

모여봄 (Moyeobom)

실시간 사람 탐지 및 혼잡도 분석 시스템입니다. YOLO11과 Norfair를 활용하여 웹캠 또는 비디오에서 사람을 탐지하고, 인원 수, 혼잡도, 이동 방향을 실시간으로 분석합니다.

주요 기능

- **실시간 사람 탐지:** YOLO11 모델을 사용한 정확한 사람 탐지
- **객체 추적:** Norfair를 활용한 다중 객체 추적 및 ID 할당
- **인원 수 카운팅:** 실시간으로 탐지된 사람 수 표시
- **혼잡도 분석:** 프레임 내 사람이 차지하는 면적 비율을 기반으로 혼잡도 측정 (여유/보통/혼잡)
- **이동 방향 분석:** 각 객체의 이동 방향(상/하/좌/우) 추적 및 통계
- **웹 인터페이스:** 실시간 비디오 스트리밍 및 데이터 시각화
- **비디오 재생:** 저장된 비디오 파일에 대한 탐지 및 분석

기술 스택

- **Backend:** Flask
- **Computer Vision:**
 - YOLO11 (Ultralytics)
 - OpenCV (cv2)
 - Norfair (객체 추적)
- **Deep Learning:** PyTorch
- **Frontend:** HTML, CSS, JavaScript

설치 요구 사항

필수 요구사항

- Python 3.10.15
- Anaconda (권장)

설치 방법

1. 가상환경 생성 및 활성화

```
conda create -n moyeobom python=3.10.15
conda activate moyeobom
```

2. 필수 패키지 설치

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

또는 개별 패키지 설치:

```
pip install ultralytics flask norfair opencv-python numpy torch
```

3. **GPU 지원 (선택사항)** GPU가 있는 경우 PyTorch GPU 버전을 먼저 설치한 후 나머지 패키지를 설치하세요:

```
# CUDA 12.1용 (대부분의 최신 GPU)
pip install torch torchvision torchaudio --index-url
https://download.pytorch.org/whl/cu121
# 또는 CUDA 11.8용
pip install torch==2.5.0 torchvision==0.20.0 torchaudio==2.5.0 --index-url
https://download.pytorch.org/whl/cu118

# 그 다음 나머지 패키지 설치
pip install -r requirements.txt
```

💡 **참고:** GPU 설치에 대한 자세한 내용은 [SETUP_GUIDE.md](#)를 참고하세요.

실행 방법

1. 프로젝트 디렉토리로 이동

```
cd moyeobom
```

2. **Flask** 애플리케이션 실행

```
python app.py
```

3. 웹 브라우저에서 접속

- 메인 페이지: <http://localhost:5000>
- 사람 수 카운팅: http://localhost:5000/count_view
- 혼잡도 분석: http://localhost:5000/area_view
- 지도 뷰: http://localhost:5000/map_view
- 비디오 뷰: http://localhost:5000/video_view

프로젝트 구조

```
moyeobom/
├── app.py           # Flask 메인 애플리케이션
├── config.json      # 설정 파일 (서버, 카메라, 모델, 추적기, 혼잡도 설정)
├── requirements.txt  # 필수 패키지 목록
├── templates/       # HTML 템플릿
│   └── index.html   # 메인 페이지
```

```
|   |   | count_view.html      # 사람 수 카운팅 뷰  
|   |   | area_view.html      # 혼잡도 분석 뷰  
|   |   | map_view.html       # 지도 뷰  
|   |   | map_view_info.html  # 지도 정보 뷰  
|   |   | video_view.html     # 비디오 뷰  
|   | static/                 # 정적 파일  
|   |   | css/                # 스타일시트  
|   |   | images/            # 이미지 파일  
|   |   | videos/            # 비디오 파일 (분석 대상)  
|   |   | fonts/             # 폰트 파일  
|   | README.md              # 프로젝트 문서
```

API 엔드포인트

페이지 라우트

- `GET /` - 메인 페이지
- `GET /count_view` - 사람 수 카운팅 페이지
- `GET /area_view` - 혼잡도 분석 페이지
- `GET /map_view` - 지도 뷰 페이지
- `GET /map_view_info` - 지도 정보 페이지
- `GET /video_view` - 비디오 목록 및 재생 페이지

