

# 물체 인식 성능

---

Youngtaek Hong, PhD


- Object detection 관련 논문을 읽다 보면 초기의 논문들은 대부분 성능에 “정확도” 지표를 사용하고 있는 것을 확인할 수 있습니다.
- Object Detection 뿐만 아니라 다양한 Task의 논문들을 살펴보면 대부분 연구 초기에는 주로 “정확도” 라는 지표를 올리기 위한 연구를 수행합니다. Object Detection에서는 이 “정확도” 라는 지표에 관한 표현

- 정확도의 계산은 주로 정답(Ground Truth, 이하 GT)와 모델이 예측한 결과(Prediction) 간의 비교를 통해 이루어집니다.
- Image Classification의 경우에는 GT가 이미지의 class인 반면, Object Detection은 이미지의 각 object의 해당하는 Bounding Box와 Box 안의 class를 의미합니다.
- 즉 정확도가 높다는 것은 모델이 GT와 유사한 Bounding Box를 예측(Regression)하면서 동시에 Box 안의 object의 class를 잘 예측(Classification)하는 것을 의미합니다. 즉 class도 정확하게 예측하면서, 동시에 object의 영역까지 잘 예측을 해야 합니다.

- 보통 Object Detection 논문에서 사용하는 정확도의 경우 Class를 예측하지 못하면 실패로 간주됩니다. Class를 올바르게 예측하였을 때의 Bounding Box의 정확도를 기준으로 정확도를 측정하게 됩니다.

# IoU (Intersection Over Union)

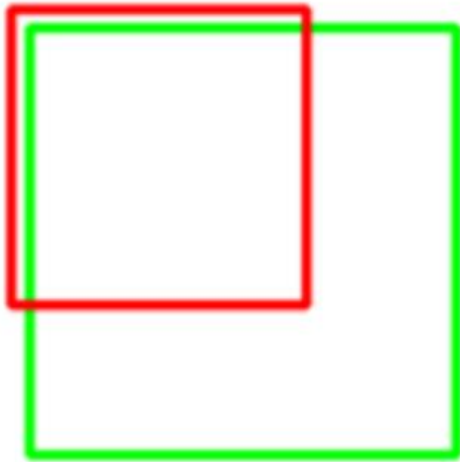
- Object Detection에서 Bounding Box를 얼마나 잘 예측하였는지는 IoU라는 지표를 통해 측정하게 됩니다.
- IoU(Intersection Over Union)는 Object Detection, Segmentation 등에서 자주 사용되며, 영어 뜻 자체로 이해를 하면 “교집합/합집합”이라는 뜻을 가지고 있습니다.
- 실제로 계산도 그러한 방식으로 이루어집니다. Object Detection의 경우 모델이 예측한 결과와 GT, 두 Box 간의 교집합과 합집합을 통해 IoU를 측정합니다.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


The diagram illustrates the calculation of Intersection over Union (IoU) for two overlapping squares. The top part shows two squares: one with a blue outline and one with a solid blue fill. The overlapping region is shaded in a darker blue. The bottom part shows the union of the two squares, represented by a single solid blue shape that covers the area of both original squares.

[그림 1. IoU 계산 예시]

IoU: 0.4034



**Poor**

IoU: 0.7330



**Good**

IoU: 0.9264



**Excellent**

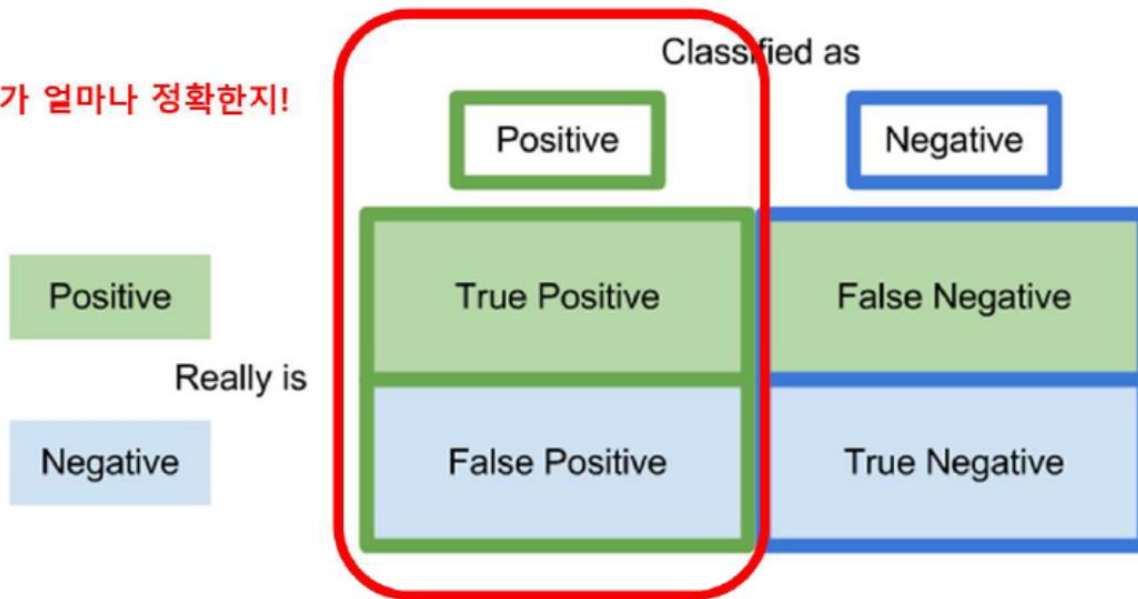
[그림 2. Object Detection 예측 결과에 따른 IoU 값 예시]

