

○ 2025.04 ○

# “ 교차로 돌발 진입 감지 ”

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

# 프로젝트 플래너

	<div><div></div>기간 계획</div>	<div><div></div>실제 시작</div>															
		<div><div></div>% 완료</div>															
활동	계획 시작	기간 계획	실제 시작	실제 기간	완료율	4월											
						3	4	7	9	10	11	14	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%												
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%												
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

01

프로젝트 개요 및 개발 배경

02

시스템 구성 및 기술 스택

03

데이터 처리 및 AI 모델 설계

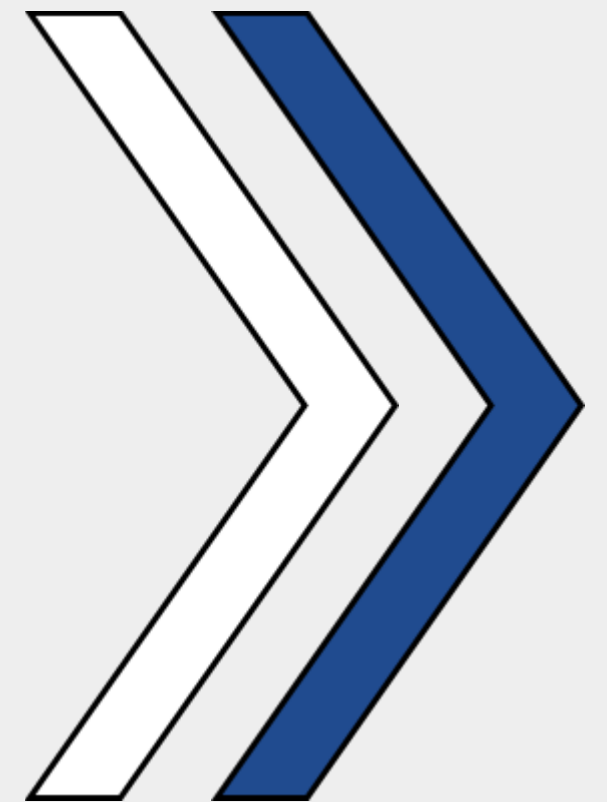
04

시스템 구현 및 작동 시연

05

결과 정리 및 향후 계획

# 프로젝트 개요 및 개발 배경



- 문제 정의 및 필요성
- 프로젝트 목표

# 01

## 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, 보행자  
무단횡단, 장애물 정지 차량 등 예측 불가능한  
이벤트가 빈번하게 발생한다.

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

### 필요성

- ☒ AI 기반 실시간 감지 시스템으로 교차로  
위험 상황을 자동 포착 가능
- 
- ☒ 기존 CCTV와 연동 가능 -> 추가 설치  
비용 최소화
- 
- ☒ 위험 발생시 즉각 경고 및 대응 가능
- 
- ☒ 사고 예방, 교통 흐름 안정화,  
시민 안전 확보에 기여

01

## 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 돕습니다.

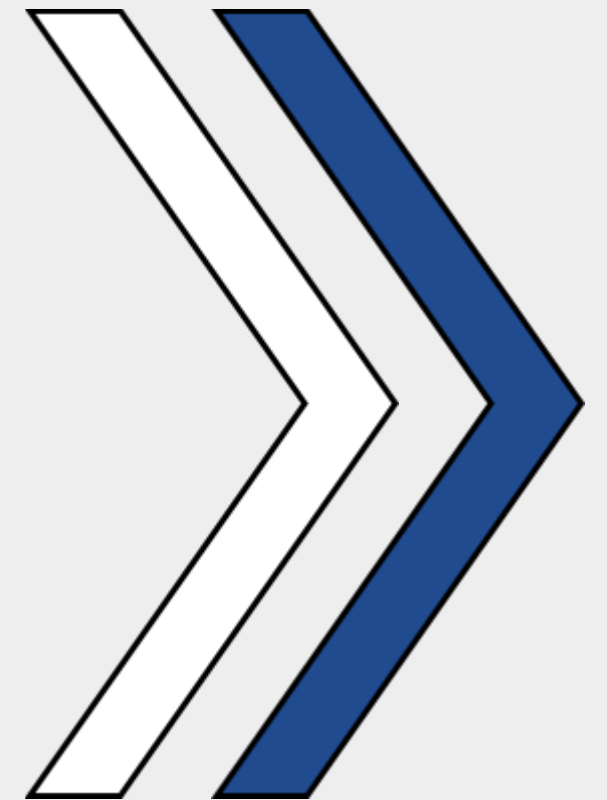


FINAL GOAL

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

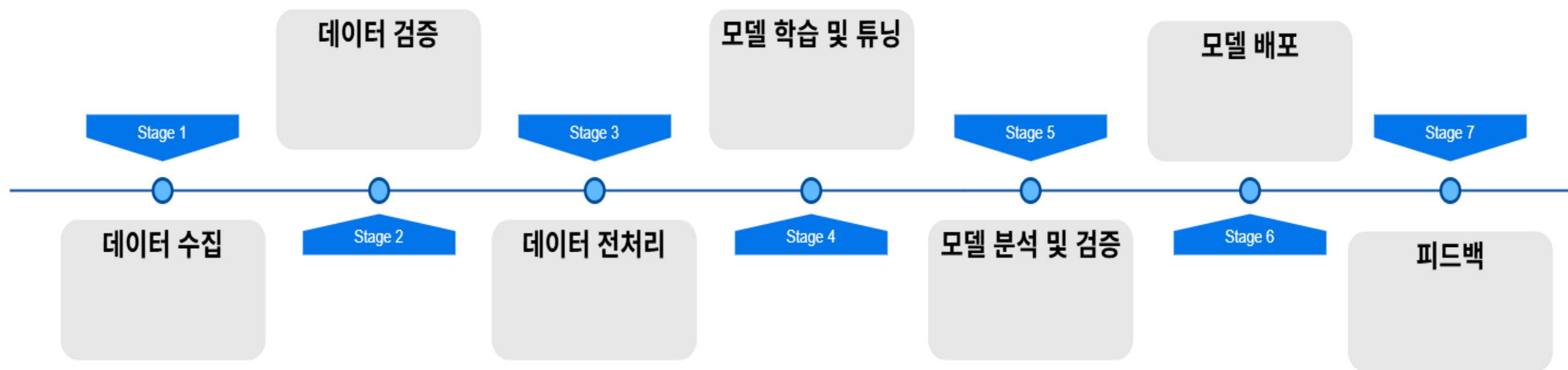
# 시스템 구성 및 기술 스택



- 전체 시스템 흐름도
- 사용 기술 및 개발 환경
- 역할 분담

## 02

# 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

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

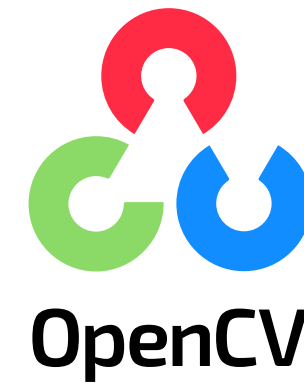
02

## 사용 기술 및 개발 환경

프레임워크



데이터 처리



AI 모델 및 성능 평가



개발 환경

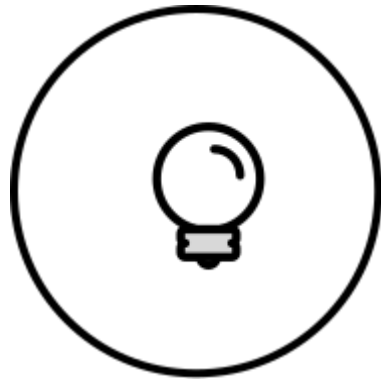


버전 관리



02

## 역할 분담(?)



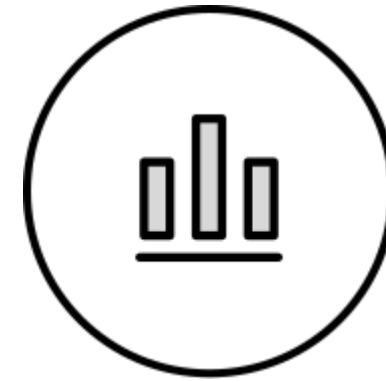
팀장, 총괄,  
데이터, 모델 훈련(약 3300  
개) 검수 및 수정,  
YOLO 모델 전처리 및  
구현 Roboflow APL 연동

이름 : 조영규



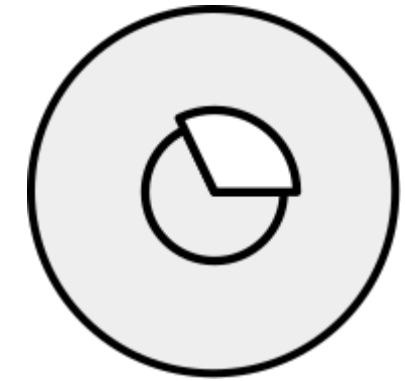
CCTV 프레임 수집(약 200만 개),  
CCTV 동영상 수집(약 200만 개),  
모델 훈련(약 3300 개), YOLO 모델  
전처리, 대시보드 텐서보드 구현 및  
연동, twili 연동

이름 : 김우혁



데이터 가공(약 1000 개  
, 발표 자료 PPT  
제작

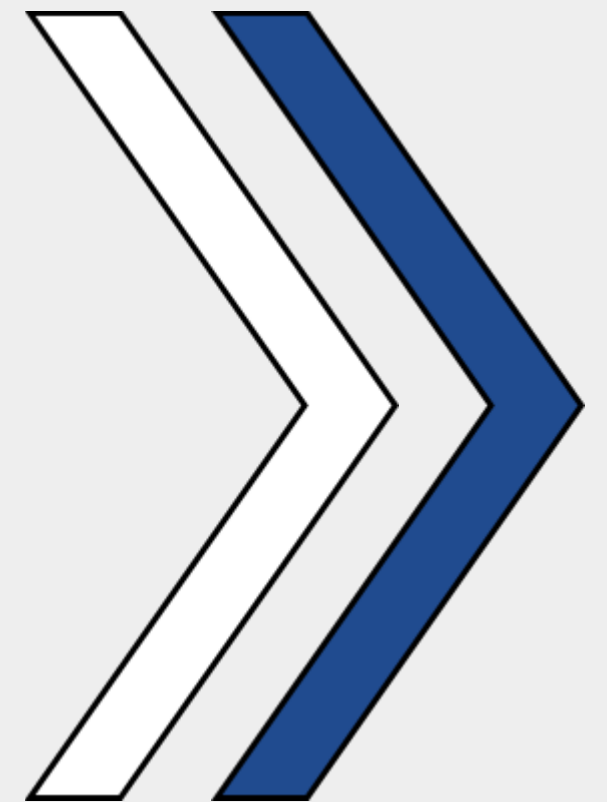
이름 : 유혜정



데이터 가공(약 10  
00 개), 발표 자료  
PPT  
제작

이름 : 정날빛

# 데이터 처리 및 AI 모델 설계



- 데이터 수집 및 데이터 전처리
- 모델 데이터 학습과정(표)

# 03

## 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)

    try:
        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()
```

# 03

## 데이터(영상) 수집 소스코드

```
import cv2, os

#CCTV URL 리스트와 이름 리스트
urls = cctv_play ["cctvurl"]
names = cctv_play ["cctvname"]

