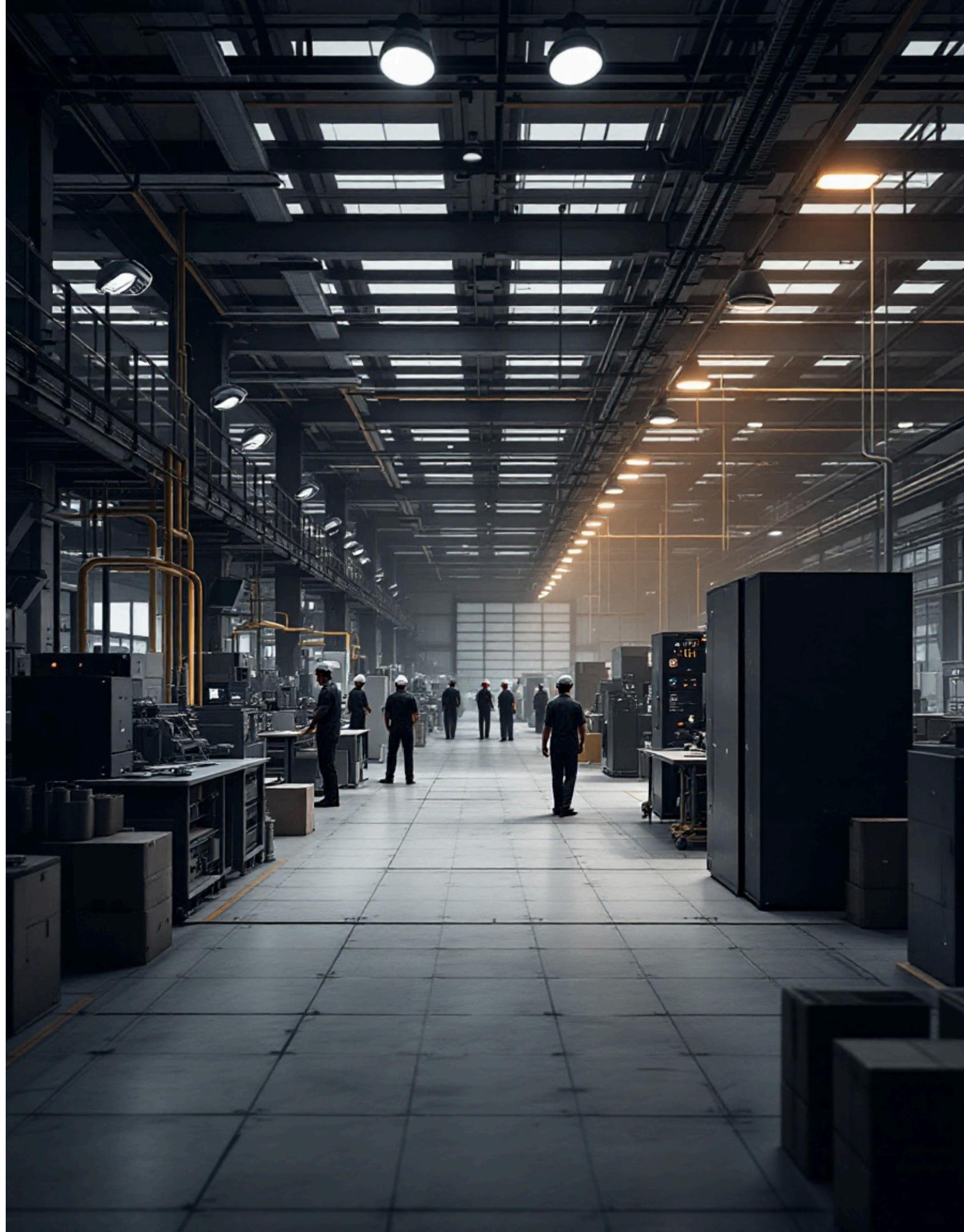


Industrial Safety Ai CCTV (ISAC)

조명 : TOST(Team Of Safe T)

조원 : 박경규(팀장), 김범규, 양영준, 장해준



CONTENTS

- 1 주제
- 2 배경 및 문제 정의
- 3 목표 및 기대 성과
- 4 시스템 구성도
- 5 과정
- 6 결과
- 7 시연
- 8 고찰
- 9 QnA



01 주제

산업 재해의 예방을 위한 CCTV

산업 현장에서 사고와 재해의 발생을 CCTV를 통해 Ai 기반으로 실시간 감지가 가능하도록 하는 프로그램.

- 사람의 낙상 및 넘어짐
- 화재
- 도움 요청 신호
- 작업 중 소화기의 유무



02

배경 및 문제정의

산업 현장 추락사고 예방대책 시급

기호일보 | 입력 2024.09.12 | 19면 | 댓글 0



한국은 OECD 국가 중 산재사망률 1위를 기록하다가 근자에 들어 그나마 3위로 내려갔다 한다. 그래도 여전히 '산재 왕국'이라는 오명에서 벗어나지 못하고 있는 우리다. 어제도 오늘도 산업현장 곳곳에서 근로자들이 작업 도중 산업재해로 다치거나 숨지고 있다.

