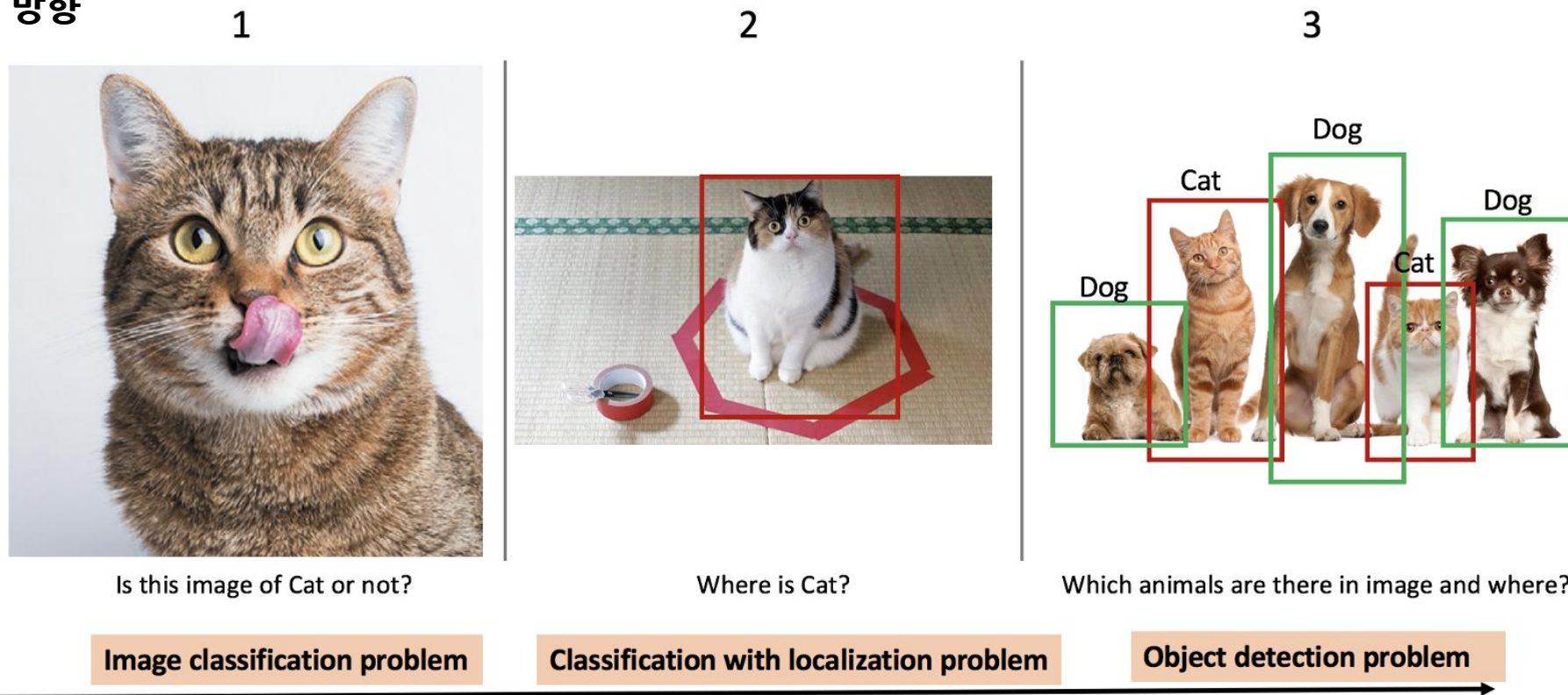

Object detection/recognition, Face detection/recognition

CONTENTS

- 1 CNN을 활용한 객체 인식 모델
- 2 객체 인식 모델의 발전 방향
- 3 R-CNN 모델 소개
