

# 시각장애인의 안전한 보행을 보조하기 위한 주변탐지 솔루션

객체인식 프로젝트  
안희성      박준섭      김빛나

최종발표 2024. 2. 29

# TOC

## I. 프로젝트 개요

1. 기획 의도
2. 목표 산출물
3. 프로젝트 개요
4. 프로젝트 R&R

## I. 프로젝트 수행 결과

1. 데이터 수집 및 분석
2. 데이터 전처리
3. 모델 선정
4. 성능 및 테스트 결과
5. 개선 방향
6. 회고 (Lessons Learned)

# I. 프로젝트 개요

# Intro Questions

-----

*여러분은 밖에서 시각장애인을 만나보신 적 있나요?*

*길을 혼자 걷는 시각장애인을 얼마나 자주 보시나요?*

*우리가 매일 오가며 이용하는 도로와 대중교통, 시각장애  
인이 혼자 다니기에는 어떨 것 같나요?*

# 기획 의도 (1)

국내 시각장애인 26만명, 저시력자 인구 40만명  
**BUT!** 현역 활동 중인 안내견 77마리,  
 양성 및 훈련비용 1마리 당 약 1억원



(출처: 보건복지부, 삼성안내견학교, tvN 유퀴즈)

## 기획 의도 (2)

음향신호기 설치된 횡단보도는 34%에 불과  
**심지어!** 수동이라 직접 버튼 눌러야 하는데  
노후 기기 고장, 안내 점자블록 없어 **무용지물인** 경우도



(출처: 경찰청, 한국장애인고용공단)

## 기획 의도 (3)

— — — .

- 개발 배경: 혼자 독립적으로 보행할 수 있는 국내 시각장애인은 소수에 불과하며, 많은 시각장애인이 동네 앞, 문밖을 나서는 것조차 어려움을 느낌 (출처: 한국저시력인협회)
- 문제 정의: '이동'이 교통수단/여객수단/도로 등을 이용해 움직이는 권리를 포괄한다면, '보행'은 이동의 가장 기본이 되는 걷는 행위임 (참고: 교통약자의 이용편의 증진법 제3조)
- 기획 의도: 시각장애인의 안전한 이동을 위한 첫걸음으로, 인공지능 객체인식 기술을 활용한 안전 보행 보조 솔루션을 제공하고자 함
- 전달 가치: 공급 및 설치가 부족한 안내견과 음향신호기를 대신하여, 시각장애인의 이동권/접근권 향상에 기여함으로써 경제적/사회적 가치 창출이 기대됨

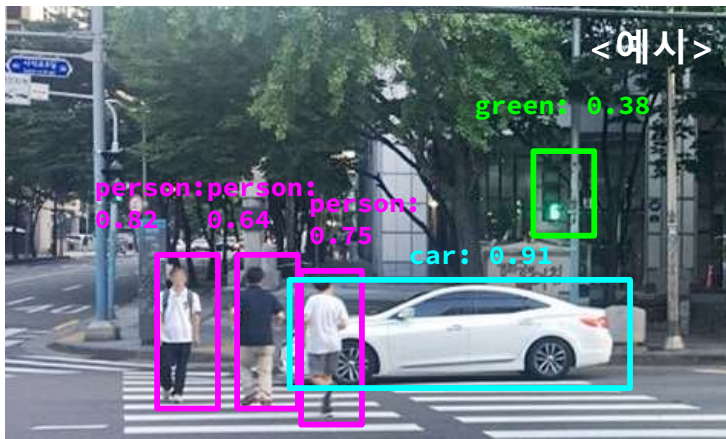
# 목표 산출물

## ● 프로젝트 구현 목표

- (Object Detection 모델) 보행 환경에서 **주요 객체를 식별**

## ● 최종 산출물 청사진

- (카메라) 영상 처리를 활용하여 **객체를 실시간 식별**하고
- (스피커) 안전한 보행을 위한 주요 정보를 **음성으로 실시간 안내**하는 서비스



"보행 신호등 초록불 입니다"  
"전방에 사람 3명, 차량 1대 있습니다"



# 프로젝트 개요

— — — .

