o 2025.04 **o**

" 교차로 돌발 진입 감지

팀 명 : JoyK | 팀장 : 조영규 | 팀원: 김우혁, 유혜정, 정날빛

프로젝트 플래너

	· · · · · · · · · · · · · · · · · · ·	기간 계획 % 완료		실제 시작											
활동	계획 시작	기간 계획	실제 시작	실제 기간	완료율	4월 3	4	7 9	9	10 1	1 1	4 15	16	17	18
요구사항 정의	3	1	3	1	100%										
시스템 기능 및 감지 대상 정의	3	1	3	1	100%										
돌발진입 상황 시나리오 수집	4	1	4	1	100%										
교차로 영상 데이터 수집	7	5	7	4	100%				- 20						
라벨링 및 전처리 (돌발진입, 장애물 등)	7	5	7	3	100%										
YOLO 모델 선정 및 개발 환경 구성	7	5	7	1	100%										
모델 학습 및 하이퍼파라미터 튜닝	7	5	10	2	100%						٠,				
모델 성능 측정 (정확도, 오탐률 등)	14	5	14	2	100%										
실시간 감지 및 실제 환경 검증	14	5	14	2	100%										
감지 결과 기반 데시보드 연동	14	5	14	2	100%							,,,,			
경고 알림 기능 구현	14	5	15	1	100%										
전체 시스템 테스트 및 성능 개선	14	5	15	1	100%										
결과 정리 및 보고서 작성	14	5	16	3	100%										
발표 자료 및 시연 영상 정리	14	5	16	3	100%										

목차

O1 프로젝트 개요 및 개발 배경

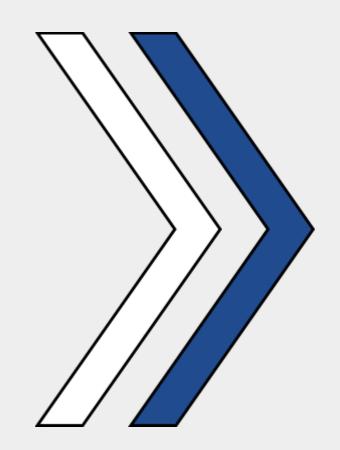
O2 시스템 구성 및 기술 스택

OS 데이터 처리 및 AI 모델 설계

O4 시스템 구현 및 작동 시연

05 결과 정리 및 향후 계획

프로젝트 개요 및 개발 배경



- 문제 정의 및 필요성
- 프로젝트 목표

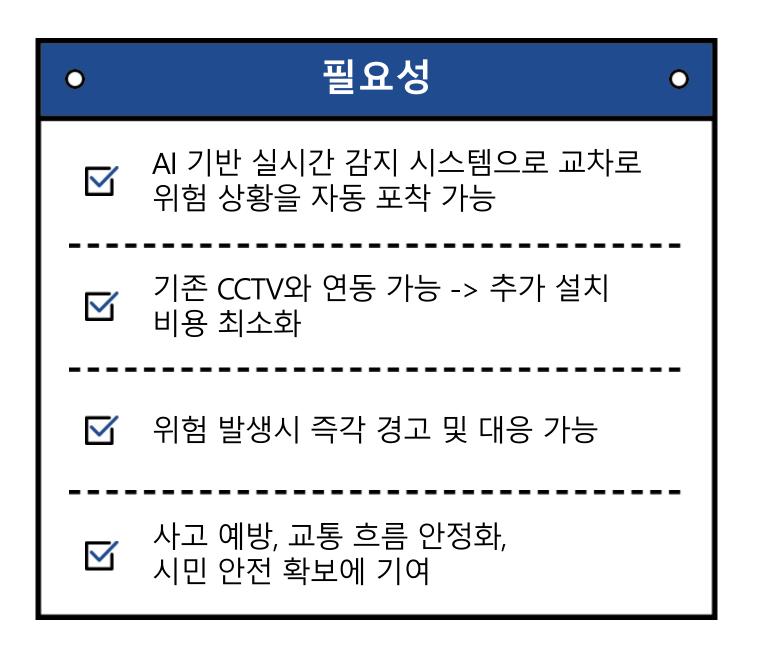
문제 정의 및 필요성

교차로에서 발생하는 교통 혼잡, 차량 대기, 차량 충돌 등 **""**

교차로는 다양한 교통 주체가 동시에 이동하는 고위험 구간이다.

특히, 돌발 차량 진입, 신호 위반, 역주행, 보행자무단횡단, 장애물 정지 차량 등 예측 불가능한이벤트가 빈번하게 발생한다.

기존 감시 시스템은 사후 확인 중심으로, 실시간 감지 및 대응이 어렵다.



프로젝트 목표

문제점

- 신호 위반, 중앙선 침범, 역주행 등
 돌발 차량 진입 사례 빈발
- 보행자나 정상 차량과의 충돌 위험
- 야간, 악천후, 교차로 진입로 사각지대 위험성 증가

프로젝트 개요

실시간 영상 분석을 통해 교차로에 진입하는 돌발 차량을 감지합니다. 이를 대시보드 형태로 시각화, 운전자가 빠르게 인식하고 반응할 수 있도록 도웁니다.