인천 유치원 보수공사 현장에서 60대 근로자가 떨어져 숨지는 사고가 발생했다는 소식이다. 사고는 근로자가 외벽에 설치된 비계에 올라 마감재 보수작업을 하다 7.5m 아래로 떨어졌다 는 것이다. 이보다 앞서 인천시 남동구 민간 재활용 처리장에서 50대 근로자가 집게차 위에서 2.3m 아래로 추락해 숨졌으며, 서구 종이상자 제조공장에서는 60대 근로자가 화물차에 물건을 싣는 작업을 하던 중 2m 아래로 추락해 치료 중 숨지기도 했다는 것이다.



GDP 대비 제조업 비중(단위=%)



03

목표 및 기대성과



05 과정

1. 협업과정(github/notion 생성)

 IndustrialSafetyAiCctv (Public)

Watch

main · 15 Branches · 0 Tags · Go to file · Add file · Code

creator928 · Update ISAC_main_UI.py · 3 hours ago · 41 Commits

ISAC · Update ISAC_main_UI.py · 3 hours ago

README.md · New line(Update README.md) · last week

how-to-cowork.txt · update how to cowork · last week

yolo11s.pt · ui alert func update · yesterday

README

Project Industrial Safety A.I CCTV(ISAC)

Team Name : TOST(Team of Safe T)

- 산업 현장에서 발생할 수 있는 다양한 위험 상황을 실시간으로 탐지하고, 구조 활동을 지원하는 감시 시스템을 구축하는 프로그램 개발 프로젝트.

주요 기능

- 위험 상황 탐지 및 구호 지원 기능
 - AI 모델을 사용하여 작업자 기절 등의 위험 상황 감지
 - Pose Estimation 기술을 사용하여 인근 구조자의 구조 신호를 실시간 감지
 - 쓰러진 작업자의 헬멧 색상을 기반으로 미리 등록된 건강 정보를 불러오며, 이를 통해 초기 대응 및 구급 활동에 필요한 정보 제공
- 위험 발생 요소 모니터링
 - 특정 위험 작업 상황에서 소화기 등 필수 오브젝트의 존재여부를 판단하는 기능 제공
 - 영상 분석을 통해 화재 및 연기를 실시간으로 탐지하고, 재난 상황을 신속하게 파악
- 위험 요소 통합 관리 지원
 - 통제실 및 관리자에게 탐지된 정보 실시간 전달 및 대응 촉구



Industrial Safety AI Cctv (ISAC)

• 산업 현장에서 발생할 수 있는 다양한 위험 상황을 실시간으로 탐지하고, 구조 활동을 지원하는 감시 시스템을 구축하는 프로그램 개발 프로젝트.

프로젝트 이름: "ISAC"

University Funding

일정

기능

▶ 구현 기능

회의록

회의

발표

▶ 특정한 분기별 정리도 추천합니다.

작업

◆ 작업

Git Hub

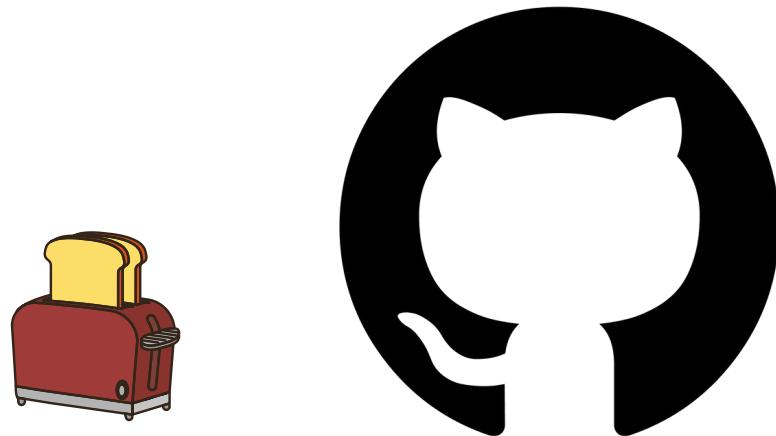
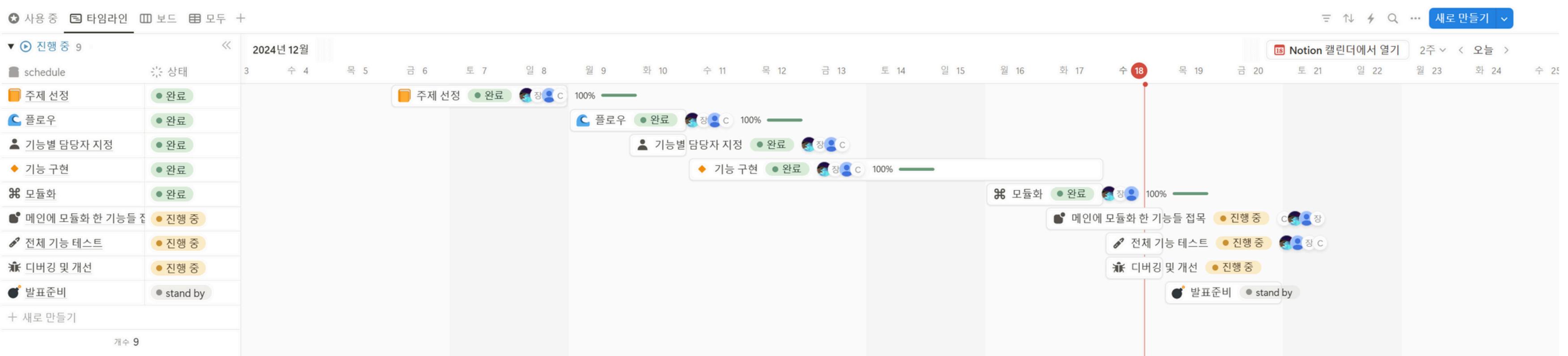
GitHub - creator928/IndustrialSafetyAiC...
Inteldx team project 01 - Team TOST(Team Of Safety) - creator928/IndustrialSafetyAiCctv...
https://github.com/creator928/IndustrialS...

