

**学科实践二**

**实验报告**

**类 别： 多模型比较的人脸表情识别**

**选 题： 表情识别**

**专 业： 计算机科学与技术**

**组 长： 鲍仁旭**

**组 员： 林宏津 许太行**

**2025年 6 月 8 日**

**一、课题任务描述**

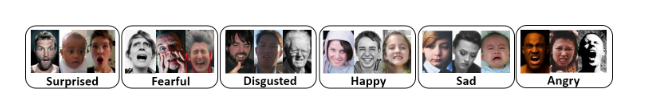
随着人工智能和计算机视觉技术的快速发展，人脸表情识别作为其中一个重要的研究方向，已广泛应用于智能监控、人机交互、心理健康评估等领域。准确识别人脸的情绪状态对于提升系统的智能化水平和用户体验具有重要意义。

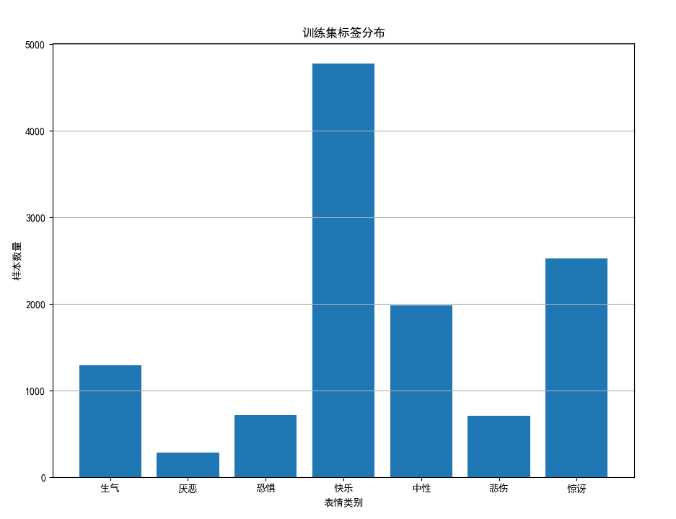
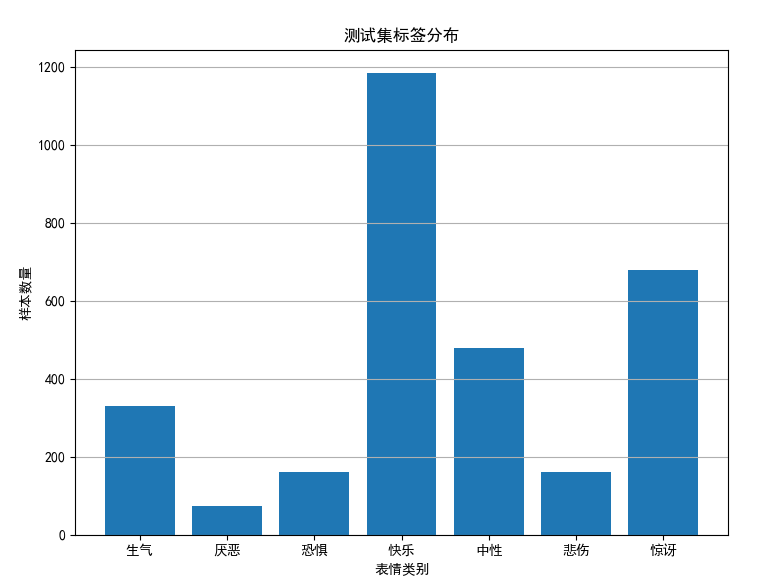
传统的人脸表情识别方法多依赖于手工设计的特征提取，受限于特征表达能力，难以适应复杂多变的现实场景。近年来，深度学习技术，尤其是卷积神经网络（CNN）和注意力机制的发展，大幅提升了表情识别的准确率和鲁棒性。基于深度学习的方法能够自动学习有效的表情特征，适应光照、姿态和表情细微变化，提高识别效果。

本项目基于**深度残差网络（ResNet）、注意力机制设计的 DAN 和GoogleNet模型**，结合多种超参数调优和数据增强技术，系统性地实现并对比分析了三种主流深度学习模型。

**二、数据集处理**

**1、数据规模**

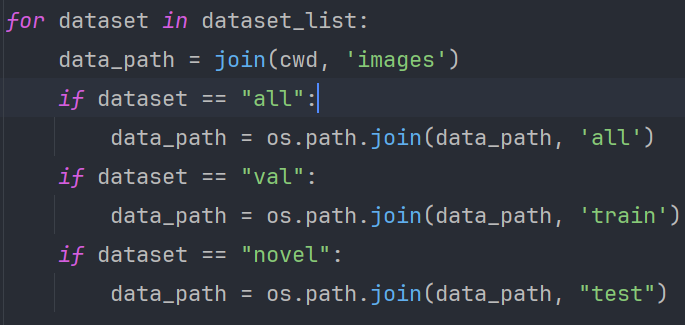
本项目采用了人脸表情数据集 RAF-DB（Real-world Affective Faces Database）作为训练和测试的主要数据来源。RAF-DB 数据集包含29670张带有多种自然表情的人脸图像，涵盖7类基本情绪类别（如愤怒、厌恶、恐惧、高兴、伤心、惊讶和中性），图像采集自多种自然场景，具备较高的多样性和复杂度。

 训练集及测试集中各个标签样本量分布如下，

**2、数据预处理**

**（1）索引整理**

本项目对原始图像数据集中的目录结构进行了统一整理，并使用 Python 脚本自动生成标准化的索引文件。实现了类别与标签的自动映射，还通过乱序打散的方式增强了后续训练的鲁棒性。

****

**（2）数据调整及数据增强**

为了提高模型泛化能力和鲁棒性小组对所有图像进行统一处理，所有图像统一进行了尺寸调整，缩放至48×48或224×224像素的RGB三通道格式。图像数据被转换为张量（Tensor），并基于ImageNet预训练模型的通道均值和标准差进行标准化处理，以加快模型收敛速度和提升泛化能力。

此外为了适应真实世界情况，缓解过拟合的问题，训练阶段对数据进行了数据增强操作，包括：

随机水平翻转：以50%的概率进行左右翻转，增强对称表情识别能力。

随机旋转：图像随机旋转±10度，模拟不同拍摄角度。

颜色抖动：调整图像的亮度和对比度，适应光照变化。

**训练结果表明数据增强能够有效缓解过拟合增强模型泛化能力。**

**三、技术路线**

**1、核心算法**

本次选题中主要使用的是ResNet50、ResNet18、基于inception块的GoogleNet及基于注意力机制设计的DAN（Dual Attention Network）网络构建模型。本文使用到了当前主流深度学习框架Pytorch及torchvison。