- 4 YOLO 모델 소개

01 CNN을 활용한 객체 인식 모델

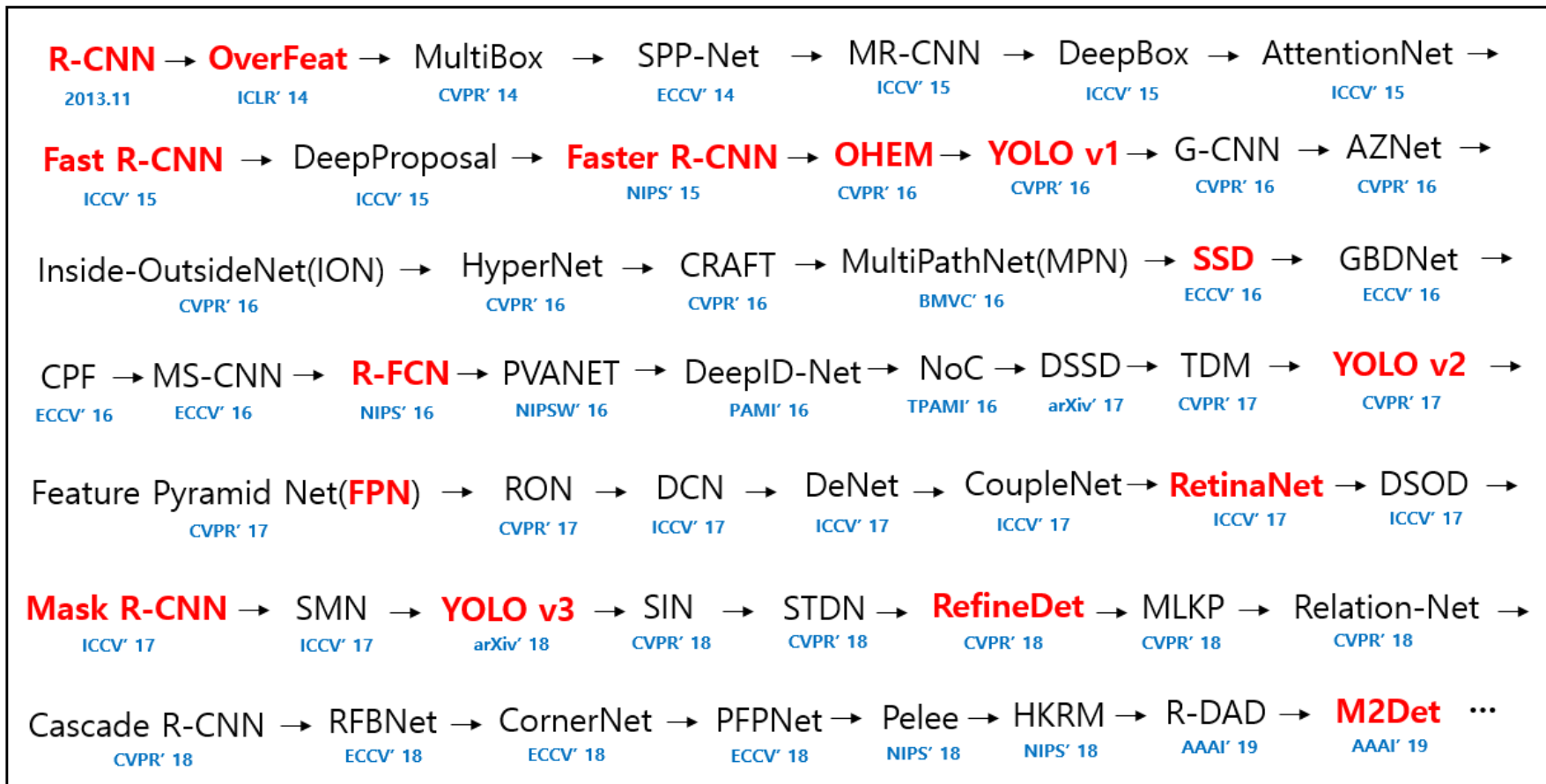
CNN 의 발전 방향



객체 인식 모델 : 객체의 존재 유무(Object Detection)를 판단할 뿐만 아니라 객체가 어떤 것인지 구분 (Recognition)