백

GitHub 임베드

05 과정

2. 프로젝트 진행 일정 및 계획 수립



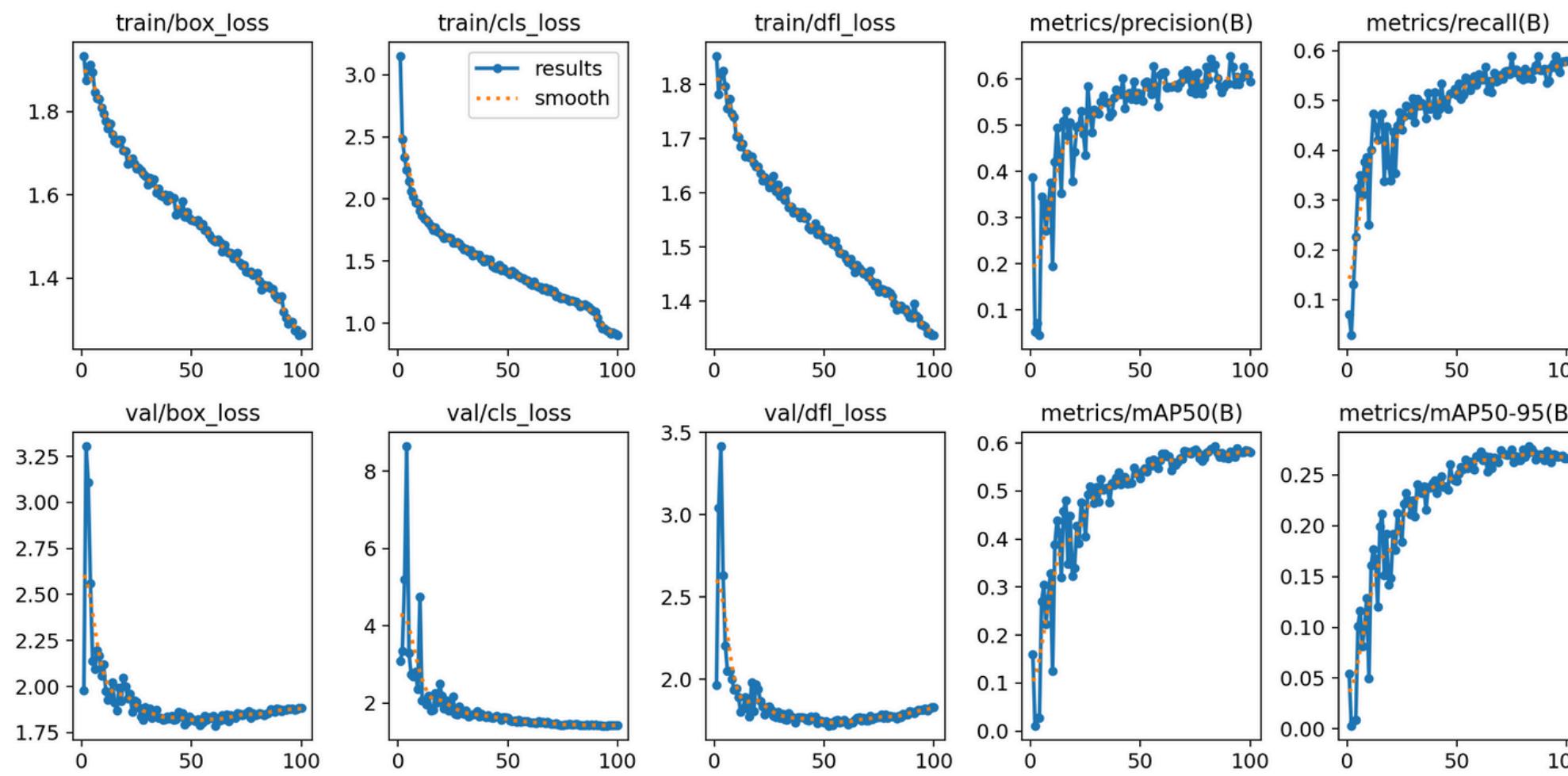
Milestone



Intel DX Team Project

