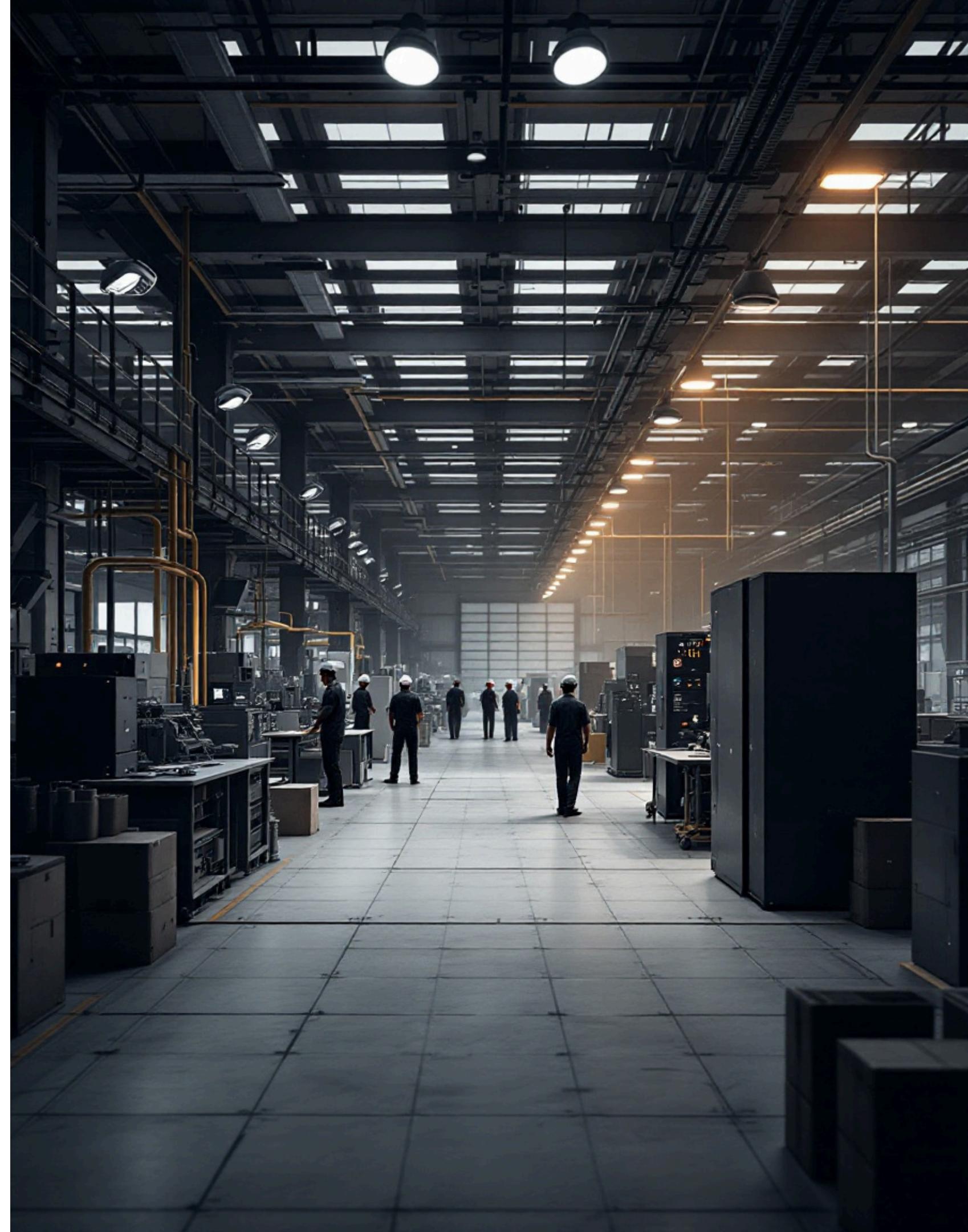


Industrial Safety Ai CCTV (ISAC)

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 안전 AI CCTV

산업 현장에서 사고, 재해의 발생을 AI 통해 실시간으로 감지가 가능하도록 하는 프로그램.

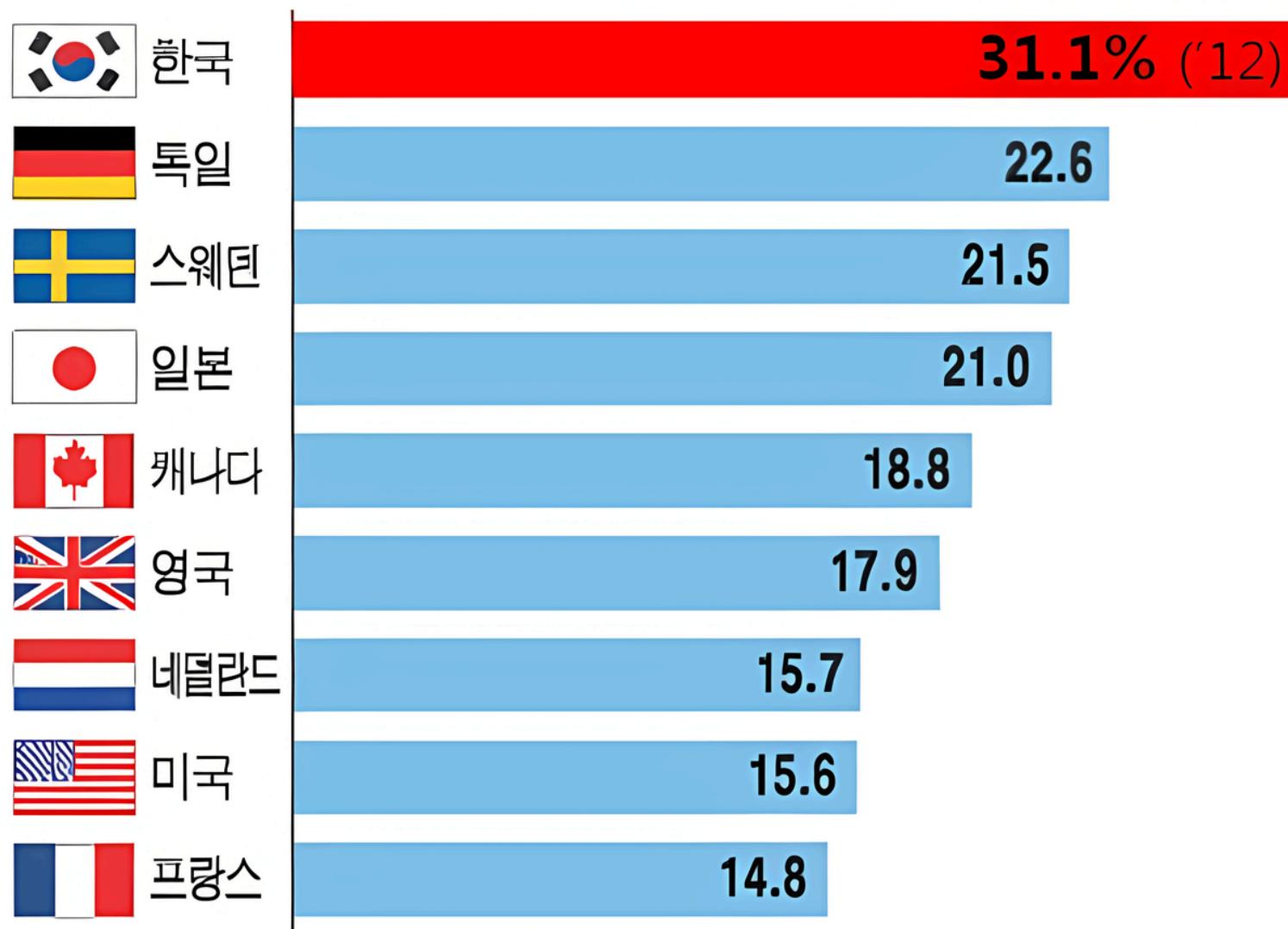
- 사람의 낙상 및 넘어짐
- 화재
- 도움 요청 신호
- 작업 중 소화기의 유무



02

배경 및 문제정의

GDP 대비 제조업 비중(단위=%)



산업 현장 추락사고 예방대책 시급

咯 기호일보 | 0 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

서울파이낸스 [창간22/위험천만 건설현장上] '안전' 쟁진다지만…곳곳이 사각지대

한국은 경제협력개발기구(OECD) 국가 중 산업재해(산재) 사망률 1위다. 이중에서도 최다 사고율을 기록하는 산업군은 단연 건설업이다. 실제 건설현장에서는 하루가...

2024. 9. 16.