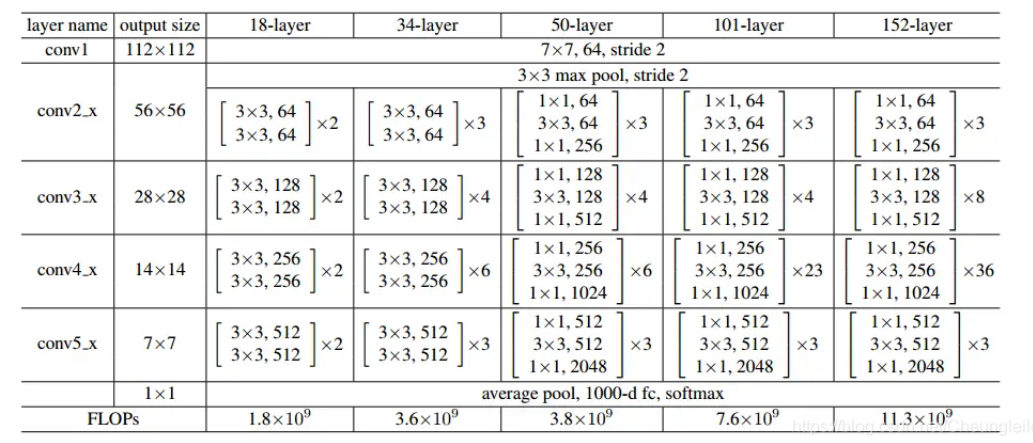
**ResNet**网络是通过引入了恒等映射来计算残差，以此缓解梯度消失/爆炸的现象，来解决神经网络隐藏层过多时，网络的准确度达到饱和而产生非过拟合的急剧退化现象。

**DAN**网络融合了空间注意力和通道注意力机制，通过动态调整特征图的空间和通道权重，增强模型对关键区域和特征的感知能力。DAN基于ResNet的骨干网络，提升了对人脸微表情的捕捉能力

**GoogleNet**是一种基于Inception模块的深度卷积神经网络，通过在同一层并行使用多种大小卷积核和池化操作，来捕获多尺度特征信息。GoogleNet结构紧凑，参数较少，但性能优异，常用于复杂图像分类任务。

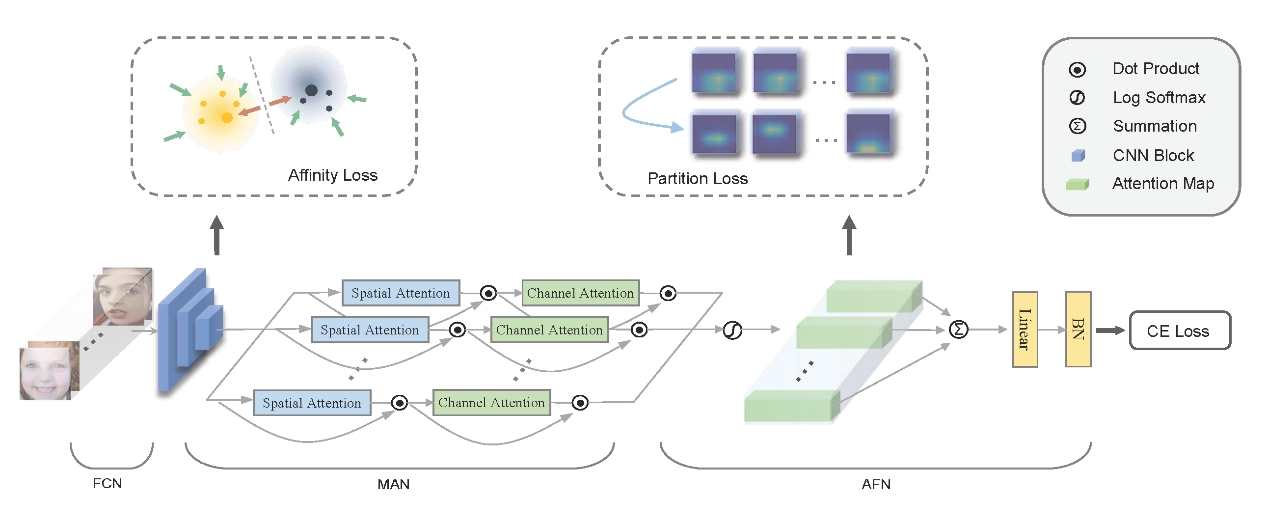
**2、ResNet系列网络**

ResNet系列网络主要包括卷积层、池化层、全连接层和残差模块，通过引入恒等映射来缓解梯度消失的问题。ResNet系列的残差连接使得模型的深度不再是限制。本项目中使用ResNet50\_vd实现对应功能。ResNet系列的结构网络图如图1所示。残差网络由残差块、瓶颈残差块组成



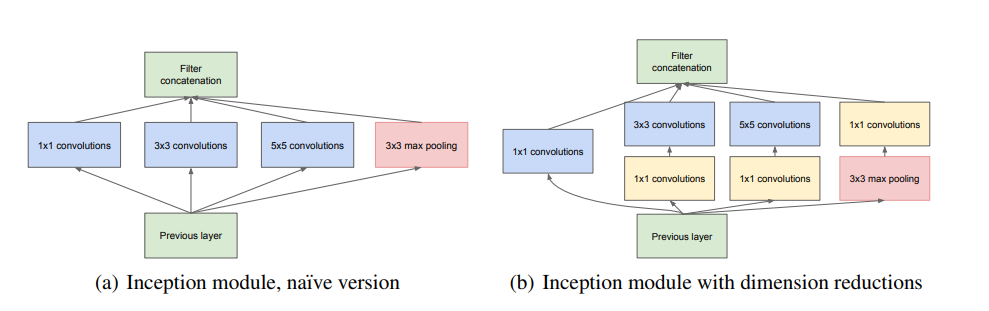
**3、基于ResNet骨干网络的DAN网络**

DAN模型由特征聚类网络（FCN）、多头注意力网络（MAN）和注意力融合网络（AFN）三部分组成，首先FCN提取具有类别区分能力的基础特征，MAN通过多个轻量级注意力头依次运用不同大小卷积核的空间注意力和通道注意力单元，捕捉并增强多个面部表情区域的注意力图，最后AFN对这些注意力图进行训练和融合，实现对表情类别的准确预测，从而高效地学习和整合多区域的表情特征[1]。