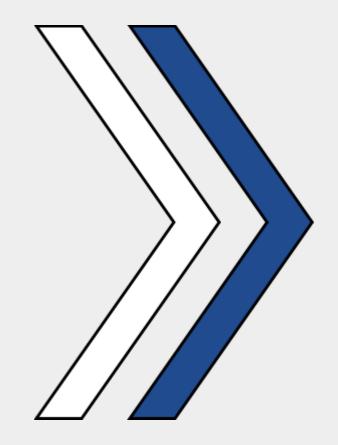


FINAL GOAL

프로젝트 최종 목표

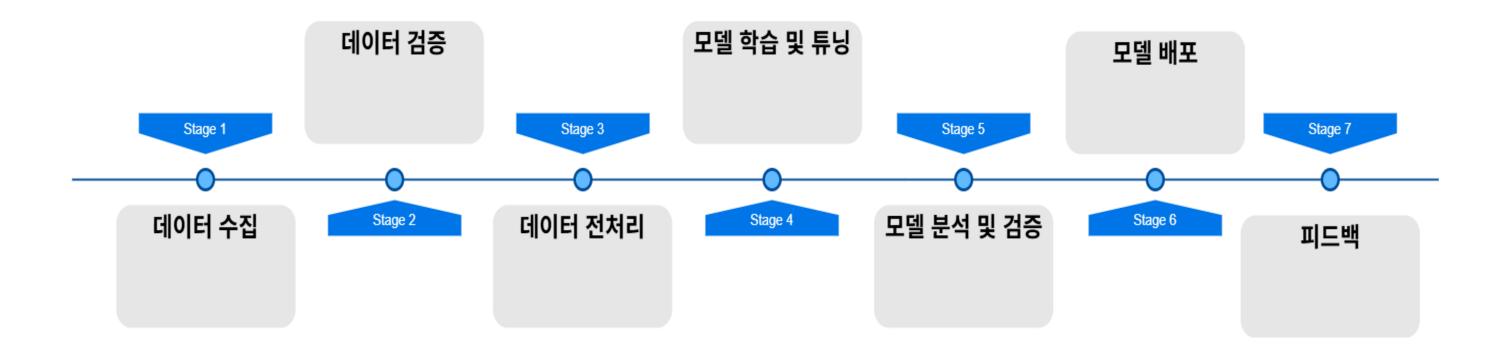
- 교차로에서 돌발 진입하는 차량을 실시간으로 감지.
- YOLO 모델을 활용하여 차량의 위치와 방향을 정확히 인식
- 돌발 진입 사건 발생 시 실시간 경고 및 알림 표시

시스템 구성 및 기술 스택



- 전체 시스템 흐름도
- 사용 기술 및 개발 환경
- 역할 분담

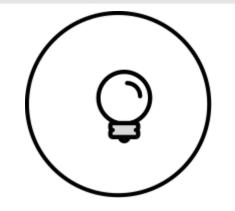
<u>02</u> 전체 시스템 흐름도



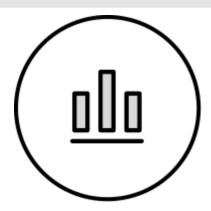
02 사용 기술 및 개발 환경

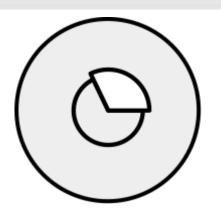
구분	기술/도구			
프로그래밍 언어	Python			
AI 모델	YOLO 11(Roboflow API)			
데이터 처리	Pandas, NumPy			
학습 프레임워크	OpenCV, YOLO			
웹 실행 환경	Streamlit			
개발 도구	Visual studio code			
버전 관리	Git, GitHub			

역할 분담









팀장, 총괄, 데이터,모델훈련(약3300 개) 검수 및 수정, YOLO모델 전처리 및 구현Roboflow APL연동

CCTV프레임 수집(약200만개), CCTV동영상 수집(약200만개), 모델훈련(약3300개), YOLO모델 전처리,대시보드 텐서보드 구현 및 연동,twili 연동

데이터가공(약1000개), 발표 자료 PPT 제작 데이터가공(약10 00개), 발표 자료 PPT 제작

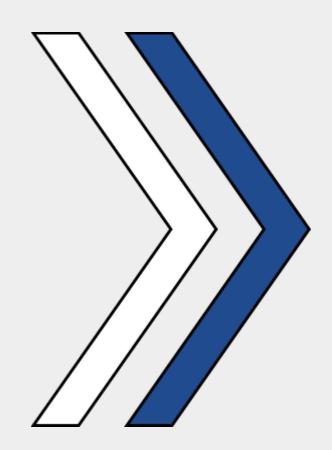
이름 : 조영규

이름 : 김우혁

이름 : 유혜정

이름 : 정날빛

데이터 처리 및 AI 모델 설계



- 데이터 수집 및 데이터 전처리
- 모델 데이터 학습과정(표)

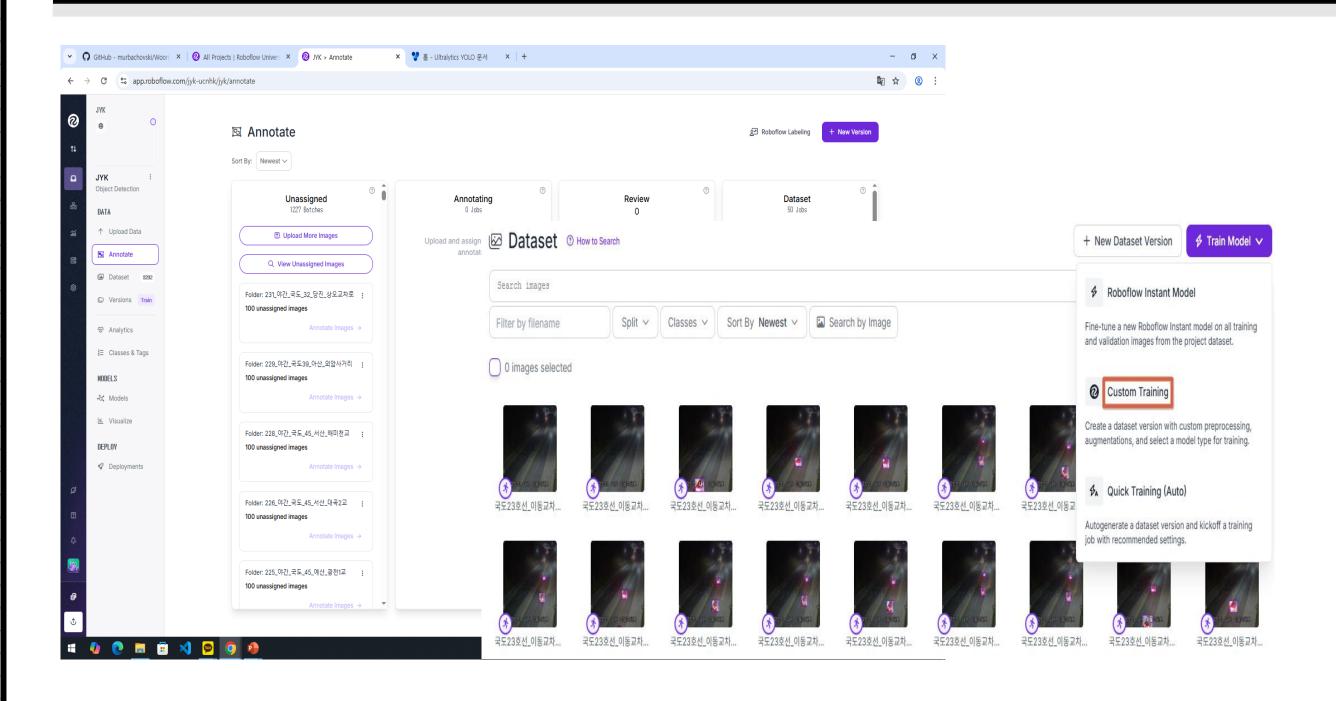
데이터(프레임) 수집 소스코드

```
import os, time, subprocess, pandas as pd
from urllib.parse import quote
from seleniumwire import webdriver
from selenium.webdriver.common.by import By
# 기본 설정
API_KEY = "API키"
df = pd.read_excel("OpenDataCCTV.xlsx")
save_dir = "cctv_videos"
os.makedirs (save_dir, exist_ok=True)
# 셀레니움 드라이버 실행
driver = webdriver.Chrome()
for idx, row in df.iterrows():
   cctvid = row["CCTVID"]
   name = quote (row.get("CCTVNAME", ""))
   url = f"https://www.utic.go.kr/jsp/map/openDataCctvStream.jsp?key={API_KEY}&cctvid={cctvid}&cctvName={name}&kind=KB"
   driver.get(url)
   time.sleep(5)
       video_url = driver.find_element(By.TAG_NAME, "source").get_attribute("src")
       if video_url.startswith("//"):
           video_url = "https:" + video_url
       output = os.path.join(save_dir, f"{cctvid}.mp4")
       subprocess.run(["ffmpeg", "-y", "-i", video_url, "-t", "10", "-c", "copy", output])
    except:
       continue
driver.quit()
```

