

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

2024.01.26

이은주

# Intro

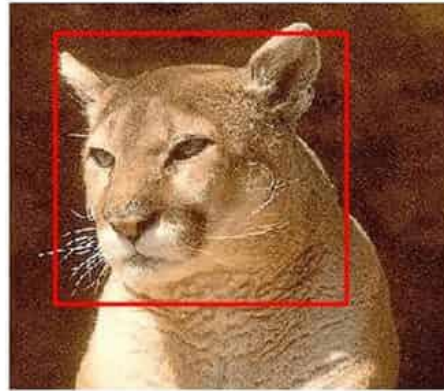
## Classification



Cougar

Output : class probability

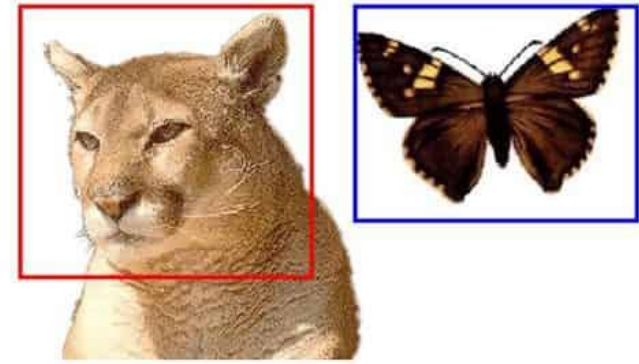
## Classification + Localization



Cougar

Output : (x, y, w, h)

## Object Detection

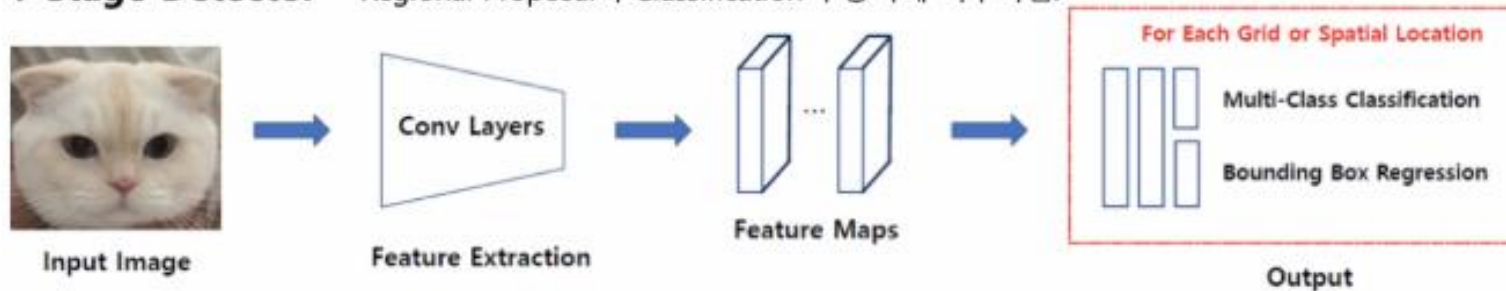


Cougar, Butterfly

Output : class probability + (x, y, w, h)

