550/arXiv.1908.08681.
- [28] STERGIOU A, POPPE R, KALLIATAKIS G. Refining activation downsampling with SoftPool[C]. Montreal, Canada: IEEE, 2021.
- [29] ZEILER M D, FERGUS R. Visualizing and understanding convolutional networks[C]. Zurich, Switzerland: Springer, 2014.
- [30] 李超, 李锋, 黄炜嘉. 基于并联卷积神经网络的水果品种识别[J]. 浙江农业学报, 2022, 34(11): 2533-2541.
- [31] WEN Y, ZHANG K, LI Z, et al. A discriminative feature learning approach for deep face recognition[C]. Amsterdam, Netherland: Springer, 2016.
- [32] HUANG M, CHIN T. A database of eight common tomato pest images[DB]. 2020. DOI: 10.17632/s62zm6djd2.1.
- [33] HUGHES D, SALATH' E M. An open access repository of images on plant health to enable the development of mobile disease diagnostics[J]. arXiv, 2015. DOI: 10.48550/arXiv.1511.08060.
- [34] 蒋清健, 姚勇, 付志军, 等. 基于改进卷积神经网络算法的番茄叶片病害识别[J]. 江苏农业科学, 2022, 50(20): 29-34.
- [35] 徐姗姗, 吕净妍, 陈芳媛. 深度卷积神经网络的遥感植被检测方法[J]. 南京林业大学学报(自然科学版), 2022, 46(4): 185-193.
- [36] 牛学德, 高丙朋, 南新元, 等. 基于改进 DenseNet 卷积神经网络的番茄叶片病害检测[J]. 江苏农业学报, 2022, 38(1): 129-134.
- [37] 棘玉, 尹显明, 严恩萍, 等. 基于相机拍照的油茶果形状特征提取研究[J]. 南京林业大学学报(自然科学版), 2022, 46(2): 63-70.

(责任编辑:王妮)