# Precision

Precision은 주로 Recall과 사용되며, Image Classification을 공부하시면서 다들 한번쯤은 보셨을 지표입니다.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

검출 결과가 얼마나 정확한지!

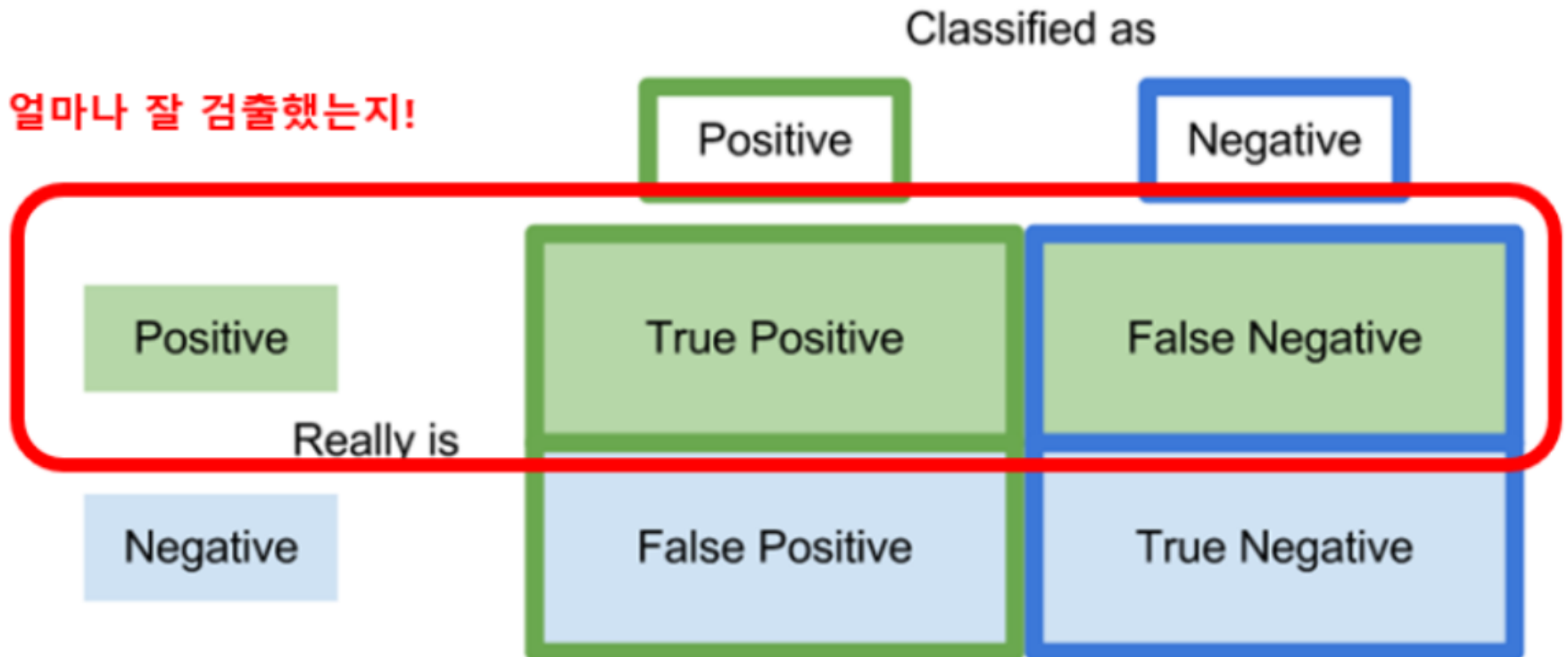


[그림 3. Precision 계산 방법]