순천광장신문 오늘 하루도 7~8명의 노동자, 일하다가 죽어

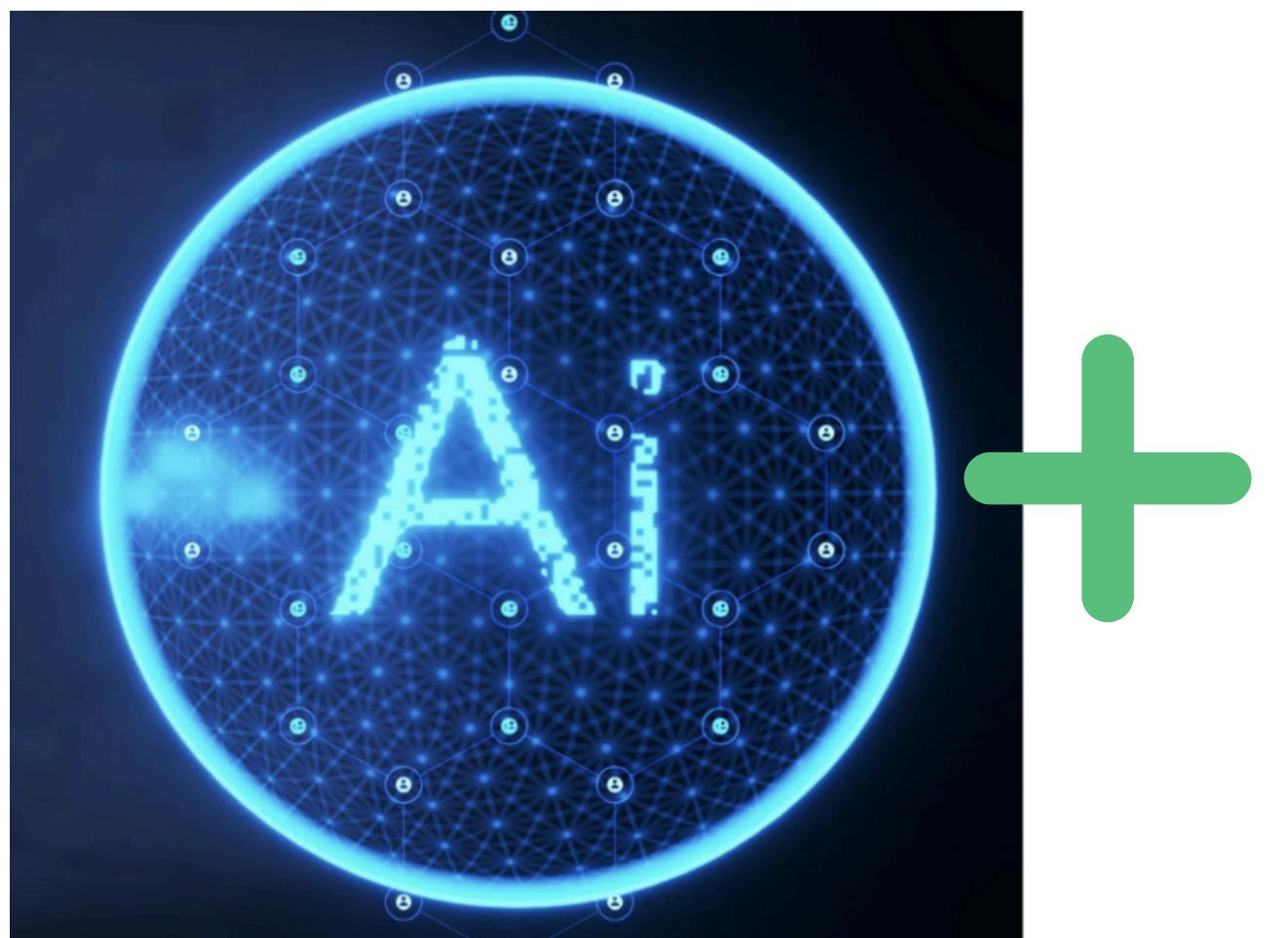
정부가 집계하는 산재사망률이 있다. 일과 관련하여 노동자 1만명당 몇 명이 죽었느냐가 기준이다. 정부는 2023년 산재사망률을 0.98명으로 발표했다.

2024. 7. 29.



03

목표



03

목표



Safety

센서 / 장비 / 영상의 조합으로 산재사고 미연에 방지



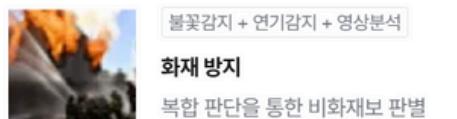
장비 안전관리

작업중 사람 진입 시 알람 및 장비 자동 멈춤



가스누출 감지

유해가스 누출 감지



화재 방지

복합 판단을 통한 비화재보 판별



장비이상 방지

장비이상으로 인한 사고예방

Health

근로자 관리 및 작업환경을 체계적으로 관리



안전장구 착용

안전장구 미착용 근로자 관리



위험작업 투입근로자 관리

위험구역 작업 근로자 관리



사업장 내 건강관리

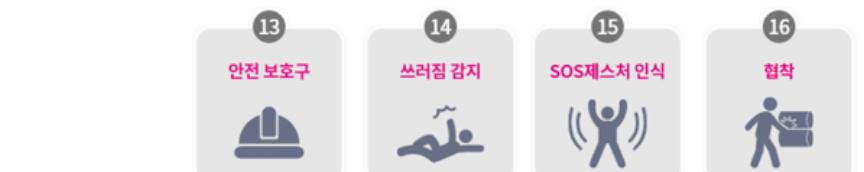
사업장내 건강 저해요소 관리



기본 기능



추가 기능



현장 요구사항 확인 후 커스텀 기능 구현

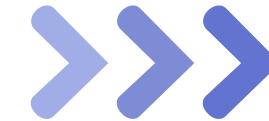
※ 12개의 영상분석 기본기능과 현장 안전사고 예방을 위한 4개의 추가기능을 제공하고,
현장의 요구사항을 반영한 업종별 특화 커스텀 기능 제공 가능

