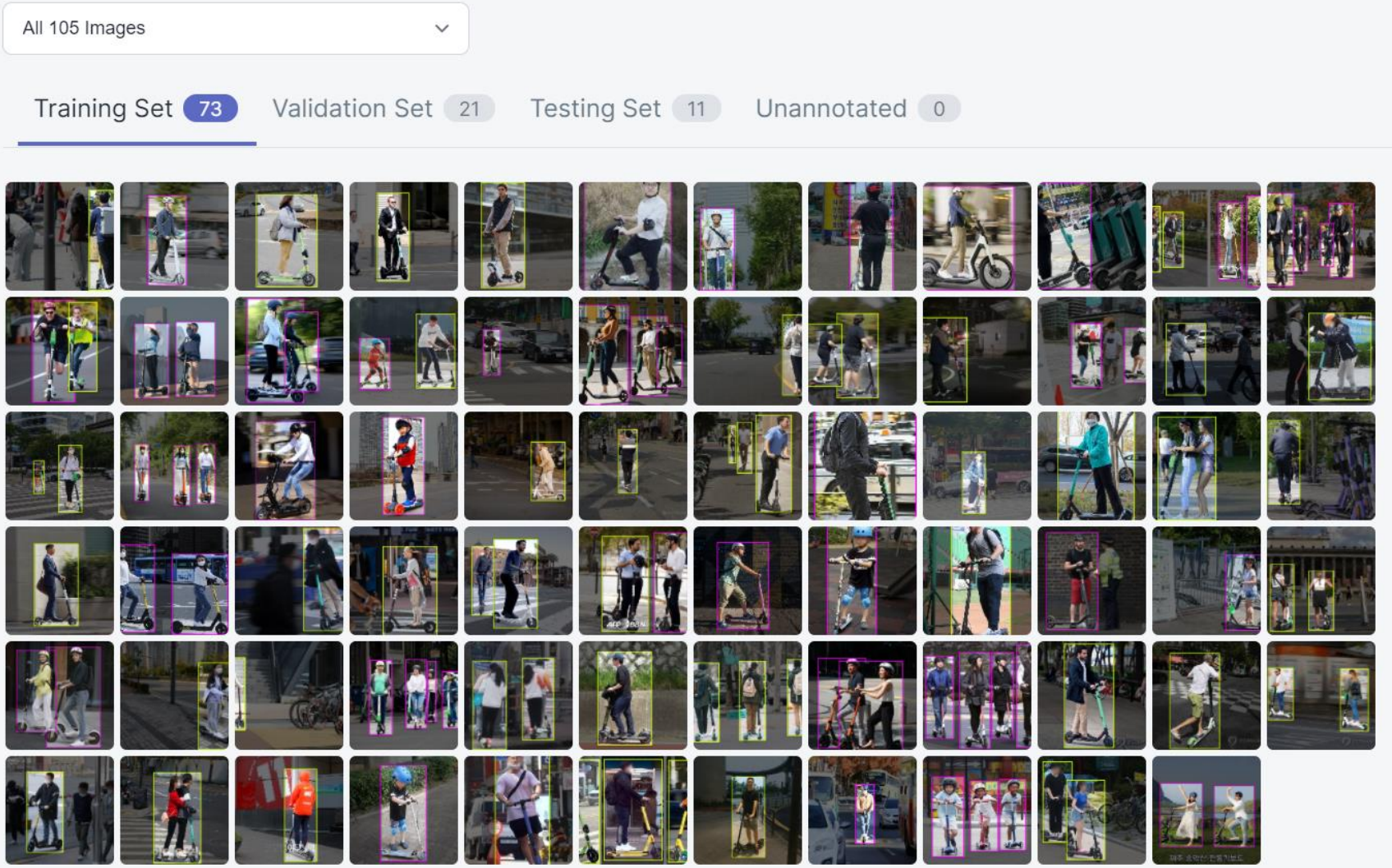
YOLO v5
Object Detection

Index

1.
Data

데이터 수집

How?	구글링
Amount?	105개
Split?	Train/Valid/Test



2.
Annotation

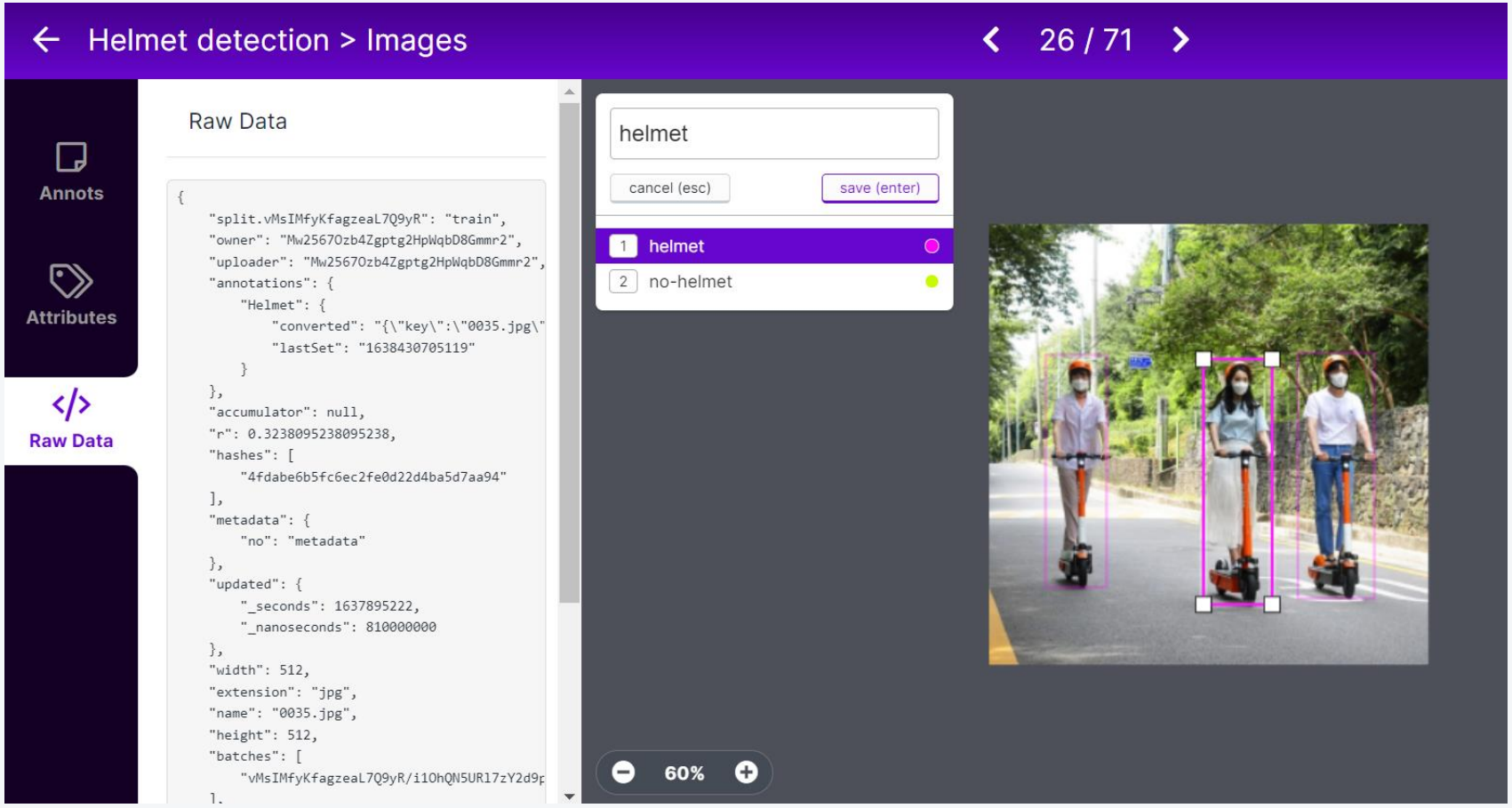
데이터 어노테이션

Tool? Roboflow

How? Bounding box

& Labeling

- helmet
- no-helmet



2.
Annotation

데이터 어노테이션

헬맷 미착용 객체 예시

- no-helmet

바운딩 박스가 하나인 경우

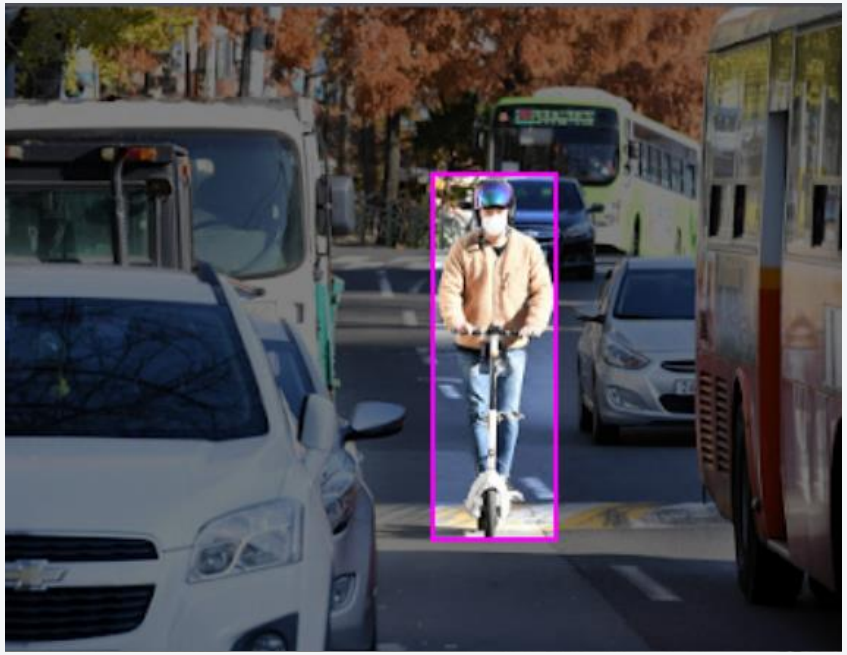
바운딩 박스가 여러 개인 경우

2.
Annotation

데이터 어노테이션

헬맷 착용 객체 예시

- helmet



바운딩 박스가 하나인 경우

바운딩 박스가 여러 개인 경우

3. Pre-Processing

데이터 전처리

Resize? 416 × 416
Rotation? ±30°
Shear? ±15° H,V

TRAIN / TEST SPLIT

Training Set 87%	Validation Set 8%	Testing Set 4%
219 images	21 images	11 images

Raw image = 73
3장씩 증폭하여 219개