# Recall

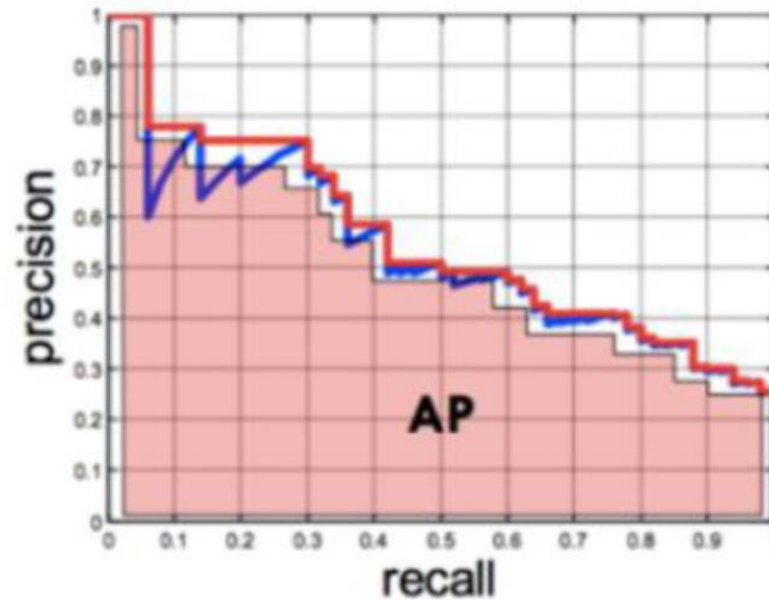
$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$



[그림 4. Recall 계산 방법]

## AP (Average Precision), mAP (mean Average Precision)

앞서 말씀드린 것처럼 Precision과 Recall은 반비례 관계를 갖기 때문에 Object Detection에서는 Average Precision, 이하 **AP** 라는 지표를 주로 사용합니다.



[그림 5. Average Precision 계산 방법]

## performance table

Detector	VOC07 (mAP@IoU=0.5)	VOC12 (mAP@IoU=0.5)	COCO (mAP)	Published In
R-CNN	58.5	-	-	CVPR'14
SPP-Net	59.2	-	-	ECCV'14
MR-CNN	78.2 (07+12)	73.9 (07+12)	-	ICCV'15
Fast R-CNN	70.0 (07+12)	68.4 (07++12)	-	ICCV'15
Faster R-CNN	73.2 (07+12)	70.4 (07++12)	-	NIPS'15
YOLO v1	66.4 (07+12)	57.9 (07++12)	-	CVPR'16
G-CNN	66.8	66.4 (07+12)	-	CVPR'16
AZNet	70.4	-	22.3	CVPR'16
ION	80.1	77.9	33.1	CVPR'16
HyperNet	76.3 (07+12)	71.4 (07++12)	-	CVPR'16
OHEM	78.9 (07+12)	76.3 (07++12)	22.4	CVPR'16
MPN	-	-	33.2	BMVC'16
SSD	76.8 (07+12)	74.9 (07++12)	-	ECCV'16
GBDNet	77.2 (07+12)	-	27.0	ECCV'16
CPF	76.4 (07+12)	72.6 (07++12)	-	ECCV'16
R-FCN	79.5 (07+12)	77.6 (07++12)	29.9	NIPS'16
DeepID-Net	69.0	-	-	PAMI'16

NoC	71.6 (07+12)	68.8 (07+12)	27.2	TPAMI'16
DSSD	81.5 (07+12)	80.0 (07++12)	-	arXiv'17
TDM	-	-	37.3	CVPR'17
FPN	-	-	36.2	CVPR'17
YOLO v2	78.6 (07+12)	73.4 (07++12)	-	CVPR'17
RON	77.6 (07+12)	75.4 (07++12)	-	CVPR'17
DeNet	77.1 (07+12)	73.9 (07++12)	33.8	ICCV'17
CoupleNet	82.7 (07+12)	80.4 (07++12)	34.4	ICCV'17
RetinaNet	-	-	39.1	ICCV'17
DSOD	77.7 (07+12)	76.3 (07++12)	-	ICCV'17
SMN	70.0	-	-	ICCV'17
YOLO v3	-	-	33.0	arXiv'18
SIN	76.0 (07+12)	73.1 (07++12)	23.2	CVPR'18
STDN	80.9 (07+12)	-	-	CVPR'18
RefineDet	<b>83.8 (07+12)</b>	<b>83.5 (07++12)</b>	41.8	CVPR'18
SNIP	-	-	<b>45.7</b>	CVPR'18
Relation-Network	-	-	32.5	CVPR'18
Cascade R-CNN	-	-	42.8	CVPR'18
MLKP	80.6 (07+12)	77.2 (07++12)	28.6	CVPR'18
RFBNet	82.2 (07+12)	-	-	ECCV'18
CornerNet	-	-	42.1	ECCV'18
Pelee	76.4 (07+12)	-	-	NIPS'18
HKRM	78.8 (07+12)	-	37.8	NIPS'18
M2Det	-	-	44.2	AAAI'19