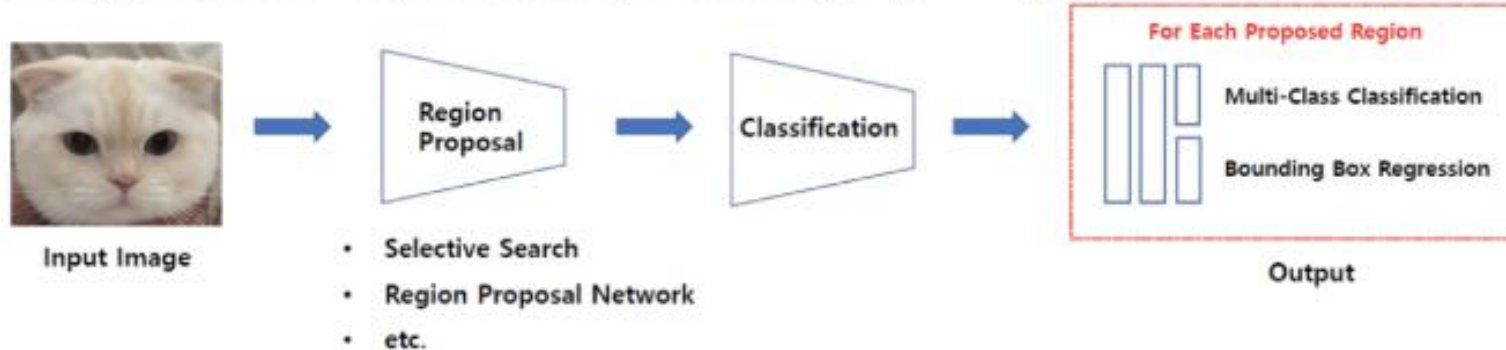
# Object Detection

**1-Stage Detector** - Regional Proposal와 Classification이 동시에 이루어짐.



이미지 내 모든 위치를 object의 잠재영역으로 보고 각 후보 영역에 대해 class 예측.  
YOLO 계열, SSD 계열

**2-Stage Detector** - Regional Proposal와 Classification이 순차적으로 이루어짐.

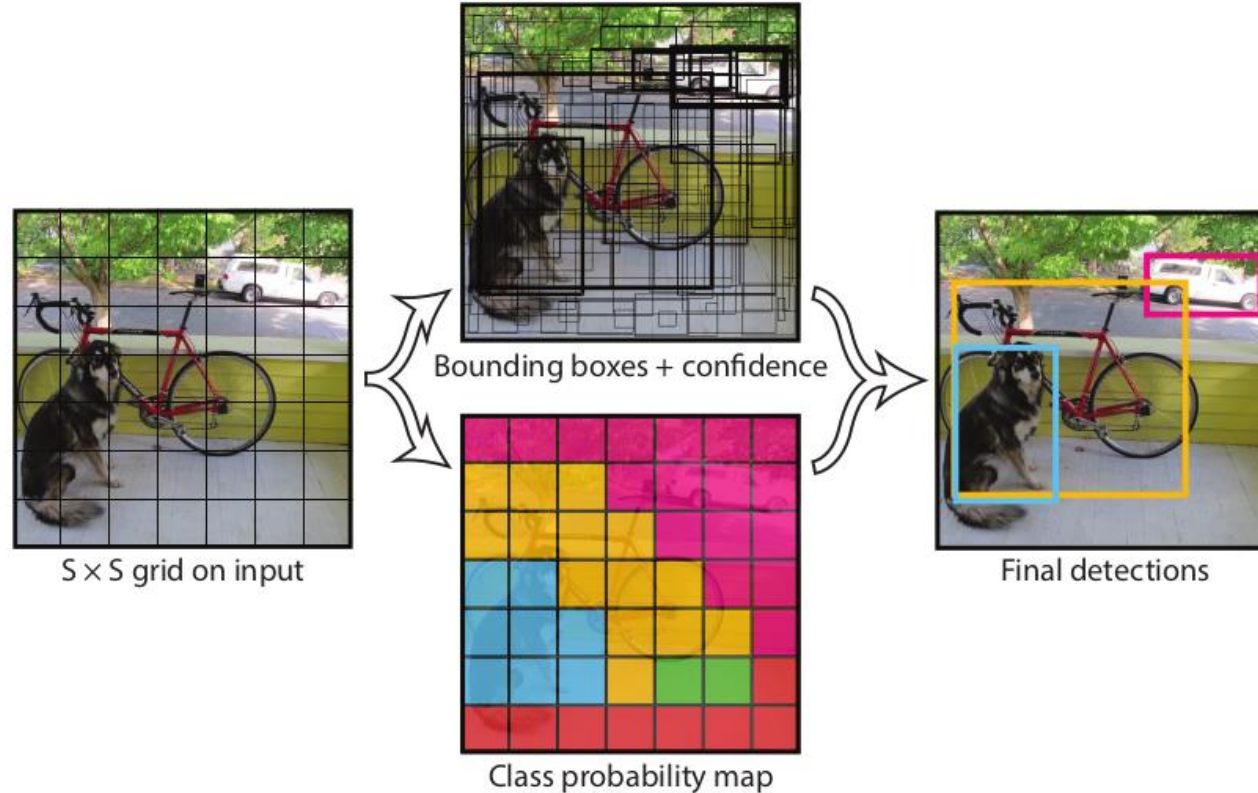


후보 object 위치 제안 후, object class 예측. R-CNN계열

# Yolo

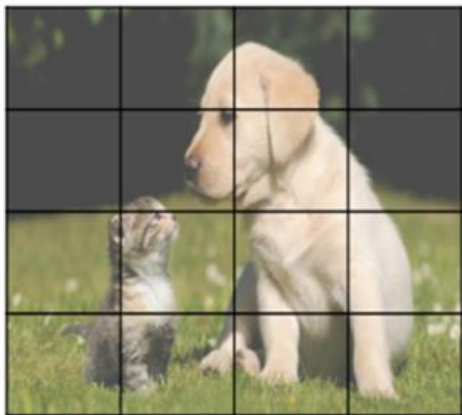
Region proposal, feature extraction, classification, bbox regression

-> one stage detection으로 통합

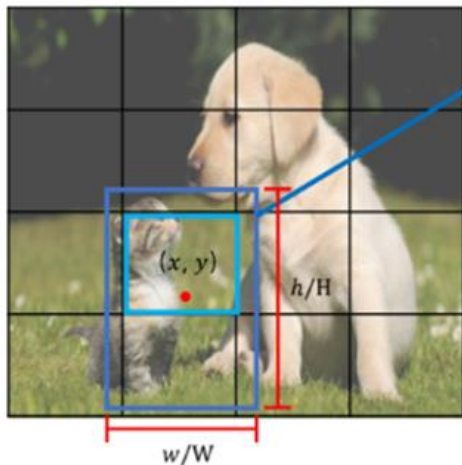


# Yolo 동작과정

예시)  $S = 4, B = 2, C = 20$