PREPROCESSING

Auto-Orient: Applied

Resize: Stretch to 416×416

AUGMENTATIONS

Outputs per training example: 3

Rotation: Between -30° and +30°

Bounding Box: Shear: ±15° Horizontal, ±15° Vertical

전처리의 일종, 데이터에 인위적인 변화를 줌

4. Modeling

모델링

Version? YOLO v5

Release? 20.06.09

Epoch? 300

Weight? Yolov5s

```
1 !python train.py --img 416 --batch 16 --epochs 300 --data /content/datasets/Helmet-detection-3/data.yaml --weights yolov5s.pt --cache
```

```
     Epoch   gpu_mem       box       obj       cls    labels  img_size
   299/299     1.55G   0.01578     0.011  0.001689        50       416: 100% 14/14 [00:07<00:00,  1.87it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.84it/s]
                  all         21         31      0.664      0.721      0.715      0.457

300 epochs completed in 0.689 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.3MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model Summary: 213 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.07it/s]
                  all         21         31      0.609      0.789      0.746      0.464
               helmet         21         12      0.639          1      0.912      0.584
            no-helmet         21         19      0.578      0.579      0.579      0.344
Results saved to runs/train/exp
```

P = precision = 참이라고 예측했을 때, 실제로 참
R = recall = 실제로 참인 것을, 참이라고 예측

⬇

Validation data 기준 mAP* 75% (목표치)

