종합설계프로젝트 2 중간발표 2

CCTV 영상 분석 기반의 차종 인식을 통한 교통량 분석

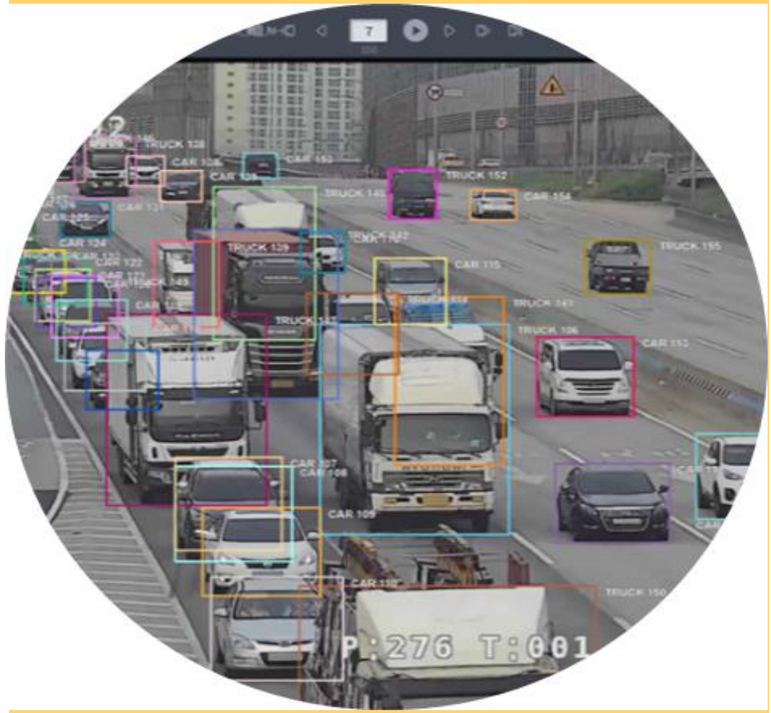
5팀 범일정보
발표 : 최현권

컴퓨터학부	18학번	김종우
컴퓨터학부	18학번	이장훈
컴퓨터학부	18학번	정상훈
컴퓨터학부	18학번	최현권
컴퓨터학부	18학번	허민석

LIST OF CONTENTS

수행 배경 및 목표

- 교통 영향 평가에 드는 시간적, 인적 비용
- 기 설치된 CCTV의 영상이나 기 촬영된 영상을 통해 다양한 데이터를 수집
- 수집한 데이터를 이용하여 교통량, 차종, 과정에 따라 불법주정차 정보를 분석
- 추가적인 활용 가능성

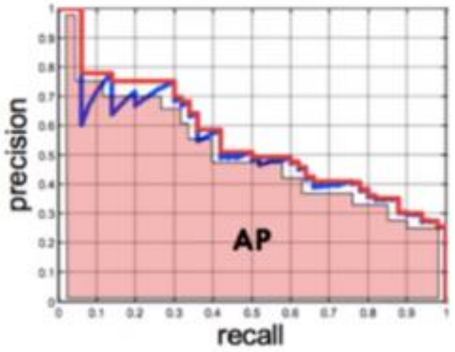


시스템 요구분석 및 정의

기능 요구사항

- 양방향 도로 동영상에서 차량 인식 기능
- YOLO v5 객체 탐지 모델을 사용한 차종 3종 (이륜차, 승용차, 화물차) 구분 기능
- 인식된 차종의 수 계산 기능
- 객체 추적 알고리즘을 사용한 객체 트래킹 기능
- 트래킹 기능을 사용한 불법 주정차 탐지 기능
- 분석 결과 저장 기능
- 도로 교통량 분석 결과 GIS기반의 모니터링 서비스 기능

비기능 요구사항



- 정확도 : 80% 이상의 mAP 성능으로 차량을 인식해야 함
- 사용성 : 사용자 친화적인 UI/UX를 제공해야 함
- 가용성 : 사용자가 원하는 순간에 시스템은 서비스를 제공해야함
- 안정성 : 오류를 최소화하고 안정적으로 작동해야 함

- Precision (정밀도) : 모델이 True라고 분류한 것 중에서 실제 True인 것의 비율
- Recall (재현율) : 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율