Resized image 를  
4x4 grid 로 분할



Grid cell 마다 bbox 2개씩 예측

bbox #1

$x$   
 $y$   
 $w$   
 $h$   
 $p_c$

bbox의 중심좌표의 위치  
(grid cell 기준)

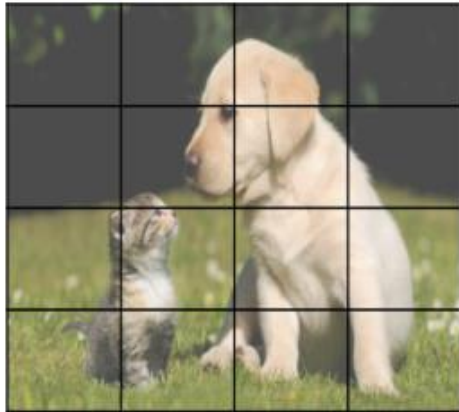
Input image  $W, H$  로 normalize

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

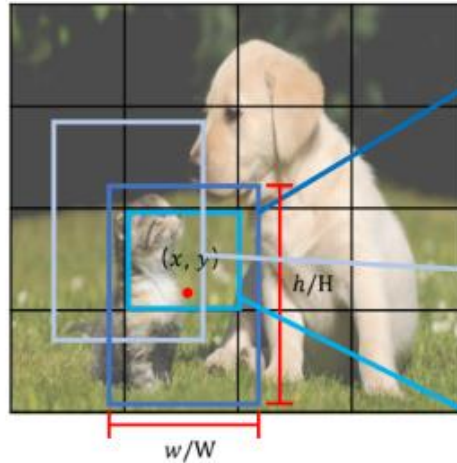
cf.  $\text{Pr}(\text{Object})$ :  
물체가 bbox 내에 있으면 1,  
없으면 0

# Yolo 동작과정

예시)  $S = 4, B = 2, C = 20$



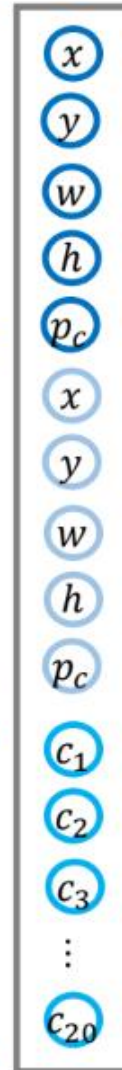
Resized image 를  
4x4 grid 로 분할



Grid cell 마다 bbox 2개씩 예측

bbox #1

bbox #2



bbox의 중심좌표의 위치  
(grid cell 기준)

