

도로 정체 분석 서비스

CCTV 이미지를 활용한 객체 탐지 서비스



2025.07.11

발표자 : 강주광

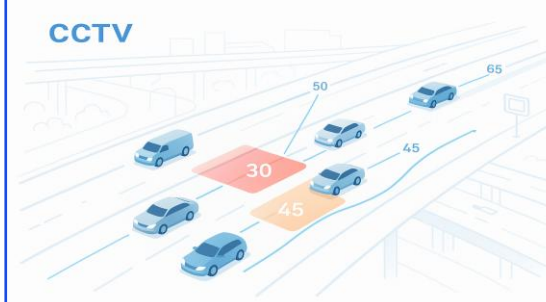
COMENOT

목차

CCTV 이미지를 활용한 객체 탐지 서비스

01 프로젝트 개요 및 기획 배경 02 모델 상세 분석 및 선정

03 데이터 전처리 04 시스템 구현 및 결과



도심 교통 정체 문제 현황 및 분석 필요성

교통 혼잡비용

* 교통혼잡비용(Traffic Congestion Cost)이란?

- 교통수요의 증가에 따른 사회적 비용

° 교통혼잡비용은 차량운행비용과 시간가치비용의 합으로 이루어짐

- 차량운행비용: 고정비(인건비, 감가상각비, 보험료, 제세공과금 등), 변동비(연료소모비, 유지정비비, 엔진오일비 등)

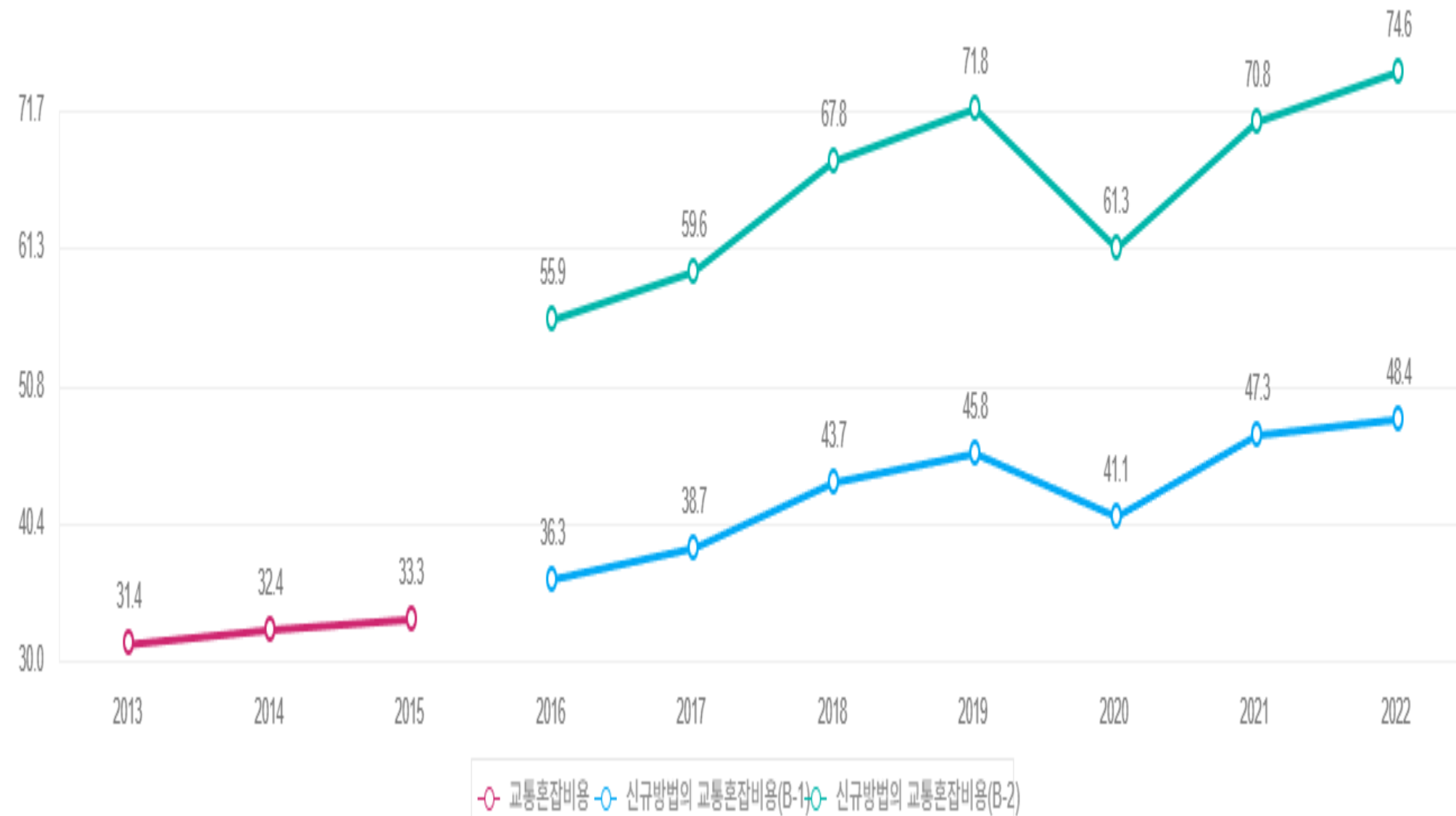
- 시간가치비용: 수단별(승용차, 버스), 목적별(업무, 비업무) 재차인원의 시간가치비용 적용

° 2020년 코로나19로 인한 예외적 하락을 제외하면 매년 꾸준히 상승세

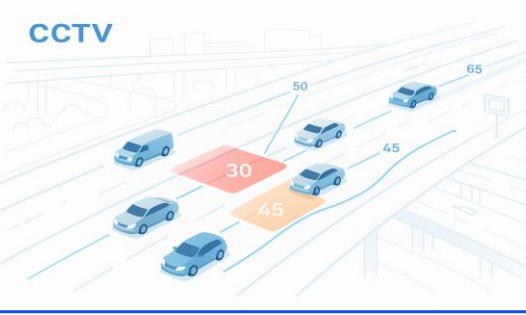
*B-2 : 시군도로를 포함한 전국 교통 혼잡비용

교통혼잡비용

연도별 교통혼잡비용



<지표누리-도로교통 혼잡비용>



도심 교통 정체 문제 현황 및 분석 필요성

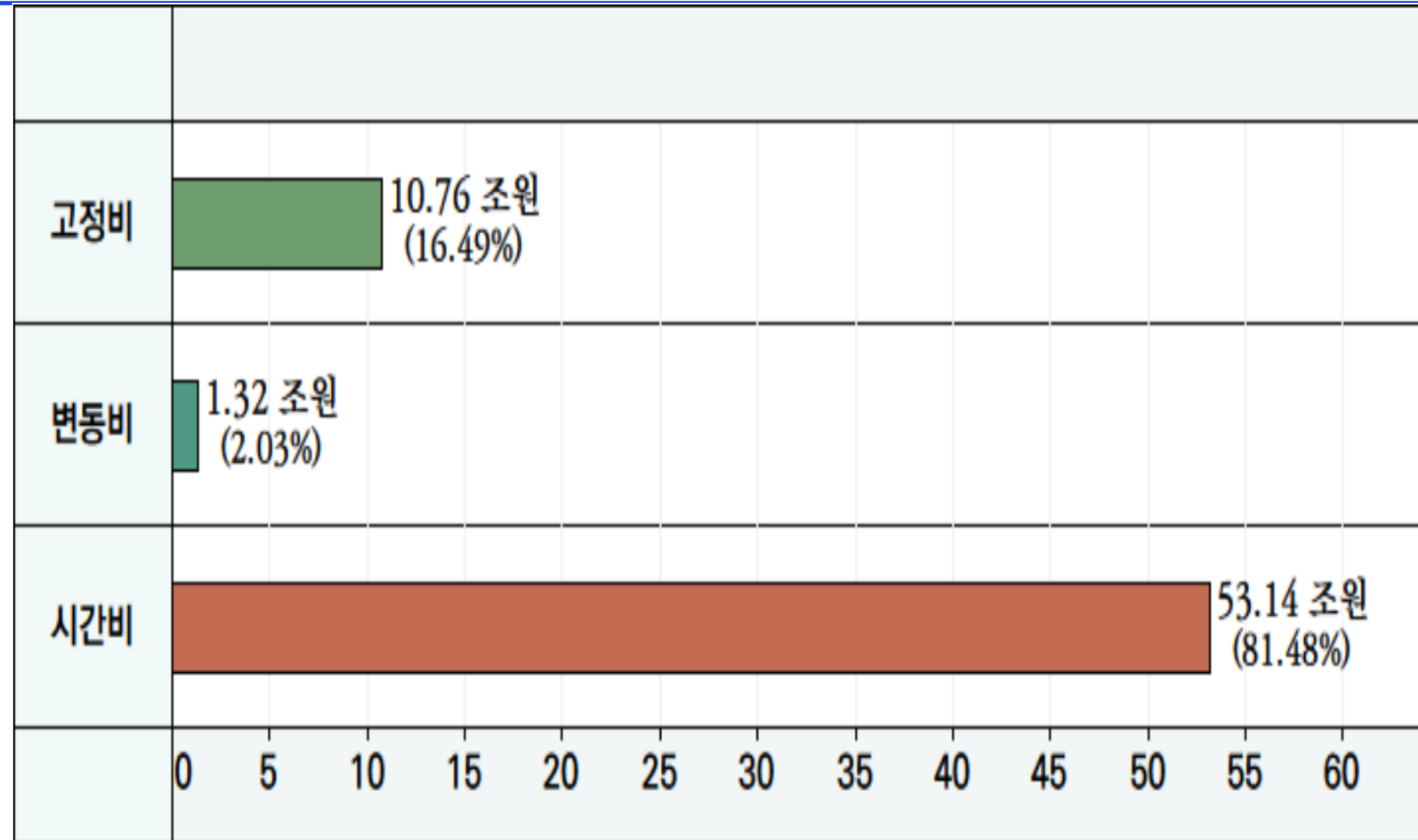
차량 정체 - 교통 혼잡비용

[구성요소별 교통혼잡비용 산정방법]