일정		사용 자원, 기법, 모델 등
2/6 - 8	주제 선정, 데이터 수집	
2/14 - 16	데이터 EDA, 전처리	PIL, OpenCV
2/16 - 18	object detection 모델링	YOLO v4 with Darknet
<b>2/19(월)</b>	<b>중간발표</b>	
2/19 - 28	데이터 전처리 및 보강	PIL, OpenCV
2/23 - 29	object detection 모델링, 성능 비교	YOLO world, v8 with PyTorch
<b>2/29(목)</b>	<b>최종발표</b>	

# 프로젝트 R&R

-----

	기획	데이터 수집	EDA/전처리	모델링	PPT/발표
안희성	○	●	●	●	
박준섭	○	●	●	●	●
김빛나	●	●	●	○	●

## II. 프로젝트 수행 결과

# 데이터 수집 및 분석

-----

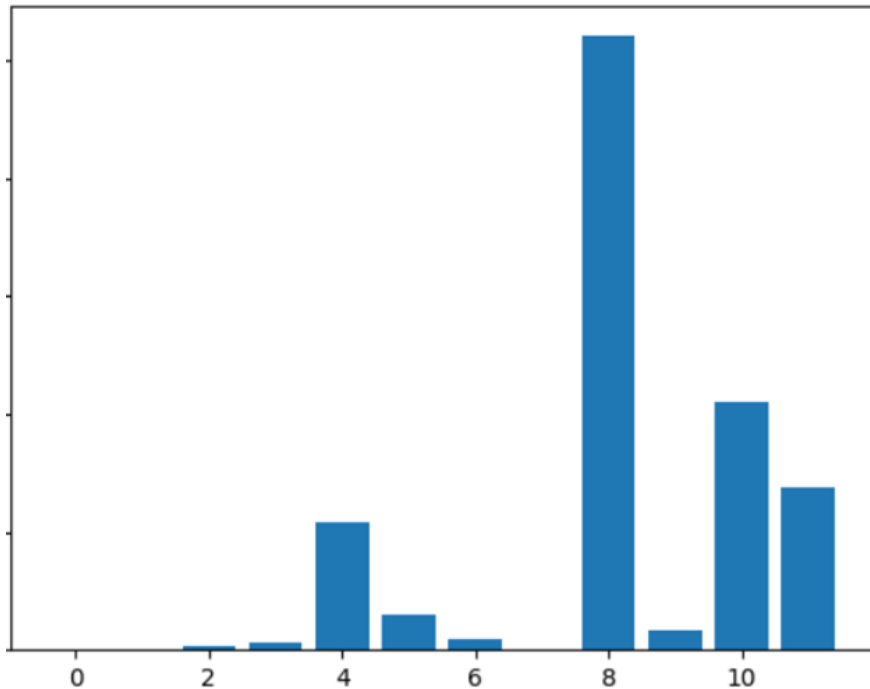
- 기초 데이터 출처: AI hub <인도보행 영상>
  - 한계: '신호등' 라벨에서 보행등 또는 차량등 여부, 색깔 미분류
  - 보완: Kaggle 및 Roboflow <보행 신호등 이미지> 'G/R' 라벨링 하여 클래스 추가
- Labeling 및 Indexing: 클래스 개수 기존 30개 이상 ⇒ 'n\_class = 11'로 재분류
  - 재분류 기준 및 고려사항 (서비스 관점)
    - (1) 시각장애인 보행 시 위험도 ⇒ 시각장애인 관점에서 안내가 필요한 객체를 포함하나?
    - (2) 클래스 내 라벨 특성의 유사도 ⇒ 안내 시 이해하기 쉽게 클래스로 범주화 되었나?
  - 중간발표 이후 추가 고려사항 (모델링 관점)
    - (3) 재분류된 클래스 별 학습 데이터 개수가 충분하고 균등하게 분포하나? ⇒ **NO**

## (참고) 라벨 재분류 결과 (최종발표)

기존 클래스	재분류	서비스 구현 시
green_light	0:green	보행등 초록불
red_light	1:red	보행등 빨간불
bicycle, motorcycle, scooter, carrier, stroller, wheelchair	2:vehicle	탈 것
person	3:person	사람
bench, chair, table	4:rest	휴식 공간
bus_stop, taxi_stop	5:stop	버스/택시 정류장
bollard	6:bollard	볼라드
fire_hydrant, parking_meter, power_controller, movable_signage, barricade	7:obstacle	기타 장애물
tree_trunk, potted_plant	8:plant	나무
bus, car, truck	9:car	차량
pole	10:pole	기둥