Input image W, H 로 normalize

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

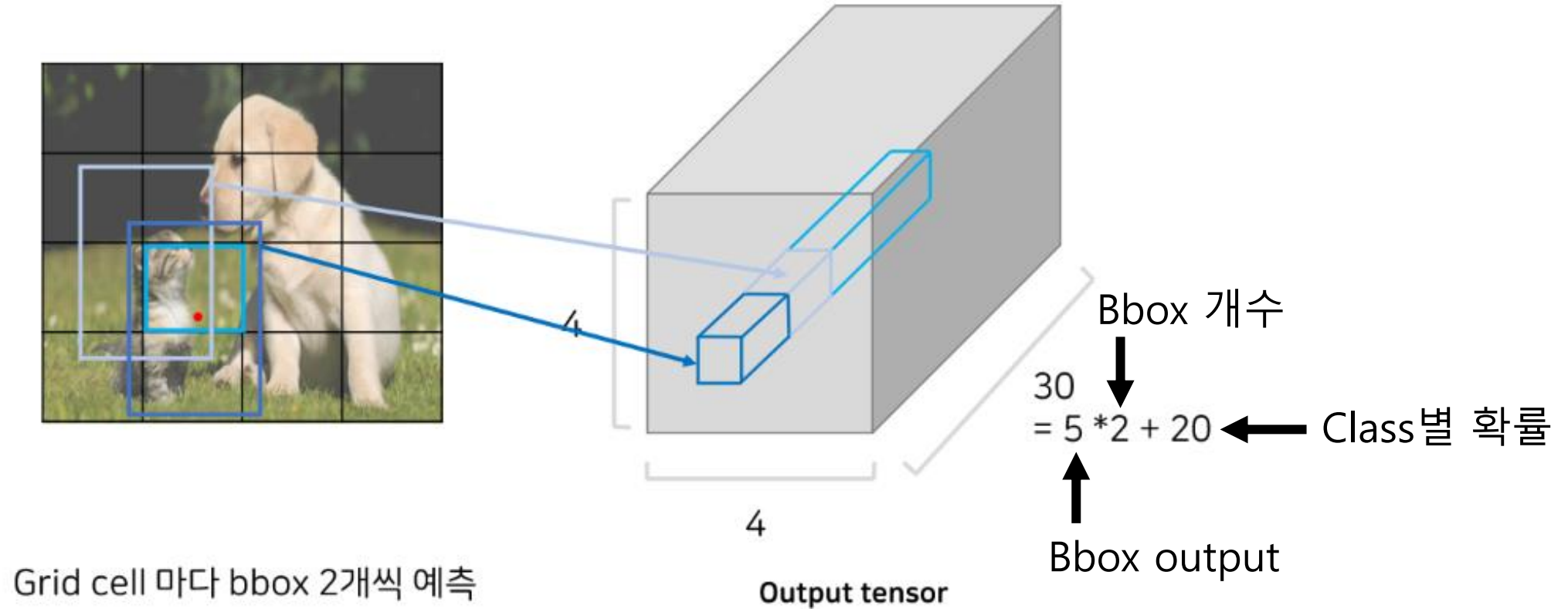
cf.  $\text{Pr}(\text{Object})$ :  
물체가 bbox 내에 있으면 1,  
없으면 0