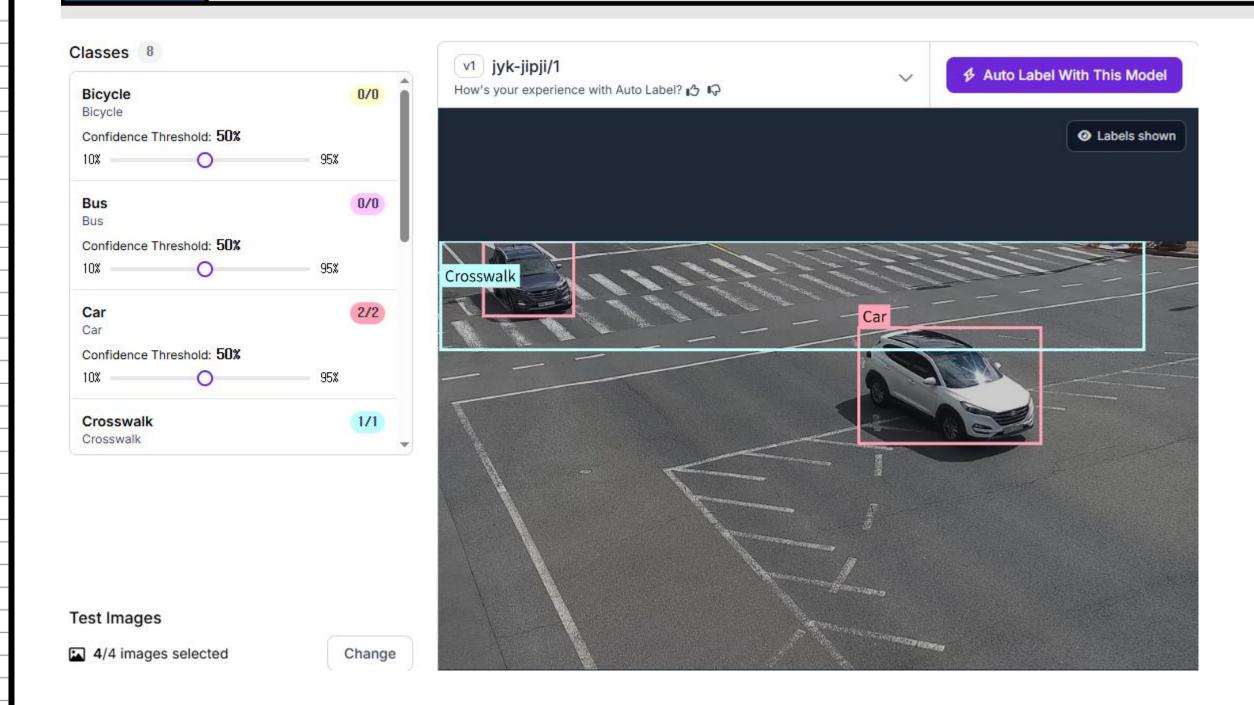
데이터(영상) 수집 소스코드

```
import cv2, os
#CCTV URL 리스트와 이름 리스트
urls = cctv_play ["cctvurl"]
names = cctv_play ["cctvname"]
# 저장할 프레일 수
frames_per_cctv = 100
for idx, (url, name_raw) in enumerate (zip(urls, names)):
   cap = cv2.VideoCapture(url)
   if not cap.isOpened():
       continue
    #폴더 이름 정제 및 생성
    name = name_raw.strip().replace("[", "").replace("]", "").replace("", "_")
    folder = f"{idx} {name}"
    os.makedirs (folder, exist_ok=True)
    # 프레임 저장
    count = 0
    while count < frames_per_cctv:</pre>
       ret, frame cap.read()
       if not ret:
           break
       filepath = os.path.join(folder, f"{name}_{count + 1:03}.jpg")
        cv2.imwrite(filepath, frame)
       count+= 1
    cap.release()
```