05 과정

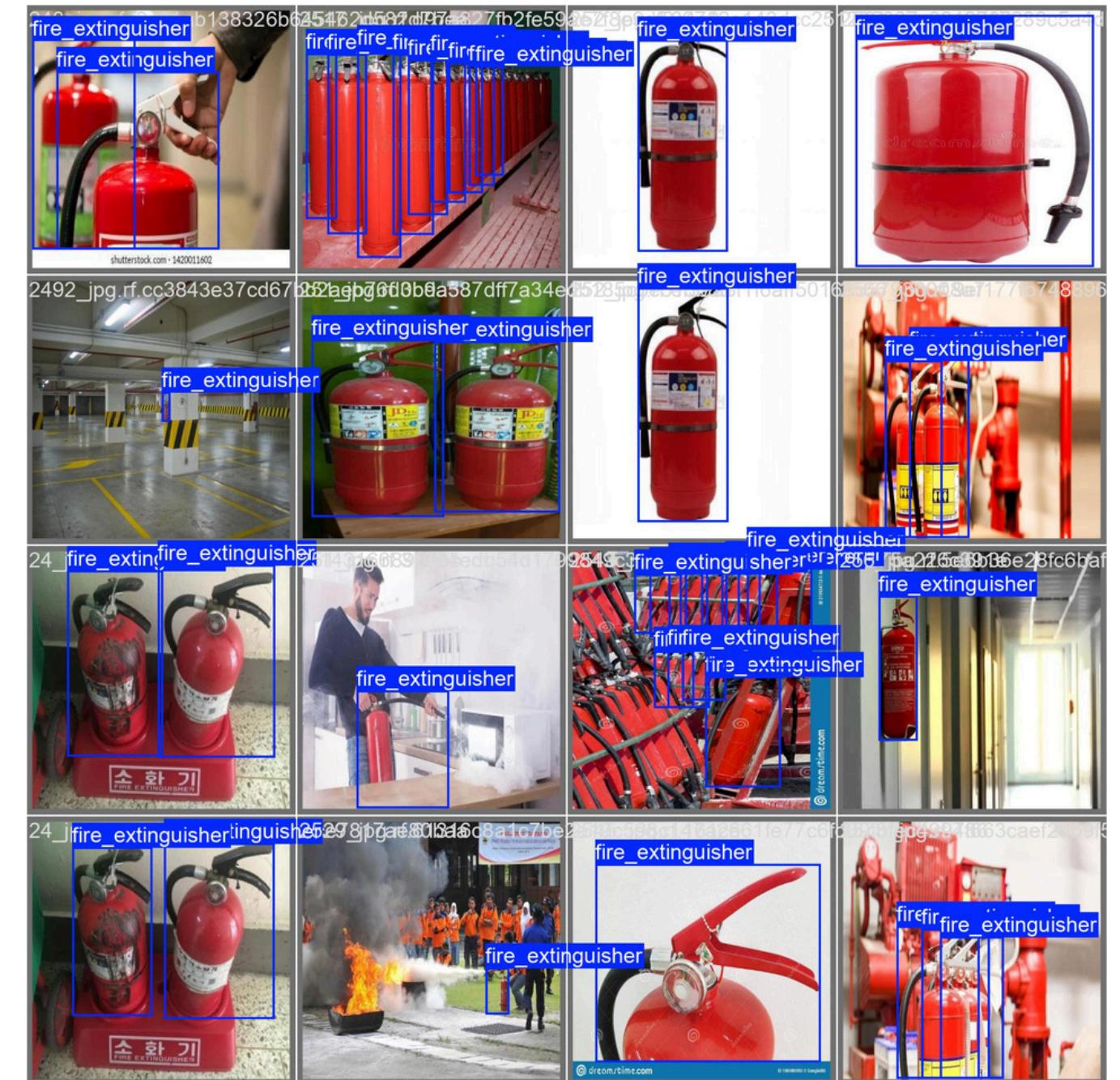
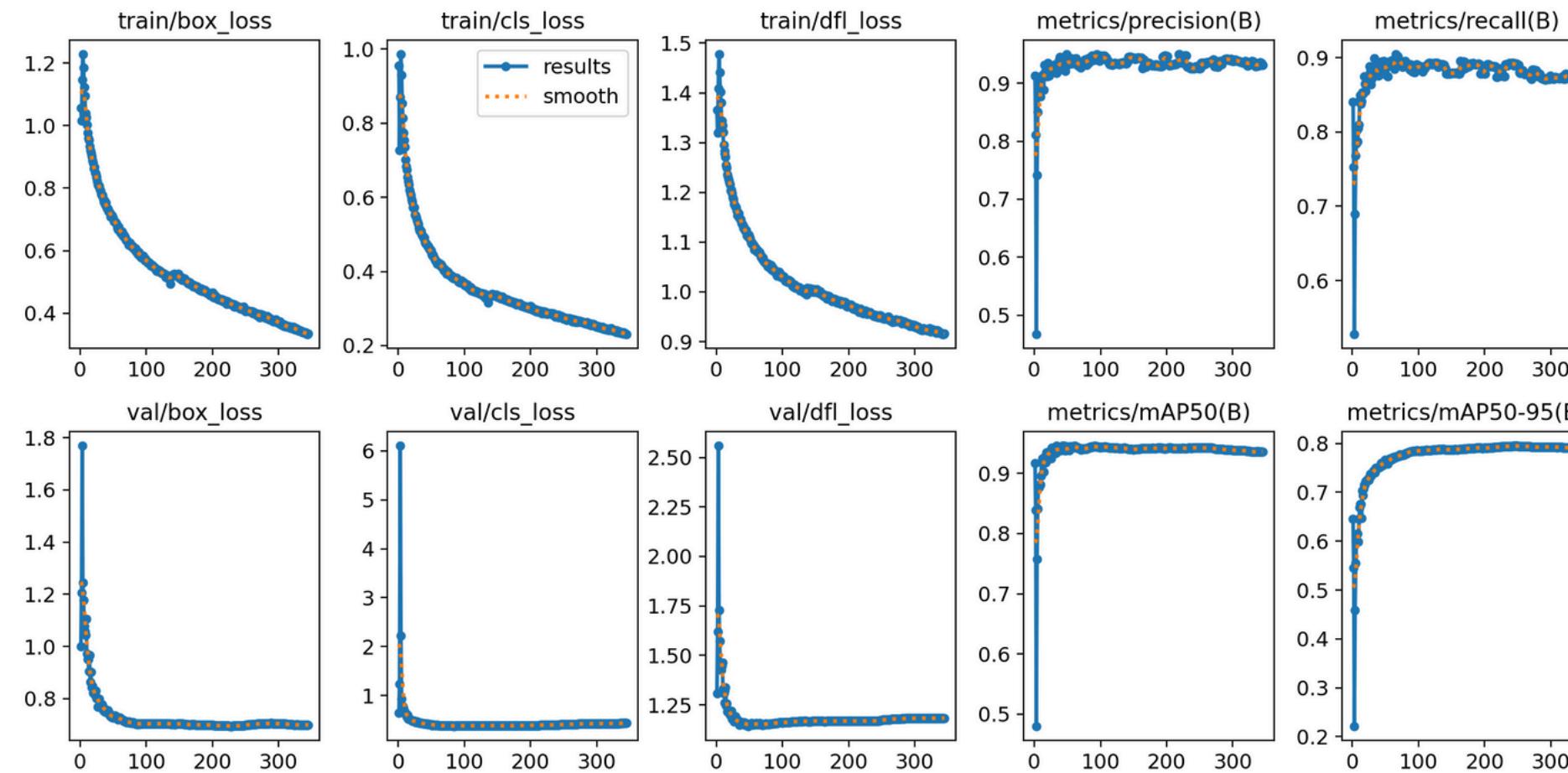
3. 모델 학습을 통한 모델 생성



