

# Object Detection

한컴엔플렉스  
AI융합기술센터

Date

2022.09.06

Presenter

박정완 센터장

# INDEX

01

Object Detection이란?

소개  
기본 분류

02

R-CNN

정의  
아이디어  
작동원리

03

YOLO

정의  
아이디어  
작동원리

04

국방데이터에 적용

YOLO v5  
스마트 지뢰탐지시스템 적용



## Object Detection이란?

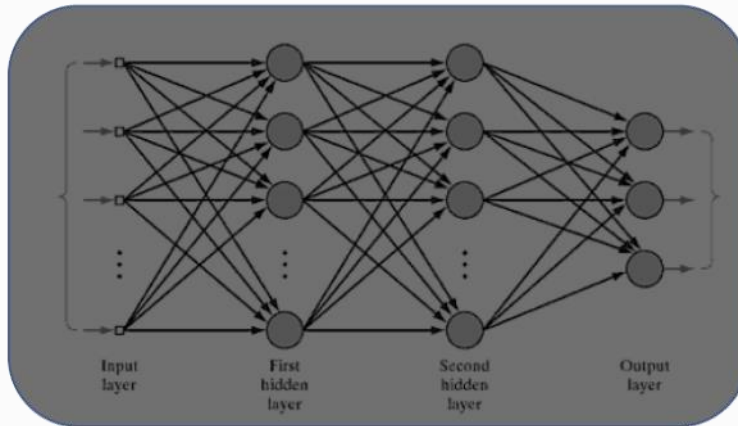


물체 인식의 정의, 종류

# Object Detection이란?

## 먼저, Image Classification이란?

- DNN에 입력으로 이미지를 넣으면 그 이미지에 해당하는 Class를 분류해내는 문제
- 아래 그림과 같이 타겟으로 하는 전체 class에 대한 확률 값들을 출력



$P_{\text{dog}} = 0.9$   
 $P_{\text{cat}} = 0.1$

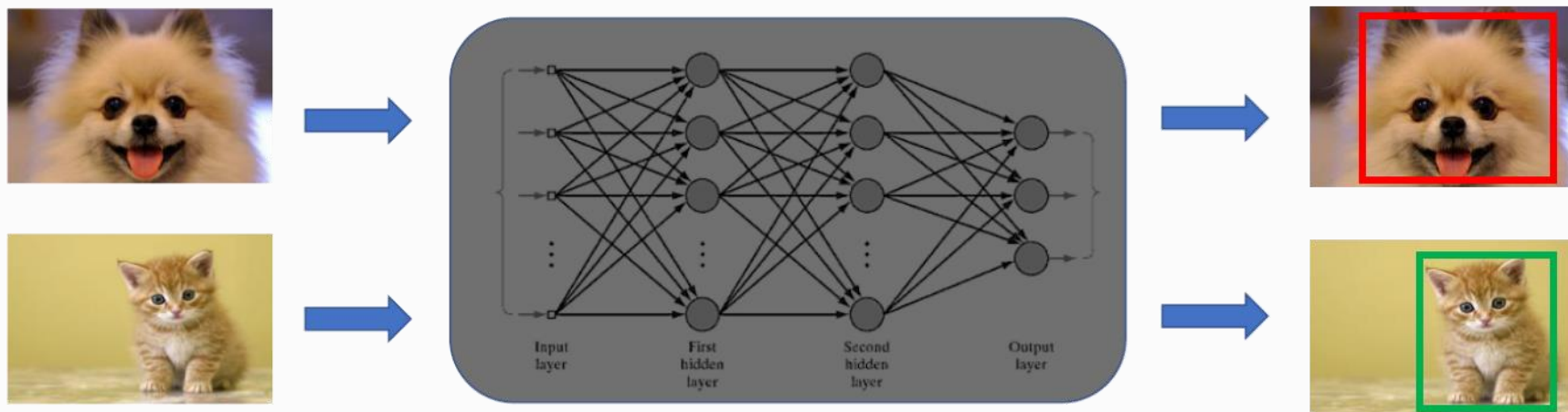


$P_{\text{dog}} = 0.05$   
 $P_{\text{cat}} = 0.95$

# Object Detection이란?

## Object Detection의 정의

- Object Detection은 Image Classification task에 사물의 위치를 Bounding Box로 예측하는 Regression task(회귀문제)가 추가된 것
- Object Detection = Image Classification task + Bounding box + **Regression task**
- Bounding box는 좌표값과 초기정보를 설정 -> 좌표정보를 추정해주기 위해 선형회귀를 통해 추정값을 얻음



# Object Detection이란?

## Object Detection 이론

1

1 stage - object detection

2

2 stage - object detection

R-CNN과 같이 탐색 영역을 찾는 Region Proposal  
(이미지 안에서 객체가 있을 만한 후보 영역을 찾아줌)

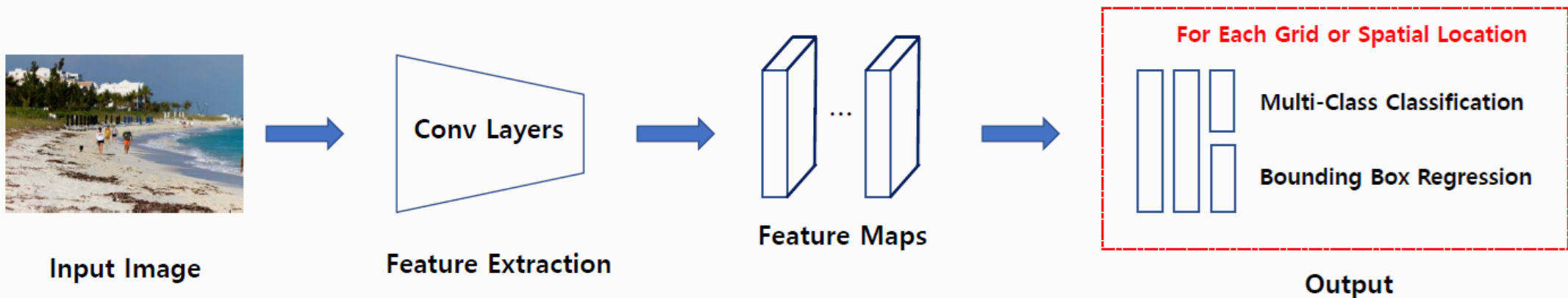


해당 영역을 분류하는 Detection (class 탐지 )