## (참고) 라벨 별 데이터 분포 (중간발표 기준)

*Too Unbalanced :(*

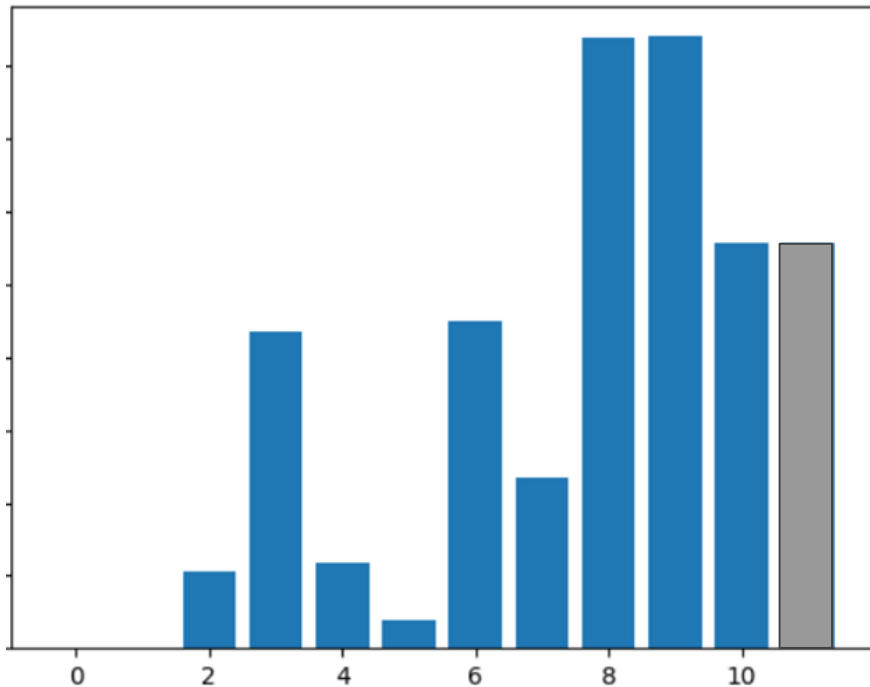


```
index_to_classname = {  
    'Green_light' : 0,  
    'Red_light' : 1,  
    'Mobility aid' : 2,  
    'Person' : 4,  
    'Rest' : 5,  
    'Stop' : 6,  
    'Kiosk' : 7,  
    'Obstacle' : 8,  
    'Bicycle' : 9,  
    'Car' : 10  
}
```

## (참고) 라벨 별 데이터 분포 (중간발표 후 재분류)

-----

*More Balanced*  
*BUT! 여전히 개선의 여지가 있음*



```
index_to_classname = {  
    'Green_light' : 0,  
    'Red_light' : 1,  
    'Vehicle' : 2,  
    'Person' : 3,  
    'Rest' : 4,  
    'Stop' : 5,  
    'Bollard' : 6,  
    'Obstacle' : 7,  
    'Plant' : 8,  
    'Car' : 9,  
    'Pole' : 10,  
    'Others' : 11  
}
```

# 데이터 전처리

— — — .

- 이미지 리사이징

- 640 x 640 (v8) (YOLO의 일반적인 입력 데이터 사이즈: 32의 배수)

- Bounding Box (Bbox) 좌표 포맷 YOLO 스타일로 변경

- 기초 데이터:  $(x1, y1, x2, y2) = (\text{좌상단 } x, \text{좌상단 } y, \text{우하단 } x, \text{우하단 } y)$
  - YOLO 스타일:  $(x, y, w, h) = (\text{center } x, \text{center } y, \text{ratio } w, \text{ratio } h)$

- 데이터 증강

- 라벨 재분류 후에도, 여전히 일부 클래스의 데이터 개수 부족/불균형
  - 추가한 신호등 이미지의 크기와 형태가 <인도보행 영상> 기초 데이터와 크게 상이함



