实验报告

大作业：LiteIResNet网络 实现的动物10分类任务

**最终：91%（见报告第15页）**

组别：第四组

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | Junhui Lin | 学号 | 202211081025 | Leader（模型调研指导，模型改进，参数全过程调整，优化PPT，优化报告） |
| 姓名 | Huang Yang | 学号 | 202211081083 | 了解模型，基准模型测试，参数辅助调整 |

目录

[1. 研究目的 1](#_Toc186460467)

[2. 研究背景和现状 1](#_Toc186460468)

[3. 实验材料 1](#_Toc186460469)

[3.1 数据集介绍 1](#_Toc186460470)

[3.2 环境配置 1](#_Toc186460471)

[4. 实验方法 1](#_Toc186460472)

[4.1 数据处理 1](#_Toc186460473)

[4.1.1 数据预处理 1](#_Toc186460474)

[4.1.2 数据增强 1](#_Toc186460475)

[4.2 模型设计 2](#_Toc186460476)

[4.2.1 MobileNetV3基线模型介绍 2](#_Toc186460477)

[4.2.2 网络结构分析 2](#_Toc186460478)

[4.2.3 设计理念 3](#_Toc186460479)

[4.3 消融实验设计 4](#_Toc186460480)

[4.4 模型训练 4](#_Toc186460481)

[4.5 早停机制 4](#_Toc186460482)

[5. 实验结果 5](#_Toc186460483)

[5.1 基线实验 5](#_Toc186460484)

[5.2 消融实验 6](#_Toc186460485)

[5.2.1 消融实验1：移除ECAAttention 6](#_Toc186460486)

[5.2.2 消融实验2：移除多尺度特征融合 7](#_Toc186460487)

[5.2.3 消融实验3：替换ArcMarginProduct 8](#_Toc186460488)

[5.2.4 消融实验4：移除ECAAttention和多尺度特征融合 9](#_Toc186460489)

[5.2.5 消融实验5：移除ECAAttention和替换分类层 10](#_Toc186460490)

[5.2.6 消融实验6：移除多尺度特征融合和替换分类层 11](#_Toc186460491)

[5.2.7 结果整理 12](#_Toc186460492)

[5.3 总结 12](#_Toc186460493)

[6. 讨论与最终设计 12](#_Toc186460494)

[6.1 实验结果分析 12](#_Toc186460495)

[6.2 改进设计 12](#_Toc186460496)

[7. 结论 18](#_Toc186460497)

[8. 研究意义 18](#_Toc186460498)

[9. 参考文献 18](#_Toc186460499)

1. 研究目的

本实验旨在设计并实现基于**深度学习**的图像分类模型，通过对**Animal-10数据集**进行图像分类任务，探索不同**模型架构设计**对分类性能的影响。通过对比**基线模型**与**改进模型**的表现，进一步优化图像分类模型的性能，旨在提升对动物图像的分类准确率。

2. 研究背景和现状

**图像分类**是计算机视觉领域的核心任务之一，近年来，深度学习技术在该领域取得了巨大的突破。尤其是**卷积神经网络（CNN）**在图像分类中的表现优异。**轻量级模型**，如MobileNetV3，因其优越的性能和较低的计算开销，在实际应用中得到了广泛关注。本实验基于**MobileNetV3架构**进行实验，探索不同**改进模块**对模型性能的提升，并进一步优化图像分类的效果。

3. 实验材料

3.1 数据集介绍

* Animal-10：包含10种动物类别（狗、马、大象、蝴蝶、鸡、猫、牛、蜘蛛、羊、松鼠）。

数据分布：

· 训练集：14,720张图像

· 验证集：1,840张图像

· 测试集：1,840张图像

分配比例：8:1:1。

3.2 环境配置

编程语言：Python 3.x

深度学习框架：PyTorch

使用的库和工具：

* seaborn、tqdm、scikit-learn、torchinfo、pywavelets

4. 实验方法

4.1 数据处理

4.1.1 数据预处理

图像缩放至**224×224**。

**归一化处理**，使用的mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]。

4.1.2 数据增强

由于 Animal-10 数据集规模较小，且图像内容多是自然场景中的动物，与 ImageNet 分布有一定相似度，我们可以直接使用 ImageNet 的均值和标准差 来做归一化，同时在训练集上使用更“自动化”或“多变”的增广策略，如 RandAugment 或 AutoAugment。

**为何建议使用 ImageNet 均值和标准差？**

（1）小数据集自己统计的均值、方差可能不够稳定；

（2）预训练模型或与自然图像分布相似时，ImageNet 的统计量是更通用且稳定的选择。

相关操作：

（1）**水平翻转**：模型不会对特定方向的特征产生偏好，提高对不同视角的适应性。

（2）**随机旋转图像**：真实场景中，动物可能以不同的角度出现，旋转增加了模型对角度变化的鲁棒性。

**防止过拟合**：通过多角度的训练，模型不易记住特定角度的特征。

（3）**随机仿射变换**：包括平移（10%）、缩放（90%-110%）、剪切（±10度）。模拟图像在不同位置、不同尺寸和不同视角下的变化，增强模型对空间变换的适应能力。

多样性：结合平移、缩放和剪切，提供更丰富的数据变体。

（4）**随机裁剪图像的一部分**：不同的裁剪区域让模型学习到图像的不同部分，提高对部分遮挡或视角变化的鲁棒性。

（5）**颜色抖动**：模拟不同光照条件下的图像，增强模型对光照变化的适应性。防止模型过于依赖特定颜色特征，提高对颜色变化的鲁棒性。

（6） **随机擦除**：模拟图像中对象被部分遮挡的情况，增强模型的鲁棒性；防止模型过度依赖特定区域的特征，提高对全局信息的学习。

4.2 模型设计

基线模型：MobileNetV3。

初始改进模型：ImprovedAnimalNetV2，具体改动如下：

|  |  |  |
| --- | --- | --- |
| 模块 | MobileNetV3 | ImprovedAnimalNetV2 |
| Stem | Conv-BN-Hardswish | Conv-BN-Hardswish |
| 倒残差块 | 使用SE模块增强通道注意力 | 使用ECAAttention模块替代SE，增强通道注意力 |
| 多尺度特征融合 | 无 | 在特定层输出中间特征，通过全局平均池化后融合 |
| 分类层 | 标准Linear层 | 使用ArcMarginProduct分类层 |

4.2.1 MobileNetV3基线模型介绍

MobileNetV3是谷歌推出的**轻量级神经网络**，主要设计目标是提供高效的图像处理能力，特别适用于计算资源有限的移动端和嵌入式设备。与传统的卷积神经网络（如VGG）相比，MobileNetV3通过优化网络结构、引入倒残差模块和SE注意力机制，显著减少了模型的参数量和计算量。尽管参数量大幅度减少，MobileNetV3仍能保持较高的分类精度，特别适用于嵌入式设备、移动终端等硬件资源受限的场景。

4.2.2 网络结构分析

1. 初始卷积（3→16）

功能：将输入图像的通道数扩展至16，采用较大的步幅，确保网络能够在开始时就有效减少空间维度，为后续特征提取提供基础。

1. 倒残差模块

功能：通过**深度可分离卷积**降低计算量，同时结合**SE模块**增强通道间的注意力分配，提高网络对关键特征的表达能力。

具体结构：

* Block1（16→16）
* Block2（16→24→24）
* Block3（24→40→40→40，含SE模块）
* Block4（40→80→80→80，含SE模块）
* Block5（80→112→112，含SE模块）
* Block6（112→160→160，含SE模块）

1. 全局平均池化

功能：通过对每个通道的全局平均，提取该通道的全局信息，并将空间维度压缩为1×1。此操作减少了特征图的维度，为后续的分类过程提供精简的特征。

1. 全连接分类层（160→10）

功能：将最终提取的特征进行分类，输出10个类别的预测结果。

**SE模块实现流程**：

* 自适应平均池化：首先通过全局平均池化提取每个通道的全局信息，为后续的权重计算提供输入
* 全连接层生成通道权重：通过两个全连接层对提取到的全局信息进行处理，生成通道的权重。经过激活函数处理后，权重映射到[0, 1]区间。
* 增强通道特征：通过将原始输入特征图与计算出的权重相乘，增强重要通道的特征表示，帮助网络聚焦于对分类任务更为重要的特征。

