

实验四：综合实验

一、实验名称

基于骨骼关键点的**动作姿态识别与分析** —— 从数据处理到模型优化的全流程实践及扩展应用。

二、实验背景

在智能监控、人机交互、体育训练等领域，动作姿态识别具有重要应用价值。骨骼关键点数据能够精准捕捉人体运动特征，结合深度学习模型可实现对复杂动作的有效识别。本实验基于提供的骨骼关键点数据集，融合数据处理、模型构建与优化等核心环节，让学生深入理解骨骼关键点在动作姿态识别中的核心作用，了解动作识别的完整技术流程，掌握深度学习模型处理时序数据的方法，同时鼓励学生在实验过程中进行创新实践，探索动作识别技术在不同场景下的优化方向与应用实现。

三、实验目标

- 数据处理能力：**掌握骨骼数据的清洗、归一化及特征提取方法，理解关节点坐标与姿态表征的关系。
- 模型构建与训练：**熟悉卷积神经网络（CNN）、循环神经网络（RNN）、LSTM（长短期记忆网络）等模型在时序数据中的应用，能根据算力调整模型结构。
- 实验流程把控：**完成从数据集加载、模型训练到测试评估的全流程操作，学会记录训练日志并分析模型性能。
- 创新与优化：**尝试不同的数据增强策略、模型架构或损失函数，探索提升识别准确率的方法。
- 扩展研究：**通过扩展研究中的多模态融合、实时应用和异常行为检测等方向的探索研究，激发对前沿技术及应用实现的钻研兴趣。

四、实验环境

- 硬件：**CPU/GPU（建议 NVIDIA 显卡以支持 CUDA 加速）
- 软件：**Python 3.x、PyTorch 深度学习框架、Pandas/NumPy 数据处理库、Matplotlib 可视化工具

五、实验数据

- 数据集来源：**自主采集的骨骼关键点数据（.csv 格式），包含多主体、多动作类别（如握手、喝水、投掷等）。
- 数据结构：**每行记录一个时间帧的关节点信息，包含关节 ID、三维坐标（X/Y/Z）、旋转四元数（W/X/Y/Z）等。
- 示例字段：**

Frame	JointID	X	Y	Z	W	RotationX	RotationY	RotationZ
232	1	0.0295	0.9973	-0.0565	0.0356	-167.19	881.21	-703.45

六、实验步骤

（一）数据预处理

1. 数据清洗

- 过滤无效帧：剔除关节坐标全为零或明显异常的时间帧。
- 去除干扰主体。
- 缺失值处理等。

2. 特征提取

- 选取关键关节：根据人体结构，提取头部、躯干、四肢等 关键关节（如颈、肩、肘、腕、髋、膝、踝等，对应joint_index列表）。
- 坐标归一化：以胸腔关节为中心，将所有关节坐标转换为相对坐标，消除个体差异与平移影响（参考body_centered_normalization函数）。
- 时序采样：按固定间隔（如每 4 帧）采样，降低数据维度并保留动作时序特征（参考for f in range(3, frames+4, 4)逻辑）。

3. 数据增强

- 随机旋转：对关节坐标施加小幅度随机旋转，模拟不同视角。
- 噪声添加：在坐标中注入高斯噪声，提升模型鲁棒性。

- 时间扭曲：随机删除或重复部分帧，模拟动作速度变化。

(二) 模型构建与训练

1. 基线模型（CNN-RNN、CNN-LSTM混合网络等）

- **输入层**：将骨骼数据转换为“通道 - 时间 - 关节”三维张量（形状为[C, T, V]，参考`data.transpose(2, 0, 1, 3)`）。
- **卷积层**：使用一维卷积（Conv1D）提取关节间空间特征，如`nn.Conv1d(in_channels=V, out_channels=64, kernel_size=3)`。
- **循环层**：采用 LSTM 或 GRU 层捕捉时序依赖，如`nn.LSTM(input_size=64, hidden_size=128, num_layers=2, bidirectional=True)`。
- **全连接层**：通过两层全连接层完成分类，输出维度为动作类别数。

2. 模型优化

- **损失函数**：使用交叉熵损失（CrossEntropyLoss）。
- **优化器**：Adam 优化器，初始学习率 0.1，权重衰减 $4e-4$ 。
- **训练配置**：批量大小 64，训练 20 epoch，记录每轮训练的损失与准确率至`log.txt`。

(三) 实验验证与分析

1. 测试集评估

- 使用分层抽样划分训练集与测试集（测试集占比 20%，参考`train_test_split`）。
- 计算测试集准确率、混淆矩阵等，分析各类别识别效果。

2. 可视化与调优

- 绘制训练曲线：对比损失值与准确率随 epoch 的变化趋势（使用 Matplotlib）。
- 姿态可视化：将骨骼数据还原为三维姿态，对比预测动作与真实动作的差异（可选工具：Motive 或自定义 3D 绘图库）。
- 模型改进方向：
 - 尝试轻量化模型（如 MobileNet、Transformer）以减少计算量。
 - 引入注意力机制（Attention）增强关键关节的特征权重。
 - 调整数据预处理参数（如采样间隔、归一化方式）。
 - 多模态融合：结合视频 RGB 数据与骨骼关键点，构建双流网络等。

七、扩展研究

- 多模态融合：**结合骨骼数据与视频图像，使用双流网络提升识别精度。
- 实时应用：**通过模型部署，实现实时动作姿态检测。
- 异常行为检测：**基于自编码器（AE）或生成对抗网络（GAN）等构建正常动作模型，检测异常动作。
- 自主研究提出：**可自主提出合理扩展应用需求方向并进行研究分析实现。

八、实验成果

1. 实验报告（Word）

- 内容结构：**包含实验目的、实验内容及步骤，包括数据处理流程、模型架构图、训练曲线、测试结果（准确率、混淆矩阵）、优化方案对比等。
- 分析讨论：**总结模型优缺点，提出进一步改进思路（如多模态融合、小样本学习等），总结遇到的问题及解决方案。
- 扩展研究：**针对（七、扩展研究），可选任其一或提出新方向，进行研究学习，包括可行性、架构设计、技术路线、应用实现等总结介绍。

2. 代码提交

- 完整 Jupyter Notebook 或 PyTorch 项目文件，包含数据预处理、模型定义、训练与测试脚本。
- 关键代码注释清晰，如数据加载函数 `preprocess_data`、模型类 `Model` 的实现细节。

3. 命名规范：文件命名为“姓名 - 学号 - 班级 - 实验四”，压缩包包含报告与代码文件夹。

九、实验安全与注意事项

- 数据隐私：**确保采集数据仅用于实验，不泄露主体身份信息。
- 算力管理：**训练大型模型时注意 GPU 内存占用，可通过减小批量大小或使用梯度累积优化。
- 版本控制：**建议使用 Git 管理代码版本，避免实验结果丢失。

通过本实验，学生将系统掌握从数据预处理到模型部署的全流程技术，深入理解深度学习在时序数据中的应用逻辑，培养问题分析与模型优化的实践能力。同时，通过扩展研究中的多模态融合、实时应用和异常行为检测等方向的探索，激发学生对前沿技术的钻研兴趣，为未来在智能监控、人机交互等领域的深入研究与实际应用奠定坚实基础。