$\text{Pr}(\text{Class}_i | \text{Object})$

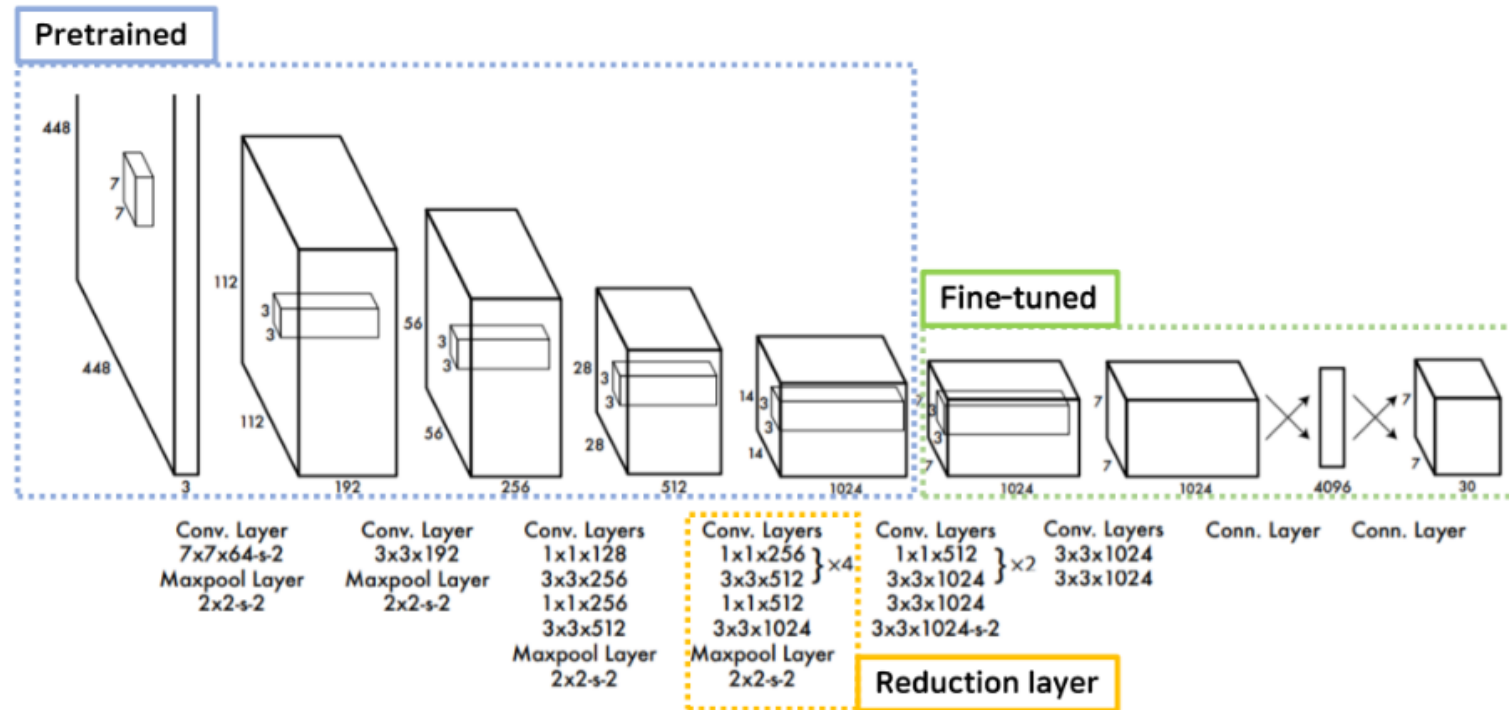
: 물체가 bbox 내에 있을 때,  
Grid cell에 있는 object가 i번째  
class에 속할 확률



# Yolo 동작과정



# Yolo Network



총 24개의 conv layer와 2개의 fc layer로 구성

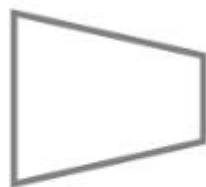
앞의 20개의 conv layer에 대해서는 1000개 클래스의 ImageNet 데이터셋으로 pretrained 된 부분  
뒤에 4개의 conv layer와 2개의 fc layer를 더 붙여서 Pascal VOC 데이터로 Fine tuning 시킨 과정  
노란색으로 표시된 중간에 1x1 reduction layer로 연산량 감소



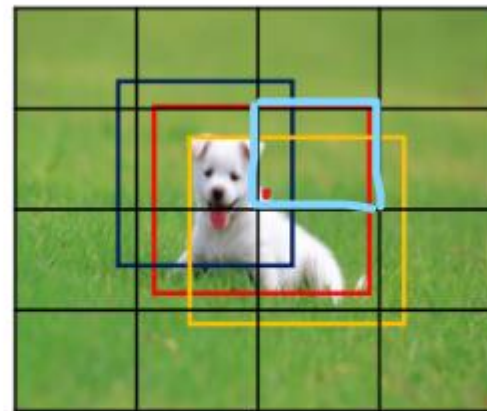
# Training



Input Image



Conv layers



Groundtruth

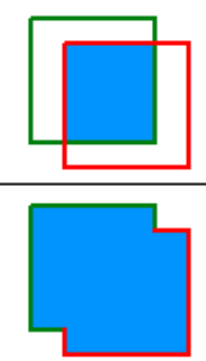


예측한 yellow bbox



예측한 blue bbox

Groundtruth 중심점이 cell 6에 위치  
즉, cell 6이 강아지 object 예측하는데 responsible한 cell이 된다.