# 데이터 증강 시도 (1) 모자이크



*모자이크 결과물, 객체 인식에 부적절*

*Then, How?*

# 데이터 증강 시도 (2) 이미지 합성

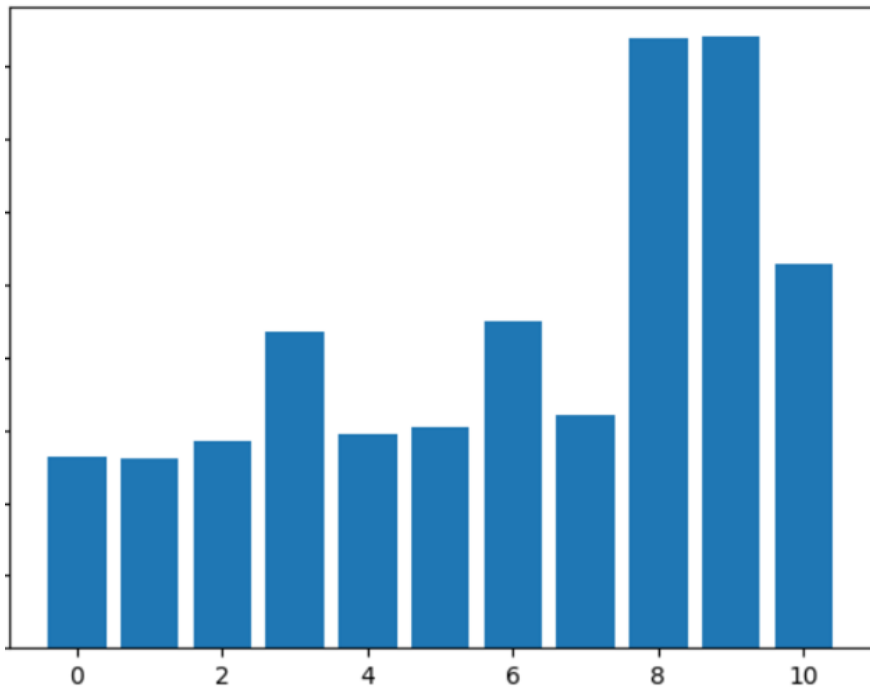
- (1) 대상 객체 **크롭** > (2) 너무 크지 않도록 max\_size 설정 & **리사이징** >  
 (3) **너무 많은 객체가 겹치지 않도록**, 배경 이미지 중 객체가 최대 5개인 **배경 추출** >  
 (4) 클래스 별 증강 비율 설정해 **객체 합성** > (5) 합성된 객체 **정보 업데이트**



(출처: AI hub, Roboflow)

## (참고) 라벨 별 데이터 분포 (데이터 증강 후 최종발표)

-----  
*이미지 합성을 통한  
객체 데이터 증강 후 ⇒ Well Balanced*

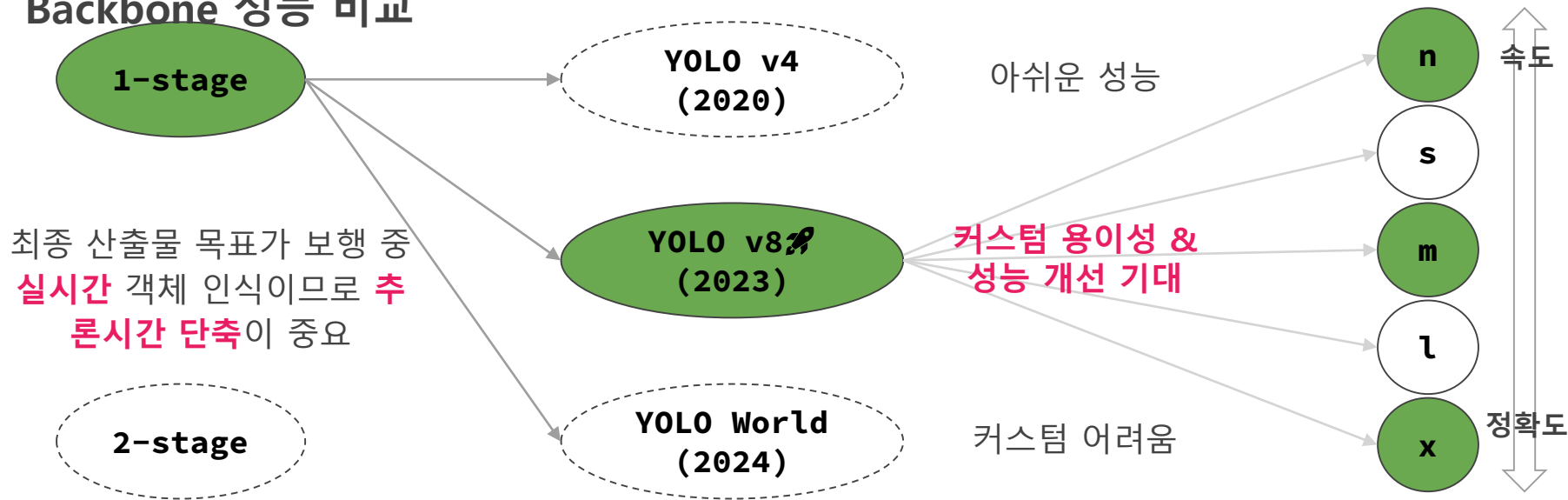


```
index_to_classname = {  
    'Green_light' : 0,  
    'Red_light' : 1,  
    'Vehicle' : 2,  
    'Person' : 3,  
    'Rest' : 4,  
    'Stop' : 5,  
    'Bollard' : 6,  
    'Obstacle' : 7,  
    'Plant' : 8,  
    'Car' : 9,  
    'Pole' : 10,  
}
```

