

## 摘要:

如今物联网设备搭载深度学习模型是当下发展趋势, 随着其发展带来的数据隐私安全问题也渐渐引起大众关心, 传统的集中式学习显然已经不能迎合当下发展要求, 用户不愿将设备采集的敏感信息上传, 使得传统集中式学习模型无法得到充足训练数据。因此亟需一种新的解决方案来解决物联网设备的数据孤岛问题。

联邦学习很好的解决了目前设计用户敏感信息的物联网设备的数据训练问题, 用户敏感数据不出本地便可很好的训练出合适的模型。但是传统的智能物联网设备往往算力不足, 往往只能选择轻量化网络进行训练, 如 MobileNet 等专门为移动设备优化过的网络。据我们所知, 很少有研究去探究这些传统轻量化网络在联邦学习中的性能表现以及如何优化其性能表现, 使其在联邦学习场景下能够发挥其更大的性能。基于上述问题, 本文改进了一种更适用于小算力物联网边缘设备在联邦学习背景下的网络模型 FEMBNet, 该模型可以有效提升联邦学习模型在现实场景下的大分辨率数据集的分类准确率, 并且经过优化过的网络结构可以有效减少联邦学习模型参数传输数量, 从而对联邦学习的通信效率产生消极影响, 为迎合数据隐私保护要求日益上涨现状的相关行业提供一个通用的解决方案和新颖的研究方向。

## Part1、Introduction

随着物联网 (IoT) 设备和边缘计算技术的迅猛发展, 许多关键任务正转移至资源受限的边缘设备上。为了实现高效的数据处理和智能决策, 必须在这些设备上部署高性能的深度学习模型。例如, 在智能城市监控、无人机遥感和自动驾驶等应用场景中, 边缘设备不仅需要处理海量数据, 还必须保证模型在各种复杂数据集上的稳定性和高效性。据我们观察, 目前对于小算力边缘设备的网络模型一

般采取轻量化网络模型，这些模型包括但不限于:MobileNet、SqueezeNet、EfficientNet 等轻量化模型。

与此同时，随着边缘设备的广泛部署，大量敏感数据（如用户行为、地理位置和图像信息）在本地生成和处理，用户并不愿意将这些数据共享。为了保护用户隐私，传统的数据集中式训练方法在此背景下变得不再可行。联邦学习[1]作为一种新兴的数据隐私保护解决方案，允许在不传输本地数据的情况下，通过多个设备协作训练模型，从而保证数据的安全性和隐私性。

据我们所知，针对搭载这些模型的边缘计算设备在联邦学习中的应用研究甚少，尤其是在资源受限的边缘设备中。在数据隐私保护愈加受到重视的今天，如何在传统轻量化模型上改进出一种可以在联邦学习框架下高效运行、并在各种数据集上表现卓越的针对小算力边缘嵌入式设备搭载的轻量化模型，是当前亟需解决的关键问题。同时如何保证该客户端层面的改进在不影响联邦学习通信效率的前提下稳步提升联邦学习聚合性能也是我们当下研究的重点。

受到这些的启发，我们有以下问题：

能否基于当下小算力边缘设备中常用的传统轻量网络，改进出一种参数量更少，性能更高，结合联邦学习表现更好的，针对联邦学习分类任务优化的、小算力边缘设备通用网络，用来为当下如火如荼般发展的边缘嵌入式设备的隐私保护问题提供联邦学习背景的解决方案，其网络传输参数量少，能够在相同通信环境下拥有更快的传输速度。并且其也能够有效减少在联邦学习背景下的聚合模型的性能损失。

本研究的贡献在于：提出了一种改进的轻量化深度学习模型 FEMBNet，专为联邦学习场景下的性能优化和应用设计，旨在解决边缘计算中对模型轻量化、

计算效率和数据隐私保护的多重挑战。通过传统轻量化网络模型和 FEMBNNet 在 cifar10 数据集和 UCMerced\_LandUse 遥感数据集上的对比实验，以及在客户端数据分布 IID 和 non-IID 下不同聚合算法和模型对聚合效果的影响对比试验，我们验证了该模型在联邦学习中的有效性和广泛适用性，展示了其在边缘设备中的实际应用潜力。这项研究不仅拓展了轻量化模型在联邦学习中的应用场景，还为未来涉及隐私数据的边缘智能系统与联邦学习的结合提供了新颖的技术解决方案，具有显著的应用前景和推广价值。

## Part2、Related Works

据我们所知，现有的针对小算力边缘设备表现的研究一般集中在如何选择一个表现更好的深度学习模型问题上，而针对小算力边缘设备优化过的网络模型有比如 MobileNet【6】，EiificientNet【12】，SqueezeNet【8】等传统轻量级网络模型，这些模型在集中式学习下已经是表现良好的模型了，而在联邦学习设置下上述研究未曾涉及。因此，【13】研究了传统轻量模型 MobileNetV3 在联邦学习设置下 cifar10 数据集的表现，其表明联邦学习是去中心化机器学习的一种可行技术。【14】探索了不同联邦学习算法在使用 CIFAR-10 数据集增强图像分类任务中的应用，其重点以 VGG16 和 MobileNet 为架构进行探究，以 VGG16 和 MobileNet 在集中学习背景下作为基线，实验表明了除联合平均（FedAvg）外，所有探索的联邦学习算法都表现出优于基线的准确性，探究了联邦学习中不同算法对不同场景设置的影响。而通信效率也是目前研究方向之一，McMahan[1]等人提出的优化方案允许客户端在每次通信轮次中进行多个本地训练轮次，而不是每次都在全局服务器同步。这意味着客户端在向服务器发送更新之前，可以在本地数据上进行更多的训练，这减少了全局同步的频率，从而减

少了通信轮次，提高了联邦学习的通信效率。考虑到梯度传输对联邦学习通信效率也会产生一定的影响，F Sattler[2]等人在【2】中提出了一种稀疏二进制压缩方法，用于在分布式深度学习中减少通信量。通过将梯度稀疏化并压缩为二进制格式，该方法显著降低了通信开销，进而减少了通信轮次，提升了联邦学习数据传输性能。同时，Luping WANG[3]等人在【3】中提出了一种正交的方法(CMFL)，该方法可以识别客户端进行的不相关更新，并防止它们被上传，以减少网络占用空间，从而达到减少通讯轮次和提升通信效率的目的。【4】【5】【15】则是在客户端上传数据后对其进行聚类等操作，执行服务端选择策略的一些聚类 and 筛选客户端方法来提升聚合效率和减少收敛时间的。

而目前这些方法仅仅只是在联邦学习背景下对小算力设备的网络模型选择上和联邦学习聚合策略上的通信效率上进行研究，并且没有一个通用的解决方案，只能是网络模型和联邦学习聚合算法的排列组合。

与上述工作相比，和研究数据异质性和研究聚合策略对通信效率以及联邦学习表现的影响不同的是，本研究更多聚焦在探究针对小算力物联网设备的联邦学习场景下的客户端训练网络层面优化问题，在不影响其通信效率的前提下对其联邦模型的聚合性能有一定提升。本文的解决方案是通过我们在联邦学习场景下的客户端层面的改进，使用改进过的网络 FEMNet 来达到减少客户端传输模型参数量，优化传输效率的同时，还能提升联邦学习最终模型的聚合性能，以此达到联邦学习性能和精度的提升与实现减少联邦学习通信开销的目的，为业界提供一个通用的解决方案，并且为今后相关行业更深入的研究提供一个思路。

## **Part3、Methodology**

### **3.1 baseline 模型概述**

### 3.1.1 baseline 选取与 baseline 实验设置

我们的模型基于小算力边缘嵌入式设备的传统网络模型之一 MobileNetV3-small 改进而来，我们选取了三个常用的针对小算力嵌入式设备优化过的网络模型作为备选，分别是 MobileNetV3-small，EfficientNet，SquzeNet，经过我们的实验证明，MobileNetV3-small 与 EfficientNet 和 SquzeNet 相比，在小分辨率数据集 (cifar10) 上 150-200 轮的分类任务训练上性能表现不如 EfficientNet 和 SquzeNet，但是在高分辨率 (256\*256 彩色遥感图像数据集) 的同样轮数的分类任务准确度上，MobileNetV3-small 的性能大幅领先 EfficientNet 和 SquzeNet，鉴于我们的实验改进的是针对现实生活中的小算力嵌入式边缘设备的改进实验，因此我们选择在高分辨率图像数据下表现好的 MobileNetV3-small 作为我们的 baseline 模型，以 cifar10 数据集和 **UCMerced\_LandUse 数据集** 作为我们的实验数据集，训练 150 轮作为我们的 baseline 实验。

### 3.1.2 MobileNetV3-small 概述

Howard A 在其文章中指出，MobileNetV3-small[6]是在其前身 MobileNetV2 的基础上创新性的引入了 SE 模块和 h-swish 激活函数，同时也重新设计了耗时层的结构，不仅降低了计算开销，同时也提高了网络性能，网络更快更高效。其网络结构如图 1 所示：