시스템 구조 및 설계

1. 시스템 실행 환경

•프로그램은 Python 언어로 작성되며, PyTorch, OpenCV, Pandas 등의 라이브러리를 사용, YOLO v5 모델을 객체 탐지에 이용, 차종의 수 계산을 NumPy로 수행하고, 결과를 Dash라는 Plotly의 웹 프레임워크를 사용하여 Plotly를 이용한 차종 구분 및 교통량 분석 결과를 인터랙티브한 웹으로 구현

PyTorch

YOLOv5

시스템 구조 및 설계

2. 시스템 구조

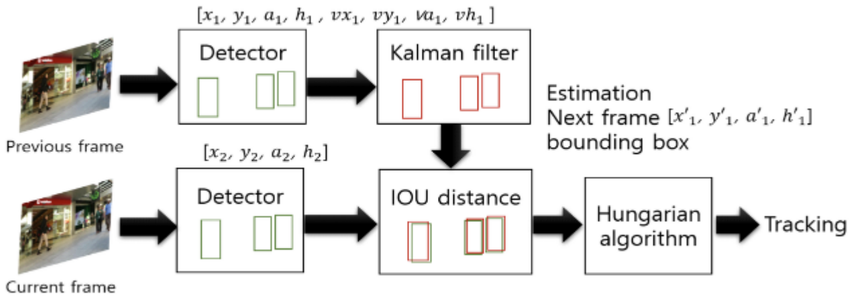
• 사용자 인터페이스 (UI) 모듈 : GIS기반의 위치 선택, 분석 결과 시각화, 재실행 등의 기능을 제공

• 객체 탐지 모듈 : 1-Stage Detector로 속도에 강점을 가지는 YOLO v5 객체 탐지 모델을 사용하여 차량 인식과 차종 구분 기능을 수행

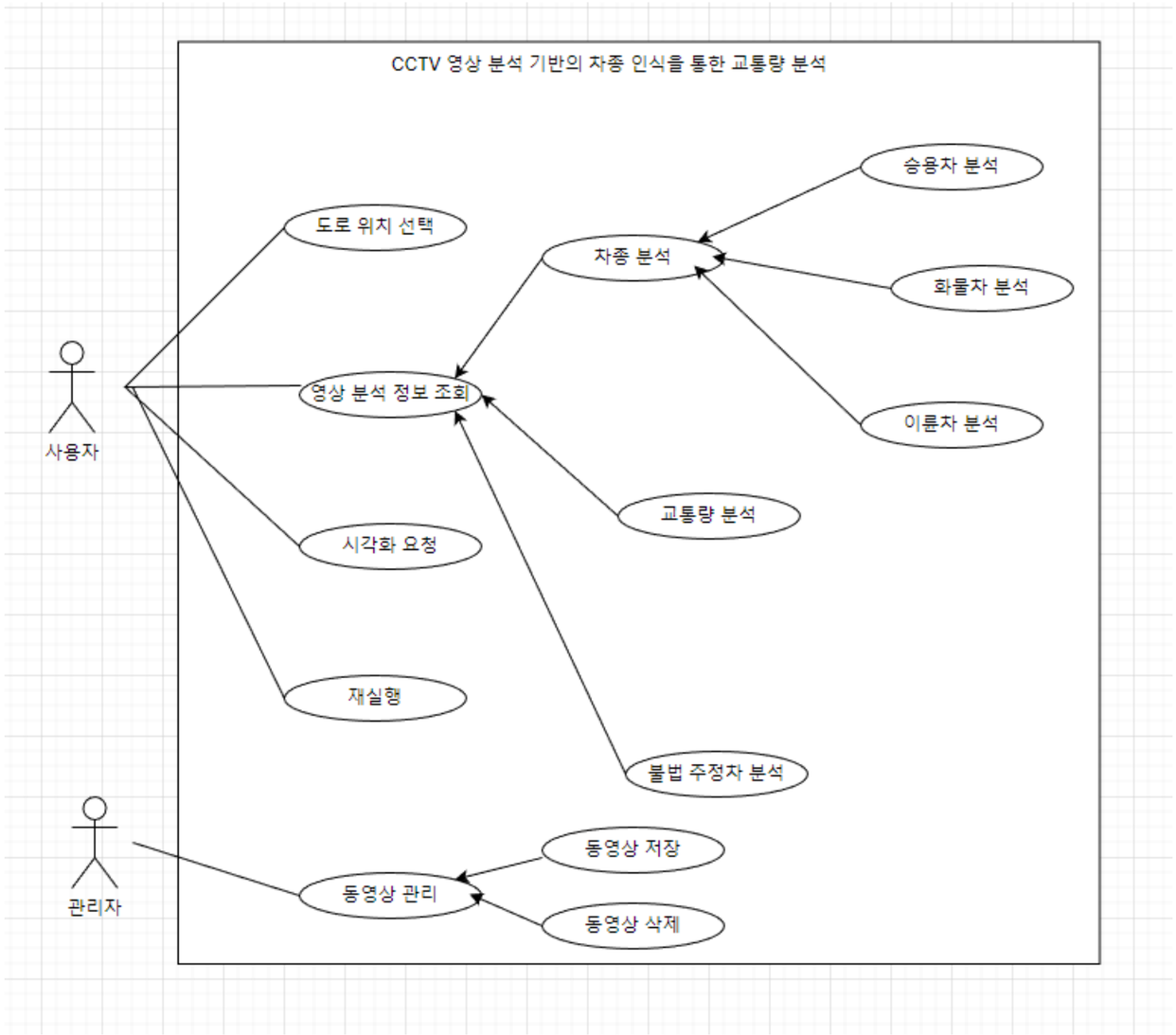
• 교통량 분석 모듈 : SORT(Simple Online and Realtime Tracking) 객체 추적 알고리즘을 사용하여 인식된 차종의 수를 계산하여 도로 교통량을 분석