# 저장할 프레임 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:
        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()
```

03

# 데이터 수집 및 전처리1

app.roboflow.com/jyk-ucnhk/jyk/annotate

JYK  
Object Detection

DATA  
Upload Data  
Annotate  
Dataset 6292  
Versions Train  
Analytics  
Classes & Tags  
MODELS  
Models  
Visualize  
DEPLOY  
Deployments

Annotate

Sort By: Newest

Unassigned  
1227 Batches

Upload More Images  
View Unassigned Images

Folder: 231\_야간\_국도\_32\_당진\_상오교차로 :  
100 unassigned images  
Annotate Images →

Folder: 229\_야간\_국도\_39\_아산\_외암사거리 :  
100 unassigned images  
Annotate Images →

Folder: 228\_야간\_국도\_45\_서산\_해미천교 :  
100 unassigned images  
Annotate Images →

Folder: 226\_야간\_국도\_45\_예산\_대곡2교 :  
100 unassigned images  
Annotate Images →

Folder: 225\_야간\_국도\_45\_예산\_광천1교 :  
100 unassigned images  
Annotate Images →

Roboflow Labeling + New Version

Annotating 0 Jobs  
Review 0  
Dataset 50 Jobs

Dataset How to Search

Search images

Filter by filename Split Classes Sort By Newest Search by Image

0 images selected

국도23호선\_이동교차... 국도23호선\_이동교차... 국도23호선\_이동교차... 국도23호선\_이동교차... 국도23호선\_이동교차... 국도23호선\_이동교차... 국도23호선\_이동교차...

국도23호선\_이동교차... 국도23호선\_이동교차... 국도23호선\_이동교차... 국도23호선\_이동교차... 국도23호선\_이동교차... 국도23호선\_이동교차... 국도23호선\_이동교차...

+ New Dataset Version Train Model

Roboflow Instant Model  