데이터 수집 및 전처리1



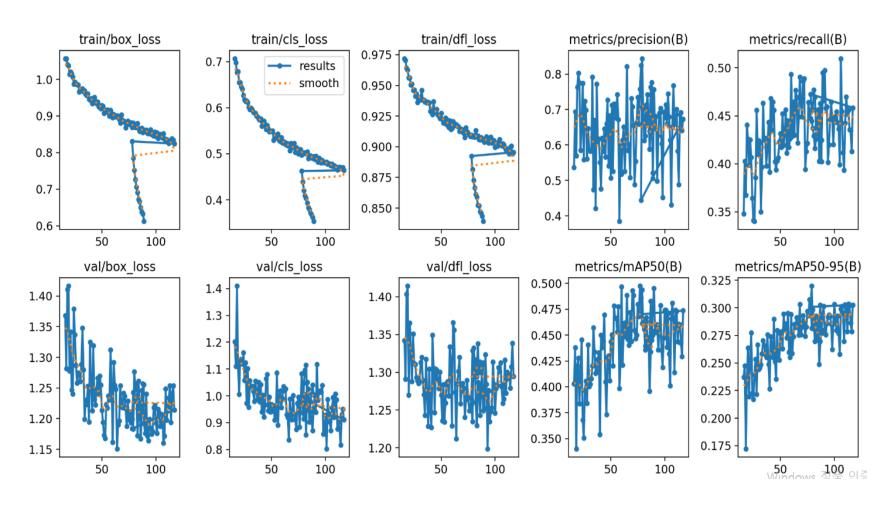
데이터 수집 및 전처리2



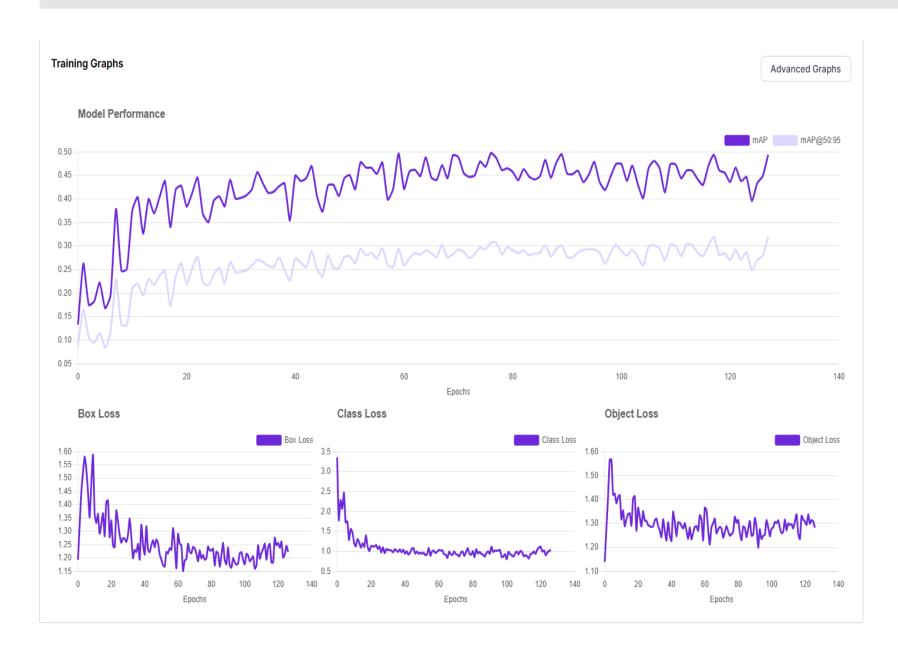
○❸ 모델 훈련 결과(전)- (120회)



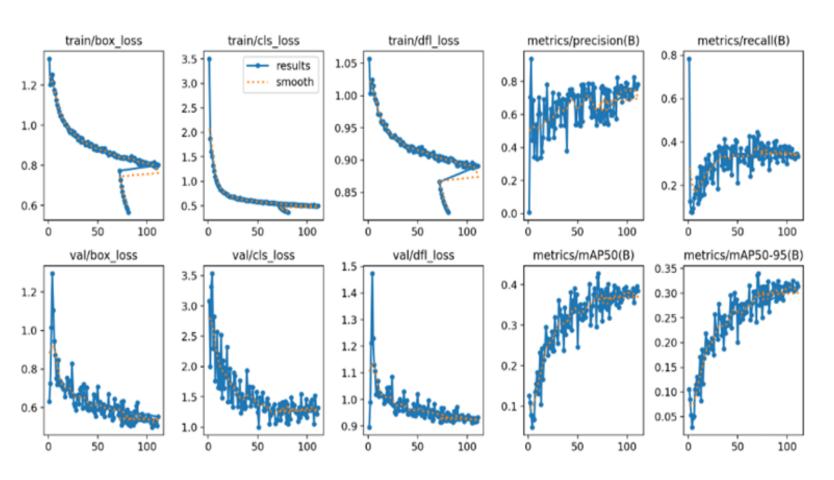
Advanced Training Graphs



○❸ 모델 훈련 결과(후)- (130회)

