프로젝트 보고서 (비전 AI 기반 행동 감지 자동 문서화 시스템 구축)

날짜	@2025년 11월 26일
발표 자료 1조_최종_프로젝트	https://www.miricanvas.com/login?redirect=%2Fv2%2Fdesign2%2F90e5b045-8d9a-4b67-92ce-18ce151986b1
깃허브	https://github.com/keulreobeu/sessac_project

▼ 목차

1. 프로젝트 개요

1.1. 주제 및 선정 배경

- **주제:** 비전 AI를 기반으로 GMP(Good Manufacturing Practice) 작업자의 행동을 자동 감지하고, 이를 이벤트 단계별로 문서화하여 기록의 일관성 및 추적성(Data Integrity)을 확보하는 시스템 구축
- **선정 배경:**
  - **Data Integrity 중요성 증대:** GMP 공정에서 기록성, 일관성, 추적성은 필수 요소임.
  - **현행 시스템의 한계:** 여전히 수작업 기록에 의존하여 누락, 지연, 오기입 등 휴먼 에러가 지속적으로 발생함.
  - **자동화 필요:** 행동 자체를 AI로 분석하고 실시간으로 문서화하여 신뢰성을 높이는 시스템이 필요함.

1.2. 문제 상황 및 해결 방안

구분	내용
문제 상황	- **휴먼 에러:** 기록 누락, 작성 지연, 수기 작성 오류 발생 - **비효율성:** QA 담당자가 모든 작업 영상을 수동으로 검토해야 함 - **검증 한계:** 행동 단계(열기-넣기-닫기)별 세부 확인이 어렵고 수기 기록의 신뢰성 저하
해결 방안	- **실시간 감지:** 카메라를 통해 작업 영상을 입력받고 랜드마크/BBOX 기반 행동 감지 - **자동 분류:** TCN/CNN 모델을 활용하여 이벤트 단계를 자동으로 분류 - **자동 문서화:** LangChain 및 LLM을 활용하여 GMP 문서를 자동 생성 - **모니터링:** 대시보드를 통해 작업 상태를 실시간 확인 및 검토

1.3. 프로젝트 구조 (End-to-End Pipeline)

1. **Input:** Pi Camera를 통한 영상 입력
2. **Pre-processing:** YOLOv8 / MediaPipe 기반 Landmark 및 BBOX 추출
3. **Classification:** TCN / CNN 시퀀스 모델을 활용한 행동 단계 분류
4. **Documentation:** LangChain + LLM 기반 문서화
5. **Storage:** Database 저장
6. **Review:** Dashboard 시각화 및 작업자 검토 후 제출