- * (고정비) 운행시간 차이 × 고정비 원단위 × 교통량
- * (변동비) 연료 소모량 차이 × 연료소모비 단가 원단위 × 교통량 × 링크길이
- * (시간가치비) 운행시간 차이 × 차량 1대당 평균 통행시간가치 원단위 × 교통량

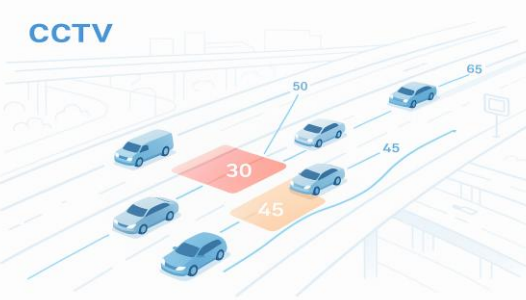
〈한국교통연구원 -국가교통비용지표〉

차량 정체로 발생하는 운행 시간 차이로 계산되는 시간비는 차량의 정체 량과의 직접적인 관계를 가진다.



〈그림 4-3〉 2021년 구성요소별 교통혼잡비용 추정결과

〈한국교통연구원 -제3권 교통혼잡비용(2021)〉



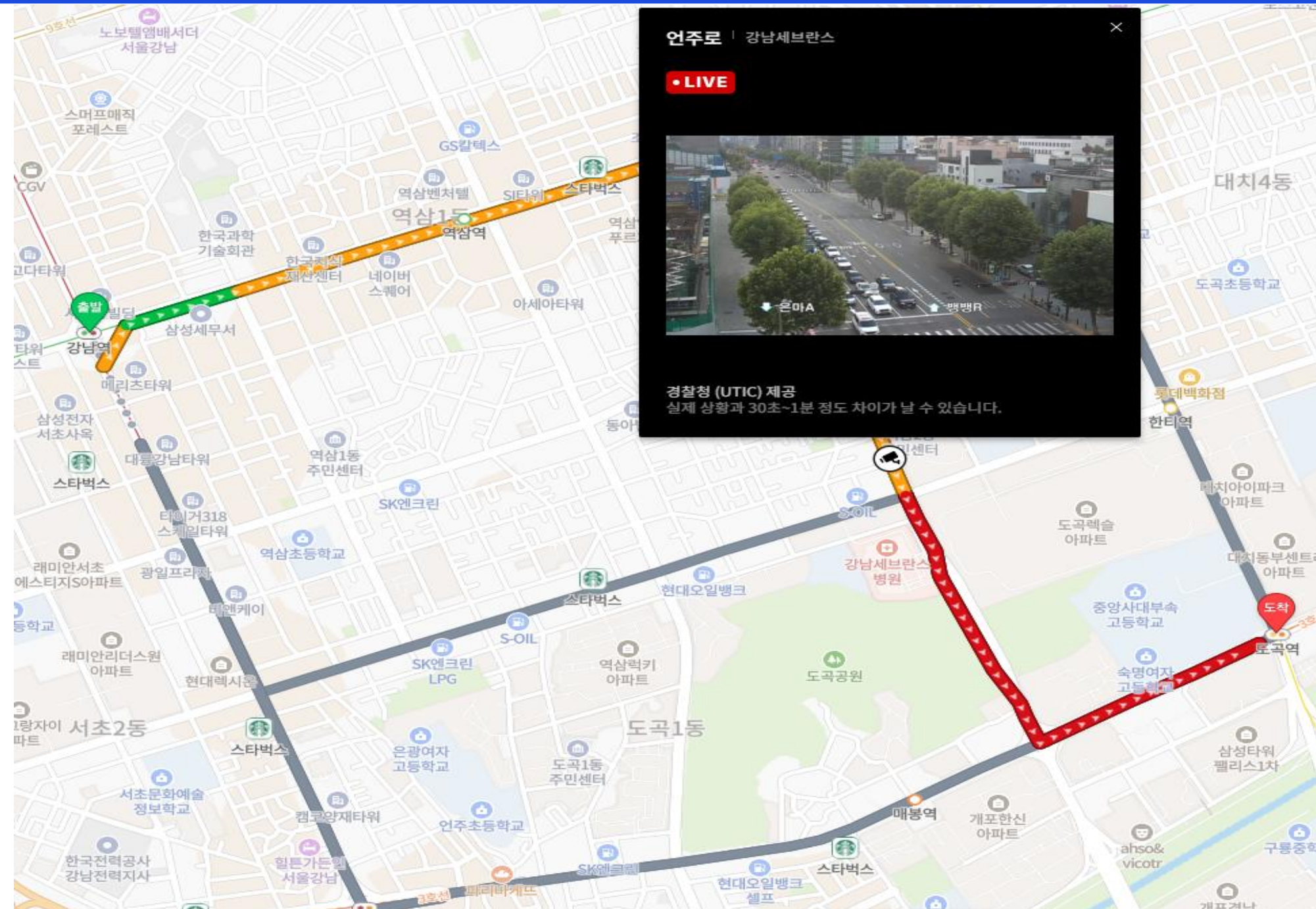
프로젝트 목표 및 기대 효과

목표

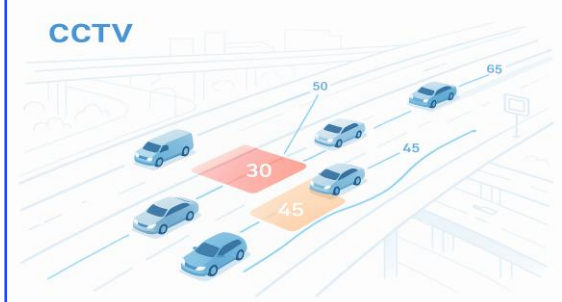
도로 cctv를 분석하여 교통현황 흐름을 파악하고
서비스 이용자에게 도로 교통 상황을 제시하여
여러 도로로 차량을 분산해 정체 예방 및 해소에 기여한다

기대효과

- 혼잡 비용 감소(시간비 감소)
- 예기치 못한 교통 흐름에 빠른 대응(자연재해, 교통사고, 날씨)

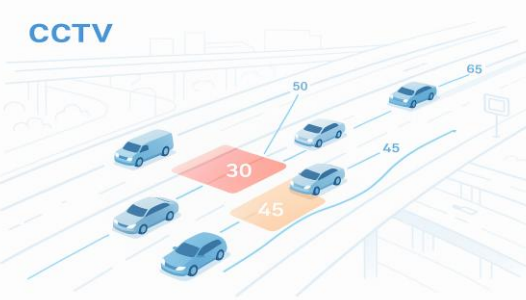


<네이버지도>



데이터 구성 및 선정 기준 (CCTV 이미지 데이터)

데이터 세트	합성 데이터	실제 도로 cctv 데이터
데이터 이미지		
제공	DACON 제공 및 비솔(VISOL)에서 생성 - 자동차 객체 탐지 합성 데이터	뉴욕 주립대 올버니 캠퍼스 구축 및 공개 - UA-DETRAC 데이터셋 논문 "UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking" 소개
데이터 수	사용할 총 이미지 데이터 수 : 6481개 Train : 3888개 Val : 1296개 Test: 1297개	사용할 총 이미지 데이터 수 : 138252개 Train: 82951개 Val: 27650개 Test: 27651개
이미지 차이점	- 노이즈가 없이 깔끔한 이미지 - 제한된 차량 수(1~3) - 객체에 대한 기초적인 정보 제공 - 초기 안정적인 학습 가능	- 실제 환경의 노이즈 유지 - 1~X 많은 차량수 - 실제 환경의 다양한 촬영 각도 반영 - 후반 실제 환경에 대한 일반화



딥러닝 개발 과정 요약



Step 1

모델
선정

- 적용 가능한 모델들을 비교 분석
- 최적의 모델 선정(최종 선정: YOLO11s)

Step 2

데이터
전처리

- 요구 데이터 수집(합성 데이터: 6,481개, 실제 데이터: 138,252개, 총: 144,733개)
- 모델 요구형태 수정(5점 좌표)& 사용 불가 데이터 삭제(객체가 없는 이미지)

Step 3

모델
학습

- 모델 학습 (epochs=200, patience=10, imgsz=640, batch=-1, workers=12)
- 1차 학습 mAP50 [0.995], mAP50-95 [0.99498]
- 2차 학습 mAP50 [0.99389], mAP50-95 [0.92949]

Step 4

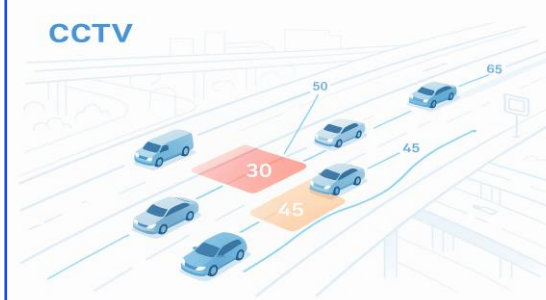
모델
검증

- Test데이터를 활용한 검증(conf : 0.7)