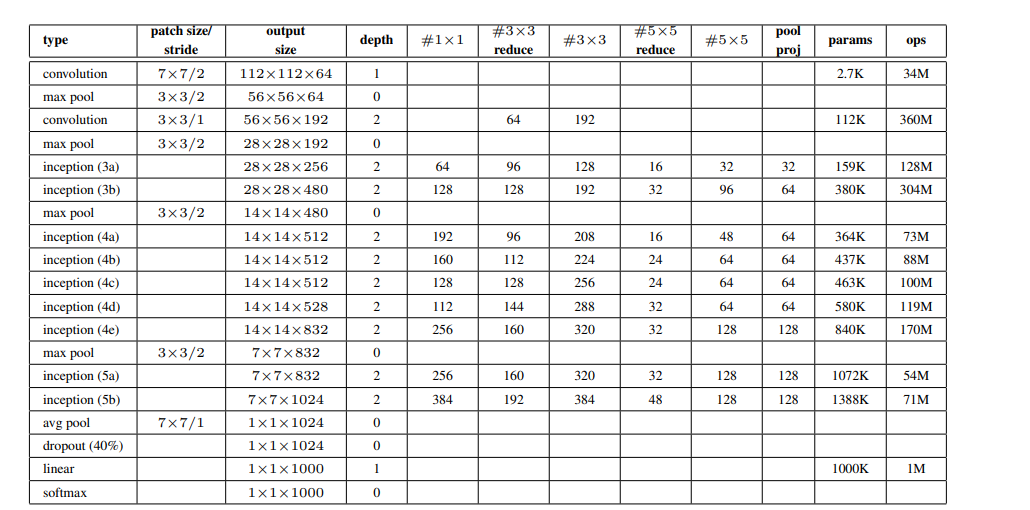


**4、基于Inception块的GoogleNet**

GoogleNet采取Inception提取图像特征，该模块通过并行使用不同尺寸的卷积核和池化操作，能够同时捕捉多尺度的特征，从而提高网络的表达能力和计算效率。Inception块的设计如下



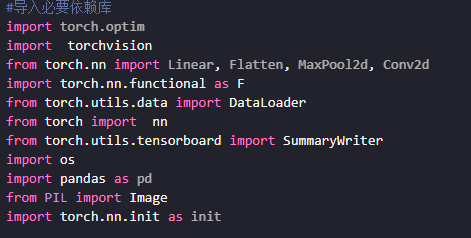
通过1x1、3x3、5x5的卷积获得小中大三种感受野的数据特征，此外3x3的最大池化提取局部不变特征，融合多尺度信息，增强了网络的表达能力和鲁棒性。GoogleNet的具体层设计如下：[2]

****

**3、技术路线**

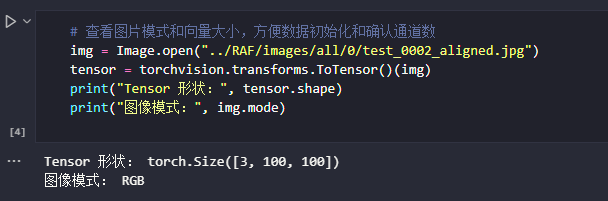
**（1）导入必要依赖库**

本项目基于pytorch，在开始之前需要导入基础库包括 PyTorch 中的优化器（torch.optim）、计算机视觉相关工具（torchvision），以及神经网络相关模块如线性层（Linear）、卷积层（Conv2d）、最大池化层（MaxPool2d）等。同时引入数据加载工具（DataLoader）、张量操作函数库（torch.nn.functional）、TensorBoard 可视化工具（SummaryWriter）、图像处理库 PIL 的 Image 模块，以及数据处理库 pandas。这些依赖为后续模型构建、训练、数据处理和可视化分析提供基础支持。

****

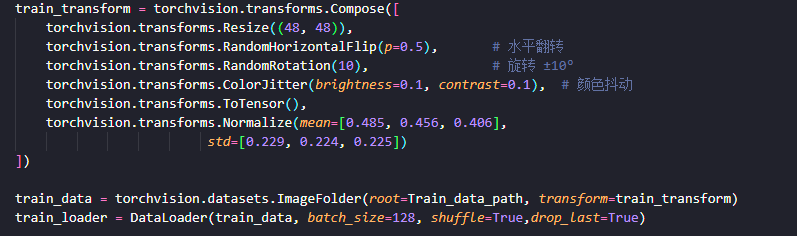
**（2）查看图片模式及向量大小**

抽取一张图片查看数据集中的图片的通道数及向量大小。

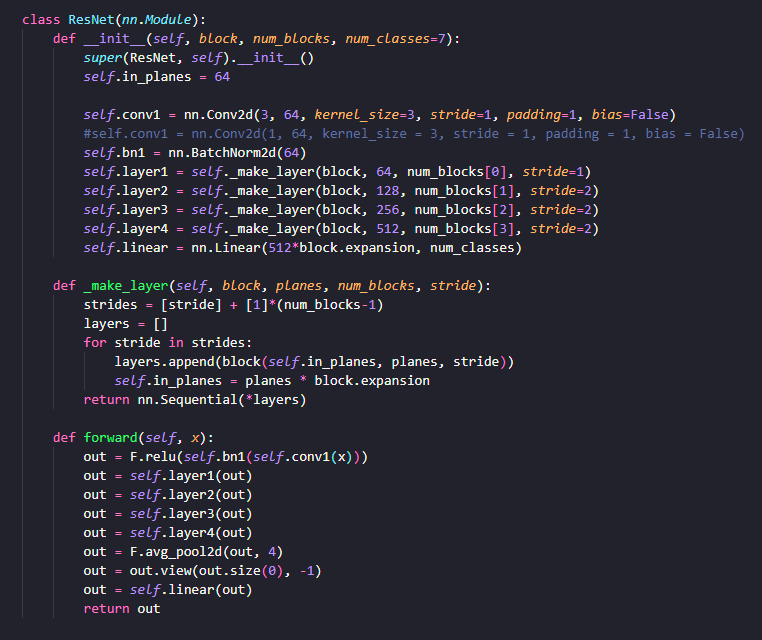


**（3）编写Transform函数并打包数据集**

完成对Transform函数的初始化设置包括统一大小、数据增强（旋转、水平翻转、颜色抖动）和归一化操作。



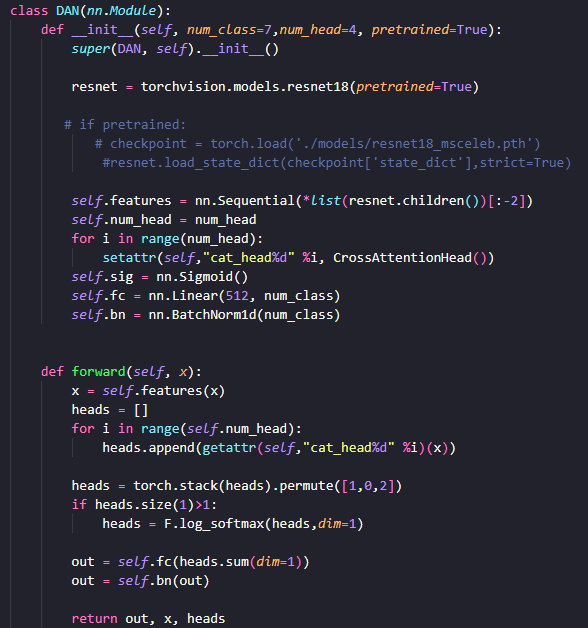
**（4）实现残差网络**

由于篇幅限制，具体的瓶颈块（Bottleneck）和基础块（BasicBlock）的详细定义未在此展开，读者可参考完整代码文件以了解其具体结构和实现细节。

**（5）实现DAN网络**

模型以预训练的 ResNet18 为基础特征提取骨干网络，去除其最后两个层，提取输入图像的基础特征。然后通过动态创建多个注意力头，分别对特征进行多头注意力计算。前向传播时，模型依次通过骨干网络提取特征，再分别调用各注意力头生成多个注意力特征，最后将这些特征按维度融合，通过全连接层和批归一化层输出最终的分类结果。