Device
Feedback



03

목표

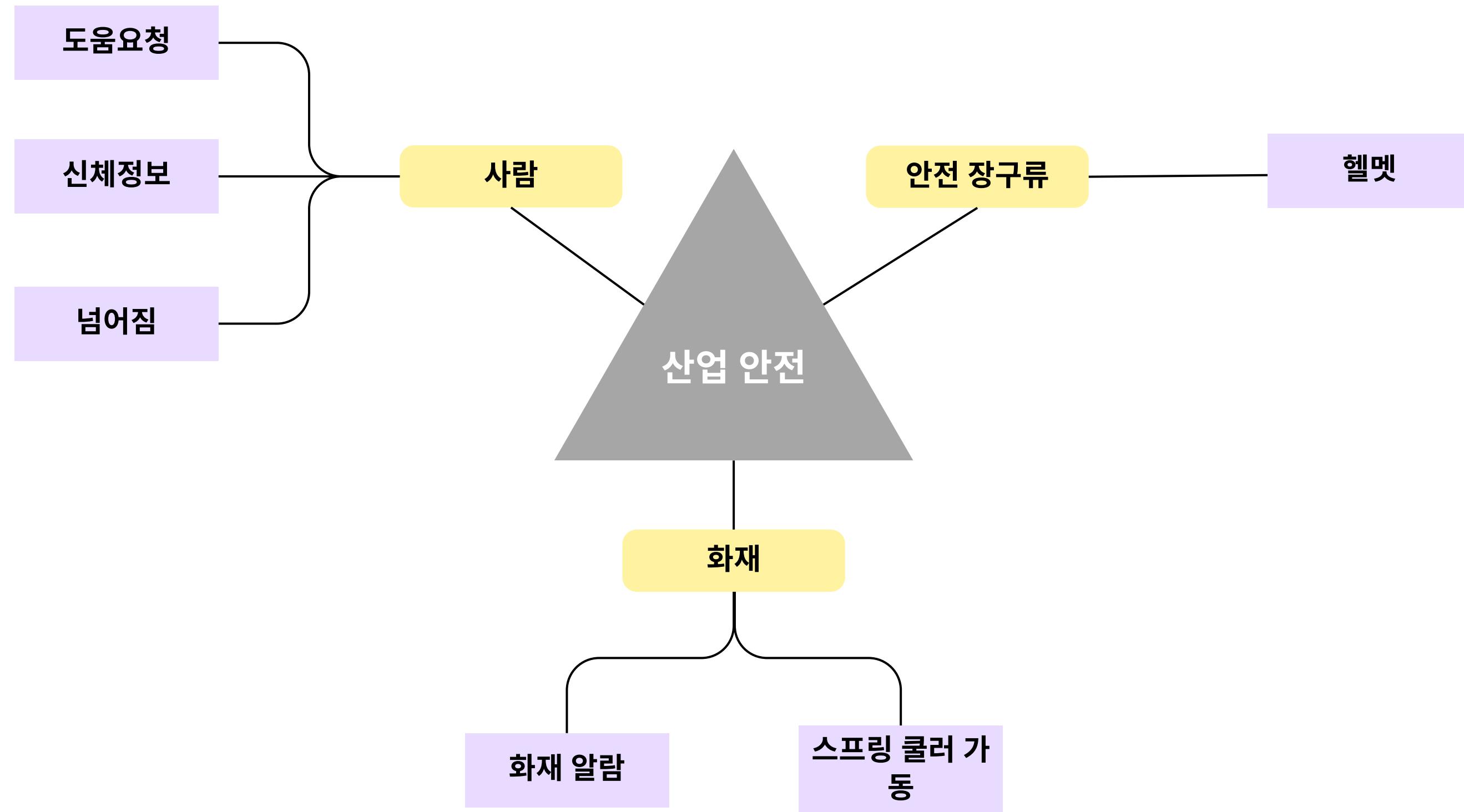


Device
Feedback



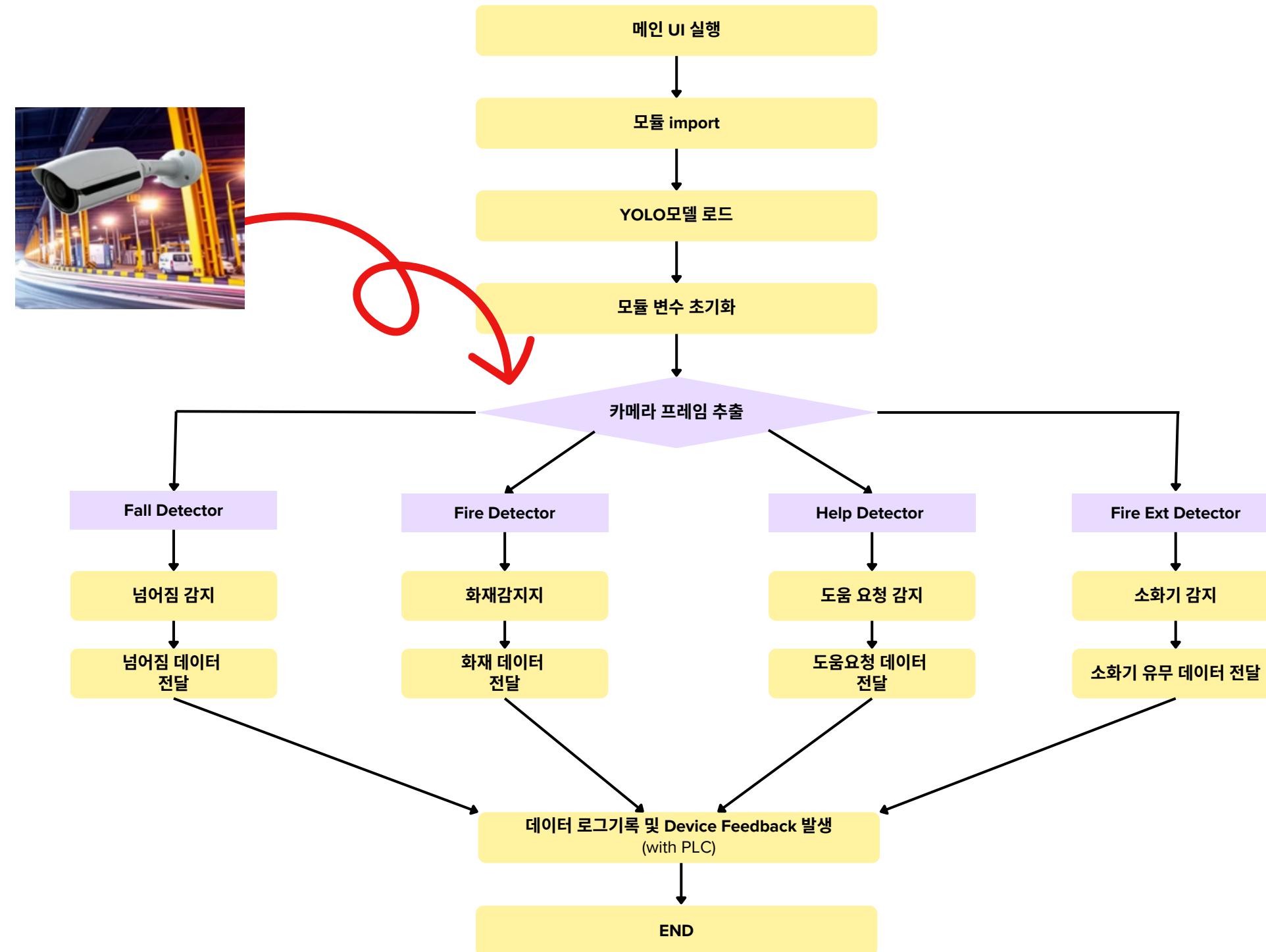
04

시스템 구성도 (안전관리 요소)



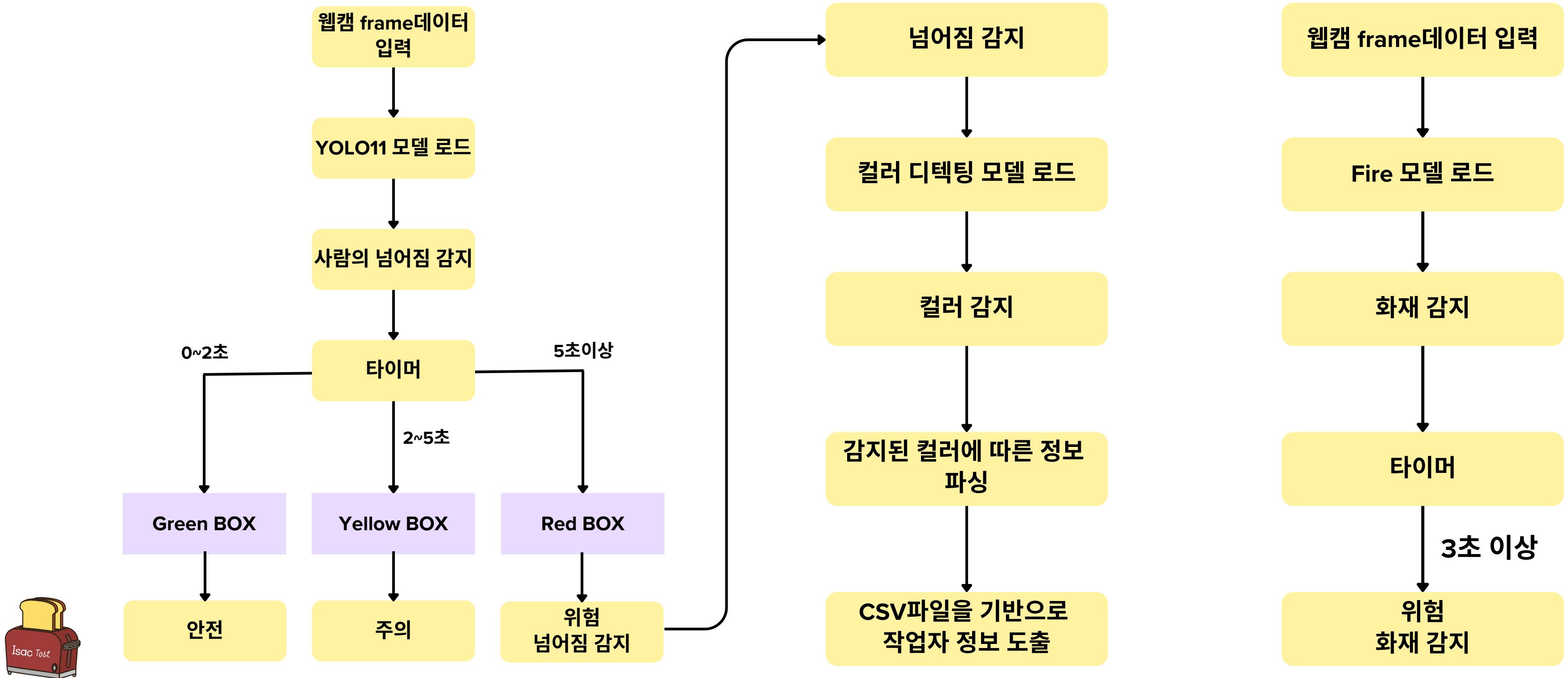
04

시스템 구성도(플로우 차트)



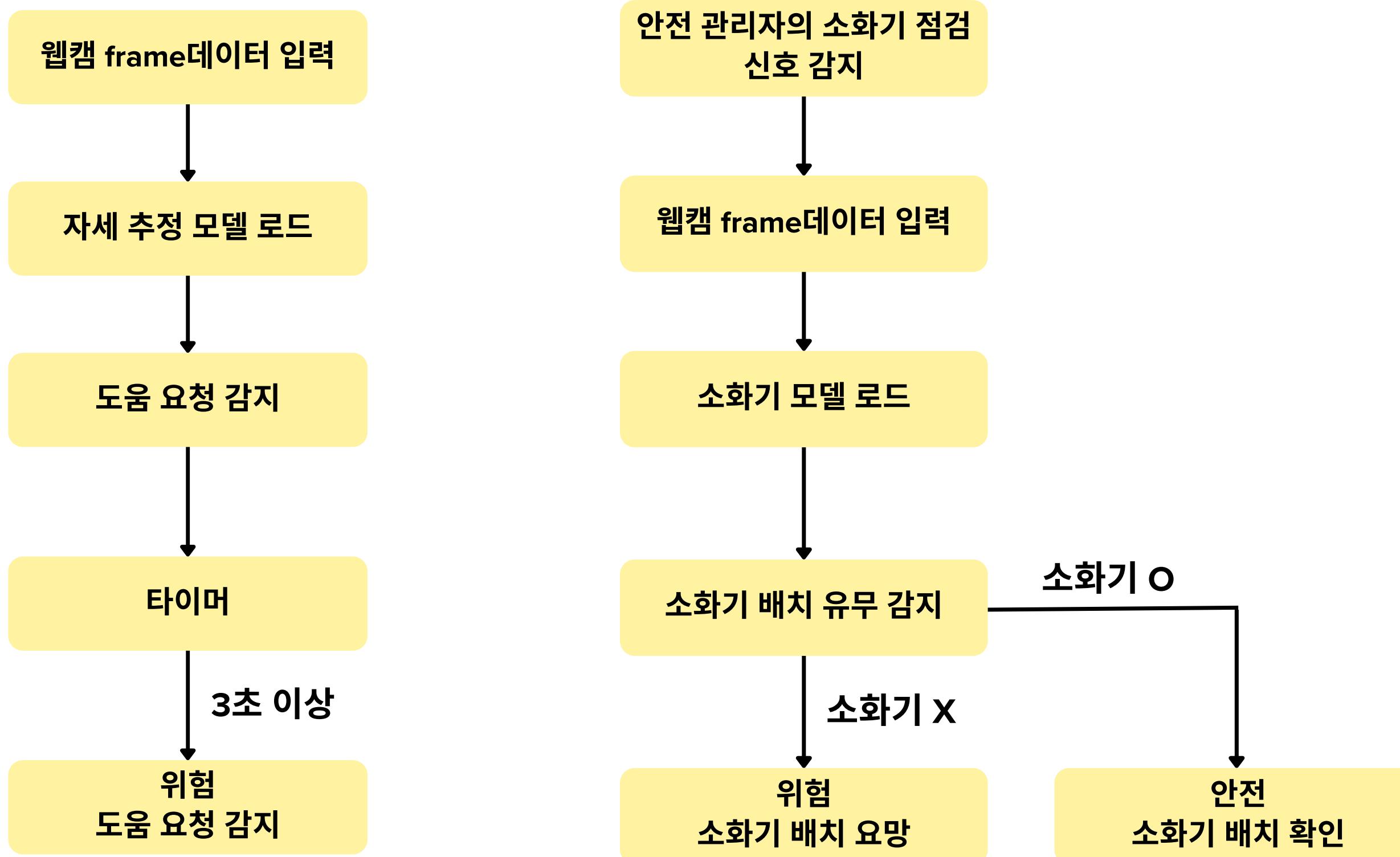
04

시스템 구성도(Fall/Fire 플로우 차트)



04

시스템 구성도 (Help/Fire_Ext 플로우 차트)



05 과정

1. 협업과정(github/notion 생성)

 IndustrialSafetyAiCctv Public



 main  15 Branches  0 Tags  Go to file  Add file  Code

 creator928 Update ISAC_main_UI.py  b8b24a7 · 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

 Isac Tost



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 SafeT) - creator928/IndustrialSafetyAiCctv  <https://github.com/creator928/IndustrialS...>