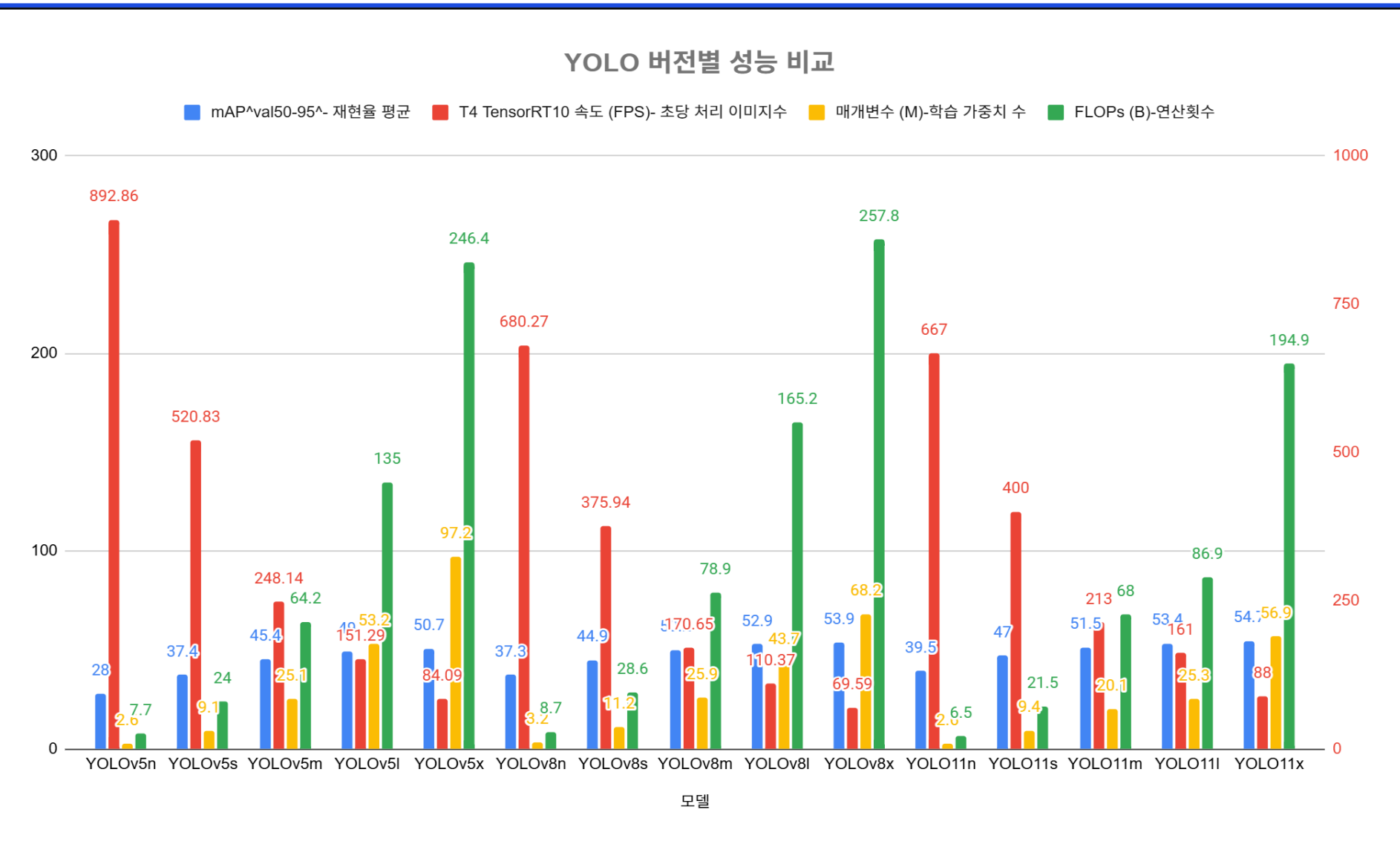
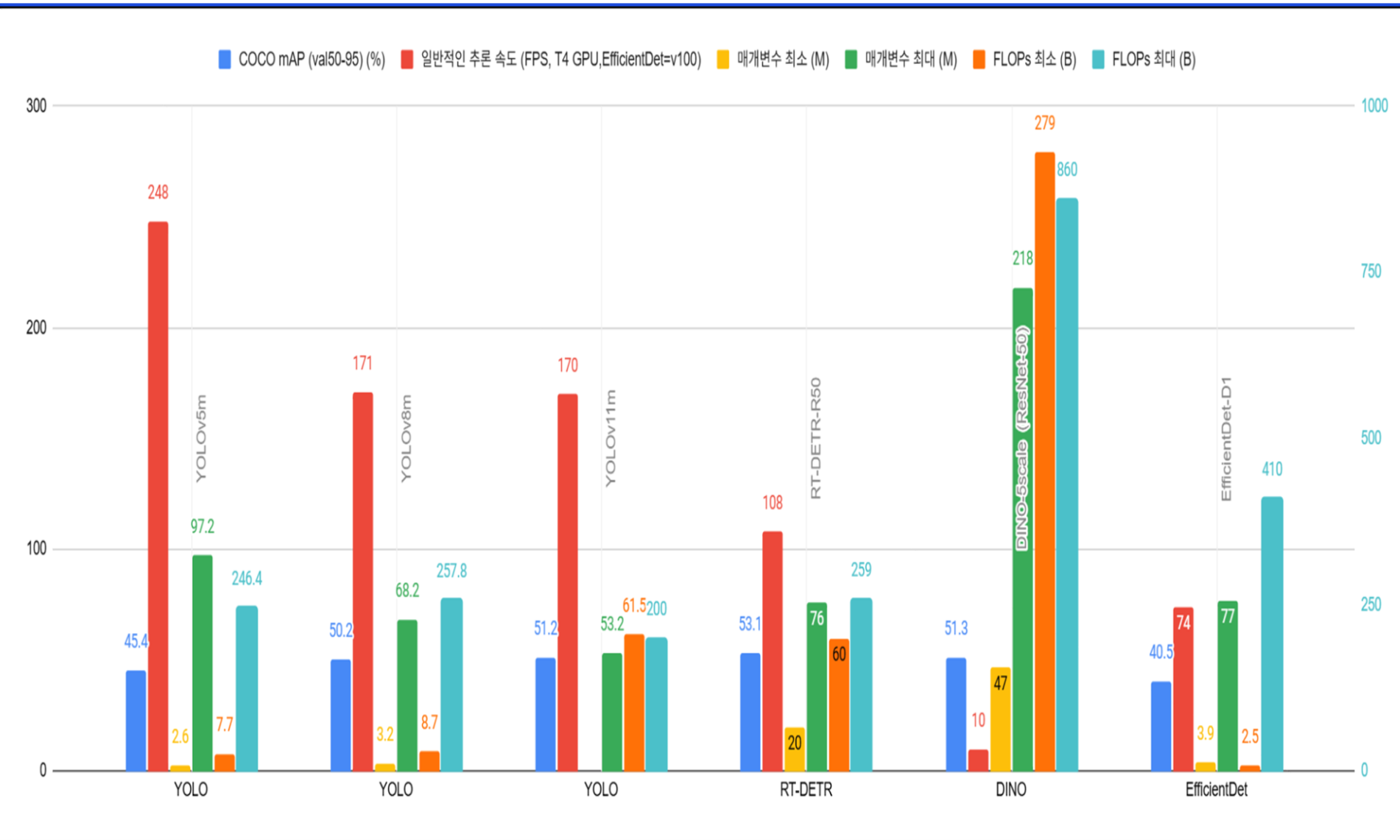
Step 5