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of overlap}}{\text{area of union}}$$


학습시에는 예측된 bbox중 하나만 사용한다. (한 개 선정 기준 : IOU)  
학습 단계에서 Groundtruth와 IOU가 가장 높은 예측 bbox 1개만 사용하여 진행

IOU blue < IOU yellow 이므로 cell 6에서 responsible한 bbox를 표시하여 loss function에 반영

# Training

Train 단계 loss function : MSE

Regression  
loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

모든 grid cell에서의 gt box좌표  
와 bbox좌표의 오차

Confidence  
loss

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2$$

모든 grid cell에서 예측한 class  
속할 확률 값과 gt값의 오차

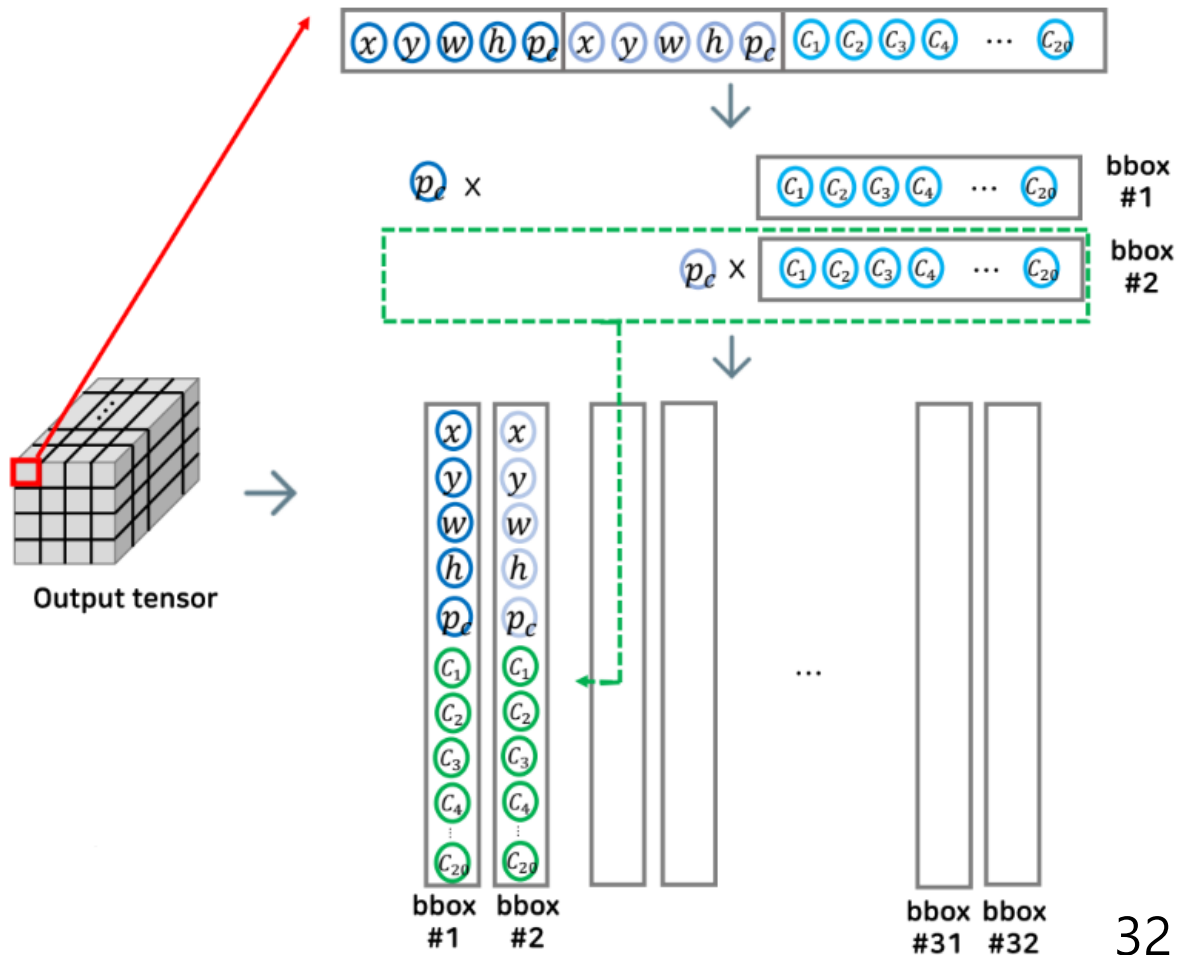
Classification  
loss

$$+ \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

모든 grid cell의 confidence  
score와 실제 정답과의 차이

i는 cell의 index , j는 bounding box predictor index

# Inference



Object당 bbox개수가 많아지므로  
NMS(알고리즘) 적용

NMS(Non-Maximum Suppression) :  
각 object에 대해 예측한 여러 bbox 중  
가장 예측력 좋은 bbox만 남김  
클래스 별로 각각 적용

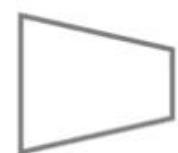
$$32 = 4 * 4 \text{ (image size)} * 2 \text{ (bbox 개수)}$$

# Inference

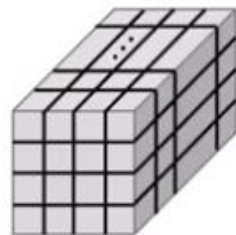
1) Object 가 1개인 경우



Input Image



Conv layers



Output tensor



## Non-Maximum Suppression

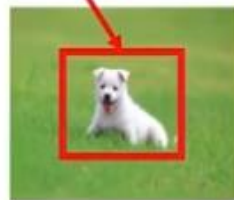