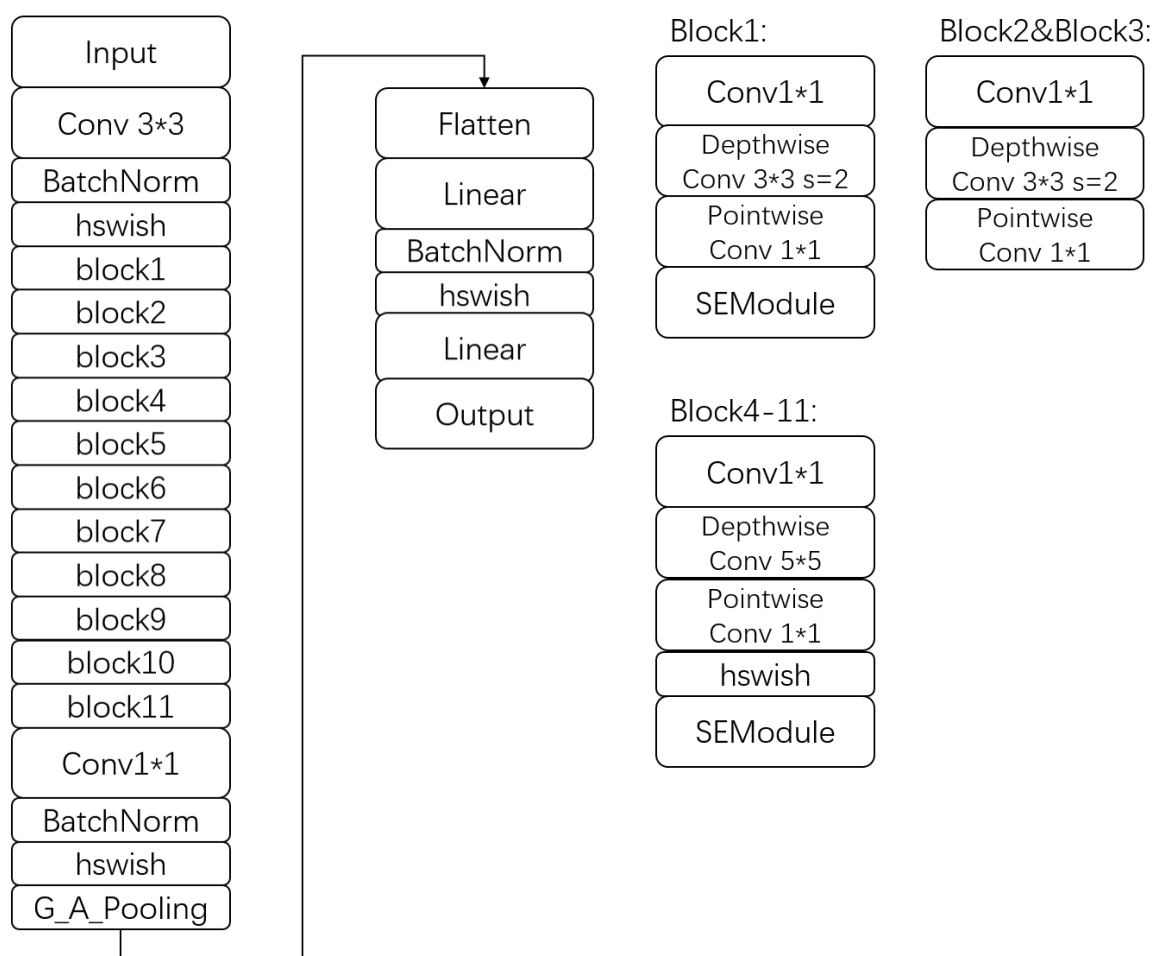


Fig.1 MobileNetV3-small 结构图

网络的输入首先经过一个普通的卷积层（3x3 卷积核）和批归一化层，再经过一个 h-swish 激活函数，这些步骤有助于提取初步的特征。其中，h-swish 激活函数[6]是一种在 swish 激活函数上改进的非线性激活函数，是谷歌于文献[6]中提出，旨在提高模型的表现能力同时减少模型计算复杂度，特别是在轻量级神经网络中，Swish 激活函数的定义为：

$$Swish(x) = x \cdot sigmoid(x) = x \cdot \frac{1}{1 + e^{-x}}$$

然而，Swish 函数的非线性计算涉及指数运算，计算成本较高。为了减少计算复杂度，h-Swish 激活函数采用了分段线性近似方式进行简化，其数学表达式为

$$h\_Swish(x) = x \cdot \frac{ReLU6(x + 3)}{6}$$

其中，ReLU6 函数定义为：

$$ReLU6(x) = \min(\max(0, x), 6)$$

h-Swish 在输入区间 $[-3, 3]$ 内近似于 Swish 函数，而在区间外则为线性映射，这一分段线性策略大幅降低了计算复杂度。

相比 Swish，h-Swish 的计算效率显著提高。h-Swish 仅需一次乘法和一次加法操作，极大地减少了计算开销。同时在 MobileNetV3 的实验中，h-Swish 与 ReLU 相比，显著提升了模型的分类精度，同时保持了低计算成本。接下来的多个块（block1 到 block11）包含了不同的卷积操作，每个块由深度可分离卷积和 SE 模块组成。深度可分离卷积将标准卷积分解为深度卷积和逐点卷积，这样不仅减少了计算量，还保留了模型的表达能力。同时 SE 模块通过自适应的通道权重调整进一步提升了模型对重要特征的关注度。

Chollet F 在文献[7]中提到，深度可分离卷积操作可以分为两个步骤：深度卷积（Depthwise Convolution）和逐点卷积（Pointwise Convolution）。

•**深度卷积**：假设输入特征图大小为  $H \times W \times C_{in}$ ，其中  $H$  是高度， $W$  是宽度， $C_{in}$  是输入通道数。对于每个通道  $c$ ，应用一个  $k \times k$  的卷积核：

$$Y_c = X_c * K_c$$

这里， $*$  表示卷积操作。

•**逐点卷积**：之后，对所有的  $C_{in}$  个通道的输出进行  $1 \times 1$  卷积，得到最终输出  $H \times W \times C_{out}$ ：

$$Y = \sum_{c=1}^{C_{in}} Y_c * K'_c$$

其中  $K'_c$  是  $1 \times 1$  的卷积核。

•**计算量**：标准卷积的计算量为  $H \times H \times C_{in} \times C_{out} \times k \times k$ ，而深度可分离

卷积的计算量为：

$$H \times W \times (C_{in} \times k \times k + C_{in} \times C_{out})$$

这比标准卷积大大减少了计算量。

在这些块处理完后，特征图经过全局平均池化层（G\_A\_Pooling），接着是一个全连接层，该层进一步提取和组合特征，最后通过 h-swish 激活函数进行处理。模型的输出部分是一个线性层，该层将最终特征映射到输出空间。在整个架构中，h-swish 激活函数被多次使用，这种非线性函数能够更好地保留信息流，尤其是在轻量级网络中。总的来说，MobileNetV3 网络通过结合深度可分离卷积、SE 模块和 h-swish 激活函数，在保持较低计算成本的同时，显著提高了模型的表现能力。

### 3.1.3 MobileNetV3 中 SE 模块概述

尽管 MobileNetV3 已经是针对小算力设备进行过优化的网络，但是其本身仍然存在一些缺陷，MobileNetV3 创新性的引入了 SE 模块，SE 模块首次由 Hu J 在文献[8]中提出，SE 模块结构图如下：

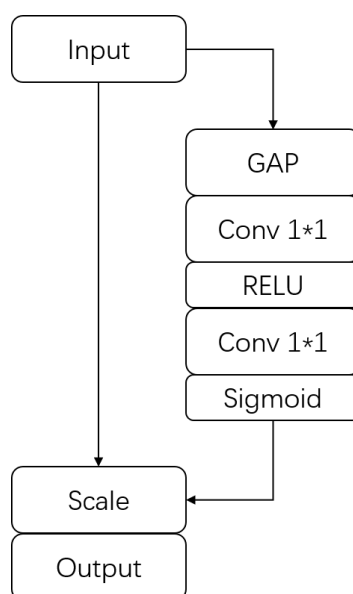


Fig.2 SE 模块结构图

SE 网络架构采用了注意力机制，通过通道上的全局上下文信息来增强特征



表达能力。输入特征图经过前层网络的处理后进入该模块进行 Squeeze 操作，也就是对输入特征图通过全局平均池化层，该层将特征图的空间维度压缩到通道维度，即每个通道仅保留一个值（通道内的平均值），从而获得每个通道的全局信息。设输入特征图为  $X \in R^{H \times W \times C}$ ，其中  $H$  是特征图的高度， $W$  是特征图的宽度， $C$  是通道数。全局平均池化的输出  $z$  为：

$$z_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W X_c(i, j)$$

其中， $z_c$  是第  $c$  个通道的全局描述。

GAP 后的结果通过一个  $1 \times 1$  卷积层进行线性变换，调整通道数目。这一操作可以看作是对通道特征进行加权以捕捉通道之间的关系。卷积后的特征图通过 ReLU 激活函数进行非线性映射。ReLU 函数将输入值中的负值置为零，从而增强网络的表达能力，并引入非线性特征。使用全连接层和激活函数对压缩后的特征进行处理，生成每个通道的缩放系数，也就是 Excitation 操作，设  $z \in R^C$  是从 Squeeze 操作得到的向量。首先通过一个全连接层（FC1）和 ReLU 激活函数获得中间特征：

$$s' = \text{ReLU}(W_1 \cdot z + b_1)$$

其中， $W_1$  和  $b_1$  是 FC1 的权重和偏置。然后，通过第二个全连接层（FC2）和 sigmoid 激活函数获得通道的缩放系数：

$$s = \sigma(W_2 \cdot s' + b_2)$$

其中， $W_2$  和  $b_2$  是 FC2 的权重和偏置， $\sigma$  是 sigmoid 函数。

激活后的特征图再通过一个  $1 \times 1$  卷积层进一步处理，以生成最终的权重图。第二个卷积层的输出通过 Sigmoid 激活函数，将结果映射到 0 到 1 的范围内。Sigmoid 函数用于生成通道上的注意力权重，这些权重将用于对原始输入特征进行调整。这一步也叫 Reweight 操作：设  $s \in R^C$  是从 Excitation 操