由于篇幅限制不展示CrossAttentionHead等计算注意力权重的函数，只展示了DAN的主体部分。



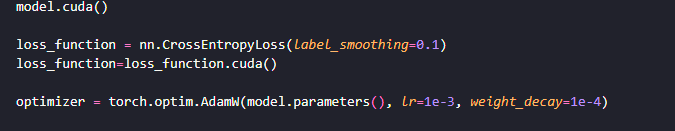
**（5）GoogleNet**

由于 GoogLeNet 模型结构复杂，包含多分支的 Inception 模块和多尺度卷积设计，手工实现难度较大且容易引入错误。为了保证模型的准确性与训练效率，本文直接调用了 PyTorch 框架中已实现并经过充分优化的 googlenet 函数，基于该预训练模型进行后续的训练与微调。

****

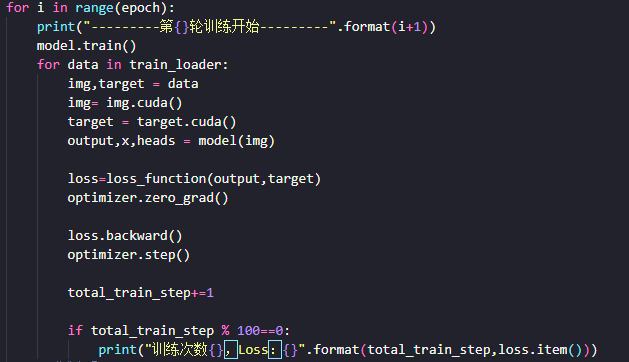
**（6）超参数设置**

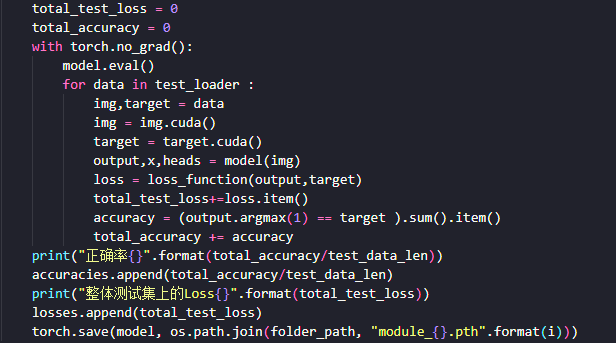
本文利用.cuda完成GPU加速，并以AdamW作为优化器，交叉熵作为损失函数。



**（7）模型训练**

本文基于pytorch，搭建模型训练、计算损失函数、梯度清零、反向传播、向前传播的深度学习训练流程。



并在每一轮结束后在测试集上测试模型参数输出在测试集上正确率和损失。

**（8）正确率函数、损失函数和困惑矩阵（confusion matrix）绘制**

通过第三方依赖库matlablib绘制正确率函数和损失函数图像，方便训练者观察是否出现梯度消失。

****

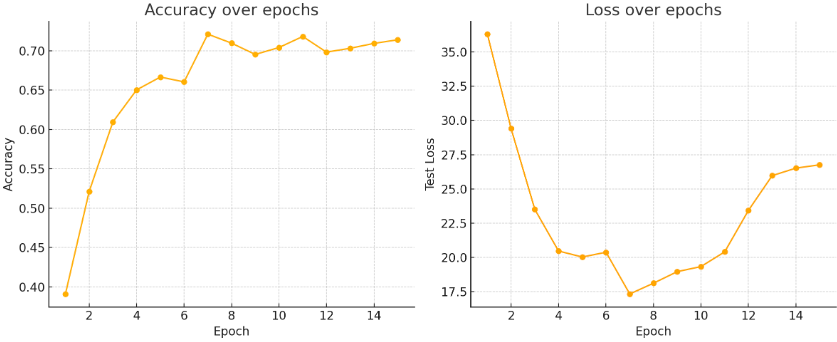
使用第三方库sklearn及seaborn绘制**困惑矩阵**方便使用者对模型训练状况进行分析和调优。

**四、结果的比较分析及参数调优**

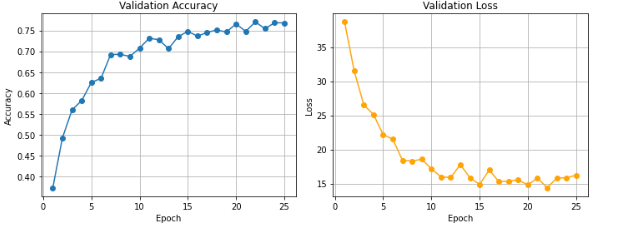
**1、各模型结果及其比较分析**

1. **ResNet50（不含数据增强）**

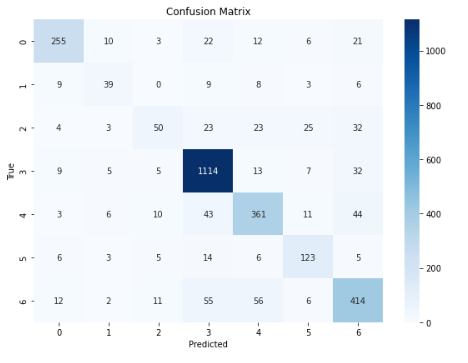
本文将ResNet50的块设置为[3,4,5,3]在没有数据增强的训练下8轮之后梯度不再下降，损失反倒变高，可能出现了较为严重的过拟合。且在这种情况下准确率最高为72%并不理想故本文采取数据增强来优化训练。

****

1. **ResNet50（含数据增强）**

****本文通过颜色抖动、水平翻转、旋转等方式来进行数据增强，来增强模型的泛化能力并避免过拟合。由下图可知，模型具有良好的损失下降趋势，说明拟合效果较好。且在这种情况下准确率有提升最好的准确率为76%。

第3类（快乐）预测非常准确，正确分类了1114个样本，对角线数字最大。第6类（惊讶）也有较好的分类效果，正确分类了414个样本。

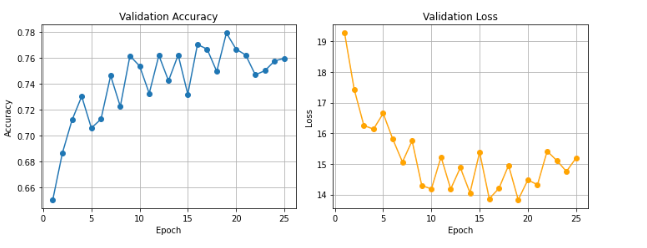
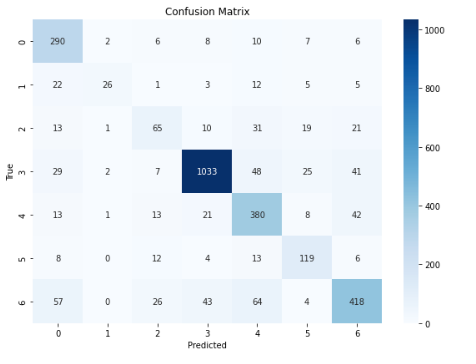
****其他类别的正确分类数相对较少，且误分类较多，例如第2类（恐惧）和第4类（中性）之间存在较多混淆。通过输出困惑矩阵可以发现从图中可以看出，模型对“快乐”和“惊讶”等类别的识别效果较好，但在“恐惧”、“中性”等相似类别之间存在较多混淆，表现出一定的识别困难。