# Object Detection이란?

## 1 Stage-Object Detection

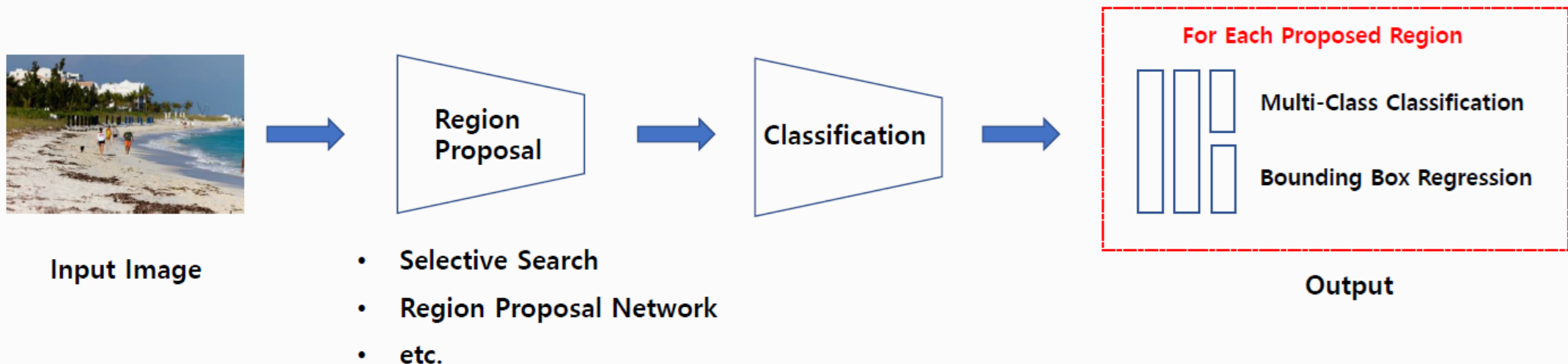
- 1-Stage Object Detector → 두 가지 과정을 한번에 처리하는 방법
- Region proposal과 Detection이 한 번에 수행
- 2-Stage Object Detector에 비해 빠르지만 부정확
- 대표적 종류 : SSD, YOLO



# Object Detection이란?

## 2 Stage-Object Detection

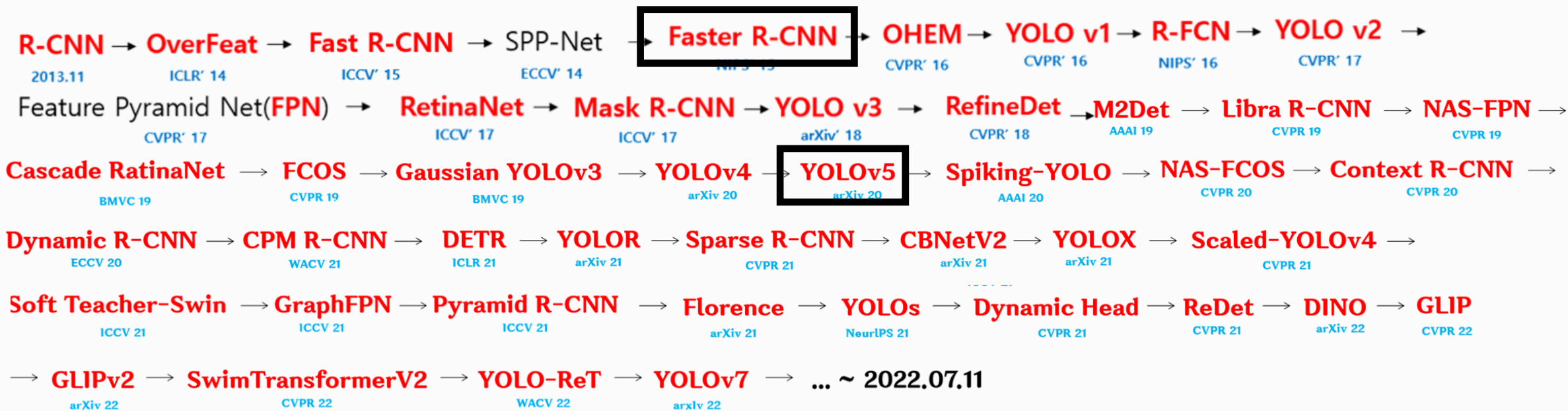
- 2-Stage Object Detector → 두 가지 과정이 순차적으로 수행되는 방법론
- Region proposal과 Detection 2단계의 과정을 순차적으로 거침
- 1-Stage Object Detector에 비해 비교적 느리지만 정확
- 대표적 종류 : R-CNN, Fast R-CNN, Faster R-CNN





# Object Detection이란?

## 기술발전과정





## R-CNN



CNN의 이해

Region Based Convolutional Neural Networks의 대표적 기술

주요 원리

주요 알고리즘(Selective 알고리즘, Greedy 알고리즘)