02 객체 인식 모델의 발전 방향

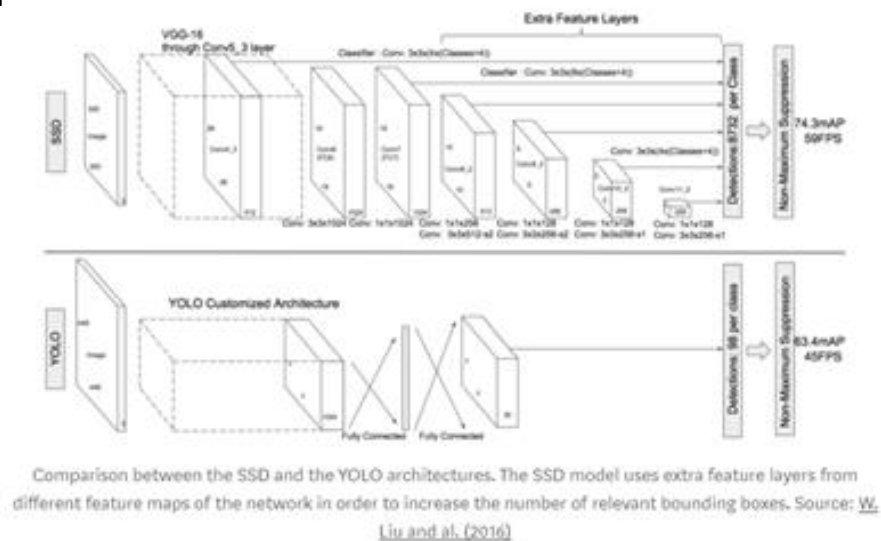
객체 인식 모델에 관한 논문



02 객체 인식 모델의 발전 방향

객체 인식 모델의 분류

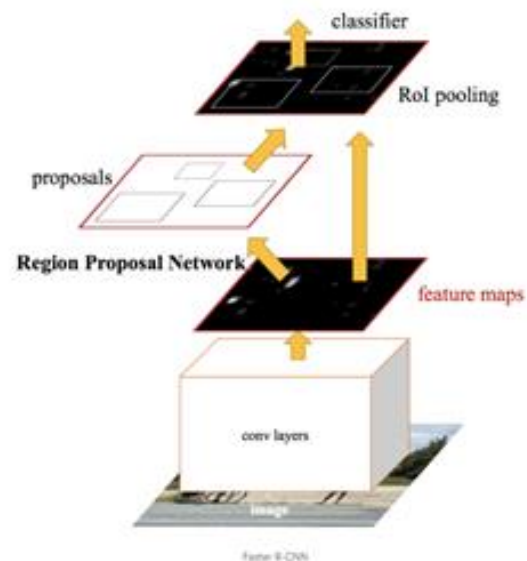
1



(1) one stage method

YOLO / SSD / RetinaNet

2



(2) two stage method

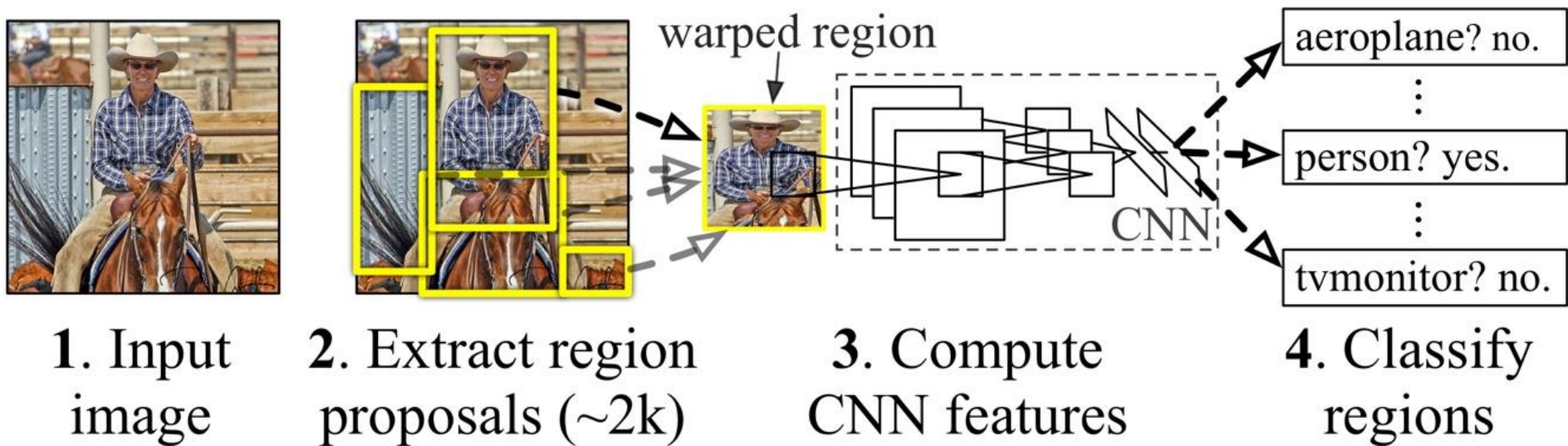
R-CNN / Fast R-CNN / Faster R-CNN

03 객체 인식 모델 R-CNN

특징

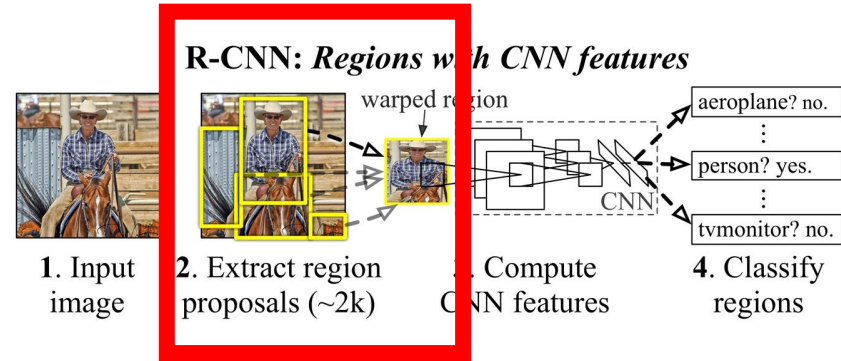
- 딥러닝을 활용한 첫 이미지 객체인식 모델
- 2 stage detector