**（3）DAN（含数据增强）**

为了进一步提高分类准确率及模型的泛化能力和鲁棒性，本文采取了基于ResNet预训练结果，加入注意力机制，使模型能够聚焦于关键的表情区域，从而更有效地提取判别特征，提升整体识别性能,提升准确率。

从训练结果可知，由于基于ResNet的预训练结果，DAN模型能再较短的时间内达到60%的正确率，并能在极短的时间内展现出快速收敛的趋势。左图显示验证准确率在前几个 epoch 内快速上升，随后在 0.74 至 0.78 之间小幅波动，说明模型逐步趋于稳定。右图的验证损失则呈现明显下降趋势，在约第 15 个 epoch 附近达到最低值，之后虽有一定波动，但整体保持在较低水平。这说明模型不仅具备较强的拟合能力，同时在验证集上也具有良好的泛化性能。

此模型下最好准确率为78.68%

**** 从下中混淆矩阵可以看出，模型在大多数表情类别上的分类性能有所提升，尤其在“快乐”（标签3）和“惊讶”（标签6）类别上表现突出，分别正确分类了1033和418个样本，显示出模型对这两类特征的高度敏感性。相比之下，“恐惧”（标签2）和“厌恶”（标签1）等类别仍存在一定程度的误分类，易被混淆为“中性”或“生气”等类别，说明模型对这些情绪的判别边界尚不够清晰。整体来看，该混淆矩阵反映了模型在强化表达性强、特征明显的表情识别方面较为可靠，但在处理模糊或相似表情时仍有优化空间。

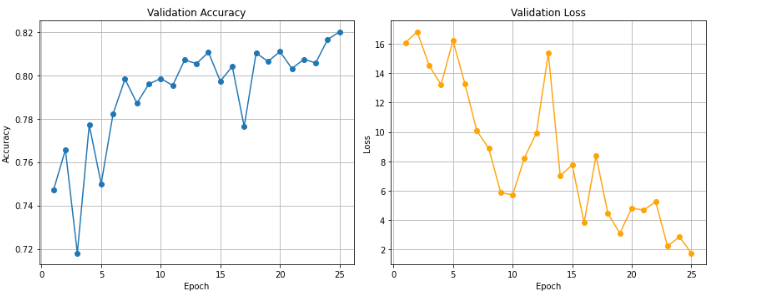
**（4）GoogleNet（含数据增强）**

为了学习更多优秀算法，本文尝试了GoogLeNet算法。GoogLeNet基于size为224x224的3通道图像。故先通过reshape函数对图像进行预处理。

从准确率曲线可以看出，模型在前几轮训练后准确率迅速提升，从约 0.72 提高至 0.80 以上，说明模型在短时间内完成了有效学习。随后准确率整体保持稳定，并在第 25 轮达到最高值 0.82，表明模型具备较强的泛化能力。

验证损失曲线整体呈下降趋势，表明模型在训练过程中持续优化，能够有效降低在验证集上的预测误差。虽然中间几轮存在一定波动，但整体收敛效果良好，最终损失降至约 1.76，进一步说明模型稳定性较高。

本模型下最好准确率为82%。



**（5）结论**

本环节设置了相同的batch\_size、lr、optimizer，可以开发现数据增强对于模型训练具有较大的提升。注意力网络对于图像识别的优势略低于多维特征提取的GoogLeNet。

| **模型名称** | **最佳准确率** | **备注说明** |
| --- | --- | --- |
| ResNet50（无增强） | 72.09% | 无数据增强，训练8轮后出现过拟合，损失反弹 |
| ResNet50（数据增强） | 76.79% | 使用颜色抖动、水平翻转、旋转等方式提升泛化能力 |
| DAN（数据增强） | 78.68% | 基于ResNet，融合注意力机制聚焦关键区域 |
| GoogLeNet（数据增强） | 82.01% | 多尺度Inception结构，收敛快，准确率提升明显 |

对Res及DAN的结果分析：

|  |  |  |
| --- | --- | --- |
| **表情类别** | **DAN平均准确率 / %** | **ResNet平均准确率 / %** |
| **生气** | **88.14** | **77.51** |
| **恐惧** | **35.32** | **52.7** |
| **厌恶** | **40.68** | **31.25** |
| **快乐** | **87.18** | **94.01** |
| **悲伤** | **79.52** | **75.52** |
| **愤怒** | **72.46** | **78.85** |
| **中性** | **68.3** | **74.46** |

由表可知，模型在“生气”和“快乐”这两类表情上的识别效果较为理想，分类准确率相对较高，说明其在处理表达特征明显的情绪时具有较强的判别能力。然而，对于“厌恶”和“恐惧”等相对模糊或特征不显著的类别，模型的识别精度明显偏低，表明其在情绪边界不清晰时仍存在一定的识别困难。

筛选数据集可知 “厌恶”表情的特征并不十分明显，难以与其他相似情绪明显区分。从图中展示的几张面部图像可见，恐惧与厌恶表情都表现为面部肌肉的变化上存在较大重叠，表现出较为模糊和不确定的情绪特征。

且从面部关键点（landmark）角度，这两类表情在关键点的位移和角度变化也较为相近，都体现为眉毛皱褶弯曲、嘴唇出现随机位移。

|  |  |  |
| --- | --- | --- |
| 恐惧 | 厌恶 | 厌恶 |
|  |  |  |
| 恐惧 | 惊吓 | 惊吓 |
|  |  |  |

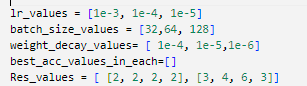
因此，在恐惧、厌恶、惊吓这类没有明显特征的表情分析上模型的精确性仍有欠缺，而对于如开心（均表现为嘴唇移动与眼角变化）、愤怒（landmark向中心偏移且面部颜色容易发生变化）等具有明显表情特征的识别较高。

此外部分表情也会出现“情绪混合”的情况，如Rohith等人在Facial Expression Recognition using Facial Landmarks[3]的研究中分析的，部分表情的混合导致模型出现误判，如下图的表情就混合了恐惧和惊吓，导致模型难以给出合理的解释和正确的分类。



