

OPENING HAZARD SYSTEMATIC DETECTION

이승현 · 나상현 · 이정현 | CESI 팀

개구부 위험요소 자동감지 시스템

최근 10년간 건설업 사고사망자 현황에서 건설업의 사고가 전반적으로 감소하였지만, 여전히 이동식 비계의 사고율은 유지되거나 증가하는 경향이 보인다. 특히 떨어짐에 관한 사고가 많았으며, 통계적으로 이동식 비계의 떨어진 사고 중 안전난간의 비율이 높았다. 따라서 이를 개선하고자 높이 조절식 안전난간을 제작하여 편리하게 난간을 설치, 해제하고 어떤 현장에서도 사용할 수 있는 대안을 제시하였다.

시공

CONSTRUCTION &
MANAGEMENT



이승현 Lee Seung Hyeon
학번 : 2001676
e-mail : dbwkdud@naver.com



나상현 Na Sang Hyeon
학번 : 2001649
e-mail : sktkdus2001@naver.com



이정현 Lee Jeong Hyeon
학번 : 2001681
e-mail : rjswlts678@gmail.com



Opening hazard automatic detection system 개구부 위험요소 자동감지 시스템

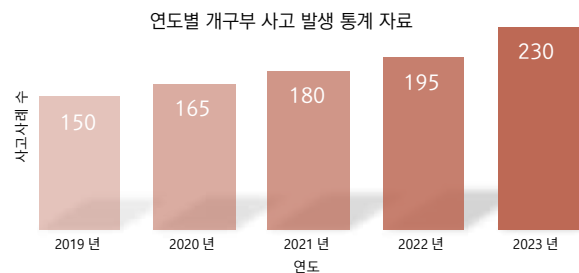
■ 건설현장 12대 기인물에 의한 사망사고 통계

2019년~2022년 사망사고 12대 기인물(단위: 명)

전체	12대 기인물	단부/개구부	철골	지붕	비계/작업발판	굴착기
566	344	51	48	40	39	28
100%	60.8%	9.8%	8.5%	7.1%	6.9%	4.9%
고소 작업대	사다리	달비계	트럭	이동식 비계	거푸집/동바리	이동식 크레인
28	22	21	19	18	17	13
4.9%	3.9%	3.7%	3.4%	3.2%	3.0%	2.3%

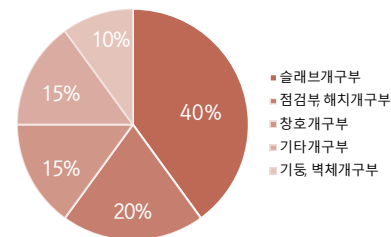
표와 같이 3년간 12대 기인물 사망사고 중 단부/개구부 사고가 가장 많은 것을 알 수 있다.

■ 연도별 개구부 사고 발생 통계 자료



〈그래프 1〉 연도별 개구부 사고 발생 통계자료 그래프 1을 통해 개구부 사고 발생이 매년 증가하는 개구부에서현상을 알 수 있다.

개구부 유형에 따른 사고 비율



〈그래프 2〉 개구부 유형에 따른 사고비율 〈그래프 2〉을 통해 개구부의 유형 중 슬래브 발생 하는 사고가 가장 많은 것을 알 수 있다.

사고사례



사고형태 | 추락(떨어짐)
개구부 유형 | 슬래브 개구부
사고개요 | 공조덕트 작업 중 미끄러져 추락
사고원인 | 덮개 미설치/추락방지망 설치 미흡
재발방지대책 | 덮개 설치, 추락방지망 설치



사고형태 | 추락(떨어짐)
개구부 유형 | 슬래브 개구부
사고개요 | 개구부 덮개를 대충 설치하여 재해자가 지나가다 밟고 추락
사고원인 | 전용 덮개 미사용/ 추락방지망 미설치
재발방지대책 | 전용 덮개 사용/ 추락방지망 설치