**倒残差模块实现流程**：

* 通道扩展：首先使用1×1卷积扩展通道数，为后续的特征提取提供更大的容量。
* 深度可分离卷积：采用深度可分离卷积将输入特征图中的每个通道分别处理，显著减少计算量。
* 通道权重分配（可选）：通过SE模块对通道进行权重调整，提升对关键通道的关注。
* 通道还原与残差连接：通过1×1卷积还原通道数，并通过残差连接避免信息丢失，帮助网络更深层次的训练。

4.2.3 设计理念

在MobileNetV3的基础上进行多次**消融实验**，通过移除或替换不同的网络模块，分析各个模块对模型性能的影响。我们旨在通过对MobileNetV3的简化与改进，找到在特定应用场景下更高效且准确的网络结构。

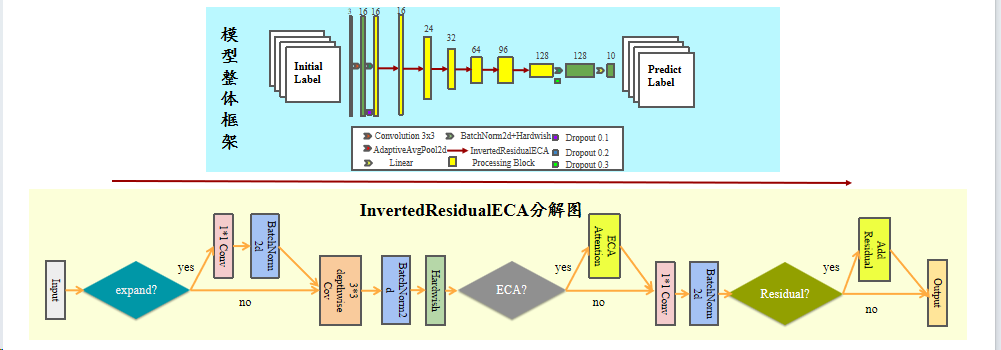


图4. 1模型结构图

* **基于MobileNetV3的轻量化设计**：MobileNetV3本身已经是一种高效且适用于移动端的轻量级网络，但仍可以通过去除或替换某些模块，进一步减少计算量和参数量。通过消融实验，我们评估了哪些模块的去除对性能影响最小，以及如何在保留较高准确度的同时进一步优化模型。
* **多模块的消融实验**：实验包括移除ECAAttention模块、去除多尺度特征融合、替换ArcMarginProduct分类层等，目的是探索各个模块对模型表现的贡献。通过这些实验，我们希望能够实现更高效的网络结构，尤其是在资源有限的设备上。
* **模型性能与训练稳定性优化**：除了消融实验外，我们还通过调整超参数（如学习率、批量大小等）和引入数据增广策略，来进一步优化模型的训练过程，确保其在较小数据集上的泛化能力。

4.3 消融实验设计

消融实验1：移除ECAAttention模块，使用传统的倒残差块。

消融实验2：移除多尺度特征融合，只使用最终特征进行分类。

消融实验3：替换ArcMarginProduct分类层，使用标准的线性分类层。

消融实验4：移除ECAAttention模块和多尺度特征融合。

消融实验5：移除ECAAttention模块和替换分类层。

消融实验6：移除多尺度特征融合和替换分类层。

4.4 模型训练

(1)损失函数 Label Smoothing：在交叉熵损失函数的基础上，对标签进行平滑处理。

通过将标签从“硬标签”（例如，某类为1，其余为0）转变为“软标签”（例如，某类为0.9，其余为0.1），减少模型对特定样本的过度依赖。让模型在训练时更注重整体趋势，而不是记忆训练数据的细节。