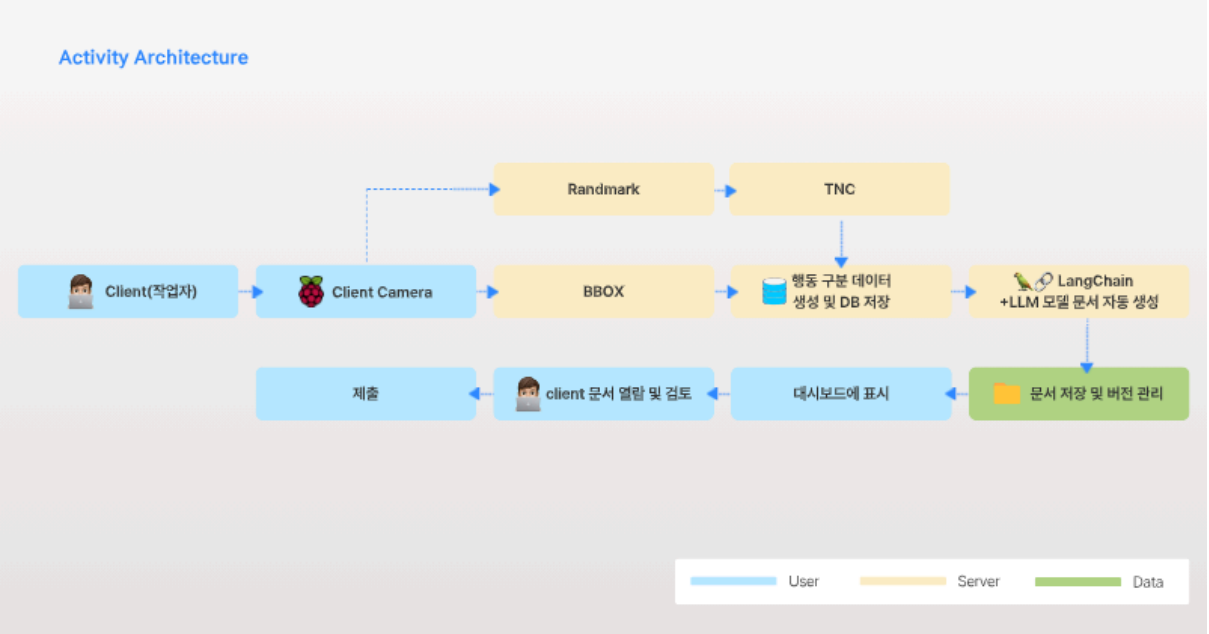
1.4. 활용 도구 및 기술 스택

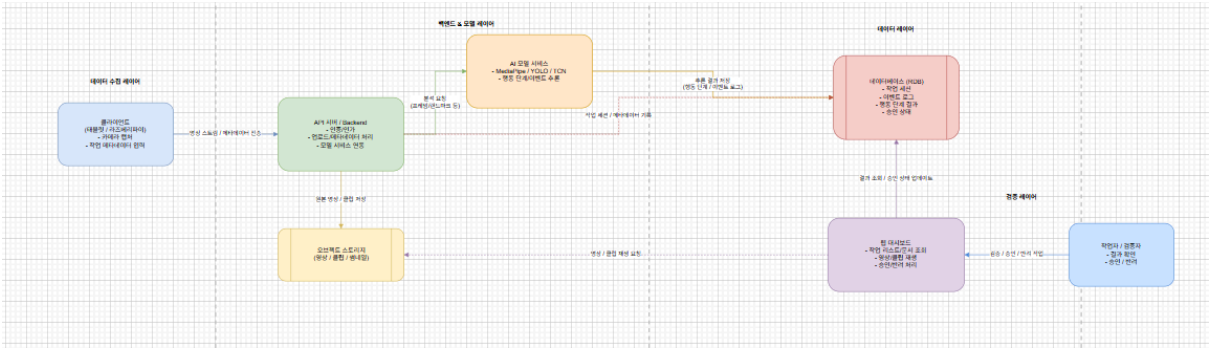
- **AI / Vision Modeling:** MediaPipe Hands, YOLOv8, TCN, CNN, BiLSTM
- **Backend / Dashboard:** FastAPI, Storage/Database, LangChain + LLM
- **Hardware:** Raspberry Pi Camera
- **Collaboration:** Jira, GitHub, Slack

1.5. 라벨링 체계 및 모델 통합

- **라벨링 정의:**
  - **이벤트 플래그:** GMP 행동 3단계(꺼내고 열기, 약 넣기, 닫고 넣기)
  - **랜드마크:** 관절 좌표(x, y, visibility)
  - **BBOX:** 상자, 약품 용기, 약 유무 상태 탐지용
- **모델 통합 전략 (Multi-head):**
  - YOLO-Pose 멀티헤드 + TCN Head 구조 채택
  - Head 1: Object Detection / Head 2: Pose Landmark / Head 3: Temporal Event Classification

1.6. 아키텍처





1.7. 기대 효과

- **기록 자동화:** 작업자 기록 부담 감소 및 생산성 향상
- **추적성 강화:** GMP 핵심 요구사항 충족 및 Data Integrity 확보
- **효율성 증대:** QA 문서 검토 시간 단축 및 휴먼 에러(누락/오류) 감소

