

# Object Detection

## 1. 라이브러리별 최상위 모델 비교

라이브러리	저자/퍼블리셔	모델	AP(on COCO-val 2017)	AP(on COCO-test 2017)	Parameters(M)	기타	Official Link
Tensorflow(Tensorflow Hub 모델)	Tensorflow	EfficientDet-D7x	54.4	55.1	77.0	- <a href="#">EfficientDet의 하위 모델 보기</a> - Tensorflow Hub 에 등록된 모델이지만 publisher는 Tensorflow	<a href="https://github.com/google/automl/tree/master/efficientdet">https://github.com/google/automl/tree/master/efficientdet</a>
Tensorflow	Tensorflow	RetinaNet(with SpineNet-143)		50	67.0		<a href="https://github.com/tensorflow/models/blob/master/official/vision/README.md">https://github.com/tensorflow/models/blob/master/official/vision/README.md</a>
Pytorch	Pytorch	Faster R-CNN(with ResNet-50 FPN backbone)	46.7		43.7	- <a href="#">Faster R-CNN의 하위 모델 보기</a>	<a href="https://pytorch.org/vision/master/models/generated/torchvision.models.detection.fasterrcnn_fpn.html">https://pytorch.org/vision/master/models/generated/torchvision.models.detection.fasterrcnn_fpn.html</a>
Pytorch(Pytorch Hub 모델)	Ultralytics	YOLOv5x6	54.4	54.4	141.8	- <a href="#">YOLOv5의 하위 모델 보기</a>	<a href="https://pytorch.org/hub/ultralytics_yolov5/">https://pytorch.org/hub/ultralytics_yolov5/</a>

\* SOTA(State Of The Art) 모델 AP는 65.4입니다. 수치가 전반적으로 낮은 이유는 정확도를 테스트하는 데이터셋(COCO)의 클래스가 80개이기 때문입니다. 저희 프로젝트에서는 클래스가 1~2개일 것이기에 모델이 훨씬 높은 정확도를 기록할 것입니다.

\* 하위 모델은 최상위 모델보다 탐지 성능은 떨어지지만 크기가 작고 학습 및 추론 시간이 짧습니다.

## 2. 모델 선정

- 큰 문제가 없는 한 가장 탐지 성능이 좋은 `EfficientDet-D7x` 을 사용하겠습니다.