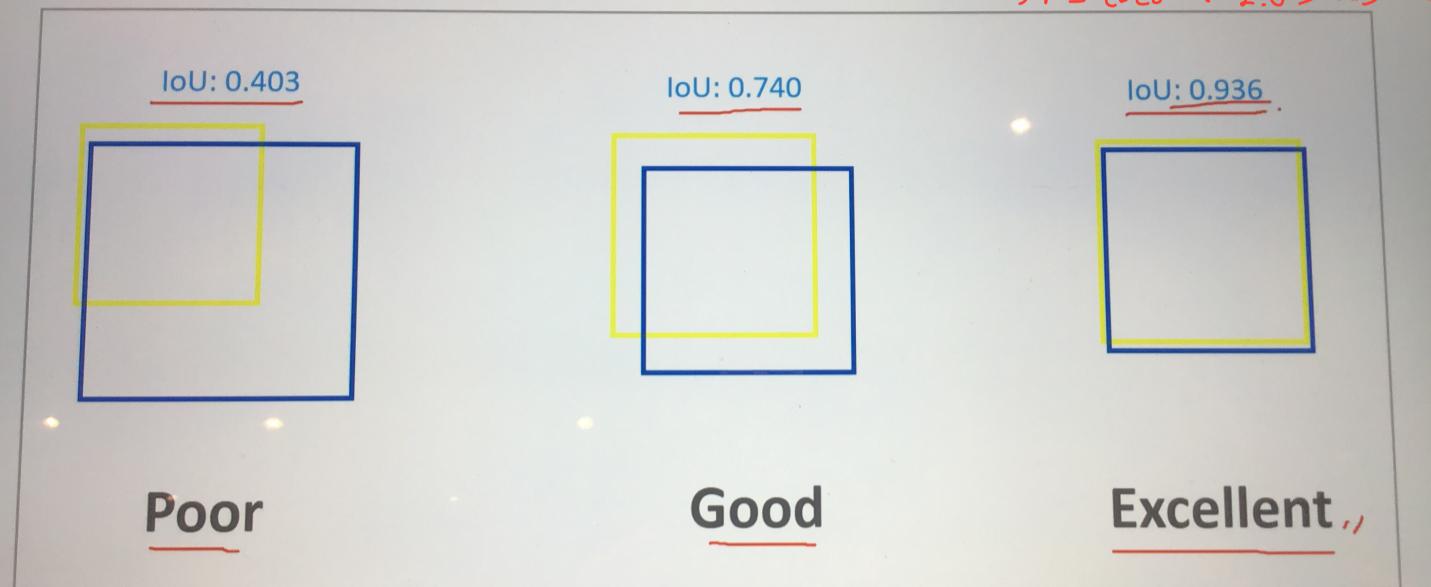


IoU에 따른 Detection 성능 \Rightarrow 절대적인 기준은 X

ex) Competition 별 1) pascal VOC : IoU > 0.5
2) MS Coco : IoU > 0.5 \rightarrow 0.6 \rightarrow ... 점차 높임.



파이썬 딥러닝 컴퓨터 비전

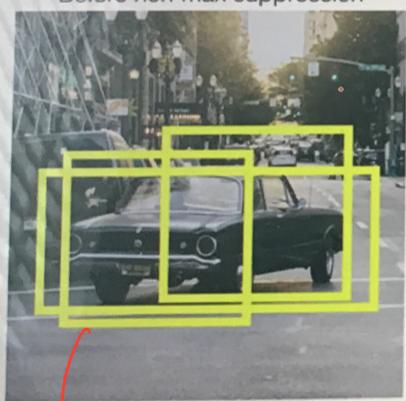
MacBook Pro

NMS(Non Max Suppression) \Rightarrow 가장 확실한 bounding box만

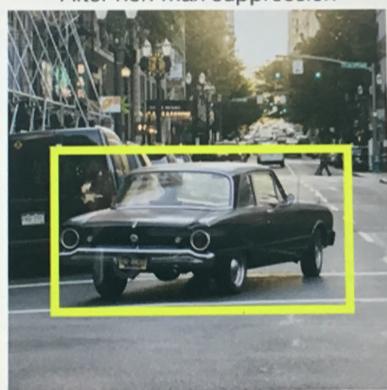
* Object detection 알고리즘 : 정확하게 Object가 있는 곳을 번호주는게 아니,
있을 만한 곳 다수를 번호는

남겨주는 기법!