모델
적용

- 학습 모델을 적용한 API구현
- Streamlit을 사용한 사이트 제공



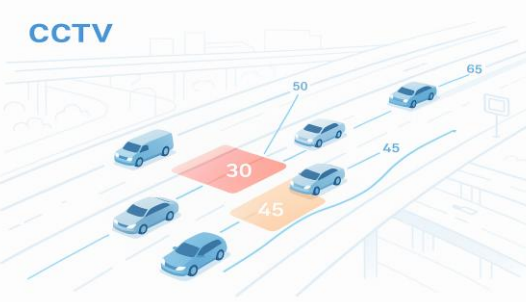
모델 상세 분석 및 선정



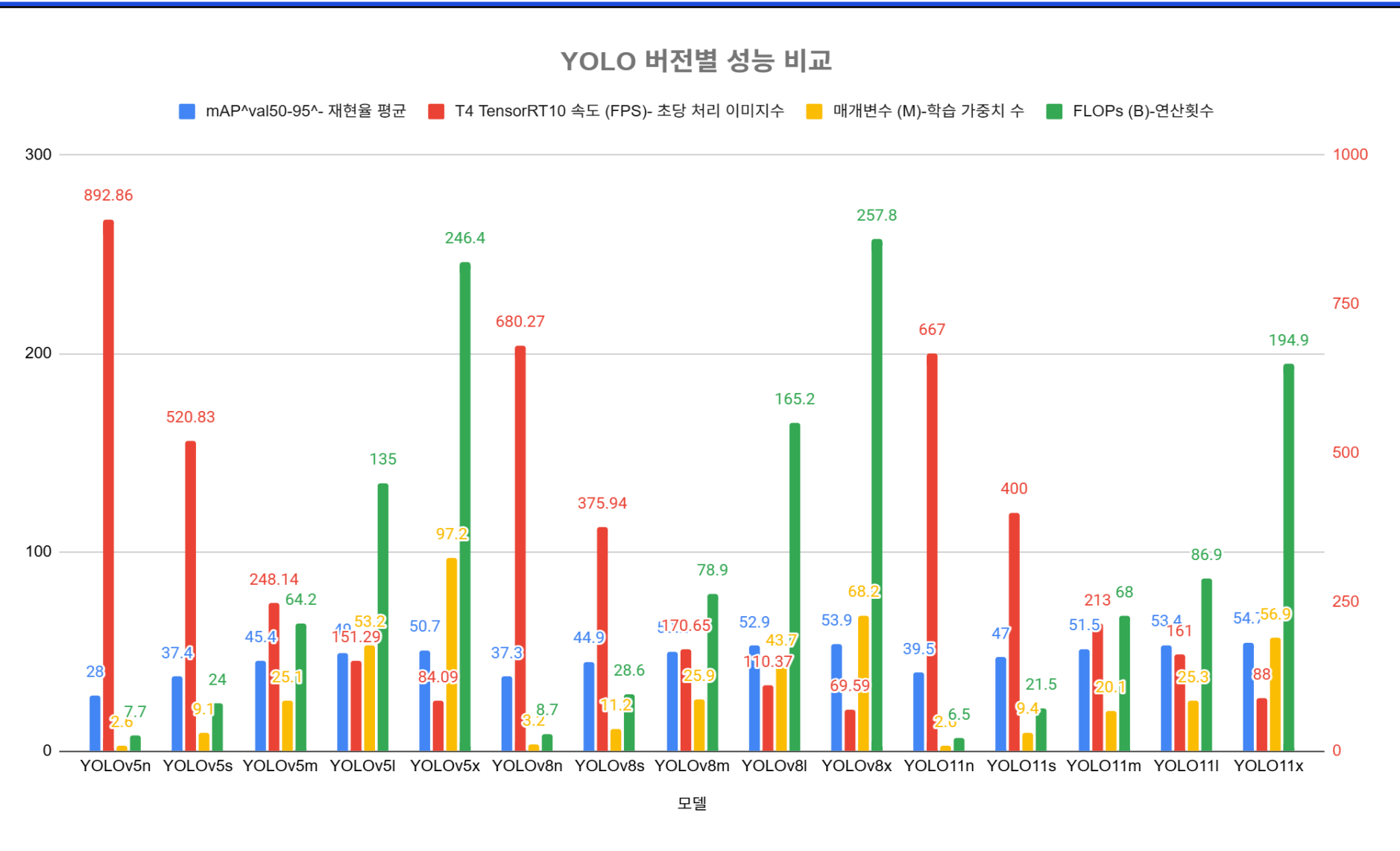
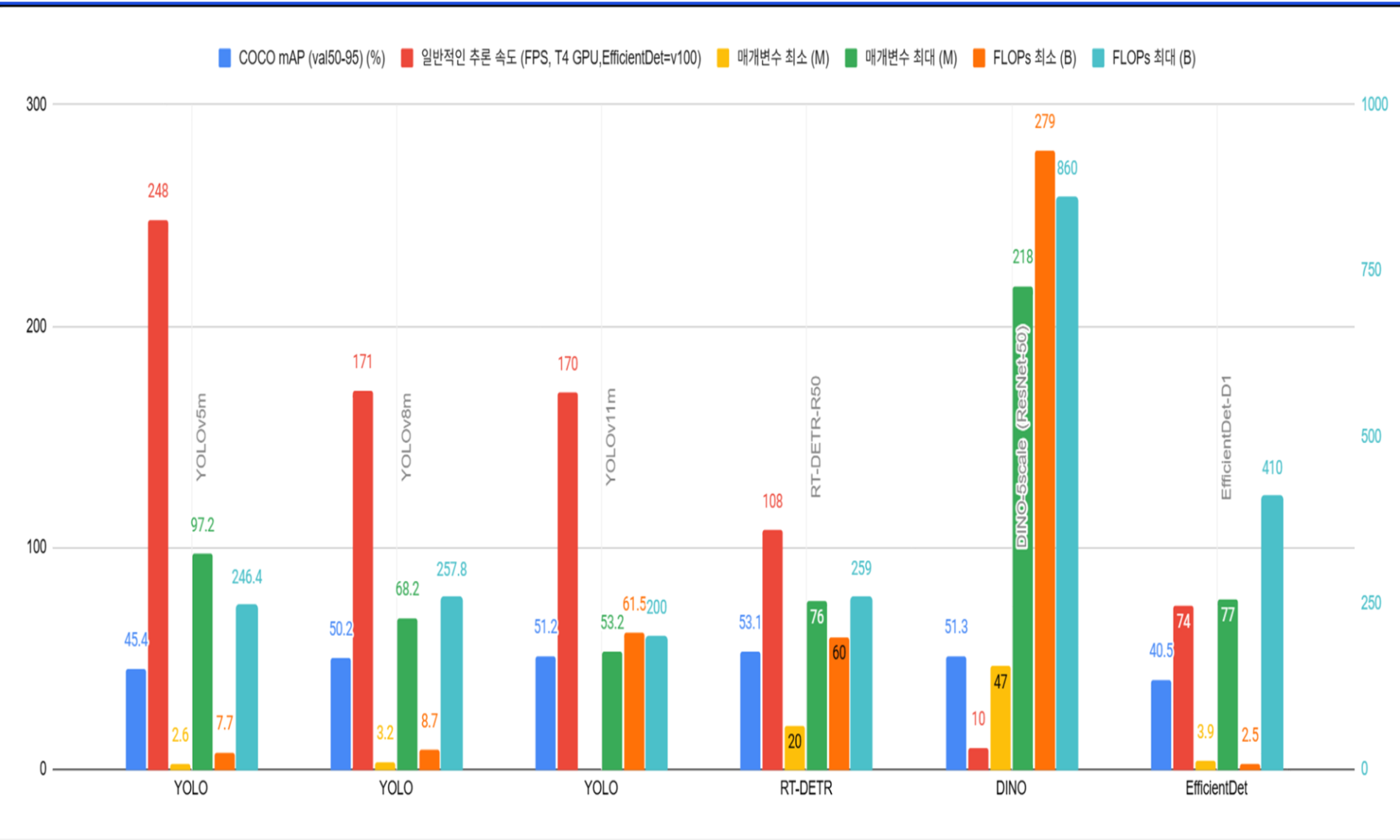
■ YOLO5, YOLO8, YOLO11, RT-DETR, DINO, EfficientDet 모델들을 비교분석한 결과 정확도와 속도 모두 높은 YOLO 모델 선정

■ YOLO의 세부 버전 v5, v8, v11 각각의 n, s, m, l, x 성능 비교 결과

■ <https://www.ultralytics.com/yolo>
License : AGPL-3.0 License



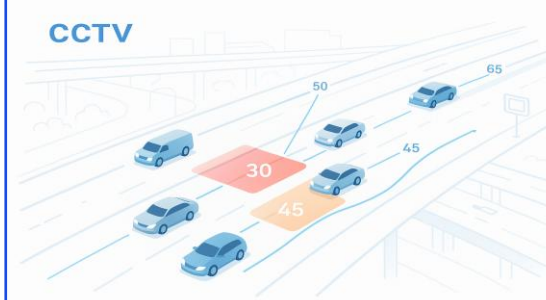
모델 상세 분석 및 선정



- 모델간 벤치마킹 결과 YOLO11s는 초당 400장의 이미지 처리가 가능
- 15개의 모델 중에서 처리 속도 4순위, mAP50-95 성적 9순위인 YOLOv11s 선정
- [베이스 라인모델 기준으로 속도와 정확도의 치우침 없는 모델을 선정]

- 추후 복수의 cctv를 일괄 처리하기 위해서는 초당 처리량이 많은 모델 yolo5n, yolo5s, yolo8n, yolo11n을 사용하는 걸 고려 가능

- <https://www.ultralytics.com/yolo>
License : AGPL-3.0 License



합성 데이터 셋 & 실제 도로 cctv

Yolo 모델의 5점 형식



합성 데이터 셋
- VISOL(DACON)

실제 도로 cctv
데이터 셋
- UA-DETRAC

* Test data(image), train data(image, label), classes.txt

1. Yolo형식(images[train, val, test], labels[train, val, test], yolo.yaml)과 데이터 구조 차이(수정)
2. Label데이터가 9점 형식(yolo는 5점 형식)차이 (수정)
3. Yolo. Yaml 의 class_id 는 차량의 종류 구별 없이 진행 (수정)

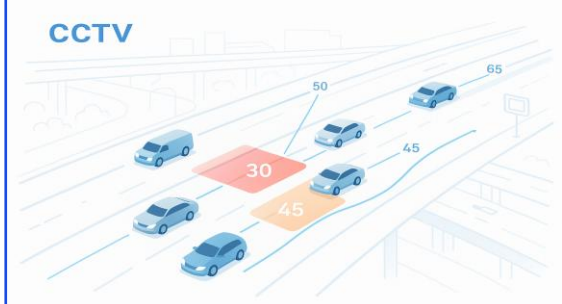
* Labels[train, val], images[train, val]