# 모델 선정

모델 선정: YOLO  
Backbone 성능 비교

다양한 버전 테스트



## (참고) YOLO v4 - 테스트 결과물 (중간발표)



사전학습된 모델 및 가중치

(yolov4.conv.137)로 custom data  
(1.7만여 개, 410x410) 학습 결과:

- 낮은 정확도(mAP@0.50: 52%)
- '장애물' 클래스만 식별함

⇒ 문제 원인 파악하여 수정 예정

- 가설1) 클래스 별 학습 데이터가 불균형해서 그런가?
- 가설2) confidence threshold를 조정해 볼까?

(참고)



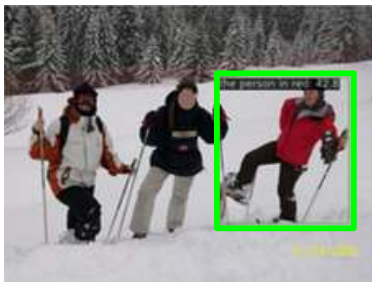
YOLO-World

(1) 소개 및 장점

● YOLO-World “**Real-Time, Zero-Shot** Object Detection”

- 2024/1/31 텐센트 AI Lab에서 공개한 **실시간, 개방형** 어휘 개체 감지 모델
- 사전 학습된 오프라인 어휘 (COCO-80 카테고리) 즉시 적용  
+ 사용자 지정 프롬프트로 커스텀 클래스 동적 지정

⇒ 재학습 없이도 새로운 도메인/어휘에 모델 즉시 적용 가능



the person in red



the brown animal



the tallest person

(출처: Ultralytics)

(참고)



YOLO-World

(2) 적용 한계

- — — .
- **BUT!** 여러 조합으로 커스텀 클래스 지정하여 테스트한 결과, 아래 문제로 **채택 안기로 함**
    - (1) 새로운 '**객체 식별**' 성능에 대한 신뢰도가 낮음
      - COCO-80으로 학습하지 않은 오프라인 객체(예: Bollard) 인식 못함
      - 신호등 초록불 or 빨간불을 식별은 하지만, 데이터마다 결과가 상이함
    - (2) 우리 프로젝트에서 원하는 커스텀 '**객체 범주화**'가 어려움
      - 예1: "도로보행 장애물" 등 주관적 조건의 범주화가 현실적으로 어려움
      - 예2: "2륜차량", "4륜차량", "탈것" 등 객관적 조건의 범주화/식별 성능도 낮음

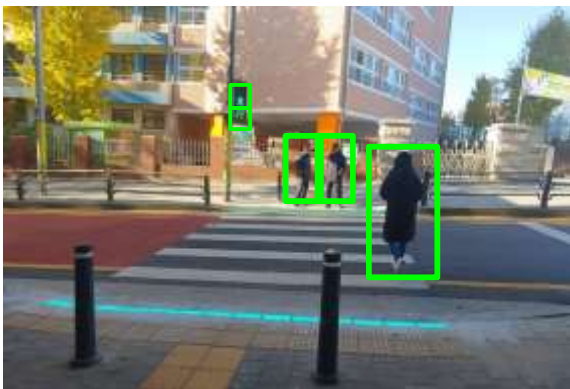


(참고)



