

Recognizer 프로젝트 폴더 구조별 기능 설명서

프로젝트 개요

MMPose 기반의 실시간 동작 인식 및 분석 시스템으로, 포즈 추정, 객체 추적, 동작 분류, 이벤트 감지 기능을 통합 제공합니다.

전체 폴더 구조

```
recognizer/
├─ action_classification/      # 동작 분류 모듈
├─ configs/                   # 설정 파일
├─ core/                       # 핵심 추론 모드
├─ docs/                       # 문서 및 가이드
├─ events/                     # 이벤트 관리 시스템
├─ main.py                     # 메인 실행 파일
├─ output/                     # 출력 결과 디렉토리
├─ pipelines/                  # 파이프라인 구현
├─ pose_estimation/            # 포즈 추정 모듈
├─ scoring/                    # 점수 계산 모듈
├─ tracking/                   # 객체 추적 모듈
├─ utils/                      # 유틸리티 함수
└─ visualization/              # 시각화 모듈
```

1. action_classification/ - 동작 분류 모듈

구조

```
action_classification/
└─ stgcn/
    ├── stgcn_classifier.py      # STGCN 기반 동작 분류기
    └─ __pycache__/
```

기능

- **STGCN (Spatial-Temporal Graph Convolutional Networks)** 기반 동작 분류
- Fight/NonFight, Normal/Falldown 등 이진 분류 수행
- 시공간 그래프 컨볼루션을 통한 고정밀 동작 인식
- MMAction2 프레임워크와 연동

주요 클래스

- `STGCNActionClassifier` : 메인 분류기 클래스
- 입력: 포즈 시퀀스 데이터 (keypoint coordinates)
- 출력: 동작 클래스 확률값

2. configs/ - 설정 파일

구조

```
configs/  
└─ config.yaml          # 메인 설정 파일
```

기능

- 전체 시스템 설정 통합 관리
- 모드별 파이프라인 설정 (annotation, inference, realtime)
- 모델별 하이퍼파라미터 설정
- 듀얼 서비스 (Fight/Falldown) 설정
- 성능 최적화 및 이벤트 관리 설정

주요 설정 섹션

- mode : 실행 모드 선택
- dual_service : 멀티 서비스 활성화
- models : 각 모델의 체크포인트 및 설정
- performance : 성능 최적화 설정
- events : 이벤트 감지 임계값

3. core/ - 핵심 추론 모드

구조

```
core/  
└─ inference_modes.py    # 추론 모드 구현체
```

기능

- inference.analysis: 배치 분석 모드
- inference.realtime: 실시간 처리 모드
- inference.visualize: 시각화 모드
- 멀티프로세싱 지원 및 GPU 분산 처리
- 성능 평가 및 보고서 생성

주요 기능

- 비디오 파일 배치 처리
- 실시간 스트림 분석
- 결과 시각화 및 저장
- 성능 메트릭 계산

4. events/ - 이벤트 관리 시스템

구조

```
events/
├─ event_logger.py          # 이벤트 로깅
├─ event_manager.py        # 이벤트 생명주기 관리
└─ event_types.py          # 이벤트 타입 정의
```

기능

- 실시간 이벤트 감지 및 관리
- Fight/Falldown 등 이벤트 상태 추적
- 연속성 검증 및 쿨다운 처리
- 이벤트 로그 생성 및 알림 시스템

주요 클래스

- EventManager : 이벤트 생명주기 관리
- EventLogger : 구조화된 이벤트 로깅
- EventType : 이벤트 타입 열거형

5. pipelines/ - 파이프라인 구현

구조

```
pipelines/
├─ analysis/                # 분석 파이프라인
├─ dual_service/           # 듀얼 서비스 파이프라인
└─ separated/              # 분리된 파이프라인
```

기능

- 모듈화된 처리 파이프라인
- 포즈 추정 → 추적 → 분류 → 이벤트 감지 체인
- 듀얼 서비스를 통한 다중 동작 감지
- 병렬 처리 및 성능 최적화

파이프라인 타입

- AnalysisPipeline : 배치 분석용
- DualServicePipeline : 실시간 멀티 서비스
- SeparatedPipeline : 단일 서비스 전용