R-CNN: *Regions with CNN features*



03 객체 인식 모델 R-CNN

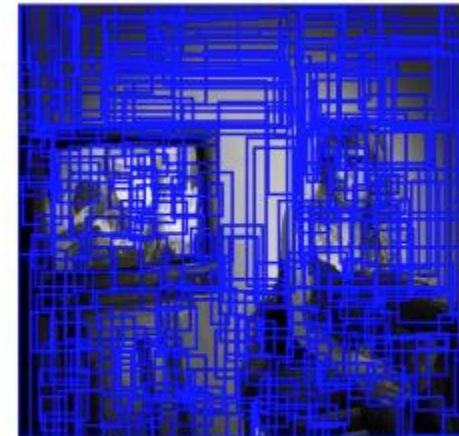
Selective Search 알고리즘



Input Image



Segmentation

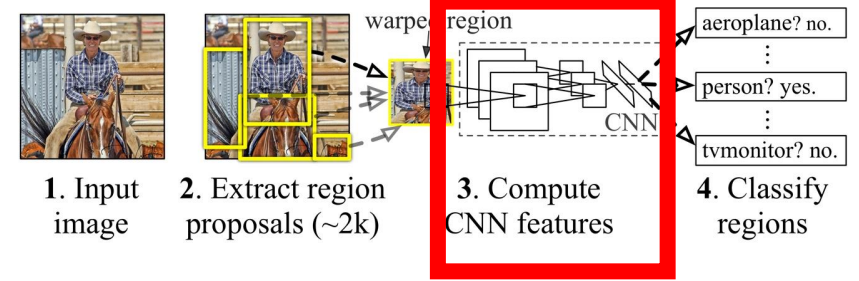


Candidate objects

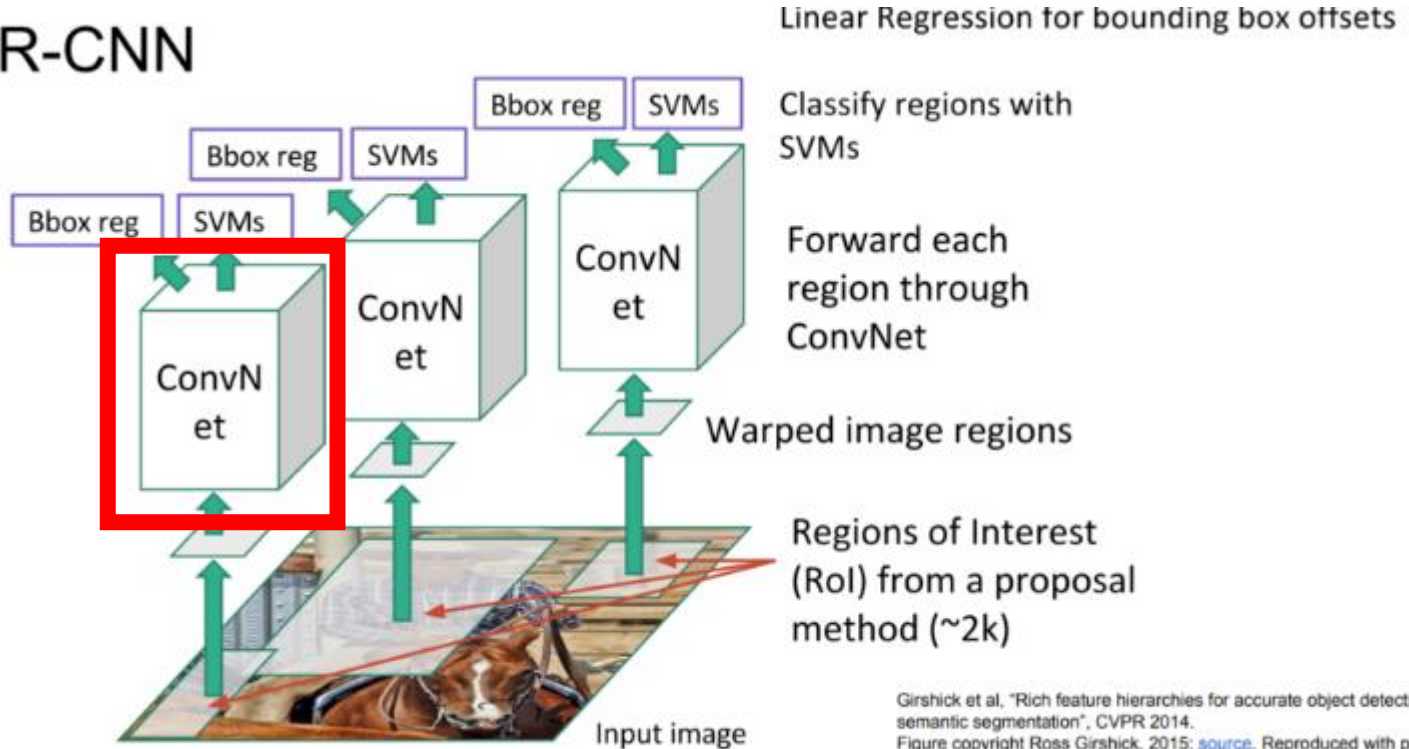