05

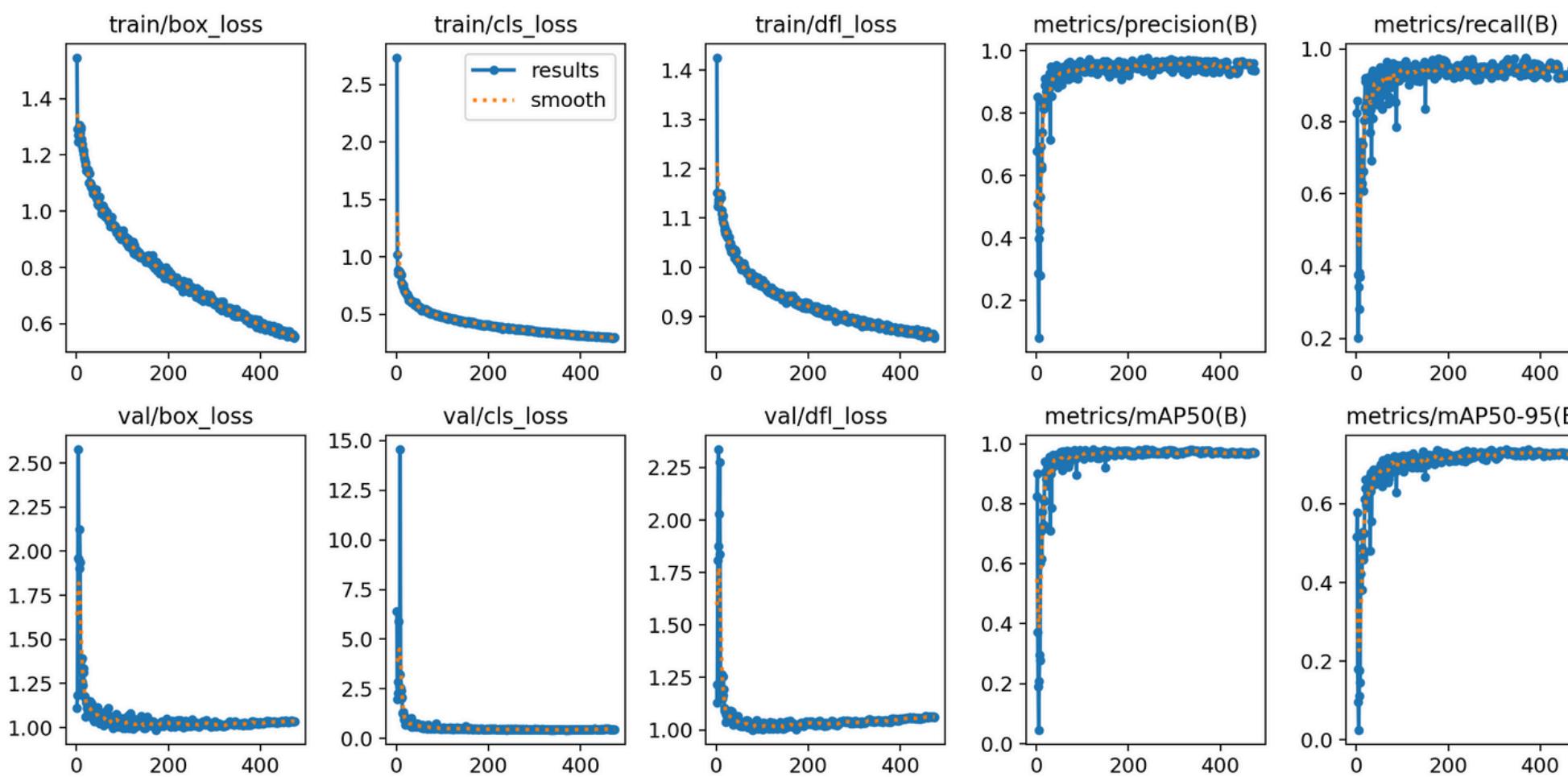
과정

3. 모델 학습을 통한 모델 생성



05 과정

3. 모델 학습을 통한 모델 생성



05

과정

4. 기존 코드 모듈화



주간 아침 회의



팀명 TOST (Team Of Safe T)