作得到的缩放系数向量。调整后的特征图  $\tilde{X}$  为：

$$\tilde{X}_c = s_c \cdot X_c$$

其中， $\tilde{X}_c$  是调整后的第  $c$  个通道特征图， $s_c$  是对应的缩放系数。之后将输出与输入进行点乘，得到最后输出，最终的输出特征图将作为后续网络层的输入。

由文献[8]我们可以得到 SE 模块主要专注于通道间的注意力，它忽略了空间维度上不同位置之间的依赖关系，从而对多尺度特征的处理能力较弱，这意味着在处理具有不同空间尺度的特征时，SE 模块可能无法充分捕捉这些尺度上的信息。同时 SE 模块在空间维度上并没有显著改进空间特征的学习和表示。这可能会导致在处理复杂的视觉任务时，特征空间的关系学习不足，从而影响模型收敛效果。在 MobileNetV3 中 SE 模块虽然有效，但其计算开销和参数量可能在处理大规模数据时成为瓶颈，特别是在高分辨率图像或复杂场景中，这无疑会对联邦学习背景下的模型聚合和通信效率产生一定影响。

#### 3.1.4 EMA 模块概述

而针对这些问题，我们考虑，有没有什么方法不仅仅只关注通道特征，而对多尺度特征有专门的处理，因为如果仅仅只关注通道特征而不关注多尺度特征，就会失去对图像细节或者宏观背景的把控，在处理大分辨率图像时就会损失一定的性能。而且同时又能在空间维度上有一定的改进从而对特征空间的关系学习上有一定优化，并且参数量应该尽可能地少一些，因为我们是要在联邦学习背景下去进行应用，应该本着参数量尽可能少的原则去设计网络的改进，从而不会对联邦学习的通信效率产生消极影响。

因此我们创新性的使用文献[9]提出的 EMANet 来针对网络中的 SENet 进行优化，EMA 模块旨在高效地捕捉多尺度的空间特征，并引入跨空间的学习机

制。EMANet 网络结构如下：

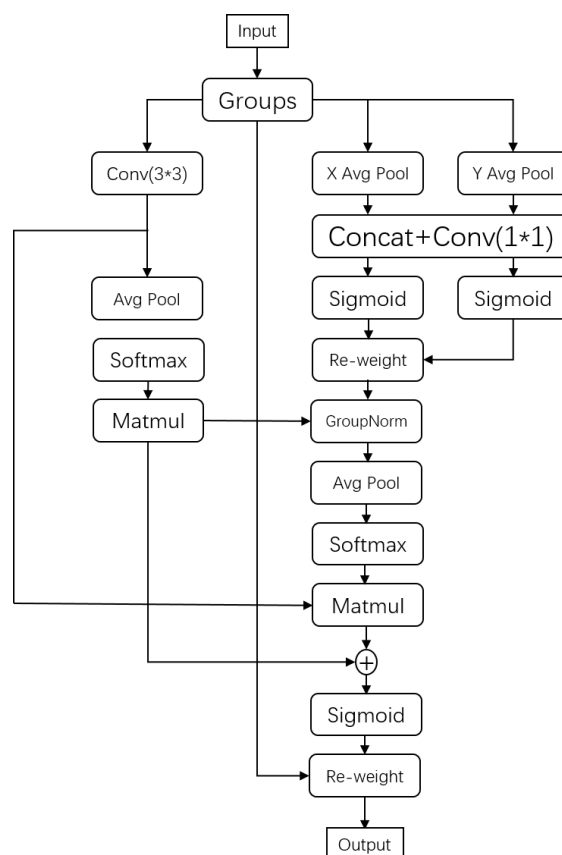


Fig.3 EMA 模块示意图

EMA 模块通过多个步骤来有效捕捉和增强特征图中的重要信息。EMA 模块的具体工作流程如下：

### 1. 特征分组与空间池化:

输入特征图被划分为  $G$  个组, 每组包含  $\frac{C}{G}$  个通道, 生成大小为  $\frac{C}{G} \times H \times W$  的特征子图。这一步骤旨在减少计算量, 同时保留足够的特征表达能力。在每个组内, 特征图分别进行横向和纵向的全局池化操作 (X Avg Pool 和 Y Avg Pool), 生成大小为  $C/G \times 1 \times W$  和  $C/G \times H \times 1$  的特征映射。假设输入特征图为  $X \in R^{C \times H \times W}$ , 将其划分为  $G$  个组, 每组特征图大小为  $X_i \in R^{\frac{C}{G} \times H \times W}$ , 其中  $i = 1, 2, \dots, G$ 。然后对每个组内的特征图进行横向和纵向的全局平均池化:

其中横向池化 (X Avg Pool) 为:

$$P_x(X_i) = AvgPool(X_i, axis = H) \in R^{\frac{C}{G} \times 1 \times W}$$

纵向池化 (Y Avg Pool) 为:

$$P_y(X_i) = AvgPool(X_i, axis = W) \in R^{\frac{C}{G} \times H \times 1}$$

## 2.特征融合

将横向和纵向池化的结果进行拼接, 然后通过  $1 \times 1$  卷积进行融合从而得到包含多尺度空间信息的特征图。其中拼接如下式:

$$P_{concat}(X_i) = Concat(P_x(X_i), P_y(X_i)) \in R^{\frac{C}{G} \times (H+W) \times 1}$$

而卷积融合如下:

$$F_i = Conv_{1 \times 1}(P_{concat}(X_i)) \in R^{\frac{C}{G} \times H \times W}$$

## 3.跨空间注意力计算

卷积后的特征图被分割为两部分, 分别表示横向和纵向的信息。每部分通过 Sigmoid 激活函数生成注意力权重, 用于重新加权原始特征图。假设卷积后的特征图  $F_i$  被分割为两部分  $F_{i,x}$  和  $F_{i,y}$ , 它们分别表示横向和纵向的信息, 注意力权重计算如下:

$$\alpha_x = \sigma(F_{i,x}) \in R^{\frac{C}{G} \times 1 \times W}, \alpha_y = \sigma(F_{i,y}) \in R^{\frac{C}{G} \times H \times 1}$$

其中,  $\sigma(\cdot)$  表示 Sigmoid 激活函数。之后重新加权:

$$X'_i = \alpha_x \cdot X_i + \alpha_y \cdot X_i$$

此操作实现了对不同空间位置的特征进行加权调整, 使得模型能够关注图像中的重要区域。

## 4.组归一化与跨空间学习

对重新加权后的特征进行 GroupNorm 归一化操作, 然后通过全局池化、Softmax 以及矩阵乘法 (Matmul) 计算出新的特征表示。首先对重新加权后的特征图  $X'_i$  进行组归一化:

$$X_i'' = \text{GroupNorm}(X_i')$$

然后进行跨空间学习：

·全局池化：

$$g_i = \text{AvgPool}(X_i'') \in R^{\frac{C}{G}}$$

·softmax 得到权重：

$$w_i = \text{Softmax}(g_i)$$

·矩阵乘法 (Matmul)：

$$Y_i = W_i \cdot X_i''$$

此过程被称为跨空间学习，通过将横向和纵向特征互相结合，进一步捕捉图像中不同位置间的依赖关系。

5.融合与输出，

将跨空间学习后的特征映射  $Y_i$  通过 Sigmoid 激活生成最终权重  $\beta_i$ ，然后对原始特征图进行加权，最后得到输出特征图  $Z$ ：

$$Z_i = \beta_i \cdot X_i$$

其中  $\beta_i = \sigma(Y_i)$ 。

最终输出特征图保留了原始空间维度  $C \times H \times W$ ，但在通道和空间信息的表达上得到了显著增强。此输出包含了跨空间、多尺度的注意力信息，极大地增强了模型对复杂视觉场景的理解能力。

因此 EMA 模块通过更细粒度的空间注意力机制和高效的计算设计，增强了网络在空间维度上的感知能力，进而提升了模型在视觉任务中的表现。通过多尺度空间特征提取与跨空间学习机制，增强了网络对复杂视觉场景的理解能力。同时，EMA 模块在计算效率和性能之间找到了良好的平衡，适用于各种计算资源有限的环境中部署。

EMA 模块相较于 SE 模块的主要改进体现在以下几个方面：**1)多尺度空间**

**特征提取：**EMA 模块采用了多种不同的池化操作，分别对特征图的水平方向和垂直方向进行全局池化，保留了更多的空间上下文信息。这与 SE 模块仅进行全局平均池化的方式相比，能够更细致地捕捉到空间特征，提升了训练效率。

**2)跨空间学习：**EMA 模块通过将水平和垂直池化的结果拼接，并使用卷积操作融合特征，进一步对不同空间位置的特征进行建模。此过程引入了跨空间的依赖关系，使得模型能够更好地理解特征图中各个位置之间的关联。**3)组归一化与高效计算：**为了在保持计算效率的同时增强特征表达，EMA 模块引入了组归一化（Group Normalization）和分组卷积等操作。这些设计不仅减少了计算复杂度，还提升了模块的训练稳定性。比较符合小算力边缘计算设备的特性。

在我们使用 EMA 模块代替 SE 模块的改进下，从而使得网络整体具有更少的参数量，更加广泛的应用场景以及更高的性能表现。而这仅仅是在小算力边缘嵌入式设备的网络模型上的提升，在联邦学习场景下的性能表现我们进行了如下实验：