• 불법 주정차 탐지 모듈 : 위와 같은 SORT 객체 추적 알고리즘을 사용하여 불법 주정차 구역을 설정한 뒤, 일정 프레임 이상 움직임이 없다면 불법 주정차로 판단하고, 그 차량 수를 계산

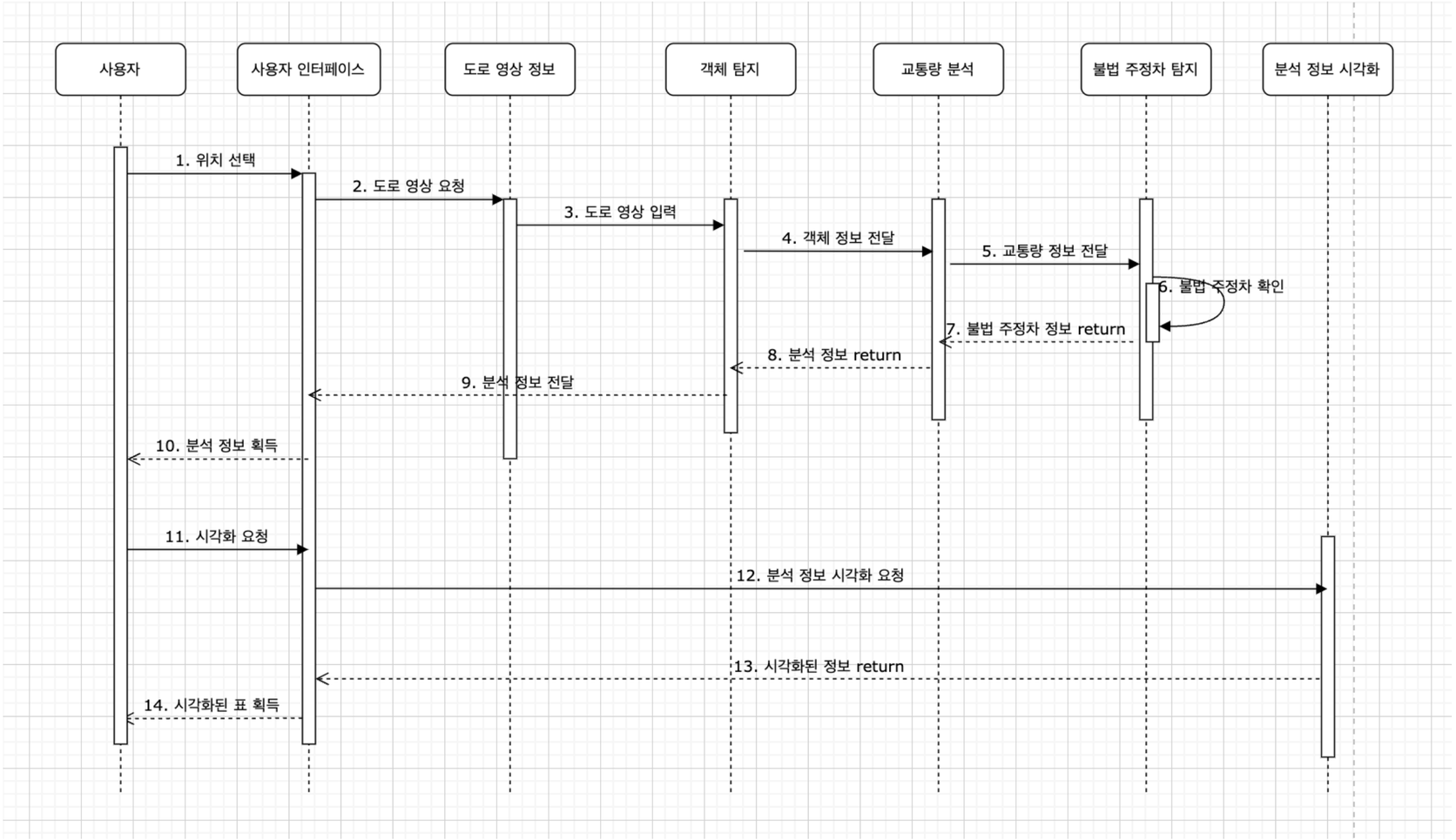
• 시각화 모듈 : 교통량 분석 정보 및 불법 주정차 탐지 정보를 Dash라는 Plotly의 웹 프레임워크를 사용하여 Plotly를 이용한 차종 구분 및 교통량 분석 결과를 인터랙티브한 웹으로 구현



