

Ambient AI Bootcamp:

Practice 4

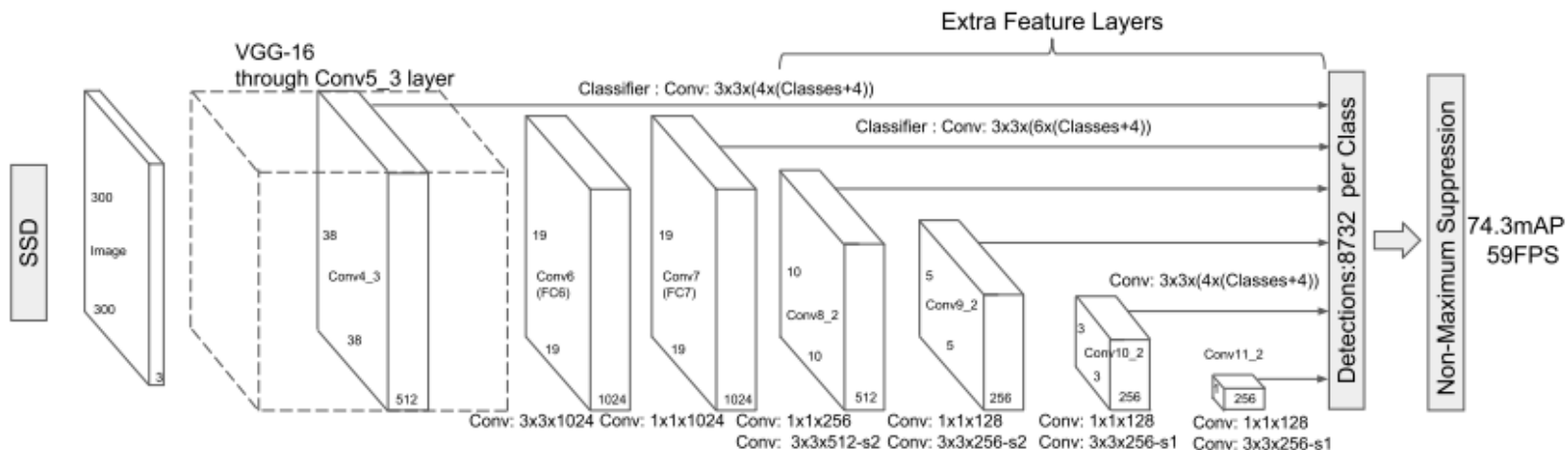


SNU Graduate School of Data Science

Object detection model 실습: MobileNetSSD

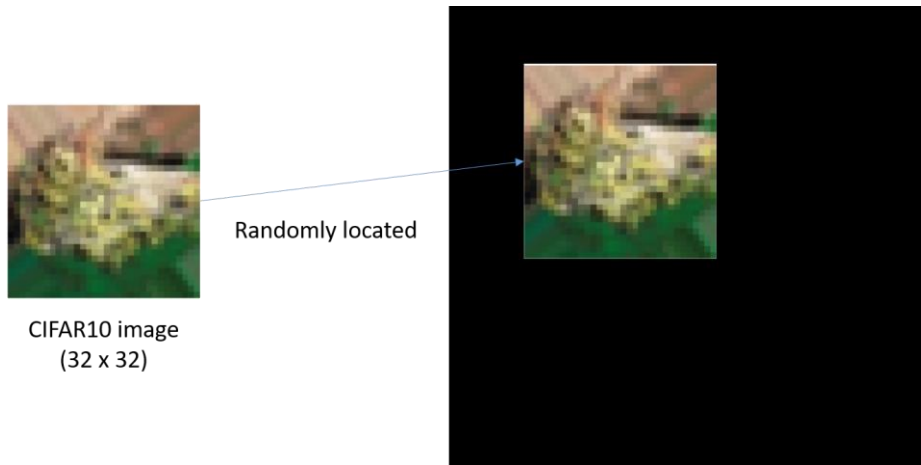
Recap: Single-shot detection model

- One-stage model: Region of Interest를 찾는 과정 없이 한번에 객체 탐색 수행
- 다양한 Grid size에서, 여러 종류의 기본 Anchor box를 만들고 이를 조정해가며 Detection 수행