(2) 学习过程

**优化器**：可以使用**AdamW**，在小数据集上泛化好。

**初始学习率**：lr=3e-4 ~ 1e-3 区间都可尝试。如果 batch\_size 很大(>=128)，可以略调大 lr；如果 batch\_size 小，可以倾向用 3e-4。

**调度器**：使用 CosineAnnealingLR ，让学习率平滑衰减。

保持 patience=10 左右更保险（你原先是 7）。小数据集有时候波动较大，多给点耐心可能会收获更好结果。

**训练设置**：

* Batch size: 64
* Epochs: 80
* 混合精度训练（AMP）

4.5 早停机制

**耐心度**：10个epoch。

最佳模型保存在best\_model.pth。

5. 实验结果

5.1 基线实验

实验结果：（见图1. 1、图1. 2）

准确率：82.66%

精确率：82.92%

召回率：82.66%

F1分数：82.59%

总结：基线实验表明，MobileNetV3在Animal-10数据集上的整体表现较为均衡，准确率达到了82.66%，但仍有改进空间。

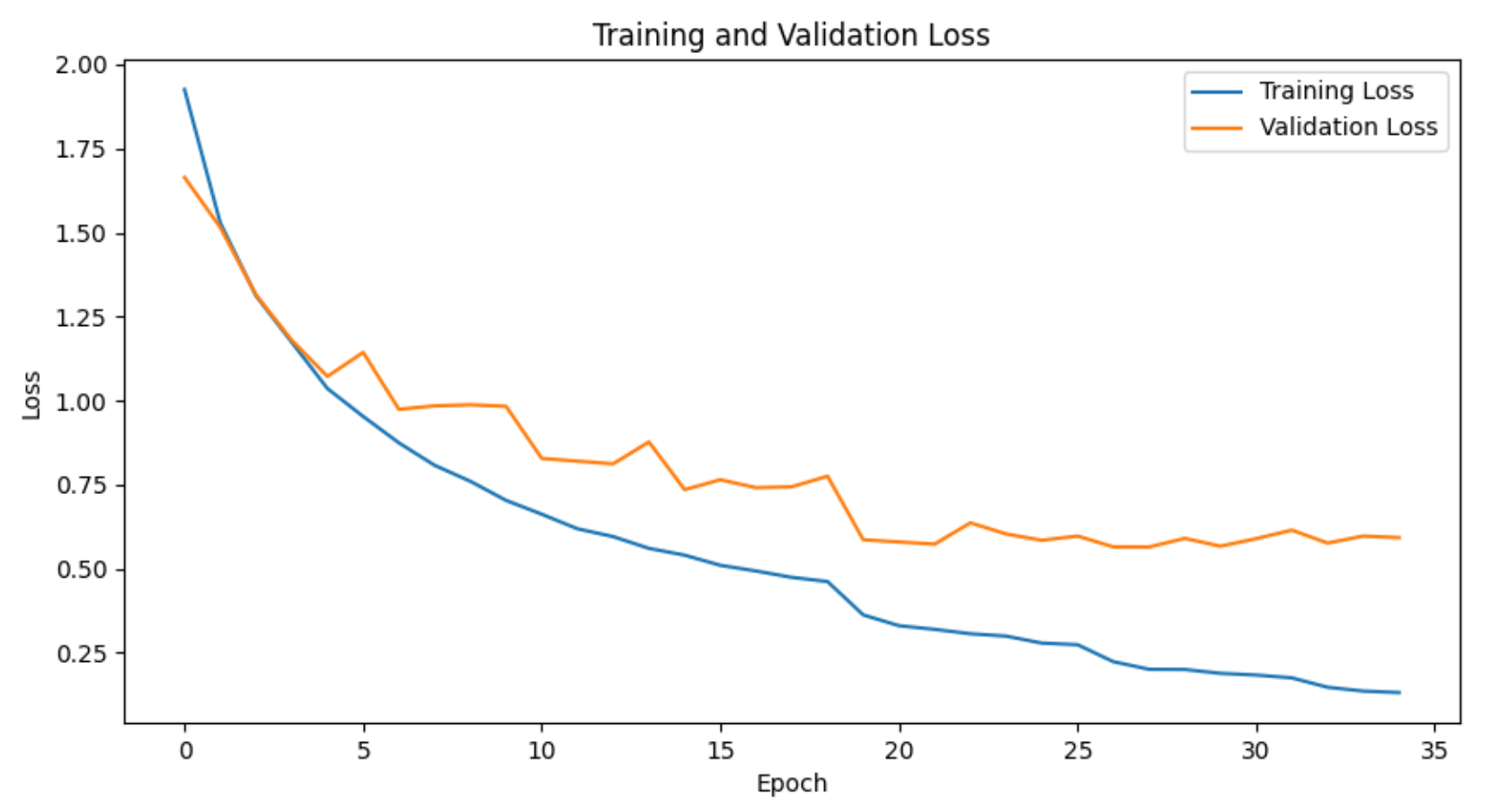


图1. 1基线实验训练与验证损失图

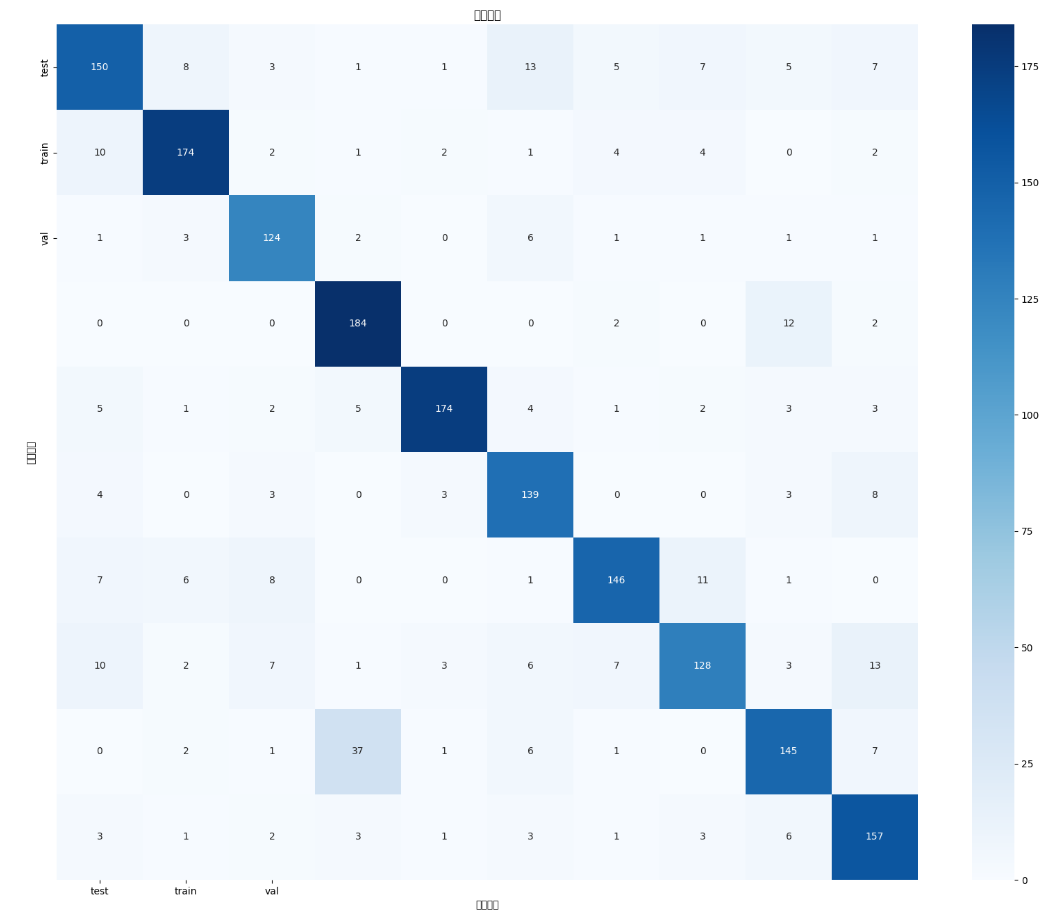


图1. 2基线实验混淆矩阵（X轴：预测标签，Y轴：真实标签）

5.2 消融实验

5.2.1 消融实验1：移除ECAAttention

实验结果：（见图2.1. 1、图2.1. 2）

准确率：83.64%

精确率：83.84%

召回率：83.64%

F1分数：83.55%



图2.1. 1消融实验1训练与验证损失图

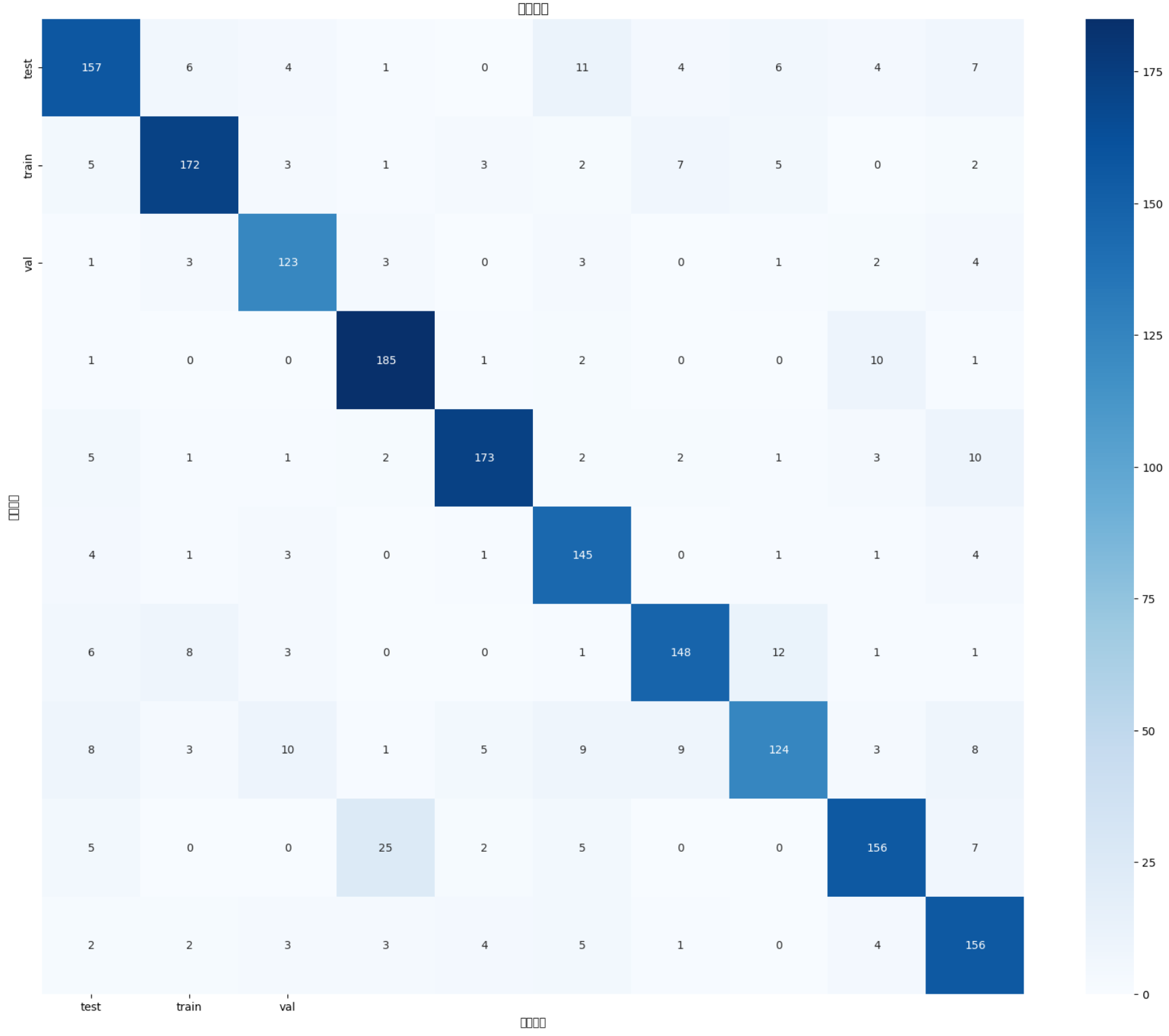


