

# You Only Look Once: Unified, Real-Time Object Detection

(Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi)

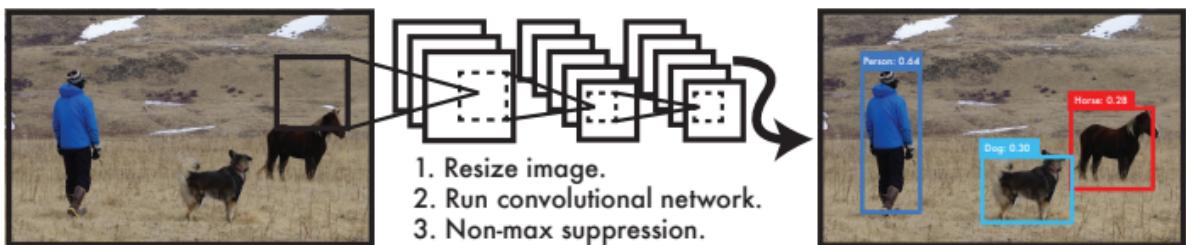
## INTRODUCTION

### 기존 object detection vs. YOLO

- 기존 object detection
  - 어떤 object를 detection하기 위해서는 그 object의 classifier를 테스트 이미지의 다양한 location에서 검증해야 함
    - e.g., DPM (deformable parts model) - sliding window approach, R-CNN - region proposal
  - 문제점: 느림 -> 복잡한 pipeline, 최적화하기 어려움 -> 각 요소가 따로 training되어야 함
- YOLO
  - Object detection을 single regression 문제로 바꿈 -> 이미지에서 바로 bbox coordinates와 class 확률을 제시
  - => 즉, 이미지를 한번만 보고 어떤 objects가 어디에 있는지 알 수 있음

### YOLO 소개

- Convolutional network 1개가 여러개의 bounding box들과 클래스 확률을 동시에 예측



### YOLO 장단점

- 장점
  - 1. 아주아주 빠르다
  - 2. 이미지 전체에 대한 예측 진행
  - 3. object에 대해 일반화 된 representation 학습
- 단점: 다른 성능이 좋은 detection system에 비해 아직 정확도가 낮음
  - BUT! 빠르고 정확하게 사물들을 이미지에서 detection 가능.

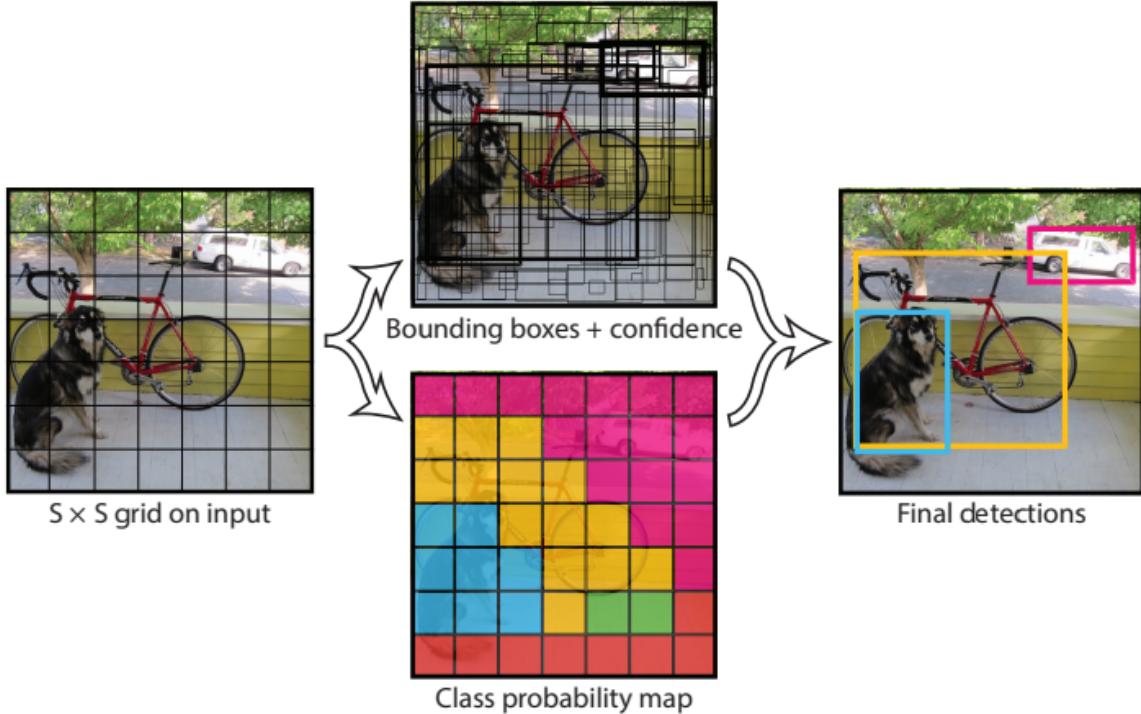
## 그래서 YOLO가 어떻게 작동하는데?

