Safezone

5 June 2024

CloseAI 이상헌 김유철 박준혁 이정훈

목차 table of contents

1

Overview

프로젝트 목표

특정 상황에서 안전 관리에 도움을 주는 모델 개발

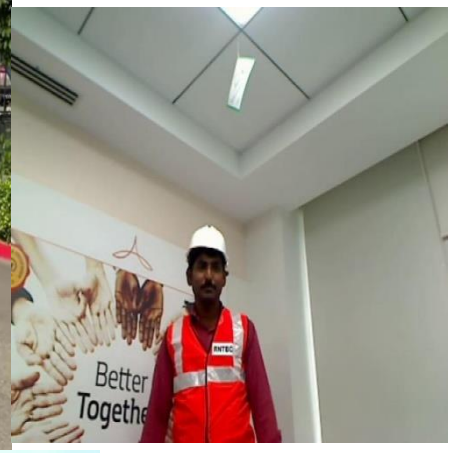
사전에 정의한 각종 위험한 상황을 탐지 및 경고

관리자가 모니터링 하고 싶은 객체를 입력 받아 탐지

2

Data

활용 데이터



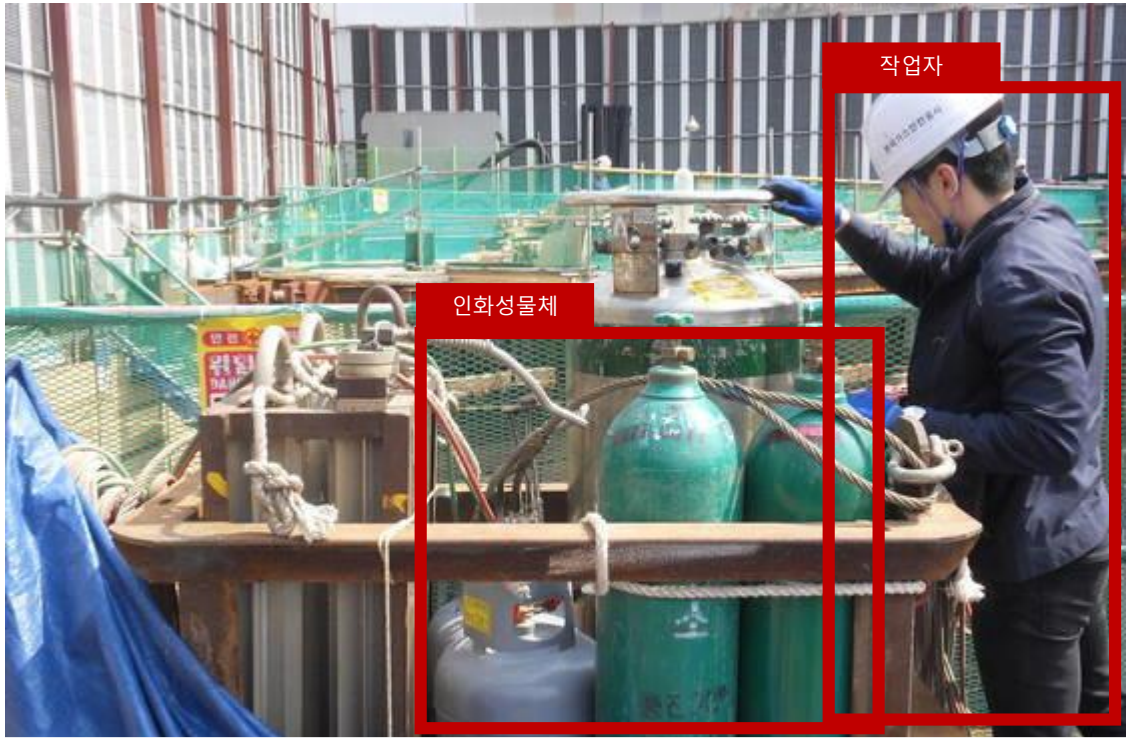
Roboflow 데이터셋 수집
AI허브 안전장비 데이터셋

출처:https://universe.roboflow.com/search?q=construction+images%3E5000+model%3Ayolov8