Bounding box regression

# R-CNN이란?

## CNN(Convolution Neural Network, 합성곱 신경망)

DNN(Deep Neural Network)에서 이미지나 영상 같은 데이터를 처리할 때 발생하는 문제점들을 개선한 방법



---

### ■ 제한사항

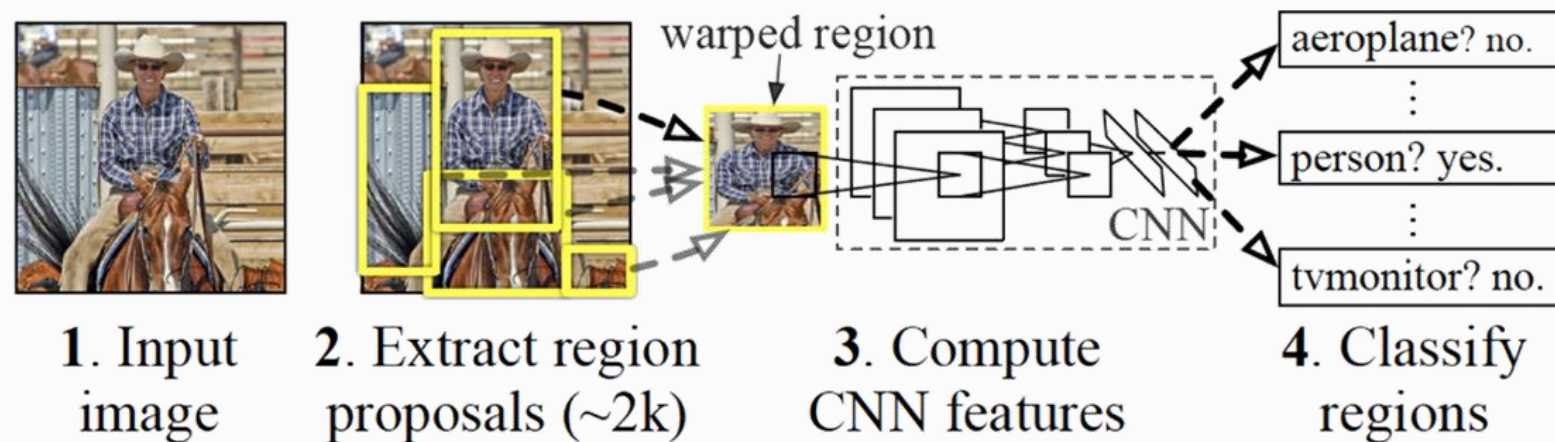
이미지 공간 정보 유실로 인한 정보 부족

인공 신경망이 특징을 추출 및 학습이 비효율적이고 정확도를 높이는데 한계 존재

# R-CNN이란?

## Region Based Convolutional Neural Network

2-Stage Object Detection의 대표적 기술



Task: →

### 1. Region Proposal

Selective 알고리즘 + Greedy 알고리즘

### 2. Region Classification

Bbox regression + classification

### 작동원리

물체의 위치를 찾는 Region Proposal + 물체를 분류하는 Region Classification

두 가지 Task를 순차적으로 적용하여 시행

# R-CNN이란?

## R-CNN 작동원리

### ■ 다량의 bbox 추출

이미지의 데이터와 레이블을 투입하여, selective 알고리즘을 적용  
박스 2000개를 추출

### ■ CNN으로 특징벡터를 추출

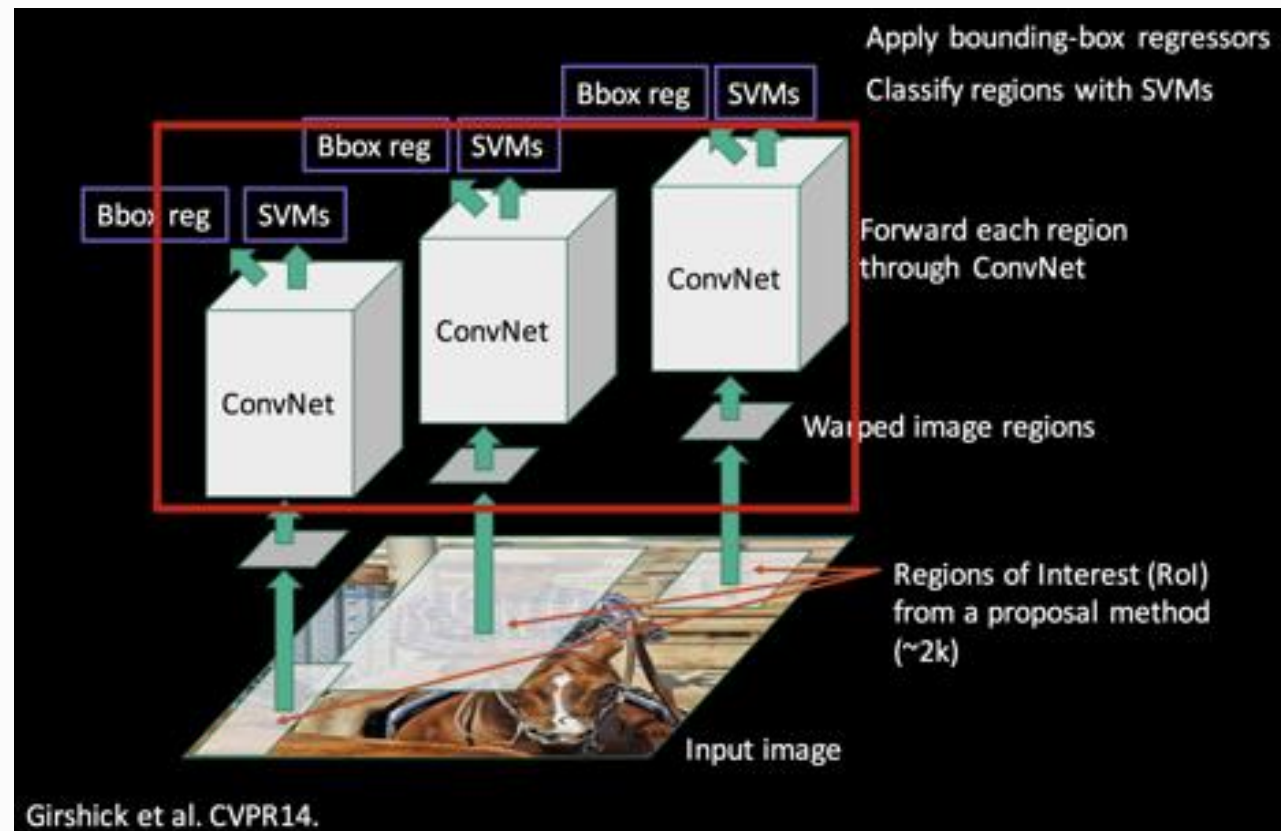
사전에 훈련 시킨 CNN을 활용  
차원의 특징 벡터를 추출하여 Feature map을 형성