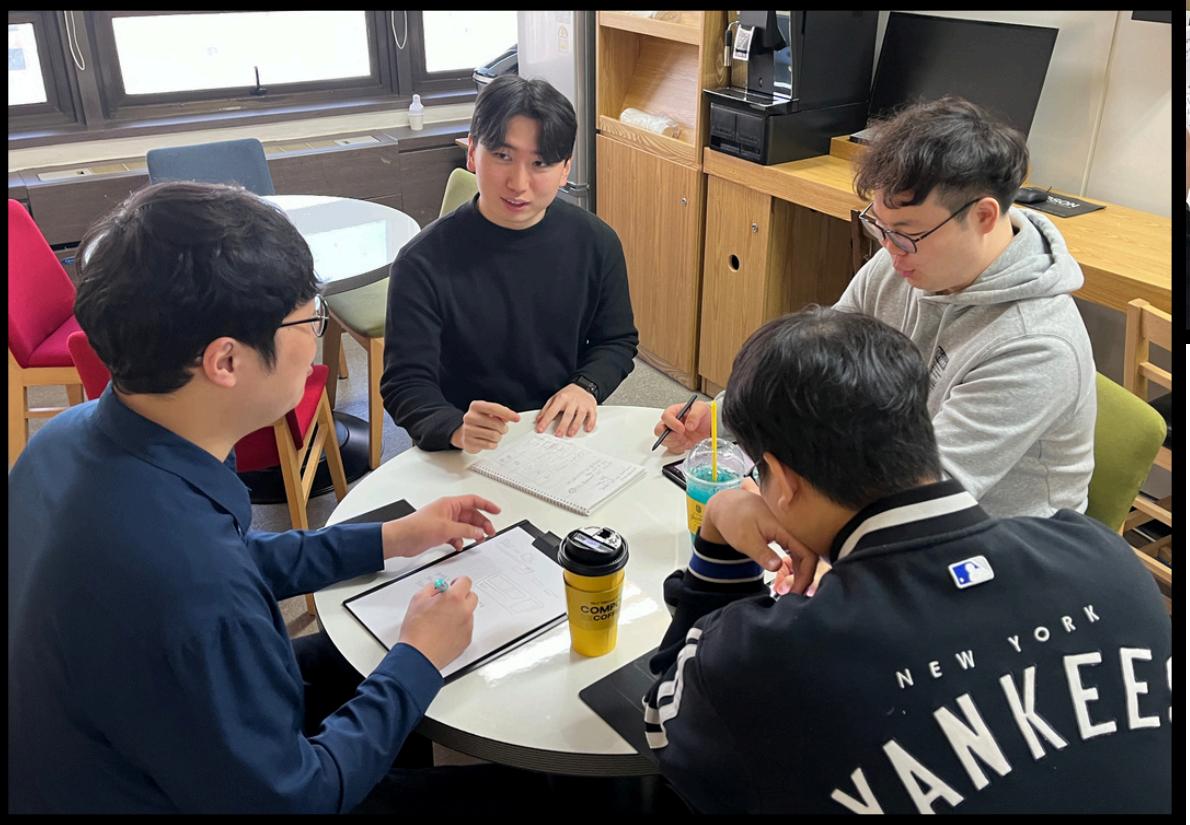
백

 GitHub 임베드

05

과정

2. 회의



2024.12.17 아침 회의 내용

TO DO LIST

소화기 감지 시스템 체크리스트 구현

담당자: @양영준

- 사람이 소화기를 직접 체크하고 확인하는 방식 도입
- 체크박스 형태의 인터페이스 구현
- 소화기 필수 배치 여부 확인 시스템 구축

목표: 소화기가 있어야 하는 작업에 소화기의 여부를 판단하기 위함

→ 이후 소화기 디텍팅 전제 조건을 사람에서 작업중으로 변경 (작업중을 위한 자세추정이 어려움)

헬멧 감지 및 크롤 기능 구현

담당자: @장해준

- 헬멧 감지 및 크롤 완료
- 헬멧 띠 부분의 색상 감지 기능 구현 예정

목표: 작업자 쓰러짐 감지 시 해당 작업자의 정보를 출력하기 위함

소화기 모델 / 지게차 모델 학습

담당자: @PKG

- 기존 학습 모델의 경량화 작업 진행
- 더 가벼운 모델로 재학습 예정

목표: 실제 기능 구현에 적합한 경량화된 모델 개발

메인프로그램 작성

담당자: @creator928forgit@gmail.com

- 기능들의 통합 작업 진행
- 최종 프로그램 UI 디자인 및 구현

목표: 각 기능들을 하나의 프로그램으로 통합하여 실제 사용 가능한 형태로 구현

05

과정

3. 업무 분배

박경규(팀장)

- TEST PC 환경 설정
- 모델 학습(OTX, YOLO)
 - 불, 연기 (YOLO/OTX)
- 불, 연기 object Detecting 모델 기반의 fire detector 프로그램 제작 및 모듈화
- python-PLC연동 및 SCADA 제작

김범규(수석 개발자)

- GitHub Repository 및 버전 관리
- 메인 프로그램 아키텍처 설계 및 구현
- 주요 기능 개발 및 최적화
 - PyQt 기반 GUI 구현
 - 소스코드 모듈화 및 기능 구현
 - 주요 클래스 및 메소드 개발 및 기능 구현
- 시스템 인테그레이션 및 최적화

양영준(서브 개발자)

- 모델학습(OTX)
 - 불,연기(OTX)
- 테스트용 데이터 수집
- 자세 추정모델 기반의 help detecto 프로그램 제작 및 모듈화
- safety gear 모델 기반의 fire_ext detector프로그램 제작 및 모듈화
- PPT 제작

장해준(AI 엔지니어)

- 모델학습(YOLO)
 - 헬멧(YOLO)
 - 소화기(YOLO)
- 테스트용 데이터 수집
- YOLO11모델 기반의 fall detector 프로그램 제작 및 모듈화



05 과정

4. 프로젝트 진행 일정 및 계획 수립

★ 사용 중 타임라인 보드 모두 +

▼ ◎ 진행 중 9

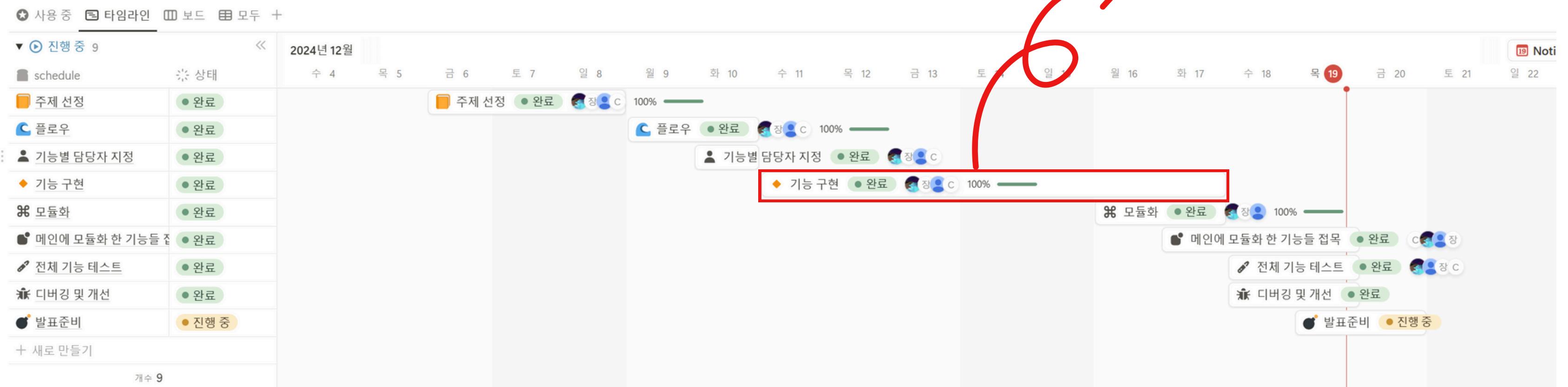
schedule	상태
주제 선정	완료
플로우	완료
기능별 담당자 지정	완료
기능 구현	완료
모듈화	완료
메인에 모듈화 한 기능들 접속	완료
전체 기능 테스트	완료
디버깅 및 개선	완료
발표준비	진행 중

개수 9

2024년 12월

수 4 목 5 금 6 토 7 일 8 월 9 화 10 수 11 목 12 금 13 토 14 일 15 월 16 화 17 수 18 목 19 금 20 토 21 일 22

Noti 19



기능 구현

상태: 완료

소유자: PKG, 장해준, 양영준, creator928forgit@gmail.com

진행률: 100%

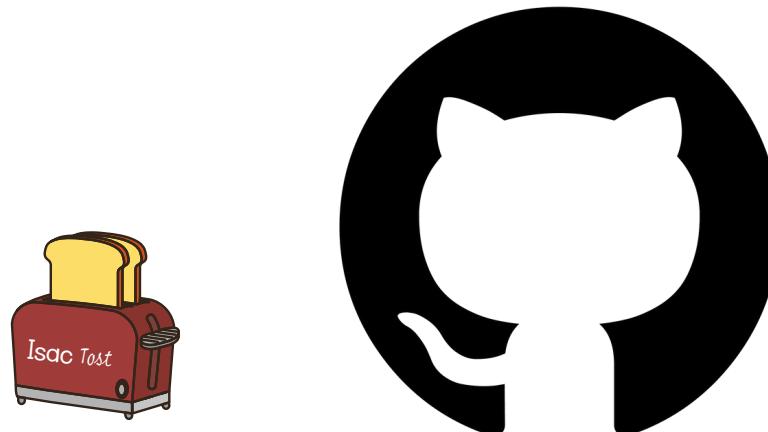
날짜: 2024년 12월 11일 → 2024년 12월 17일

속성 4개 추가

작업

- 메인 프로그램 및 UI 완료 creator928forgit@gmail.com 2024년 12월 9일 → 2024년 12월 16일
- 작업자 낙상 감지 완료 장해준 2024년 12월 10일 → 2024년 12월 12일
- 헬멧 완료 장해준 2024년 12월 12일 → 2024년 12월 13일
- 모선 감지 완료 양영준 2024년 12월 13일 → 2024년 12월 15일
- 안전장치 완료 양영준 2024년 12월 16일 → 2024년 12월 17일
- 불,연기 감지 완료 PKG 2024년 12월 11일 → 2024년 12월 16일
- 작업중 안전 장치 여부 완료 양영준 2024년 12월 17일
- 헬멧(띠부분) 색상 확인 완료 장해준 2024년 12월 18일