비디오 스트리밍

- `GET /video_feed` - 실시간 웹캠 스트리밍 (사람 탐지 및 추적)
- `GET /area_feed` - 실시간 웹캠 스트리밍 (혼잡도 분석)
- `GET /cam_feed` - 기본 웹캠 스트리밍
- `GET /video_feed/<video_name>` - 비디오 파일 스트리밍 (탐지 포함)

데이터 API

- `GET /detection_data_feed` - 실시간 탐지 데이터 (JSON)
 - 반환 데이터: `ratio_code`, `people_count`, `most_movement_direction`, `most_movement_count`
- `GET /get_count_data` - 사람 수 및 이동 방향 데이터
- `GET /get_ratio_data` - 혼잡도 데이터
- `GET /get_ratio_code` - 혼잡도 코드 (1: 여유, 2: 보통, 3: 혼잡)

사용 방법

1. 실시간 사람 수 카운팅

- `/count_view` 페이지에서 실시간으로 탐지된 사람 수와 주요 이동 방향을 확인할 수 있습니다.
- 각 사람은 고유 ID로 추적되며, 이동 방향이 시각화됩니다.

2. 혼잡도 분석

- `/area_view` 페이지에서 프레임 내 사람이 차지하는 면적 비율을 기반으로 혼잡도를 확인할 수 있습니다.
- 혼잡도는 다음 세 단계로 표시됩니다:

- **여유**: 면적 비율 < 30%
- **보통**: $30\% \leq \text{면적 비율} < 70\%$
- **혼잡**: 면적 비율 $\geq 70\%$

3. 비디오 분석

- `/video_view` 페이지에서 `static/videos/` 폴더에 저장된 비디오 파일을 선택하여 분석할 수 있습니다.
- 비디오 재생 시 실시간으로 사람 탐지 및 추적이 수행됩니다.

설정

프로젝트의 모든 설정은 `config.json` 파일에서 관리됩니다. 코드를 수정할 필요 없이 설정 파일만 변경하면 됩니다.

설정 파일 구조

`config.json` 파일에는 다음 설정들이 포함되어 있습니다:

- **서버 설정**: 호스트, 포트, 디버그 모드
- **카메라 설정**: 웹캠 인덱스
- **모델 설정**: YOLO 모델 파일 경로
- **추적기 설정**: 거리 함수 및 임계값
- **혼잡도 설정**: 여유/보통/혼잡 기준 비율

주요 설정 변경 방법

혼잡도 임계값 조정

`config.json` 파일에서 `complex_ratio` 값을 수정하세요:

```
"complex_ratio": {  
  "low_threshold": 30,  
  "high_threshold": 70  
}
```

웹캠 변경

여러 개의 웹캠이 연결되어 있을 때 다른 웹캠을 사용하려면:

```
"camera": {  
  "index": 0  
}
```

`"index": 0`을 `"index": 1` 또는 `"index": 2` 등으로 변경하세요.

추적기 민감도 조정

객체 추적이 너무 민감하거나 둔감할 때:

```
"tracker": {  
  "distance_threshold": 120  
}
```

- 값이 클수록: 더 멀리 있는 객체도 같은 객체로 인식 (더 민감)
- 값이 작을수록: 가까운 객체만 같은 객체로 인식 (덜 민감)

포트 번호 변경

다른 프로그램과 포트가 충돌할 때:

```
"server": {  
  "port": 5000  
}
```

"port": 5000을 다른 번호로 변경하세요 (예: "port": 5001).

⚠ **주의:** `config.json` 파일을 수정한 후에는 프로그램을 다시 시작해야 합니다.

주의사항

- 웹캠이 연결되어 있어야 실시간 스트리밍이 작동합니다.
- YOLO11 모델(`yolo11x.pt`)은 첫 실행 시 자동으로 다운로드됩니다.
- GPU가 없는 환경에서는 처리 속도가 느릴 수 있습니다.
- 비디오 파일은 `static/videos/` 디렉토리에 저장해야 합니다.

참고사항

- 이 프로젝트는 Flask를 메인 프레임워크로 사용합니다.
- `django/` 폴더에는 초기 개발 시 사용했던 Django 프로젝트가 남아있지만, 현재는 사용되지 않습니다. 모든 기능은 Flask 애플리케이션(`app.py`)에서 구현되어 있습니다.

라이선스

이 프로젝트는 개인/교육 목적으로 사용할 수 있습니다.

기여

버그 리포트나 기능 제안은 이슈로 등록해주세요.