基于 STM32 的农业物联网病虫害图像识别算法研究

许柏涛，陈翔
（中山大学电子与信息工程学院，广东 广州 510006）

摘　要：在现代农业物联网系统中，边缘计算是不可或缺的组成部分。在此背景下，可将轻量级病虫害图像识别任务置于边缘设备上，然而受限于设备计算和存储能力，该任务面临着不小的挑战。为了解决这些问题，提出了一种以经济实用的 STM32 为边缘设备进行病虫害图像识别的方法。该方法针对 STM32 的特点，基于 MobileNetv2 结构做出改进，并应用量化感知训练技术对神经网络模型进行压缩，提高了模型的可移植性。同时，模型使用 X-CUBE-AI 部署并进行了性能评估。实验结果表明，改进模型不仅保证了图像分类准确率，而且相较于其他轻量级神经网络，该模型对 STM32 的 Flash 与 RAM 资源的占用有所减小。

关键词：农业物联网；边缘计算；病虫害识别；STM32
中图分类号：TP389.1
文献标志码：A
doi: 10.11959/j.issn.2096−3750.2023.00365

Research on agricultural IoT pest and disease image recognition algorithm based on STM32

XU Botao, CHEN Xiang
School of Electronic and Information Engineering, Sun Yat-sen University, Guangzhou 510006, China

Abstract: In modern agriculture IoT systems, edge computing is an indispensable component. In this context, it is feasible to deploy lightweight pest and disease image recognition tasks on edge devices. However, due to the constraints of device computation and storage capabilities, this task faces significant challenges. To address these challenges, an economically practical method was proposed for pest and disease image recognition on STM32 edge devices. Specifically, the MobileNetv2 structure was improved to better suit the characteristics of STM32, quantization-aware training technique was used to compresses the network, model portability was enhanced. Meanwhile, the X-CUBE-AI was used to arrange the model and evaluate the performance. Experimental results demonstrate that the proposed model not only ensures image classification accuracy but also reduces the Flash and RAM resource consumption on STM32 compared to other lightweight networks.

Key words: agricultural IoT, edge computing, pest and disease recognition, STM32

0　引言

随着技术的不断进步，农业物联网的应用已经成为现代农业的重要组成部分[1-2]。通过传感器技术、物联网技术以及云计算技术的结合，农业生产的各个环节都可以实现自动化、信息化、智能化的管理，从而提高农业生产效率和质量[3]。

在农业物联网中，边缘计算作为一种新兴的计算模式，正在被越来越多的人所关注和应用[4-6]。相较于传统的云计算模式[7-9]，边缘计算将计算资源和

收稿日期：2023−04−14 ；**修回日期：**2023−07−27
通信作者：陈翔，chenxiang@mail.sysu.edu.cn
基金项目：广东省现代农业产业技术创新团队专项基金资助项目（No.2023KJ122）
Foundation Item: The Guangdong Provincial Special Fund for Modern Agriculture Industry Technology Innovation Teams (No.2023KJ122)

数据处理能力下放到网络边缘的设备，如传感器、路由器、交换机、物联网网关等。该策略使得数据处理更加及时、高效，减少了网络时延和带宽消耗，并且可以更好地保护数据的隐私和安全。因此，在农业物联网中，利用边缘设备进行数据处理和人工智能应用能够提供更高的实时性和灵活性，更适用于复杂和动态的环境下的应用场景。此外，边缘设备通常具备经济实惠的特点，能够使更多农民受益。

在农业生产当中，农业病虫害是制约农业生产的一个重要因素，传统的人工方法对农业病虫害的监测和诊断存在效率低下、准确率不高[10-11]的问题。因此，结合农业物联网和边缘计算技术，建立基于人工智能的农作物病虫害识别系统，对农业病虫害的智能监测和预警具有重要意义。在该系统中，利用边缘设备的计算和存储能力，实现数据的实时采集、预处理和分析，进一步优化识别算法的性能，提高农业病虫害的识别准确率和效率，为农业生产提供更好的支持[12]。

随着微控制器算力的提升，STM32 开始被用作边缘设备[13]。作为一款低功耗、高性能且经济实惠的处理器，STM32 有助于在边缘设备上进行人工智能计算，从而在农业生产中提高效率和降低成本。首先，STM32 是一款性能稳定、功耗低、经济的嵌入式芯片，特别适合在边缘设备中使用。其次，STM32 具备较强的扩展性和兼容性，可以与多种传感器、执行器、通信模块等硬件设备连接[14]。此外，STM32 还拥有完善的软件生态环境，支持多种编程语言和开发环境，能够满足多种开发需求。因此，选择 STM32 作为边缘设备是合理且可行的选择。

然而，将 STM32 作为病虫害图像识别边缘节点仍面临不少困难。首先，类似的微控制单元（MCU, microcontroller unit）由于计算能力和存储容量的有限性，尤其是存储容量的限制，可能难以支持较复杂的神经网络模型的运行和部署。这间接导致当前的软件堆栈在 MCU 上部署深度学习模型方面缺乏充分的软件支持。其次，农业场景的复杂性增加了病虫害识别的难度。例如，光照、阴影、天气等因素可能影响图像质量，从而影响模型的准确率。