图2.1. 2消融实验1混淆矩阵

5.2.2 消融实验2：移除多尺度特征融合

实验结果：（见图2.2. 1、图2.2. 2）

准确率：82.01%

精确率：82.42%

召回率：82.01%

F1分数：81.92%



图2.2. 1消融实验2训练与验证损失图

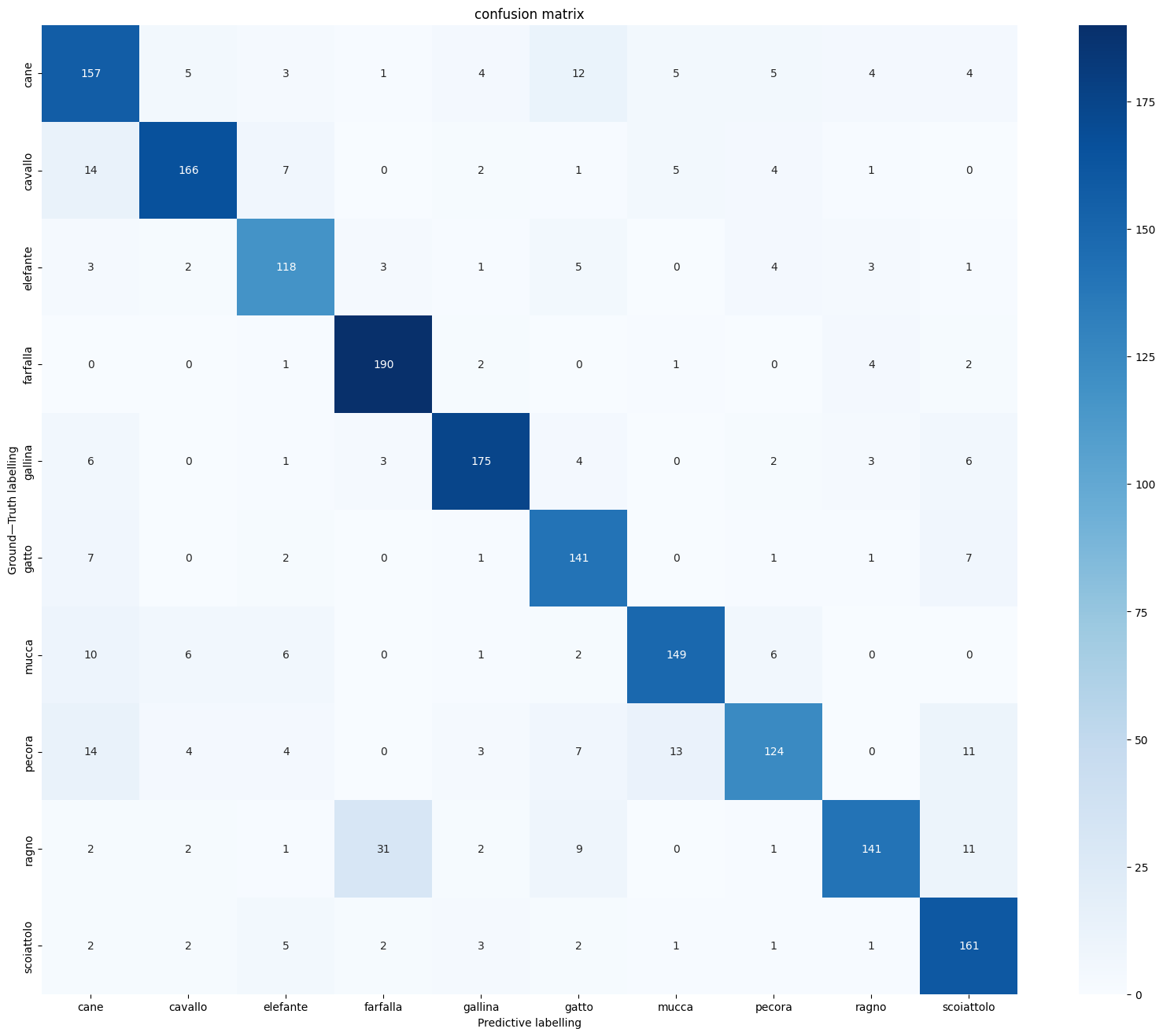


图2.2. 2消融实验2混淆矩阵

5.2.3 消融实验3：替换ArcMarginProduct

实验结果：（见图2.3. 1、图2.3. 2）

准确率：83.64%

精确率：83.67%

召回率：83.64%

F1分数：83.60%



图2.3. 1消融实验3训练与验证损失图

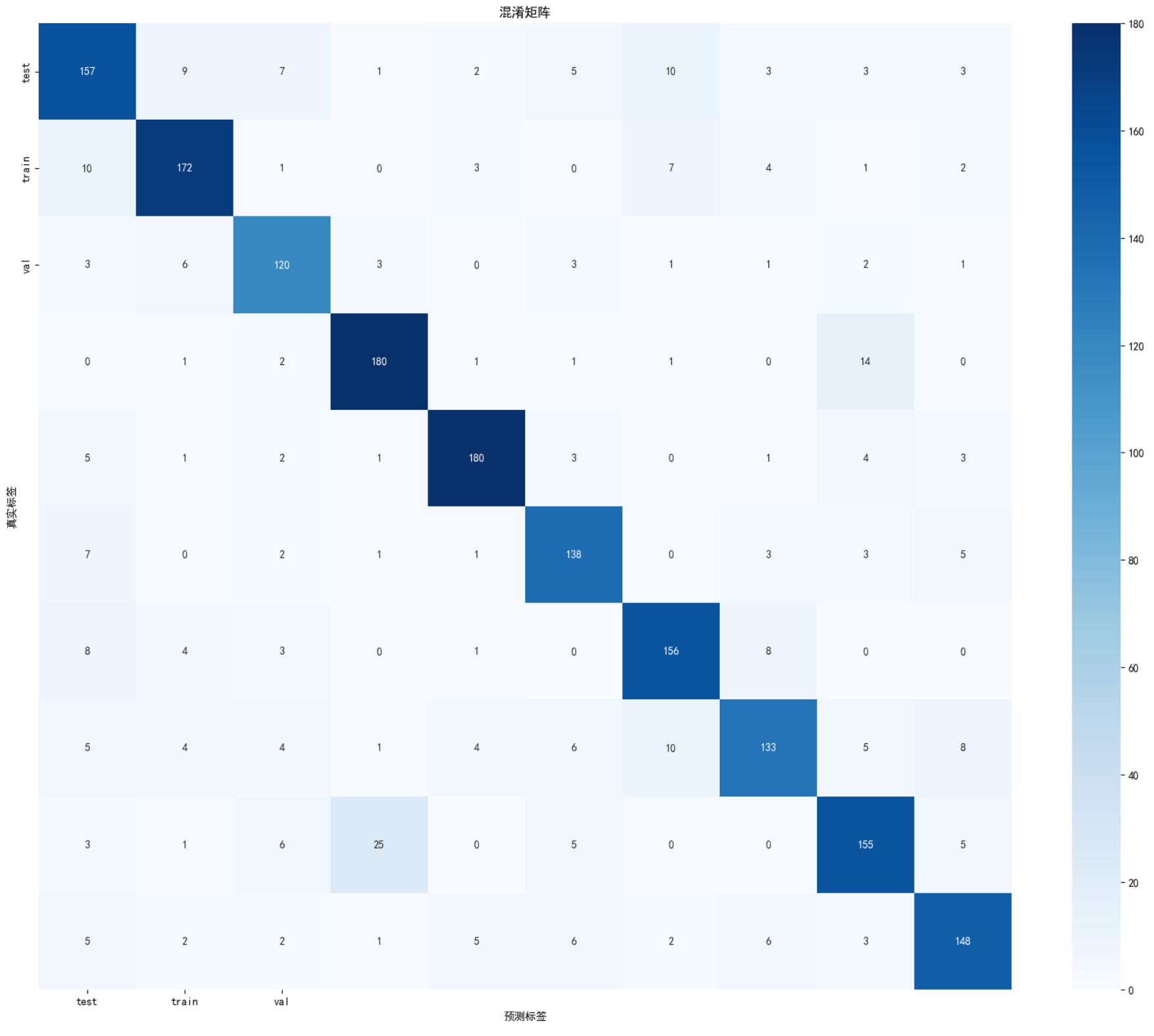


图2.3. 2消融实验3混淆矩阵

5.2.4 消融实验4：移除ECAAttention和多尺度特征融合

实验结果：（见图2.4. 1、图2.4. 2）

准确率：83.64%

精确率：83.83%

召回率：83.64%

F1分数：83.57%



图2.4. 1消融实验4训练与验证损失图

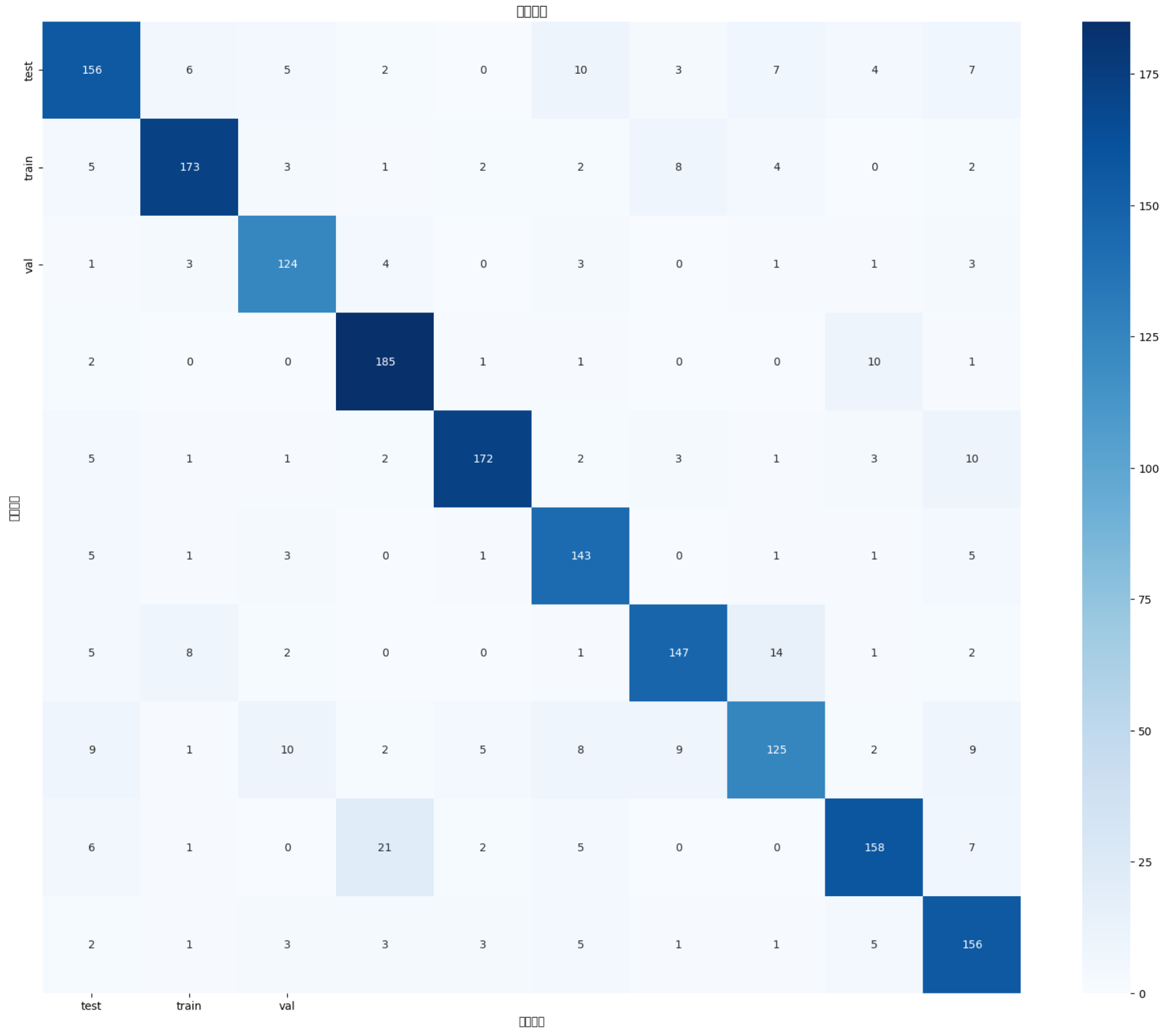


图2.4. 2消融实验4混淆矩阵

5.2.5 消融实验5：移除ECAAttention和替换分类层

实验结果：（见图2.5. 1、图2.5. 2）

准确率：84.13%

精确率：84.31%

召回率：84.13%

F1分数：84.05%

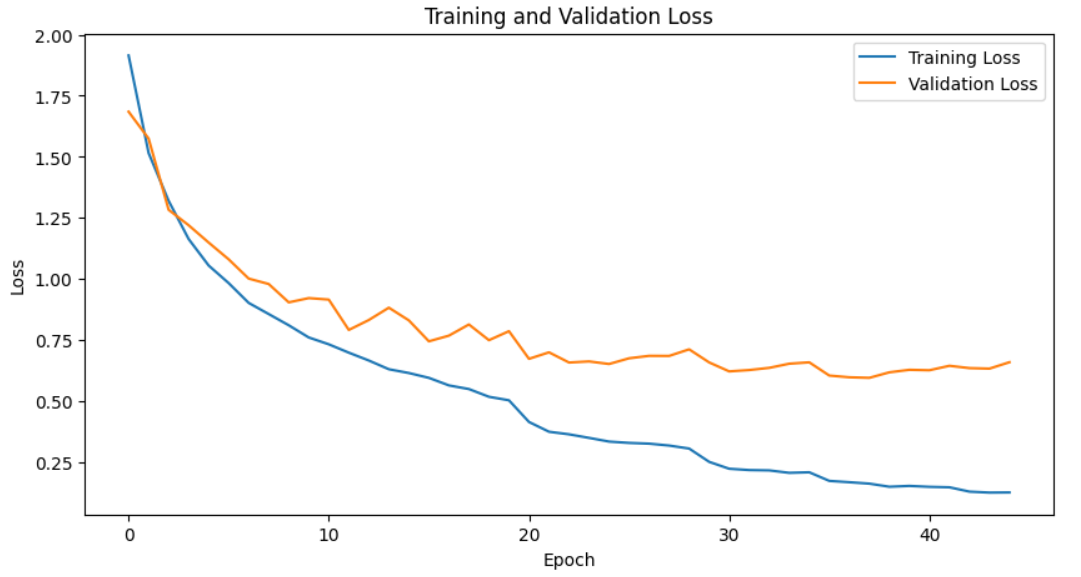


图2.5. 1消融实验5训练与验证损失图

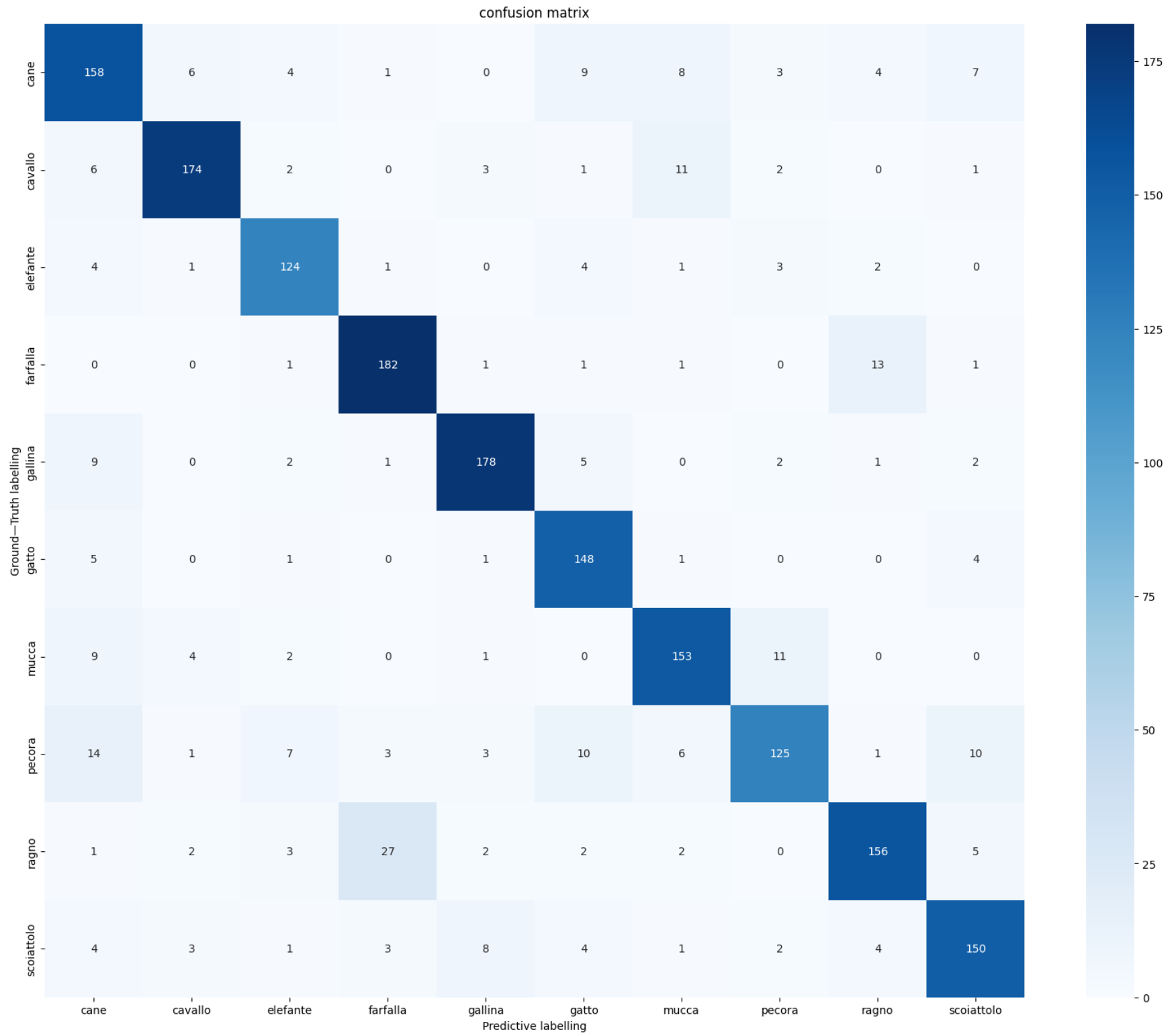


图2.5. 2消融实验5混淆矩阵