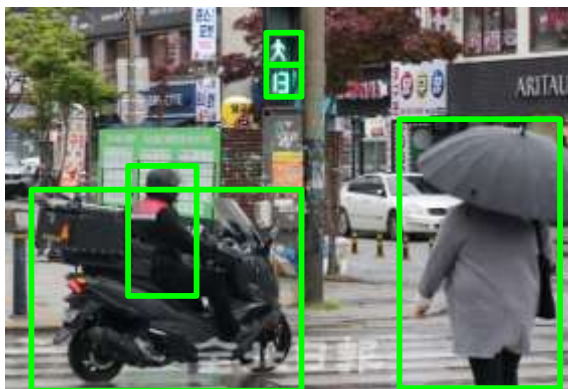
## (3) 테스트 결과물

```
# Define custom classes
model.set_classes(["bollard", "4 wheel vehicle", "2 wheel vehicle", "bike",
"person", "green_traffic_light", "red_traffic_light", "obstacle_for_passenger"])
```



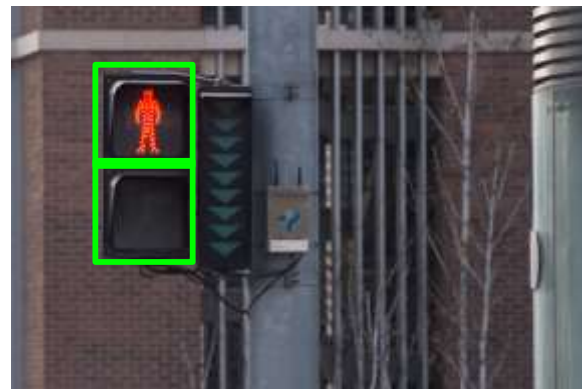
```
3 persons,
2 green_traffic_lights
```

**볼라드는?**



```
1 bike, 2 persons,
1 green_traffic_light
```

**오토바이를 2륜차가 아닌  
자전거로 인식하네?**



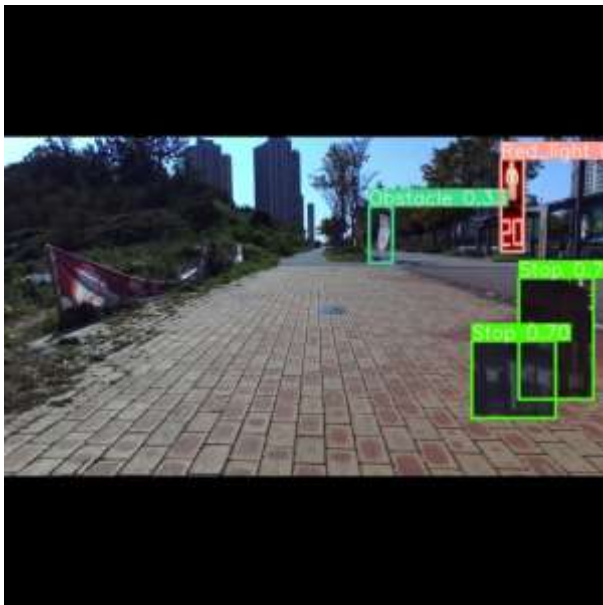
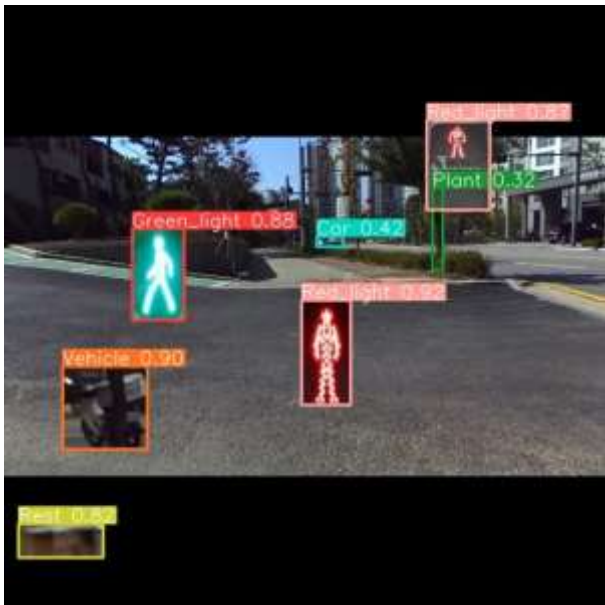
```
2 green_traffic_lights
```