针对上述神经网络在 STM32 上的部署难点，本文以轻量级神经网络 MobileNetv2[15]为基础，进行了一系列针对性优化，并提出了新的模型。首先，针对 STM32 的存储限制，在保持精度的前提下，对神经网络结构进行了优化和裁剪，改进并替换了部分结构，以达到降低存储需求的目的。其次，采用量化技术对改进模型进行了压缩，将神经网络的浮点参数转化为整型参数，从而减少了神经网络的存储空间，并降低计算复杂度。最后使用意法半导体发布的 X-CUBE-AI 软件实现神经网络的部署[16]。通过这些优化措施，能够在 STM32 上成功部署小分类病虫害图像识别模型，克服了存储限制和计算复杂度的挑战。

1　基于 MobileNetv2 的结构改进

1.1　MobileNetv2 结构

MobileNetv2 是一种由 Google 于 2018 年提出的轻量级深度神经网络结构，旨在保持高准确率的同时减少计算量和参数量，以适应移动设备和嵌入式系统等资源受限的环境。

MobileNetv2 的核心构建包括残差块和轻量级深度可分离卷积。整个神经网络结构由一系列的块组成，每个块由深度卷积和点卷积组合而成，分别用于卷积处理和降维、升维处理。MobileNetv2 可以有效地减少计算量和参数数量，同时保持较高的准确率。

在 MobileNetv2 中还引入了一种叫作倒置残差块（inverted residual block）的结构，也被称为瓶颈结构（bottleneck），用于解决轻量级模型中的信息瓶颈问题。该结构先进行点卷积升维，然后进行深度卷积，最后进行点卷积降维输出。倒置残差块结构可以有效地增强模型的表达能力，同时保持较少的计算量和参数数量。倒置残差块结构见表 1，输入通道 k 首先经过点卷积升维到 tk，然后经过深度卷积，最后经过点卷积降维输出为 k'。

表 1　　倒置残差块结构

输入	运行	输出
$h\times w\times k$	1×1 Conv2d, ReLU6	$h\times w\times tk$
$h\times w\times tk$	3×3 dwise, ReLU6	$\frac{h}{s}\times\frac{w}{s}\times tk$
$\frac{h}{s}\times\frac{w}{s}\times tk$	1×1 Conv2d	$\frac{h}{s}\times\frac{w}{s}\times k'$

总体而言，MobileNetv2 具有简洁且轻量的结构，因此非常适合在移动设备和嵌入式系统等资源受限的环境中执行图像识别和物体检测等任务。其设计的关键在于通过深度可分离卷积和倒置残差结构，实现高效的特征提取和信息处理，同时最大限度地减

少计算量和参数数量。这使得 MobileNetv2 能够在资源有限的情况下，仍保持较高的准确率和性能表现。

1.2 结构改进与模型设计

与服务器或其他内存更充足的嵌入式设备相比，将神经网络模型部署在 STM32 这种内存资源匮乏的设备上会面临更多挑战。除了模型大小，运行神经网络模型时的内存峰值大小也需要考虑，即 Flash 和 RAM 的占用情况。特别是涉及运行时 RAM 的情况，相较于 MobileNetv2，其他轻量级神经网络如 MobileNet[17] 和 FD_MobileNet[18] 更适合在 STM32 上运行，因为它们不使用残差连接[19]。在神经网络推理过程中，残差连接在输入和输出处产生中间值，这些中间值需要暂存在 RAM 空间中等待使用。因此，针对 STM32 的优化，需要尽量减少残差连接的使用。然而，必要的残差连接仍然是不可或缺的，因为它们可以在一定程度上减少神经元的死亡，有助于解决梯度消失的问题，避免信息的损失，保持模型的精度，并使神经网络在训练过程中更容易收敛，从而提高神经网络训练的鲁棒性。

同时，仍需考虑 Flash 资源。在目前的轻量级神经网络中，深度可分离卷积和 MobileNetv2 的瓶颈结构被广泛使用，它们使用大量的 1×1 点卷积，导致神经网络参数仍然较多，不利于在空间有限的 STM32 上部署。

为了缓解上述问题，本文提出了改进方案，旨在减少模型参数与模型中间值的数量，从而降低内存资源占用。

1.2.1 减少残差连接

在 MobileNetv2 模型中，除了最初的常规卷积和卷积步长为 2 的下采样部分，其余部分都采用残差连接结构。然而，在内存资源受限的 STM32 上，这种设计可能对 RAM 资源造成巨大压力。为了减少 RAM 占用，可以通过减少残差连接的方式进行优化。具体而言，在特征提取过程中，由于下采样操作导致图像分辨率降低，信息可能会有所丢失。为了最小化下采样的信息损失所引起的对精度的影响，仅在可能导致精度损失的下采样操作之后的卷积层级中应用残差连接。这样可以有效减少中间值的内存开销，并避免过度使用残差连接对 RAM 资源造成过大的压力，同时维持精度。

优化策略的理论基础可以从以下角度进行更严谨的分析和说明。

在深度神经网络中，下采样操作导致特征图分辨率降低，可能会引起信息的丢失。残差连接的作用是通过跳跃连接将输入和输出之间的特征进行融合，以便更好地传递梯度和保持信息的流动性。然而，在下采样之前的卷积层级中，由于特征图的分辨率相对较高且信息未经过显著的损失，残差连接的引入可能不会对神经网络的学习效果产生显著影响。

相反，在下采样之后的卷积层级中，特征图的分辨率降低，此时神经网络需要更多地关注重要的高频细节信息，并且有可能发生信息的瓶颈现象。在这种情况下，使用残差连接可以帮助恢复和利用特征图中的重要细节信息，以增强神经网络的表达能力，保持模型的准确率。