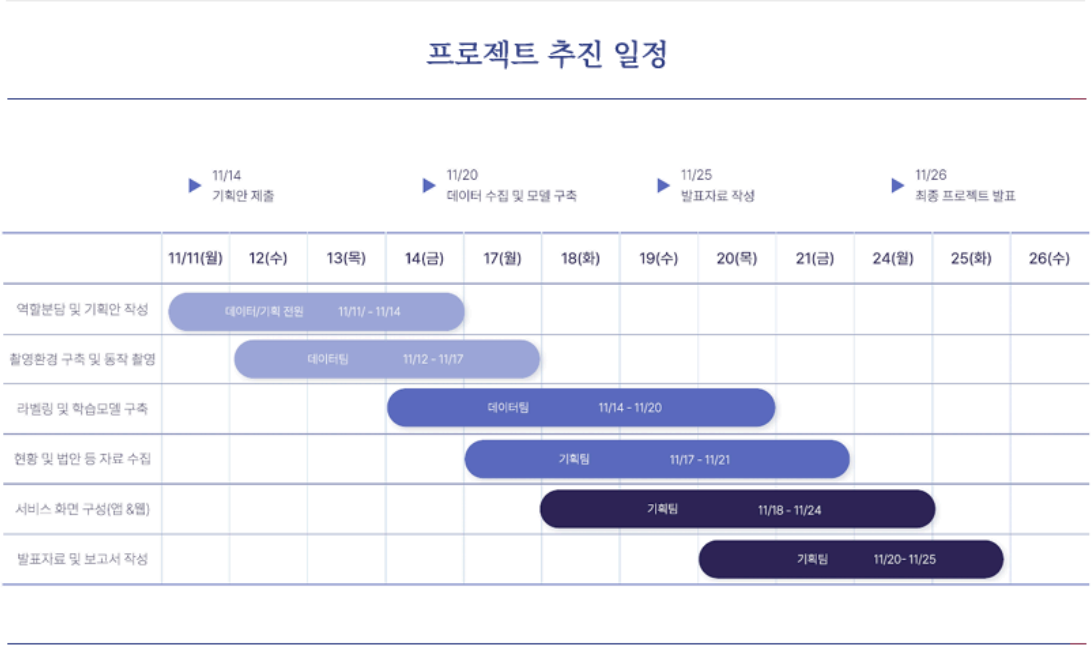
---

2. 프로젝트 팀 구성 및 역할

성명	역할	담당 업무
김인선	기획 총괄	- 개념 정리 및 시장 분석
김지현	데이터 수집 및 발표	- 데이터 수집 환경 구축 및 동작 촬영 - 학습 데이터 수집 및 시연 영상 촬영 - 프로젝트 발표
양선민	UX/UI 디자인	- 작업자 및 관리자 페르소나 설정 - UX/UI 및 앱/웹 화면 구성 및 구현
이진호	모델링 및 개발	- 데이터 수집/라벨링/전처리 및 윈도우 구성 - 행동 탐지 모델 비교 및 최종 구현 - 객체 탐지 모델 학습 및 모델 통합 출력 구현
장은정	기획 및 문서화	- 기획안 및 최종 발표 자료 작성 - LangChain 구축 및 자동 문서화 템플릿 제작 - 인터뷰 및 보고서 양식 작성, 시장 조사, 페르소나 설정

성명	역할	담당 업무
조창현	아키텍처 및 보고서	- 시스템 아키텍처 구성 - 동작 촬영 전반 및 객체 탐지 라벨링 - 최종 보고서 작성

3. 프로젝트 수행 과정



4. 프로젝트 수행 결과

1) 데이터 수집

- 구성 환경
  - 웹캡 또는 USB 카메라 사용
  - 720P HD이미지
  - 30fps를 목표로 수집하였지만 수집 환경의 문제로 7.5fps로 수집됨
  - OpenCV기반 이미지 촬영 진행
- 촬영 스크립트 기능
  - SPACE: 녹화 시작 및 종료
  - A/S/D: 이벤트 플래그 기록

    - Q/ESC: 종료
- 녹화 파일 구조

```
data
└── video/
    ├── normal/
    │   ├── video_normal_001/
    │   │   ├── frame_000000.jpg
    │   │   ├── frame_000001.jpg
    │   │   ├── frame_000002.jpg
    │   │   └── ...
    │   ├── video_normal_001_events.csv
    │   ├── video_normal_002/
    │   │   ├── frame_000000.jpg
    │   │   ├── frame_000001.jpg
    │   │   └── ...
    │   ├── video_normal_002_events.csv
    │   └── ...
    │
    ├── missing1/
    │   ├── video_missing1_A_001/
    │   │   ├── frame_000000.jpg
    │   │   ├── frame_000001.jpg
    │   │   └── ...
    │   ├── video_missing1_A_001_events.csv
    │   ├── video_missing1_B_001/
    │   │   ├── frame_000000.jpg
    │   │   └── ...
    │   ├── video_missing1_B_001_events.csv
    │   └── ...
    │
    ├── missing2/
    │   ├── video_missing2_A_001/
    │   │   ├── frame_000000.jpg
    │   │   ├── frame_000001.jpg
    │   │   └── ...
    │   ├── video_missing2_A_001_events.csv
    │   ├── video_missing2_C_001/
    │   │   ├── frame_000000.jpg
    │   │   └── ...
    │   ├── video_missing2_C_001_events.csv
```

```
│   └── ...
│
└── idle/
    ├── video_idle_001/
    │   ├── frame_000000.jpg
    │   ├── frame_000001.jpg
    │   └── ...
    ├── video_idle_001_events.csv
    └── ...
```