세부 회의



2024년 12월 16일



양영준 PKG 장해준 creator928forgit@gmail.com



비어 있음



속성 4개 추가



댓글 추가



코드 구조 예시

```
# 예시 코드 구조

class FallDetect:
    def __init__(self):
        self.model = self.load_model()

    def load_model(self):
        # 모델 로딩 로직
        pass

    def detect(self, frame):
        """
        낙상 감지 기능을 수행합니다.

        Args:
            frame: numpy array 형태의 입력 이미지

        Returns:
            detected_frame: bbox가 그려진 출력 이미지
            bbox_list: [[x1,y1,x2,y2,conf,class_id], ...]
        """
        detected_frame = frame.copy()
        bbox_list = []
        # 감지 로직
        return detected_frame, bbox_list
```

모듈 사용 방법

```
# 메인 코드에서 사용 예시
from fall_detect import FallDetect

detector = FallDetect()
result_frame, bboxes = detector.detect(input_frame)
```

05

과정

4. 기존 코드 모듈화



```
# 진짜번호
first_time = time.time()
count_dict = {} # 각 사람별 count를 저장할 딕셔너리
maintime_dict = {} # 각 사람별 maintime을 저장할 딕셔너리

# 모델 로드
model = YOLO('yolov8s-pose.pt') # 사용하려는 모델 경로 입력

# 웹캡처 시작
cap = cv2.VideoCapture(0) # 0번 웹캠 사용

if not cap.isOpened():
    print("Error: Could not open webcam.")
    exit()

while True:
    ret, frame = cap.read()
    if not ret:
        print("Error: Could not read frame.")
        break

    # 모델을 사용하여 프레임 추론
    results = model.track(frame, verbose=False, persist=True)

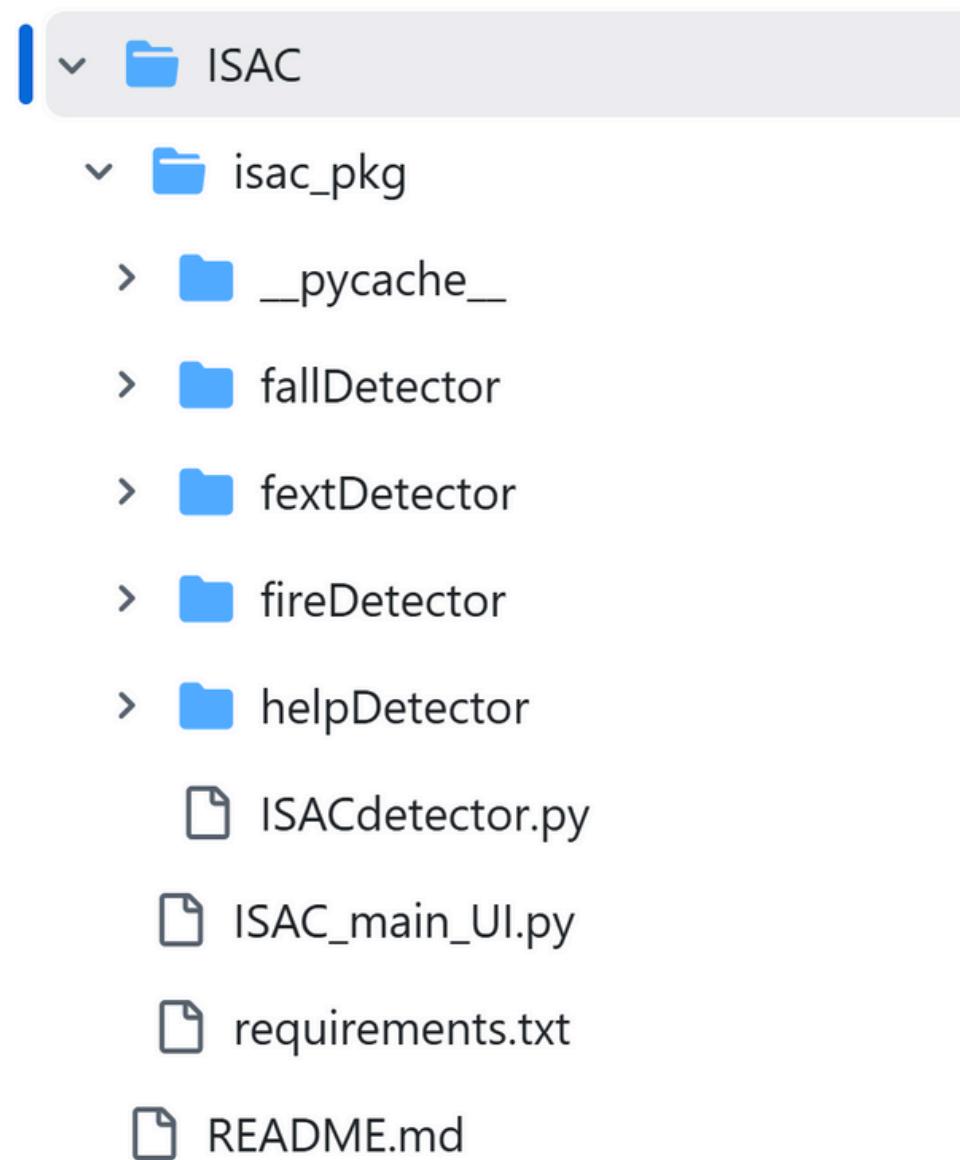