因此，优化策略选择仅在可能影响精度的下采样操作之后使用残差连接的原因是，它充分弥补了下采样之后特征图分辨率降低的缺陷。通过在下采样后的卷积层级中引入残差连接，可以增加神经网络对重要细节信息的关注，使得学习效果至少不比原来差。在同时考虑内存限制和模型准确率的前提下，这种优化策略可以提高内存资源的利用效率，并确保神经网络性能的稳定和准确率。

1.2.2 优化 1×1 卷积比例

借鉴文献[20]的思想，本文提出一种优化策略，使用少量的参数生成一组相似的特征图。文献[20]发现，在传统卷积过程中，很多特征图具有相似的特征表示。相似的特征图之间可以通过线性变换的方式互相生成。因此，可以采用更经济的线性变换方法代替卷积操作，从而提取特征图。具体而言，对于给定的一幅特征图，通过应用线性变换，可以得到一组相似但具有不同特征的特征图。接着，将这些生成的特征图与原始特征图进行融合操作，以综合利用它们所包含的不同信息。在使用简单的线性变换获得数量相当的特征图的同时参数也随之下降，这意味着相同参数量情况下，能获得更多的信息。

本文使用了深度可分离卷积生成与原输出通道一半数量的特征图。接着，使用 3×3 深度卷积作为线性变换，对前面生成的特征图进行处理，生成另一半特征图。最后，将传统卷积的特征图和线性变换生成的特征图进行特征融合。本文将这一模块称为“STM32-block”，其结构如图 1 所示，并将其作为代替 MobileNetv2 中无下采样与残差连接的 bottleneck 块的一种解决方案。

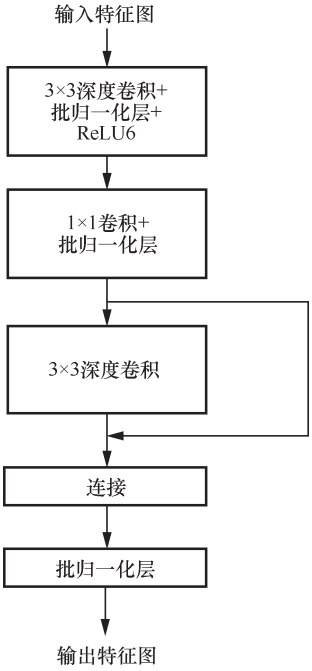


图 1　STM32-block 结构

这种优化策略的核心思想是引入线性变换和特征融合，以较少的参数生成一组相似的特征图，降低 1×1 卷积比例。这样可以在相同的参数量下获得更多的信息，适应内存资源受限的 STM32 的部署需求，并保持模型的性能稳定和准确率。

同时，在下采样的 bottleneck 块中，本文采用不同的扩张因子平衡速度和内存的消耗。具体而言，在下采样阶段，采用较小的扩张因子（如 4 倍），以微弱的精度损失，换取更快的速度和更小的内存占用。而在残差连接部分，仍然使用更大的扩张因子（如 6 倍），以确保具有足够的感受野。

此外，相较于 MobileNetv2 原有结构，本文减少了一次下采样操作，以保持输入分辨率并获取更多的空间信息。换言之，如果将两个下采样之间的层视为一个子网络，则该子网络必须具有足够的感受野来编码空间信息，而保持输入分辨率可以保留更多的空间信息，为下一次下采样提供更好的信息基础。

STM32-MobileNet 结构见表 2。每一行描述一个由 1 个或多个相同（模步幅）层组成的序列。其中，t 表示扩张因子，在下采样阶段被固定为 4；c 表示输出通道数，即每个序列中的所有层具有相同的输出通道数；n 表示操作的重复次数；s 表示步长为 2 的下采样操作，每个序列的第一个层具有步长 s，而其他层的步长为 1。

表 2　STM32-MobileNet 结构

输入	运行	t	c	n	s
$160^2\times3$	Conv2d	–	32	1	2
$80^2\times32$	STM32-block	–	16	1	1
$80^2\times16$	bottleneck	6	24	2	2
$40^2\times24$	STM32-block	–	32	2	1
$40^2\times32$	bottleneck	6	64	2	2
$20^2\times64$	STM32-block	–	96	3	1
$20^2\times96$	bottleneck	6	160	2	2
$10^2\times160$	STM32-block	–	320	1	1
$10^2\times320$	Avgpool	–	–	1	–
1×1×320	Conv2d 1×1	–	1 280	1	1
1×1×1 280	Conv2d 1×1	–	k		

通过以上优化措施，神经网络结构能够在减小内存消耗的同时平衡速度和精度。使用不同的扩张因子以及保持分辨率的设计策略，提升模型的感受野和空间信息的编码能力，同时适应内存资源受限的 STM32，并保持模型的性能稳定和准确率。

1.2.3　模型量化

要在 STM32 上部署深度学习模型，仅减小模型的大小是不够的，即使使用轻量级神经网络，其资源需求仍然超出 STM32 的限制。因此，为了在 STM32 上有效部署模型，需要采用更高效的方法进一步减小模型的大小并降低计算需求。其中，量化技术[21]是一种能够有效减小模型大小的技术。