Fine-tune a new Roboflow Instant model on all training and validation images from the project dataset.

Custom Training  
Create a dataset version with custom preprocessing, augmentations, and select a model type for training.

Quick Training (Auto)  
Autogenerate a dataset version and kickoff a training job with recommended settings.



03

## 데이터 수집 및 전처리2

Classes 8

**Bicycle**

Bicycle

0/0

Confidence Threshold: 50%

10%  95%

**Bus**

Bus

0/0

Confidence Threshold: 50%

10%  95%

**Car**

Car

2/2

Confidence Threshold: 50%

10%  95%

**Crosswalk**

Crosswalk

1/1

Test Images


 4/4 images selected

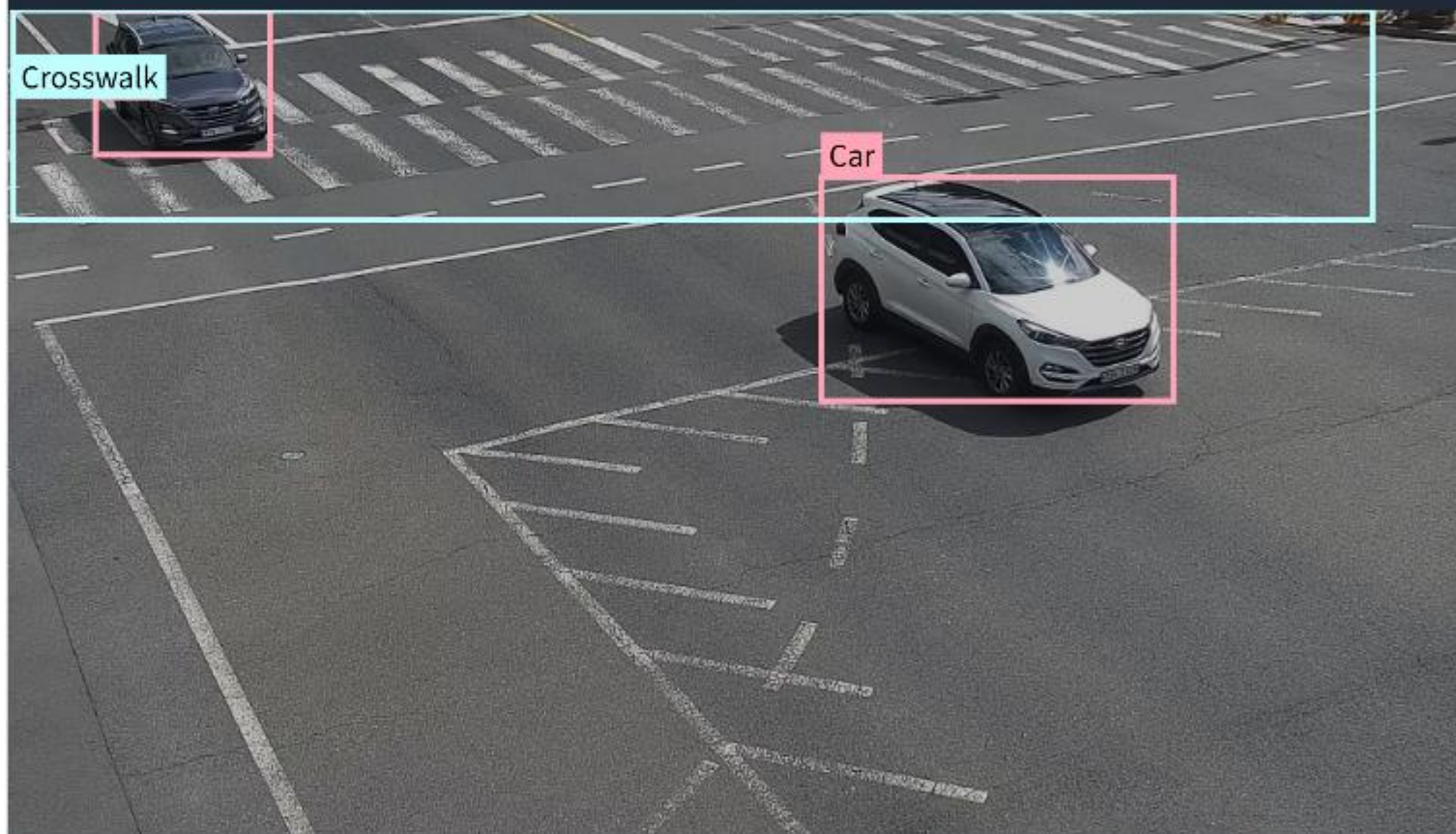
Change

v1 jyk-jipji/1

How's your experience with Auto Label?  