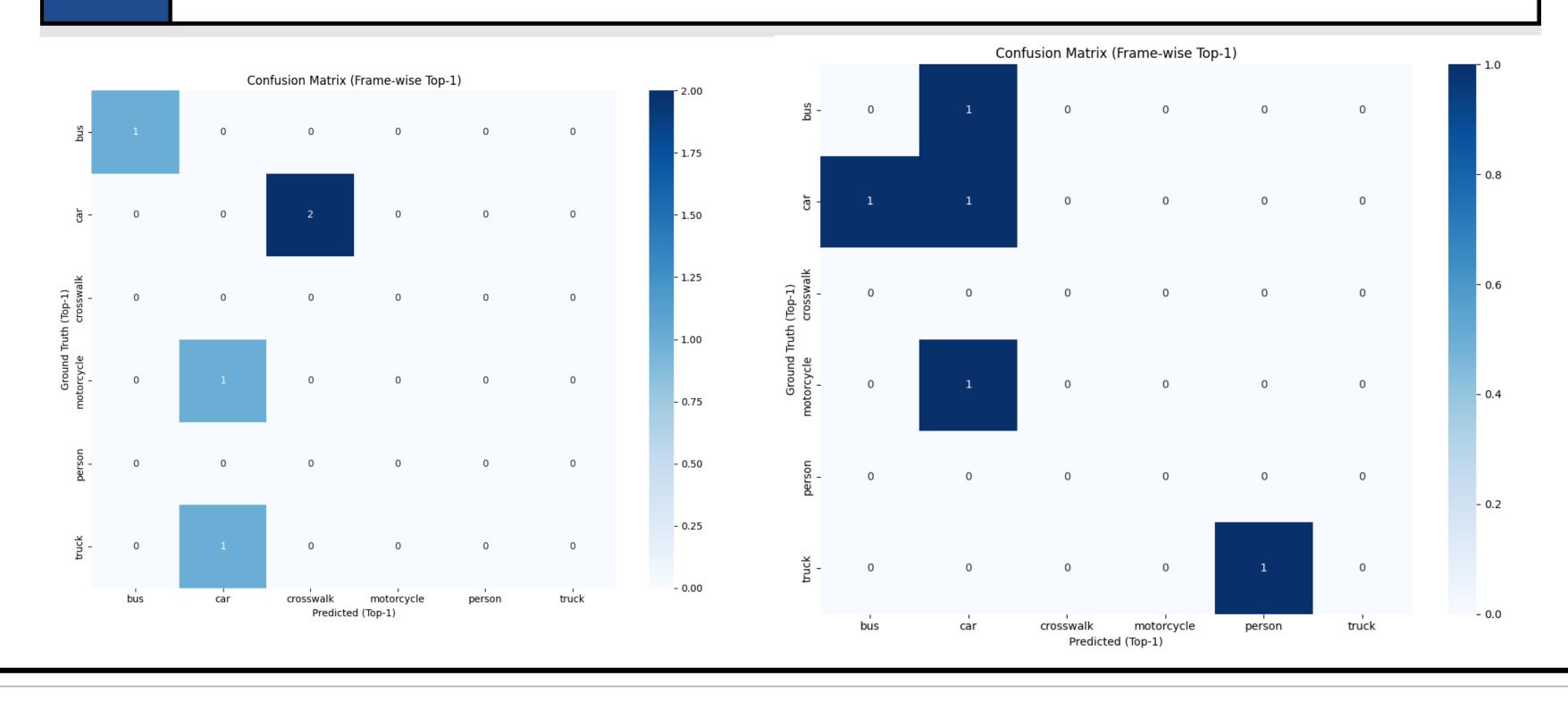


Advanced Training Graphs

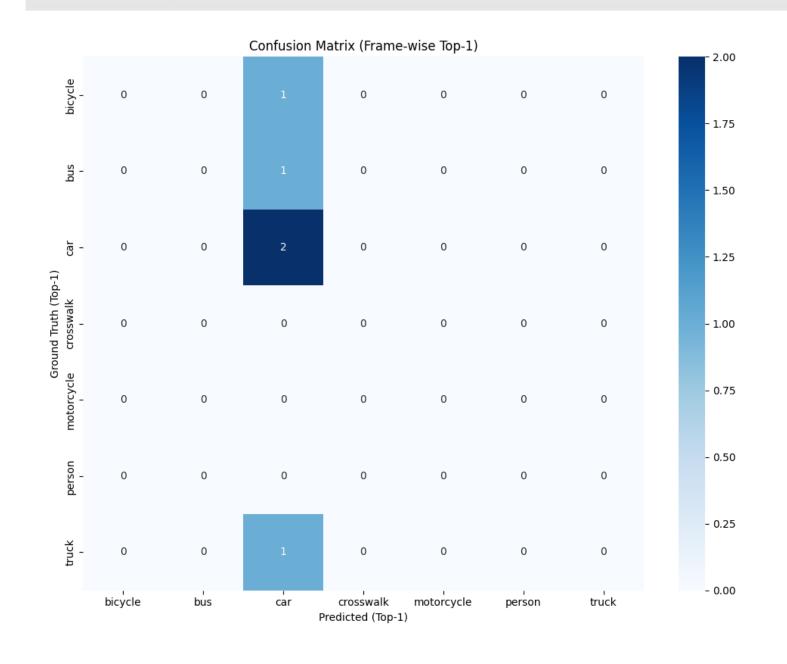


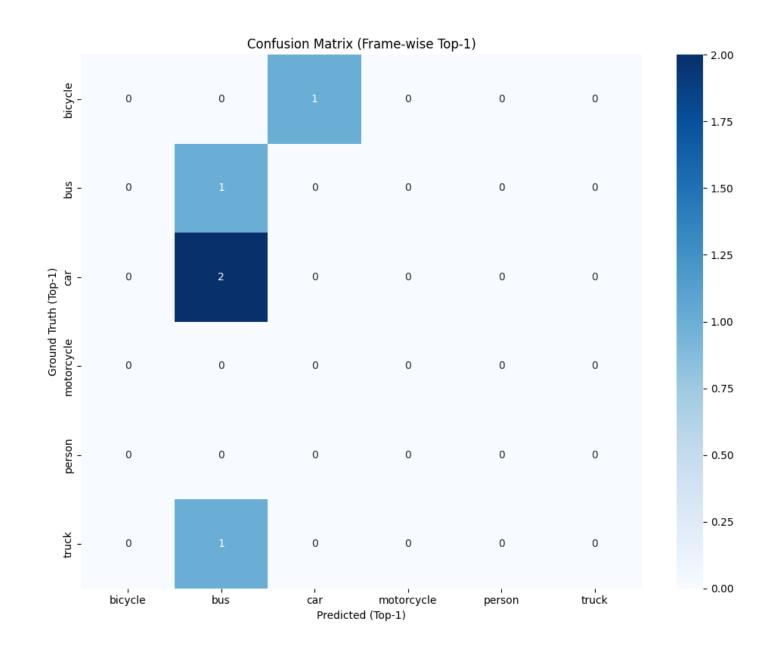
 \times

OB 모델 데이터 학습 - Confusion Matrix

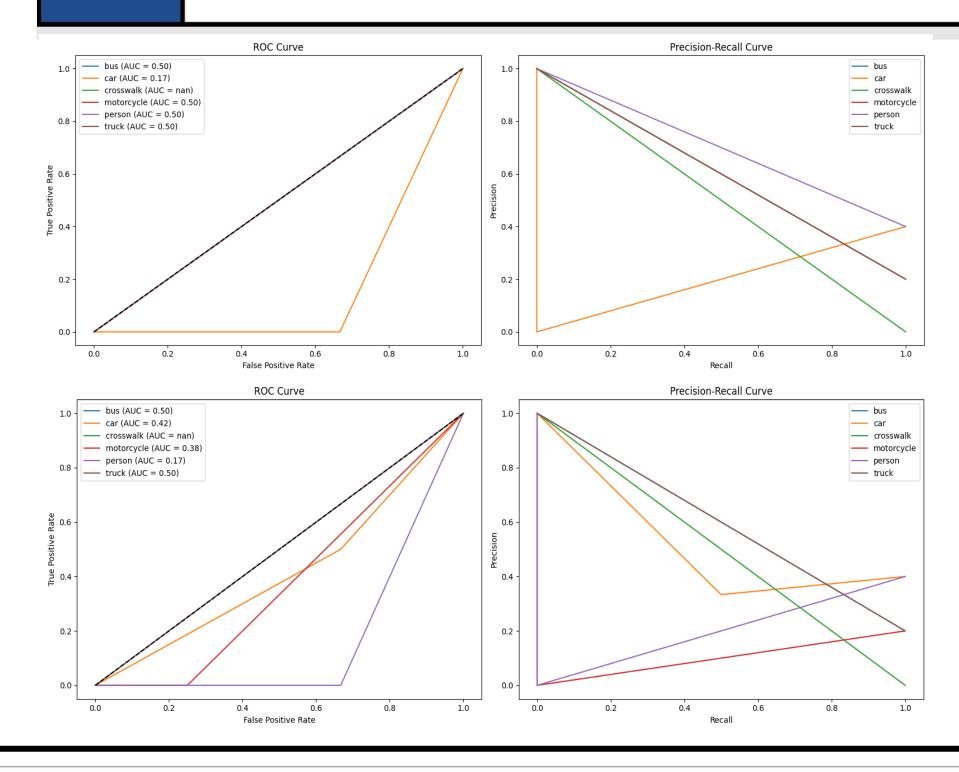


OB 모델 데이터 학습 - Confusion Matrix

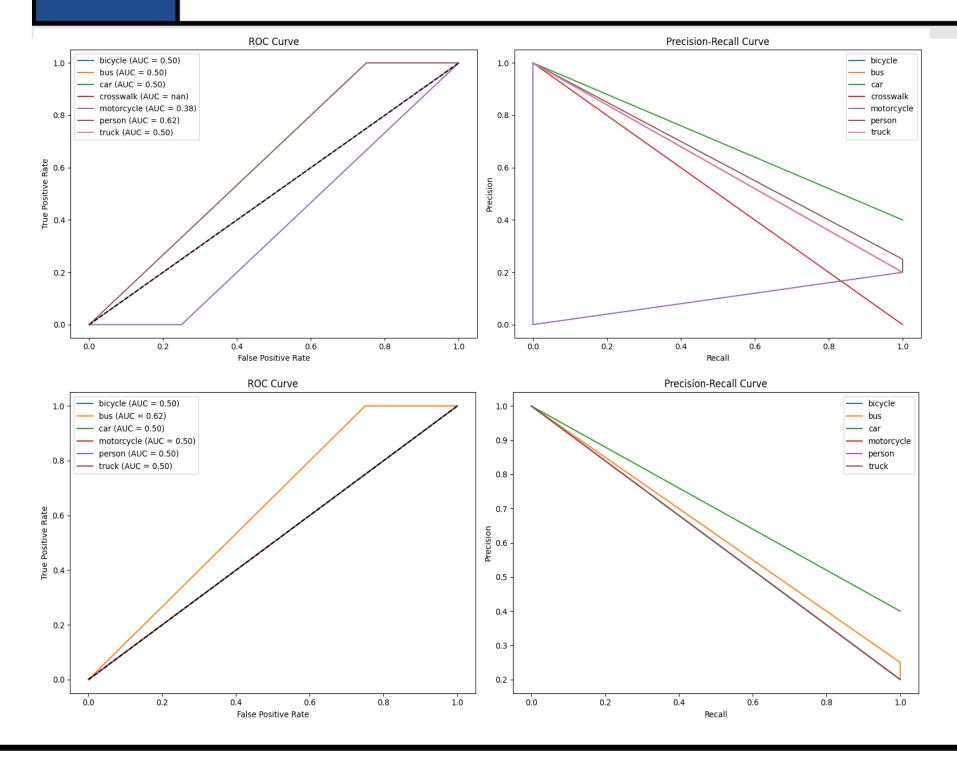