Prepare data

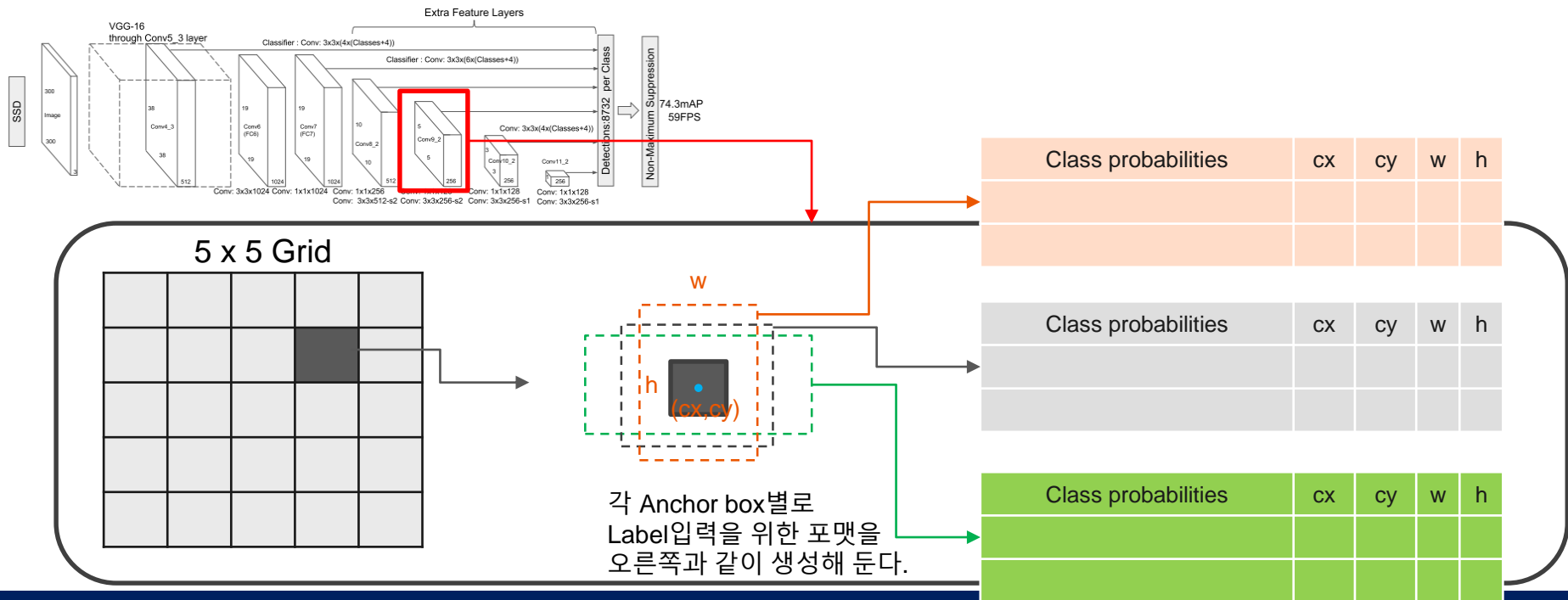
- 실습 목적으로 데이터를 인위적으로 만들어서 사용.
- CIFAR-10 image를 64x64로 확대한 후 224 x 224 이미지에 랜덤하게 배치.



- Label은 어떻게 만들어야 할까?
 - 2종류의 Label이 필요: Class, Box location
 - Model의 Output과 같은 형태로 준비해 두어야 함.