 Auto Label With This Model

 Labels shown





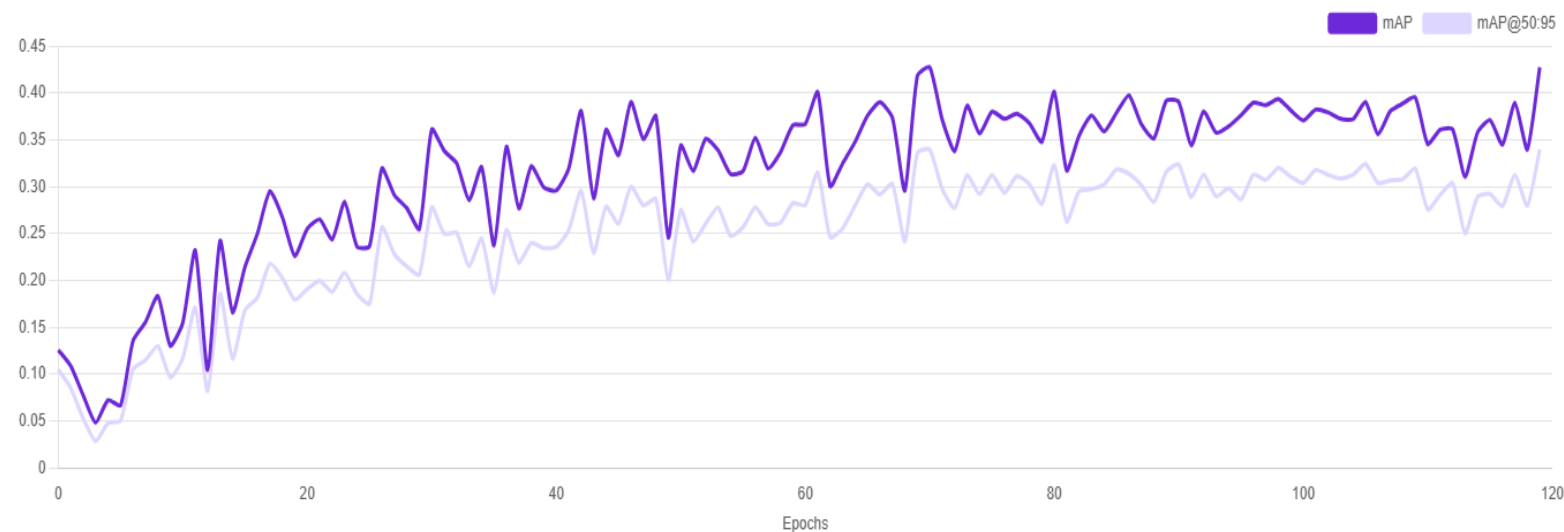
03

# 모델 훈련 결과(전)- (120회)

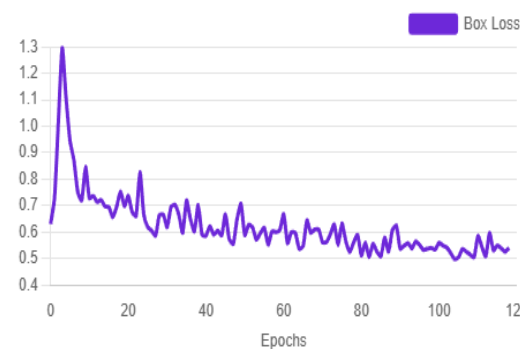
Training Graphs

Advanced Graphs

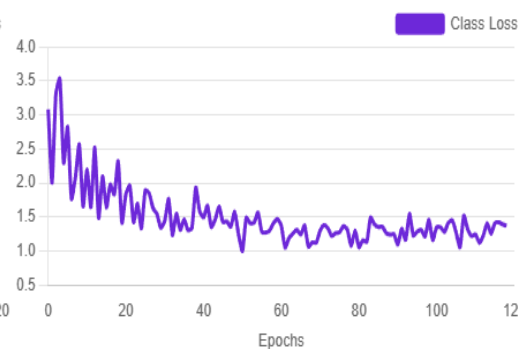
Model Performance



Box Loss



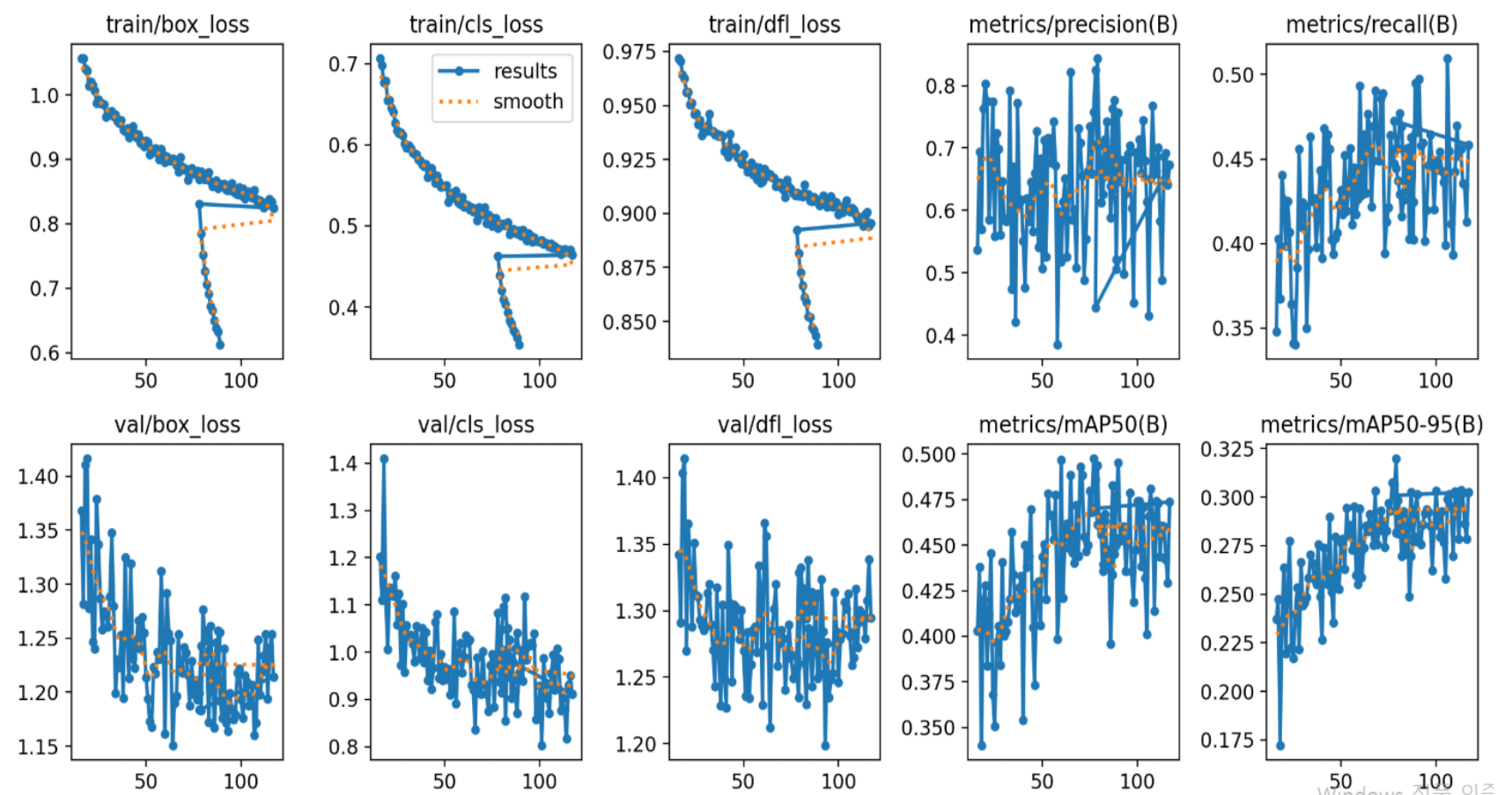
Class Loss



Object Loss



Advanced Training Graphs



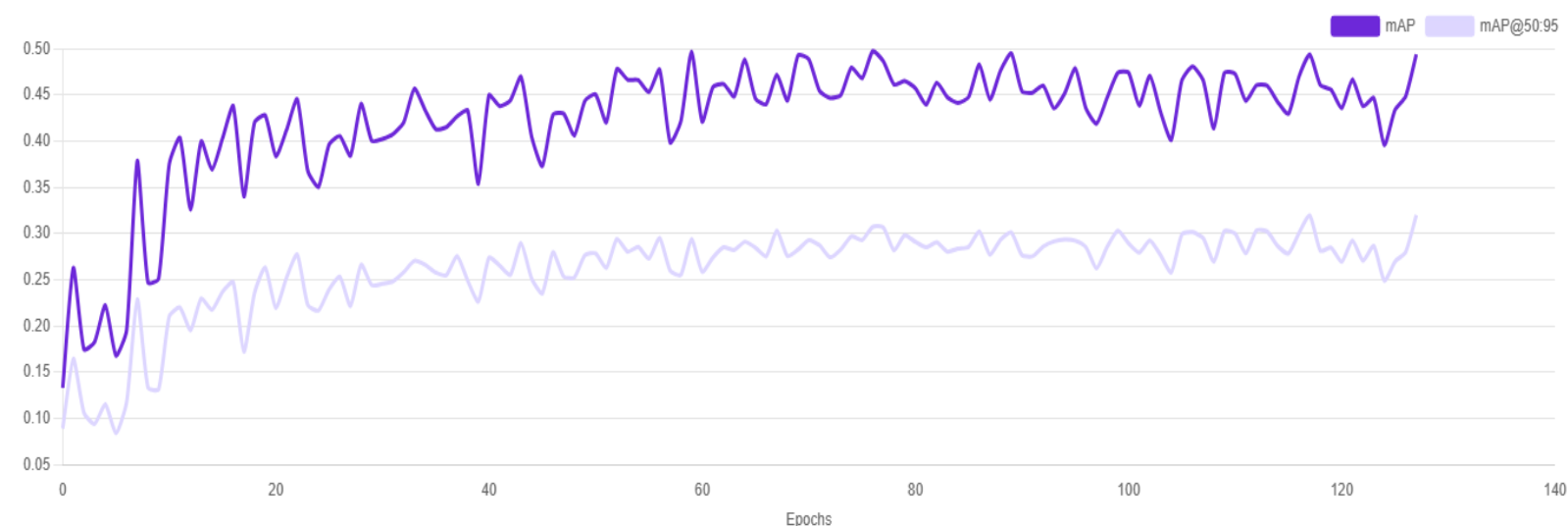
03

# 모델 훈련 결과(후)- (130회)

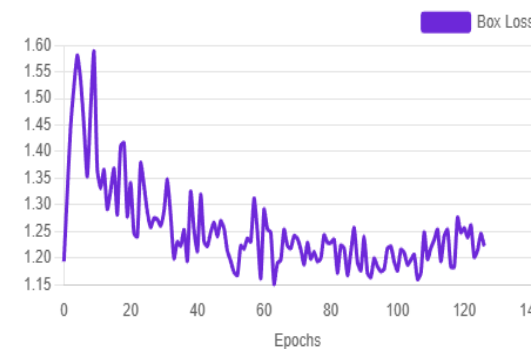
Training Graphs

Advanced Graphs

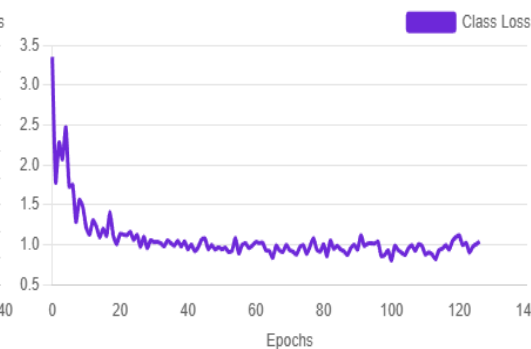
Model Performance



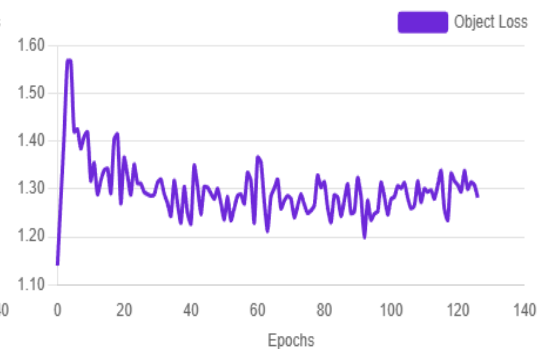
Box Loss



Class Loss

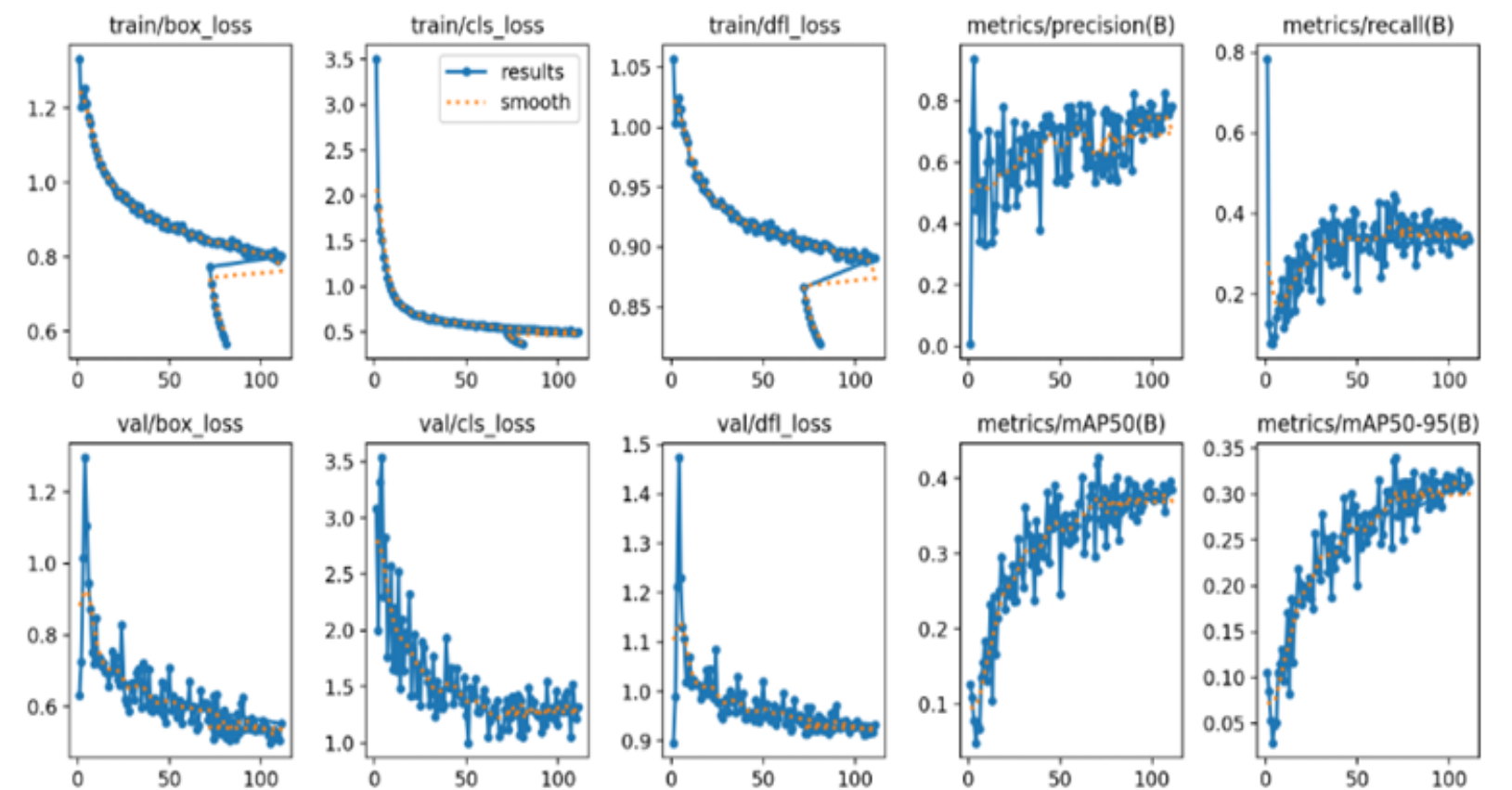