- 약 200~300개 영상 클립
- MediaPipe & YOLO로 Landmark/BBOX 자동 추출

---

2) 데이터 처리

- 데이터 가공 process
  0. 영상 -> 프레임 변환
  1. 이벤트 플래그 -> 프레임 라벨로 변환
     - 이벤트 플래그를 학습 가능한 프레임별 라벨링으로 변환함
     - 이벤트 플래그로 학습시 데이터 불균형으로 인한 학습의 어려움이 발생.
  2. 랜드마크 추출(MediaPipe 사용)
     - google의 MediaPipe를 이용하여 손의 각 관절부의 랜드마크를 추출하여 저장함.
  3. Dataset 구성
     - 촬영시 10회를 묶음으로 촬영을 진행하여 학습시에 세트 단위를 유지하며 분할 할 수 있도록 함.

---

3) 모델 선정

① TCN (Temporal Convolutional Network)

설명

- TCN은 시계열(sequence) 데이터를 처리하기 위해 고안된 1D Convolution 기반 신경망 구조이다.
- RNN 계열(LSTM, GRU)에 비해 병렬 처리 효율이 높고, 장기 의존성(long-term dependency)을 더 안정적으로 학습할 수 있다.
- Dilated Convolution을 사용하여 긴 시퀀스를 빠르게 처리할 수 있으며, Temporal CNN 구조로 행동 인지(gesture/action recognition)에 널리 활용된다.

선정 이유

- 본 프로젝트는 포즈 랜드마크(관절 좌표) 의 시간적 패턴을 학습하여 행동 단계를 분류하는 것이 핵심이므로 적합한 구조이었다.

- LSTM보다 학습이 빠르고, overfitting이 적으며, baseline 구조에서도 안정적인 성능 (0.7152) 을 보여주었다.
- 시계열 기반의 행동 단계를 분류하는 데 CNN 기반의 구조가 효과적임을 확인할 수 있었다.

활용

- MediaPipe Pose/Hands 등에서 추출한 프레임 단위 랜드마크 시퀀스 → 윈도우 단위로 입력하여 행동 클래스(약넣기, 닫기, Idle 등)를 분류하는 데 사용.

---

② MLP + Temporal Average Pooling (Baseline 경량 모델)

설명

- MLP(Multilayer Perceptron)에 Temporal Average Pooling 방식을 결합한 단순한 구조.
- 시퀀스 전체 길이를 평균 pooling하여 대표 벡터를 만들고, 이를 MLP로 분류하는 방식.

선정 이유

- 가장 단순한 baseline 모델을 구성하여 다른 모델들과 비교할 기준값을 만들기 위해 도입.
- 구조가 매우 가볍고 학습이 빠르기 때문에 초기 실험용으로 적합.
- 그러나 시계열 정보를 충분히 학습하지 못하는 구조적 한계가 있다.

활용

- 비교 baseline용으로 사용하여 “시계열 모델 적용의 필요성”을 확인하는 데 활용하였다.
- 성능은 가장 낮은 0.4335로 시계열 패턴이 중요한 업무에는 부적합하다는 결론 도출.

---

③ 1D CNN Classifier

설명

- Conv1D 기반의 시계열 처리 모델로, 짧은 temporal dependency를 효율적으로 학습한다.
- LSTM보다 빠르고, TCN보다 더 단순한 구조로 구성할 수 있다.

선정 이유

- baseline TCN보다 **더 단순한 CNN 기반 구조로 어느 정도 성능이 나오는지 비교**하기 위한 목적.
- 포즈의 연속적인 움직임 패턴을 convolution filter가 잘 감지하는지를 검증하기 위한 모델.
- 실제 baseline TCN 보다 약간 더 높은 **0.7253** 성능을 내며 효과적인 대안임을 입증.

활용

- 포즈 좌표 시퀀스를 입력하여 단기 행동 패턴 인식에 활용.
- TCN 대비 더 경량화된 대안으로 비교 실험 수행.