**2、基于ResNet的参数调优**

1. **参数调优设计**

****本文通过在ResNet18（block设计为[2,2,2,2]）及ResNet50（block设计为[3,4,6,3]）上测试不同的学习率、学习策略、batch\_sized进行参数调优。由于在上一个环节中所有模型都呈现优秀且明显的明显的额梯度下降趋势，所以本文不再对epoch进行调整，统一设置为30。

由于有54次循环，预计训练时间为

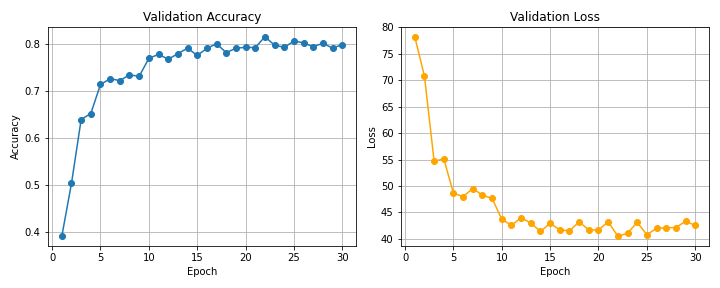
1. **结果**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| groupNumber | Block\_value | Learning Rate | Weight Decay | Batch Size | Epochs | Best Accuracy |
| 1 | [2, 2, 2, 2] | 0.001 | 0.0001 | 32 | 30 | 0.813559 |
| 2 | [2, 2, 2, 2] | 0.001 | 0.0001 | 64 | 30 | 0.814537 |
| 3 | [2, 2, 2, 2] | 0.001 | 0.0001 | 128 | 30 | 0.792373 |
| 4 | [2, 2, 2, 2] | 0.001 | 0.00001 | 32 | 30 | 0.810626 |
| 5 | [2, 2, 2, 2] | 0.001 | 0.00001 | 64 | 30 | 0.79824 |
| 6 | [2, 2, 2, 2] | 0.001 | 0.00001 | 128 | 30 | 0.778683 |
| 7 | [2, 2, 2, 2] | 0.001 | 0.000001 | 32 | 30 | 0.814211 |
| 8 | [2, 2, 2, 2] | 0.001 | 0.000001 | 64 | 30 | 0.80867 |
| 9 | [2, 2, 2, 2] | 0.001 | 0.000001 | 128 | 30 | 0.783572 |
| 10 | [2, 2, 2, 2] | 0.0001 | 0.0001 | 32 | 30 | 0.800196 |
| 11 | [2, 2, 2, 2] | 0.0001 | 0.0001 | 64 | 30 | 0.784876 |
| 12 | [2, 2, 2, 2] | 0.0001 | 0.0001 | 128 | 30 | 0.758149 |
| 13 | [2, 2, 2, 2] | 0.0001 | 0.00001 | 32 | 30 | 0.804107 |
| 14 | [2, 2, 2, 2] | 0.0001 | 0.00001 | 64 | 30 | 0.779661 |
| 15 | [2, 2, 2, 2] | 0.0001 | 0.00001 | 128 | 30 | 0.753911 |
| 16 | [2, 2, 2, 2] | 0.0001 | 0.000001 | 32 | 30 | 0.803129 |
| 17 | [2, 2, 2, 2] | 0.0001 | 0.000001 | 64 | 30 | 0.782595 |
| 18 | [2, 2, 2, 2] | 0.0001 | 0.000001 | 128 | 30 | 0.747066 |
| 19 | [2, 2, 2, 2] | 0.00001 | 0.0001 | 32 | 30 | 0.719687 |
| 20 | [2, 2, 2, 2] | 0.00001 | 0.0001 | 64 | 30 | 0.699804 |
| 21 | [2, 2, 2, 2] | 0.00001 | 0.0001 | 128 | 30 | 0.679596 |
| 22 | [2, 2, 2, 2] | 0.00001 | 0.00001 | 32 | 30 | 0.725554 |
| 23 | [2, 2, 2, 2] | 0.00001 | 0.00001 | 64 | 30 | 0.698501 |
| 24 | [2, 2, 2, 2] | 0.00001 | 0.00001 | 128 | 30 | 0.672751 |
| 25 | [2, 2, 2, 2] | 0.00001 | 0.000001 | 32 | 30 | 0.725554 |
| 26 | [2, 2, 2, 2] | 0.00001 | 0.000001 | 64 | 30 | 0.711864 |
| 27 | [2, 2, 2, 2] | 0.00001 | 0.000001 | 128 | 30 | 0.677314 |
| 28 | [3, 4, 6, 3] | 0.001 | 0.0001 | 32 | 30 | 0.808018 |
| 29 | [3, 4, 6, 3] | 0.001 | 0.0001 | 64 | 30 | 0.800522 |
| 30 | [3, 4, 6, 3] | 0.001 | 0.0001 | 128 | 30 | 0.778357 |
| 31 | [3, 4, 6, 3] | 0.001 | 0.00001 | 32 | 30 | 0.802477 |
| 32 | [3, 4, 6, 3] | 0.001 | 0.00001 | 64 | 30 | 0.79661 |
| 33 | [3, 4, 6, 3] | 0.001 | 0.00001 | 128 | 30 | 0.772164 |
| 34 | [3, 4, 6, 3] | 0.001 | 0.000001 | 32 | 30 | 0.795306 |
| 35 | [3, 4, 6, 3] | 0.001 | 0.000001 | 64 | 30 | 0.794329 |
| 36 | [3, 4, 6, 3] | 0.001 | 0.000001 | 128 | 30 | 0.767927 |
| 37 | [3, 4, 6, 3] | 0.0001 | 0.0001 | 32 | 30 | 0.795306 |
| 38 | [3, 4, 6, 3] | 0.0001 | 0.0001 | 64 | 30 | 0.78455 |
| 39 | [3, 4, 6, 3] | 0.0001 | 0.0001 | 128 | 30 | 0.750326 |
| 40 | [3, 4, 6, 3] | 0.0001 | 0.00001 | 32 | 30 | 0.797588 |
| 41 | [3, 4, 6, 3] | 0.0001 | 0.00001 | 64 | 30 | 0.77575 |
| 42 | [3, 4, 6, 3] | 0.0001 | 0.00001 | 128 | 30 | 0.755215 |
| 43 | [3, 4, 6, 3] | 0.0001 | 0.000001 | 32 | 30 | 0.800847 |
| 44 | [3, 4, 6, 3] | 0.0001 | 0.000001 | 64 | 30 | 0.777053 |
| 45 | [3, 4, 6, 3] | 0.0001 | 0.000001 | 128 | 30 | 0.747718 |
| 46 | [3, 4, 6, 3] | 0.00001 | 0.0001 | 32 | 30 | 0.724576 |
| 47 | [3, 4, 6, 3] | 0.00001 | 0.0001 | 64 | 30 | 0.700782 |
| 48 | [3, 4, 6, 3] | 0.00001 | 0.0001 | 128 | 30 | 0.675033 |
| 49 | [3, 4, 6, 3] | 0.00001 | 0.00001 | 32 | 30 | 0.735007 |
| 50 | [3, 4, 6, 3] | 0.00001 | 0.00001 | 64 | 30 | 0.700782 |
| 51 | [3, 4, 6, 3] | 0.00001 | 0.00001 | 128 | 30 | 0.677314 |
| 52 | [3, 4, 6, 3] | 0.00001 | 0.000001 | 32 | 30 | 0.731095 |
| 53 | [3, 4, 6, 3] | 0.00001 | 0.000001 | 64 | 30 | 0.700456 |
| 54 | [3, 4, 6, 3] | 0.00001 | 0.000001 | 128 | 30 | 0.675033 |

