

GAIDE: 臨床步態分析與診斷輔助系統

GAIDE (Gait Analysis and Insight for Diagnostic Excellence) 是一個基於電腦視覺的臨床步態分析系統，旨在從低成本的RGB攝影機影片中，自動提取3D人體骨架，並利用部分空間法與機器學習/深度學習模型，對步態相關疾病(ASD, DHS, LCS, HipOA)進行分類與分析，為臨床診斷提供可解釋的量化依據。

專案特色

- 模組化流程: 整個分析流程被拆分為數據預處理、特徵提取、資料集建立、模型訓練與視覺化分析等多個獨立模組。
- 命令列驅動: 透過main.py和各個獨立腳本提供強大的命令列介面，可以分步執行、監控進度。
- 可解釋特徵: 結合部分空間法，提取具有物理意義的步態特徵(如「速度」和「平滑度」)，為模型的判斷提供依據。
- 多模型支援: 支援傳統機器學習模型(SVM)和先進的深度學習模型(ST-GCN)，並包含後期融合(Late Fusion)架構。
- 多樣化分析: 提供混淆矩陣、平均波形圖、頻譜分析等多種視覺化分析工具，以深入洞察步態模式。

專案結構

```
human_gait/
├── data/
│   ├── raw_videos/          # 存放原始步態影片
│   └── processed_skeletons/  # 存放預處理後的3D骨架.npy檔
├── results/
│   ├── gifs/                # 存放部分空間法產生的GIF動圖
│   ├── subspace_npy/        # 存放部分空間法產生的特徵.npy檔
│   ├── figures/             # 存放混淆矩陣、波形圖等分析圖表
│   ├── models/              # (建議) 存放訓練好的模型檔案
│   ├── svm_features.npy     # 最終用於SVM訓練的特徵矩陣
│   ├── labels.npy           # 對應的標籤向量
│   └── stgc_npaths.npy      # 供ST-GCN使用的骨架檔案路徑
├── src/
│   ├── cv_motion3d_public-main/ # 外部的部分空間法腳本
│   ├── __init__.py           # 將src標記為Python包
│   ├── config.py             # 全局路徑與參數設定
│   ├── data_processing.py    # 數據預處理函數
│   └── subspace_method.py    # 部分空間法呼叫函數
```

— train.py	# SVM模型訓練與評估函數
— visualize.py	# 3D骨架視覺化函數
— stgcn_dataset.py	# ST-GCN的PyTorch Dataset類別
— stgcn_layer.py	# 自定義的ST-GCN層
— stgcn_models.py	# ST-GCN模型架構定義
— engine.py	# 深度學習訓練與評估引擎
— main.py	# 主執行腳本 (預處理, 部分空間法)
— build_master_dataset.py	# 建立最終訓練資料集的腳本
— train_stgcn.py	# 訓練ST-GCN模型的主腳本
— evaluate_stgcn.py	# 評估已儲存的ST-GCN模型
— analyze_waveforms.py	# 波形分析與視覺化腳本
— process_normal_control.py	# 處理正常對照組樣本的腳本
— requirements.txt	# 專案依賴的Python套件

安裝與設定

1. 複製專案：

```
git clone [您的專案Git網址]
cd human_gait
```

2. 建立Conda環境：

建議使用conda來管理環境，以更好地處理PyTorch等複雜依賴。
 # 建立一個新的Python 3.10環境

```
conda create -n gait_analysis python=3.10
conda activate gait_analysis
```

3. 安裝依賴套件：

使用提供的requirements.txt檔案來安裝所有必要的Python套件。

```
pip install -r requirements.txt
```

注意：*torch*和*torch_geometric*的安裝可能需要根據您的硬體(CPU, CUDA, Apple Silicon)進行調整。*requirements.txt*中提供的是適用於Apple Silicon (Mac)的版本。

4. 設定路徑：

打開src/config.py檔案，確認其中的資料夾路徑(RAW_VIDEO_DIR, PROCESSED_SKELETON_DIR等)與您的設定一致。

使用流程

整個研究流程被設計為可以分步執行。請依照以下順序操作。

步驟 1: 數據預處理

此步驟會將data/raw_videos/中的所有影片，透過MediaPipe轉換為經過插值和平滑化處理的3D骨架.npy檔案，並儲存到data/processed_skeletons/。

```
python main.py preprocess
```

- 此指令會自動跳過已經處理過的檔案。
- 若要強制重新處理所有檔案，請加上--force參數：python main.py preprocess --force

步驟 2: 部分空間法分析

此步驟會讀取processed_skeletons中的骨架檔案，執行部分空間法分析，並產生兩種結果：

- 視覺化的.gif動圖，儲存於results/gifs/。
- 量化的特徵時間序列.npy檔，儲存於results/subspace.npy/。

```
python main.py subspace
```

- 此指令同樣會自動跳過已處理的檔案。
- 若要強制重新處理，請使用--force參數。
- 若只想測試單一檔案，可使用--file參數：python main.py subspace --file [檔案名稱].npy

步驟 3: 建立最終訓練資料集

此步驟會掃描所有中間檔案，整合特徵(透過填充/截斷)，並從檔名中提取標籤，最終生成三個供模型訓練使用的檔案。

```
python build_master_dataset.py
```

- 輸出：
 - results/svm_features.npy (給SVM用的特徵矩陣 X)
 - results/labels.npy (對應的標籤向量 y)
 - results/stgcn_paths.npy (給ST-GCN使用的骨架檔案路徑)

步驟 4: 模型訓練 (ST-GCN)

此步驟會讀取上一步產生的數據，訓練ST-GCN模型，並將訓練日誌、評估報告、混淆矩陣圖和最佳模型權重，儲存到一個以實驗參數命名的獨立資料夾中。

- 訓練純**ST-GCN**基準模型：

```
python train_stgcn.py --model baseline --lr 0.001 --epochs 50
```
- 訓練後期融合模型：

```
python train_stgcn.py --model late_fusion --lr 0.0005 --epochs 100
```

步驟 5: 評估已儲存的模型

使用evaluate_stgcn.py腳本來載入一個訓練好的模型，並在測試集上進行一次詳細的評估。

```
python evaluate_stgcn.py --model baseline --weights "results/[實驗資料夾名稱]
/best_model.pth"
```

其他分析工具

單一骨架視覺化

快速預覽某個預處理好的骨架動畫。

```
python main.py visualize [檔案名稱].npy
```

波形深度分析

analyze_waveforms.py腳本提供了更進階的視覺化分析功能。

- 繪製平均波形圖：比較不同疾病類別的「平均步態模式」。
分析 'first' 特徵，並疊加正常對照組的波形
python analyze_waveforms.py waveforms --feature first --control_id Normal_Control_01
- 執行頻譜分析：
python analyze_waveforms.py fft

此功能會計算步頻、頻譜熵等特徵，並產生對應的箱形圖。