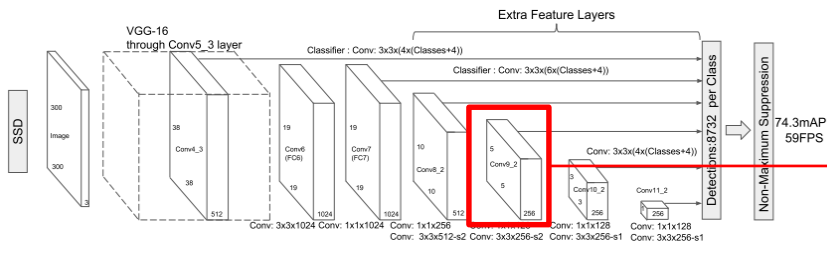
SSD ground truth label

- Model에서 Output을 어떻게 생성하는지 먼저 파악하고 Output과 동일한 형태로 GT Label 생성
- 각 Detection layer마다 Grid로 나누어지고, 각 Grid는 여러 종류의 Anchor box를 생성함



SSD ground truth label

- Image와 GT Box가 주어졌을 때 Anchor Box 중 매칭이 많이 되는 Box들을 찾는다.(예: IoU가 0.3 이상)



5 x 5 Grid



해당 Anchor box가 GT box와 많이 겹친다고 가정

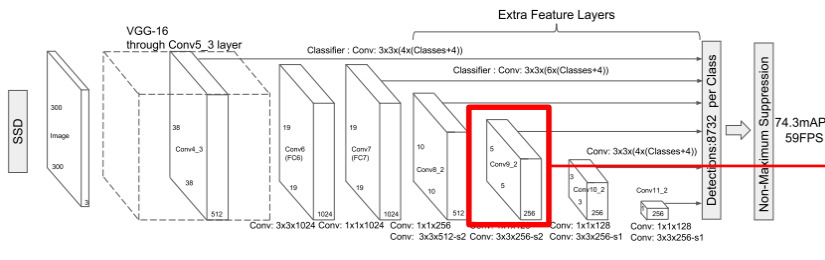
| Background | Class 1 | Class 2 | ... | Class N | cx | cy | w | h |
|------------|---------|---------|-----|---------|----|----|---|---|
| 0 | 0 | 1 | | 0 | | | | |
| 1 | 0 | 0 | | 0 | | | | |

매칭되는 Anchor box의 해당되는 Class label을 1로 부여한다.

매칭되지 않는 Anchor box는 Background class label을 1로 부여한다.

SSD ground truth label

- Image와 GT Box가 주어졌을 때 Anchor Box 중 매칭이 많이 되는 Box들을 찾는다.(예: IoU가 0.5 이상)



5 x 5 Grid



Box의 중심과 가로, 세로 Label은 아래와 같이 부여한다.