---

④ BiLSTM Classifier

설명

- 양방향 LSTM(BiLSTM)은 시퀀스를 앞 방향 + 뒤 방향으로 모두 처리하여 더 풍부한 시계열 정보를 학습한다.
- 자연어 처리나 장기 의존성이 중요한 시퀀스 문제에 자주 사용된다.

선정 이유

- 행동 인지와 같은 시계열 데이터에서 RNN 기반 모델 성능을 확인하기 위해 적용.
- LSTM이 temporal smoothing 효과가 있기 때문에 잡음이 많은 랜드마크 데이터에서 강점이 있을 것으로 예상.

활용

- 포즈 시계열 전체를 입력하여 행동 단계를 분류.
- 성능은 **0.6836** 수준으로 CNN 기반 모델 대비 다소 낮았으나, RNN 계열의 특성을 확인하는 데 의미 있음.

---

⑤ TCN (개선된 하이퍼파라미터 버전)

설명

- 기존 TCN을 기반으로 채널 수, kernel size, dropout, window 크기 등 주요 하이퍼파라미터를 조정한 버전.
- baseline 대비 expressiveness와 regularization이 개선된 구조.

선정 이유

- baseline TCN에서 우수한 성능을 보였기에, 더 최적화된 설정을 탐색하여 성능 극대화를 목표로 개선.
- 결과적으로 모든 Fold에서 성능이 향상되고 평균 **0.7416**으로 2위 성능 달성.

활용

- 개선된 구조를 통해 실제 현장의 더 다양한 행동 패턴을 안정적으로 처리 가능.
- 최종 후보 모델 중 하나로 평가됨.

---

⑥ CNN (개선된 하이퍼파라미터 버전)

설명

- 1D CNN의 필터 수, kernel size, stride, regularization 등을 수정한 개선 버전.
- 시계열 필터링 구조를 강화하여 행동 패턴 인식을 최적화한 형태.

선정 이유

- baseline 1D CNN이 우수한 성능을 보였기 때문에, CNN 기반 구조를 개선하여 최적 성능을 확보하려는 목적.
- 모든 Fold에서 성능 향상이 발생하며 **전체 모델 중 최고 val_acc = 0.7515** 기록.

활용

- 포즈 기반 행동 인식에서 CNN이 TCN 못지않게 강력한 모델임을 확인.
- 최종 후보 모델 중 가장 높은 성능을 보여 최종 모델로 선정할 가치가 높음.

---

4) 모델 학습

1. 행동 탐지
    - 사용한 모델
        - MLP + Temporal Average Pooling
            - 프레임 단위 특징(랜드마크)을 평균을 구하여 하나의 고정 길이 벡터로 만든 뒤
            - MLP(다층 퍼셉트론)으로 분류하는 단순 구조
            - 시간 정보가 사라지기 때문에 시계열 데이터 학습이 적절한지 비교용 모델
        - 1D CNN(Temporal Convolution)
            - 시간축을 따라 슬라이딩 커널(CN)로 패턴을 학습하는 모델
            - short-term 패턴(0.1~0.5초) 탐지에 강함
            - 멀리 떨어진 프레임간 의존성을 잘 잡지 못함
        - BiLSTM
            - LSTM을 앞 -> 뒤, 뒤-> 앞 두 방향으로 학습
            - 시간 순서 기반의 long-term dependency를 학습
            - 프레임 간 의미적 흐름을 파악함
            - 앞뒤 문맥을 모두 보며 학습을 함.
        - TCN(Temporal Convolutional Network)
            - Dilated Conv(팽장 합성곱)을 사용하여 긴 시간 의존성을 CNN 방식으로 학습
            - BiLSTM과 달리 병렬화가 가능하여 성능이 좋음
        - 최종적으로 TCN 모델을 선택하여 학습을 진행함.
2. 객체 탐지
    - Yolov8을 통한 객체탐지
        - 기능