Object Loss



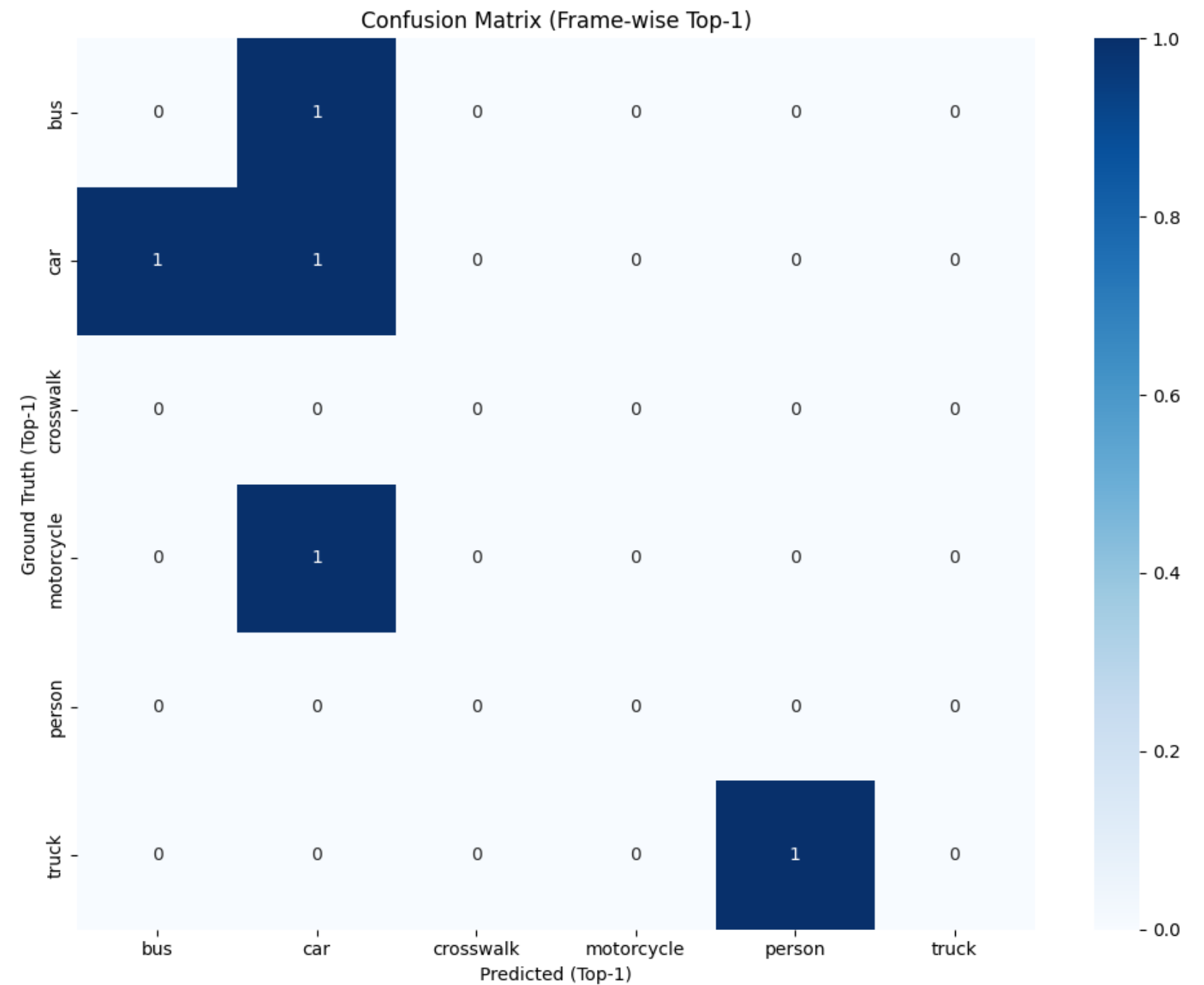
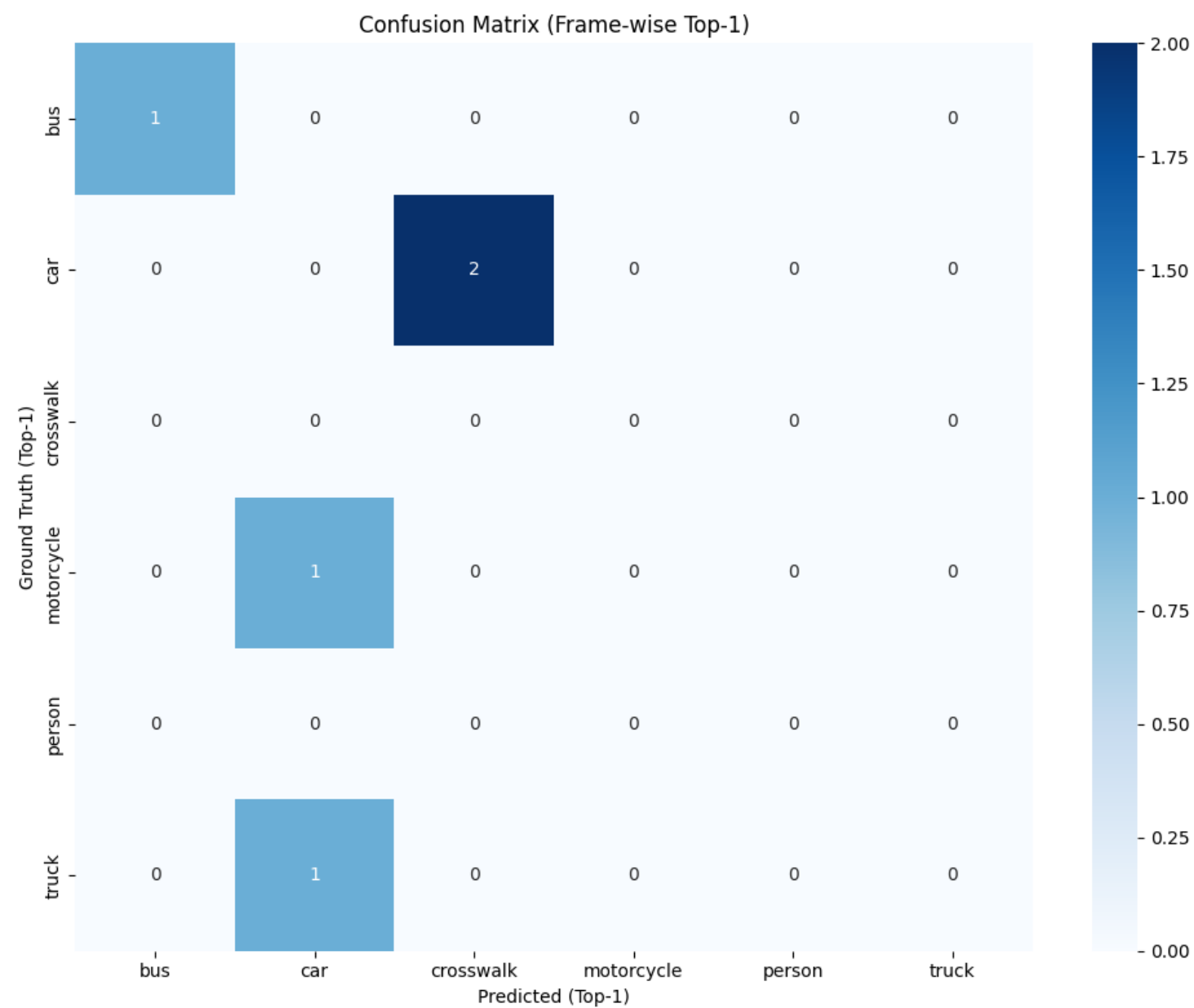
Advanced Training Graphs

×



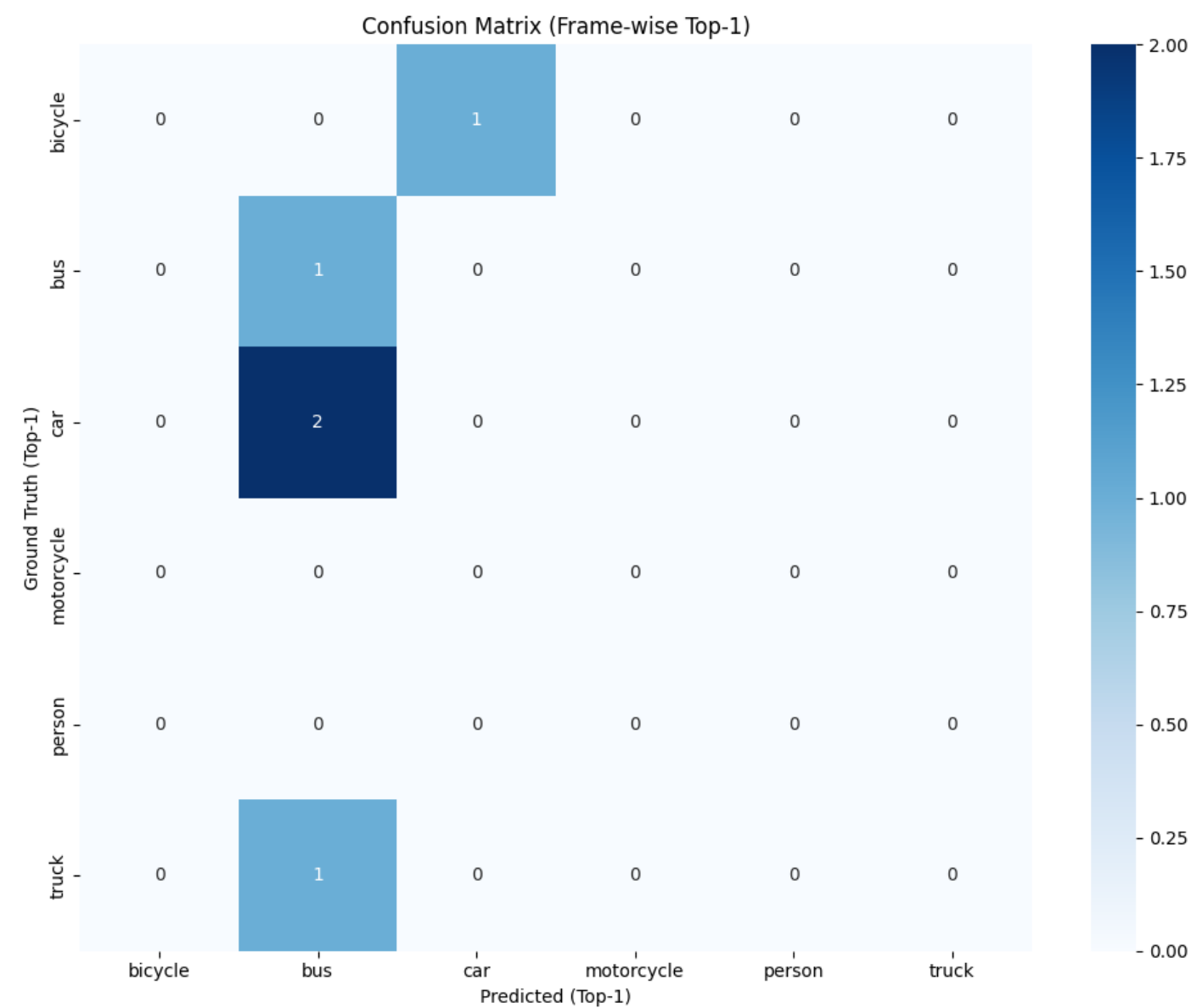
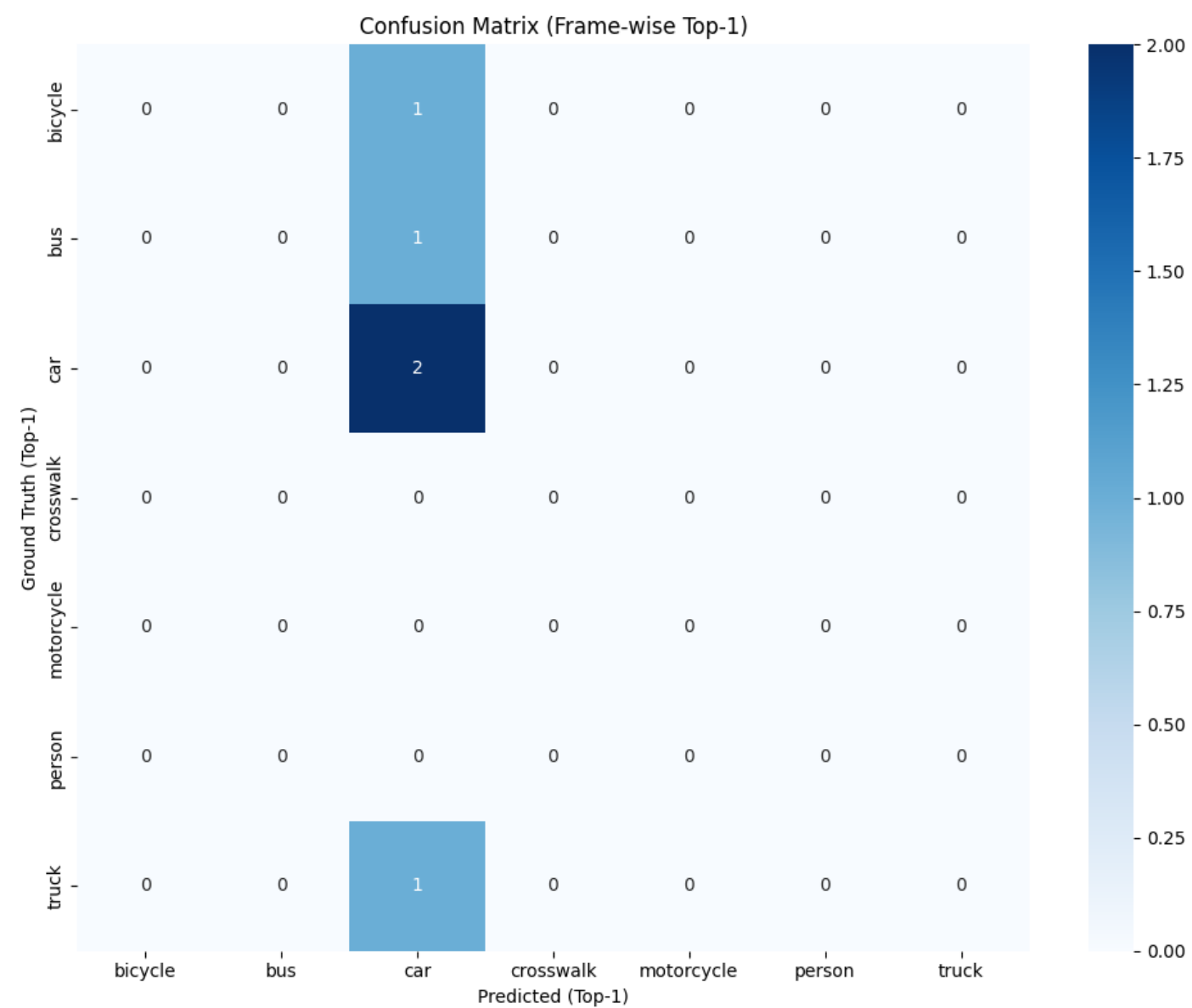
03

# 모델 데이터 학습 - Confusion Matrix



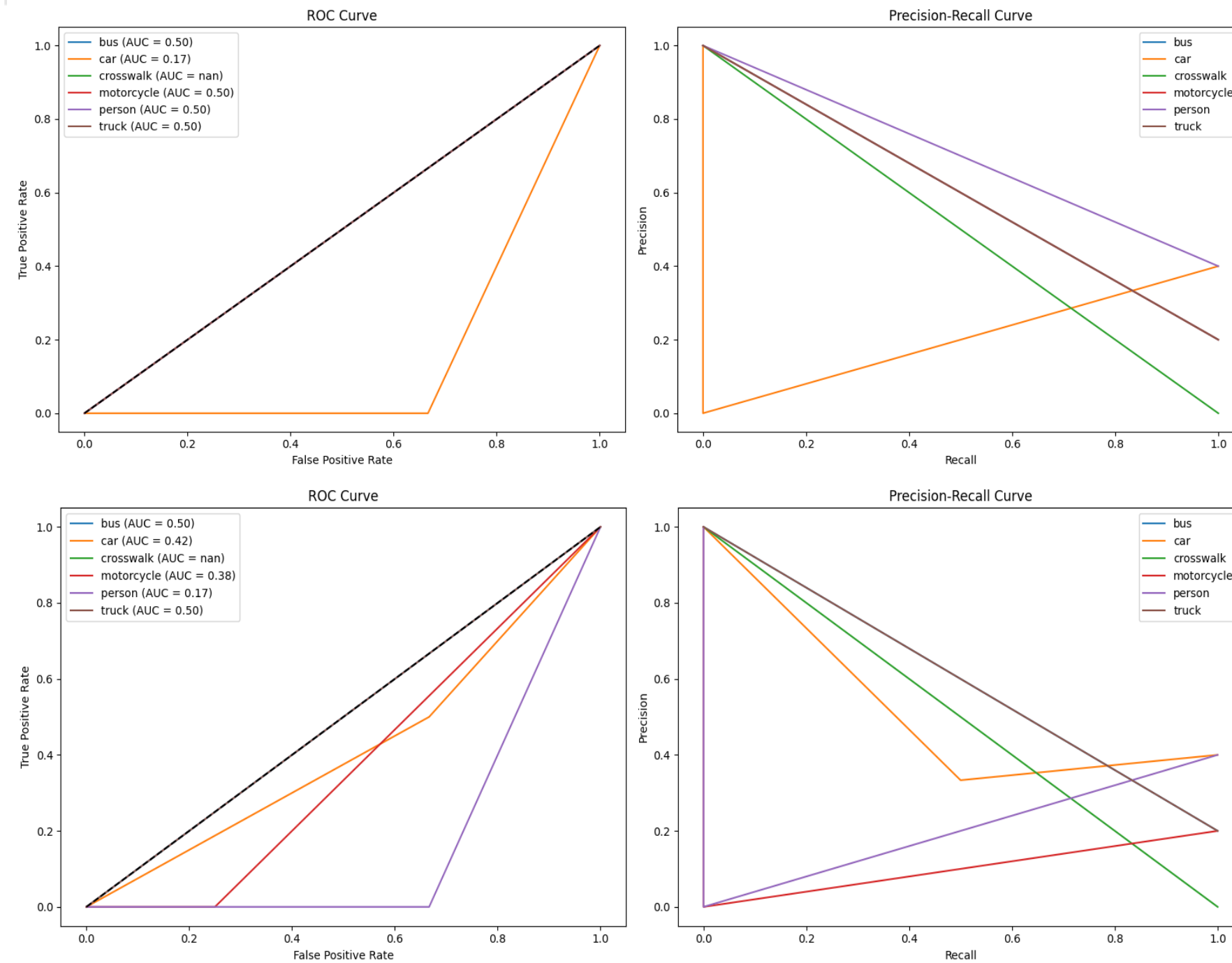
## 03

## 모델 데이터 학습 - Confusion Matrix



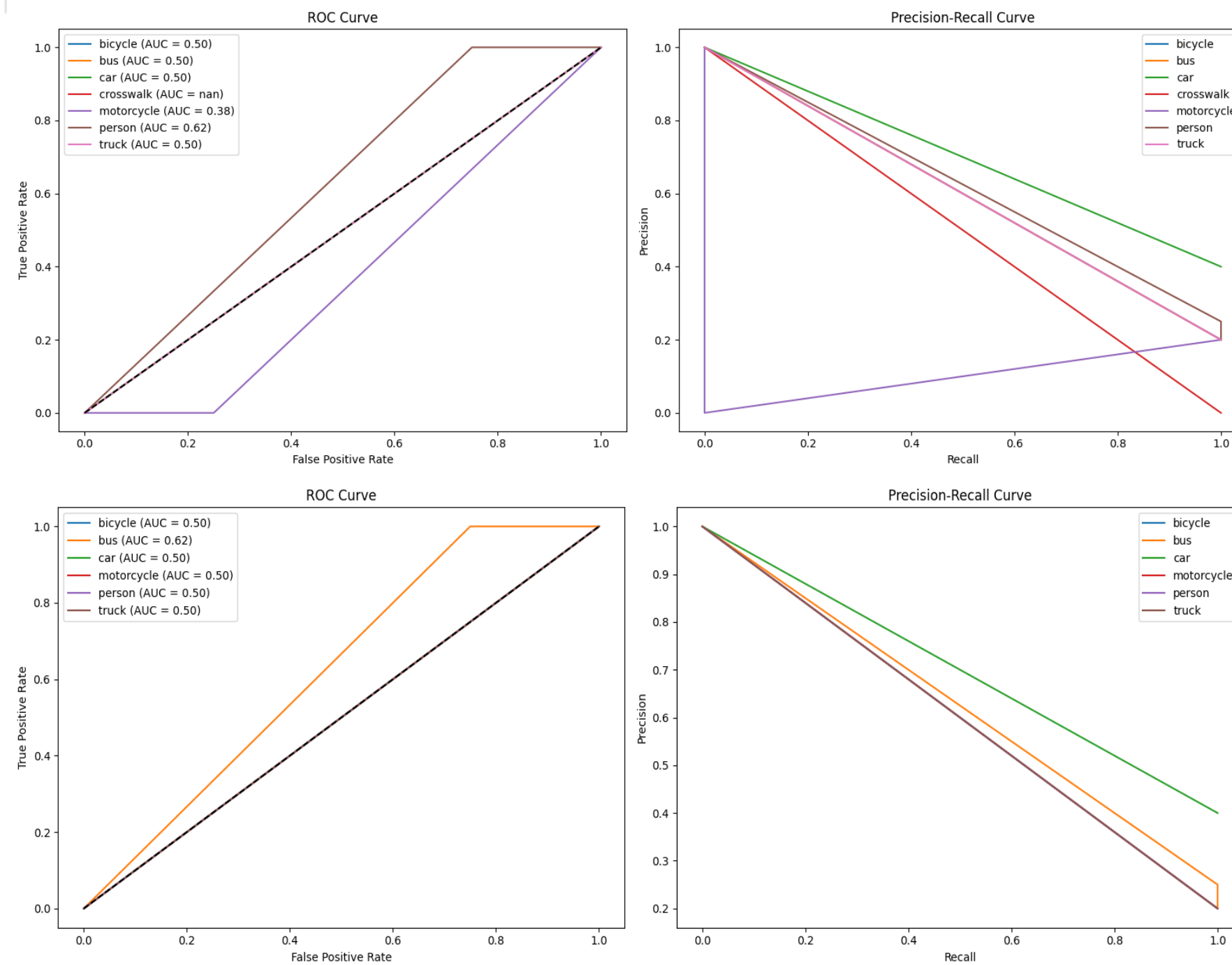
03

# 모델 데이터 학습 - ROC Curve, PR Curve



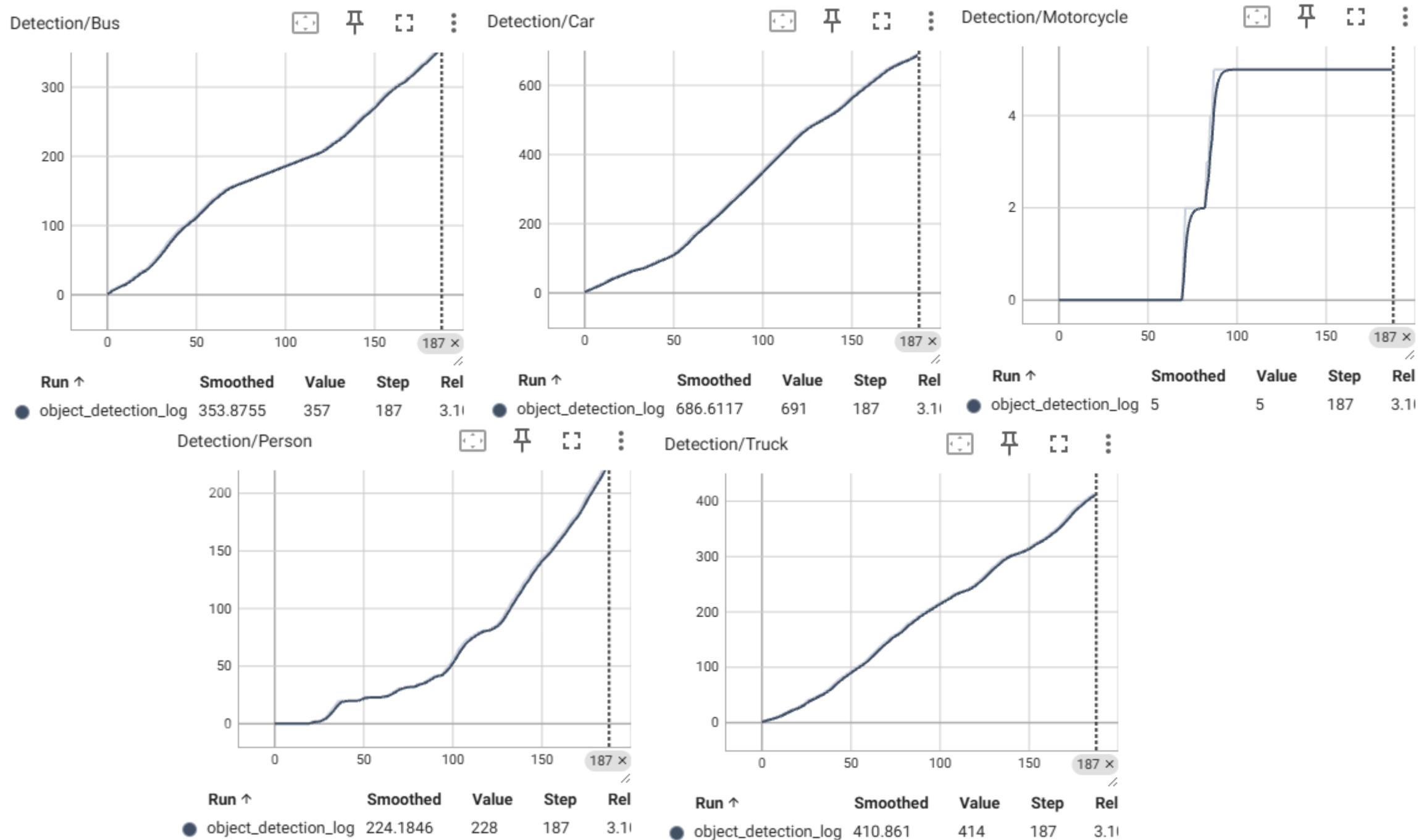
03

# 모델 데이터 학습 - ROC Curve, PR Curve



03

# 모델 데이터 학습 - tensorboard





# 03

## 모델 데이터 학습 - dashboard

### 실시간 위반 탐지 대시보드

영상 업로드 (.mp4, .avi)



Drag and drop file here

Limit 200MB per file • MP4, AVI, MPEG4