    # 추론 결과에서 포즈 데이터 그리기
    annotated_frame = results[0].plot() # YOLOv8의 plot 기능으로 그리기

    # 모든 사람의 키포인트 처리
    for person_idx, keypoints in enumerate(results[0].keypoints):
        # ----- 키 포인트를 가져와 리스트로 변경 -----
        kxy = keypoints.xy[0] # 키포인트 가져옴
        kxy_list = [(row[0].item(), row[1].item()) for row in kxy] # 리스트로 변환

        # ----- 키 포인트들과 딕셔너리로 합성 -----
        body_parts = {
            0: "코 (Nose)",
            1: "왼쪽 눈 (Left Eye)",
            2: "오른쪽 눈 (Right Eye)",
            3: "왼쪽 귀 (Left Ear)",
            4: "오른쪽 귀 (Right Ear)",
            5: "왼쪽 어깨 (Left Shoulder)",
            6: "오른쪽 어깨 (Right Shoulder)",
            7: "왼쪽 팔꿈치 (Left Elbow)",
            8: "오른쪽 팔꿈치 (Right Elbow)",
            9: "왼쪽 손목 (Left Wrist)",
            10: "오른쪽 손목 (Right Wrist)",
            11: "왼쪽 엉덩이 (Left Hip)",
            12: "오른쪽 엉덩이 (Right Hip)",
            13: "왼쪽 무릎 (Left Knee)",
            14: "오른쪽 무릎 (Right Knee)",
            15: "왼쪽 발목 (Left Ankle)",
            16: "오른쪽 발목 (Right Ankle)"
        }

        # 키포인트 딕셔너리 생성
        kepoint_dict = {i: {"body_part": body_parts[i], "x,y": kxy_list[i]} for i in range(len(kxy_list))}

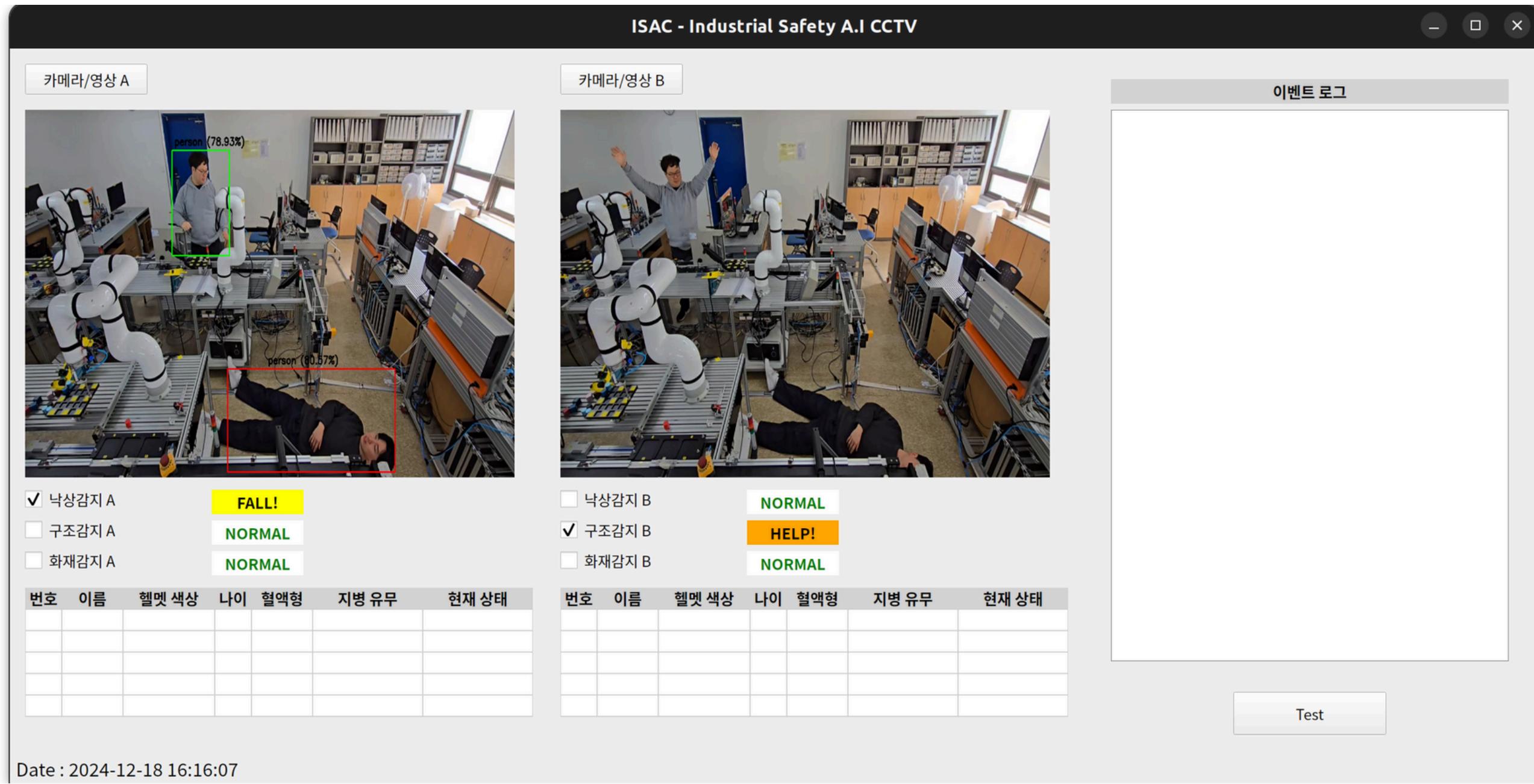
        # 코 좌표
        if 0 in kepoint_dict and kepoint_dict[0]["x,y"] != (0, 0):
            nose_coordinates = kepoint_dict[0]["x,y"]
```