03 객체 인식 모델 R-CNN

CNN 총 이용 방법

R-CNN: Regions with CNN features



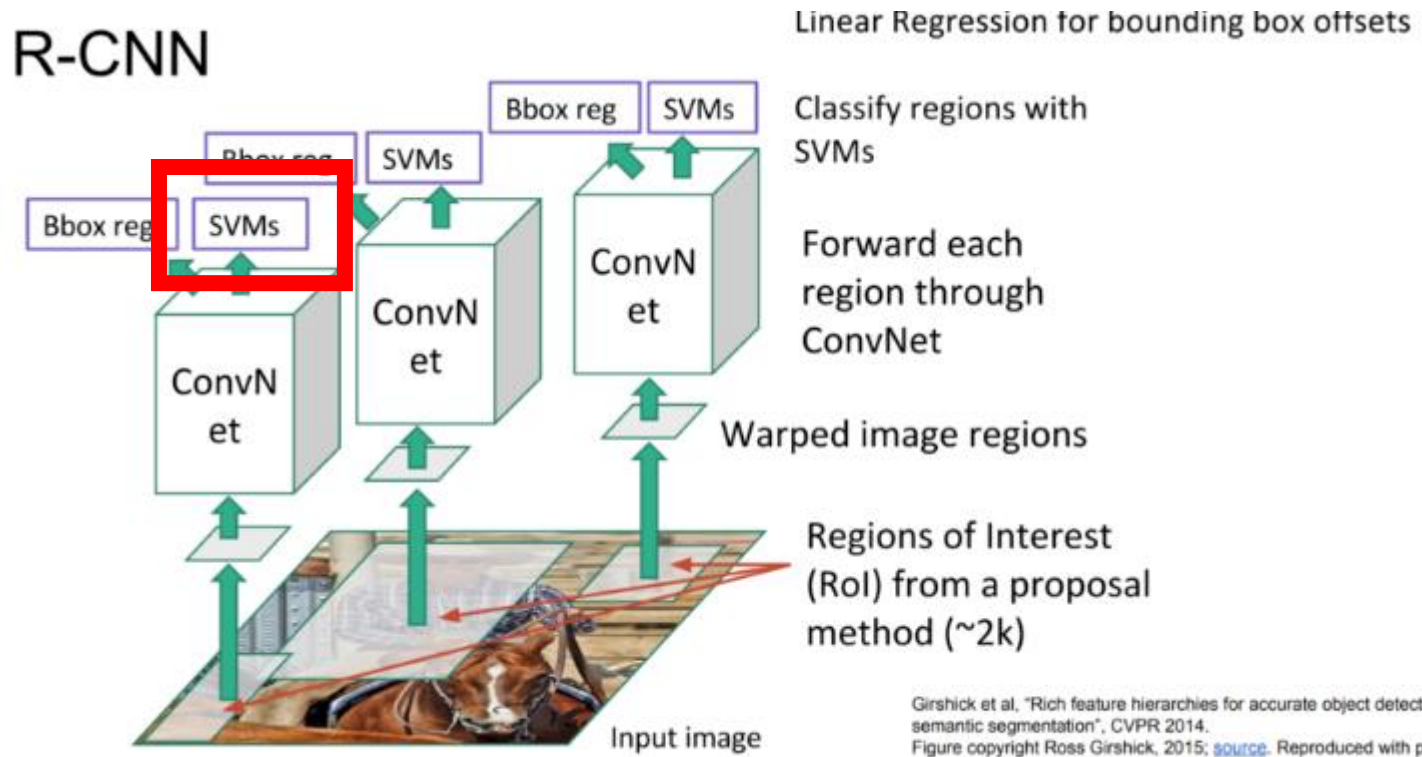
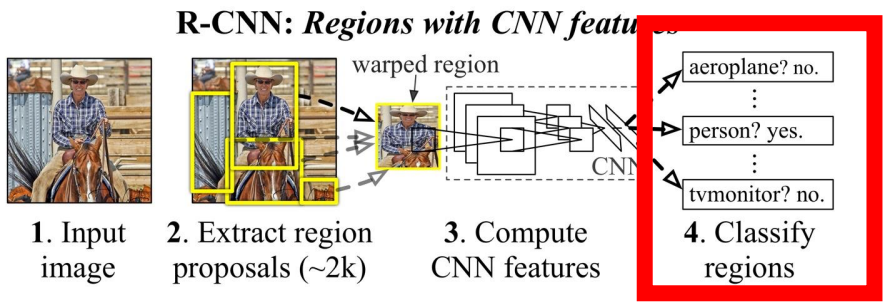
R-CNN



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

03 객체 인식 모델 R-CNN

SVM 모델 (이진 분류기)