3

Model

Case 1



사전에 정의한 특정 이벤트 발생시 경고

문자 수신
<인화성 물질에 작업자 접근. 확인 요망>

Case 2

사용자가 원하는 특정 상황을 입력 받아 탐지 가능

예시) 입력 : 안전모를 착용하지 않은 사람을 찾아줘

YOLO v8

모델 선정 이유

1. 높은 정확도와 속도
2. 다양한 객체 탐지
3. 간편한 학습 및 튜닝
4. 고해상도 이미지 처리

다른 모델과 비교

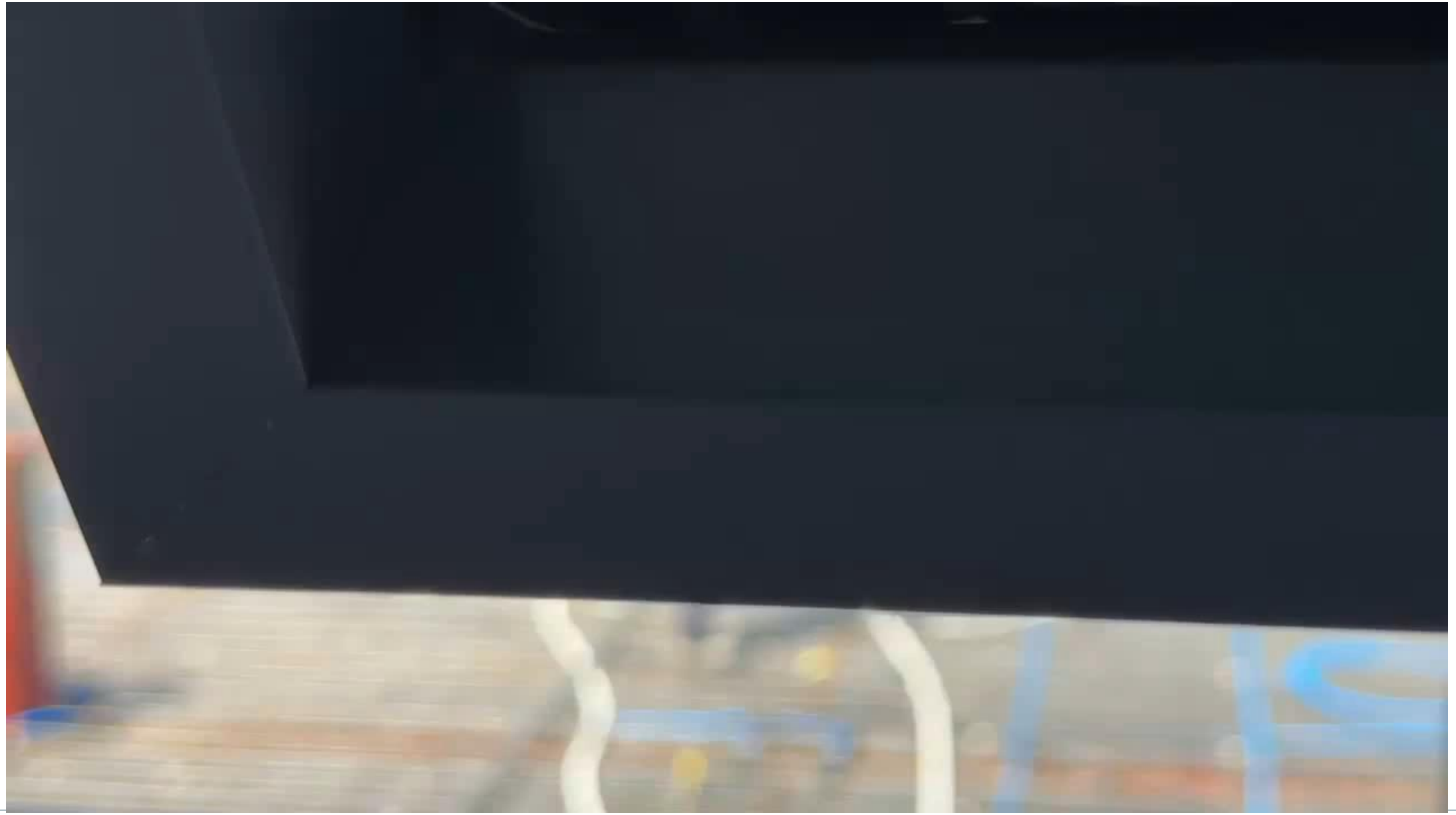
YOLO 버전 비교

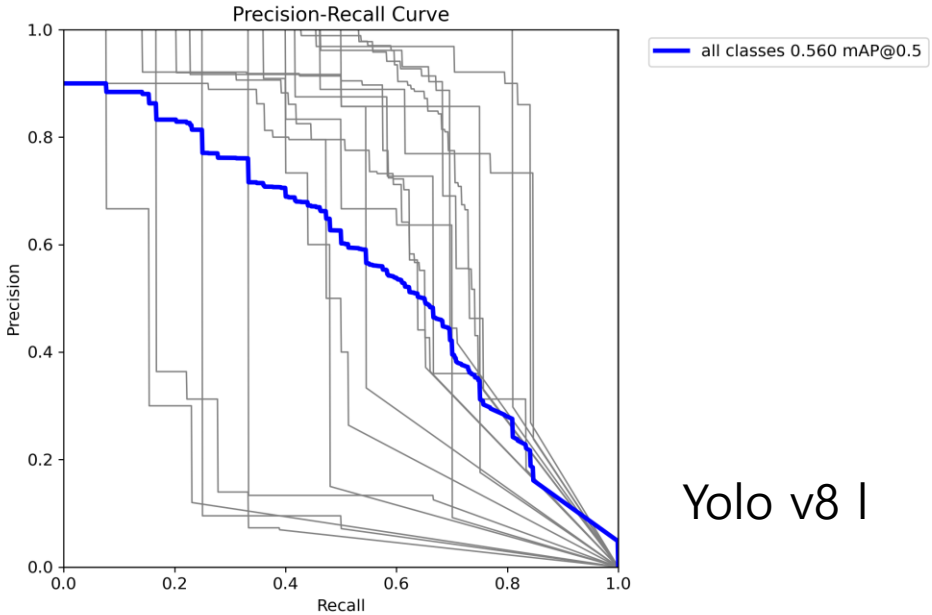
버전	구조와 효율성	성능
YOLO6	경량화와 속도 최적화에 중점	작은 객체 탐지에서 강점, 빠른 속도
YOLO7	효율적 구조와 훈련기법으로 성능 향상	다양한 벤치마크에서 높은 성능
YOLO8	최신 기술 통합과 모듈화로 성능 극대화	최신 기술 적용으로 최고성능