시스템 구조 및 설계 – Use Case Diagram

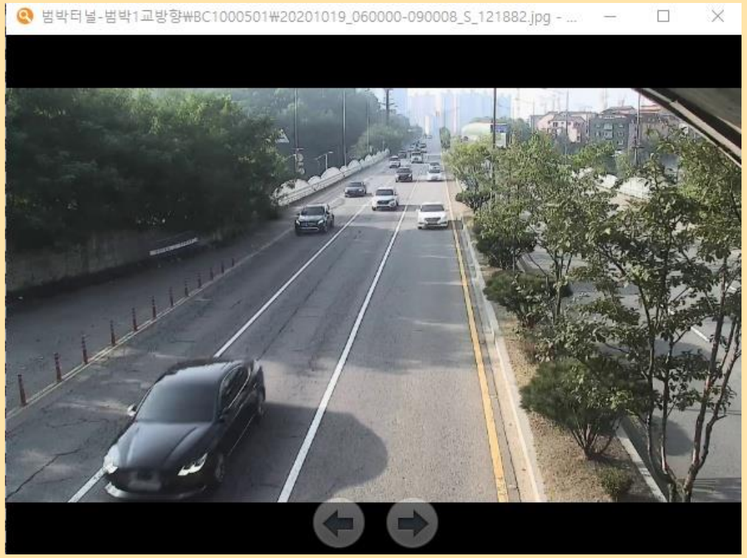


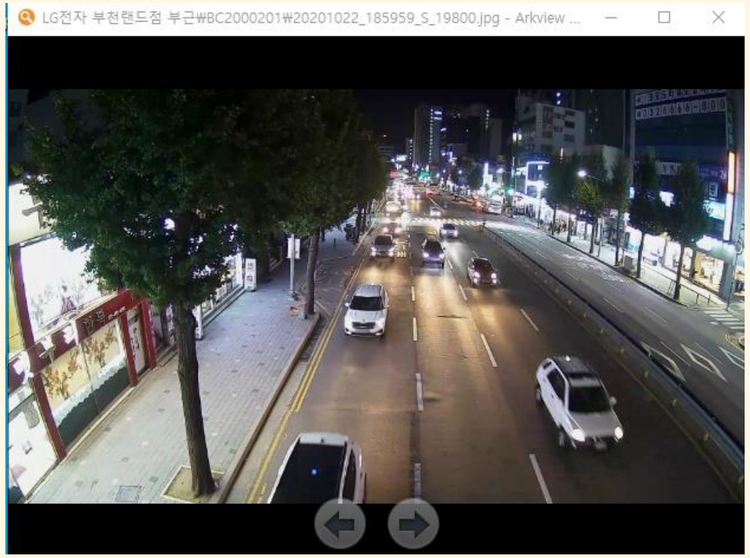
시스템 구조 및 설계 – Sequence Diagram

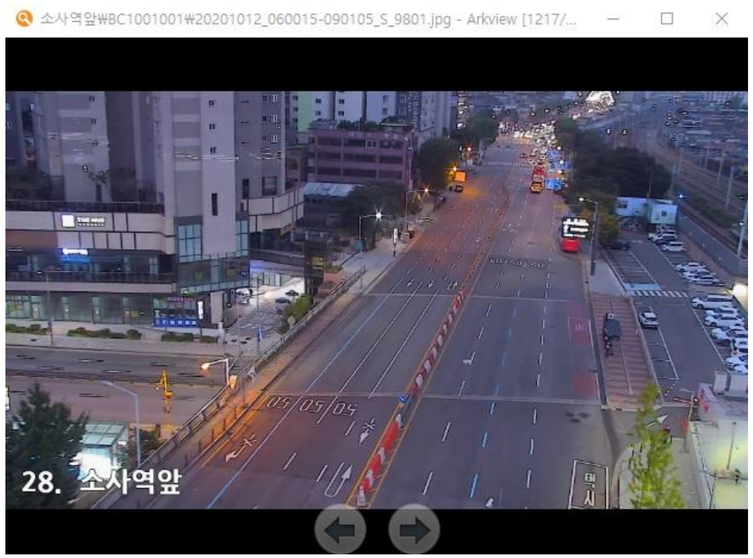


진행 상황

< 라벨링 작업 및 데이터 셋 구축 >







• AI hub, 기업에서 제공하는 CCTV 동영상 파일을 통해 데이터 셋을 구축하고,
bounding box로 구분하여, class 0, 1, 2에 따라 car, truck, motorcycle 로 라벨링

진행 상황

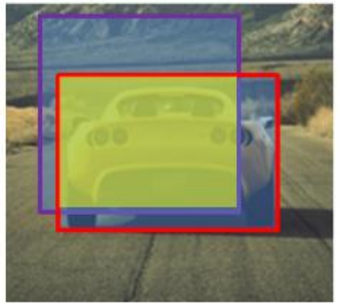
< 2400장 학습 >

```
      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      29/29      3.84G    0.06074    0.03808   0.008547         65        640: 100%|██████████| 151/151 [00:39<00:00,  3.85it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:07<00:00,  2.53it/s]
                   all        607       5385      0.696      0.478      0.565      0.363

30 epochs completed in 0.407 hours.
Optimizer stripped from runs\train\exp20\weights\last.pt, 14.4MB
Optimizer stripped from runs\train\exp20\weights\best.pt, 14.4MB

Validating runs\train\exp20\weights\best.pt...
Fusing layers...
Model summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:08<00:00,  2.14it/s]
                   all        607       5385      0.666        0.5      0.568      0.368
                   car        607       3945      0.593      0.585      0.629      0.437
                 truck        607       1304      0.821      0.524      0.652      0.437
            motorcycle        607        136      0.584      0.392      0.422      0.231
Results saved to runs\train\exp20
```

- Precision (정밀도) : 0.666
- Recall (재현율) : 0.5
- mAP50 : 0.568
- mAP50-95 : 0.368

- Precision (정밀도) : 모델이 True라고 분류한 것 중에서 실제 True인 것의 비율
- Recall (재현율) : 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율
- mAP50 : Pascal VOC의 mAP 평가 방식 (IoU > 0.5 인 detection은 true, 그 이하 false)
- mAP50-95 : COCO의 mAP 평가 방식 (IoU > 0.5, IoU > 0.55, .., IoU > 0.95
  각각을 기준으로 계산 후 평균)

IoU 계산 방법은 두 바운딩 박스를 가지고 IoU = 교집합 영역 넓이 / 합집합 영역 넓이 를 계산해 나온 값이 IoU이다.