05

과정

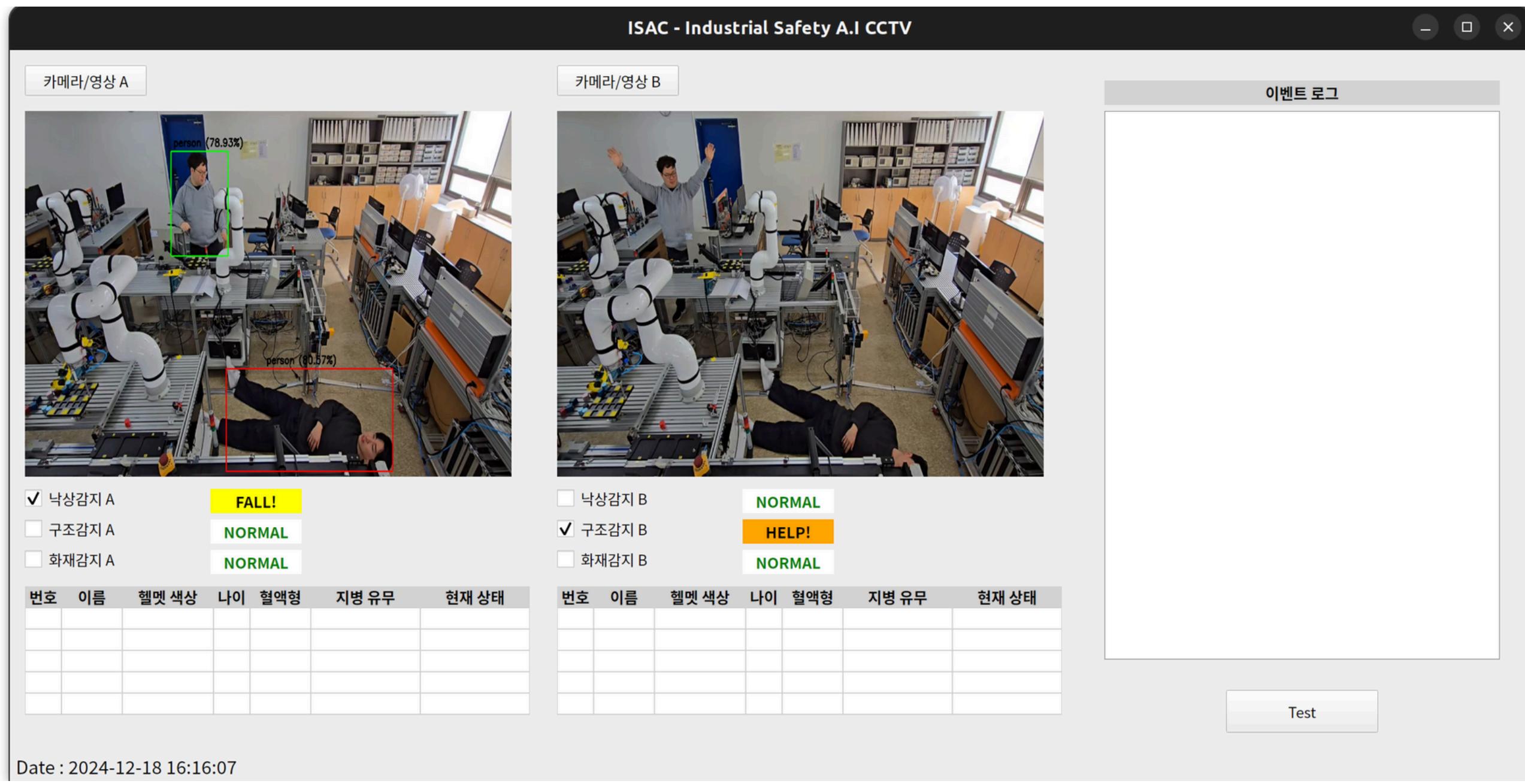
5. UI 구상도



05

과정

5. 기능구현 및 GUI를 통한 시각화



06

결과



07

시연



08

고찰





THANK YOU

감사합니다