5.2.6 消融实验6：移除多尺度特征融合和替换分类层

实验结果：（见图2.6. 1、图2.6. 2）

准确率：84.51%

精确率：84.92%

召回率：84.51%

F1分数：84.53%



图2.6. 1消融实验6训练与验证损失图

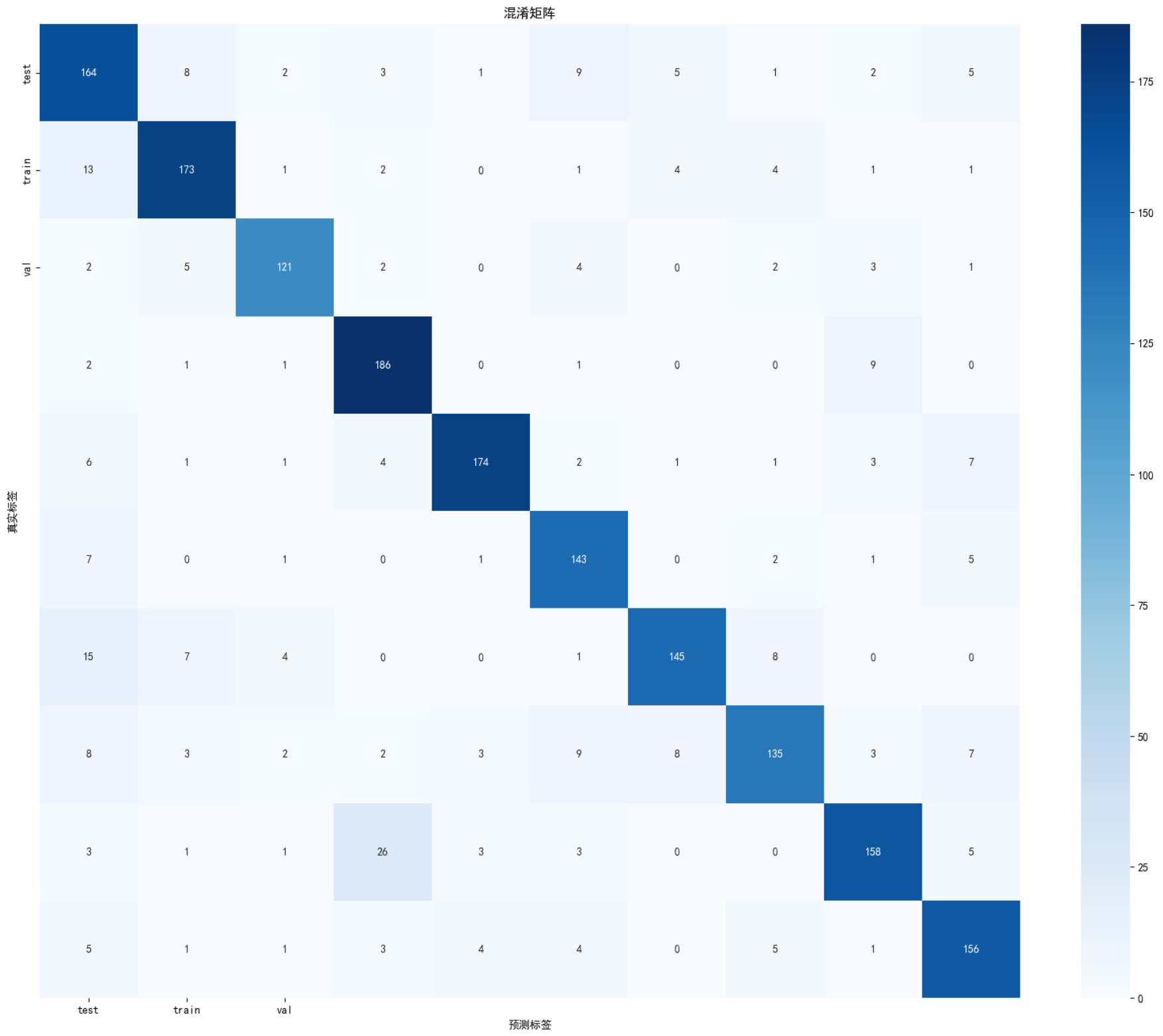


图2.6. 2消融实验6混淆矩阵

5.2.7 结果整理

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **实验配置** | **准确率** | **精确率** | **召回率** | **F1分数** |
| 移除ECAAttention | 83.64% | 83.64% | 83.64% | 83.55% |
| 移除多尺度融合 | 82.01% | 82.42% | 82.01% | 81.92% |
| 替换分类层 | 83.64% | 83.67% | 83.64% | 83.60% |
| 移除ECAAttention和多尺度融合 | 83.64% | 83.83% | 83.64% | 83.57% |
| 移除ECAAttention和替换分类层 | 84.13% | 84.31% | 84.13% | 84.05% |
| **移除多尺度融合和替换分类层** | **84.51%** | **84.92%** | **84.51%** | **84.53%** |
| 注：消融实验6（移除**多尺度融合**和替换**分类层**）表现最佳，准确率达到84.51% | | | | |

5.3 总结

从实验结果可以看出，通过对MobileNetV3的不同改进和消融实验，模型在Animal-10数据集上的分类性能逐步得到优化。基线模型MobileNetV3的表现已经达到82%的准确率，但在进一步优化后，**消融实验6**的配置（移除多尺度融合和替换ArcMarginProduct）取得了最好的效果，最终准确率达到85%。