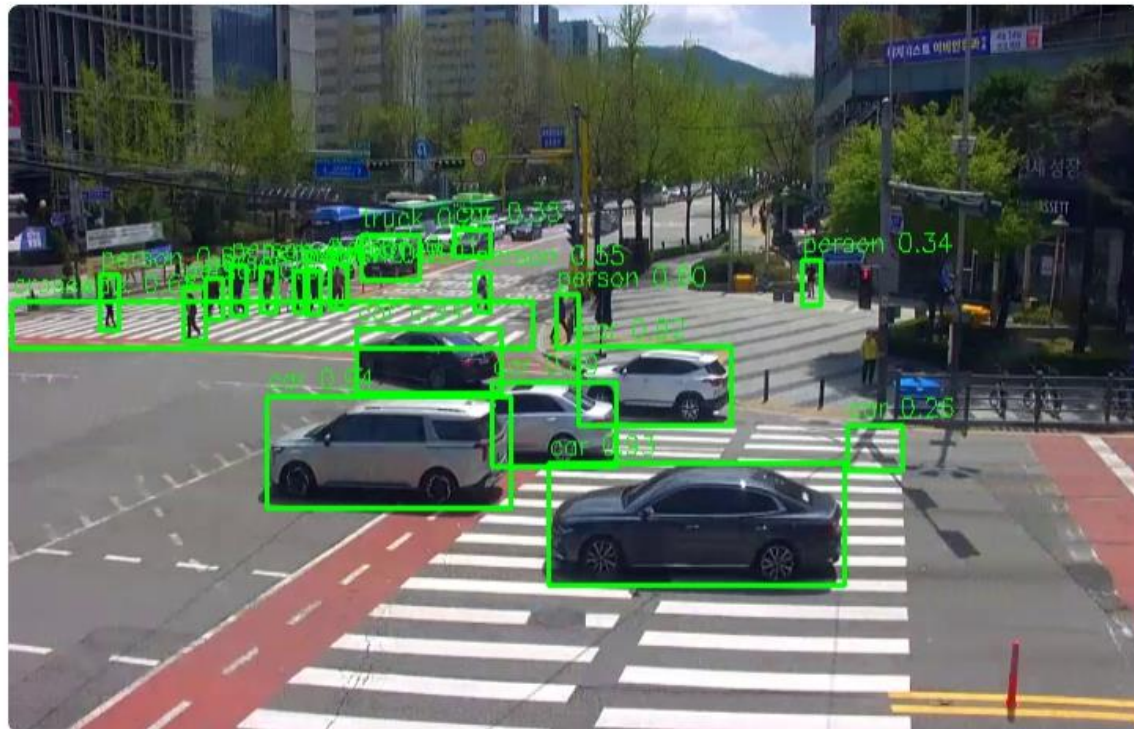


KakaoTalk\_20250415\_123136238.mp4 1.7MB

처리할 최대 프레임 수

1000

100



### 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

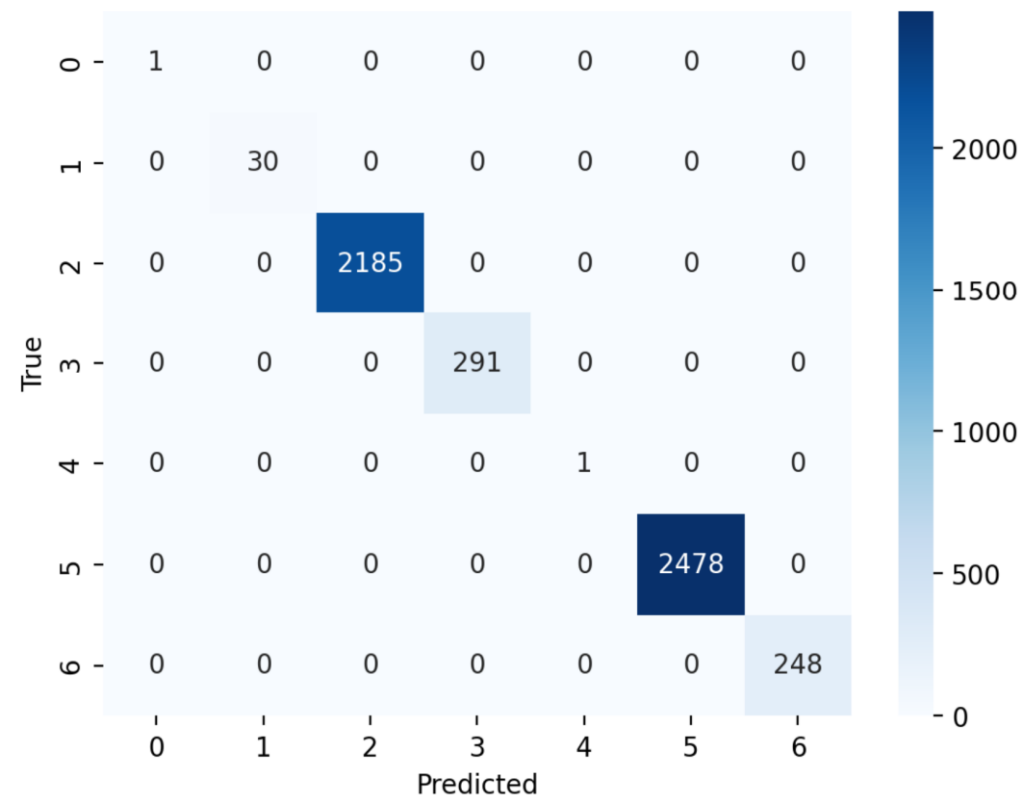
평균 정밀도 (Average Precision): 0.78



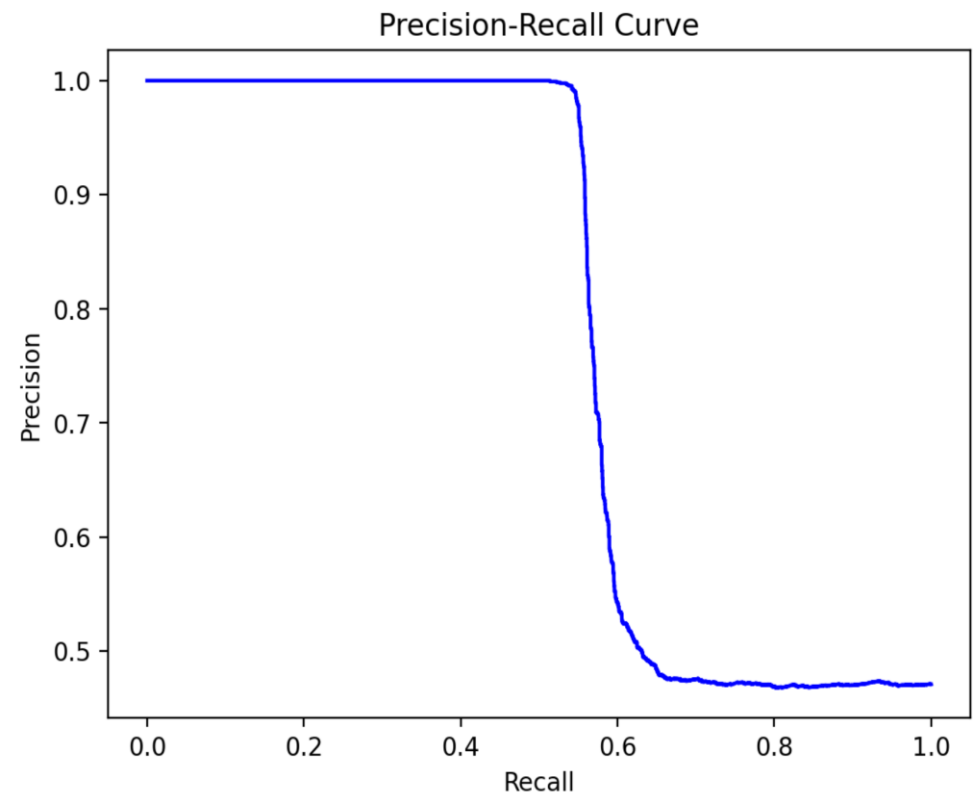
# 03

# 모델 데이터 학습 - dashboard

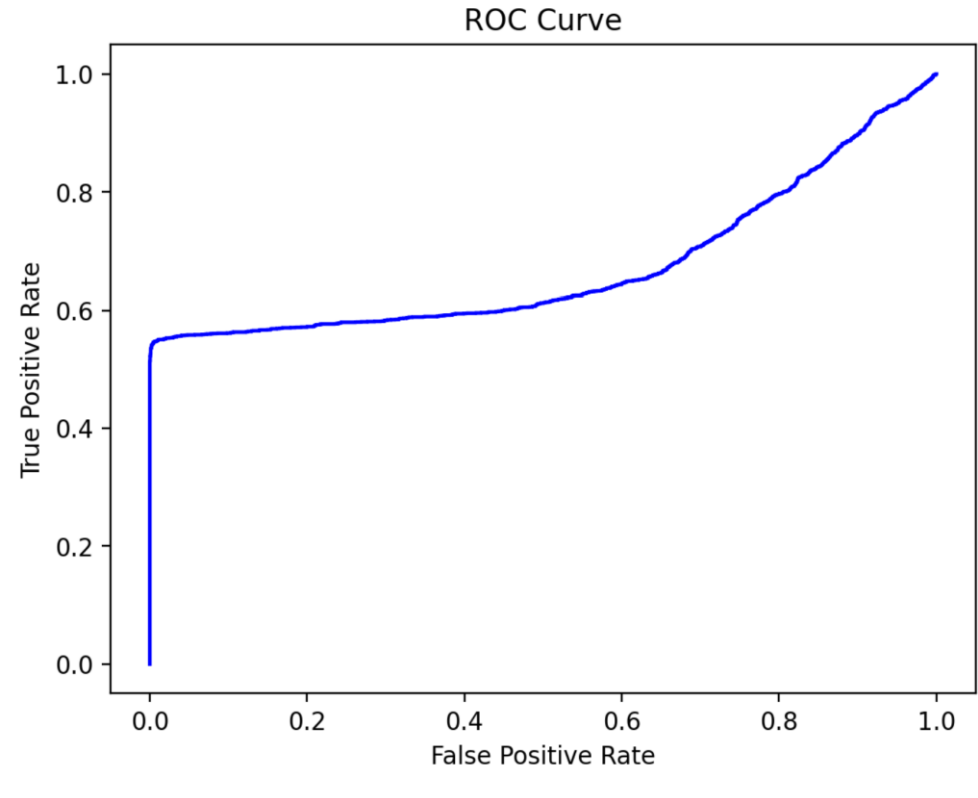
Confusion Matrix



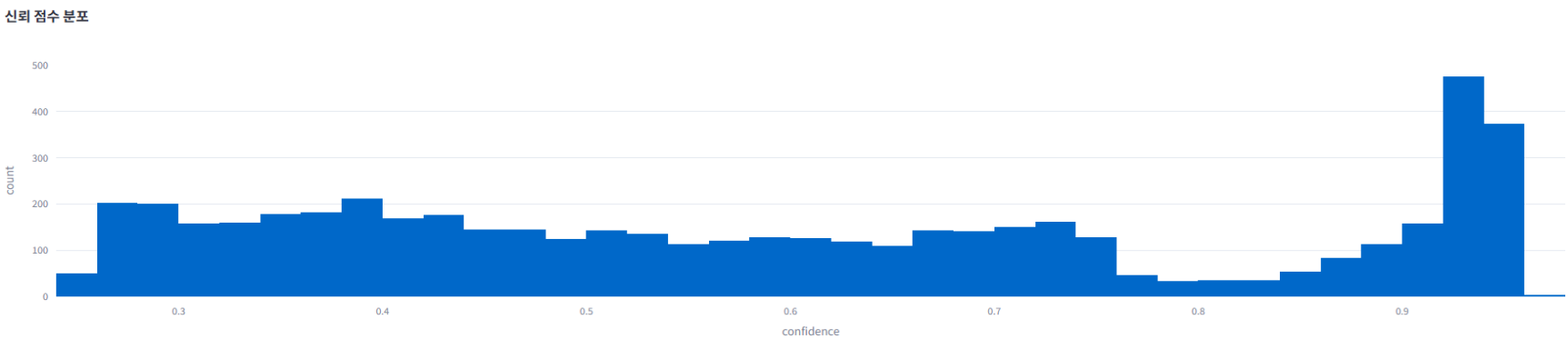
Precision-Recall 곡선



ROC 곡선

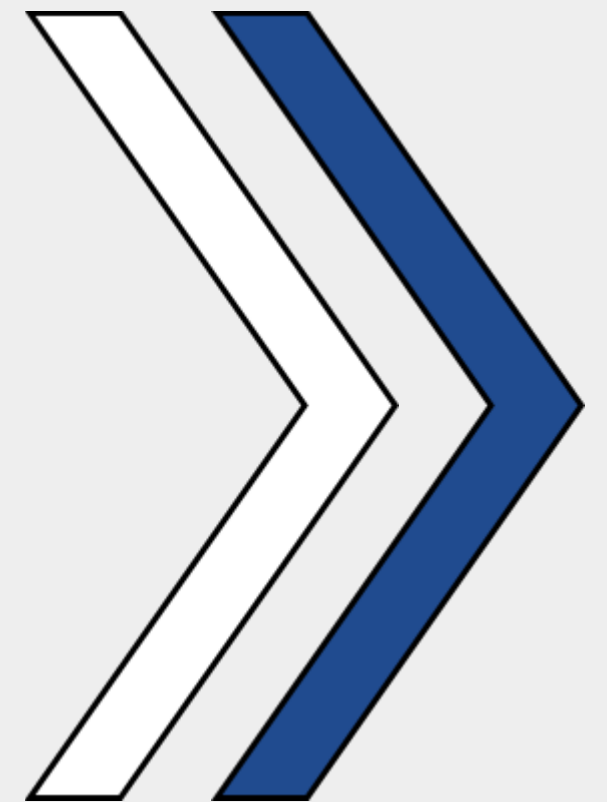


신뢰 점수 분포



영상 처리 완료!

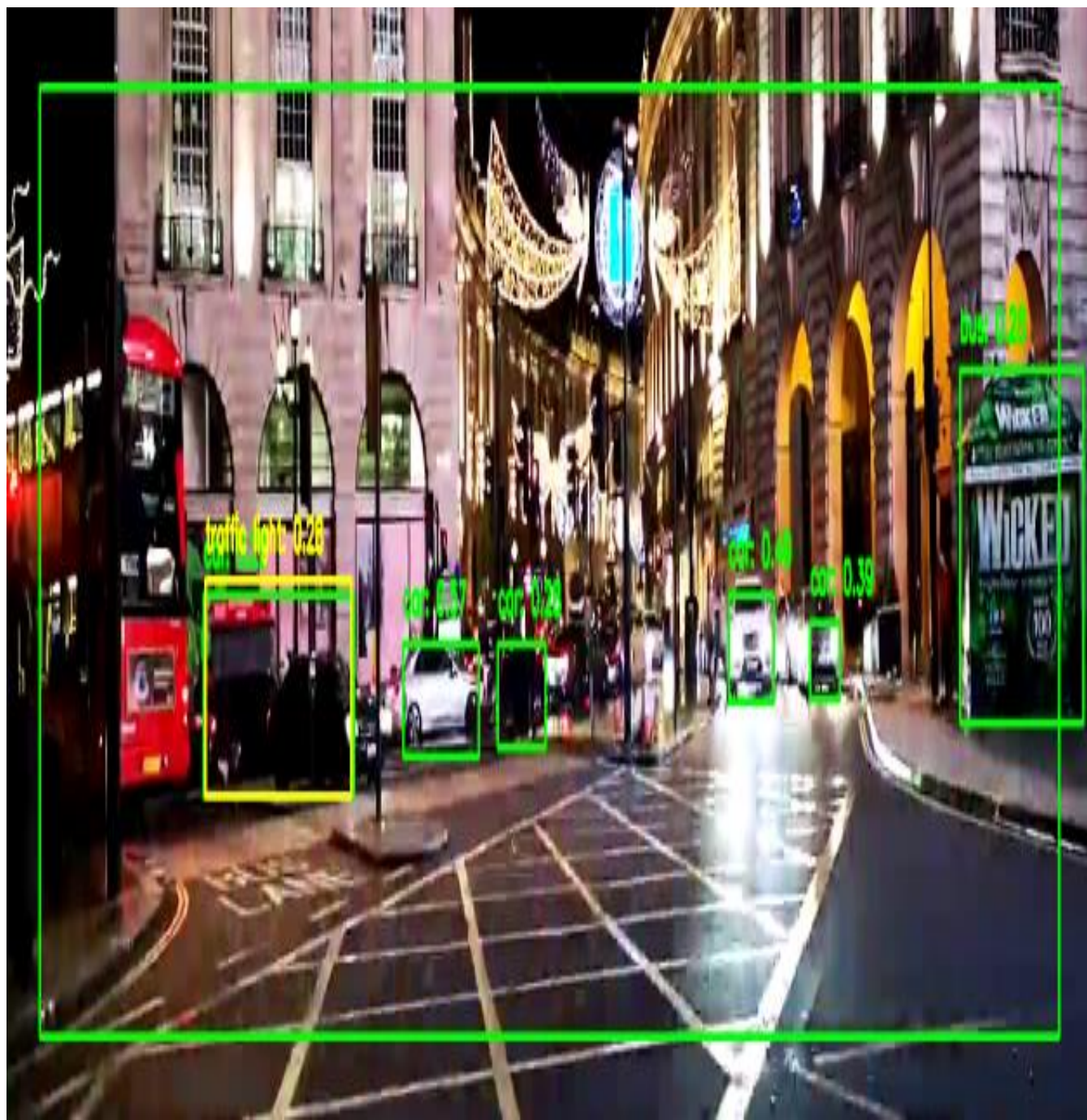
# 시스템 구현 및 작동 시연



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

## 04

## 구현 화면 소개(전)

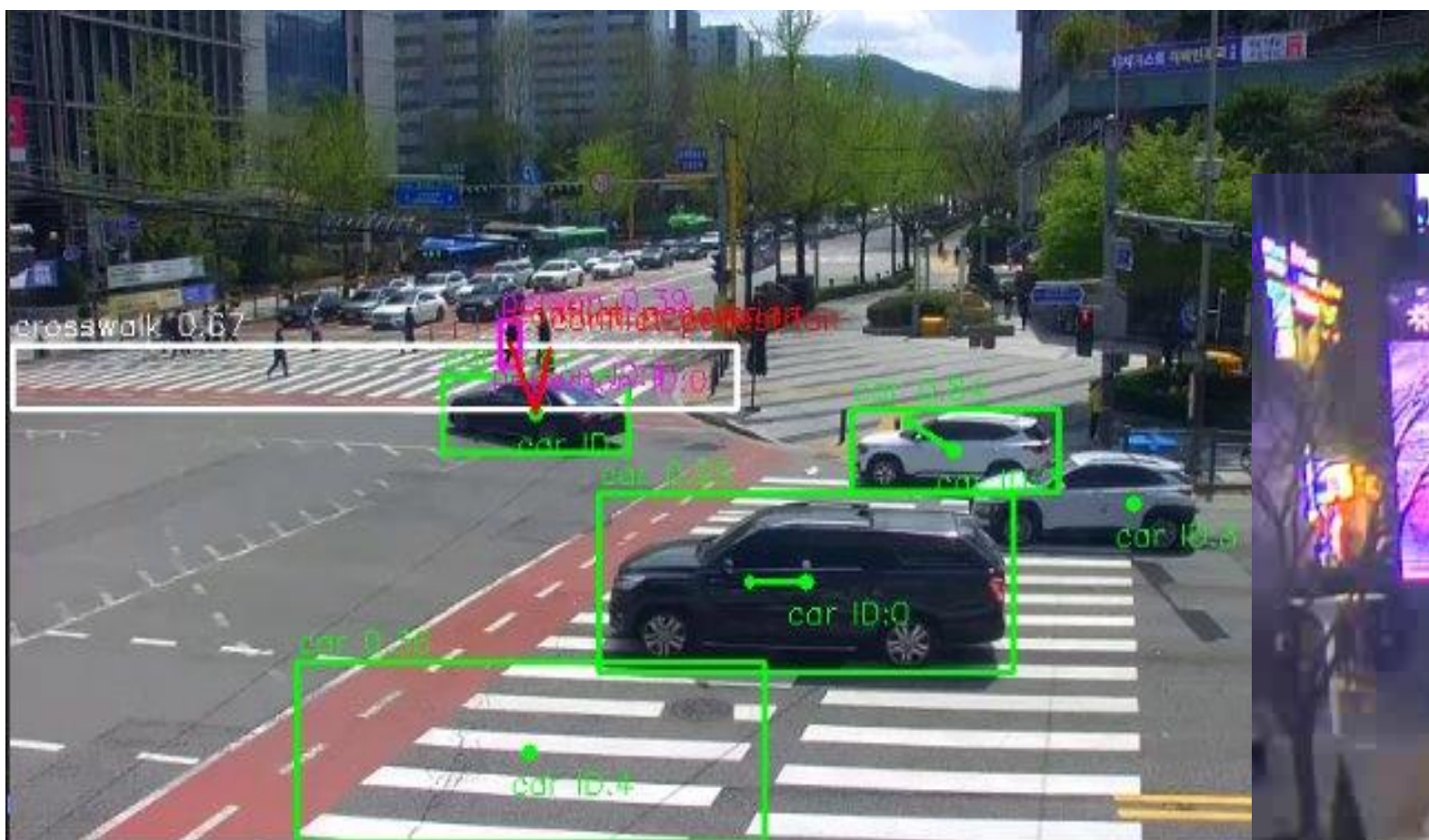