- 박스 개수 감지
- 열린/닫힌 박스 수 감지
- 물건 있음/없음 감지
- 프레임 단위 로그 생성

---

5) 예측 및 후처리

- 각각의 단일 모델로 예측을 할 경우 정확한 값을 얻을 수 없음
  - TCN 행동 탐지 모델: 전반적으로 행동 위치는 맞으나,
    각 행동 구간별 끊기는 지점 + 오탐으로 인한 노이즈 등 파편화 된 데이터가 형성되어 있음
  - Yolo 객체 탐지 모델: 객체가 정상적으로 보인다는 가정 하에 압도적인 정확도를 보이나, 손, 장해물 등 객체 탐지가 안되는 상황 + 다른 객체 오탐 등으로 인해 안정적인 구간 예측이 힘듬
- 위의 두 모델의 단점을 서로 보완하여 예측 알고리즘을 구성
- 예측 알고리즘 간략 설명
  1. TCN 결과와 YOLO 결과를 프레임 단위로 합친 후
  2. TCN 에서 나온 이벤트 구간을 노이즈 보정 후 파악
  3. 각 구간 안에서 시작 지점과 끝 지점을 yolo 객체탐지 데이터를 통하여 구함
  4. 이벤트 플레그 작성 완료

6) 모델 평가 (CV 결과)

모델별 평균 val_acc

모델명	Fold 0 (acc)	Fold 1 (acc)	Fold 2 (acc)	Fold 3 (acc)	평균 v
TCN	0.603	0.798	0.769	0.691	**0.715**
MLP + Temporal AvgPooling	0.349	0.549	0.476	0.360	**0.433**
1D CNN Classifier	0.605	0.786	0.782	0.728	**0.725**
BiLSTM Classifier	0.584	0.776	0.730	0.644	**0.683**
TCN (개선 버전)	0.621	0.826	0.786	0.733	**0.741**
CNN (개선 버전)	0.639	0.837	0.796	0.733	**0.751**

결론

- **TCN 모델**
  - 시퀀스를 길게(60프레임 이상) 볼 수 있을 때 성능이 크게 향상됨
  - 높은 FPS(30fps 이상) 환경에서 실제 작업 흐름을 잘 반영함
  - 정상적인 촬영 환경에서는 **TCN이 더 적합한 모델**
- **CNN 모델**
  - 현재 데이터는 7.5fps로 프레임 수가 매우 낮음
  - CNN은 짧은 구간(약 6~20프레임)에서도 동작 가능해 **낮은 프레임 환경에서 유리**
  - 우리의 데이터 환경에서는 **CNN이 상대적으로 더 높은 성능을 보여줌**
- **현재 상황 요약**
  - 실제 의도된 환경(고 FPS, 충분한 프레임 수)에서는 **TCN이 정답에 가까운 모델**
  - 하지만 **현재 수집된 프레임 수가 적은 환경(20프레임 수준)**에서는

    CNN 모델이 더 적합하고 더 좋은 성능을 냄

⇒ 이론적·실제 프로세스 기반으로는 TCN이 더 적합한 모델이지만 현재 데이터 수집 환경(FPS 부족)에서는 CNN이 최적의 선택이다.

---

7) 시연 영상

attachment:259a68ed-fdf0-472c-a35e-93bdc7d0082e:최종영상.mp4

---

5. 프로젝트 평가

한계점 및 개선점

- **실제 HACCP 현장과 환경 차이**
  - 현장과 촬영 환경이 달라 일반화 성능 저하 가능성 존재
- **외부 학습 자료 부족**
  - 유사 사례 및 참고 데이터 부족으로 서비스 기획 및 모델 검증 범위 제한
- **데이터 수집 환경 문제 :**
  - 목표 30fps → 실제 7.5fps 로 수집

    - 윈도우당 프레임 수 15장으로 감소
    - CNN은 6~7프레임, TCN은 60프레임 이상을 요구하므로
      → CNN 기반 모델 성능에 불리하게 작용
    - 정상적인 FPS 확보 시 CNN 성능 향상 가능성 높음
- **물리적인 시간 제한 :**
    - 총 약 280개(1분 영상 기준) 수집
    - 최소 가정한 7개 이슈 학습에는 매우 적은 수준
    - K-Fold 마다 학습 편차 심함 → 데이터 부족 영향 명확
    - 데이터 증대 시 더 높은·안정적인 성능 기대
- **서비스 개선·피드백 반영 시간 부족**

향후 개선 계획

- 알람 기능 추가
- 관리자
- 작업자 화면 고도화
- 실시간 행동 탐지
- 손·도구에 의한 가림(occlusion) 문제
- 상자 크기 조정(작업 공간 확보)
- 부분 가림 데이터(Augmentation) 추가
- 필터/전처리 활용한 객체 탐지 강화
- 다양한 촬영 각도 확보
- 다양한 환경(조명·배경)에서 데이터 수집
- 일반화 성능 개선