### Unified Detection

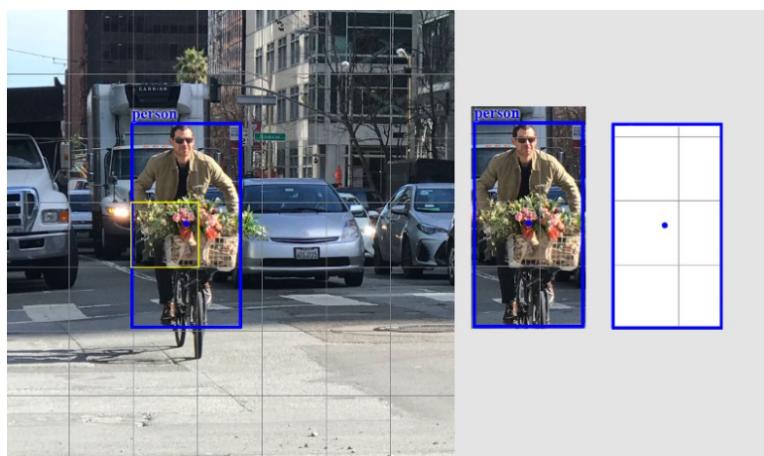
- Object detection에서 각각의 요소를 하나의 신경망에 통일시킴

- 전체 이미지의 feature들을 각 bounding box를 예측하는 데에 사용
- && 모든 클래스에 대한 bounding box를 예측
- => end-to-end training & real-time & high average precision (느디어!!!)

## Model

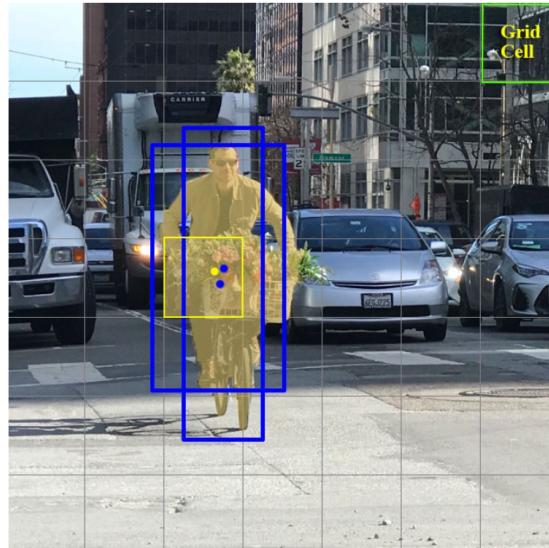


- 우선, input image가 시스템에 들어오면  $S \times S$  grid로 나눔  $\Rightarrow$  각 grid cell은 object 1개만 예측함
  - 만약 어떤 object의 중심이 grid cell에 들어갔다면, 그 grid cell을 이용해서 해당 object를 발견할 수 있음



- 각 grid cell은  $B$  bounding box와 각 박스의 confidence score를 예측  $\Rightarrow$  confidence score를 통해 모델이 box 안에 object가 있고 + 얼마나 정확하게 그 object를 예측할 수 있는지 표현