在过去，训练后量化[22]是一种常见的压缩神经网络模型的方法。然而，训练后量化通常需要对已训练好的模型进行重新量化，文献[23]指出训练后量化方式虽然在大模型上效果较好（如 ResNet101），但是在小模型上会导致显著的准确率下降（如 MobileNet），因为小模型不同通道的输出范围相差可能会非常大，但训练后量化要求同一层的所有通道需要量化为相同的分辨率，这导致范围较小的通道的权值相对误差要高得多，导致精度的损失。

为了避免上述问题，本文采用 TensorFlow Lite（简称 TFLite）提供的量化感知训练技术[23]压缩神经网络模型。

在理解量化技术之前，首先需要说明量化误差的来源。量化的核心思想就是将浮点数区间的参数映射到 int8 的离散区间中，计算式为

$$r = S(q - Z) \tag{1}$$

其中，r 为 float32 的浮点数，q 为 int8 的量化值，S、Z 分别为缩放因子和零点。

S 的计算式为

$$S=\frac{r_{\max}-r_{\min}}{q_{\max}-q_{\min}} \tag{2}$$

其中，$r_{\max}$、$r_{\min}$ 分别表示 float32 浮点数的最大值和最小值，$q_{\max}$、$q_{\min}$ 分别表示量化后的 int8 的最大值和最小值。

在 int8 量化中，假设将原始值 r 映射至[−128,127]整数区间。则此时 $q_{\max}$=127，$q_{\min}$=−128，Z 表示 float32 浮点数 r=0 对应的量化值。

因此，模型量化的精度损失主要是量化过程中丢失了一些浮点数的精度信息所引起的误差。为了应对这一问题，提出了量化感知训练方法。在训练过程中，可以插入伪量化节点模拟量化误差。伪量化过程如图 2 所示，通过在卷积操作前后插入伪量化节点，将数据从浮点数量化为整数再反量化为浮点数输入，并在反向传播时获取量化误差。在梯度更新时，将标签误差和量化误差一同考虑，以减小量化误差，保持量化模型与浮点模型精度。这就是量化感知训练的原理。该方法不仅可以提高低精度模型的精度，还可以使低精度模型与高精度模型之间的误差保持在一定范围内，从而使得低精度模型在实际应用中更加有效。

相比传统的量化方法，TFLite 的量化感知训练有以下几个优点。

- 更高的精度：量化感知训练可以将低精度模型的精度提高到与浮点模型相似的水平。而传统的量化方法，特别是在小模型上，往往会降低模型的精度。
- 更好的可移植性：TFLite 的量化感知训练可以在高精度模型和低精度模型之间实现平滑转换，使得模型在不同设备上的性能更加一致。
- 节省更多的内存和计算资源：TFLite 的量化感知训练可以大大降低模型的大小和计算需求，从而在资源受限的设备上实现更高效的运行。

2 实验结构与分析

2.1 实验配置与训练方法

本文采用 TensorFlow 作为深度学习框架，TensorFlow 被广泛应用于图像分类、音频处理、推荐系统和自然语言处理等领域。TensorFlow 具有强大的计算平台，支持 Python 和 C++编程语言，并在 CPU 和 GPU 上高效运行。在本文中，TensorFlow 为模型的训练和推理提供了可靠的计算支持。

为了在边缘设备上实现高效的神经网络推理，并且减少量化过程中的精度损失，本文采用了一种更准确的训练方法，即量化感知训练。通过 TensorFlow 提供的 API，本文实现了量化感知训练。具体而言，在实现过程中，本文利用了 TensorFlow 的量化感知训练优化工具和接口，在需要进行量化的层中插入了伪量化节点，并在不支持的操作中配置了自定义的量化形式。在训练过程中，在前向传播中使用浮点数进行计算，而在反向传播中使用量化后的数值进行计算，通过计算得到的梯度来更新神经网络的权重。训练完成后，需要将模型转换为 TFLite 模型，此时浮点模型正式量化为 int8 模型。转换过程使用 TensorFlow Lite Converter 将模型转换为 TFLite 格式。

模型训练的超参数设置批次大小为 32，迭代次数为 500，优化器为 Adam，学习率设置为 0.001，随机失活（dropout）为 0.3。

2.2 实验数据集

本文使用了两组数据集进行测试。第一组是包含 5 类苹果叶子病害数据的数据集[24]，部分苹果叶子病害图像如图 3 所示。在使用该数据集之

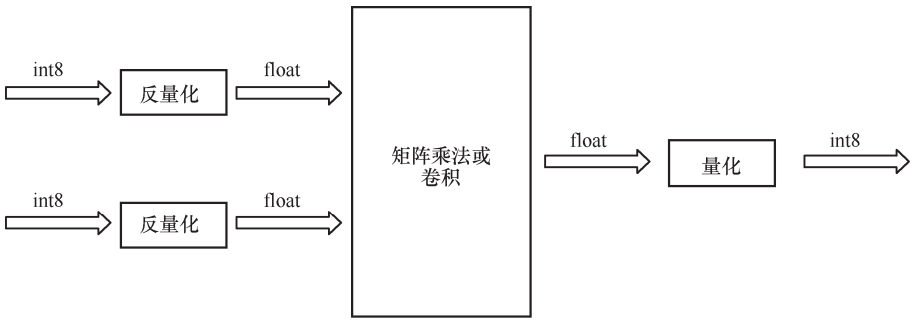


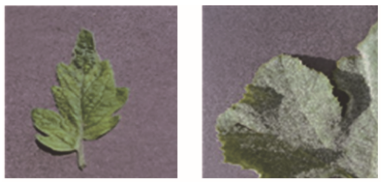
图 2 伪量化过程

前，原作者在文献[24]中说明已对该数据集进行了预处理。预处理过程包括图像旋转、水平和垂直镜像、锐度、亮度、对比度调整以及高斯模糊等 11 项操作，以扩充数据集。最终得到了共计 24 348 张有效图像。