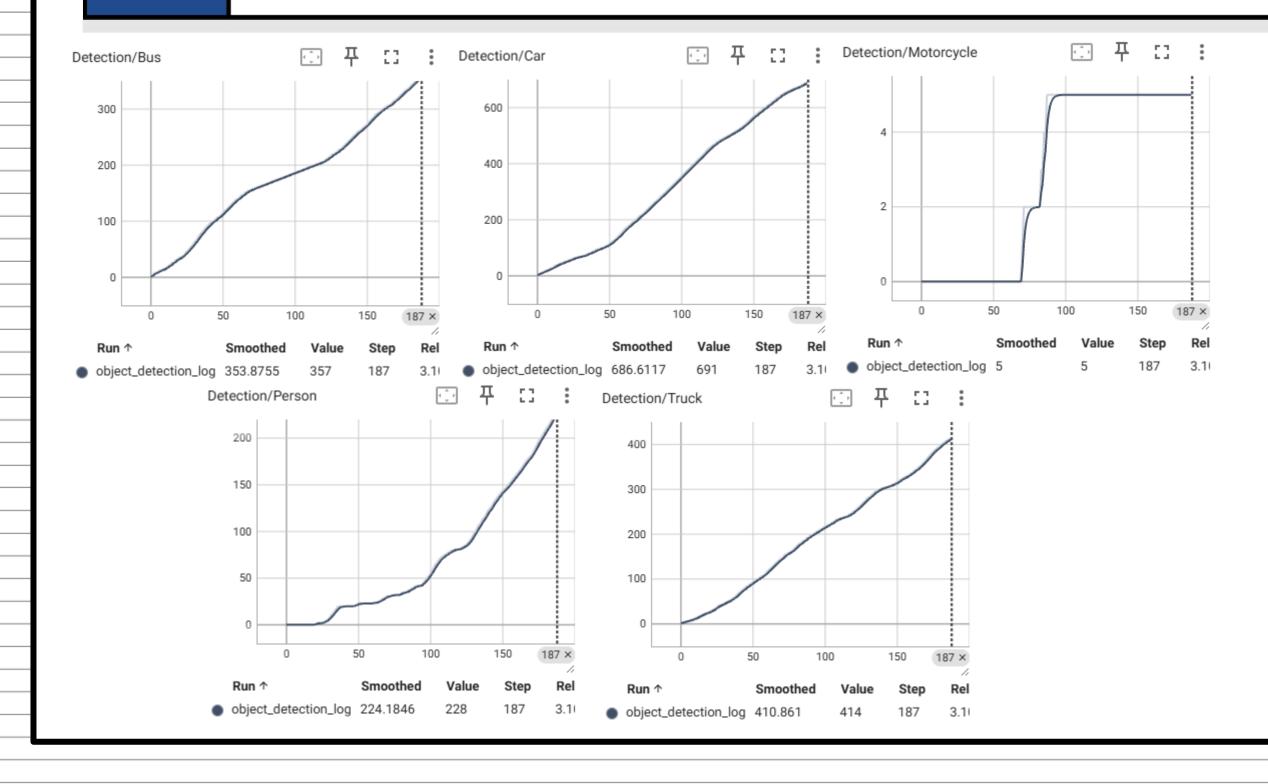
OB 모델 데이터 학습 - ROC Curve, PR Curve



OB 모델 데이터 학습 - ROC Curve, PR Curve



OB 모델 데이터 학습 - tensorboard



모델 데이터 학습 - dashboard

③ 실시간 위반 탐지 대시보드 ③ 영상업로드 (mp4, avi) Drag and drop file here Limit 200MB per file • MP4, AVI, MPEG4 MakaoTalk_20250415_123136238.mp4 1.7MB 제 처리할 최대 프레임수 1000