### ■ Classification

SVM(Support Vector Machine)으로 분류분석  
Feature map을 활용하여 선형지도학습모델 적용

### ■ Bounding box regression

적절한 크기와 위치를 지정하도록 진행



# R-CNN이란?

## Selective Search(SS)알고리즘

Object가 있을 만한 영역(region)을 찾는 알고리즘(Region Proposal, RP)의 종류 중 하나

### ■ Input이미지에 sub-segmentation 진행

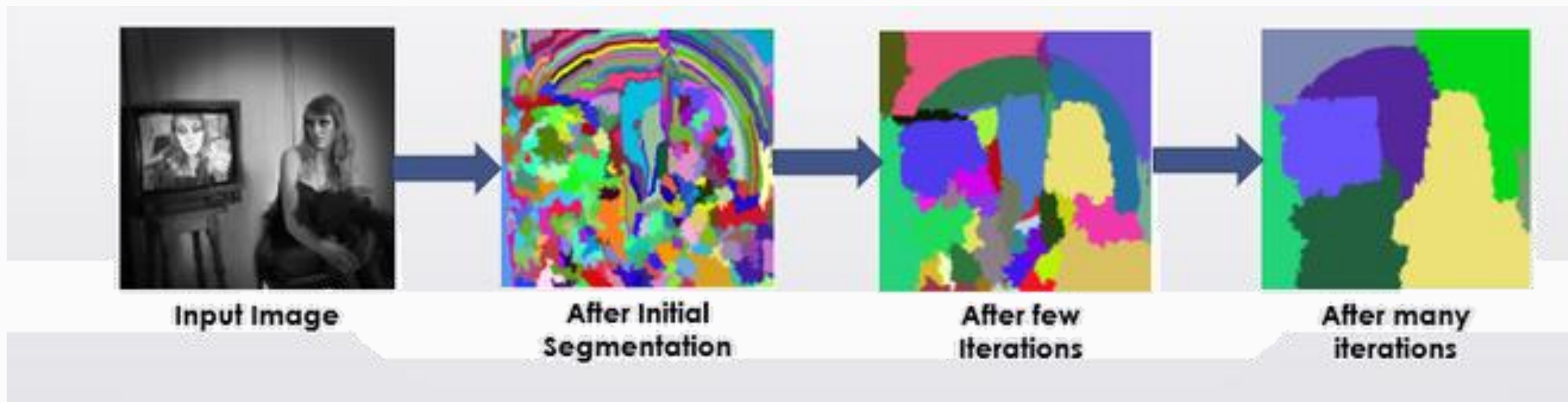




# R-CNN이란?

## Selective Search(SS)알고리즘

- 반복적으로 작은 영역을 큰 영역으로 결합(greedy 알고리즘)

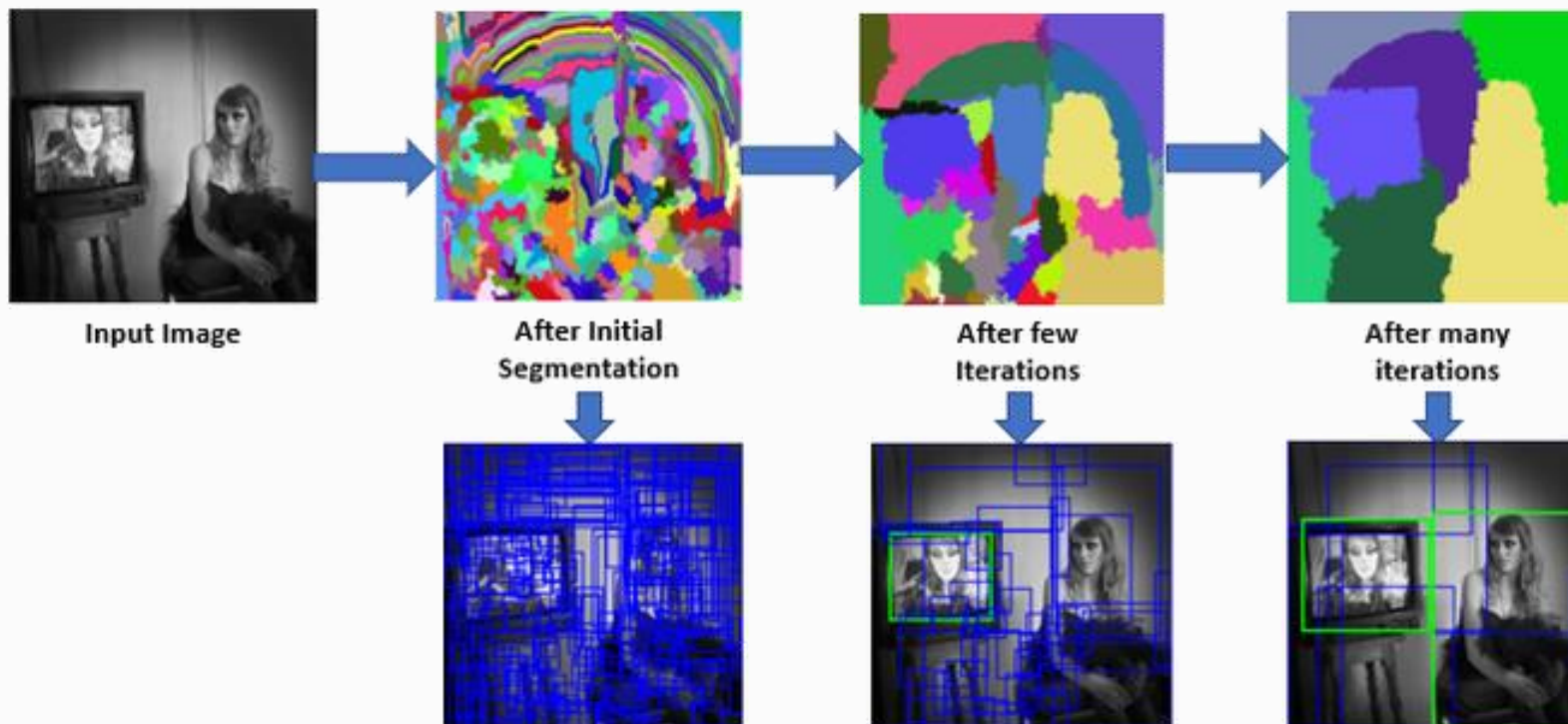


# R-CNN이란?

## Selective Search(SS)알고리즘

### ■ 수가 적어지면 후보 위치를 생성

Segmented region proposals(분할된 지역 제안)를 사용하여 candidate object location(후보개제지역)를 생성