특히, helmet에 대해 Recall 1, mAP 91%

* 클래스 1개당 AP (Average Precison)을 구하고 클래수의 mean 값을 구한 것

4.
Modeling

모델링

Version?	YOLO v5
Release?	20.06.09
Epoch?	300
Weight?	Yolov5s

```
!python train.py --img 416 --batch 16 --epochs 300 --data /content/datasets/Helmet-detection-3/data.yaml --weights yolov5s.pt --cache
```

```
     Epoch   gpu_mem       box       obj       cls    labels  img_size
   299/299     1.55G   0.01578     0.011  0.001689        50       416: 100% 14/14 [00:07<00:00,  1.87it/s]
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.84it/s]
                  all         21         31      0.664      0.721      0.715      0.457

300 epochs completed in 0.689 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.3MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model Summary: 213 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.07it/s]
                  all         21         31      0.609      0.789      0.746      0.464
               helmet         21         12      0.639          1      0.912      0.584
            no-helmet         21         19      0.578      0.579      0.579      0.344
Results saved to runs/train/exp
```

no-helmet 상대적 낮은 이유

음영

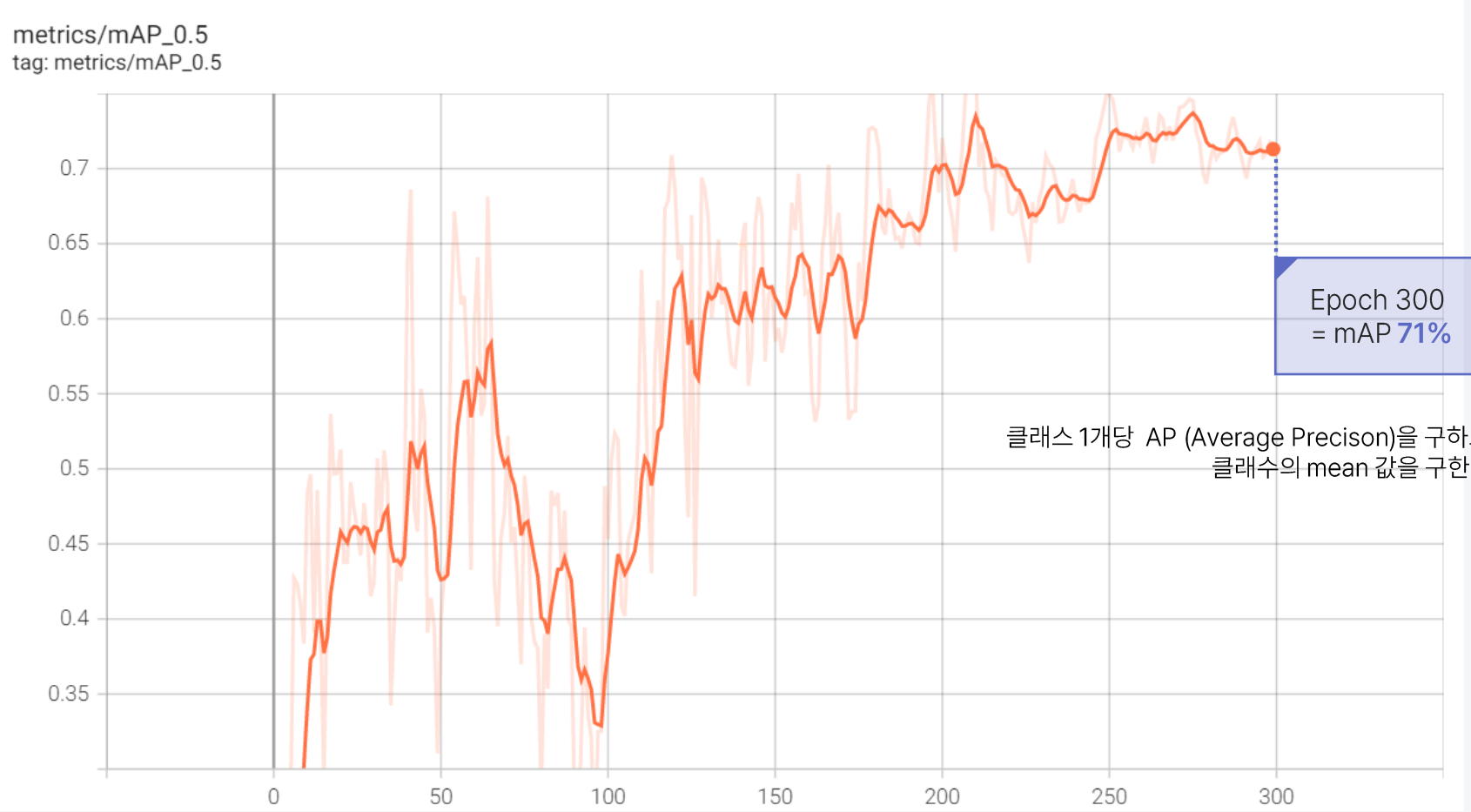
모자

4.
Modeling

모델링

모델 학습 결과

mAP 71%



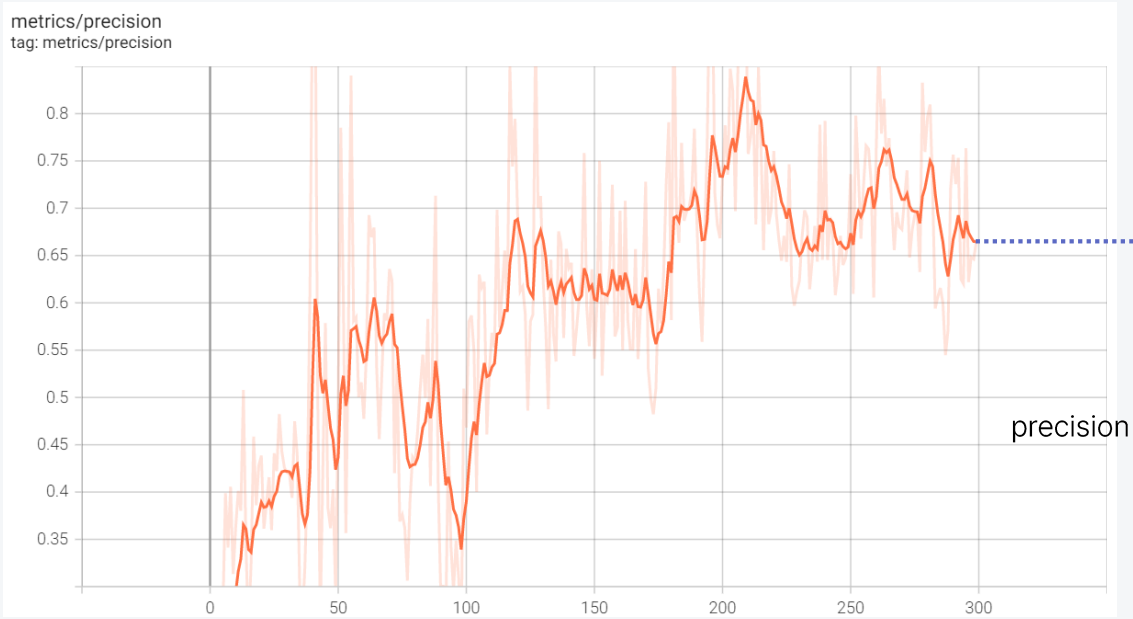
4.
Modeling

모델링

모델 학습 결과

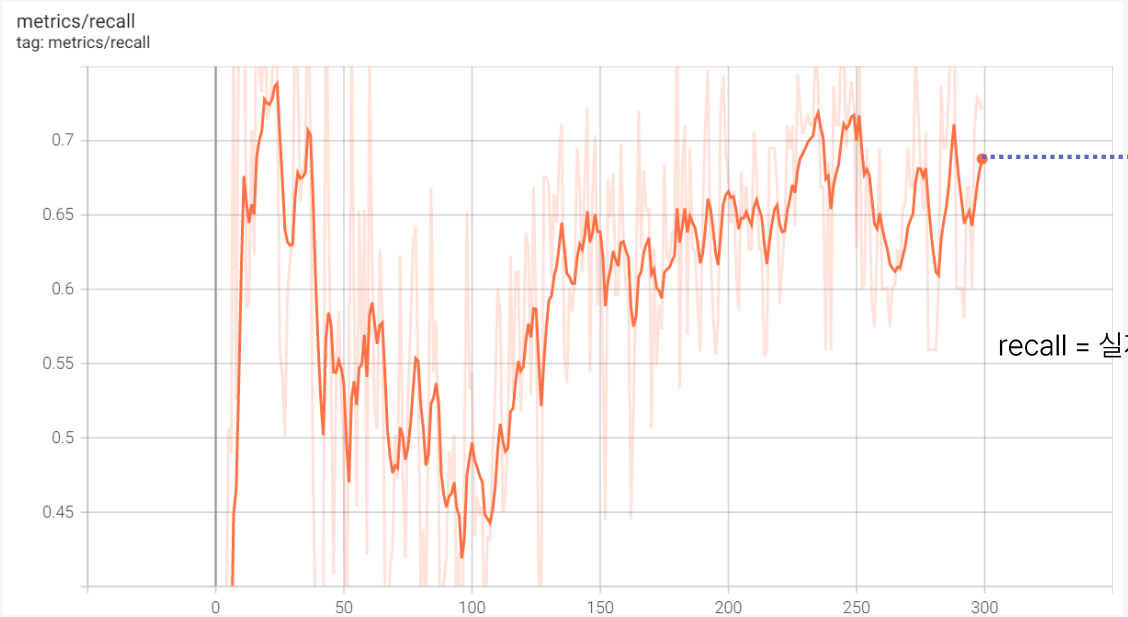
Precision 66%

Recall 72%



Epoch 300
= precision **66%**

precision = 참이라고 예측했을 때, 실제로 참



Epoch 300
= recall **72%**

recall = 실제로 참인 것을, 참이라고 예측

5.
Inference

추론 / 테스트

Good Case

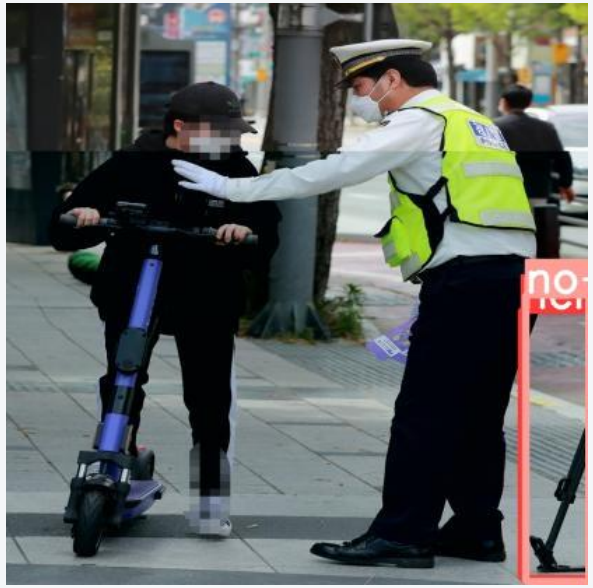
☺

5.
Inference

추론 / 테스트

Bad Case

☹

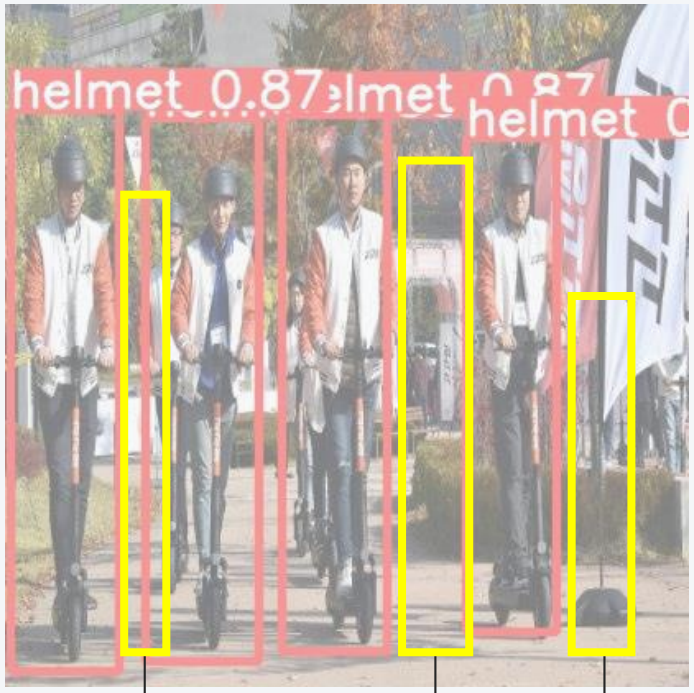


5.
Inference

추론 / 테스트

개선된 부분

Before



(1) 객체 사이 공백 인식

(2) 비인물 인식 (e.g., 깃대, 나무)

After (현재 최선)