4

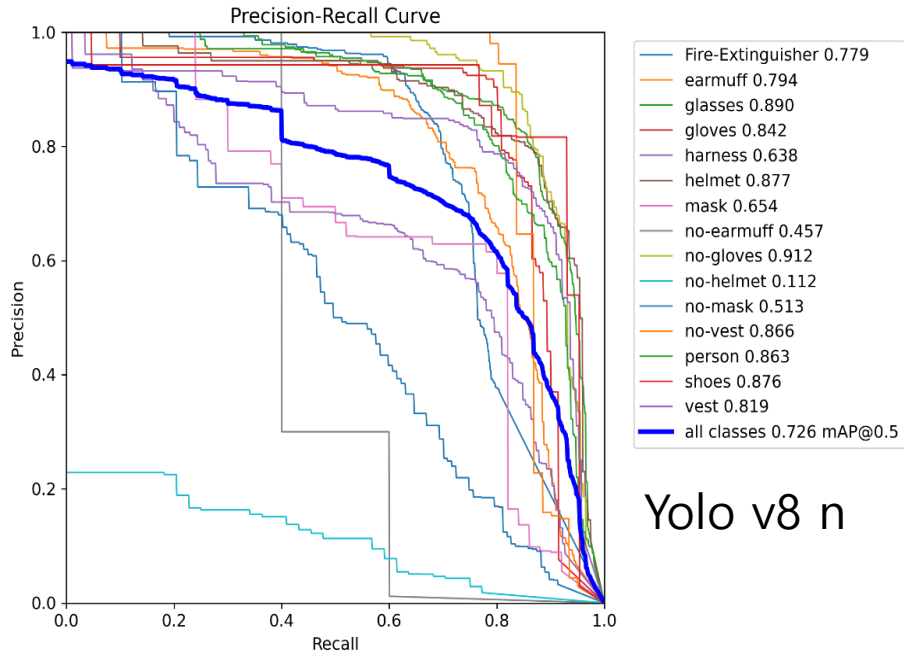
프토로타입

YOLO v8 구현 결과

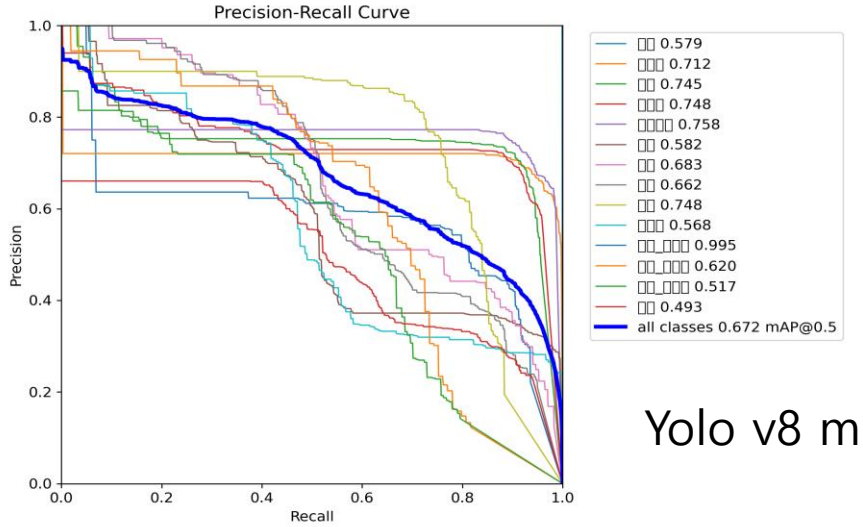




Yolo v8 l



Yolo v8 n



Yolo v8 m

5

핵심 코드

핵심 코드

```python
from ultralytics import YOLO



# Load a model
model = YOLO("yolov8l.yaml")  # build a new model from scratch
model = YOLO("yolov8l.pt")  # load a pretrained model (recommended for training)

# Use the model
model.train(data="data.yaml", epochs=500, batch=-1, patience=100)  # train the model
metrics = model.val()  # evaluate model performance on the validation set
# results = model("https://ultralytics.com/images/bus.jpg")  # predict on an image
# results = model("/home/saka/바탕화면/123.jpg")  # predict on an image

path = model.export(format="onnx")  # export the model to ONNX format
```

핵심 코드

```python
import cv2
from ultralytics import YOLO
import numpy as np
from PIL import ImageFont, ImageDraw, Image

# Load a model
model = YOLO("best.pt")  # load a pretrained model

# 비디오 파일 경로 설정
video_path = "/home/saka/문서/카카오톡 받은 파일/bbb.mp4"
output_path = "/home/saka/yolov8/output/annotated_video_123.mp4"

# 비디오 캡처 객체 생성
cap = cv2.VideoCapture(video_path)

# 비디오 저장 객체 설정
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
fps = cap.get(cv2.CAP_PROP_FPS)  # 원본 비디오의 프레임 속도 가져오기
slow_fps = fps / 2  # 비디오 속도를 절반으로 줄이기
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
out = cv2.VideoWriter(output_path, fourcc, slow_fps, (width, height))
```