我们在数据集的准备依然使用 cifar10 数据集和遥感数据集，客户端的数据集分布分为 IID 和 non-IID，聚合算法为 FedAvg 针对 IID 分布情况下的数据集，FedProx 和 FedNova 针对 non-IID 分布情况下的客户端数据集，联邦学习框架使用 flower 框架。接下来将介绍我们使用 EMA 模块代替 MobileNetV3-small 中 SE 模块的网络在联邦学习中的实验设置

## 3.2. 联邦学习概述

### 3.2.1 联邦学习模型概述

**联邦学习**<sup>[1]</sup> (Federated Learning, FL) 是一种分布式机器学习方法，旨在多个客户端（如移动设备或边缘设备）上训练模型，同时保持数据的本地化。

与传统的集中式学习方法不同，联邦学习能够在数据隐私和安全性方面提供显著优势，因为它不需要将数据集中到中央服务器，而是通过在本地进行训练并聚合模型更新的方式来优化全局模型，从而保证了本地数据的隐私安全性，更加符合现代社会对数据隐私安全性的需要。

在联邦学习中，多个客户端共同参与模型的训练过程。每个客户端在其本地数据上独立训练模型，并在完成本地训练后，将模型参数（如梯度或权重）发送给中央服务器。服务器对收到的参数进行聚合，并更新全局模型。这个过程通常通过以下步骤实现：

**模型初始化:**服务器初始化全局模型参数  $\theta_0$  并分发给所有参与客户端。

**本地训练:**在第  $t$  轮中，每个客户端  $k$  使用其本地数据集  $D_k$  对模型进行训练，计算局部更新。假设参与方  $k$  的本地数据量为  $n_k$ ，其损失函数为  $L_k$ ，本地梯度计算公式为：

$$\nabla L_k(\theta_k) = \frac{1}{n_k} \sum_{i=1}^{n_k} \nabla l(\theta_k, x_i, y_i)$$

其中  $(x_i, y_i)$  是样本对， $l$  是损失函数。每个参与方计算其本地模型的梯度，并进行本地更新：

$$\theta'_k = \theta_t - \eta \nabla L_k(\theta_k)$$

其中  $\eta$  是学习率， $\theta_t$  是第  $t$  次迭代的全局模型参数。

**全局模型聚合:**

中央服务器接收到所有参与方的本地模型更新后，使用加权平均的方式更新全局模型。常见的聚合方法包括：

$$\theta_{t+1} = \theta_t - \eta \left( \frac{\sum_{k=1}^K n_k \nabla L_k(\theta_k)}{\sum_{k=1}^K n_k} \right)$$

其中  $K$  是参与方数量， $n_k$  是第  $k$  个参与方的样本量。

在联邦学习中，通常优化的目标是最小化全局损失函数  $L$  其公式为：

$$L(\theta) = \frac{1}{k} \sum_{k=1}^k \frac{n_k}{n} L_k(\theta)$$

其中  $n$  是所有参与方样本的总数。

联邦学习通过这些步骤实现了数据不出本地，将计算任务分散到各个参与方，实现了在保护数据隐私的同时进行有效的模型训练。

### 3.2.2 联邦学习算法概述

在联邦学习服务端聚合策略中，往往不同的聚合算法要根据其场景进行选择，文献[1]指出，客户端数据异质一般分为客户端数据同分布情况（IID）和客户端数据分布不均情况（non-IID）。在现实场景中我们既有可能遇到客户端数据分布 IID，也会遇到客户端数据分布是 non-IID 情况，比如我们需要聚合一位用户的习惯，收集到的数据有来自其汽车上的驾驶习惯数据，有来自其手机的浏览爱好数据等，这些都有一定程度的数据异质。因此我们在聚合算法的选择上应充分选择不同场景下的聚合方法。我们选择[1]中提出的 FedAvg 算法去解决 IID 情况下的数据分布场景，而针对 non-IID 也有不同的划分，比如客户端数据量差异较大的情况以及客户端数据分布差异较大的情况，针对这两种常见的情况，我们分别选用[10]中提出的 FedProx 和[11]中提出的 FedNova 来处理这两种情况。其中 FedProx[10]能有效地处理由于数据异质性带来的模型训练问题，特别是当客户端的数据分布非常不均匀时，能够提供更稳定的全局模型训练效果。而 FedNova[11]适用于需要处理客户端数据量差异较大的情况，并且希望在模型更新中保持较高的公平性。在客户端更新的规模差异较大时，FedNova 的归一化处理能有效改善全局模型的训练效果。

#### 1. FedAvg 算法：



文献[1]指出 FedAvg 算法的核心思想是：在每轮训练中，每个客户端基于其本地数据对模型进行训练，然后将本地模型的更新发送到服务器，其中本地训练按照以下方法进行模型更新：

$$w_k^{t+1} = w^t - \eta \nabla L_k(w^t; D_k)$$

其中：  $w_k^{t+1}$  是客户端  $k$  在第  $t + 1$  轮后的模型参数，  $\eta_k$  是客户端  $k$  的学习率，  $\nabla L_k(w^t; D_k)$  是损失函数  $L_k$  关于模型  $w$  的梯度。

服务器在接收到这些更新后计算其更新后模型参数的加权平均值，并用其更新全局模型。全局模型按照以下方法更新

$$w^{t+1} = \frac{1}{K} \sum_{k=1}^K w_k^{t+1}$$

通常 FedAvg 算法适用于客户端数据分布服从 IID 分布的情况下，而在[10]的研究中表明，其在客户端数据分布服从 non-IID 分布情况下一般表现不好。

## 2. FedProx 算法

文献[10]指出： FedProx 算法旨在通过改进传统的 FedAvg 算法，使其更好地适应数据异质性的挑战。在 FedProx 中，每个客户端在本地训练时，除了优化其本地损失函数，还引入了一个近端项来限制本地模型更新与全局模型的差异。在每轮训练中，每个客户端  $k$  使用当前全局模型参数  $w^t$  在其本地数据集  $D_k$  上进行训练。在本地训练过程中，客户端优化带有 proximal 项的本地损失函数，更新后的模型参数为  $w_k^{t+1}$ 。proximal 项的引入可以用以下公式表示：

$$w_k^{t+1} = \operatorname{argmin}[L_k(w; D_k) + \frac{\mu}{2} \|w - w^t\|^2]$$

其中，  $w$  是参数，  $L_k(w; D_k)$  是客户端  $k$  在其本地数据集上的损失函数，  $\mu$  是近端项的正则化参数，用于控制模型更新与全局模型差异，  $\|w - w^t\|^2$  是模型更新的近端正则化项。接下来服务器将所有客户端上传上来的更新后的模型参数按照以下

方法进行聚合：

$$w^{t+1} = \frac{1}{K} \sum_{k=1}^K w_k^{t+1}$$

其中：  $w_k^{t+1}$  是第  $t + 1$  轮后的模型参数，  $K$  是参与训练的客户端数量。最后重复上述步骤直到满足预设条件。这种方法帮助减少由于数据异质性带来的偏差，使得全局模型能够在客户端数据分布异质的情况下更稳定地收敛。

### 3.FedNova 算法

在文献[11]中，FedNova 算法的主要思想是通过归一化处理客户端的本地更新，使得每个客户端对全局模型的贡献在训练过程中保持一致。传统的 FedAvg 算法在处理客户端的数据分布不均时，可能导致全局模型训练效果不佳，而 FedNova 也是针对 FedAvg 算法进行改进，通过对本地更新进行标准化，成功的提升了全局模型在客户端数据分布不均时的表现能力。在每轮训练中，每个客户端  $k$  使用当前全局模型参数  $w^t$  在其本地数据集  $D^k$  上进行训练。每个客户端在本地训练中对其损失函数进行优化，并计算本地模型更新。客户端的本地更新参数  $w_k^{t+1}$  可以通过以下公式表示：

$$w_k^{t+1} = w^t - \eta \nabla L_k(w^t; D_k)$$

其中，  $w_k^{t+1}$  是客户端  $k$  在第  $t + 1$  轮后的模型参数，  $\eta_k$  是客户端  $k$  的学习率，  $\nabla L_k(w^t; D_k)$  是损失函数  $L_k$  对模型参数  $w$  的梯度。

接下来，对每个客户端的模型更新进行归一化处理。具体地，对于每个客户端  $k$ ，将其模型更新  $w_k^{t+1}$  进行归一化处理，以便在全局模型更新中进行公平的合并。归一化后的更新可以表示为：