■ 슬래브 개구부의 사고 발생 원인 및 문제



- 대부분 바닥에 뚫린 형태로 존재하여 눈에 잘 띄지 않는다.
- 병행작업 상황에서는 작업자의 주의력이 분산된다.
- 주변에 자재 등으로 인해 개구부가 시각적으로 가려져 개구부의 위치 인식이 미흡하다.
- 덮개 및 안전시설의 설치가 미흡하다.
- 작업 공간의 제한과 이동 동선의 문제가 있다

■ 개구부 병행작업의 종류와 위험 원인

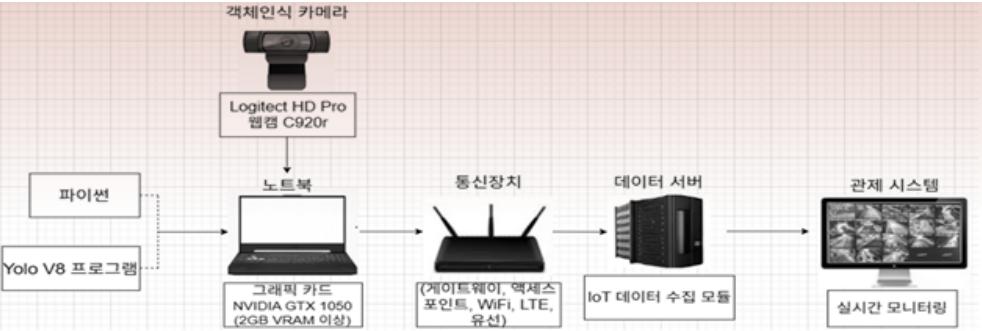
- 개구부 주변 정리정돈 작업
- 슬래브 기동 철근마감 작업
- 개구부 인근에서 슬래브 알폼 조립 작업
- 개구부를 통해 자재 인양(알폼) 중 알폼 조립 병행 작업



설계 목표

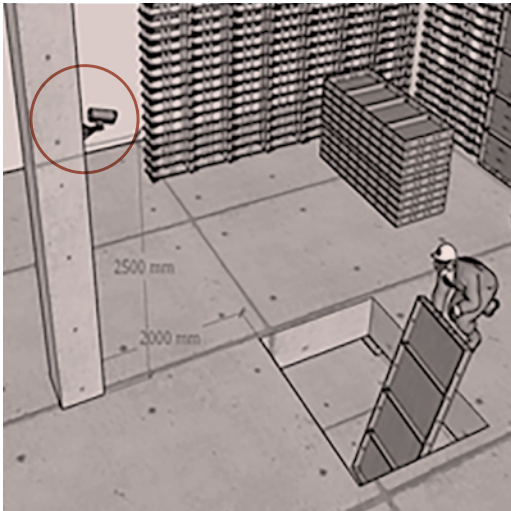
- 개구부의 덮개가 전용 덮개로 안전하게 덮여 있는지 인식
- 개구부 위험 환경내에 작업자가 있는지 인식
- 개구부를 통한 자재 인양 외에는 위험요소로 구별하여 인식
- 작업 종료 후 개구부 덮개를 개방한 후 일정 시간이 지나면 위험요소로 구별되어 관리자에게 연락이 가도록 설정

설계 구성요소



카메라 설치 위치 및 각도

- 설치위치 : 개구부 모서리로부터 대각선 방향으로 약 2m 거리의 2.5m~3m 높이의 벽이나 기둥에 설치
- 설치각도 : 개구부 중심을 기준으로 약 45도 아래로 설치