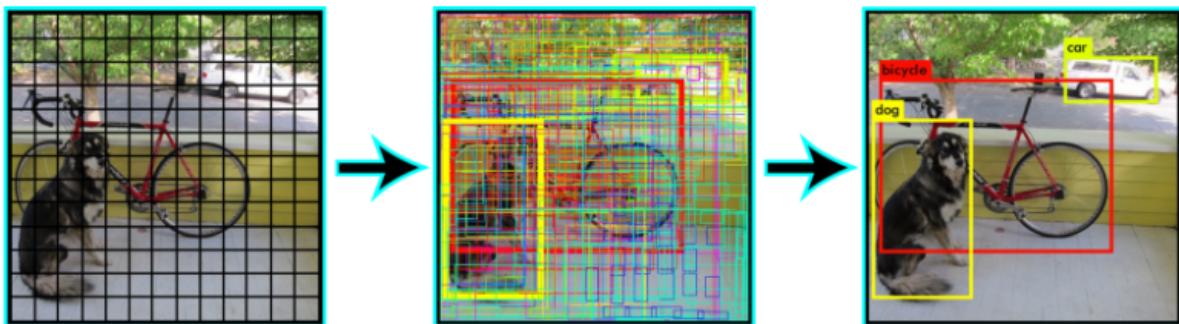
- confidence score =  $\Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$
- confidence score = 0  $\Rightarrow$  grid cell에 object가 한개도 없다!



- 각 grid cell은 C conditional class probability를 예측
  - $P(\text{Class}_i | \text{Object})$
  - bounding box의 개수와 상관없이 각 grid cell별로 class probability를 계산
- 테스트 시 conditional class probability와 각 box의 confidence prediction을 곱해줌 => 각 box의 class-specific confidence score 갖!
  - 의미: 어떤 box에 어떤 클래스가 나올 확률과 예측한 box가 그 object에 fit되는 확률

$$\Pr(\text{Class}_i | \text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}} \quad (1)$$

- 정리:
  - bbox 정보에서 물체가 있을 확률(confidence score) 회귀
  - conditional class probability를 통해 물체가 있다고 가정했을 때 어떤 클래스일지 확률 회귀
  - 최종적으로 그 grid에 class가 있을 확률(=confidence score \* conditional class probability)를 계산

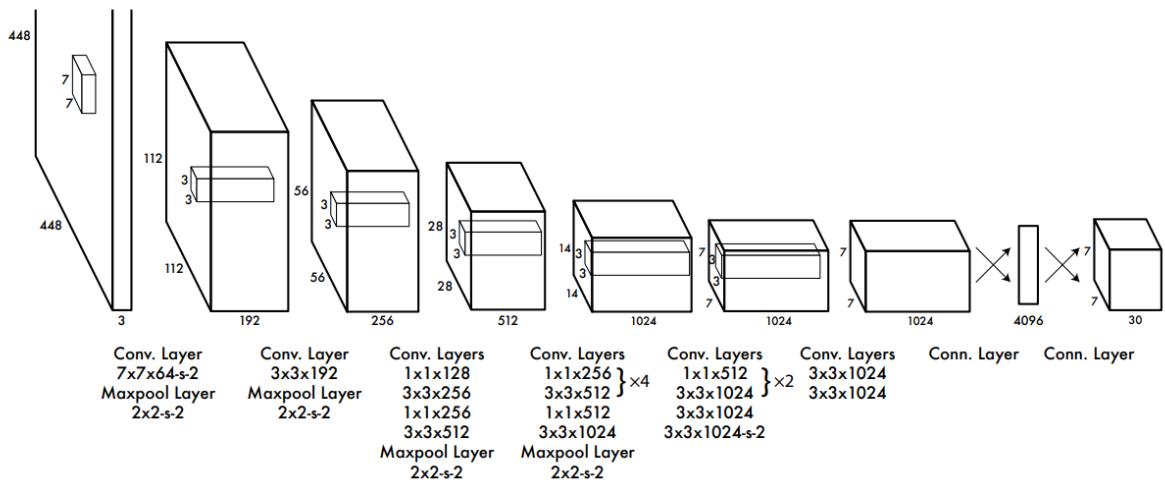


- 예측값들은  $S \times S \times (B * 5 + C)$  tensor에 인코딩됨
  - 논문:  $(7, 7, 2*5+20) \Rightarrow (7, 7, 30)$

### Architecture / Network Design (Figure 3)

- GoogLeNet에서 영감 받은 구조
- 24 convolutional layers + 2 fully connected layers
- 첫 convolutional layer들을 통해 이미지의 feature들을 추출

- fully connected layer들을 통해 output 확률과 좌표를 예측



## YOLO의 한계

- object들이 너무 가깝게 있으면 detection이 잘 안 됨 \*\*grid cell당 object 1개씩 예측하기 때문