（a）斑点落叶病　（b）褐斑病　（c）锈病

图 3　部分苹果叶子病害图像

第二组数据集是包含 7 类番茄叶子病害数据的数据集。该数据集没有经过额外的扩增处理，部分番茄叶子病害图像如图 4 所示。



（a）番茄花叶病毒　（b）白粉病

图 4　部分番茄叶子病害图像

两组数据集数量的划分比例均为训练集 60%、验证集 20%、测试集 20%，并按此比例随机划分。五分类苹果病害数据集见表 3，七分类番茄病害数据集见表 4。

表 3　五分类苹果病害数据集

类别	数量/张
Alternaria_Boltch	5 343
Brown_Spot	5 655
Grey_Spot	4 810
Mosaic	4 875
Rust	5 694

表 4　七分类番茄病害数据集

类别	数量/张
Early_Blight_Fungus	792
Healthy	1 381
Late_Blight_Water_Mold	1 513
Leaf_Mold_Fungus	755
Powdery_Mildew	1 469
Septoria_Leaf_Spot_Fungus	1 403
Spider_Mite_Damage	929

基于病虫害数据集的相对稀缺性，本文选择了两组类别数较小、数据量较大的数据集作为验证代表。这样的选择出于对模型的全面测试和评估的考虑。虽然这些数据集与实际病虫害数据集之间存在一定的差异，但它们能够提供充足的样本数量和多样性，以验证所提出的模型在 STM32 上执行小类别病害分类任务的性能。

为了提高模型的鲁棒性，应对不同图像质量和场景，以及数据泄露问题，在训练过程中引入了随机色调变换、随机模糊和随机裁剪等数据增强技术。

随机色调变换随机调整输入图像的颜色属性，如亮度、对比度和饱和度等，以模拟实际应用中可能遇到的不同光照条件和色彩变化。这样的处理有助于提高模型的鲁棒性，能够在不同环境下有效地处理图像。此外，随机模糊技术可模拟图像在拍摄或传输过程中可能产生的模糊效果，使模型能够更好地应对图像模糊的情况。而随机裁剪技术则通过对输入图像进行随机区域的裁剪，进一步提高了训练数据的变化性，以增强模型对图像不同部分的识别能力。

通过引入这些数据增强技术，模型在训练过程中能够更全面地学习和适应不同图像变化和噪声环境的特征，从而提高模型的鲁棒性和泛化能力。这种数据增强策略有助于减少过拟合现象，并使模型能够更好地适应实际应用中的图像输入。在实际应用中，根据具体场景和数据集特点，可能需要对数据增强策略进行定制化调整，以进一步提升模型在特定环境下的性能和适应性。

2.3　部署平台与评价指标

为了在 STM32 等嵌入式设备上实现神经网络，本文采用 X-CUBE-AI 工具进行神经网络部署。X-CUBE-AI 是一个专为 STM32 设计的人工智能开发平台，提供一套完整的神经网络部署和优化方案[25]，可以帮助开发人员快速高效地在 STM32 上部署和优化神经网络[26-27]。

同时，X-CUBE-AI 工具提供了硬件平台仿真功能，开发人员可以使用该工具对在 STM32 上部署的 C 语言模型进行测试和验证[28-29]。通过该仿真功能，可以评估模型在 STM32 上的适用性，包括模型的内存使用情况、计算速度等。使用 X-CUBE-AI 工具进行模型验证能够全面评估模型的准确率、可靠性和适用性，并为 STM32 上的人工智能开发提供重要的参考依据。

本文在测试过程中使用 X-CUBE-AI 工具对训练并转换为 TFLite 格式的模型进行转换和分析，并观察资源占用和验证精度[30]。在个人计算机上，使用 X-CUBE-AI 工具进行分析，其中的“analyze”功能可以获取转换后的 C 语言模型的相关信息，其中包括模型的复杂度，即乘法和累加操作、Flash 占用和 RAM 占用等，analyze 分析模型资源占用如图 5 所示。

```
Complexity: -
Used Flash: - (.00 B over 2.00 MiB Internal)
Used Ram: - (.00 B over 1.03 MiB Internal)
Achieved compression: -
Analysis status: -
```

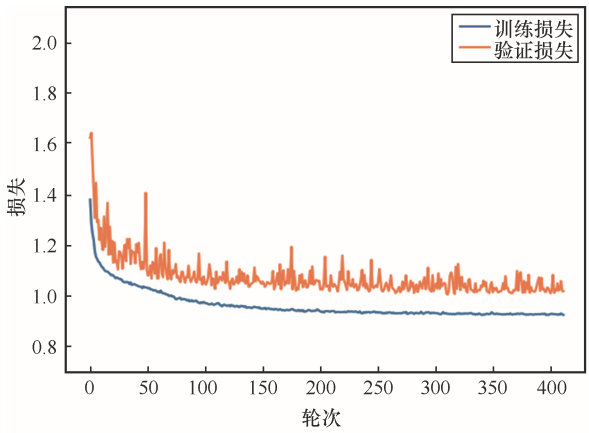
图 5　analyze 分析模型资源占用

为了验证模型分类准确率，需要将验证数据集转换为 one-hot 格式的标签，并将图像数据与相应的标签以 npy 格式存储。将这些数据导入工具，并利用工具提供的“validation on desktop”功能，在个人计算机上对模型进行验证。通过这个验证功能，比较原始的 Python 模型和转换后的 C 语言模型的性能差异，并查看验证集在模型上的测试结果。这种验证方式能够提供有关模型在 STM32 上真实运行情况的参考，并且能够方便地在桌面端评估模型的适用性和性能。通过验证模型的准确率和可靠性，并评估模型在嵌入式系统中的适用性，能够更好地了解模型在实际应用中的表现。