📊 탐지 통계

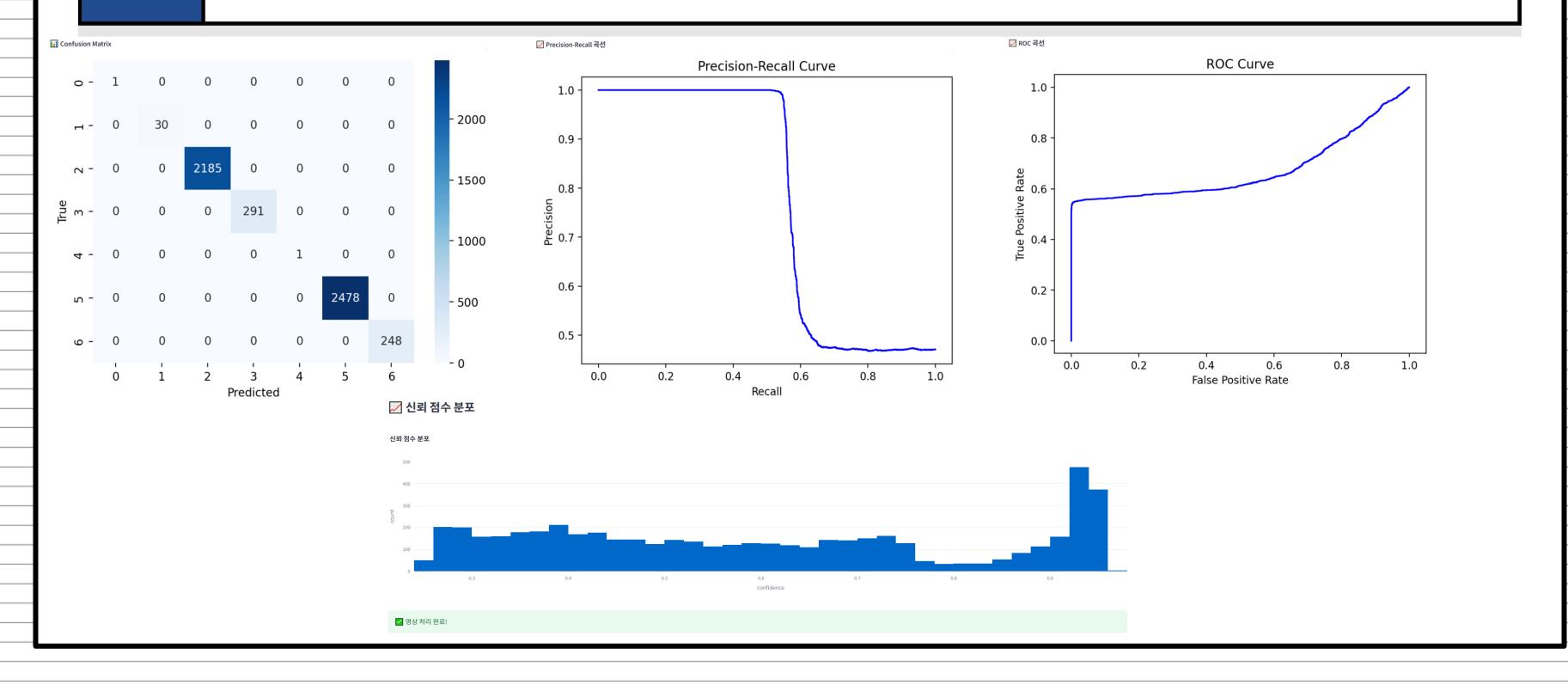
```
"car": 2185
"person": 2478
"crosswalk": 291
"truck": 248
"bus": 30
"bicycle": 1
"motorcycle": 1
}

CSV 로그 저장 완료: logs/detection_log_20250417_185613.csv
```

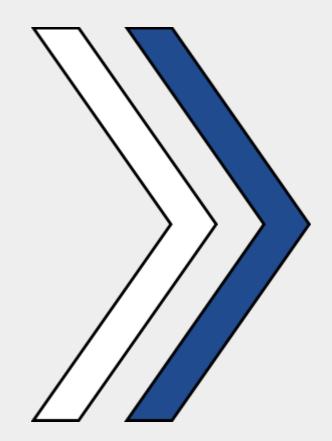
☑ 모델 평가 지표

```
정확도 (Accuracy): 1.00
정밀도 (Precision): 1.00
재현율 (Recall): 1.00
F1-Score: 1.00
평균 정밀도 (Average Precision): 0.78
```