$$\tilde{w}_k^{t+1} = \frac{w_k^{t+1} - w^t}{\|w_k^{t+1} - w^t\|}$$

其中，  $\tilde{w}_k^{t+1}$  是归一化后的客户端  $k$  模型更新。

服务器通过对所有归一化后的模型更新进行加权平均来更新全局模型参数。

全局模型参数的更新公式为：

$$w^{t+1} = w^t + \frac{1}{K} \sum_{k=1}^K \tilde{w}_k^{t+1}$$

其中： $w_k^{t+1}$  是第  $t + 1$  轮后的模型参数， $K$  是参与训练的客户端数量。最后重复上述步骤直到满足预设条件。

### 3.3 实验模型架构

本研究考虑了一个通用的联邦学习框架下的小算力边缘物联网设备的联邦学习分类任务的通用网络改进，整体流程如下：

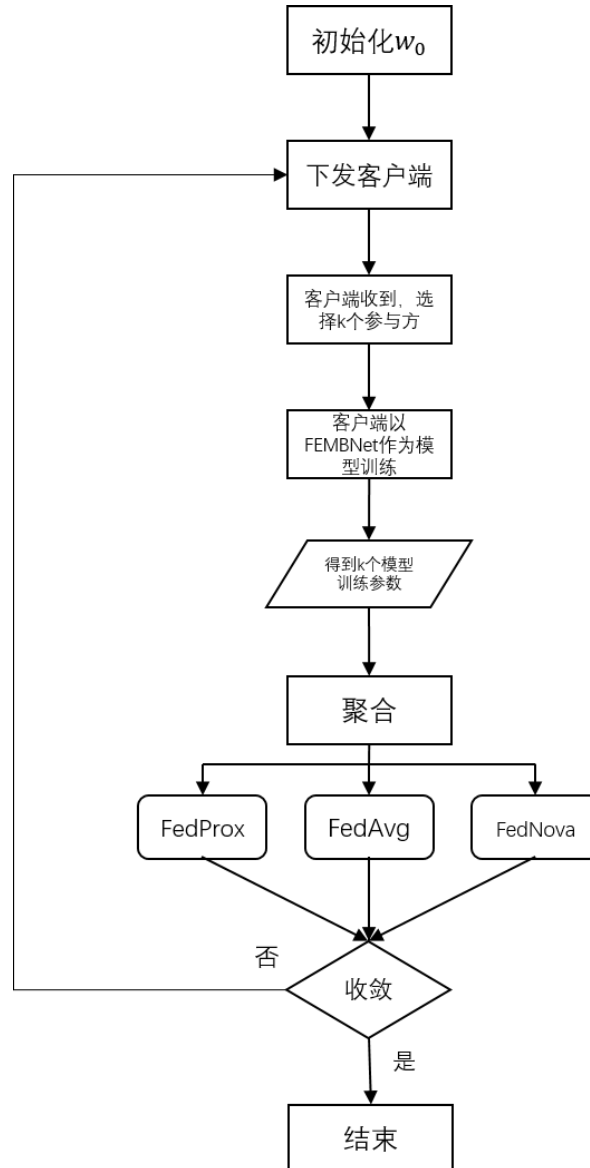


Fig.4 系统整体流程图

其中有服务器  $s$  和  $m$  边缘设备节点, 其中  $m=\{n_1, n_2, \dots\}$  表示  $m$  是由  $n_i$  个边缘设备组成的集合, 我们假设有本地数据  $D_i$ , 用  $D=\{D_1, D_2, D_3, D_4, \dots\}$  表示  $D$  是各个节点本地数据的集合, 边缘设备使用我们改进过的 FEMNet 模型进行训练, 每次实验服务端聚合各个边缘设备计算模型参数时使用的聚合算法相同, 由于聚合算法方面的优化不是我们的关注点, 为了简单起见, 本研究只考虑模拟当小算力物联网边缘设备采集到的数据分布为独立同分布 (IID) 时和数据分布非独立同分布 (non-IID) 时的通用聚合算法, 与传统的机器学习和分布式学习

不同，现实生活中的边缘设备采集到的数据分布并不一定总是 IID 的，数据分布异质化才是现实生活中的主旋律。在本研究中，我们选用经典的 FedAvg 处理 IID 时的模型聚合，同时我们使用 FedProx 和 FedNova 去处理 non-IID 时的模型聚合。

如图，首先服务器初始化一个初始模型参数 $w_0$ ，下发到客户端，客户端根据其设定的选择策略随机选择 $k$ 个客户端作为参与方进行训练，训练模型为我们改进过的 FEMNet，训练完成后得到  $k$  个训练结果模型参数，将  $k$  个模型参数上传至服务器，服务器选择聚合策略，其中可选的聚合策略有针对客户端数据均匀分布情况下的 FedAvg 聚合算法，也有针对客户端数据分布异质的 FedProx 和 FedNova 聚合算法，将聚合结果  $w$  作为下一轮联邦学习的初始模型进行下发，重复这个过程，直到收敛或者达到预设的停止条件。

通过以上的实验方法还有对模型的改进，与 baseline 方法相比，我们的改进模型能够在集中学习背景下在 cifar10 数据集上提升 3% 的准确率，在联邦学习背景下能够在相同设置下提升聚合模型聚合效率 3%~4%，non-IID 分布下提升 3%~4%，并且我们将该模型结合现实中采集的遥感图像数据集 UCMerced\_LandUse，也有明显提升效果，很好的证明了我们的方法是具有普遍适用性的。下一章节我们将展示我们的实验步骤和实验结果。

## Part4 实验结果

### 4.1 传统轻量网络性能比较

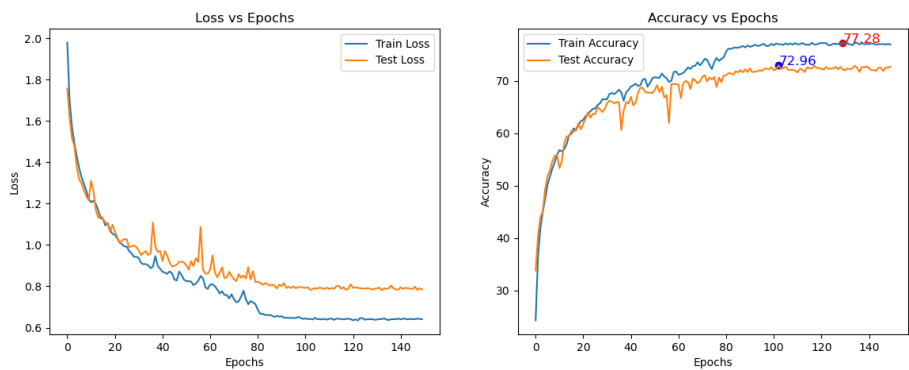
我们选取了三个传统轻量级网络 MobileNetV3-small，EfficientNet，SqueezeNet 作为初始预选模型，分别比较了他们的网络参数量和和在 UCMerced\_LandUse 遥感图像数据集上的最佳准确率表现和其网络参数数量

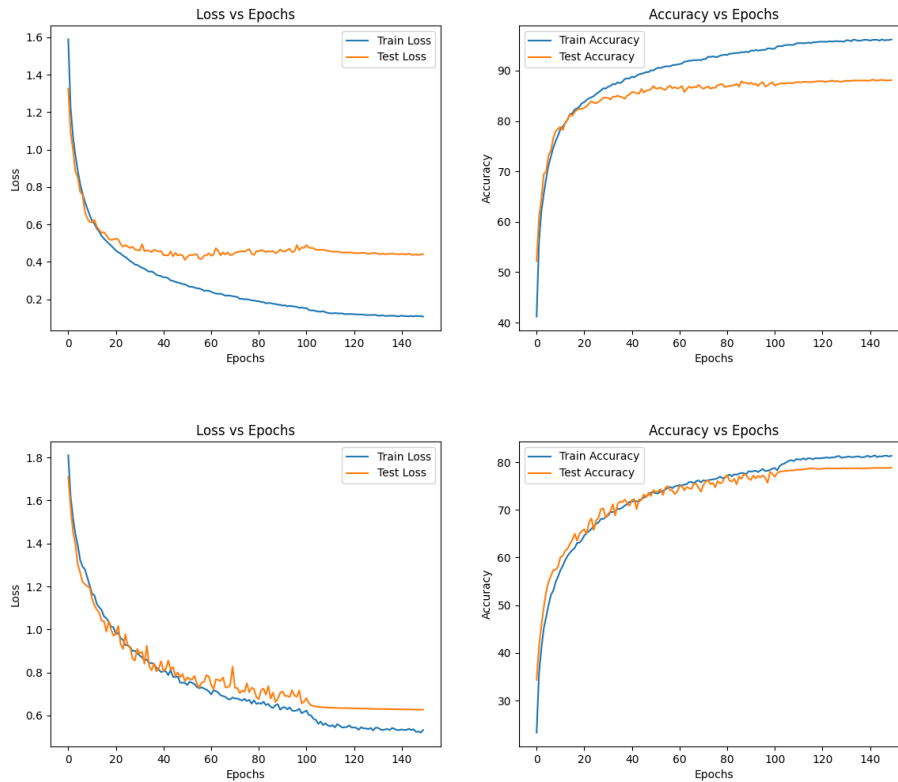
以及浮点数运算次数。其整体结果见表 1。

	cifar10	UCMerced_LandUse	Params	FLOPs
MobileNetV3-small	72.96%	89.04%	1,241,558	0.766G
EfficientNet	88.22%	59.85%	1,392,650	0.87G
SqueezeNet	78.85%	65.19%	734,956	3.39G

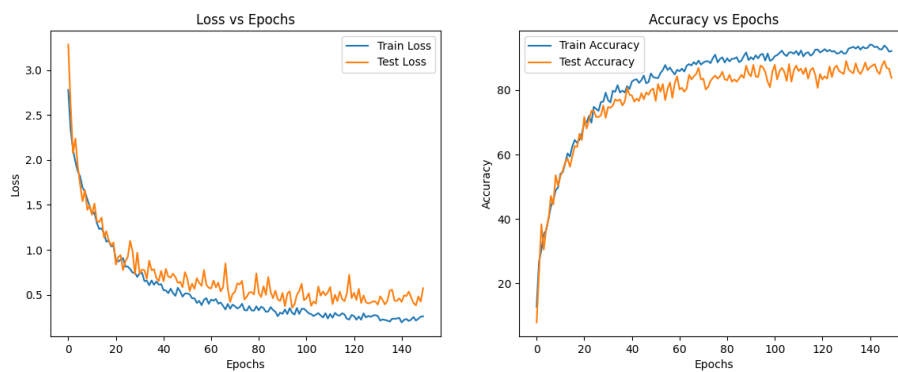
Table1:三个传统网络在 cifar10 数据集和 UCMerced\_LandUse 遥感图像数据集上的最佳准确率表现以及其网络模型参数数量和 FLOPs