6. 讨论与最终设计

6.1 实验结果分析

通过对比各个消融实验的性能，我们发现，尽管增加复杂模块（如多尺度融合和ArcMarginProduct）能够在某些任务中提高精度，但在Animal-10数据集上，复杂设计并未带来显著提升，反而可能导致训练的不稳定或**过拟合**。相比之下，简化模型后的性能显著改善，且训练过程更为稳定。这表明该数据集的特征适合通过简化模型来进行分类。

在实验过程中，我们注意到以下几个关键因素对模型性能有显著影响：

* **Dropout层的作用**：在训练过程中加入Dropout层有效地防止了过拟合，尤其是在小数据集上。Dropout通过随机丢弃部分神经元，迫使模型学习到更加稳健的特征表示，从而减少了对训练数据的过度依赖。通过这种**正则化**手段，模型在验证集上的表现得到了显著提高。
* **批量大小与学习率调整**：减少批量大小后，增加了梯度估计的噪声，帮助模型跳出局部最优，从而提升了**泛化能力**。适当的学习率调整也对模型的训练稳定性和最终精度产生了重要影响。在实验中，我们通过调整学习率，确保训练过程更加平稳，并提高了模型的**收敛速度**。
* **数据增广的重要性**：为了提高模型的**鲁棒性**，我们采用了多种数据增广策略（如水平翻转、随机旋转、仿射变换等）。这些增广方法有效地增加了数据的多样性，帮助模型适应输入变换，提高了模型的泛化能力，减少了过拟合的风险

6.2 改进设计

设计灵感来源于知名的 MobileNetV3，并结合了 ECA（Efficient Channel Attention） 注意力机制（消融实验证明改进有效的），以提高特征表达能力和分类性能。该网络旨在在保持高效计算和低参数量的同时，提供优异的分类效果，适合在资源受限的设备上部署。