1. **结果分析**
2. **性能最佳**

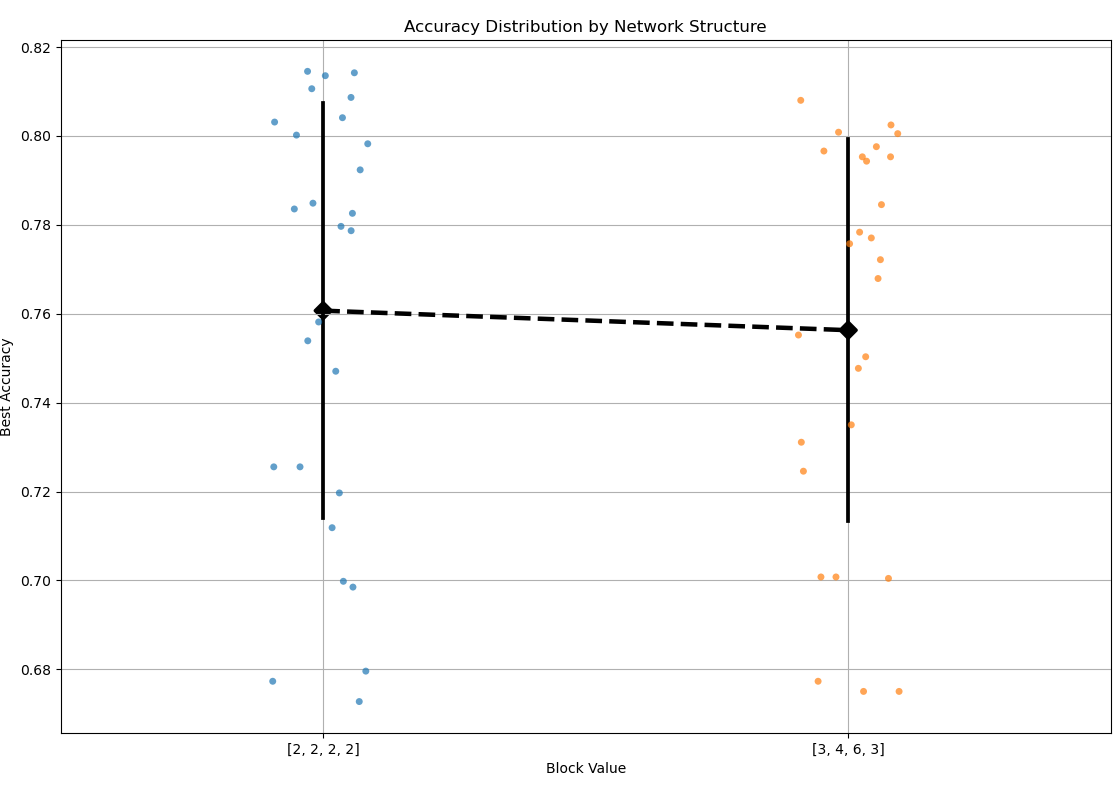
在本数据集上，性能最佳的为组别2，最佳准确率为81.4%，其配置为：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| groupNumber | Block\_value | Learning Rate | Weight Decay | Batch Size | Epochs | Best Accuracy |
| 2 | [2, 2, 2, 2] | 0.001 | 0.0001 | 64 | 30 | 0.814537 |

其正确率图像和损失函数图像如下图所示：

可以看出，在该参数配置下模型具有良好的收敛效果，正确率维持在80%左右。后续研究可以在此基础上进行微调。

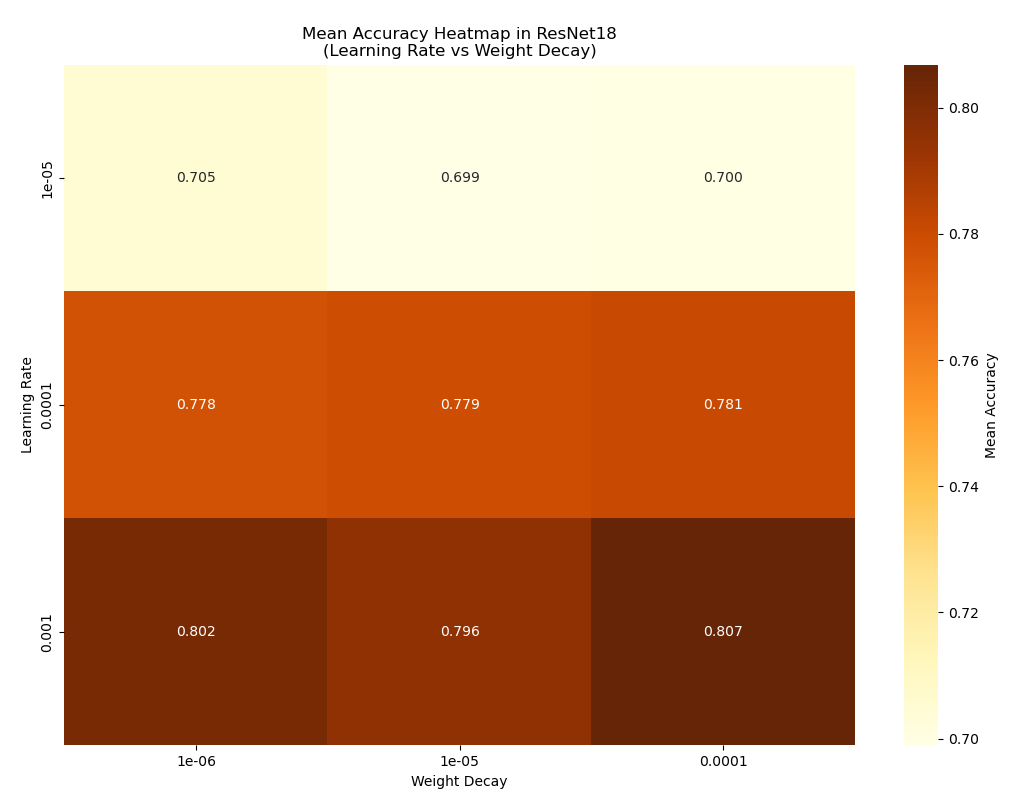
1. **网络结构在本数据集上的表现**

通过计算不同的网络结构ResNet18[2,2,2,2]及ResNet50[3,4,6,3] 的准确率分布，绘制准确率、标准差表及分布图。

| **网络结构** | **平均准确率（Mean）** | **标准差（Std）** |
| --- | --- | --- |
| ResNet18 | 0.7607 | 0.0476 |
| ResNet50 | 0.7563 | 0.0438 |