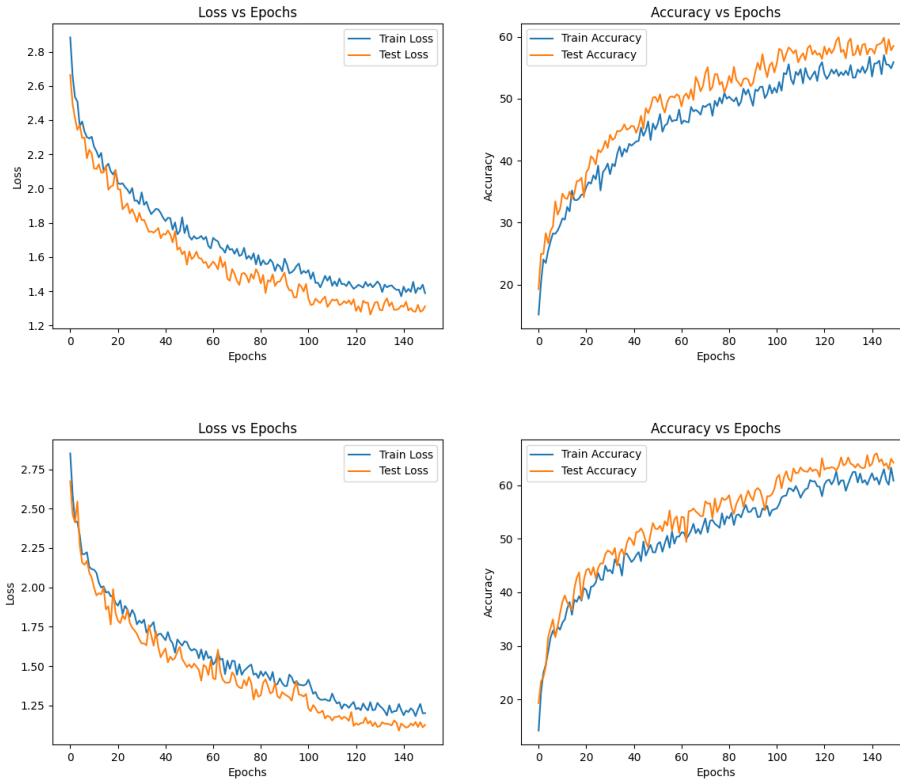
从表中可以看出，SqueezeNet 的参数数量最少，接下来是 MobileNetV3-small，最多的参数数量的网络是 EfficientNet，鉴于其参数量巨大且在现实场景下分辨率图像数据集上表现最差，因此我们优先选择 SqueezeNet 和 MobileNetV3-small 网络作为我们的最终预选模型。我们将初始预选模型在 cifar10 数据集和 UCMerced\_LandUse 数据集上做了对比，在 150 轮的训练周期上表现见下图：





**Fig1 :三个传统轻量网络在 cifar10 数据集上的 150 轮表现，从上往下依次为 MobileNetV3-small, EfficientNet, SqueezeNet 在 cifar10 数据集上集中训练 150 轮次的表现 loss 曲线和准确率 ACC 曲线图。其中在 MobileNetV3-small 上最佳准确率为 72.96%，在 EfficientNet 上最佳准确率为 88.22%，在 SqueezeNet 上最佳准确率为 78.85%。**





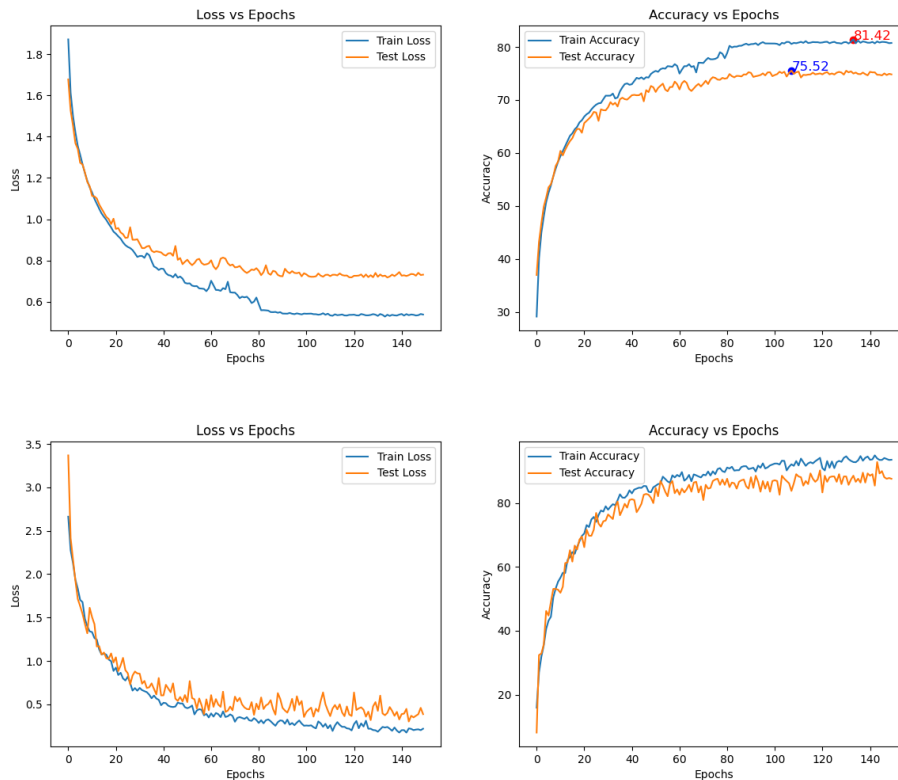
**Fig2: 三个传统轻量网络在 UCMerced\_LandUse 数据集上的 150 轮表现，从上往下依次为 MobileNetV3-small, EfficientNet, SqueezeNet 在 cifar10 数据集上集中训练 150 轮次的表现 loss 曲线和准确率 ACC 曲线图。其中在 MobileNetV3-small 上最佳准确率为 89.04%，在 EfficientNet 上最佳准确率为 59.85%，在 SqueezeNet 上最佳准确率为 65.19%**

可以看出，MobilenetV3-small 在 cifar10 数据集上表现不如其他网络，但是在更贴近现实的大分辨率数据集 UCMerced\_LandUse 上表现遥遥领先，鉴于我们针对现实场景下的嵌入式设备采集的数据进行实验，因此应该以大分辨率数据集上表现良好为主。因此我们选择 MobileNetV3-small 作为我们的 baseline 模型。

## 4.2 FEMBNNet 在数据集上的表现

按照上一章我们改进过的 FEMBNNet 网络，我们将其与上一节 baseline 相同实验设置下以 cifar10, UCMerced\_LandUse 数据集进行集中式训练，实验结果如图 2，并且与 baseline 结果进行比较，比较结果如表 2：





**Fig2:FEMBNet 在 cifar10 数据集与 UCMerced\_LandUse 数据集上的表现，其中第一个图是 FEMBNet 在 cifar10 数据集上的表现，最佳准确率为 75.52%，第二个图是 FEMBNet 在 UCMerced\_LandUse 数据集上的表现，最佳准确率为 92.85%**

	Cifar10	UCMerced_LandUse	Params
<b>Baseline</b>	<b>72.96%</b>	<b>89.04%</b>	<b>1,241,558</b>
<b>FEMBNet</b>	<b>75.52%</b>	<b>92.85%</b>	<b>1,227,692</b>

**Table2:FEMBNet 和 baseline 方法的在不同数据集上表现的对比，以及其网络模型参数量的对比，其中 baselien 方法在 cifar10 上最佳准确率表现 72.96%；在 UCMerced\_LandUse 数据集上最佳准确率表现为 89.04%；而 FEMBNet 分别为 75.52%和 98.58%。并且 Baseline 的网络模型参数量略高于 FEMBNet。**

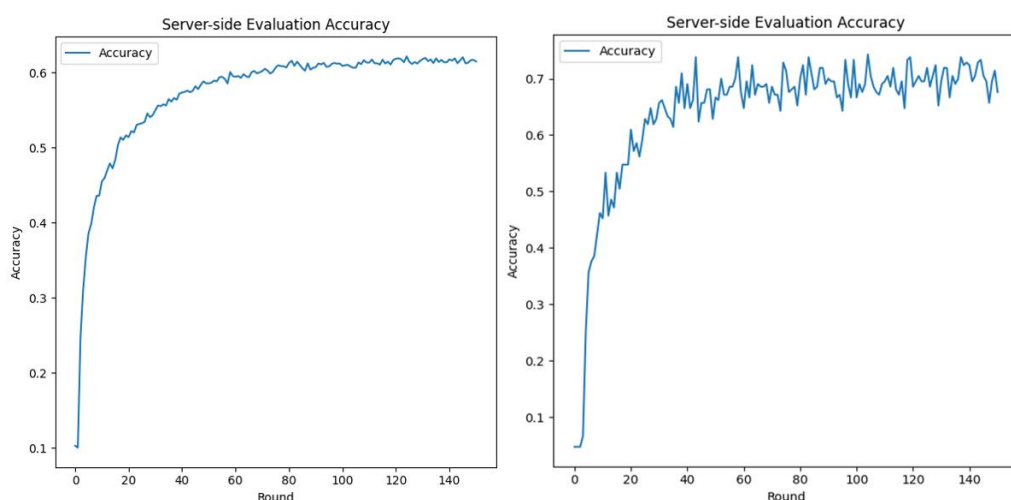
由图 2 和表 2 中我们可以看出，FEMBNet 在集中式学习下在两个数据集上表现都优于 baseline。FEMBNet 在 cifar10 数据集上性能表现高于 baseline 性能表现 2.56%，在 UCMerced\_LandUse 数据集上性能表现高于 baseline 性能表现 3.81%，并且参数量上 FEMBNet 相较 baseline 的更少，由此可以得出，