예시) Sort

|          |          |          |          |     |         |         |
|----------|----------|----------|----------|-----|---------|---------|
| 0.9      | 0.88     | 0.75     | 0.6      | ... | 0.0     | 0.0     |
| bbox #12 | bbox #13 | bbox #16 | bbox #17 |     | bbox #2 | bbox #1 |

32개

Th 못 넘은 것 remove

강아지



- class: 강아지  
- confidence: 0.9  
- bbox: #12

나머지 bbox들은 bbox12와 IOU계산  
나머지 bbox들은 IOU가 높아 NMS에 의해 제거 됨.  
bbox12와 bbox13의 IOU가 높다.  
즉, 두 bbox는 같은 object를 detect한다는 의미.

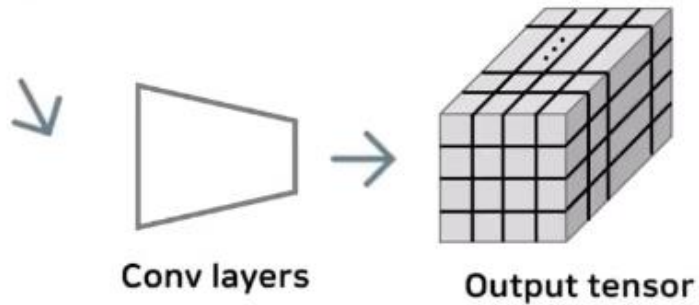
- Class 강아지를 가장 잘 예측하는 bbox는 #12 로 결정
- 나머지 bbox는 bbox#12와의 IOU가 높아서 NMS에 의해 모두 제거됨.

# Inference

2) 같은 class 속하는 Object 가 2개인 경우



Input Image



## Non-Maximum Suppression

예시)

|          |          |          |          |     |         |         |
|----------|----------|----------|----------|-----|---------|---------|
| 0.9      | 0.88     | 0.85     | 0.83     | ... | 0.0     | 0.0     |
| bbox #12 | bbox #13 | bbox #16 | bbox #17 |     | bbox #2 | bbox #1 |

강아지



Bbox12와 bbox16은 IOU낮음.

즉, 두 bbox는 다른 object라는 의미. 제거x

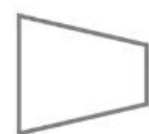
- Bbox#12 와 bbox#13의 IOU가 높으므로 NMS에 의해 제거됨.