Class probabilities

0 0 0 1 0 0 0 0

cx

$$\widehat{g}_j^{cx} = \frac{g_j^{cx} - d_i^{cx}}{d_i^w}$$

cy

$$\widehat{g}_j^{cy} = \frac{g_j^{cy} - d_i^{cy}}{d_i^h}$$

w

$$\widehat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right)$$

h

$$\widehat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right)$$

(g_j^{cx}, g_j^{cy}) : GT Box의 중심

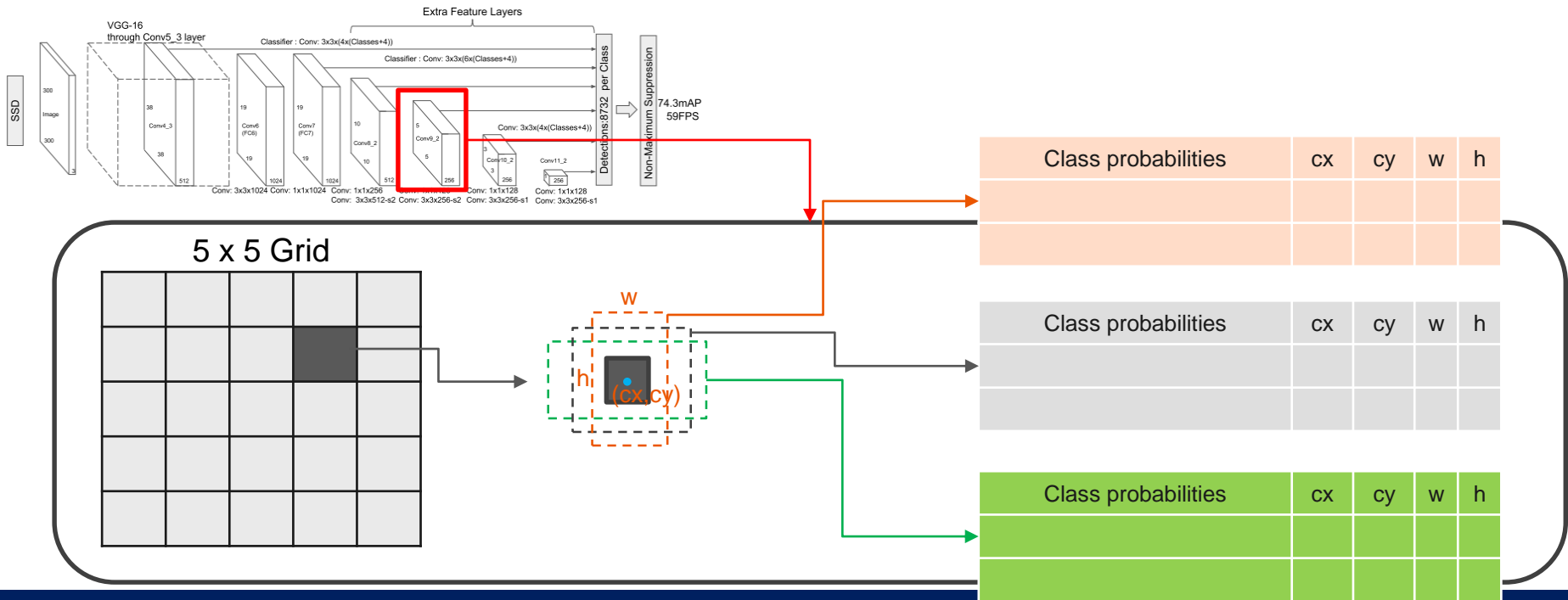
(d_j^{cx}, d_j^{cy}) : Anchor Box의 중심

(g_j^w, g_j^h) : GT Box의 가로, 세로

(d_j^w, d_j^h) : Anchor Box의 가로, 세로

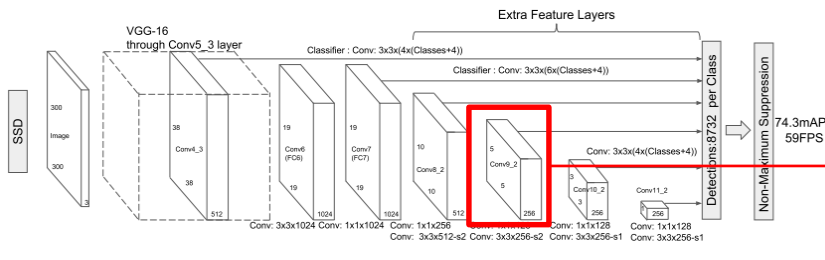
SSD Model

- 각 Detection layer마다 Grid로 나누어지고, 각 Grid는 여러 종류의 Anchor box를 생성함
- Model은 각 Anchor box로부터 조정해 가며 Prediction box를 생성