集中式学习下，我们的改进模型拥有比传统模型更高的准确率，且拥有更少的网络参数，相较于 baseline 方法更轻更快。

### 4.3 MobileNetV3-small 在联邦学习场景下的表现：

#### 4.3.1 客户端数据均匀分布时：

当客户端数据均匀分布时，也就是对应 IID 情况下，我们选择服务端为 FedAvg 聚合算法，使用 Flower 框架在 cifar10 数据集和 UCMerced\_LandUse 数据集上创建 10 个客户端进行 150 个全局轮次的训练，并且每轮训练随机选择 3 个客户端进行训练，得到结果如下：



**Fig3.MobileNetV3-small 在 cifar10 数据集和 UCMercedLandUse 数据集上的联邦学习客户端数据分布 iid 情况下采用服务端 FedAvg 算法的性能表现，左边的曲线图是 cifar10 数据集上的准确率表现，最佳准确率为 60.23%；右边的曲线图是 UCMerced\_LandUse 数据集上的准确率表现，最佳准确率为 75.28%**

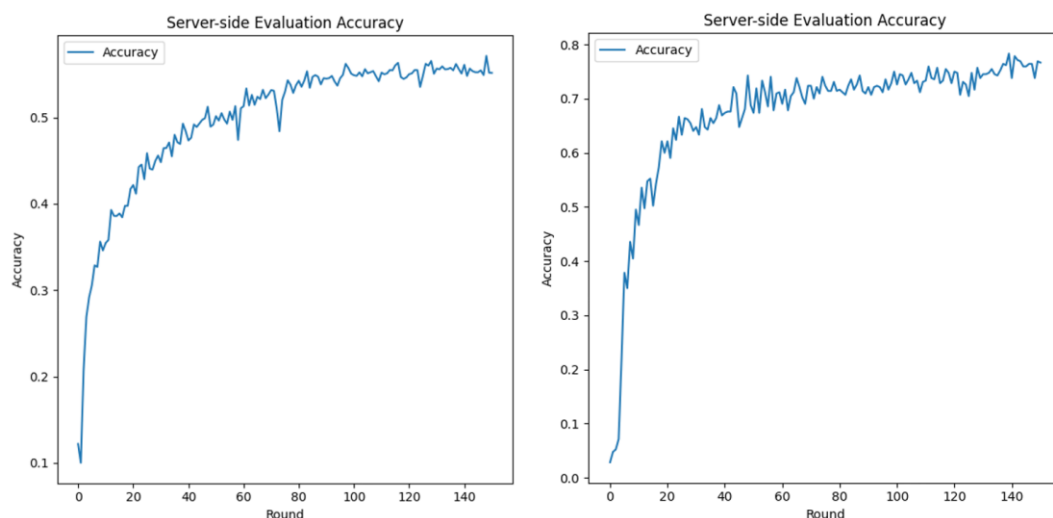
我们以此作为联邦学习场景下的 baseline 实验结果，可以看出，在 IID 场景下的联邦学习 baseline 实验和集中式学习表现类似，但是 baseline 实验在 IID 联邦学习场景下的表现要略低于集中式学习的表现。

#### 4.3.2 客户端数据分布异质：

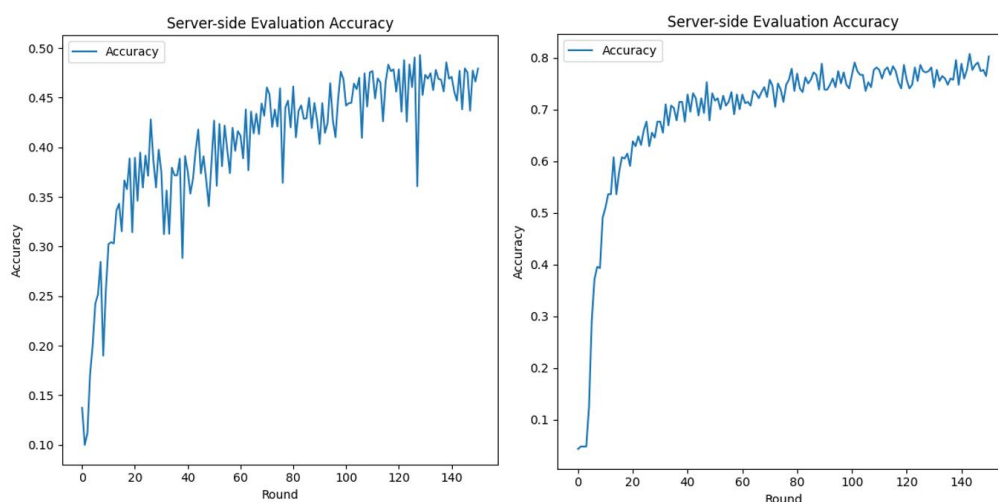
当客户端数据分布异质时，也就是对应 non-IID 情况下，我们选择服务端为

FedProx 和 FedNova 作为聚合算法来模拟不同的 non-IID 场景下的聚合策略。

使用 Flower 框架在 cifar10 数据集和 UCMerced\_LandUse 数据集上创建 10 个客户端进行 150 个全局轮次的训练，并且每轮训练随机选择 3 个客户端进行训练，得到结果如下：



**Fig4. MobileNetV3-small 在 cifar10 数据集和 UCMercedLandUse 数据集上的联邦学习客户端数据分布 non-iid 情况下采用服务端 FedProx 算法的性能表现，左边的曲线图是 cifar10 数据集上的准确率表现，最佳准确率为 57.14%；右边的曲线图是 UCMerced\_LandUse 数据集上的准确率表现，最佳准确率为 78.33%**



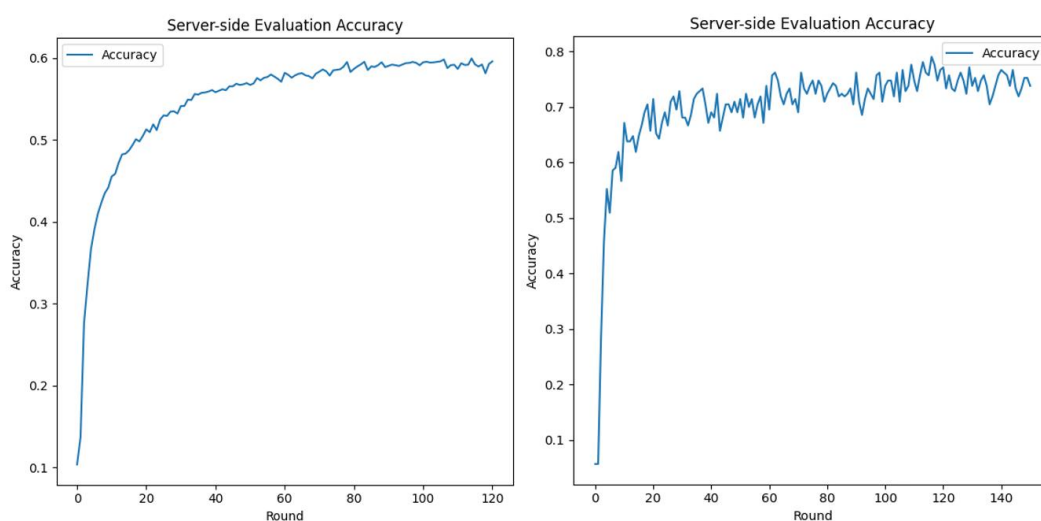
**Fig5. MobileNetV3-small 在 cifar10 数据集和 UCMercedLandUse 数据集上的联邦学习客户端数据分布 non-iid 情况下采用服务端 FedNova 算法的性能表现，左边的曲线图是 cifar10 数据集上的准确率表现，最佳准确率为 48.95%；右边的曲线图是 UCMerced\_LandUse 数据集上的准确率表现，最佳准确率为 80.71%**

截至目前我们已经有了联邦学习下的 baseline 在不同数据集下不同客户端数据分布的实验表现结果, 接下来我们将 FEMBNet 作为本地客户端的模型进行联邦学习训练。

#### 4.4 FEMBNet 在联邦学习设置下的表现

##### 4.4.1 客户端数据均匀分布

当客户端数据均匀分布时, 即 IID 情况下, 我们依然选择 FedAvg 算法作为我们的聚合算法, 使用 Flower 框架在 cifar10 数据集和 UCMerced\_LandUse 数据集上创建 10 个客户端进行 150 个全局轮次的训练, 并且每轮训练随机选择 3 个客户端进行训练, 得到结果如下:



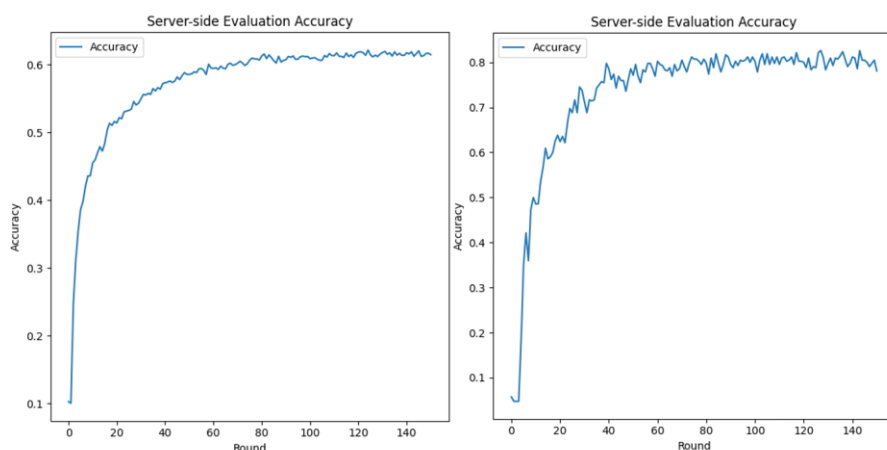
**Fig6.FEMBNet 在 cifar10 数据集和 UCMercedLandUse 数据集上的联邦学习客户端数据分布 iid 情况下采用服务端 FedAvg 算法的性能表现, 左边的曲线图是 cifar10 数据集上的准确率表现, 最佳准确率为 62.61%; 右边的曲线图是 UCMerced\_LandUse 数据集上的准确率表现, 最佳准确率为 79.52%**