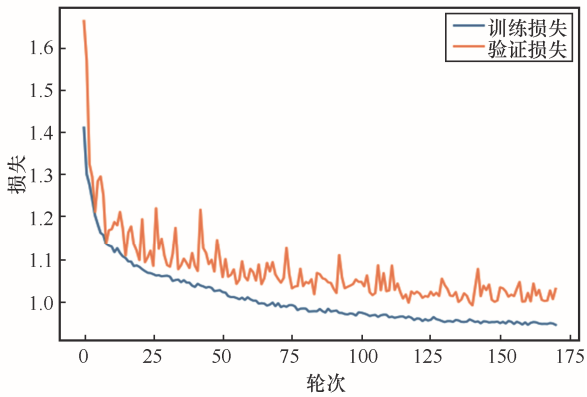
2.4　实验结果

本文旨在研究针对 STM32 的特点进行优化的神经网络模型，并将其与 MobileNet、FD_MobileNet 和 MobileNetv2 进行比较，以评估它们在轻量级图像分类任务中的性能表现。

首先，在五分类苹果叶子病害任务上对这些神经网络进行了实验。在训练过程中，采用早停（early stopping）策略提高模型训练的效率和减少过拟合的风险。早停容忍度被设置为 100 个轮次（epoch），即如果模型在连续 100 个轮次中没有进一步改善，训练过程将提前结束。在五分类数据集的训练中，不同早停容忍度下训练和验证损失如图 6 所示，通过观察图 6（a）发现改进的模型在 100～200 个轮次内已经达到了收敛状态，即模型的验证集损失不再有明显下降趋势。为了进一步提高训练的效率并防止过拟合，调整了早停容忍度，将其减少至 30 个轮次。如图 6（b）所示，通过这一调整，观察到模型仍能在早停容忍度内保持良好的性能，同时训练时间显著缩短。



(a) 早停容忍度为100个轮次



(b) 早停容忍度为30个轮次

图 6　不同早停容忍度下训练和验证损失

五分类苹果叶子病害数据集测试结果见表 5。根据表 5 的结果，MobileNet 和 MobileNetv2 在准确率方面表现接近，但它们的模型参数量、Flash 和 RAM 占用较大。FD_MobileNet 虽然在 RAM 占用方面表现较好，但 Flash 占用较高，并且在该任务中的准确率低于其他两个轻量级神经网络。相比之下，STM32-MobileNet 具有最少的模型参数、最低的 Flash 资源占用，虽然 RAM 占用略高于 FD_MobileNet，但其准确率与其他两个神经网络相当。

七分类番茄叶子病害数据集测试结果见表 6。从表 6 中可以观察到 STM32-MobileNet 在准确率上接近 MobileNetv2，但两项资源占用更少，而 FD_MobileNet 对 RAM 的消耗较低，但准确率也低。

为了更全面地评估这些模型在实际部署时的性能，本文进一步进行了对推理运行时间的统计。

本文使用计算机端的 X-CUBE-AI 工具，模拟了在 STM32 上不同模型的推理运行时间差异。针对五分类任务，本文使用了包含 5 278 张图像的测试集进行推理，并记录了每个模型的推理运行时间

表 5　　五分类苹果叶子病害数据集测试结果

网络模型	准确率（C model）	准确率（Python model）	Flash/KB	RAM/KB	参数量	CPU 推理时间/s
STM32-MobileNet（×0.35）	90.69%	91.89%	256.52	137.26	139 038	512.745
FD_MobileNet（×0.35）	86.39%	88.72%	324.49	96.12	251 429	300.733
MobileNet（×0.35）	90.71%	93.67%	425.40	234.63	319 232	978.16
MobileNetv2（×0.35）	90.67%	92.44%	498.41	165.20	420 667	661.169

表 6　　七分类番茄叶子病害数据集测试结果

网络模型	准确率（C model）	准确率（Python model）	Flash/KB	RAM/KB	参数量	CPU 推理时间/s
STM32-MobileNet（×0.35）	87.28%	87.40%	257.84	137.37	139 919	158.373
FD_mobilenet（×0.35）	80.01%	84.37%	325.20	96.12	252 147	98.934
MobileNetv2（×0.35）	87.52%	89.76%	436.23	237.25	320 134	216.960

（该时间包含结果打印和显示的时间，后续的推理运行时间同理）。根据表 5 的结果，MobileNet 的推理运行时间约为 978 s，MobileNetv2 约为 661 s，FD_MobileNet 约为 301 s，而 STM32-MobileNet 约为 513 s。同样，对于七分类任务，本文使用了包含 1 651 张图像的测试集进行推理，并得到了相应的推理运行时间。统计结果显示：MobileNetv2 约为 217 s，FD_MobileNet 约为 99 s，而 STM32-MobileNet 约为 158 s。通过以上统计结果，可以观察到改进的模型相比经典的轻量级模型 MobileNet 系列，在推理速度上有所提升。