→ NMS 이전, 그럴듯한 bounding box 다수를 잡아줌
Before non-max suppression



Non-Max Suppression
→



* Non Max Suppression (NMS)
: Max가 아닌 값들은 '남겨주는' 기법
(제거X)

- Object Detection 알고리즘은 Object가 있을 만한 위치에 많은 Detection을 수행하는 경향이 강함.
- NMS는 Detected 된 Object의 Bounding box 중에 비슷한 위치에 있는 box를 제거하고 가장 적합한 box를 선택하는 기법

'차'와 비슷한 것과 보이는 Region을 반환

파이썬 딥러닝 컴퓨터 비전

MacBook Pro

NMS 수행 로직

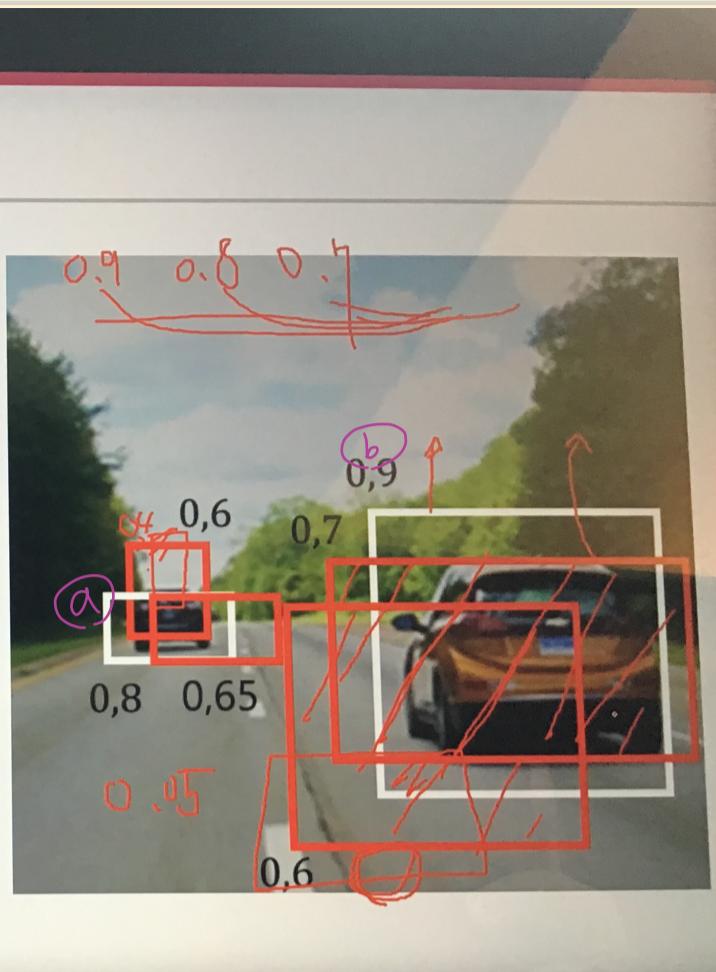
$$\left(\frac{CS}{I} \right) = 0.5 \text{ 이하}$$

$$\left(\frac{I}{I} \right) = 0.4$$

- Detected 된 bounding box 별로 특정 Confidence threshold 이하 bounding box는 먼저 제거(confidence score < 0.5)
- 가장 높은 confidence score를 가진 box 순으로 내림차순 정렬하고 아래 로직을 모든 box에 순차적으로 적용.
 - 높은 confidence score를 가진 box와 겹치는 다른 box를 모두 조사하여 IOU가 특정 threshold 이상인 box를 모두 제거(예: IOU Threshold > 0.4) ↳ 비슷한 것 제거이니까.
- 남아 있는 box만 선택

Confidence score가 높을 수록 ↑
 IOU Threshold가 낮을 수록 많은 Box가 제거됨. ↓

파이썬 딥러닝 컴퓨터 비전



* NMS
 'Max' 이건 아지만,

Confidence Score가 MAX인 것만 선택하는 건 아니다.

①과 ②는 각각 Confidence Score가 0.9, 0.8이지만, 0.9는 냉기지 않고, 내림차순으로 IOU 기준을 적용해 필터링 (다중 object의 detection 시, 클래스 구분 위해)

Object Detection 성능 평가 Metric - mAP

실제 Object가 Detected된 재현율(Recall)의 변화에 따른 정밀도(Precision)의 값을 평균한 성능 수치

- IOU
- Precision-Recall Curve, Average Precision
- Confidence threshold

⇒ softmax 값이 'confidence threshold' 이상이면, 'object'라고 인식하는 기준