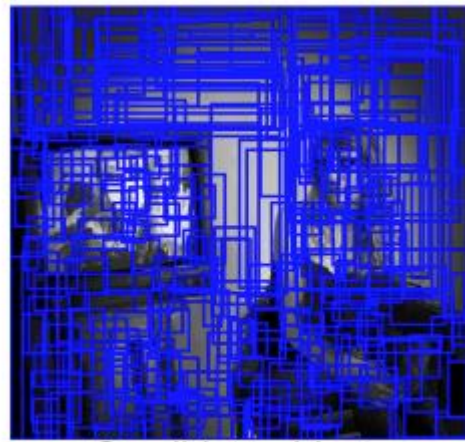


Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

03 객체 인식 모델 R-CNN

학습 방법

위치에 대한 점수로 활용되는 IOU



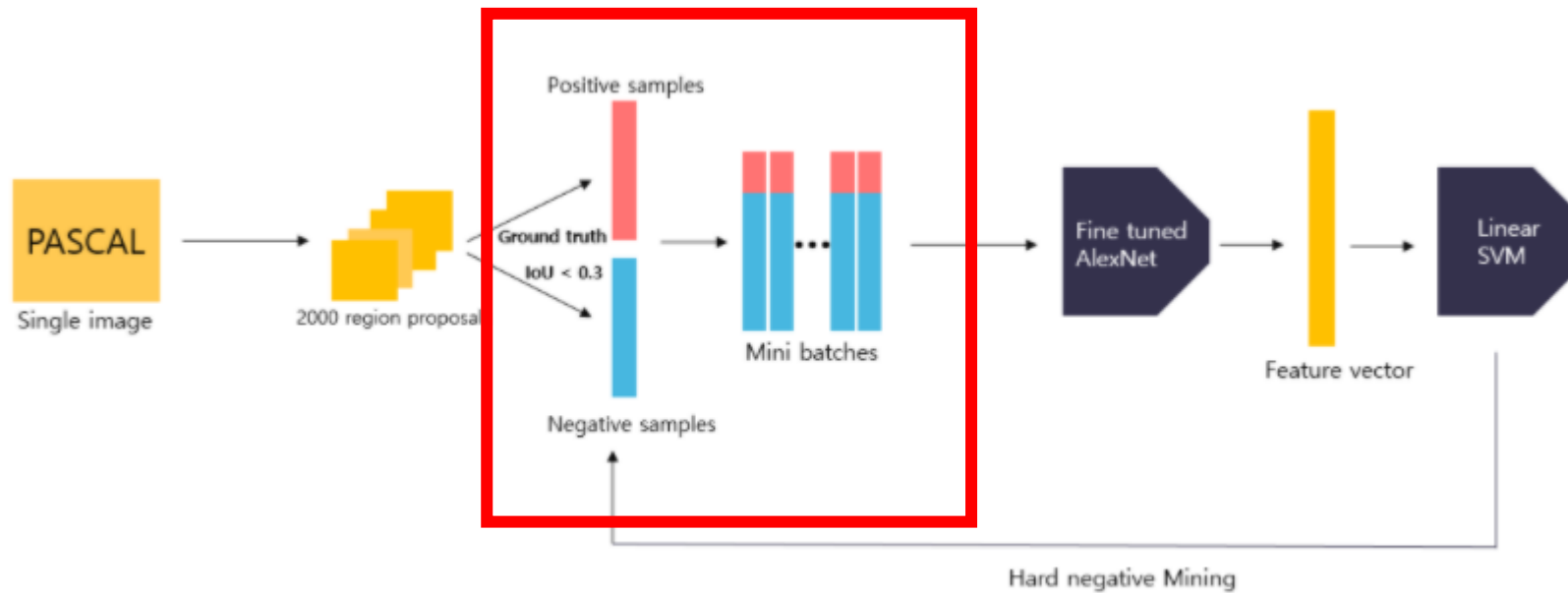
Candidate objects

$$\text{IoU} = \frac{\text{Overlapping Region}}{\text{Combined Region}}$$

03 객체 인식 모델 R-CNN

학습 방법

CNN 층 , SVM 모델 훈련 데이터 전처리



03 객체 인식 모델 R-CNN

속도 및 정확도

1. 테스트 시에 R-CNN은 이미지 하나 당 GPU에서는 13초, CPU에서 54초 소요
속도 저하의 가장 큰 병목 구간 : selective search를 통해서 찾은 2천개의 영역에 모두 CNN 을 통과 시
2. 정확도의 경우 Pascal VOC 2010을 기준으로 53.7%를 기록
당시 기존의 기록들을 모두 갈아치우며 획기적으로 Object Detection 분야에 발전을 이끌었던 스코어

04 객체 인식 모델 YOLO

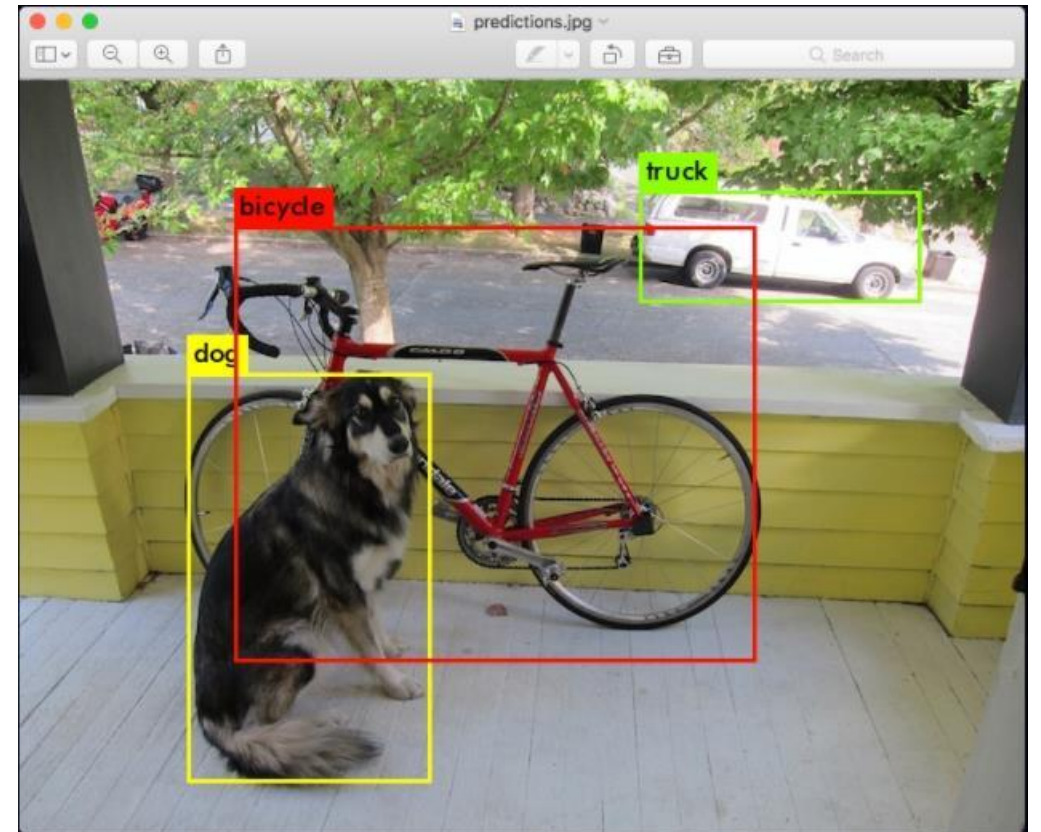
특징

- 기존의 multi-task 문제를 하나의 회귀(regression) 문제로 재정의
- YOLO는 이미지 전체에 대해서 한 번의 계산만으로 bounding box와 클래스 확률(class probability)을 예측
- 빠른 처리 속도 (1초에 45 프레임 처리, Fast YOLO는 1초에 155 프레임 처리)