+ 새로 만들기



Milestone



Intel DX Team Project

05

과정

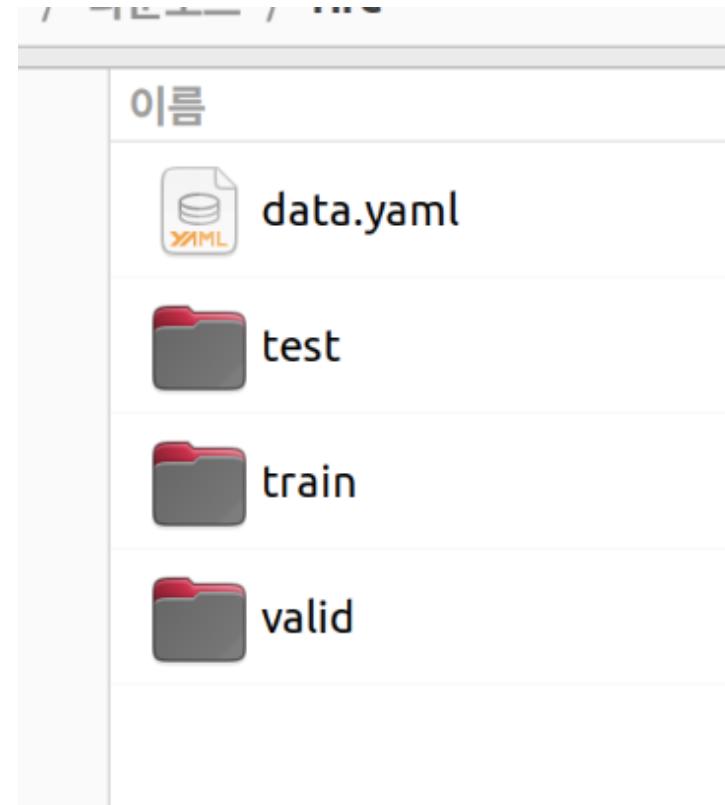
5. 환경구축



05

과정

6. 모델 학습을 통한 모델 생성



```
team998@intel-ESC4000-E10: ~/workspace/pkg
team998@intel-ESC4000-E10: ~/workspace/pkg 140x49

      all    1006    1905    0.932    0.852    0.917    0.696

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
8/400   9.10    0.6962   0.4536   1.067     52       640: 100% [██████████] 164/164 [00:59<00:00,  2.78it/s]
          Class    Images  Instances   Box(P)     R       mAP50  mAP50-95: 100% [██████████] 8/8 [00:04<00:00,  1.87it/s]
          all    1006    1905    0.934    0.837    0.914    0.688

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
9/400   9.16G   0.7015   0.4581   1.071     57       640: 100% [██████████] 164/164 [00:58<00:00,  2.81it/s]
          Class    Images  Instances   Box(P)     R       mAP50  mAP50-95: 100% [██████████] 8/8 [00:04<00:00,  1.82it/s]
          all    1006    1905    0.924    0.85     0.914    0.697

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
10/400  9.25G   0.6991   0.4556   1.071     70       640: 100% [██████████] 164/164 [00:59<00:00,  2.76it/s]
          Class   Images  Instances   Box(P)     R       mAP50  mAP50-95: 100% [██████████] 8/8 [00:04<00:00,  1.72it/s]
          all    1006    1905    0.922    0.866    0.928    0.72

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
11/400  9.58G   0.697    0.4526   1.066     113      640: 100% [██████████] 164/164 [00:59<00:00,  2.76it/s]
          Class   Images  Instances   Box(P)     R       mAP50  mAP50-95: 100% [██████████] 8/8 [00:04<00:00,  1.75it/s]
          all    1006    1905    0.931    0.847    0.927    0.72

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
12/400  9.22G   0.6991   0.4534   1.071     71       640: 100% [██████████] 164/164 [00:58<00:00,  2.82it/s]
          Class   Images  Instances   Box(P)     R       mAP50  mAP50-95: 100% [██████████] 8/8 [00:04<00:00,  1.70it/s]
          all    1006    1905    0.928    0.849    0.916    0.7

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
13/400  9.58G   0.7065   0.4569   1.071     63       640: 100% [██████████] 164/164 [00:57<00:00,  2.83it/s]
          Class   Images  Instances   Box(P)     R       mAP50  mAP50-95: 100% [██████████] 8/8 [00:04<00:00,  1.75it/s]
          all    1006    1905    0.925    0.858    0.919    0.707

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
14/400  9.25G   0.6985   0.452    1.073     72       640: 100% [██████████] 164/164 [00:59<00:00,  2.74it/s]
          Class   Images  Instances   Box(P)     R       mAP50  mAP50-95: 100% [██████████] 8/8 [00:04<00:00,  1.75it/s]
          all    1006    1905    0.918    0.859    0.92     0.708

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
15/400  9.57G   0.6958   0.4548   1.068     65       640: 100% [██████████] 164/164 [01:00<00:00,  2.72it/s]
          Class   Images  Instances   Box(P)     R       mAP50  mAP50-95: 100% [██████████] 8/8 [00:04<00:00,  1.74it/s]
```