综合考虑以上结果，可以得出结论：在 STM32 上部署时，STM32-MobileNet 相对于其他 3 个轻量级神经网络更适合，不仅表现良好，而且具有较低的资源占用和合理的推理运行时间。

针对上述结果，可以进行更深入的讨论和分析，探讨可能影响分类准确率的因素如下。

1) 模型参数量

模型参数量是一个重要的影响因素，因为参数量越多的模型通常具有更强的表达能力，可以学习更多的特征和复杂的模式，从而提高分类准确率，如 MobileNet。即使没有残差连接，模型依然保持了较高的准确率。然而，在资源受限的嵌入式设备上，较大的参数量可能导致显著的资源占用，如 Flash 和 RAM。因此，在选择模型时需要权衡模型的表达能力和资源占用，并根据实际需求做出合理的折中。

2) 残差连接

在一定程度上，神经网络越深表达能力越强，性能越好。在较小的模型中，为了充分发挥每层的作用，残差连接是一种常用的技术。残差连接可以使得信息更顺畅地传递，能够减少梯度消失，进行有效的反向传播，有助于提高模型的学习能力和分类准确率。在实验中，是否采用残差连接对模型的性能和准确率可能有一定的影响。

3) 模型结构和特征表达能力

不同的模型结构在不同场景条件下对图像分类任务具有不同的特征表达能力。如 FD_MobileNet 在准确率方面稍低，这可能与其较低的模型复杂性和特征表达能力有关，其结构上通过快速下采样实现参数量的下降，同时也没有必要的残差连接，可能导致在下采样前信息的提取不够充分。尽管这种以时间换空间的优化策略带来了更快的推理速度，但在准确率方面有所牺牲。相比之下，MobileNetv2 与本文改进的模型结构更加合理，特征表达能力也更强，因此在轻量化的同时能够保持较好的准确率。

3　结束语

本文提出了一种改进的轻量级神经网络模型，主要针对 STM32 的特点进行优化，以进一步降低其资源占用。该算法以 MobileNetv2 为基础，采用更加资源友好的线性变换方法获取相同数量的特征，并减少部分残差连接，仅保留必要的连接。此外，还减少了下采样操作，以保留更多图像信息。

为了在 STM32 上实现改进模型，本文使用 X-CUBE-AI 工具进行神经网络部署，并通过工具提供的验证指标评估模型在 STM32 上的适用性和性能。实验结果表明，在轻量级的农业病虫害图像分类任务中，该算法表现出色。相对于其他轻量级神经网络，改进模型能够更好地适应 STM32 的硬件资源限制，同时降低神经网络参数量和资源占用，并保持较高的分类准确率。

本文旨在将神经网络模型更高效地应用到以 STM32 为推理模块的经济型边缘设备。这种应用可以在保证图像分类准确率的同时，降低边缘设备成本，并提高农业生产效率。值得强调的是，边缘设备的成本和资源限制往往是制约边缘计算应用的重要因素。因此，在这方面进行改进的研究具有重要意义。此外，该研究在农业领域的应用有助于农民在病虫害监测和治理方面更加高效地进行决策，从而提高作物产量和质量。

综上所述，本文改进模型有潜在的应用价值，可降低边缘设备成本并提高农业效率。通过减少资源占用并保持较高的分类准确率，该模型有望实现降低边缘设备成本和提高农业效率的目标。

参考文献：

[1] 聂鹏程, 张慧, 耿洪良, 等. 农业物联网技术现状与发展趋势[J]. 浙江大学学报(农业与生命科学版), 2021, 47(2): 135-146.
NIE P C, ZHANG H, GENG H L, et al. Current situation and development trend of agricultural internet of things technology[J]. Journal of Zhejiang University (Agriculture & Life Sciences), 2021, 47(2): 135-146.

[2] TZOUNIS A, KATSOULALS N, BARTZANAS T. Internet of things in agriculture, recent advances and future challenges[J]. Biosystems Engineering, 2017(164): 31-48.

[3] 陆林峰, 管孝锋, 黄海龙, 等. 基于农业物联网的应用平台构建[J]. 浙江农业科学, 2020, 61(7): 1455-1457.
LU L F, GUAN X F, HUANG H L, et al. Construction of application platform based on agricultural internet of things[J]. Journal of Zhejiang Agricultural Sciences, 2020, 61(7): 1455-1457.

[4] ZHOU Y Q, TIAN L, LIU L, et al. Fog computing enabled future mobile communication networks: a convergence of communication and computing[J]. IEEE Communications Magazine, 2019, 57(5): 20-27.

[5] ZHOU Y Q, LIU L, WANG L, et al . Service-aware 6G: an intelligent and open network based on the convergence of communication, computing and caching[J]. Digital Communications and Networks, 2020, 6(3): 253-260.

[6] ZHANG X H, CAO Z Y, DONG W B. Overview of edge computing in the agricultural internet of things: key technologies, applications, challenges[J]. IEEE Access, 2020(8): 141748-141761.

[7] SUNYAEV A. Cloud computing[M]//Internet computing. Cham: Springer International Publishing, 2020.