YOLO: Real-Time Object Detection

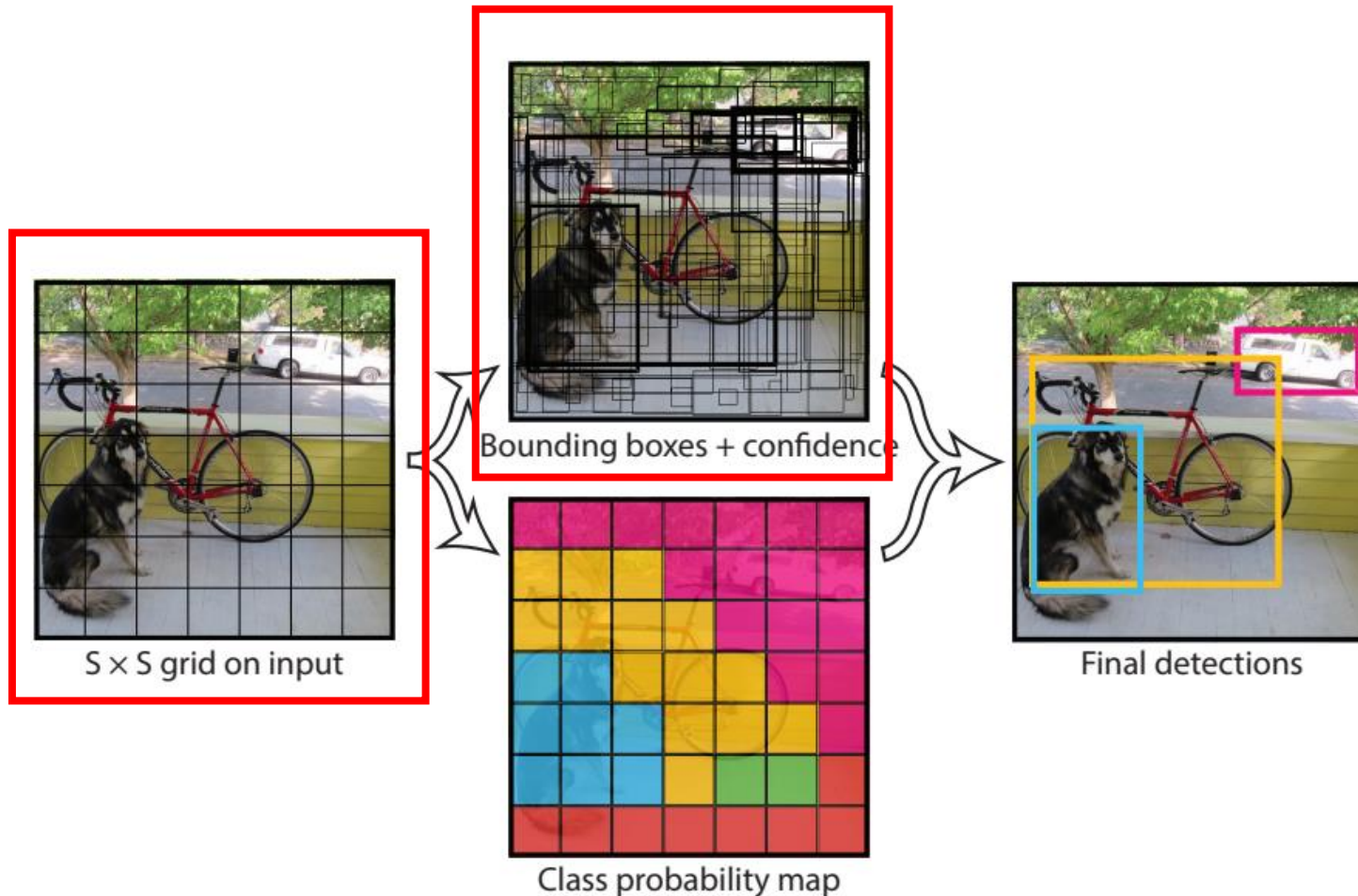
You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev.



04 객체 인식 모델 YOLO

YOLO 검출 과정

이미지에 대한 Grid를 그어 한 Cell마다 b개의 bounding box를 작성 (x,y,w,h)
각 박스마다 Confidence score(object 신뢰도 수치화)를 산출