# R-CNN이란?

## bounding box regression

- Selective search(알고리즘)를 통해 찾은 박스 위치는 정확하지 않음
- Predicted box(예측값)과 ground truth box(실제값)과의 차이를 줄이는 것이 목적

하나의 박스(예측한 박스)에 대한 위치 값

$$P^i = (P_x^i, P_y^i, P_w^i, P_h^i)$$

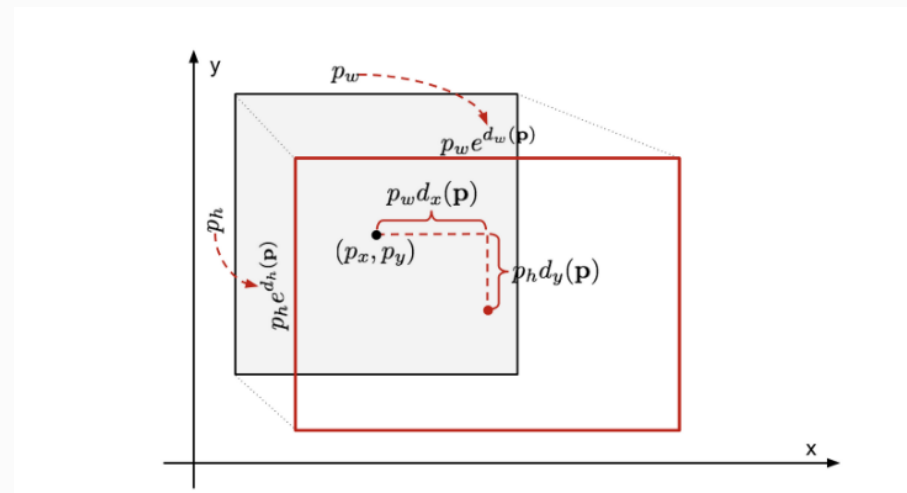
$(x, y)$  = 이미지의 중심좌표

$(w, h)$  = 이미지의 너비와 높이

실제 박스에 대한 위치 값

$$G = (G_x, G_y, G_w, G_h)$$

- 목표 : P에 해당하는 박스를 G에 가깝게 이동시키는 함수를 학습시키는 것





**YOLO**



YOLO 특징  
작동원리  
NMS알고리즘  
YOLO순서

# YOLO란?

## You Only Live Once 특징

### ■ 1-Stage Object Detection의 대표적인 기술

### ■ 이미지 전체를 한번만 인식

이전의 R-CNN은 이미지를 여러장으로 분할

YOLO는 분할 과정없이 한번에 인식

### ■ 실시간으로 객체 탐지 가능

R-CNN은 실시간 탐지 불가능

YOLO는 실시간 탐지가 가능

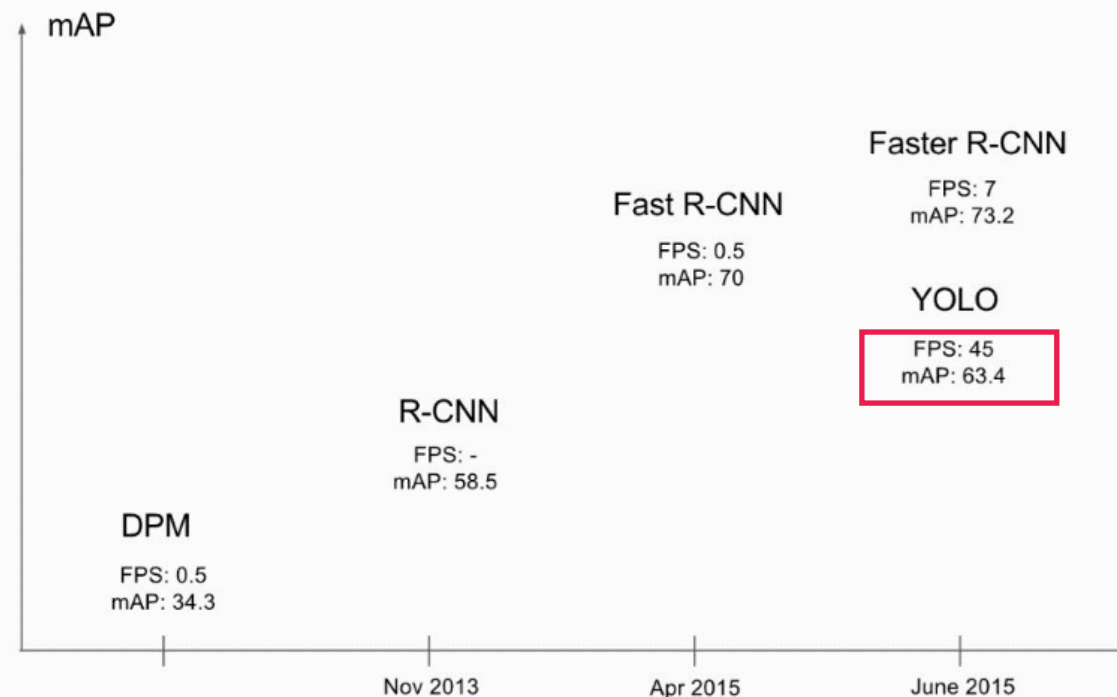
### ■ 통합된 모델을 사용

기존은 전처리 모델, 인공 신경망을 결합하여 사용

통합된 모델을 사용하여 활용가능

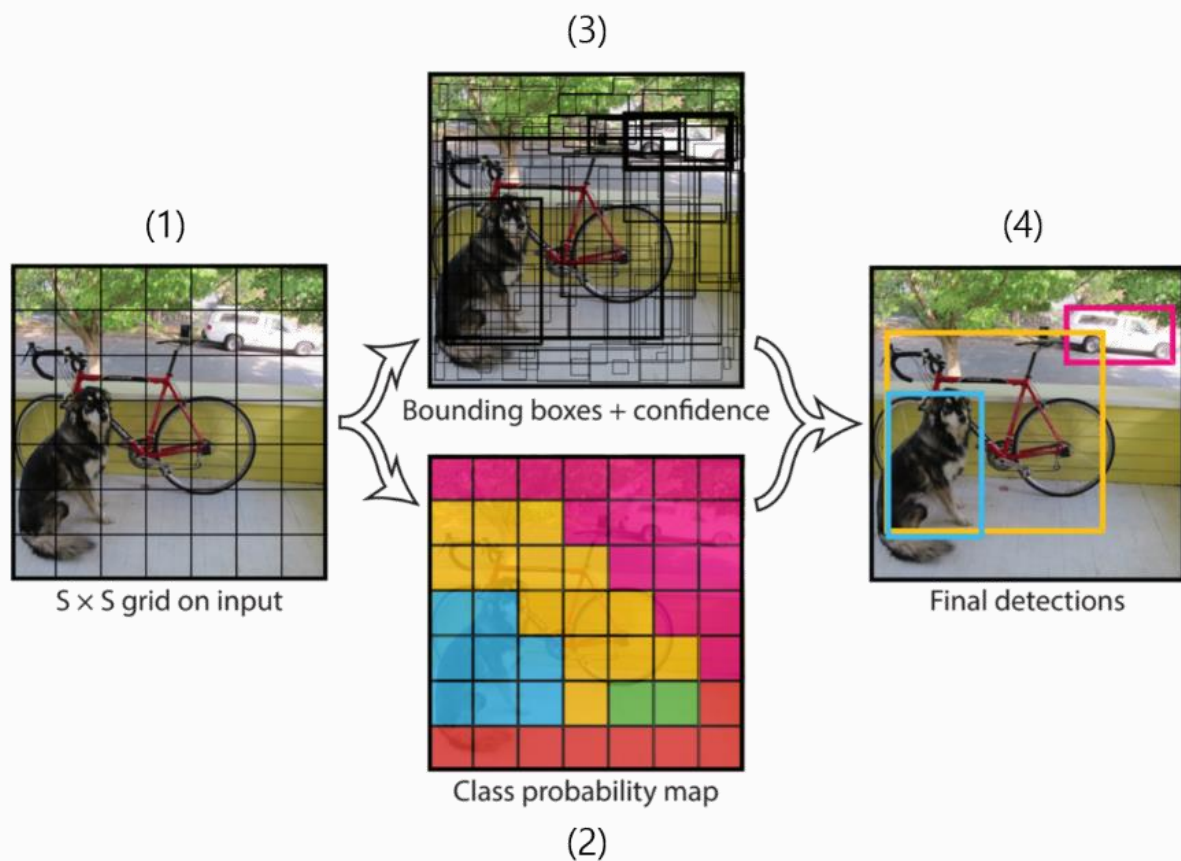
mAP(mean Average Precision) : 영상인식 정확도 평가지표

FPS(Frame Per Second) : 초당 프레임. (초당 25프레임은 끊기지 않는다고 판단)



# YOLO란?

## 작동 원리



입력된 이미지를  $S \times S$  사이즈로 나눔

각각의 grid cell은 B개의 bounding box와 그에 대한 confidence score를 가짐

각각의 bbox는  $x, y, w, h$ 와 confidence로 구성

$(x, y)$  = bbox의 중심점(grid셀의 크기 기준 1, 중심 0.5, 0.5)