我们可以看到联邦学习下 FEMBNet 在不同数据集下的表现, 与联邦学习 baseline 实验设置下的对比可以看出, FEMBNet 在联邦学习 IID 设置中性能表现均优于联邦学习 baseline 实验的 IID 设置下的性能表现, 其中在相同实验设置下, 在 cifar10 数据集上性能提升 2.38%, 在 UCMerced\_LandUse 数据集上

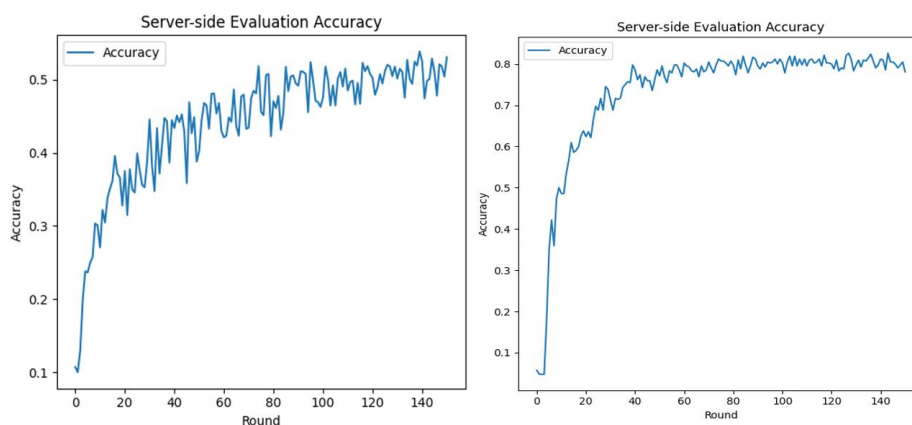
性能提升了 4.24%。可以看出在客户端分布 IID 情况下时，我们改进的模型可以有效提升联邦学习聚合模型的性能。接下来我们测试在 non-IID 设置下模型的对照试验。

#### 4.4.2 客户端数据分布异质：

当客户端数据分布异质时，我们仍然按照联邦学习 baseline 实验设置相同的设置进行实验，选择 FedProx 和 FedNova 算法进行聚合，在 cifar10 数据集和 UCMerced\_LandUse 数据集上进行 150 轮训练，实验结果如下图：



**Fig6.FEMBNet 在 cifar10 数据集和 UCMercedLandUse 数据集上的联邦学习客户端数据分布 non-iid 情况下采用服务端 FedProx 算法的性能表现，左边的曲线图是 cifar10 数据集上的准确率表现，最佳准确率为 61.04%；右边的曲线图是 UCMerced\_LandUse 数据集上的准确率表现，最佳准确率为 82.61%**



**Fig7. FEMBNet 在 cifar10 数据集和 UCMercedLandUse 数据集上的联邦学习客户端数据分布 non-**

iid 情况下采用服务端 FedNova 算法的性能表现，左边的曲线图是 cifar10 数据集上的准确率表现，最佳准确率为 53.85%；右边的曲线图是 UCMerced\_LandUse 数据集上的准确率表现，最佳准确率为 82.85%

我们也可以看出，在联邦学习设置下，FEMBNet 对比联邦学习 baseline 模型 MobileNetV3-small 在客户端数据分布异质下的表现也有不同程度的提升，相同设置下，在 cifar10 数据集上，使用 FedProx 聚合算法时 FEMBNet 能提升 2.9%的聚合模型准确率，使用 FedNova 聚合算法时 FEMBNet 能提升 4.9%聚合模型准确率。同时在 UCMerced\_LandUse 数据集上，使用 FedProx 聚合算法时 FEMBNet 能提升 4.28%的聚合模型准确率，使用 FedNova 聚合算法时 FEMBNet 能提升 2.14%聚合模型准确率。这证明我们改进的模型具在 non-IID 场景下也能有效提升聚合模型性能表现的能力。联邦学习下不同数据集上各种场景的表现见表 3 表 4：

	IID(FedAvg)	Non-IID(FedProx)	Non-IID(FedNova)
Baseline	60.23%	57.14%	48.95%
FEMBNet	62.61%	61.04%	53.85%

Table3:Cifar10 数据集上不同联邦学习客户端数据分布设置下的聚合模型最佳准确率对比。由表中可以看出，FEMBNet 对于 Baseline 在客户端 IID 和 Non-IID 的三种不同聚合算法下分别提升 2.38%，3.9%，4.9%

	IID(FedAvg)	Non-IID(FedProx)	Non-IID(FedNova)
Baseline	75.28%	78.33%	80.71%
FEMBNet	79.52%	82.61%	82.85%

Table4:UCMerced\_LandUse 数据集上不同联邦学习客户端数据分布设置下的聚合模型最佳准确率对比，由表中可以看出，FEMBNet 对于 Baseline 在客户端 IID 和 Non-IID 的三种不同聚合算法下分别提升了 4.24%，4.28%，2.14%

通过我们的实验，我们改进的模型不但在集中式学习中可以有效提升小算力嵌入式设备的性能指标,比传统模型提升了 2%~4%,并且参数量减少了 13.87k,

在联邦学习背景中, 依然能够在传统轻量模型的联邦学习任务基础上有较好的性能提升, 提升聚合模型 2%~5%的精度。证明了我们的模型在不对联邦学习通信效率造成负面影响的同时还能有效提升联邦聚合模型的性能表现, 并且通过不同数据集的对比也很好证明了我们的模型具有一定的通用性。

## **Part5 Conclusion**

我们通过对传统应用在小算力嵌入式设备的轻量网络 MobileNetV3-small 网络结构中固有的不足进行改进, 提出了改进的新网络模型 FEMNet, 该网络专门针对小算力嵌入式设备在联邦学习中分类任务性能的提升做了优化, 通过一系列对比实验证明了相比传统网络具有更少的参数量和更高的精确度, 验证了其相比于传统网络在不同联邦设置下性能能提升 2%~5%, 同时还通过在不同的数据集上进行实验, 证明了其在小算力嵌入式设备的分类任务中具有一定的通用性。我们的网络模型可以应用在任何小算力嵌入式设备的分类任务中。同时也为后续研究提供了新的研究思路。

## **Part6 future works**

我们的研究是在联邦学习客户端网络层面的改进从而影响最终聚合模型的效果, 当然这只是很小一部分, 在我们未来的工作里, 我们会着重探究参与方客户端的行为问题比如是否有恶意客户端干扰的问题, 以及当客户端愿意共享部分数据作为预训练初始化模型时, 如何选择客户端作为预训练客户端并且最终聚合模型性能是否会受影响等方面, 同时我们的工作只聚焦在联邦学习中小算力边缘嵌入式设备的分类任务这个点, 后续也会去探究分割任务或者其它类型的任务在联邦学习背景下的性能提升。

## Part7 reference

- [1] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial intelligence and statistics. PMLR, 2017: 1273-1282.
- [2] Sattler F, Wiedemann S, Müller K R, et al. Sparse binary compression: Towards distributed deep learning with minimal communication[C]//2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019: 1-8.
- [3] Luping W, Wei W, Bo L I. CMFL: Mitigating communication overhead for federated learning[C]//2019 IEEE 39th international conference on distributed computing systems (ICDCS). IEEE, 2019: 954-964.
- [4] Briggs C, Fan Z, Andras P. Federated learning with hierarchical clustering of local updates to improve training on non-IID data[C]//2020 international joint conference on neural networks (IJCNN). IEEE, 2020: 1-9.
- [5] Zhang W, Wang X, Zhou P, et al. Client selection for federated learning with non-iid data in mobile edge computing[J]. IEEE Access, 2021, 9: 24462-24474.
- [6] 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.
- [7] Chollet F. Xception: Deep learning with depthwise separable convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1251-1258.
- [8] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.
- [9] Ouyang D, He S, Zhang G, et al. Efficient multi-scale attention module with cross-spatial learning[C]//ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023: 1-5.
- [10] Li T, Sahu A K, Zaheer M, et al. Federated optimization in heterogeneous networks[J]. Proceedings of Machine learning and systems, 2020, 2: 429-450.
- [11] Wang J, Liu Q, Liang H, et al. A novel framework for the analysis and design of heterogeneous federated learning[J]. IEEE Transactions on Signal Processing, 2021, 69: 5234-5249.
- [12] Tan M. Efficientnet: Rethinking model scaling for convolutional neural networks[J]. arXiv preprint arXiv:1905.11946, 2019.
- [13] Negi V S, Chinara S. Study of MobileNets Model in Federated Learning[C]//2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDDCom/CyberSciTech). IEEE, 2023: 1040-1044.
- [14] Alam E. Enhancing Image Classification With Federated Learning: a Comparative Study of VGG16 and Mobilenet on CIFAR-10[J]. Available at SSRN 4813254, 2024.
- [15] Liu L, Xu X. Marvel: Towards Efficient Federated Learning on IoT Devices[J]. Computer Networks, 2024, 245: 110375.