```
def draw_rectangle(event, x, y, flags, param):  
    global drawing, roi_points, roi_rect, current_roi  
  
    if event == cv2.EVENT_LBUTTONDOWN:  
        drawing = True  
        roi_points = [(x, y)]  
  
    elif event == cv2.EVENT_MOUSEMOVE:  
        if drawing:  
            roi_rect = (roi_points[0][0], 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 <= x <= x2 and y1 <= y <= y2) or (x2 <= x <= x1 and y2 <= y <= y1):  
                roi_list.pop(i)  
                break
```



04

## 구현 화면 소개 (후)



## 04

## 구현 화면 소스코드 (후)

```

1  import streamlit as st
2  import cv2
3  import tempfile
4  from roboflow import Roboflow
5  import numpy as np
6  from collections import deque
7
8  st.set_page_config(layout="wide")
9
10 # 제목
11 st.title("🚗 실시간 위반 탐지 대시보드")
12
13 # 영상 업로드
14 video_file = st.file_uploader("📹 영상 업로드 (.mp4)", type=["mp4", "avi"])
15
16 # Roboflow 모델 불러오기
17 rf = Roboflow(api_key="KL1cHdVtvytxtpDiXA0W")
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):
26     cap = cv2.VideoCapture(video_path)
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

```

```

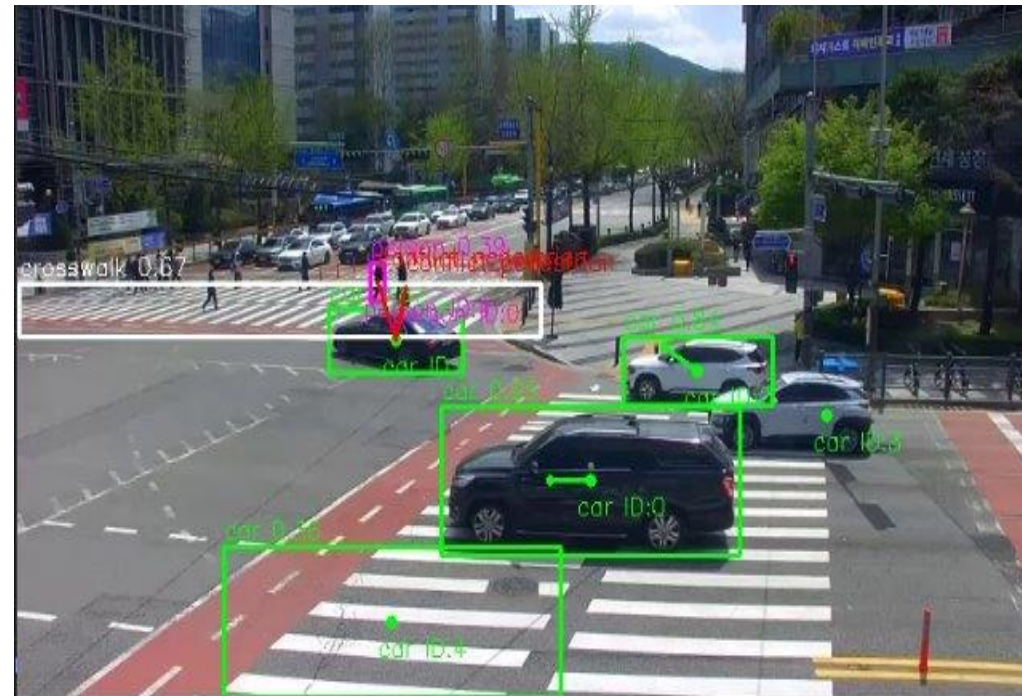
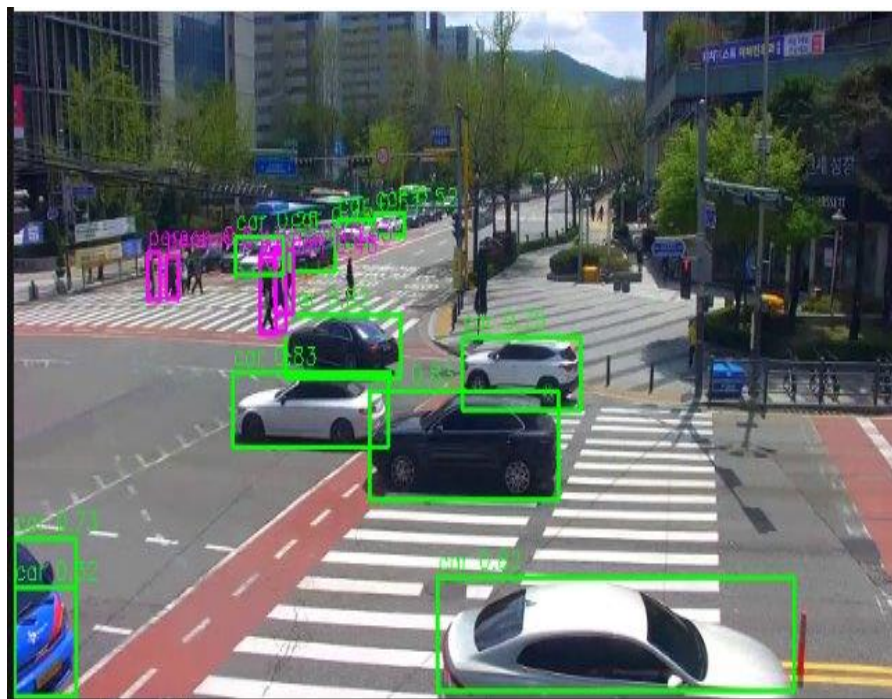
--
36     try:
37         results = model.predict(frame, confidence=CONFIDENCE_THRESHOLD, overlap=IOU_THRESHOLD)
38         predictions = results.json()["predictions"]
39         for pred in predictions:
40             cls = pred["class"].lower()
41             x = int(pred['x'] - pred['width'] / 2)
42             y = int(pred['y'] - pred['height'] / 2)
43             w = int(pred['width'])
44             h = int(pred['height'])
45             conf = pred['confidence']
46
47             if cls in stats:
48                 stats[cls] += 1
49                 color = (0, 255, 0)
50                 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:
54             st.warning(f"예측 중 오류 발생: {e}")
55
56         frame = cv2.resize(frame, (720, 480))
57         frame_placeholder.image(frame, channels="BGR")
58         frame_count += 1
59
60     cap.release()
61     return stats
62
63 # 업로드된 비디오 처리
64 if video_file:
65     tfile = tempfile.NamedTemporaryFile(delete=False)
66     tfile.write(video_file.read())
67     stats = process_video(tfile.name)
68
69 # 결과 통계 표시
70 st.subheader("📊 탐지 통계")
71 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

## 04

# 향후 반영 계획



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