$(w, h)$  = 이미지의 너비와 높이(grid셀의 크기 기준 1, 크기 10%는 0.1, 0.1)

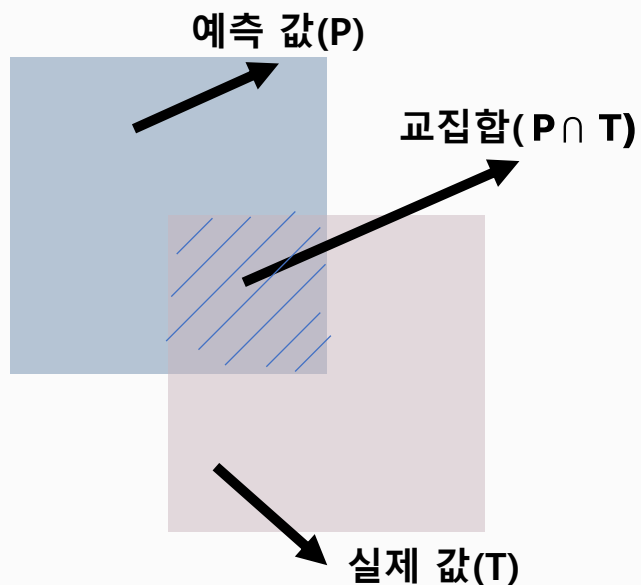
각각의 grid cell은 C(confidence class probability)를 가짐

평가할 때 C와 각 박스의 class-specific confidence score를 주는 confidence prediction을 곱함

# YOLO란?

## Confidence Score

$$\text{confidence score} = P_r(\text{object}) \times IOU_{pred}^{truth}$$



■ Bbox내에 object가 있을 확률을 예측하는 지표  
Cell낸 object가 없으면 0

■ 오브젝트일 확률이며, Softmax(활성화 함수)로  
0~1사의 값으로 모두 정규화

총합은 항상 1이 되는 특성

$$y_k = \frac{e^{a_k}}{\sum_{i=1}^n e^{a_i}}$$

■ IoU(Intersection Over Union)는 두 영역이 겹쳐져  
있을 때 두 영역의 교집합을 합집합으로 나눈 것

$$IoU = (P \cap T \text{의 면적}) / (P \cup T \text{의 면적})$$

# YOLO란?

## Conditional class probability & class-specific confidence score

$C = \text{conditional class probability} : P_r(\text{class}_i | \text{object})$

| **C**는 **class**의 개수

| 각각의 **grid cell**은 **C**개의 조건부확률을 가짐

*Class Specific Confidence Score*

$= \text{Conditional Class Probability} \times \text{Confidence Score}$

$= P_r\{\text{class}_i | \text{object}\} \times P_r(\text{objects}) \times IOU_{pred}^{truth}$