fire_ext.pt



fire.pt

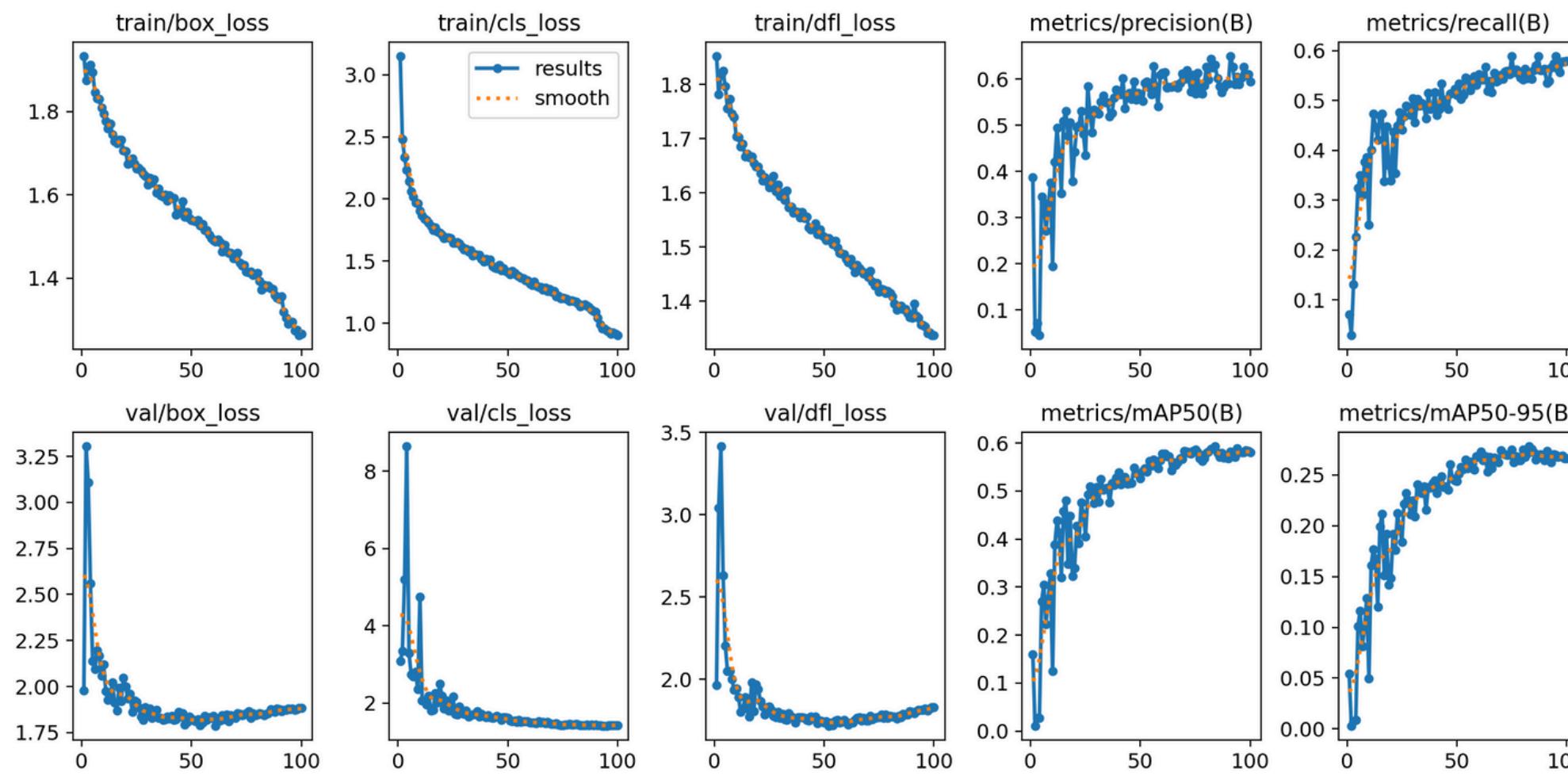


best_
helmet.pt



05 과정

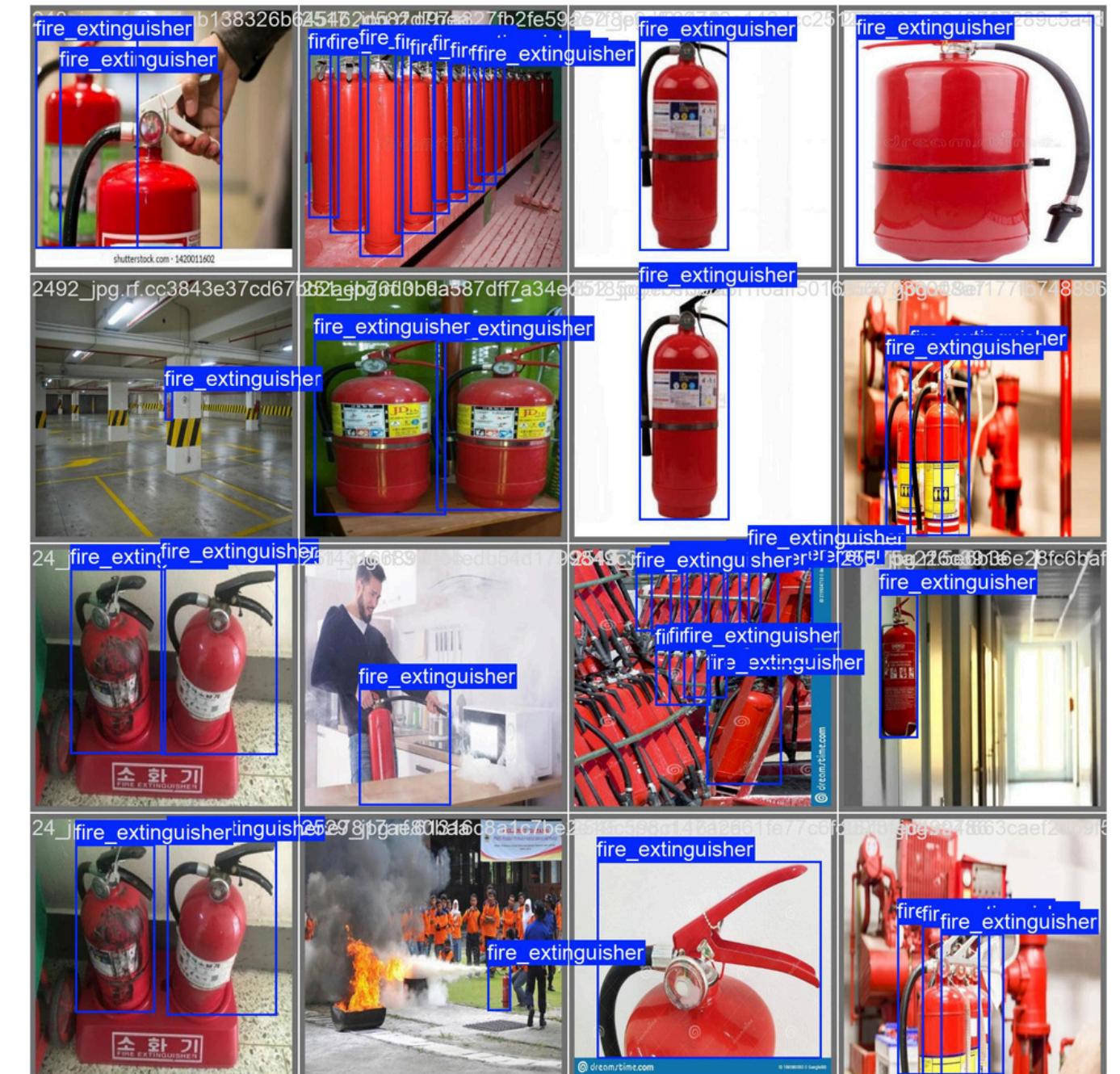
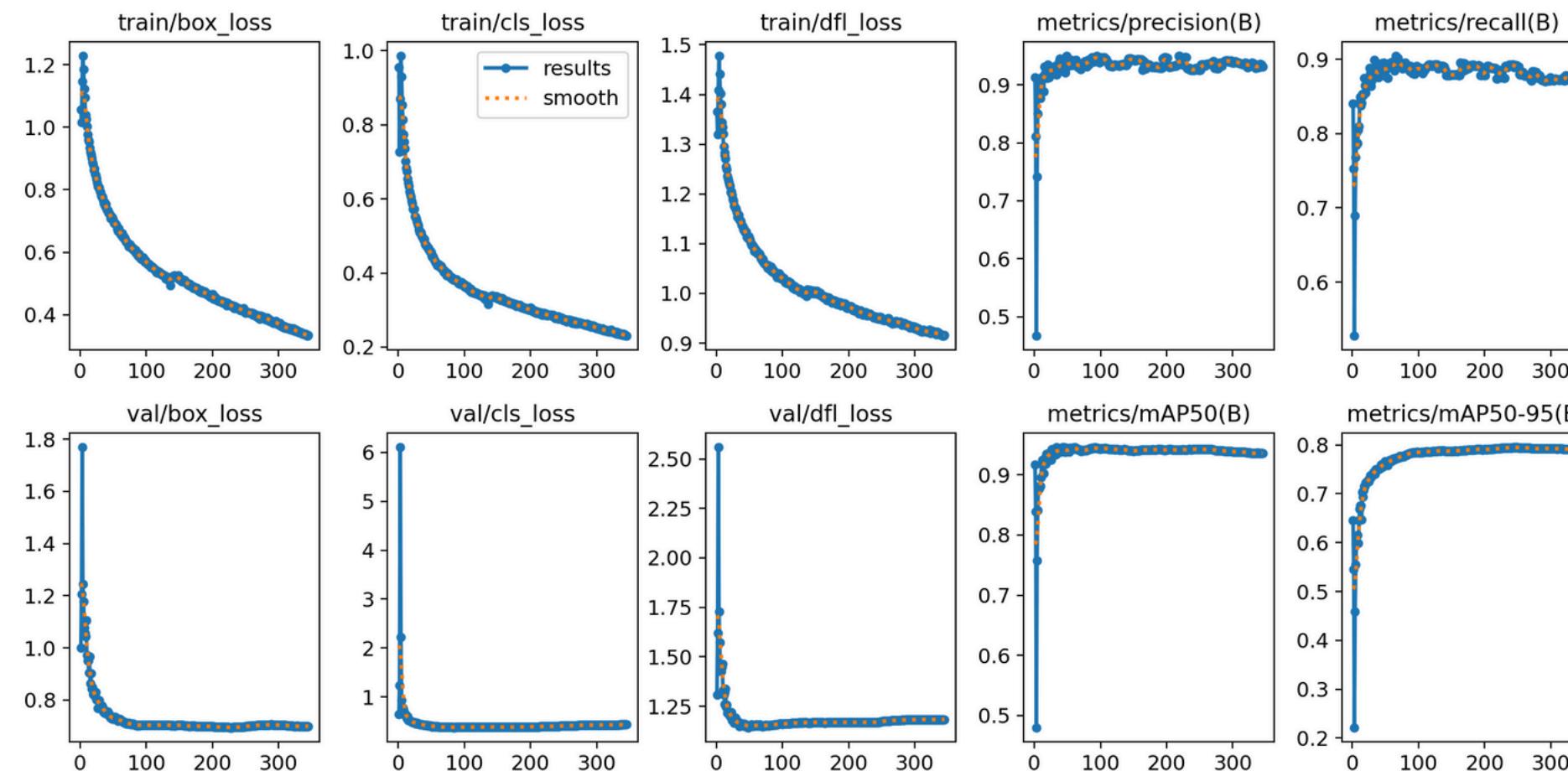
6. 모델 학습을 통한 모델 생성