6. pose_estimation/ - 포즈 추정 모듈

구조

```
pose_estimation/
├── base.py                # 포즈 추정 기본 클래스
├── rtmo/                  # RTMO 구현체
│   ├── enhanced_rtmo_extractor.py
│   ├── rtmo_estimator.py  # PyTorch 버전
│   ├── rtmo_onnx_estimator.py  # ONNX 버전
│   ├── rtmo_tensorrt_estimator.py # TensorRT 버전
│   └── multiclass_nms.py   # NMS 구현
```

기능

- 실시간 멀티 퍼슨 포즈 추정
- RTMO (Real-Time Multi-Object) 모델 기반
- PyTorch, ONNX, TensorRT 다중 백엔드 지원
- 17개 keypoint 기반 Body7 포맷

최적화 기능

- GPU 메모리 관리 및 배치 처리
- 동적 입력 크기 조정
- 다중 스레드 후처리

7. scoring/ - 점수 계산 모듈

구조

```
scoring/
├── motion_based/
│   ├── fight_scorer.py    # Fight 점수 계산
│   └── falldown_scorer.py # Falldown 점수 계산
```

기능

- 동작별 맞춤형 점수 계산
- 모션 기반 특징 추출 및 점수화
- 시간적 일관성 및 품질 평가
- 가중치 기반 종합 점수 산출

Fight Scoring

- 움직임 강도, 상호작용, 위치 관계 분석
- 시간적 패턴 인식

Falldown Scoring

- 높이 변화, 자세 각도, 지속성 분석
- 쓰러짐 패턴 특화 알고리즘

8. tracking/ - 객체 추적 모듈

구조

```
tracking/
├── base.py           # 추적 기본 클래스
├── bytetrack/       # ByteTrack 구현
│   ├── byte_tracker.py # 메인 추적기
│   ├── core/         # 칼만 필터 등 핵심 로직
│   ├── models/       # 추적 모델
│   └── utils/        # 유틸리티 함수
```

기능

- 실시간 다중 객체 추적
- ByteTrack 알고리즘 기반
- 칼만 필터를 통한 예측 및 보정
- ID 일관성 유지 및 재연결 처리

주요 특징

- 높은 정확도의 ID 할당
- 가려짐 상황 처리
- 경량화된 연산으로 실시간 처리

9. utils/ - 유틸리티 함수

구조

```
utils/
├── config_loader.py # 설정 로더
├── factory.py       # 팩토리 패턴
├── onnx_base.py     # ONNX 기본 클래스
└── window_processor.py # 윈도우 처리기
```

기능

- 공통 유틸리티 함수 제공
- 설정 파일 파싱 및 검증
- 모듈 팩토리 및 등록 시스템
- 슬라이딩 윈도우 처리
- ONNX 모델 공통 인터페이스

10. visualization/ - 시각화 모듈

구조

```
visualization/
└── pkl_visualizer.py # PKL 결과 시각화
```

기능

- 결과 데이터 시각화
- 포즈 스켈레톤 오버레이
- 추적 경로 표시
- 이벤트 상태 표시
- 실시간 및 배치 시각화 지원

11. output/ - 출력 결과 디렉토리

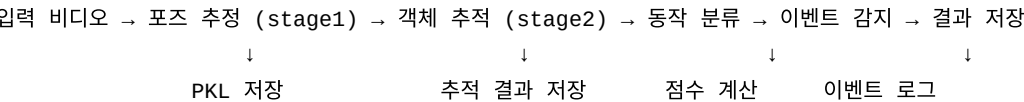
구조

```
output/  
└─ [dataset_name]/  
   └─ stage1_poses/      # 1단계: 포즈 추정 결과  
   └─ stage2_tracking/  # 2단계: 추적 결과  
   └─ stage3_dataset/   # 3단계: 최종 데이터셋
```

기능

- 단계별 결과 저장
- JSON, PKL 포맷 결과 파일
- 시각화 오버레이 비디오
- 데이터셋 분할 결과 (train/val/test)

전체 데이터 플로우



주요 특징

1. **모듈화 설계**: 각 기능이 독립적으로 구현되어 유지보수 용이
2. **다중 백엔드**: PyTorch, ONNX, TensorRT 지원으로 다양한 환경 대응
3. **실시간 처리**: 스트리밍 데이터 처리 및 실시간 이벤트 감지
4. **확장성**: 새로운 동작 분류기나 추적기 쉽게 추가 가능
5. **성능 최적화**: GPU 메모리 관리, 멀티프로세싱, 배치 처리 지원