Intersection over union (IoU)

$$= \frac{\text{size of}}{\text{size of}}$$

"Correct" if IoU $\geq 0{,}5$

IoU 계산 방법

진행 상황

< 3600장 학습 >

```
      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      29/29      3.87G    0.04937    0.03375   0.006952         11        640: 100%|██████████| 223/223 [00:53<00:00,  4.18it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:05<00:00,  3.21it/s]
                   all        607       5385      0.736      0.493      0.597      0.394

30 epochs completed in 0.496 hours.
Optimizer stripped from runs\train\exp21\weights\last.pt, 14.4MB
Optimizer stripped from runs\train\exp21\weights\best.pt, 14.4MB

Validating runs\train\exp21\weights\best.pt...
Fusing layers...
Model summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:07<00:00,  2.54it/s]
                   all        607       5385      0.757      0.486      0.607      0.395
                   car        607       3945      0.657      0.535      0.627      0.439
                 truck        607       1304      0.937      0.475       0.71      0.484
            motorcycle        607        136      0.677      0.449      0.483      0.263
Results saved to runs\train\exp21
```

- Precision (정밀도) : 0.757
- Recall (재현율) : 0.486
- mAP50 : 0.607
- mAP50-95 : 0.395

- Precision (정밀도) : 모델이 True라고 분류한 것 중에서 실제 True인 것의 비율
- Recall (재현율) : 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율
- mAP50 : Pascal VOC의 mAP 평가 방식 (IoU > 0.5 인 detection은 true, 그 이하 false)
- mAP50-95 : COCO의 mAP 평가 방식 (IoU > 0.5, IoU > 0.55, .., IoU > 0.95
  각각을 기준으로 계산 후 평균)

< 5800장 학습 >

```
      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      29/29      3.84G    0.02761     0.0306   0.003736         81        640: 100%|██████████| 356/356 [01:41<00:00,  3.50it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:07<00:00,  2.64it/s]
                   all        607       5385       0.67      0.517      0.605       0.41

30 epochs completed in 0.916 hours.
Optimizer stripped from runs\train\exp22\weights\last.pt, 14.4MB
Optimizer stripped from runs\train\exp22\weights\best.pt, 14.4MB

Validating runs\train\exp22\weights\best.pt...
Fusing layers...
Model summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:08<00:00,  2.19it/s]
                   all        607       5385      0.724      0.532      0.631      0.421
                   car        607       3945      0.715      0.573      0.689      0.486
                 truck        607       1304      0.916      0.558      0.754      0.519
            motorcycle        607        136      0.541      0.463      0.451      0.257
Results saved to runs\train\exp22
```

- Precision (정밀도) : 0.724
- Recall (재현율) : 0.532
- mAP50 : 0.631
- mAP50-95 : 0.421

진행 상황

< 9000장 학습 >

```
      Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
      29/29      3.85G    0.02617    0.03188   0.003666         82        640: 100%|██████████| 554/554 [02:07<00:00,  4.36it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:05<00:00,  3.63it/s]
                   all        607       5385      0.584      0.546      0.561      0.383

30 epochs completed in 1.310 hours.
Optimizer stripped from runs\train\exp23\weights\last.pt, 14.4MB
Optimizer stripped from runs\train\exp23\weights\best.pt, 14.4MB

Validating runs\train\exp23\weights\best.pt...
Fusing layers...
Model summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:07<00:00,  2.64it/s]
                   all        607       5385      0.734      0.546      0.603      0.397
                   car        607       3945      0.609      0.611      0.616      0.433
                 truck        607       1304      0.923      0.549      0.721      0.494
            motorcycle        607        136      0.669      0.478      0.472      0.264
```