05

과정

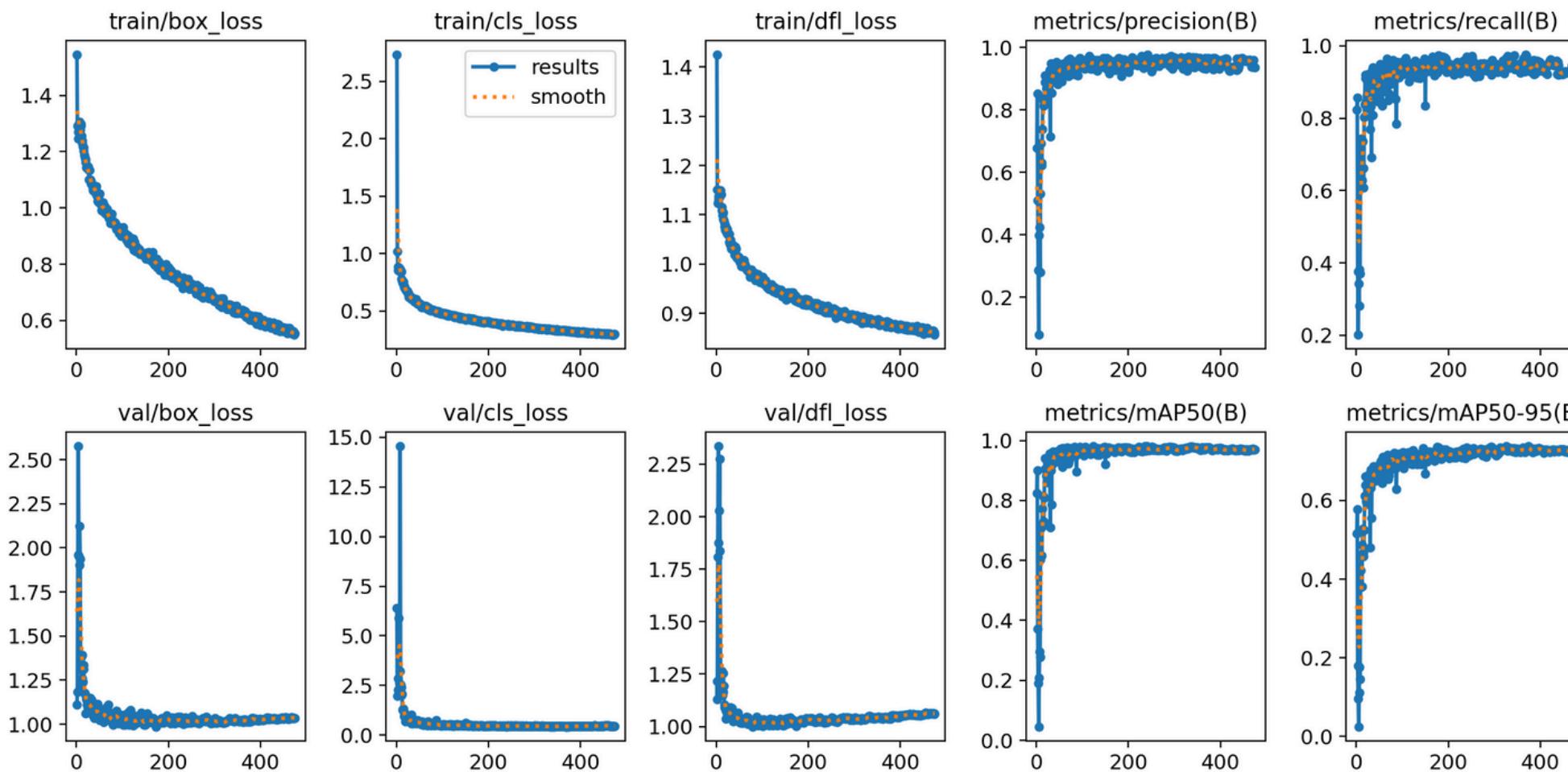
6. 모델 학습을 통한 모델 생성



05

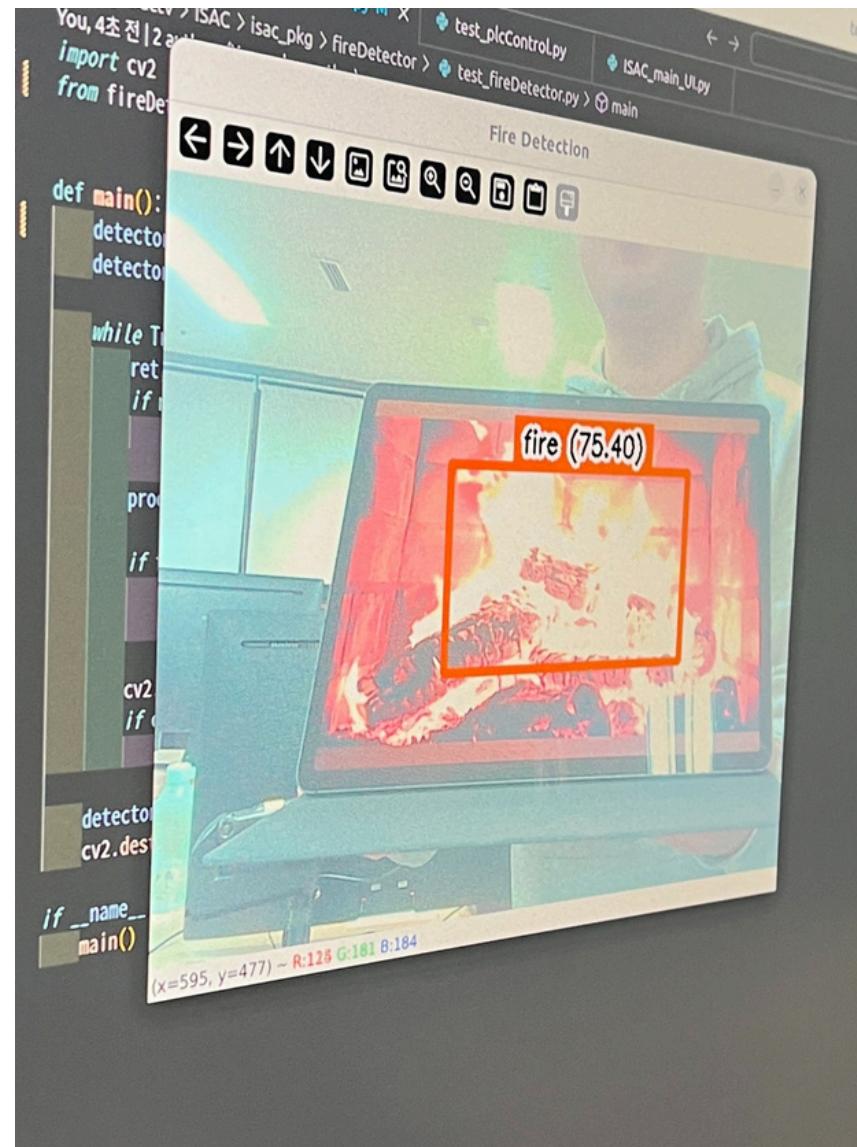
과정

6. 모델 학습을 통한 모델 생성



05 과정

7. 모델 테스트 및 기능 구현



05

과정

8. 기존 코드 모듈화



주간 아침 회의



TOST (Team Of Safe T)