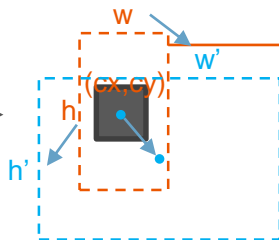
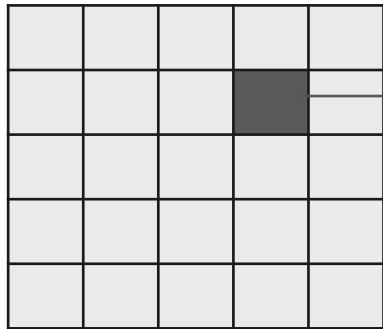


SSD Model

- 각 Detection layer마다 Grid로 나누어지고, 각 Grid는 여러 종류의 Anchor box를 생성함
- Model은 각 Anchor box로부터 조정해 가며 Prediction box를 생성



5 x 5 Grid



Class probabilities

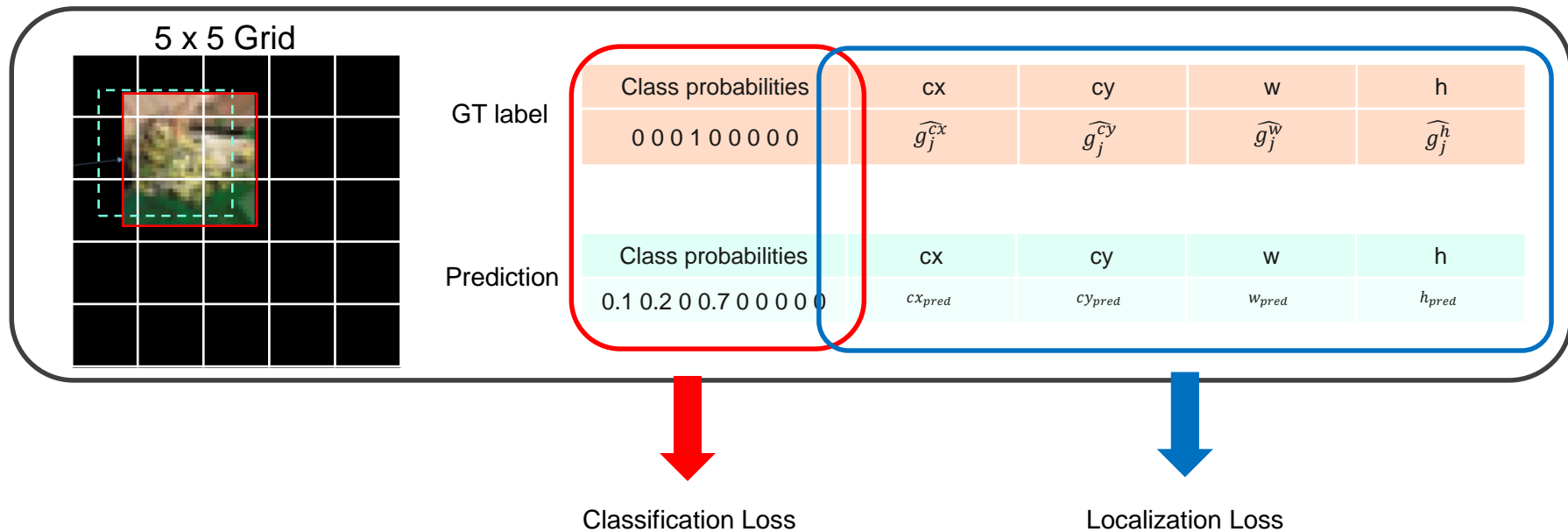
| Cx_{pred} | Cy_{pred} | w_{pred} | h_{pred} |
|-------------|-------------|------------|------------|
| | | | |
| | | | |
| | | | |

모델이 예측하는 값은 각 Anchor box 대비 조정의 크기임. 최종 Box는 모델이 예측하는 값과 각 Anchor box의 크기를 이용하여 계산됨. 예) $w' = cx_{pred} * w + cx$

(GT label 생성시 입력했던 값과 연결지어서 생각해 보면 좋음)