©을 모델 데이터 학습 - dashboard

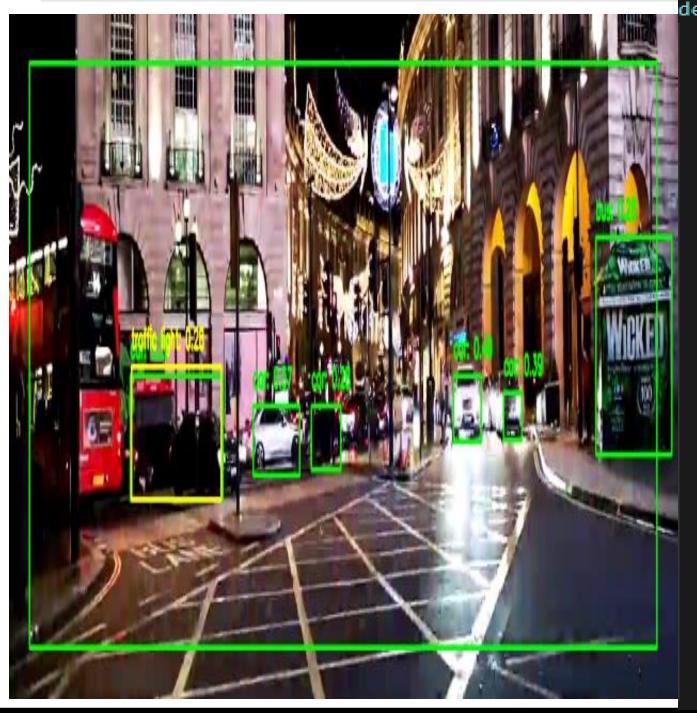


시스템 구현 및 작동 시연



- 구현 화면 소개
- 감지 시나리오 시연(영상 or 이미지)
- 모델 분석 및 검증

이4 구현 화면 소개(전)



```
def draw_rectangle(event, x, y, flags, param):
    global drawing, roi_points, roi_rect, current_roi
    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = True
        roi points = [(x, y)]
    elif event == cv2.EVENT MOUSEMOVE:
        if drawing:
            roi_rect = (roi_points[0][0], roi_points[0][1], x, y)
    elif event == cv2.EVENT LBUTTONUP:
        drawing = False
        roi_rect = (roi_points[0][0], roi_points[0][1], x, y)
        if roi rect:
            roi_list.append(roi_rect)
        roi_points = []
        roi rect = None
    elif event == cv2.EVENT RBUTTONDOWN:
        # 마우스 오른쪽 버튼으로 ROI 삭제
        for i, roi in enumerate(roi list):
            x1, y1, x2, y2 = roi
            if (x1 \le x \le x2 \text{ and } y1 \le y \le y2) or (x2 \le x \le x1 \text{ and } y2 \le y \le y1):
                roi list.pop(i)
                break
```

04 구현 화면 소개 (후)

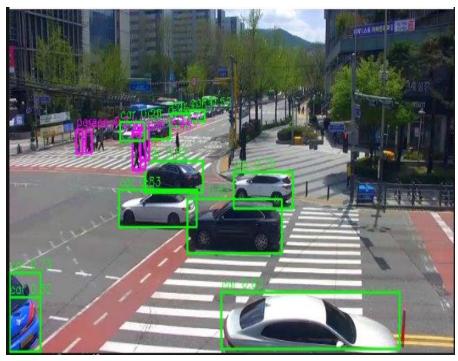


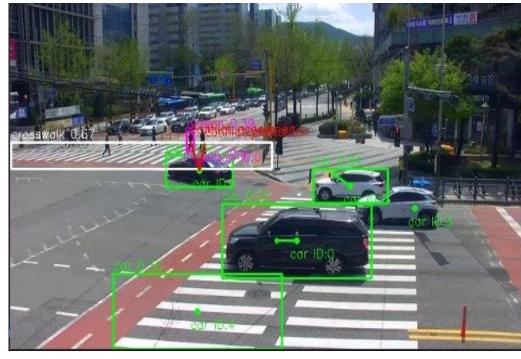
O4 구현 화면 소스코드 (후)

```
import streamlit as st
       import cv2
       import tempfile
       from roboflow import Roboflow
       import numpy as np
       from collections import deque
       st.set_page_config(layout="wide")
10
       # 제목
       st.title("🛭 실시간 위반 탐지 대시보드")
11
12
      # 영상 업로드
13
       video_file = st.file_uploader("▲ 영상 업로드 (.mp4)", type=["mp4", "avi"])
14
15
16
       # Roboflow 모델 불러오기
17
       rf = Roboflow(api_key="KLlcHdVtvytxtpDiXA0W")
18
       project = rf.workspace("joyk").project("jyk-jipji")
19
       version = project.version(2)
20
       model = version.model
21
22
       CONFIDENCE_THRESHOLD = 0.25
23
       IOU_THRESHOLD = 0.45
24
25 🗸
       def process_video(video_path):
           cap = cv2.VideoCapture(video_path)
26
27
           frame_placeholder = st.empty()
28
           stats = {"person": 0, "car": 0, "bus": 0, "truck": 0, "motorcycle": 0, "conflict_pedestrian": 0}
29
           frame_count = 0
30
31
           while cap.isOpened():
32
              ret, frame = cap.read()
33
              if not ret or frame_count > 300:
34
                  break
35
```

```
36
37
                   results = model.predict(frame, confidence=CONFIDENCE_THRESHOLD, overlap=IOU_THRESHOLD)
38
                   predictions = results.json()["predictions"]
39
                   for pred in predictions:
                      cls = pred["class"].lower()
                      x = int(pred['x'] - pred['width'] / 2)
                      y = int(pred['y'] - pred['height'] / 2)
                       w = int(pred['width'])
                       h = int(pred['height'])
                       conf = pred['confidence']
                      if cls in stats:
                          stats[cls] += 1
                       color = (0, 255, 0)
                       cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
51
                       cv2.putText(frame, f"{cls} {conf:.2f}", (x, y - 5),
52
                                  cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 1)
53
               except Exception as e:
                   st.warning(f"예측 중 오류 발생: {e}")
55
               frame = cv2.resize(frame, (720, 480))
57
               frame_placeholder.image(frame, channels="BGR")
               frame_count += 1
60
           cap.release()
61
           return stats
62
       # 업로드된 비디오 처리
       if video_file:
65
           tfile = tempfile.NamedTemporaryFile(delete=False)
          tfile.write(video_file.read())
67
           stats = process_video(tfile.name)
69
          # 결과 통계 표시
70
           st.subheader("in 탐지 통계")
           st.write(stats)
```

04 모델 분석 및 검증





● YOLO11 모델/ 제작 모듈 차이점 ●

○ 인식되는 클래스가 더 많다(새로만든 클래스 : crosswalk 생성)

✓ 거리가 가까우면 거리표시와 함께 위험을 알린다.

✓ 시간과 날씨 상관없이 인식이 가능하다.

☑ 벡터 방향 표시로 불법유턴 같은 상황도 감지할수 있다.

04 결과

4월 17일 목요일 [국외발신] Sent from your Twilio trial account - 🌉 위반 차량 감지됨 - 이미지 저장 완료: captur [국외발신] es/ violation_2025041 **7**_124626.jpg

오후 12:46

● 교차로에서 돌발 진입하는 차량을 실시간으로 감지.

- YOLO 모델을 활용하여 차량의 위치와 방향을 정확히 인식
- 돌발 진입 사건 발생 시 실시간 경고 알림 표시

☞ 반영계획



향후 반영 계획

- 인도 및 차도 구분
- 차량과 차량 사이의 거리 계산 및 알림 서비스
- 여러 종류의 차량 데이터 수집 및 가공
- 여러 종류의 동물 데이터 수집 및 가공
- 라벨링 신뢰도 조정

Q&A

감사합니다.