세부 회의



2024년 12월 16일



양영준 PKG 장해준 creator928forgit@gmail.com



비어 있음



속성 4개 추가



댓글 추가



Isaac Tost

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

8. 기존 코드 모듈화

```
# 진짜변수
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)

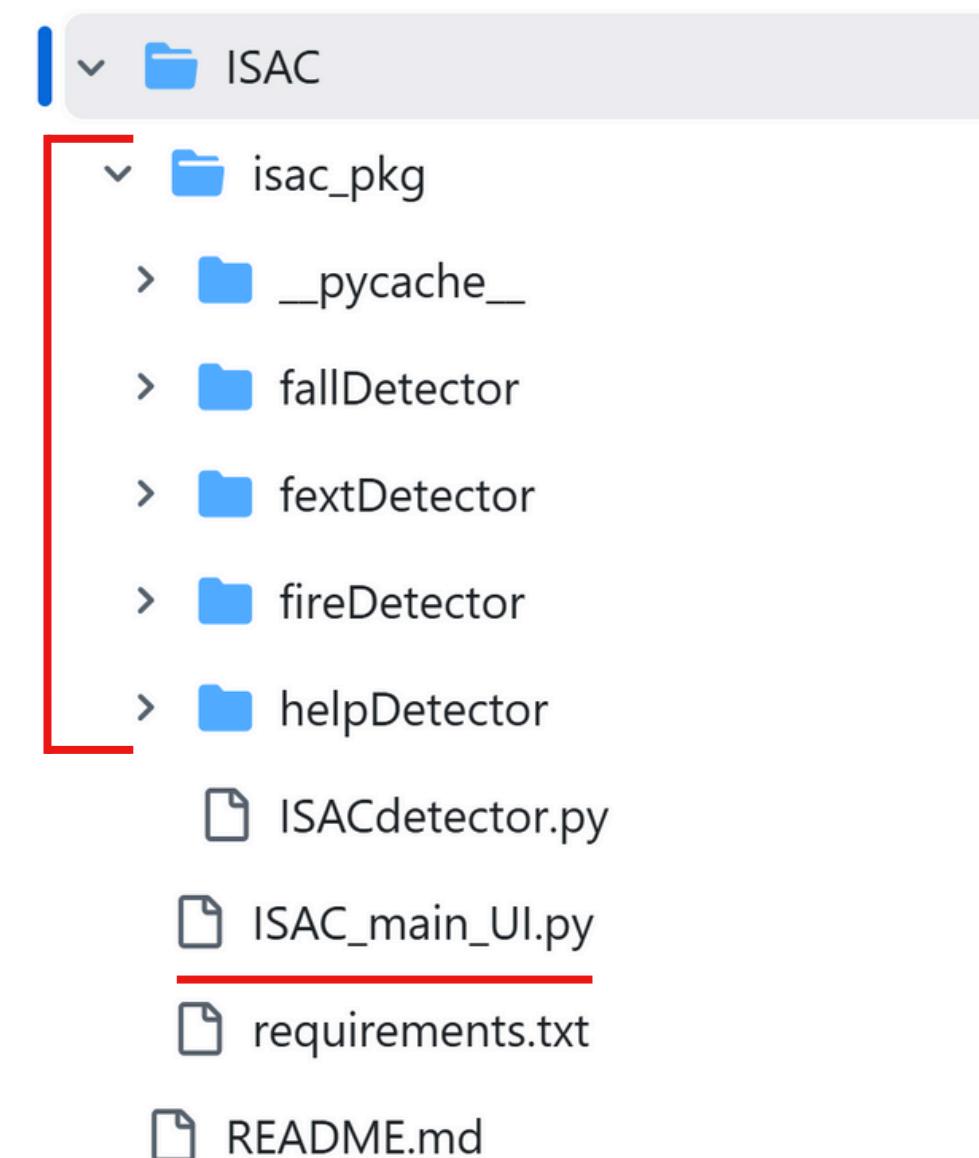
    # 추론 결과에서 포즈 데이터 그리기
    annotated_frame = results[0].plot() # YOLOv8의 plot 기능으로 그리기

    # 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

9. GUI 구상도



05

과정

9. GUI 시각화



05

과정

10. 기능 접목

ISAC - Industrial Safety A.I CCTV

영상 A 카메라 A

영상 B

이벤트 로그

Date: 2024-12-20 12:17:05

Isac Toast

낙상감지 A	NORMAL
구조감지 A	NORMAL
화재감지 A	NORMAL
안전장비감지 A	NORMAL

낙상감지 B	NORMAL
구조감지 B	NORMAL
화재감지 B	NORMAL
안전장비감지 B	NORMAL

스프링클러 작동

소방장비 점검

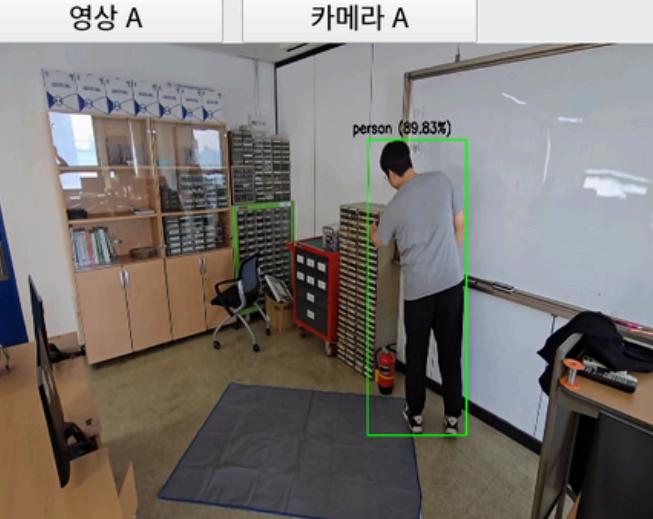


06

결과

ISAC - Industrial Safety A.I CCTV

영상 A 카메라 A



영상 B



이벤트 로그

- [2024-12-20 12:09:26] NO SAFETY 상황 발생 (B 카메라)
- [2024-12-20 12:09:38] NO SAFETY 상황 발생 (A 카메라)
- [2024-12-20 12:10:26] HELP! 상황 발생 (B 카메라)

날씨 감지 A: NORMAL
구조감지 A: NORMAL
화재감지 A: NORMAL
안전장비감지 A: NO SAFETY

날씨 감지 B: NORMAL
구조감지 B: HELP!
화재감지 B: NORMAL
안전장비감지 B: NO SAFETY

스프링클러 작동
소방장비 점검

Date: 2024-12-20 12:10:33

Isac Test

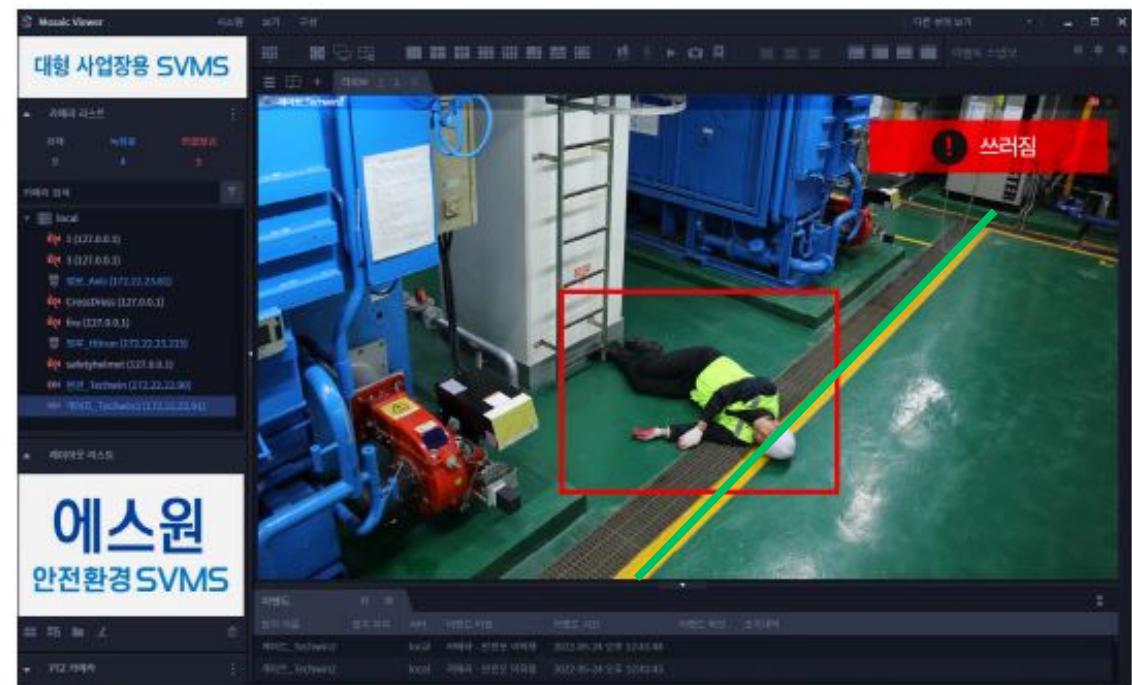


07 시연

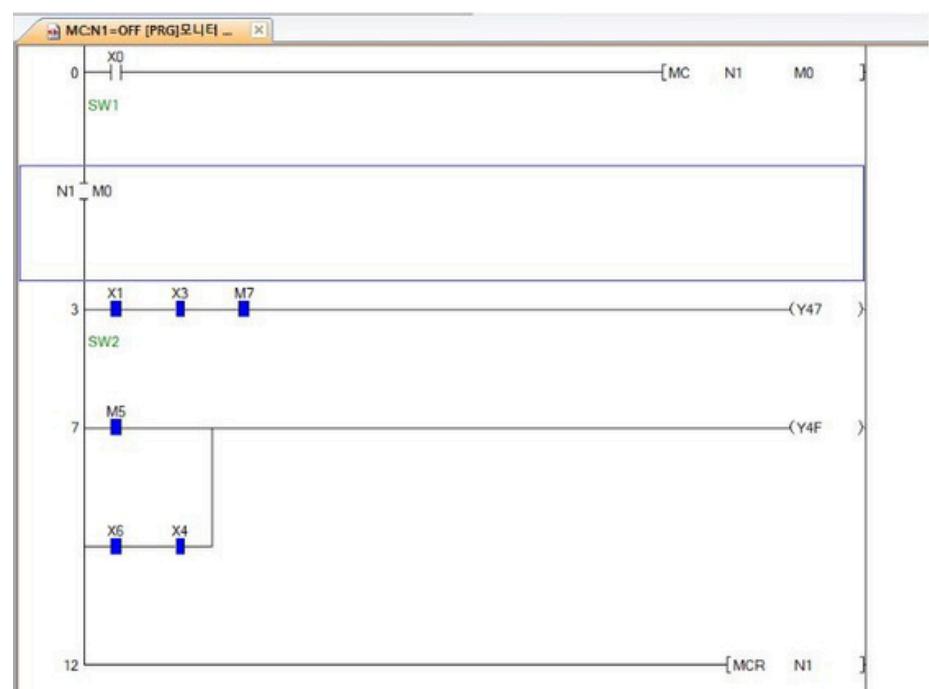
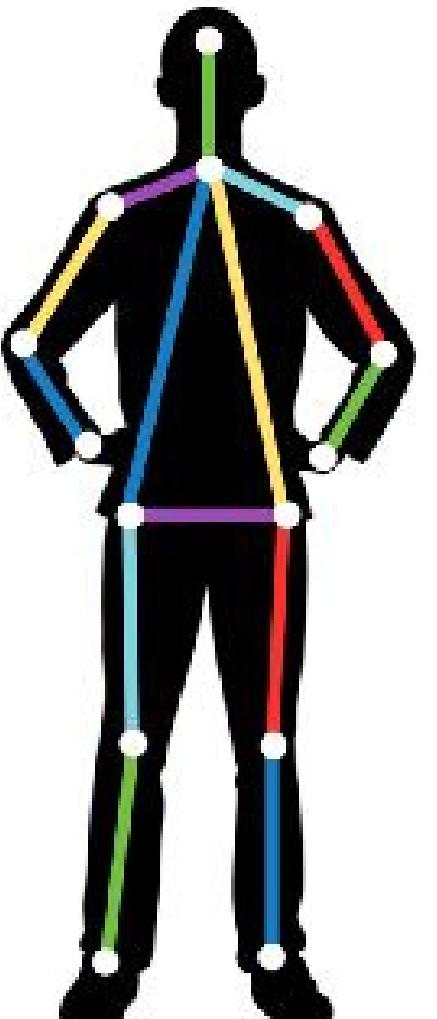
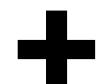


08

고찰

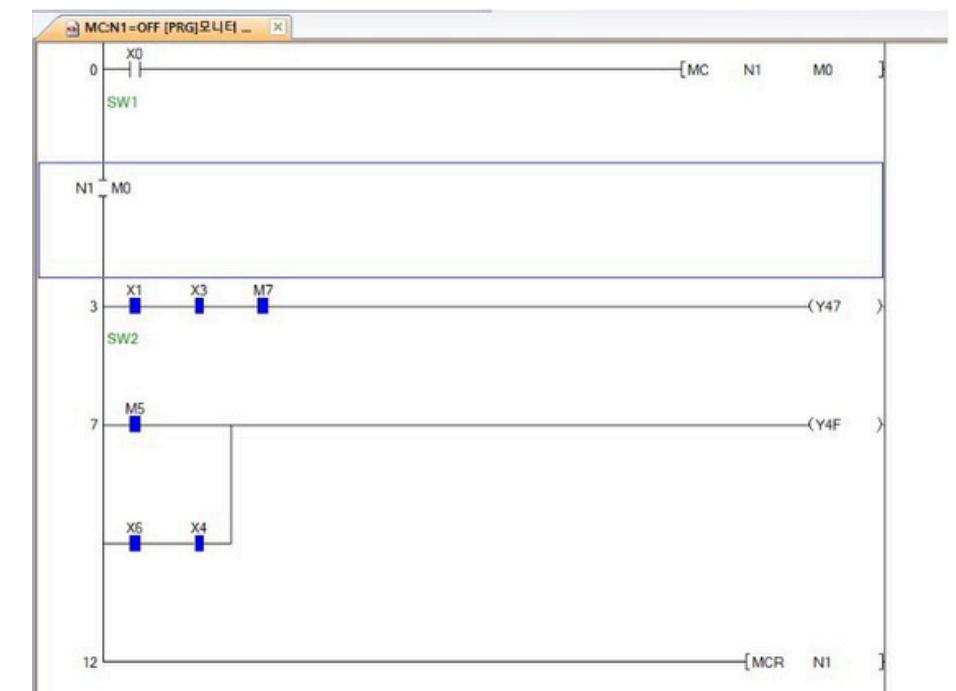


에스원 '안전환경 SVMS'에서 작업자의 '넘어짐'을 감지, 자동 알람이 발생되고 있다. 에스원 제공



08

고찰



08

고찰



+



감시 범위 ↑
사각지대 ↓

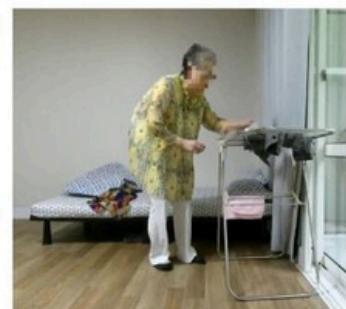


08

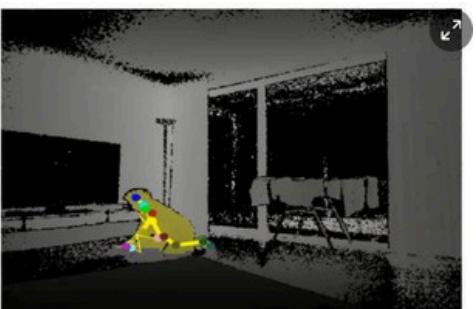
고찰



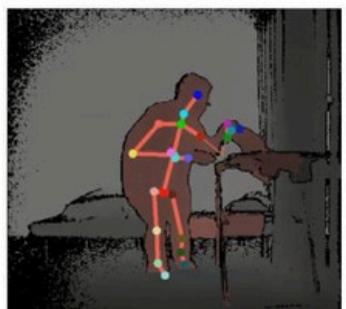
걸레로 바닥 닦기(RGB)



빨래 널기(RGB)



걸레로 바닥 닦기(DS)



빨래 널기(DS)





THANK YOU

감사합니다



Isaac Toast