[8] DILLON T, WU C, CHANG E. Cloud computing: issues and challenges[C]//Proceedings of 2010 24th IEEE International Conference on Advanced Information Networking and Applications. Piscataway: IEEE Press, 2010: 27-33.

[9] ALAM T. Cloud computing and its role in the information technology[J]. IAIC Transactions on Sustainable Digital Innovation (ITSDI), 2020, 1(2): 108-115.

[10] 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.
YANG Y R, WU H R, ZHANG Y, et al. Tomato disease recognition using leaf image based on complex environment[J]. Journal of Chinese Agricultural Mechanization, 2021, 42(9): 177-186.

[11] 赵立新, 侯发东, 吕正超, 等. 基于迁移学习的棉花叶部病虫害图像识别[J]. 农业工程学报, 2020, 36(7): 184-191.
ZHAO L X, HOU F D, LYU Z C, et al. Image recognition of cotton leaf diseases and pests based on transfer learning[J]. Transactions of the Chinese Society of Agricultural Engineering, 2020, 36(7): 184-191.

[12] 钟林忆, 刘海峰, 董力中, 等. 计算机视觉下的农作物病虫害图像识别研究[J]. 现代农业装备, 2021, 42(1): 51-55.
ZHONG L Y, LIU H F, DONG L Z, et al. Image recognition of crop diseases and insect pests based on computer vision[J]. Modern Agricultural Equipments, 2021, 42(1): 51-55.

[13] 季力. 基于 STM32 芯片的电参数测量与数据传输[J]. 自动化与仪器仪表, 2010(3): 137-139.
JI L. Power measurement and data transmission based on STM32 chip[J]. Automation & Instrumentation, 2010(3): 137-139.

[14] JACKO P, BEREŠ M, KOVÁČOVÁ I, et al. Remote IoT education laboratory for microcontrollers based on the STM32 chips[J]. Sensors (Basel, Switzerland), 2022, 22(4): 1440.

[15] SANDLER M, HOWARD A, ZHU M L, et al. MobileNetv2: inverted residuals and linear bottlenecks[C]//Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2018: 4510-4520.

[16] OSMAN A, ABID U, GEMMA L, et al. TinyML platforms benchmarking[C]//International Conference on Applications in Electronics Pervading Industry, Environment and Society. Cham: Springer, 2022: 139-148.

[17] CHEN Y P, DAI X Y, CHEN D D, et al. Mobile-former: bridging MobileNet and transformer[C]//Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2022: 5260-5269.

[18] QIN Z, ZHANG Z N, CHEN X T, et al. FD-MobileNet: improved mobilenet with a fast downsampling strategy[C]//Proceedings of 2018 25th IEEE International Conference on Image Processing (ICIP). Piscataway: IEEE Press, 2018: 1363-1367.

[19] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2016: 770-778.

[20] HAN K, WANG Y H, TIAN Q, et al. GhostNet: more features from cheap operations[C]//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 1577-1586.

[21] YANG J W, SHEN X, XING J, et al. Quantization networks[C]//Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 7300-7308.

[22] LIU Z, WANG Y, HAN K, et al. Post-training quantization for vision transformer[J]. Advances in Neural Information Processing Systems, 2021(34): 28092-28103.

[23] JACOB B, KLIGYS S, CHEN B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[C]//Proceedings of 2018 IEEE/CVF Conference on Computer Vision

and Pattern Recognition. Piscataway: IEEE Press, 2018: 2704-2713.

[24] 周敏敏. 基于迁移学习的苹果叶面病害 Android 检测系统研究[D]. 杨凌: 西北农林科技大学, 2019.
ZHOU M M. Research on android detection system of apple leaf diseases based on transfer learning[D]. Yangling: Northwest A & F University, 2019.

[25] JORDAN A A, PEGATOQUET A, CASTAGNETTI A, et al. Deep learning for eye blink detection implemented at the edge[J]. IEEE Embedded Systems Letters, 2021, 13(3): 130-133.

[26] FALBO V, APICELLA T, AURIOSO D, et al. Analyzing machine learning on mainstream microcontrollers[C]//International Conference on Applications in Electronics Pervading Industry, Environment and Society. Cham: Springer, 2020: 103-108.

[27] ALONGI F, GHIELMETTI N, PAU D, et al. Tiny neural networks for environmental predictions: an integrated approach with miosix[C]//Proceedings of 2020 IEEE International Conference on Smart Computing (SMARTCOMP). Piscataway: IEEE Press, 2020: 350-355.

[28] SAILESH M, SELVAKUMAR K, PRASANTH N. A novel framework for deployment of CNN models using post-training quantization on microcontroller[J]. Microprocessors and Microsystems, 2022(94): 104634.

[29] MERENDA M, PORCARO C, DELLA CORTE F G. LED junction temperature prediction using machine learning techniques[C]//Proceedings of 2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON). Piscataway: IEEE Press, 2020: 207-211.

[30] CAPOTONDI A, RUSCI M, FARISELLI M, et al. CMix-NN: mixed low-precision CNN library for memory-constrained edge devices[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2020, 67(5): 871-875.

[作者简介]

许柏涛（1998– ），男，中山大学电子与信息工程学院硕士生，主要研究方向为物联网、图像识别、边缘计算等。

陈翔（1980– ），男，博士，中山大学电子与信息工程学院教授，主要研究方向为无线与移动通信、卫星通信、物联网、电信大数据。