1. Yolo형식(images[train, val, test], labels[train, val, test], yolo.yaml)과 데이터 구조 차이(수정)
2. Yolo. Yaml 의 class_id 는 차량의 종류 구별 없이 진행 (생성)
3. Test데이터 분할

zidane.txt

0	0.481719	0.634028	0.690625	0.713278
0	0.741094	0.524306	0.314750	0.933389
27	0.364844	0.795833	0.078125	0.400000

class_index x_center y_center width height



모델 검증

모델 학습 방법

테스트 이미지 1



테스트 이미지 2



1차 학습 : 합성 데이터-사전학습



2차 학습 : 실제 도로-미세조정

차량 탐지 개수
(conf=0.01)테스트 1 : 0개
테스트 2 : 0개테스트 1 : 10개
테스트 2 : 4개탐지된 평균 conf
(conf=0.01)테스트 1 : 0
테스트 2 : 0테스트 1 : 0.03651
테스트 2 : 0.79675

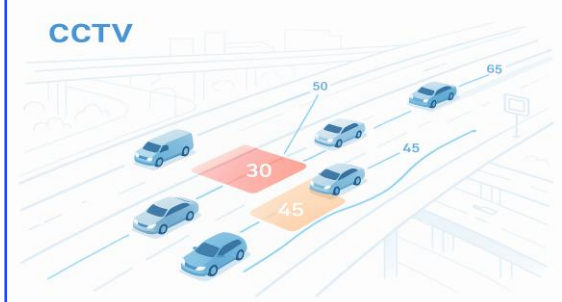
모델 mAP

mAP50 : 0.995
mAP50-95 : 0.99498mAP50 : 0.99389
mAP50-95 : 0.92949

문제점

높은 mAP를 가지지만 실제 도로에서는 탐지 실패가 발생함
=도메인 격차 발생(domain gap)

도메인 격차가 완화 되었지만 mAP를 높이는 추가 학습이 필요함



도메인 격차(Domain Gap)

도메인 격차란?

모델의 학습 데이터와 실제 적용되는 대상의 데이터 간의 차이로 인해 성능 저하가 발생하는 현상

발생 원인

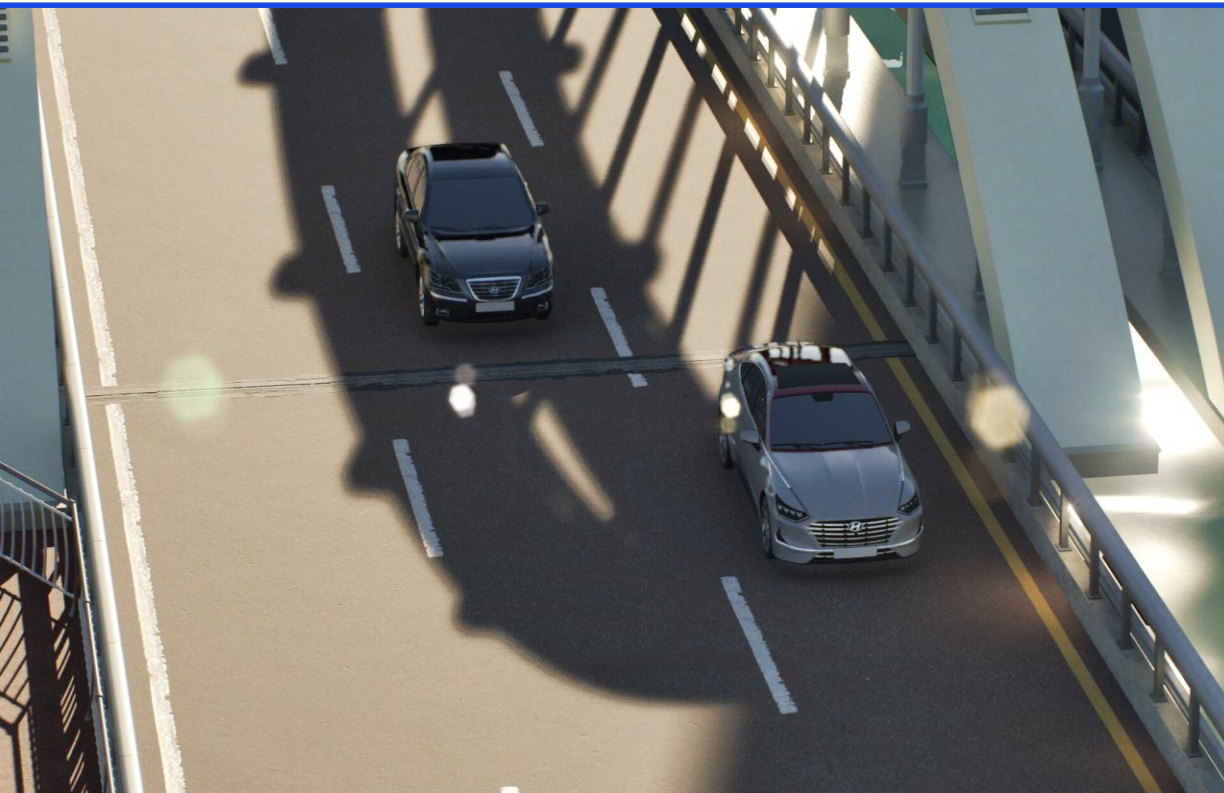
합성데이터와 실제 적용될 데이터 간의 카메라 각도, 화질, 조명, 객체 구성, 배경 노이즈 등 이미지의 여러 특성이 다른 문제로 인해 발생

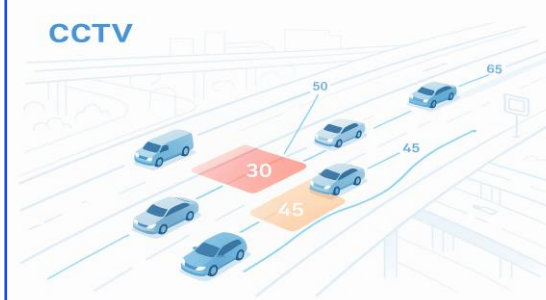
해소 방안

합성 데이터로 학습한 모델에 실제 적용될 데이터와 유사한 추가 데이터로 미세조정(Fine-tuning)학습하는 과정을 통해 도메인 격차 완화

1차 합성 데이터 -> 2차 실제 데이터 학습의 장점

사전 학습된 합성 데이터 기반 모델은 객체의 형태 및 특징에 대한 일반화된 표현을 갖고 있어, 실제 데이터에 대한 미세조정(Fine-tuning) 과정에서 적은 데이터와 적은 시간으로도 우수한 성능 향상이 가능





FastAPI 구현

FastAPI 0.1.0 OAS 3.1
/openapi.json

default

POST /analysis Cctv Analysis Api

• cctv이미지를 통해 도로의 차량 정체를 분석하는 api input file(uploadfile) : 분석할 cctv이미지(원수) output : 이미지에서 차량의 숫자

Parameters

No parameters

Request body required multipart/form-data

file * required string(binary)