시스템의 원리

– 학습된 데이터를 통한 객체 인식(학습된 모델 불러오기)

```
from ultralytics import YOLO
import cv2, os
# 학습된 모델 불러오기
model = YOLO(r"C:\Users\USER\Desktop\W00000\runs\yolov8_train\weights\best.pt")
# 입력 영상 경로
video_path = r"C:\Users\USER\Desktop\W123.mp4"
cap = cv2.VideoCapture(video_path)
# 출력 영상 저장 경로
base_name = os.path.basename(video_path)
name_only, ext = os.path.splitext(base_name)
output_dir = r"C:\Users\USER\Desktop\영상 저장"
os.makedirs(output_dir, exist_ok=True)
output_path = os.path.join(output_dir, f"{name_only}_result{ext}")
fourcc = cv2.VideoWriter_fourcc(*"mp4v")
out = None
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    if out is None:
        fps = cap.get(cv2.CAP_PROP_FPS)
        w = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
        h = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
        out = cv2.VideoWriter(output_path, fourcc, fps, (w, h))
    results = model(frame, conf=0.25, verbose=False)
    annotated_frame = results[0].plot()
    # 탐지된 클래스명 수집
    detections = results[0].boxes
    detected_classes = []
    for box in detections:
        cls_id = int(box.cls.cpu().item0)
        cls_name = model.names[cls_id].lower()
        detected_classes.append(cls_name)
```