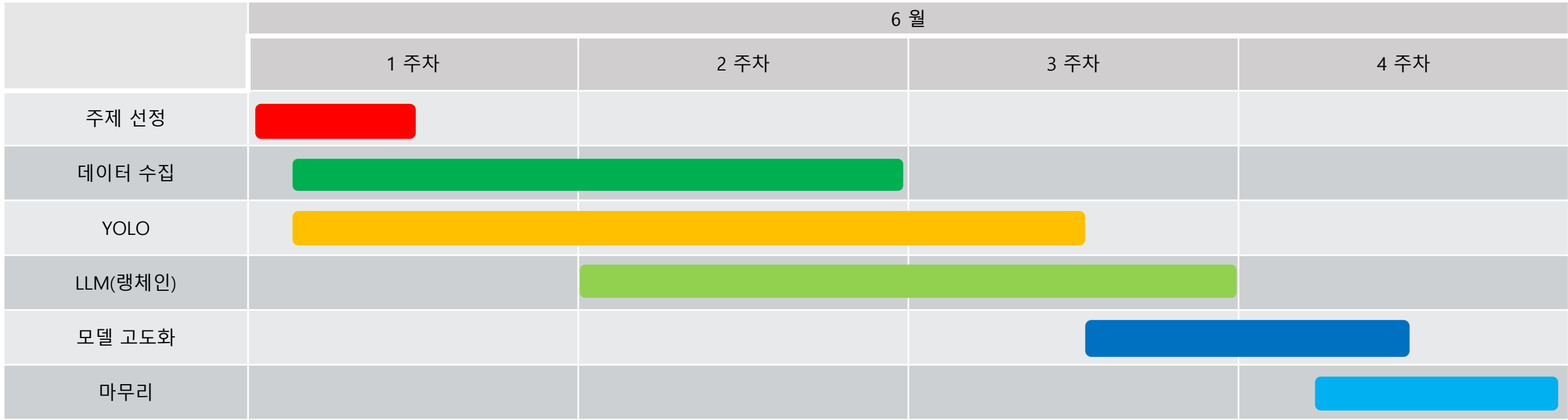
핵심 코드

```python
# 예측 결과에서 바운딩 박스 정보 추출
predictions = results[0].boxes  # Access the first result's boxes

# OpenCV 이미지를 Pillow 이미지로 변환
frame_pil = Image.fromarray(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
draw = ImageDraw.Draw(frame_pil)

# 프레임에 주석 추가
for pred in predictions:
    x1, y1, x2, y2 = map(int, pred.xyxy[0])  # Get bounding box coordinates
    conf = pred.conf[0]  # Confidence
    cls = int(pred.cls[0])  # Class
    print(f"Detected class ID: {cls}, confidence: {conf}")  # 클래스 ID 출력
    label = f"{model.names[cls]} {conf:.2f}"
    color = colors[cls].tolist()  # 클래스에 해당하는 색상 가져오기
    draw.rectangle(((x1, y1), (x2, y2)), outline=tuple(color), width=2)
    draw.text((x1, y1 - 50), label, font=font, fill=tuple(color) + (255,))

# Pillow 이미지를 다시 OpenCV 이미지로 변환
frame = cv2.cvtColor(np.array(frame_pil), cv2.COLOR_RGB2BGR)

# 주석이 추가된 프레임을 비디오 파일에 저장
out.write(frame)
```

6

Time Line

Time Line



7

사용 기술

사용 기술


TensorFlow


PyTorch


OpenCV


ultralytics
YOLOv8


scikit
learn


matplotlib


NumPy

8

Reference

Reference

Real-Time Flying Object Detection with YOLOv8 (Dillon Reis, 22 May 2024 https://arxiv.org/pdf/2305.09972**)**

You Only Look Once: Unified, Real-Time Object Detection (Joseph Redmon, 9 May 2016 https://arxiv.org/pdf/1506.02640**)**

9

QnA