# Inference

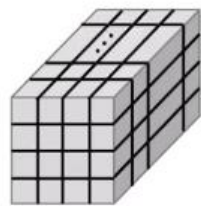
3) 다른 class 속하는 Object



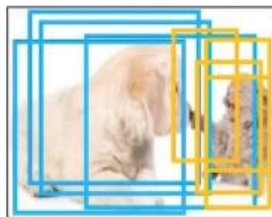
Input Image



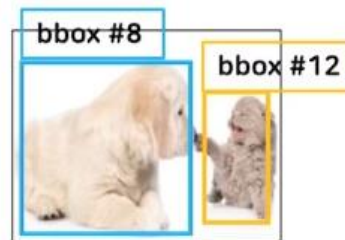
Conv layers



Output tensor



NMS 적용 전



NMS 적용 후

## Non-Maximum Suppression

예시)

|          |          |          |          |     |         |         |
|----------|----------|----------|----------|-----|---------|---------|
| 0.9      | 0.88     | 0.75     | 0.6      | ... | 0.0     | 0.0     |
| bbox #8  | bbox #9  | bbox #10 | bbox #11 |     | bbox #2 | bbox #1 |
| 0.84     | 0.81     | 0.7      | 0.65     |     | 0.0     | 0.0     |
| bbox #12 | bbox #13 | bbox #16 | bbox #17 |     | bbox #2 | bbox #1 |

강아지

고양이

클래스별로 NMS 알고리즘 수행하므로 강아지, 고양이 별도로 진행



# Comparison to Other Real-Time Systems

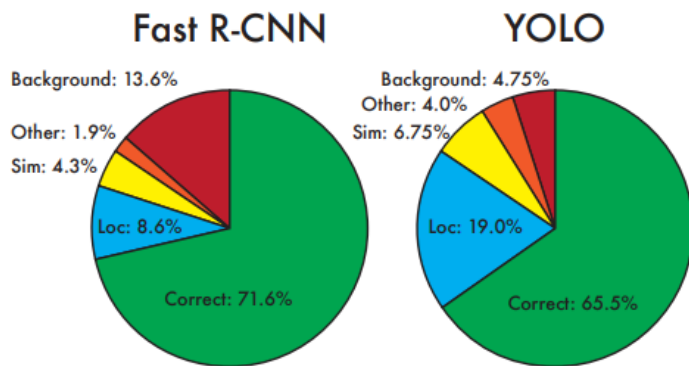
| Real-Time Detectors     | Train     | mAP         | FPS        |
|-------------------------|-----------|-------------|------------|
| 100Hz DPM [31]          | 2007      | 16.0        | 100        |
| 30Hz DPM [31]           | 2007      | 26.1        | 30         |
| Fast YOLO               | 2007+2012 | 52.7        | <b>155</b> |
| YOLO                    | 2007+2012 | <b>63.4</b> | 45         |
| Less Than Real-Time     |           |             |            |
| Fastest DPM [38]        | 2007      | 30.4        | 15         |
| R-CNN Minus R [20]      | 2007      | 53.5        | 6          |
| Fast R-CNN [14]         | 2007+2012 | 70.0        | 0.5        |
| Faster R-CNN VGG-16[28] | 2007+2012 | 73.2        | 7          |
| Faster R-CNN ZF [28]    | 2007+2012 | 62.1        | 18         |
| YOLO VGG-16             | 2007+2012 | 66.4        | 21         |

속도(FPS) : Fast Yolo > Yolo > DPM, RCNN

One stage > Two stage

성능(mAP) : Faster-RCNN > Fast-RCNN > Yolo > DPM

One stage < Two stage



Fast R-CNN + Yolo  
=> mAP 3.2% 향상  
(mean Average Precision)

|              | VOC 2007<br>AP | Picasso<br>AP Best $F_1$ | People-Art<br>AP |
|--------------|----------------|--------------------------|------------------|
| <b>YOLO</b>  | <b>59.2</b>    | <b>53.3</b>              | <b>45</b>        |
| R-CNN        | 54.2           | 10.4                     | 26               |
| DPM          | 43.2           | 37.8                     | 32               |
| Poselets [2] | 36.5           | 17.8                     | 0.271            |
| D&T [4]      | -              | 1.9                      | 0.051            |

AP성능은 Yolo가 다른 모델에 비해 좋음  
(Average Precision)