SSD Model Training

- 각 Anchor box별로 앞에서 생성한 GT label과 Prediction하는 값이 같아지도록 트레이닝



SSD Model Loss function

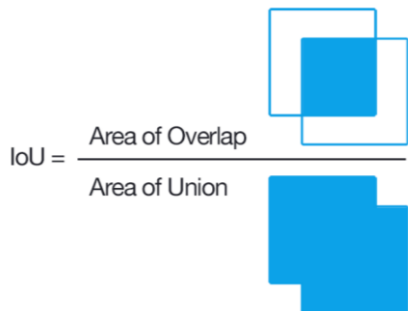
- Loss = Classification loss + Localization loss
- Classification loss
 - Positive: Anchor box 중 GT label과 매칭이 된 Box (해당 Class로 Prediction해야 함)
 - Negative: GT label과 매칭이 되지 않은 Box (Background Class로 Prediction해야 함)
 - Negative box가 Positive에 비해 훨씬 많으므로 대부분 Negative box로 예측하게 되는 경향이 있음
 - 따라서, Negative의 갯수가 Positive의 일정 비율 이하가 되도록 조절함 (저자들은 Neg:Pos = 3:1로 설정함)
- Localization loss
 - 논문에서는 Positive box에 대하여 SmoothL1 Loss (Huber loss)를 이용하였음

$$Loss_{\delta}(y, f(x)) = \begin{cases} \frac{1}{2}((y_i - f(x_i))^2 & \text{for } |y_i - f(x_i)| \leq \delta, \\ \delta |y_i - f(x_i)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

- 우리는 L2 Loss를 이용하여 구현해 볼 예정임

Object detection prediction metric: mAP

- Object detection에서 가장 많이 활용하는 지표는 mAP이며, mean Average Precision의 약자이다.
- IoU (Intersection over Union): 얼마나 Box prediction이 잘 되었는지 나타내는 지표이다.

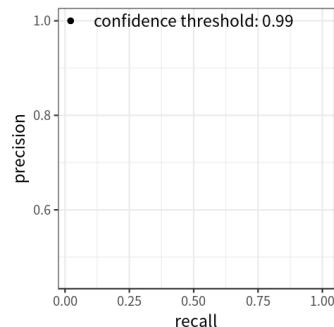


[Image source: Mean Average Precision \(mAP\) Explained: Everything You Need to Know \(v7labs.com\)](#)

- True Positive / True Negative / False Positive / False Negative
 - True Positive: 모델에서 Object 로 예측하였고, GT Label도 Object 인 경우
 - False Positive: 모델에서 Object 로 예측하였으나, GT Label은 Background인 경우
 - False Negative: 모델에서 Background 로 예측하였으나, GT Label은 Object 인 경우
 - True Negative: 모델에서 Background로 예측하였고, GT Label도 Background 인 경우
- Precision = $TP / (TP + FP)$
- Recall = $TP / (TP + FN)$

Object detection prediction metric: mAP

- 모델이 Object로 예측했다는 것은 어떻게 정할 것인가? (Positives)
 - 2개의 Threshold를 이용
 - **Class Threshold**: 해당 Class로 예측한 Confidence score가 일정 수준 이상이어야 한다.
 - **IoU Threshold**: 예측한 Box와 GT Box가 일정 넓이 이상 겹쳐야 한다.
- IoU Threshold를 고정한 상태에서 Class Threshold를 변경해가다 보면 Precision / Recall이 변하게 됨



- mean Average Precision
 - AP_i : Class i 에 대해 Precision-Recall Curve 아래 영역의 넓이
 - $mAP = \frac{1}{K} \sum_{i=1}^K AP_i$: 모든 Class에 대한 AP_i 의 평균
- IoU Threshold에 따라 다른 mAP가 나오게 됨
 - COCO에서는 IoU Threshold를 0.5 ~ 0.95까지 0.05씩 변경해가며 mAP를 구하고, 이를 평균한 것을 평가 Metric으로 사용