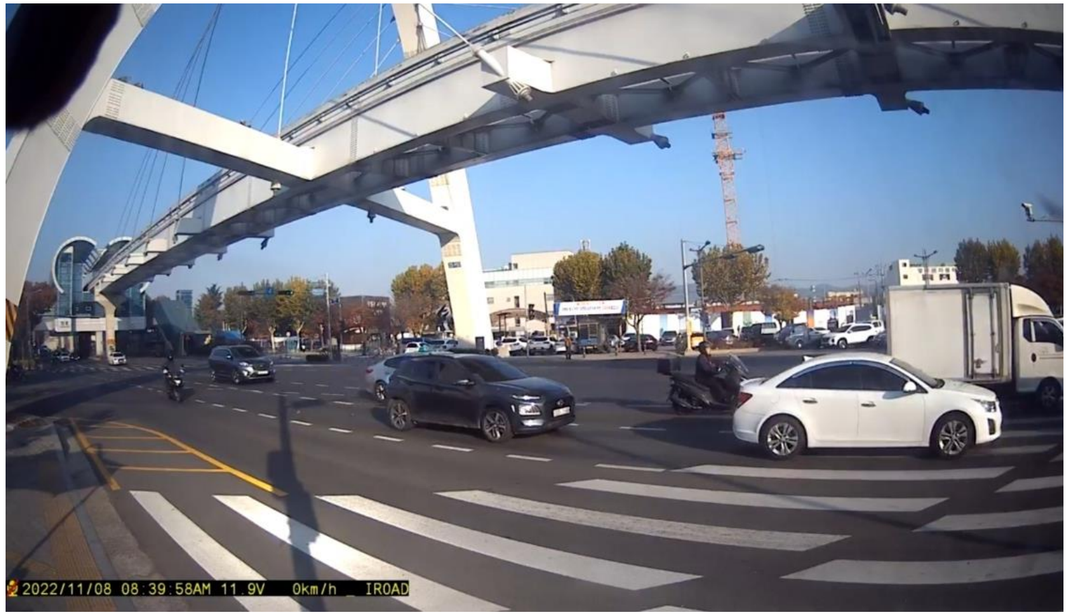
- Precision (정밀도) : 0.734
- Recall (재현율) : 0.546
- mAP50 : 0.603
- mAP50-95 : 0.397

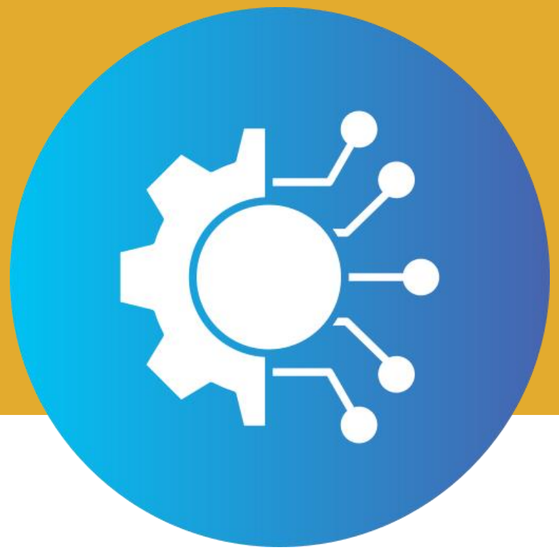
- Precision (정밀도) : 모델이 True라고 분류한 것 중에서 실제 True인 것의 비율
- Recall (재현율) : 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율
- mAP50 : Pascal VOC의 mAP 평가 방식 (IoU > 0.5 인 detection은 true, 그 이하 false)
- mAP50-95 : COCO의 mAP 평가 방식 (IoU > 0.5, IoU > 0.55, .., IoU > 0.95
  각각을 기준으로 계산 후 평균)

- 훈련 데이터를 5800장에서 9000장으로 늘렸음에도, 드라마틱한 성능향상은 되지 않았으며, 오히려 떨어진 부분도 있었음.

이슈 사항 및 해결 방안

1. 5800장 이상 학습 시 성능 향상 문제
2. 불법 주정차 기능 구현 관련 문제
3. ViewPoint Variation
4. Occlusion
5. Illumination Variation





추가적인 학습을 통한
성능 개선

교통량 계산 구현

불법 주정차 탐지 구현

웹 기능 개발

향후 일정

THANKS FOR WATCHING