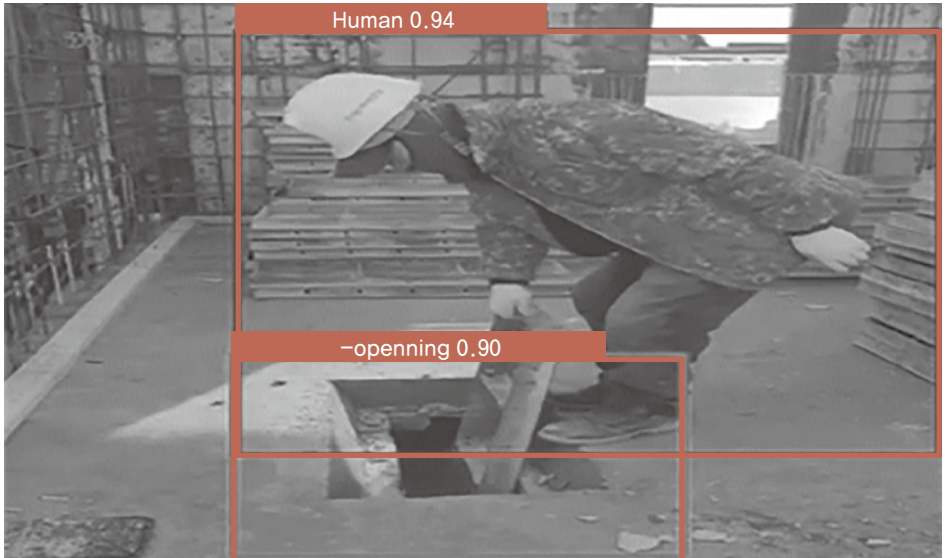
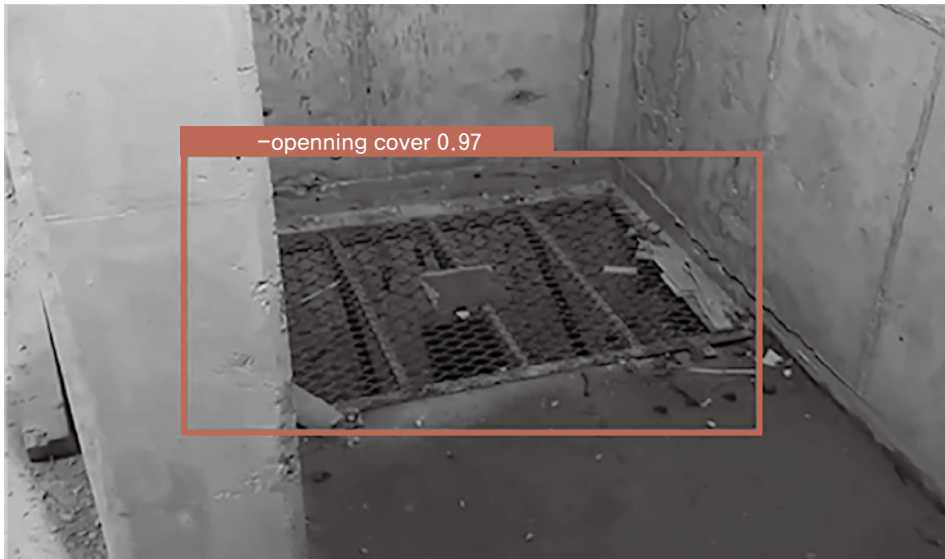
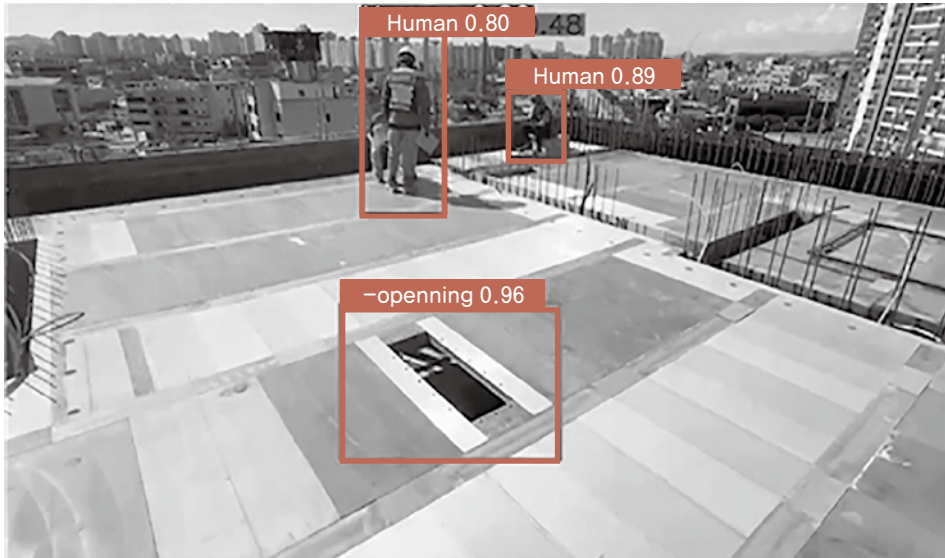
– 실시간 영상 분석을 통한 객체 인식 및 분류

```
# YOLO 예측
results = model(frame, conf=0.25, verbose=False)
annotated_frame = results[0].plot()
# 탐지된 클래스명 수집
detections = results[0].boxes
detected_classes = []
for box in detections:
    cls_id = int(box.cls.cpu().item0)
    cls_name = model.names[cls_id].lower()
    detected_classes.append(cls_name)
```

– 위험알림 조건에 따른 데이터 분석 및 도출

```
# 조건 체크 (-opening 기준)
if "-opening" in detected_classes and "al-form" not in detected_classes:
    status_text = "WARNING: Opening without Al-Form!"
    color = (0, 0, 255) # 빨강
else:
    status_text = "Safe"
    color = (0, 255, 0) # 초록
# 상태 텍스트 표시
cv2.putText(annotated_frame, status_text, (50, 50),
            cv2.FONT_HERSHEY_SIMPLEX, 2, color, 3)
# 출력 및 저장
cv2.imshow("Result", annotated_frame)
out.write(annotated_frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
cap.release()
if out:
    out.release()
cv2.destroyAllWindows()
print(f"결과 영상 이 저장되었습니다: {output_path}")
```

최종 모델



기대효과

실시간 객체 인식으로 인하여 위험 요소 감소

- 위험 요소 감지시 관리자에게 알림이 전송되어 빠른 조치 가능
- 개구부의 덮개 상태 및 개방 여부를 인식하여 안전하게 작업 가능
- 자재 인양 작업 시 작업자의 신체가 개구부를 통하는지 위험 감지 가능