**a. ECA注意力模块（ECAAttention）**

作用：ECAAttention 模块用于增强网络对重要特征的关注，通过动态调整每个通道的重要性，提升模型的表达能力。

**工作原理：**

1.全局平均池化：将特征图的空间维度（高度和宽度）压缩为一个数值，得到每个通道的全局信息。

2.一维卷积：通过一维卷积捕捉通道之间的相互关系，生成注意力权重。

3.Sigmoid激活：将注意力权重归一化到 [0,1] 之间。

4.特征重标定：将注意力权重与原始特征图相乘，增强重要通道的特征。

**优势：**

高效性：相比传统的注意力机制，ECAAttention 采用一维卷积，减少了计算量和参数数量。

自适应性：能够根据输入特征自适应地调整通道权重，提高模型的灵活性。

**b. 基础卷积模块（conv\_bn）**

**作用：**这是一个基础的卷积模块，结合了卷积层、批归一化（Batch Normalization）和激活函数，用于初步提取特征。

**组成：**

1.卷积层：提取空间特征。

2.批归一化：稳定训练，加速收敛。

3.激活函数：引入非线性，提高模型表达能力。

**优势：**

模块化：将常用的卷积操作、归一化和激活函数组合在一起，便于复用。

高效性：使用轻量级的激活函数 Hardswish，减少计算开销。

**c. 倒残差块（InvertedResidualECA）**

**作用：**倒残差块是网络的核心组成部分，负责深度特征提取，同时保持计算效率。

**工作原理：**

1.扩展：通过1x1卷积将输入通道数扩展，增加特征的表达能力。

2.深度卷积：采用3x3的深度可分离卷积，独立处理每个通道，提取空间特征。

3.注意力模块（可选）：增强重要特征。

4.压缩：通过1x1卷积将通道数压缩回原始维度，减少计算量。

5.残差连接：当输入和输出尺寸相同时，将输入直接加到输出上，促进信息流动和梯度传播。

**优势：**

计算效率高：深度可分离卷积和倒残差结构显著减少计算量和参数数量。

信息保留：残差连接有效防止信息丢失，增强模型的表达能力。

接下来我们小组将按照数据从输入到输出的顺序，逐层解析网络的结构与功能。

**a. 输入层与前置卷积（Stem）**

self.stem = conv\_bn(3, 16, 3, 2)

输入通道数：3（RGB图像）。

输出通道数：16。

卷积核大小：3x3。

步幅：2（图像尺寸减半）。

填充：1（保持卷积后的尺寸）。

**流程：**

1.卷积操作：提取初步特征，同时将图像尺寸从 (H, W) 减半到 (H/2, W/2)。

2.批归一化与激活：稳定特征并引入非线性。

**b. 特征提取模块**

block\_setting = [

# t, c, n, s, use\_eca

(1, 16, 1, 1, False),

(4, 24, 2, 2, False),

(4, 32, 3, 2, False),

(4, 64, 3, 2, True),

(6, 96, 2, 1, True),

(6, 128, 2, 2, True),

]

input\_channel = 16

features = []

for t, c, n, s, eca in block\_setting:

for i in range(n):

stride = s if i == 0 else 1

block = InvertedResidualECA(input\_channel, c, stride, t, use\_eca=eca)

features.append(block)

input\_channel = c

self.features = nn.Sequential(\*features)

**参数解释：**

t（扩展率）：控制输入通道数的扩展倍数。

c（输出通道数）：每组块的输出通道数。

n（块数）：每组块的重复次数。

s（步幅）：控制空间下采样的步幅。

use\_eca：是否在该组块中使用ECA注意力模块。

**流程：**

1.配置解析：根据 block\_setting 列表，逐组创建倒残差块。

2.逐层堆叠：每组块中，第一个块可能进行下采样（步幅2），其余块步幅为1。

3.组合模块：所有块通过 nn.Sequential 组合成一个整体特征提取模块。

**c. 全局平均池化（Global Average Pooling）**

self.avgpool = nn.AdaptiveAvgPool2d(1)

**流程：**

1.池化操作：对每个通道进行全局平均池化，将特征图尺寸变为1x1。

2.输出：得到一个固定长度的特征向量，适用于全连接层处理。

**d. 分类器（Linear Classifier）**

self.classifier = nn.Linear(input\_channel, num\_classes)