在 RAF-DB 表情识别任务中，浅层结构反而取得了略高的平均准确率（0.7607%），这说明深层网络的表达能力没有被充分利用，甚至可能导致过拟合或训练困难的问题。但是浅层网络对超参数的反应较为敏感，表现为标准差较大，准确率波动明显。但是深层网络由于层数较多，对不同配置有更强的“缓冲”，具有良好的稳定性。

1. **学习率对性能的影响**

****在本实验中，我们分别设置了 0.001、0.0001 与 0.00001 三个数量级的学习率，观察其在不同网络结构与 weight-decay下的表现。

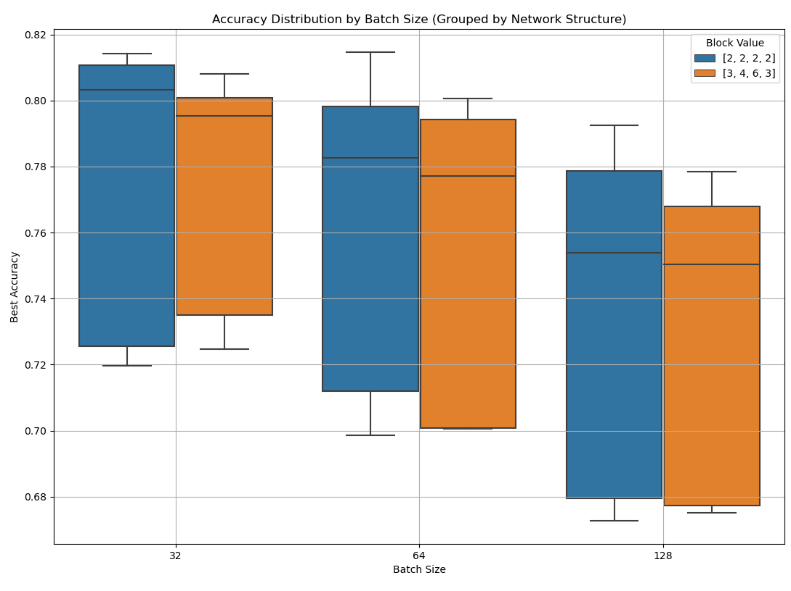
由上图可知，横向（weight decay）变化时，准确率浮动较小，纵向（learning rate）变化时，准确率差异明显，出现明显的颜色分层。即learning-rate对于模型正确率的影响远大于weight-decay。

小学习率模型训练缓慢，**欠拟合严重，未能有效提取特征，但是由于训练轮次设置**已经较大（epoch=30），即使收敛也不会具有良好的泛化能力和鲁棒性。

Weight-decay即使在 0.00001 或 0.000001，表现也差距不大；说明在学习率合理的前提下，weight decay 对性能的容忍度较大，但并非无关紧要。

1. **Batch\_size对性能的影响**

|  |  |  |
| --- | --- | --- |
| Batch\_size | 训练图像 | 训练特点 |
| 32 |  | 训练过程平稳收敛快速  但是在中后期存在梯度消失情况，准确率和损失变化不够明显 |
| 64 |  | 验证损失在中后期出现**较明显的波动** |
| 128 |  | 验证准确率整体**上升速度更慢且具有较大波动**； |

不同批次大小的准确率箱线图如下：

可以看出，在本数据集上中小规模的batch\_size效果较好。其中小规模效果最好，表现为高均值和低方差，具有较高的准确率和高稳定性的特点。但是在较深的网络中或较高学习率下易存在小批次存在梯度消失的问题，在训练过程中可以考虑调整网络结构等方式缓解。

**五、课题总结**

**1、简要回顾**

本课题聚焦于基于深度学习的人脸表情识别，通过在 RAF-DB 数据集上分别实现并对比了 ResNet18、ResNet50、DAN 与 GoogleNet 四种主流模型，系统性地探索了不同网络结构与超参数设置对模型性能的影响。并通过数据增强、参数调优、迁移学习的方式提高模型的准确度。

**2、课题的实际应用效果**

在加入数据增强与参数调优后，所有模型在识别准确率上均有明显提升。其中 GoogleNet 在本数据集上表现最为优异，达到 82.01% 的最佳准确率，DAN 紧随其后，达到 78.68%。从混淆矩阵可见，各模型对“快乐”“生气”等高辨识度表情分类效果良好，对“恐惧”“厌恶”等模糊类别仍存在误判，反映出深度模型在真实情绪识别中的适应性与局限性。

**3、实验中遇到的问题及解决方法**

（1）

问题：FAR-2013数据集自身准确度为65+-5%不高

解决办法：更换成自身准确率更高的学术数据集RAF-DB并进行合理的数据清洗和预处理。

（2）

问题：GoogleNet的实现过于复杂，涉及多个private函数难以复现

解决办法：使用pytorch的函数googlenet辅助实现

（3）

问题：参数调优训练时间过长（>=100h）

解决办法：固定epoch的轮次，不对epoch进行迭代，减少循环迭代轮次

**4、未来改进的方向**

当前实验已在 ResNet18 和 ResNet50 上完成了系统性的参数调优，并找到了在本数据集上表现最优的超参数组合。未来工作可基于该最佳配置，继续对该组参数进行微调，进一步提升 ResNet 系列模型的性能与稳定性。同时，仿照 ResNet 的训练流程与调优思路，对 DAN 与 GoogLeNet 两种结构开展系统性的超参数调优与模型优化工作,以构建更加精细化、泛化能力更强的最佳模型配置。

此外，还可引入自动超参数优化方法（如 Optuna、Bayesian Optimization），提升调参效率，辅助模型在不同数据集与场景中的迁移与部署。

**六、小组分工**

|  |  |
| --- | --- |
| 姓名 | 工作内容 |
| 鲍仁旭 | 模型训练、报告撰写、ppt制作、汇报、统筹分工 |
| 林宏津 | 模型训练、报告撰写、ppt制作 |
| 许太行 | 模型训练、报告撰写、ppt制作 |

参考文献：

[1] Zhengyao Wen, Wenzhong Lin, Tao Wang and Ge Xu. Fujian Provincial Key Laboratory of Information Processing and Intelligent Control. Distract your attention: multi-head cross attention network for facial expression recognition

[2] Christian Szegedy. Google Inc. Going deeper with convolutions

[3] Rohith Raj S, Pratiba D, Ramakanth Kumar P. Facial Expression Recognition using Facial Landmarks: A novel approach[R]. Bengaluru: RV College of Engineering, Department of Computer Science and Engineering, 2023.