$= P_r(\text{class}_i) \times IOU_{pred}^{truth}$

# YOLO란?

## 최종결과값

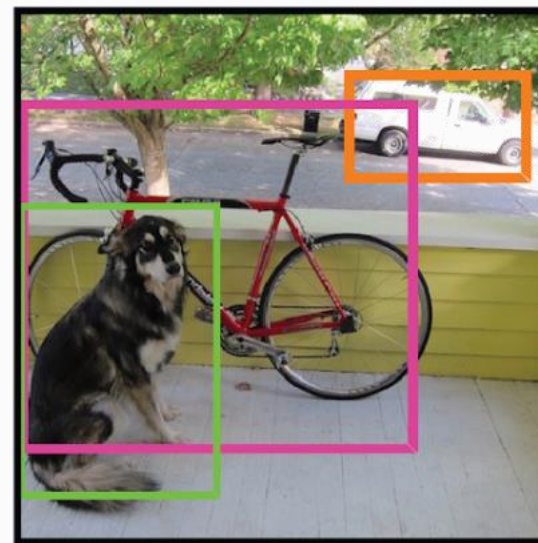
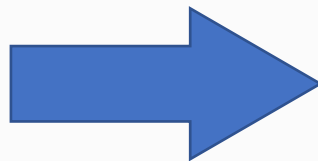
*Final prediction result =  $7 \times 7 \times 30$  tensor*

$$\text{Output Size} = S \times S \times (B \times 5 + C) = 7 \times 7 \times (2 \times 5 + 20) = 7 \times 7 \times 30$$

- 각 셀 격자는  $(B \times 5 + C)$  크기의 *vector* 를 가짐
- 이미지 전체  $(S \times S \times (B \times 5 + C))$  크기의 *tensor* 를 가짐
- $B \times 5$  = Bounding box  $B$  개의  $x, y, w, h, confidence$  5개의 변수