max_car integer ☐ Send empty value

streamlit_json boolean ☒ Send empty value

Execute **Clear**

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8000/analysis' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@test.png;type=application/png' \
  -F 'max_car=10' \
  -F 'streamlit_json=false'
```

Request URL

http://localhost:8000/analysis

Server response

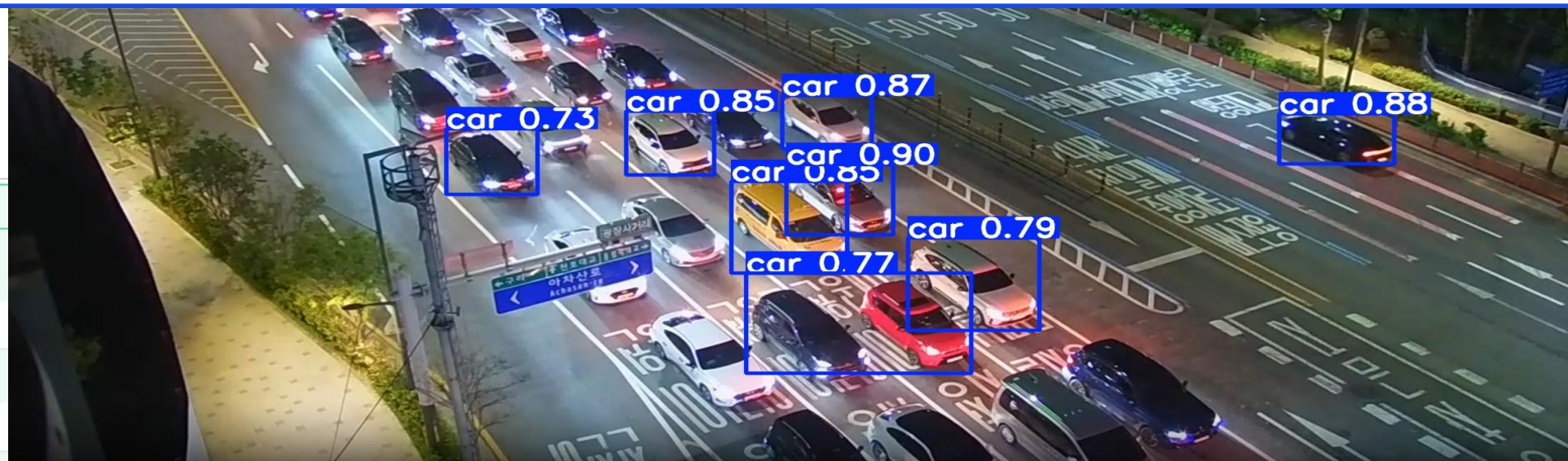
Code **Details**

200

Response body

"정체(0.8)"

Download



Response body

"정체(0.8)"

max_car
integer

10

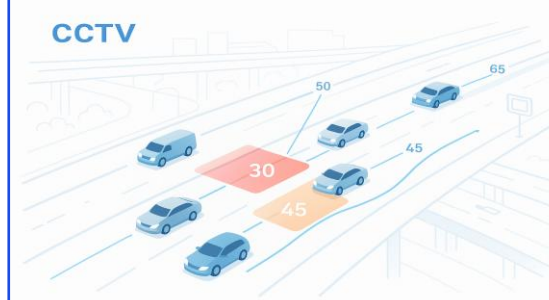
$X = (\text{찾은 차량수} / \text{최대 차량수})$

$X \leq 0.3$: 원할

$0.3 < X \leq 0.7$: 서행

$0.7 < X$: 정체

- Input: 분석 대상 이미지, 이미지에서 발생 가능한 최대 차량 수
- Output (String): 차량의 정체도(분석된 수치)
- conf = 0.7
- 문제점 : Cctv 위치당 촬영되는 차선의 수, 도로의 폭, 도로의 길이가 일관되지 못한다는 문제점
- 한계점 1 : 도로 정보를 정량화 못한다는 문제점으로 인해 최대 차량 수를 사용자에게 입력 받아야 하는 한계
- 한계점 2 : 객체의 면적으로 정체도를 파악하는 방법은 이미지의 3D -> 2D로 사영과 카메라의 각도, 굴절로 인해 일관된 분석이 안되는 한계



Streamlit 구현(test1)

차량 혼잡도 분석

cctv 이미지를 분석하여 차량 탐지 및 정체도를 판단합니다.

점유율 기준 최대 차량수

10

- +

여기에 CCTV 이미지 파일(PNG)을 업로드해주세요...

Drag and drop file here

Limit 200MB per file • PNG

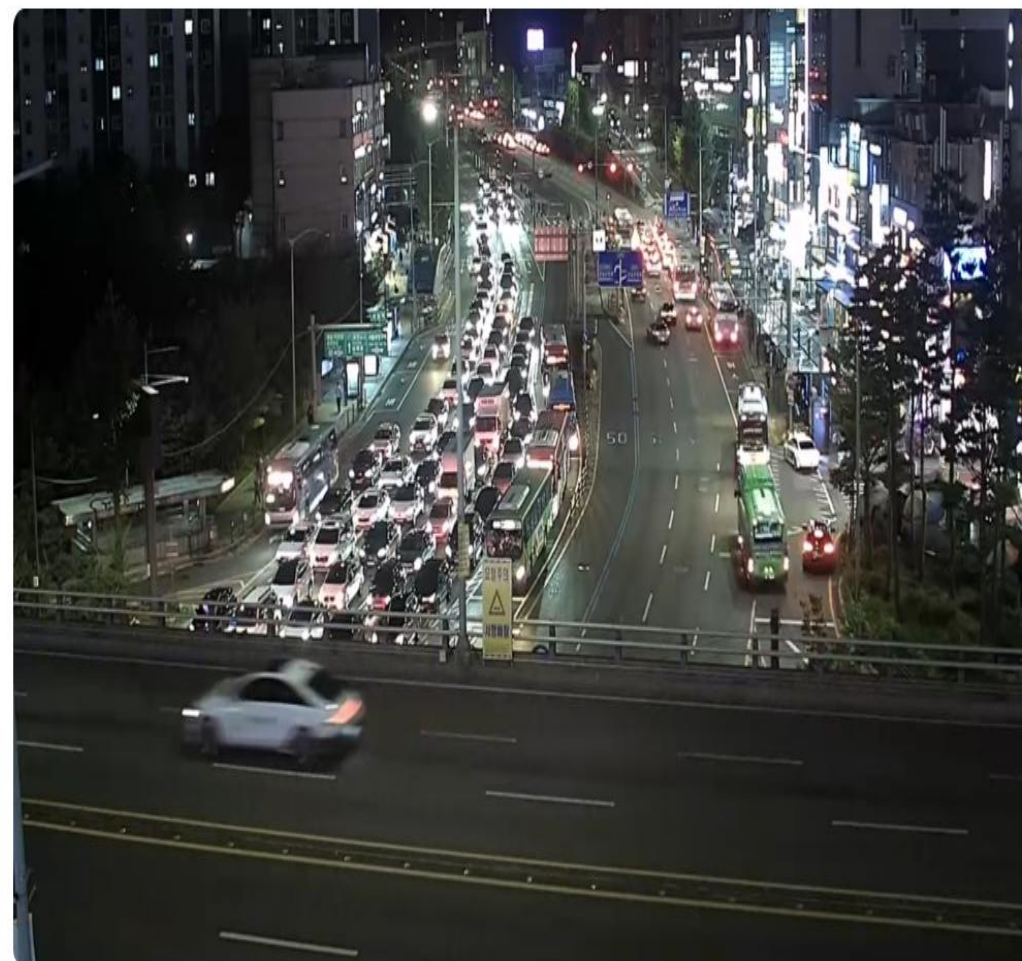
Browse files