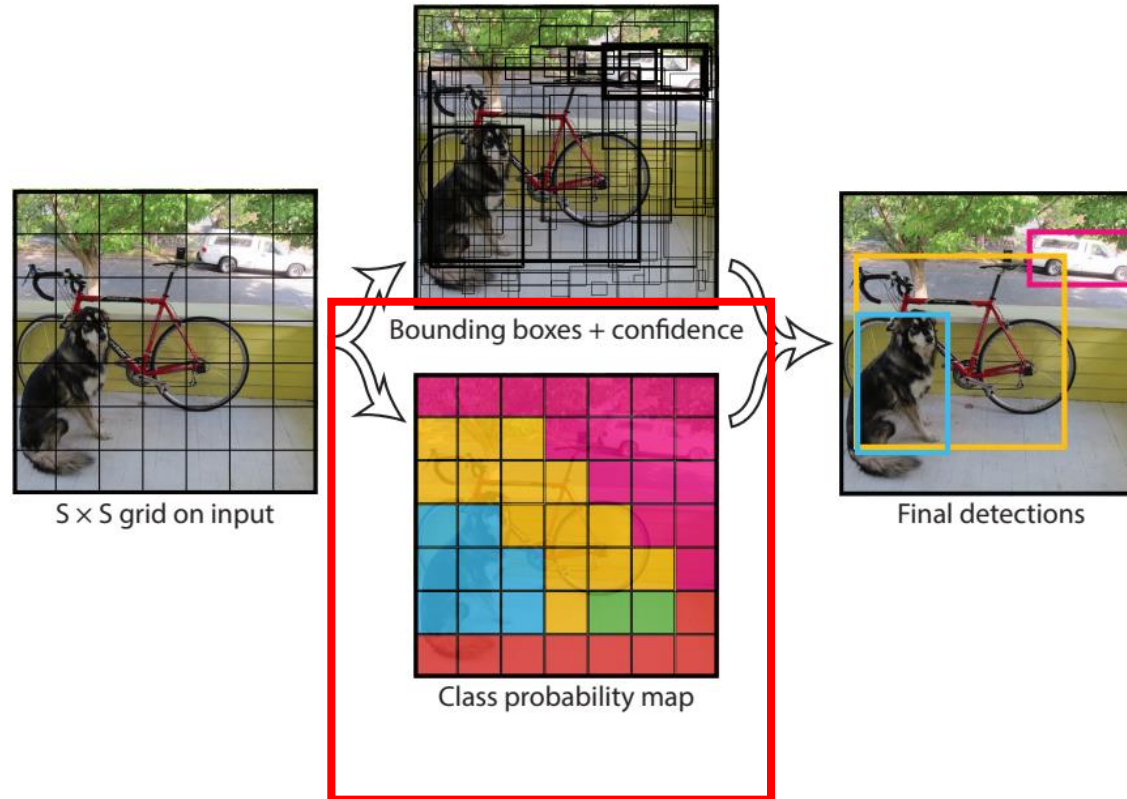
Confidence score

$$\text{Pr}(\text{Object}) * IOU_{pred}^{\text{truth}}$$

04 객체 인식 모델 YOLO

Conditional class probabilities(C) (점수 지표)

$$C(\text{conditional class probabilities}) = \Pr(\text{Class}_i | \text{Object})$$



04 객체 인식 모델 YOLO

YOLO 검출 과정

$$\begin{aligned} & \textit{class specific confidence score} \\ &= \Pr(\textit{Class}_i | \textit{Object}) * \Pr(\textit{Object}) * IOU_{pred}^{truth} \\ &= \Pr(\textit{Class}_i) * IOU_{pred}^{truth} \end{aligned}$$

04 객체 인식 모델 YOLO

YOLO Training

Leaky Relu 함수

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases}$$

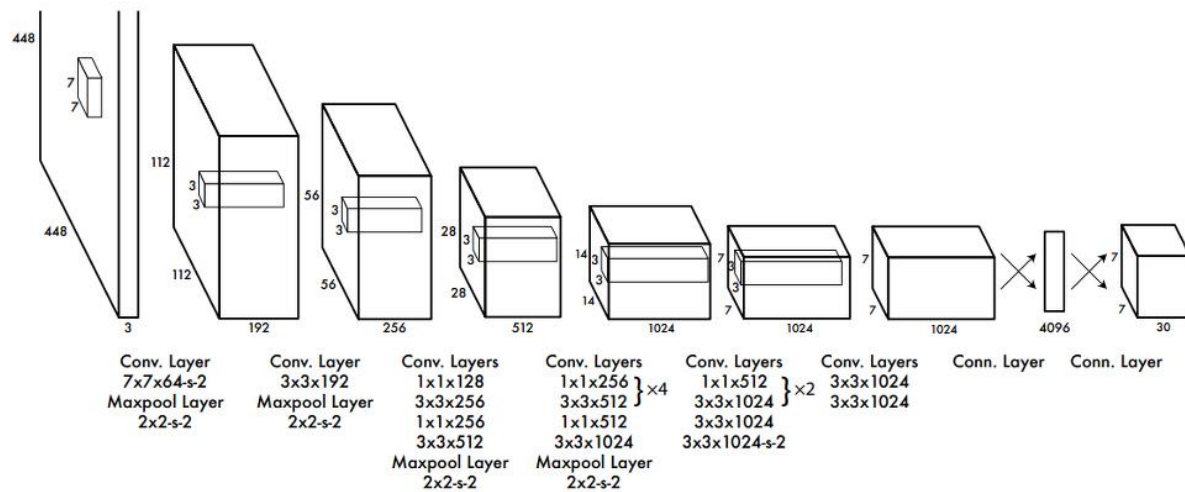


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

04 객체 인식 모델 YOLO

YOLO 손실 함수

1. bounding box의 위치를 얼마나 잘 예측했는지에 대한 loss

- 같은 박스의 크기에 동일한 가중치로 계산하면 안되는 점
작은 박스가 큰 박스보다 작은 위치 변화에 더 민감.
박스의 너비와 높이에 루트를 취해주어 개선

2. 클래스를 얼마나 잘 예측했는지에 대한 loss

- 이미지 내 대부분의 그리드 셀에는 객체가 없어서 객체가 있는 그리드 셀에 가중치를 부여

04 객체 인식 모델 YOLO

손실 함수

$$\begin{aligned} & \textbf{1} \quad \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & \textbf{2} \quad + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & \textbf{3} \quad + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\ & \textbf{4} \quad + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\ & \textbf{5} \quad + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

04 객체 인식 모델 YOLO

YOLO, R-CNN 비교

1. YOLO는 각 그리드 셀의 공간적 제약 때문에 하나의 객체가 여러 번 검출되는 경우가 R-CNN에 비해 적음.
2. 예측하는 bounding box의 개수 : YOLO < R-CNN
 - 한 이미지 당 bounding box 2,000개인 R-CNN에 비해 YOLO는 98개
3. YOLO : 단일 모델 , R-CNN : 다수의 모델로 구성

감사합니다