输入特征维度：input\_channel（通过特征提取模块的最后一层）。

输出特征维度：num\_classes（分类任务的类别数，如10类动物）。

**流程：**

1.展平操作：将 (batch\_size, channel, 1, 1) 转换为 (batch\_size, channel)。

2.线性变换：通过全连接层将特征向量映射到类别得分。

3.输出：每个类别的得分，用于最终分类。

**设计理念与参考**

**a. 参考经典网络架构**

**MobileNetV3：**

倒残差结构：高效的特征提取与信息保留。

轻量级激活函数（Hardswish）：提供良好的非线性表达能力，同时保持计算效率。

高效注意力机制：结合ECAAttention，提升特征表达能力。

**EfficientNet：**

网络扩展策略：通过均衡调整网络的深度、宽度和分辨率，实现高效的特征提取。

**b. 高效的注意力机制**

ECAAttention 取代传统的 SE 模块，通过一维卷积实现通道间的局部依赖关系捕捉，减少了参数和计算量，同时提升了注意力机制的效果。

**c. 轻量化与高效性**

深度可分离卷积：减少计算量和参数数量。

倒残差结构：有效保留信息，防止梯度消失。

模块化设计：各部分（如卷积模块、注意力模块）高度模块化，便于调整和优化。

**d. 简化分类器设计**

使用标准的线性分类器而非复杂的分类头（如 ArcFace），简化模型结构，降低训练复杂性，同时保持良好的分类性能。

**超参数优化**

经验：

1. **Dropout 层的重要性**：Dropout 是一种正则化技术，通过在训练过程中随机“丢弃”一部分神经元，防止模型对训练数据过拟合。

2. **批量大小减少与学习率调整**：较小的批量大小会增加梯度估计的噪声，有助于模型跳出局部最优，从而潜在地提高泛化能力。

3. **第0类等类别动物准确率较低**：特征空间相似

4. **数据增广的重要性**：数据增广通过对训练数据进行随机变换（如裁剪、颜色扰动等），增加了数据的多样性，帮助模型更好地泛化到未见过的数据。

**（这意味着我们需要重点关注图像裁剪，颜色变换！）**

# 调整后的超参数

**batch\_size = 32 # 增大批量大小**

**learning\_rate = 1e-3 # 调整学习率**

**weight\_decay = 1e-4 # 保持权重衰减**

# 使用 AdamW 优化器

**optimizer = optim.AdamW(model.parameters(), lr=learning\_rate, weight\_decay=weight\_decay)**

# 使用 Cosine Annealing 学习率调度器

**scheduler = optim.lr\_scheduler.CosineAnnealingLR(optimizer, T\_max=10)**

# 调整训练轮次

**epochs = 100**

**解释：**

AdamW：相比 Adam，AdamW 更好地处理权重衰减，有助于防止过拟合。

Cosine Annealing：学习率呈余弦曲线下降，有助于模型在训练后期找到更好的局部最优解。

批量大小增加：利用更大的批量大小，提升训练效率和稳定性。

**结果：**

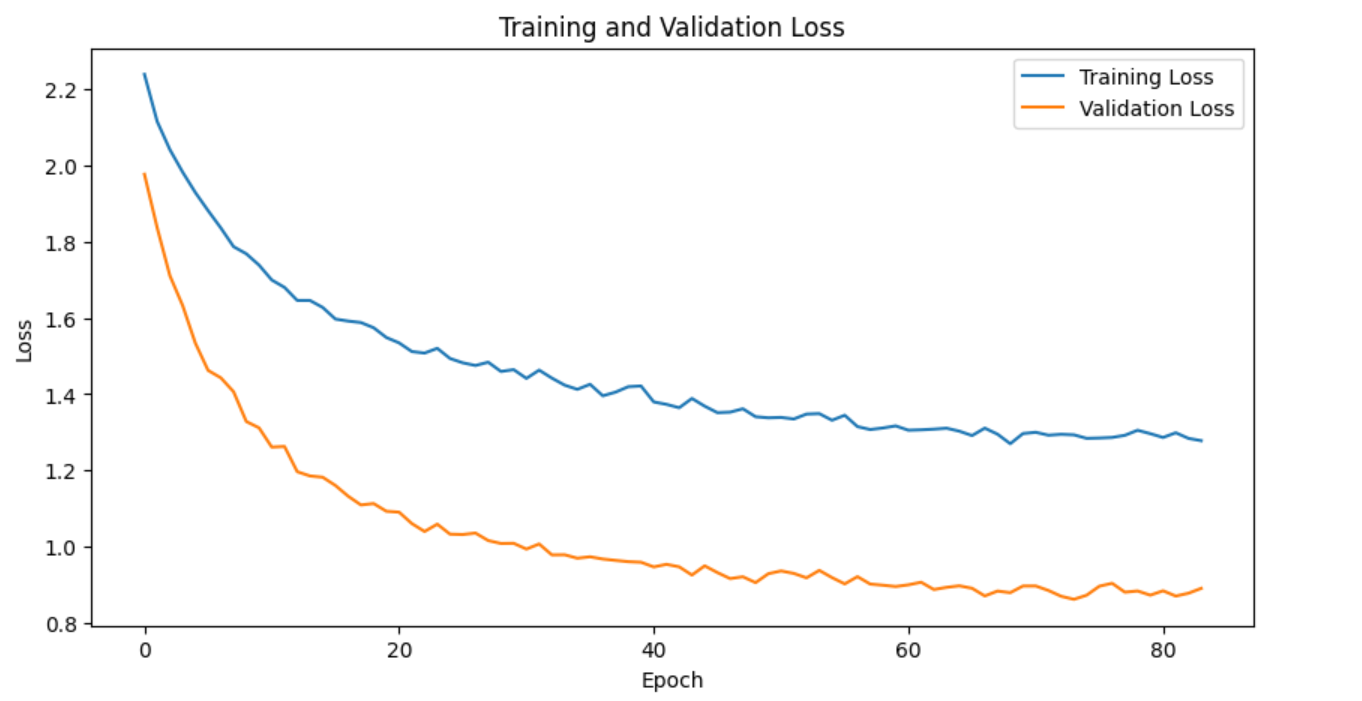


图6.2.1 训练与验证误差曲线

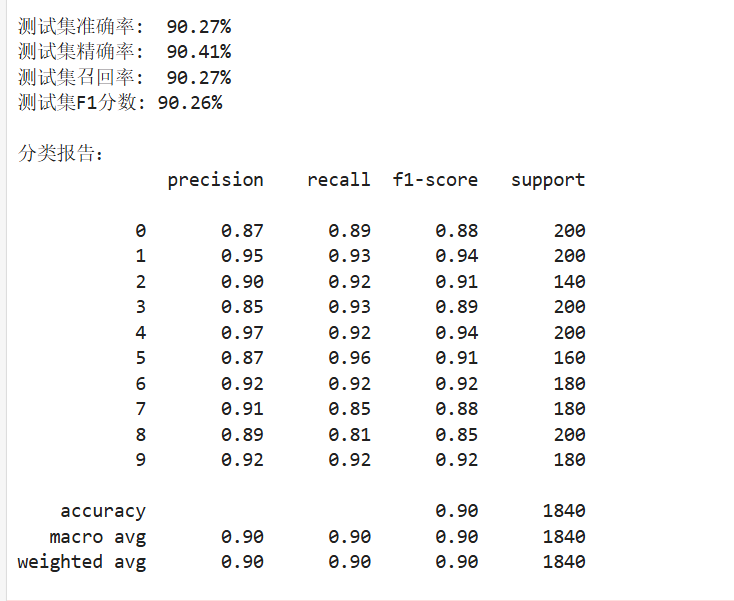


图6.2.2 准确率报告

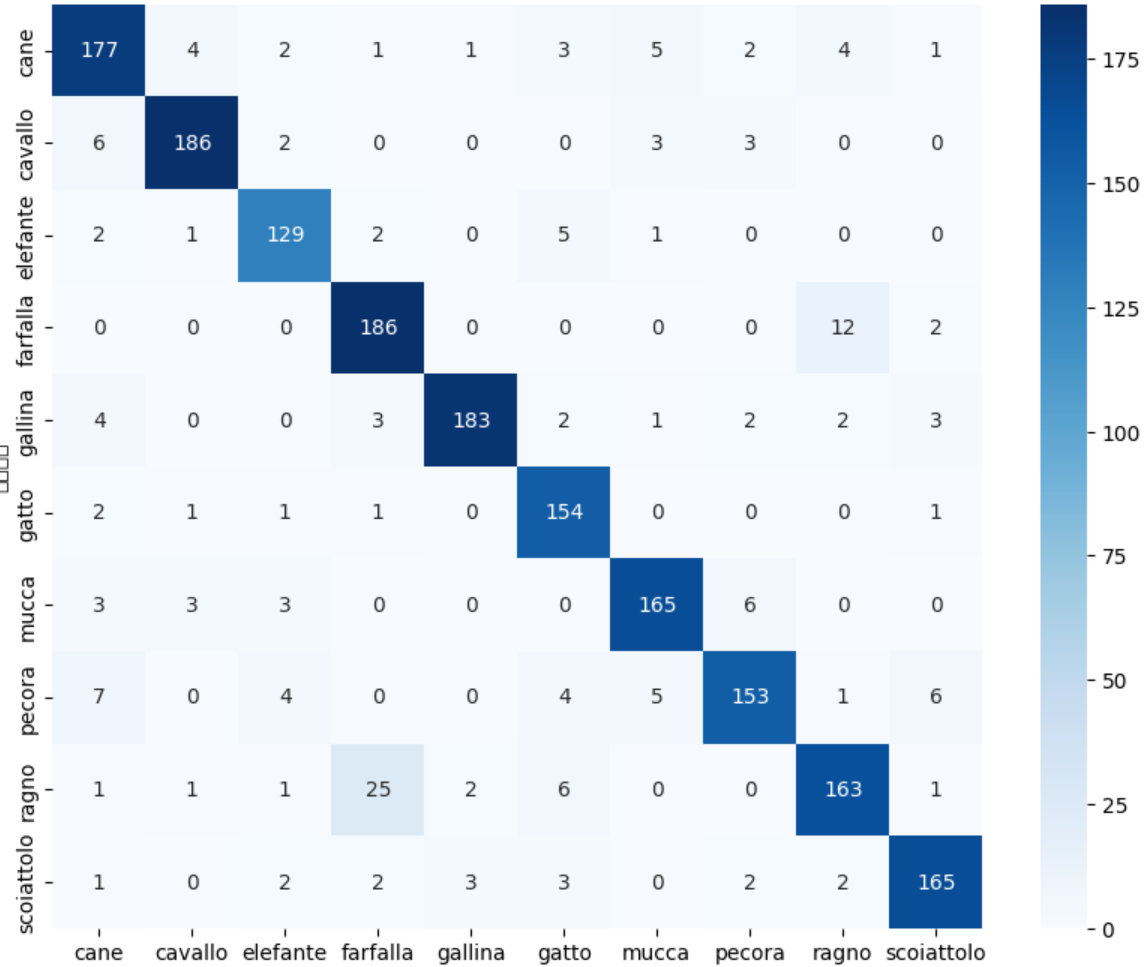


图6.2.3 混淆矩阵

7. 结论

本实验通过对**MobileNetV3**模型进行改进和一系列**消融实验**，进一步改进与设计，最终确定了最佳的模型配置。这一改进使得模型的分类准确率达到了**90%**，相较于基线模型的82.66%有较多的提升。**此外，消融实验结果表明，分类层等并未显著提高性能，反而简化模型结构在此任务中表现更为出色。**

因此，本实验**成功优化**了MobileNetV3架构，在Animal-10数据集上取得了更为稳定和高效的分类表现。

8. 研究意义

本实验展示了如何通过合理的**网络架构改进**和**消融实验**，优化深度学习图像分类模型的性能，尤其在**动物图像分类**任务中取得了较好的结果。实验结果为图像分类任务中**模型简化**提供了有价值的参考，对于如何平衡**模型复杂度**与**性能**，有重要的现实意义。此外，针对**轻量级模型**的优化方案也能为其他类似的图像分类任务提供借鉴。

9. 参考文献

[1] Wang, Qilong, et al. "ECA-Net: Efficient channel attention for deep convolutional neural networks." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.

[2] Sinha, Debjyoti, and Mohamed El-Sharkawy. "Thin mobilenet: An enhanced mobilenet architecture." 2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON). IEEE, 2019.

[3] Koonce, Brett, and Brett Koonce. "ResNet 50." Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization (2021): 63-72.

[4] 动手学深度学习[2021]，李沐著