

项目文档说明

一、项目概述

项目用于时序动作定位的深度学习模型，采用点监督学习方式，通过分层可靠性传播机制提升动作片段检测与定位精度，适用于视频内容分析，在精准识别视频中动作发生时间的场景下表现良好。

二、功能说明

(一) 两阶段训练模式

- 第一阶段 (Stage1)：利用点监督数据训练 S-Model。S-Model 包含可靠记忆模块 (Reliable Memory)、编码器 (Encoder) 和分类器 (Classifier)。可靠记忆模块存储和更新各类动作特征原型，编码器基于可靠性感知块 (ReliabilityAware Block) 对输入特征编码，分类器对编码后特征进行动作分类预测，输出每个视频片段属于各类动作的概率及可靠性特征。
- 第二阶段 (Stage2)：使用第一阶段生成的提案训练 I-Model。I-Model 包含边界完整性回归器 (BoundaryCompletenessRegressor)，对第一阶段提案细化，通过回归边界完整性优化动作片段定位，提高定位精度。

(二) 数据集支持

主要以 THUMOS14 数据集为例开发和测试。THUMOS14 是常用时序动作定位数据集，含多种体育运动动作。模型可根据数据集的文件路径、视频特征维度、帧率等参数灵活配置，适配不同数据集。

(三) 多维度评估指标

使用平均精度均值 (mAP) 作为主要评估指标，涵盖不同交并比 (IoU) 阈值下的 mAP 计算，如平均 mAP (0.1 - 0.7)、平均 mAP (0.1 - 0.5)、平

均 mAP (0.3 - 0.7) 以及特定阈值下的 mAP (如 mAP@0.1 至 mAP@0.7 等) , 全面衡量模型在不同动作定位宽松程度下的性能表现。

三、技术架构

(一) 数据处理

数据加载与预处理：通过 dataset.py 文件中的 dataset 类加载视频数据，根据数据集路径读取视频特征文件、标注文件等。对视频特征采样处理，支持随机采样和均匀采样，适应不同训练需求。同时加载视频标签信息，包括视频级别标签和点标签，用于模型训练过程中的监督学习。

提案生成与处理：第一阶段训练结束后，利用 ranking.py 文件中的 reliability_ranking 函数生成提案，包括正提案 (PP) 、可靠提案 (RP) 和负提案 (NP) 。正提案基于模型预测的动作片段，可靠提案通过匹配点标注与正提案得到，负提案从背景区域生成用于平衡训练数据。提案存储为 JSON 文件，供第二阶段训练使用。

(二) 模型结构

S-Model：由 reliable_memory、encoder 和 classifier 三部分组成。reliable_memory 存储动作特征原型，用于对比学习和特征更新；encoder 采用多层可靠性感知块对输入特征编码，捕捉特征间相关性和可靠性信息；classifier 对编码后特征分类，预测每个时间点属于各类动作的概率及背景概率。

I-Model：主要包含边界完整性回归器 (BoundaryCompletenessRegressor) ，对输入的特征和提案处理。通过卷积层和全连接层对特征变换，预测提案的边界偏移量和置信度，优化动作片段定位。

(三) 训练流程

第一阶段训练 (Stage1) ：在 main.py 文件中，当 stage 参数设置为 1 时，启动 S-Model 的训练流程。模型初始化后，使用 Adam 优化器进行参数更新，通过自定义的损失函数 (包括视频级别分类损失、帧级别分类损失、背景分

类损失和对比损失) 计算训练误差, 并定期在测试集上评估模型性能, 保存最优模型参数。

第二阶段训练 (Stage2) : 当 stage 参数设置为 2 时, 启动 I-Model 的训练流程。同样使用 Adam 优化器, 基于自定义的损失函数 (包括边界回归损失和置信度回归损失) 对模型训练, 利用第一阶段生成的提案作为输入, 优化动作片段的边界定位。

四、使用说明

(一) 环境配置

本项目基于 Python 语言开发, 依赖 PyTorch 深度学习框架及其他常用库如 NumPy、Pandas、Tqdm 等。需确保 Python 环境已正确安装, 并通过 pip 安装项目所需的依赖包。

(二) 参数配置

项目参数通过 options.py 文件中的 parse_args 函数进行配置。主要参数包括数据集名称、数据集路径、模型训练阶段 (Stage1 或 Stage2)、训练迭代次数、学习率、批量大小等。用户可根据实际需求修改这些参数, 适应不同训练场景和数据集特性。

(三) 模型训练与测试

训练模型: 在 main.py 文件中, 将 mode 参数设置为 “train”, 并指定相应的训练阶段 (stage 参数)。运行 main.py 文件, 模型将按照配置参数进行训练。训练过程中, 会在日志文件中记录训练进度、损失值等信息, 并定期在测试集上评估模型性能, 最终保存最优模型参数到指定的模型保存路径。

测试模型: 将 mode 参数设置为 “test”, 并指定测试阶段的模型参数文件路径。运行 main.py 文件, 模型将加载训练好的参数, 在测试集上进行推理, 输出测试结果, 包括各类动作的定位结果以及评估指标值。