**초록불이라고?**



# 성능 및 테스트 결과 (YOLO v8🚀)

- 성능(mAP@50): N 18% → M 19% → (L: OutOfMemoryError) → XLarge 21%
- 아래 Sample Data: Validation set, Model: YOLOv8n, Epoch: 30, Optimizer: Adam, LR: 0.01





**NO . . .  
WHY?????**

## ● Object 별로는, 최고 성능:

### 신호등 VS 최저 성능: 기둥

신호등은 형태가 일관된 반면,  
기둥은 형태가 유사한 다른 객체들 많이 정확히 구별 못함

## ● 성능 저하 원인 가설

### 1) 데이터 관점:

이미지 합성 시 객체 위치를 랜덤 설정했는데, 기존 객체가 합성 객체에 가려지는 경우 학습에 혼선이 발생했을까?

### 2) 모델 관점:

YOLO v8 학습 시 Epoch 횟수가 충분하지 않아서일까?

# 개선 방향

## As-Is


- **(프로세스 측면)** 클라우드에 저장한 대용량 데이터를 전처리 할 때, 여러 파일과 단계로 나눠서 불러오고 처리하는 과정에서 크게 지연됨
- **(데이터 측면)** 데이터 증강 목적으로 크롭/합성을 했지만, 기존 객체를 거의 변형 없이 사용해 실제 학습 데이터 개수에서는 허수로 감안해야 함
- **(모델 측면)** 다양한 사전학습된 모델들을 비교해 봤으나, 구조 상 튜닝을 통한 성능 향상 여지는 적음
- **(성능 측면)** 데이터 전처리 후 최신 버전 모델을 돌렸음에도, 오히려 성능이 저하됨(v4 50% → v8 20%)
- **(서비스 구현 측면)** 시각장애인이 독립적으로 안전한 보행을 하기에는 기능적으로 한계가 있음

## To-Be

- 데이터를 로컬에 저장하거나, 데이터를 처리하는 코드를 하나의 단일 파일에서 작성/실행할 경우 시간 단축이 기대됨
- 데이터 증강이 필요 없도록 원 데이터를 선별하거나, 데이터 증강 시 다양한 변형 기법을 시도해볼 수 있음
- 시간 한계 상 Object Detection만 시도했지만, 추후 Segmentation도 도전해보고 최근 공개된 YOLO v9도 돌려보고 싶음
- 데이터 증강 접근법이 문제인지, 모델 학습이 충분하지 않았는지 문제 원인을 찾고자 함
- 노면 인식, 위험 객체 Tracking, 깜빡이는 초록불 Alert 등 추가 기능도 구현하고 싶음

# 회고 (Lessons Learned)

— — — .

- 이번 프로젝트에서는 데이터뿐 아니라 **모델 관점에서도 더 다양한 실험**을 하여 모델 간 성능 비교와 성능 향상 시도를 하고자 했다
  - 프로젝트에 적합한 커스텀 데이터를 만들기 위한 **전처리 과정에 예상보다 시간이 많이 소요** 되었으나, 이전에 비해서는 **보다 다양한 모델들을 실험**하고 비교해볼 수 있었다
- 성능 향상을 위해 **데이터 단에서 노력과 시간 리소스**를 많이 투입했음에도 불구하고, 오히려 **최신 버전 모델에서 성능이 저하되어 많이 속상했다** 
  - 그래도, 원 데이터셋에서 못한 **보행등 색깔 구분을 구현**한 점에서 희망을 보았다
- 이전 프로젝트에서 겪었던 **프로세스 차원의 낭비를 줄이기 위해 커뮤니케이션을 보다 원활히 하는 등 노력을 다같이 하여 불필요한 시간낭비나 실수는 확연히 개선되었다**
  - 후반부 시간 부족으로 직접 촬영한 영상으로 테스트 시연을 보여주지 못해 아쉬웠다

감사합니다