test1.png

2.5MB

X



업로드된 이미지

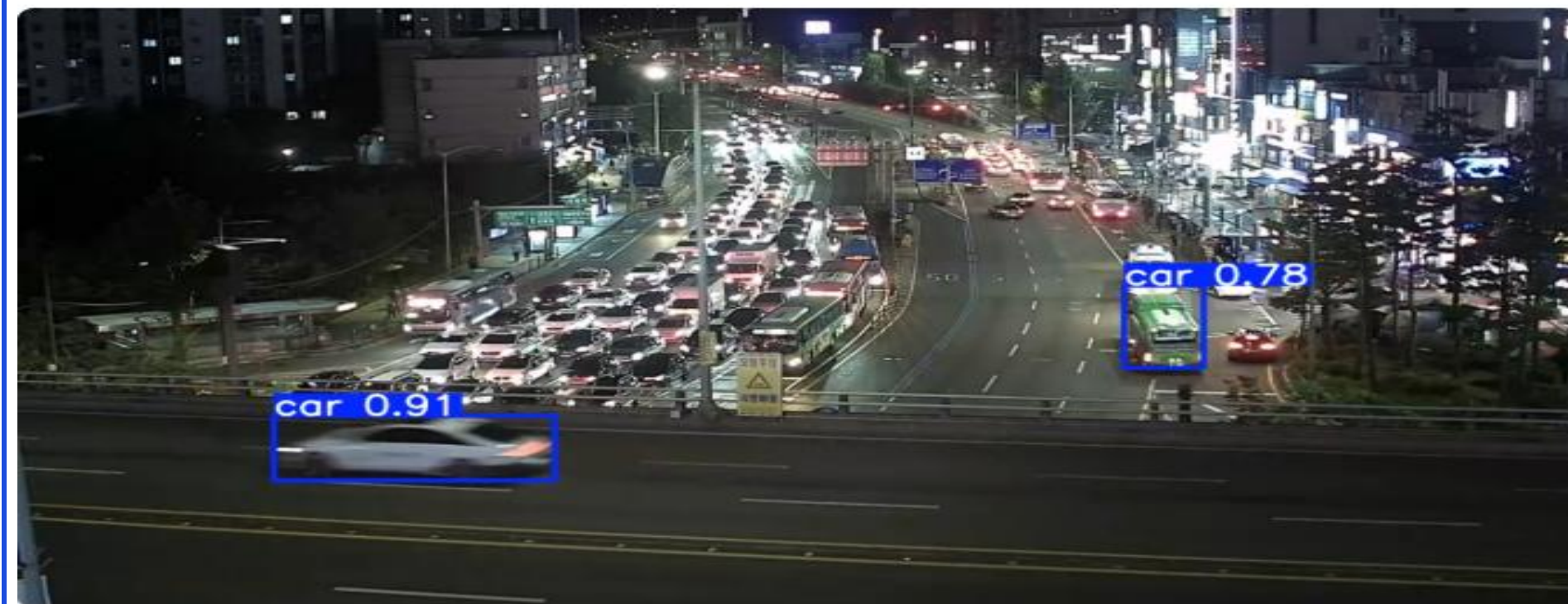
이미지를 분석 중입니다. 잠시만 기다려 주세요...

분석 완료!

Deploy

분석 결과:

파일명: test1.png



분석한 이미지

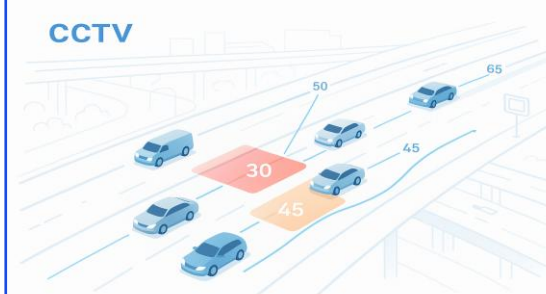
탐지된 차량 수: 2

탐지된 점유율: 0.2

탐지 결과: 원활(0.2)

탐지된 차량 목록:

	class_name	confidence	bbox
0	car	0.9092	315.3066101074219 732.4378051757812 661.287109375 848.0117797851
1	car	0.7752	1361.52197265625 498.2326354980469 1459.5751953125 646.760986328



Streamlit 구현(test2)

차량 혼잡도 분석

cctv 이미지를 분석하여 차량 탐지 및 정체도를 판단합니다.

점유율 기준 최대 차량수

10

여기에 CCTV 이미지 파일(PNG)을 업로드해주세요...

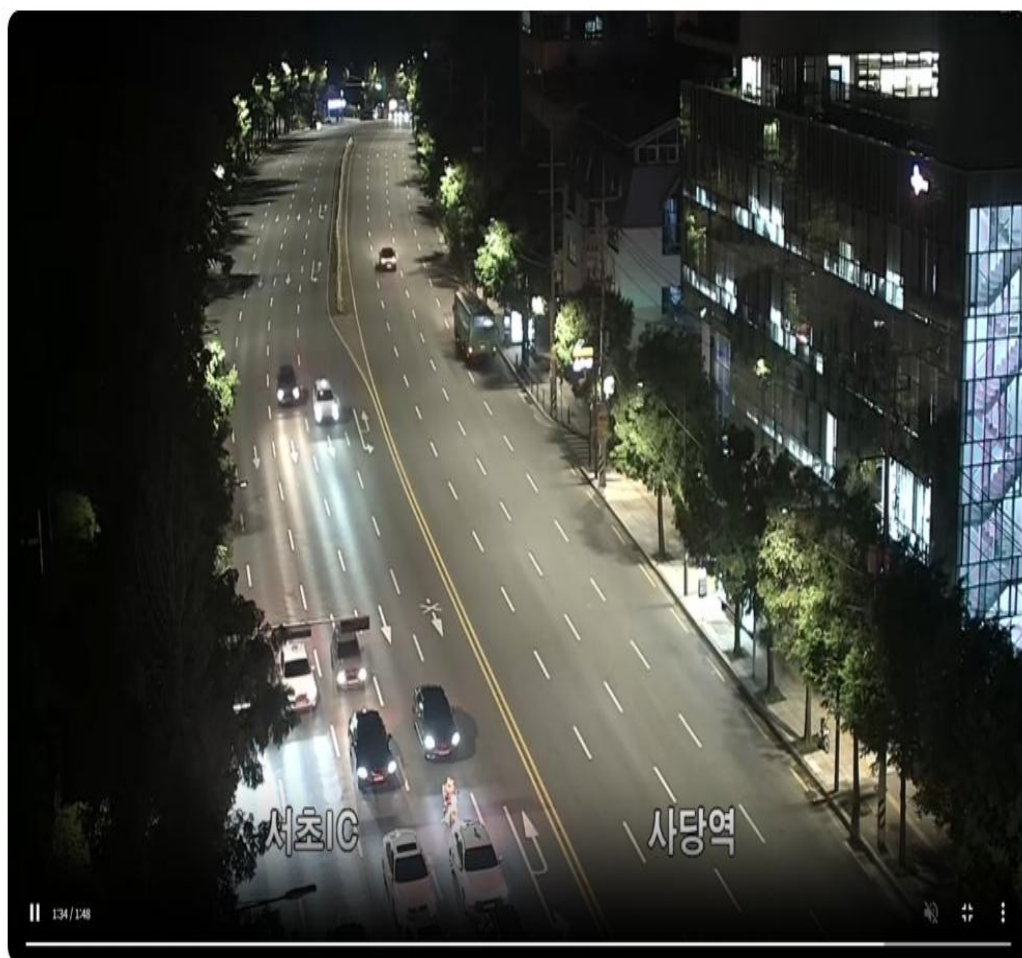
Drag and drop file here

Limit 200MB per file • PNG

Browse files



test2.png
2.1MB



업로드된 이미지

이미지를 분석 중입니다. 잠시만 기다려 주세요...

분석 완료!

Deploy

분석 결과:

파일명: test2.png



분석한 이미지

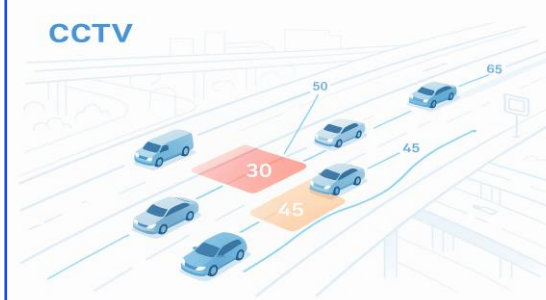
탐지된 차량 수: 3

탐지된 점유율: 0.3

탐지 결과: 원활(0.3)

탐지된 차량 목록:

	class_name	confidence	bbox
0	car	0.9023	760.567138671875 758.6365356445312 851.1087646484375 847.6516113
1	car	0.8558	609.052734375 690.3848876953125 678.0785522460938 767.8304443359
2	car	0.7376	638.27001953125 786.774658203125 734.4793701171875 881.814086914



한계

1

차량 객체 인식률 저하
(conf = 0.7 기준)

