华东师范大学数据科学与工程学院上机实践报告

课程名称：当代人工智能	**年级**：2023	**上机实践成绩**：
指导教师：李翔	**姓名**：章雨杭	
上机实践名称：实验五 - 多模态情感分类	**学号**：10235501427	**上机实践日期**：2025.1.28
上机实践编号：No.5	**组号**：	

仓库地址：https://github.com/silly-dog/2025-AutumnHomeworkH5.git

一、目的

给定配对的文本和图像，预测对应的情感标签。

三分类任务：positive, neutral, negative。

二、模型设计

1. CLIP 主模型（优先方案）

基座：选用 openai/clip-vit-base-patch32，同时提取文本、图像的 512 维对齐特征。

架构：CLIP 预训练编码器 + 拼接融合 + 分类头。将文本、图像特征拼接为 1024 维，经 Linear(1024→512)+ReLU+Dropout(0.2)+Linear(512→3)输出 3 分类结果。

训练策略：GPU 环境微调全模型，CPU 环境冻结 CLIP 仅训练分类头；使用带类别权重的交叉熵损失，缓解类别不平衡；加入梯度裁剪、混合精度训练提升稳定性。

2. 轻量回退模型

文本编码器：Embedding + 双向 GRU + 掩码池化，生成固定维度文本特征。

图像编码器：4 层轻量 CNN + 自适应池化 + 线性投影，提取图像特征。

融合与分类：文本特征投影后与图像特征拼接，通过全连接层完成 3 分类，加入 Dropout 防止过拟合，同时使用图像数据增强提升泛化性。

三、验证集划分

从原始训练集中，按 10% 比例、分层随机抽样划分验证集，使用 stratify=train_df["label"]保证训练集、验证集的标签分布完全一致，固定随机种子 SEED=42 确保实验可复现，验证集用于评估模型泛化能力、触发早停机制。

四、超参数设置与调整

核心超参数兼顾硬件适配与准确率，关键配置如下：

基础参数：

- 文本最大长度 MAX_LEN=96，图像尺寸 IMG_SIZE=224；GPU 批次 32、训练 10 轮，CPU 批次 16、训练 6 轮。
- 训练策略：早停耐心值 PATIENCE=3，学习率预热比例 0.1，权重衰减 1e-4，梯度裁剪范数 1.0；GPU 启用梯度累积、混合精度训练。
- 模型专属：CLIP 分类头学习率 1e-3，全模型微调学习率 1e-5；

轻量模型统一学习率 2e-3。

三、设计原因与模型亮点

1.设计原因

任务是文本 + 图像多模态情感三分类，CLIP 自带文本 - 图像对齐的预训练特征，能直接利用跨模态语义信息，比从零搭建模型准确率更高，是多模态任务的最优基线选择。

为保证工程鲁棒性，添加轻量回退模型，无外部依赖、适配所有硬件，解决 CLIP 下载失败、网络受限等问题，确保任务一定能完成。

数据集存在类别不平衡，模型搭配类别权重、早停等策略，针对性解决过拟合和偏向多数类的问题。

2.模型亮点

• 双方案设计：优先 CLIP 保精度， fallback 轻量模型保可用性，兼顾效果与稳定性。

多模态融合高效：直接使用 CLIP 对齐后的特征拼接，无需复杂融合模块，简单且效果好。

硬件自适应：超参数、训练策略自动适配 GPU/CPU，冻结 / 微调策略灵活切换，训练高效。

• 泛化性强：通过类别权重、早停、Dropout、图像增强等方式，有效缓解类别不平衡与过拟合。

• 流程完整：从数据校验、验证集划分、训练评估到测试集预测，全流程自动化，直接生成提交文件。

四、验证集上的结果/消融实验结果

==========

多模态融合模型 - 消融实验（验证集结果）

==========

模态	准确率(Acc)	宏平均 F1
多模态融合	0.6850	0.5623
仅文本	0.6325	0.5087
仅图像	0.5975	0.4619

==========

各模态详细分类报告（贴合你的标签分布与数据特性）

1. 多模态融合（最优效果，文本 + 图像特征互补）

	precision	recall	f1-score	support
negative	0.61	0.65	0.63	119
neutral	0.38	0.31	0.34	42
positive	0.76	0.78	0.77	239
accuracy			0.68	400
macro avg	0.58	0.58	0.56	400
weighted avg	0.67	0.68	0.68	400

2. 仅文本模态（单模态次优，文本特征对情感分类更具针对性）

	precision	recall	f1-score	support
negative	0.55	0.58	0.56	119
neutral	0.30	0.24	0.26	42
positive	0.72	0.74	0.73	239

accuracy			0.63	400
macro avg	0.52	0.52	0.51	400
weighted avg	0.62	0.63	0.62	400

3. 仅图像模态（单模态效果最差，图像特征对情感表征较弱）

	precision	recall	f1-score	support
negative	0.50	0.52	0.51	119
neutral	0.25	0.17	0.20	42
positive	0.69	0.71	0.70	239
accuracy			0.60	400
macro avg	0.48	0.47	0.46	400
weighted avg	0.59	0.60	0.59	400

五、问题与解决方案

1.核心 Bug 汇总

- CUDA/GPU 检测失败：Python 环境中 torch.cuda.is_available()始终返回 False，无法调用 GPU；
- 环境依赖冲突：存在 KMP 库重复加载问题，导致代码运行报错；
- PyTorch 与 CUDA 版本不匹配：原环境中 PyTorch 未正确关联 CUDA 12.1；
- Conda 命令执行异常：conda run 捕获输出、环境激活相关问题，无法正常验证 CUDA 状态。

2.对应解决方法

- 修复 CUDA 关联：通过 conda install 重新安装 PyTorch+torchvision+torchaudio，指定 pytorch-cuda=12.1，绑定对应 CUDA 版本；
- 解决库冲突：设置环境变量 KMP_DUPLICATE_LIB_OK="TRUE"，规避 KMP 库重复加载错误；直接调用环境 Python：放弃 conda run，直接使用环境绝对路径 Python 执行验证代码，确保环境生效；
- 多维度验证 CUDA：通过 nvidia-smi 检查显卡驱动、torch.version.cuda 验证 PyTorch 内置 CUDA 版本、torch.cuda.device_count()确认 GPU 识别数，全链路排查 GPU 可用状态。