- 일부 차량을 인식 자체를 못하는 문제
- 탐지된 차량의 conf가 낮은 문제

2

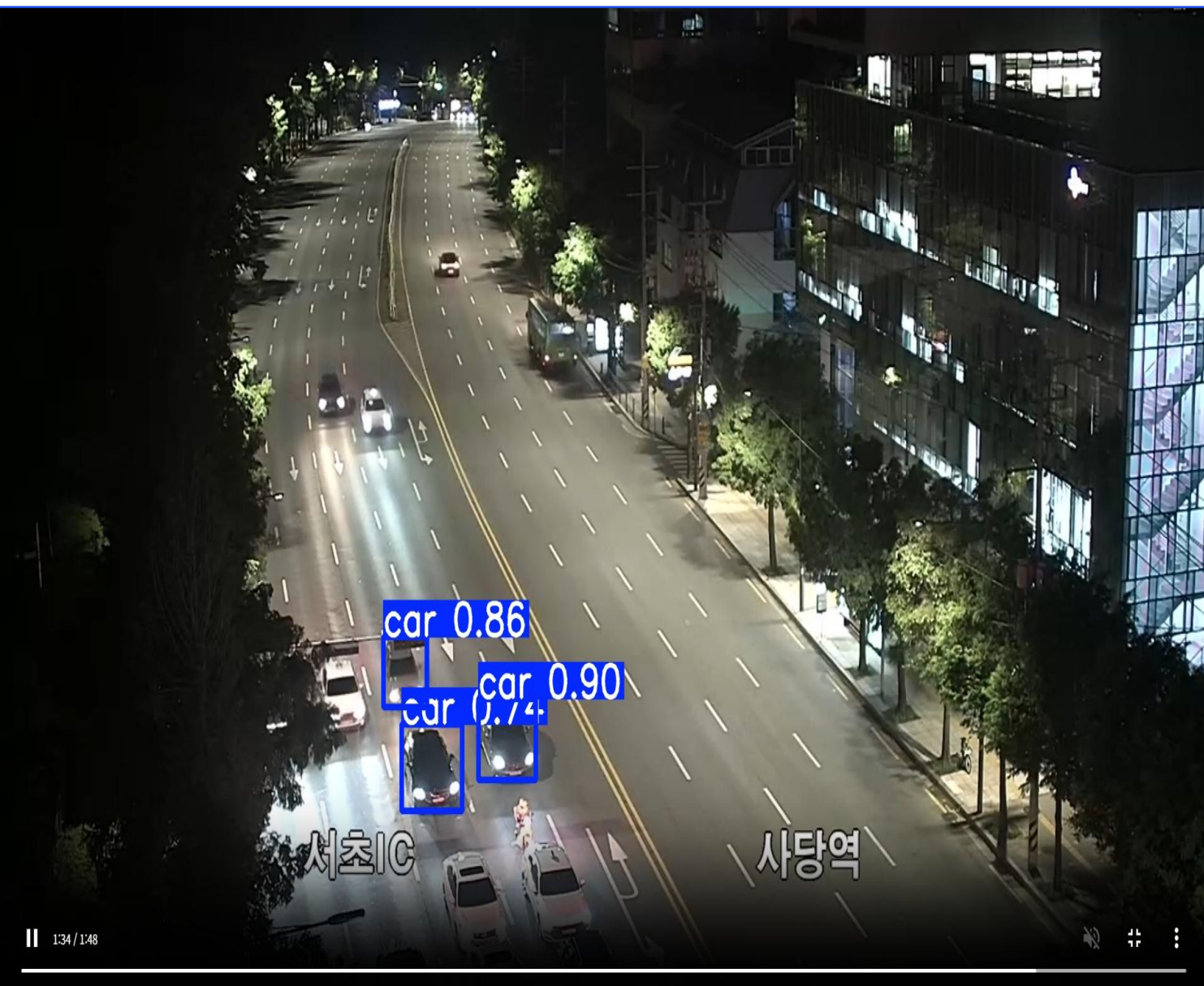
사용자의 주관 개입

- 사용자의 최대 차량 수를 필수로 입력 받는 문제
- 입력 내용의 일관성이 떨어지는 문제

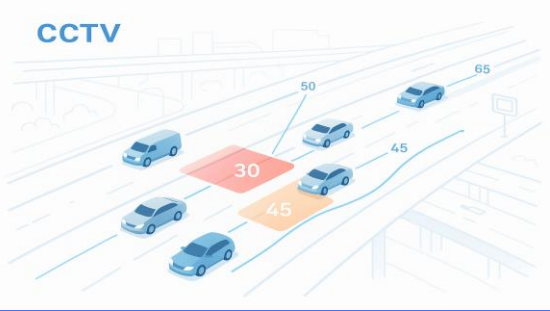
3

최종 서비스제공 방식

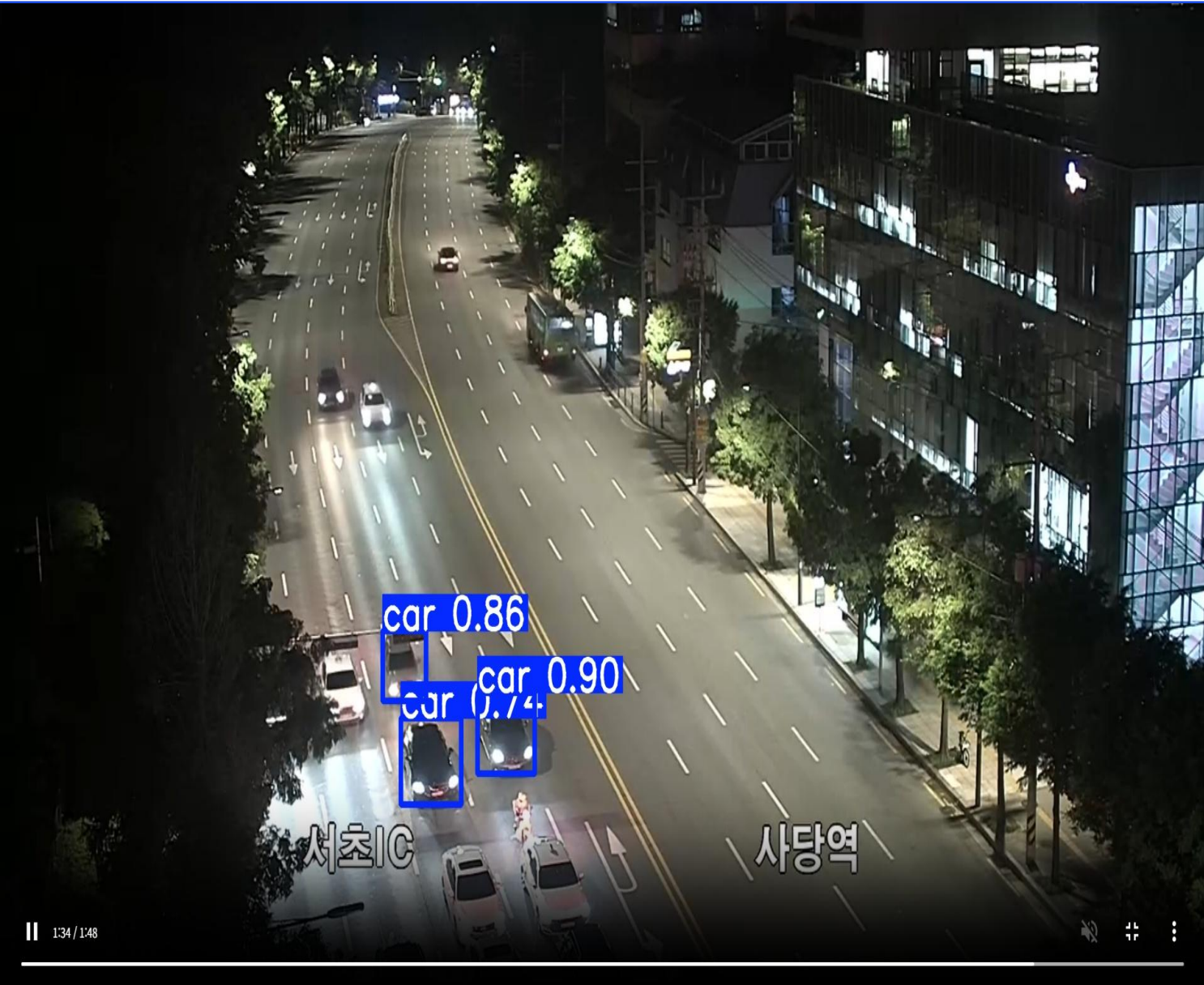
- 최종 서비스 제공 형식이 미정인 문제
- 복수, 단일 cctv처리에 따라서 모델이 변경되는 문제



	class_name	confidence
0	car	0.9023
1	car	0.8558
2	car	0.7376

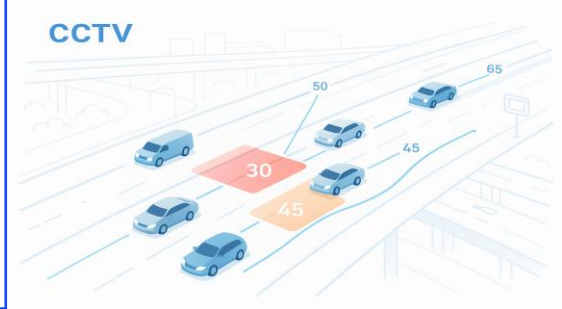


개선 방안



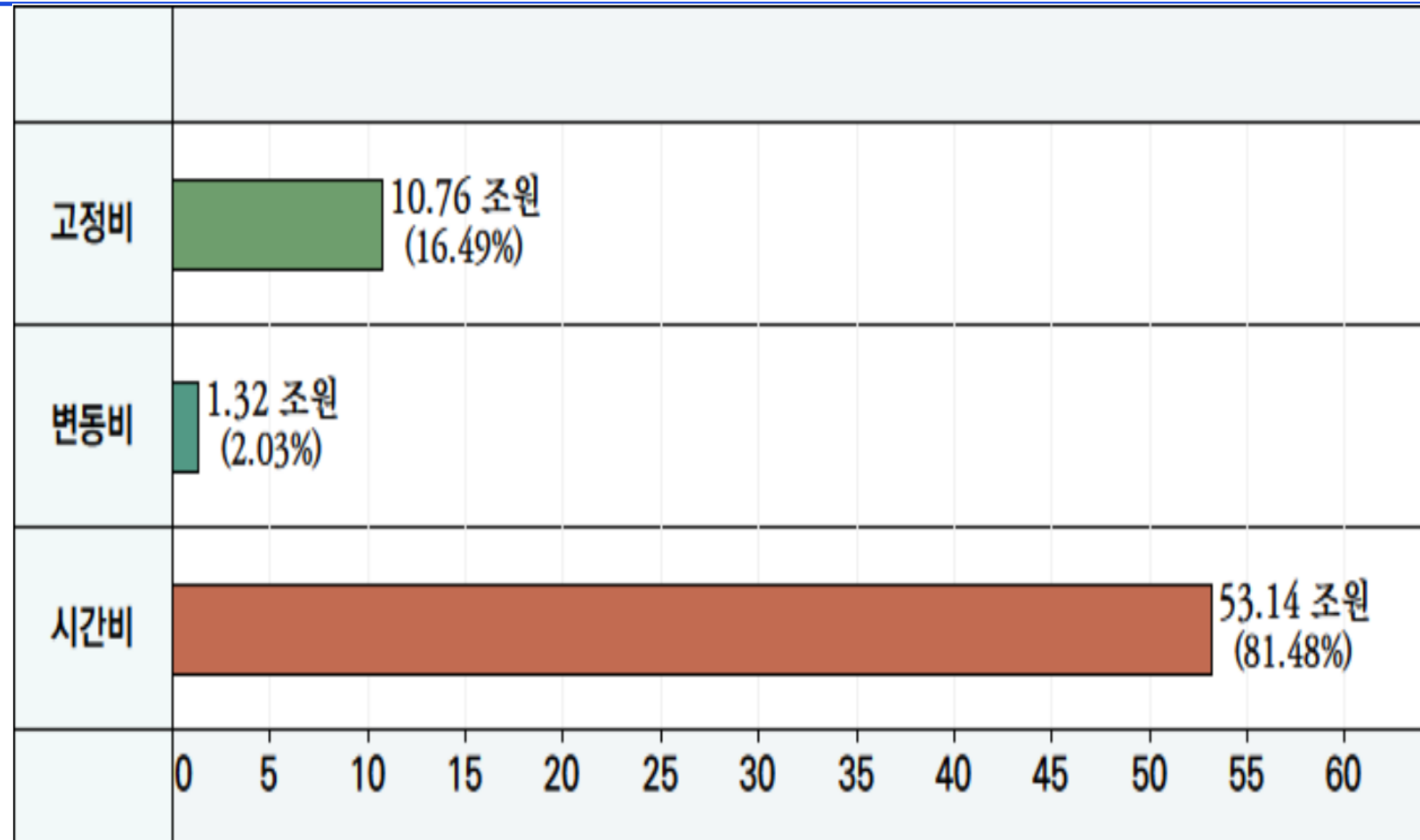
	class_name	confidence
0	car	0.9023
1	car	0.8558
2	car	0.7376

개선 단계	개선 방법	기대 효과
1단계	소량의 적용될 cctv 촬영 이미지로 모델의 미세조정	1. 객체의 인식을 상승 2. 객체당 신뢰도 상승
2단계	<p>도로의 정보만 추출하는 세그먼트 모델 추가 운영 및 추출된 도로에 카메라 캘리브레이션+호모그래피를 통한 도로의 정량화</p> <ul style="list-style-type: none">카메라 캘리브레이션 :카메라 렌즈로 인해 발생하는 이미지의 왜곡 감소 방법호모그래피 : 이미지를 정사형 시켜 버드 뷰의 이미지로 변환	1. 사용자 주관 배체 2. 분석 결과의 신뢰도 상승 3. 촬영 cctv의 변동으로 인한 모델 유지보수 난이도 하강
3단계	<p>서비스의 중요 사항에 따라서 모델 변경</p> <p>속도 우선 : Yolo11n모델 정확도 우선 : Yolo11x모델</p>	1. 속도 우선시 복수의 cctv를 분석에 용의 2. 정확도 우선시 분석의 오차 감소 및 일관성 상승



서비스 이용을 통한 기대효과

- 끼어들기, 급정거 등으로 발생하는 유령정체에 빠른 정보 전달
- 정체 구간 파악으로 차량 분산 유도
- 최종적으로 시간비 감소를 통해 연간 발생하는 70조원 가량의 교통 혼잡 비용에서 큰 폭의 감소를 기대



〈그림 4-3〉 2021년 구성요소별 교통혼잡비용 추정결과

THANK
YOU

감사합니다