# YOLO란?

## 결과 화면

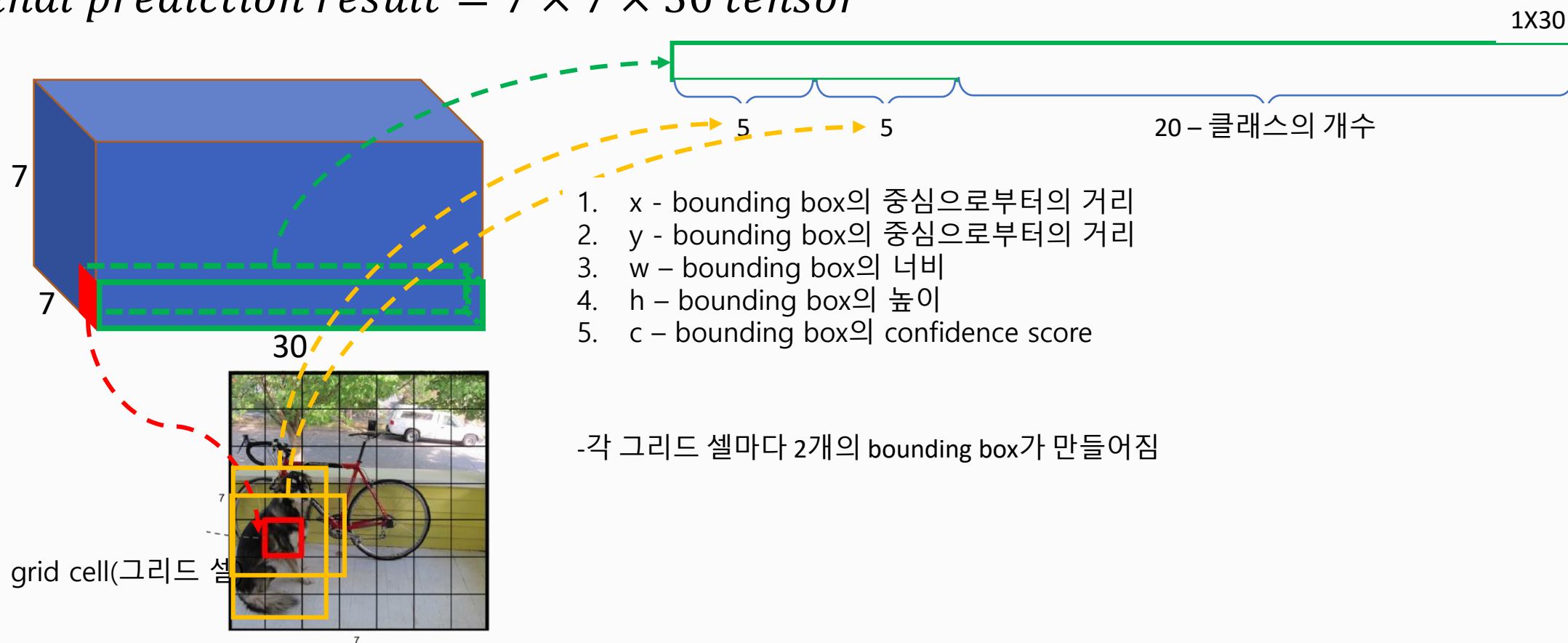




# YOLO란?

## YOLO 작동 원리

*Final prediction result =  $7 \times 7 \times 30$  tensor*

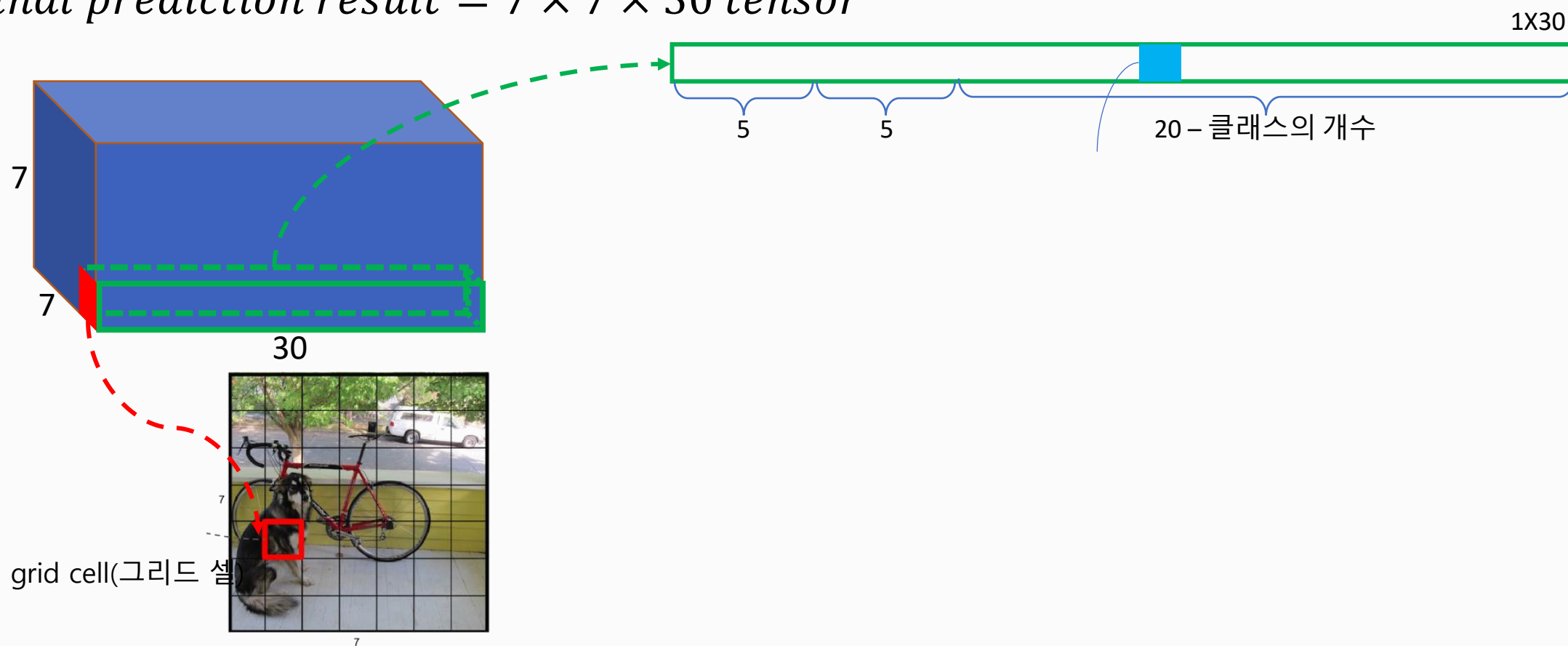


-각 그리드 셀마다 2개의 bounding box가 만들어짐

# YOLO란?

## YOLO 작동 원리

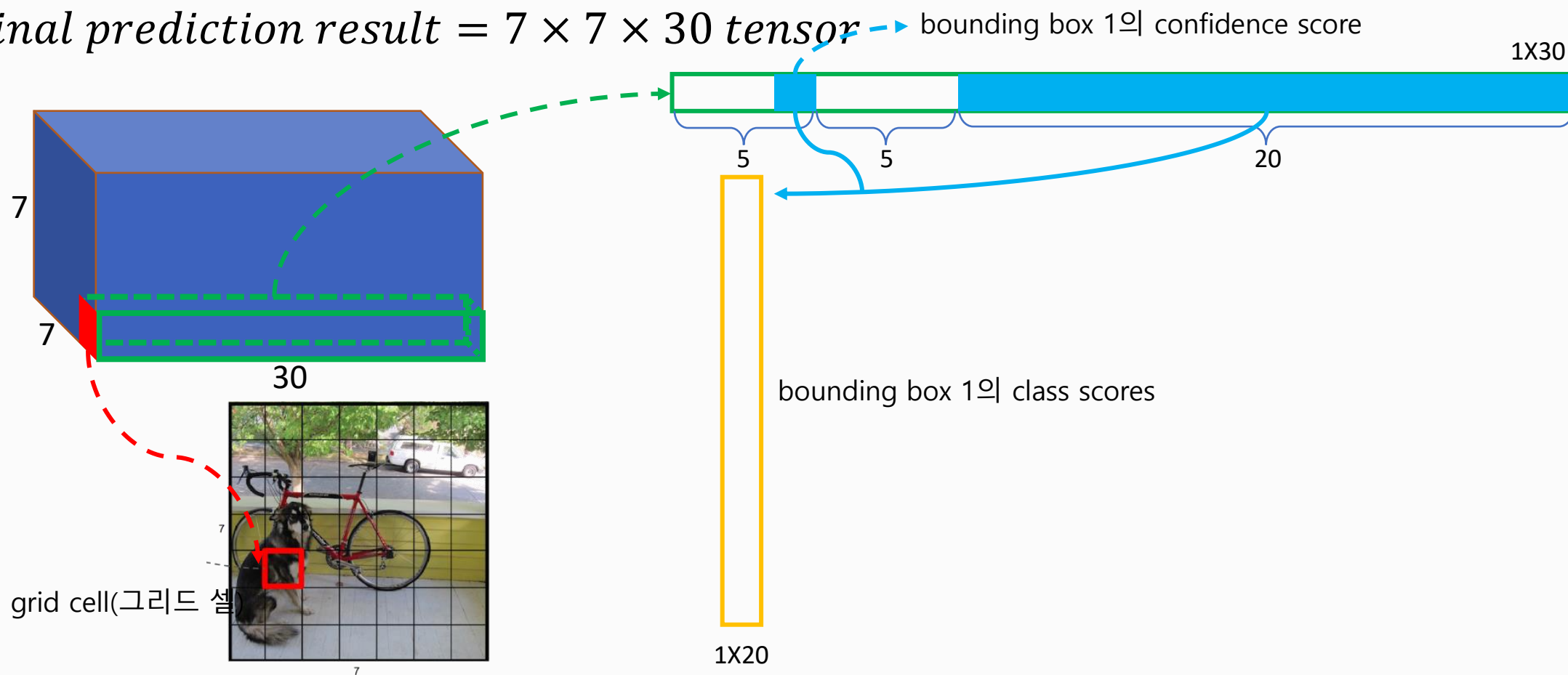
*Final prediction result =  $7 \times 7 \times 30$  tensor*



# YOLO란?

## YOLO 작동 원리

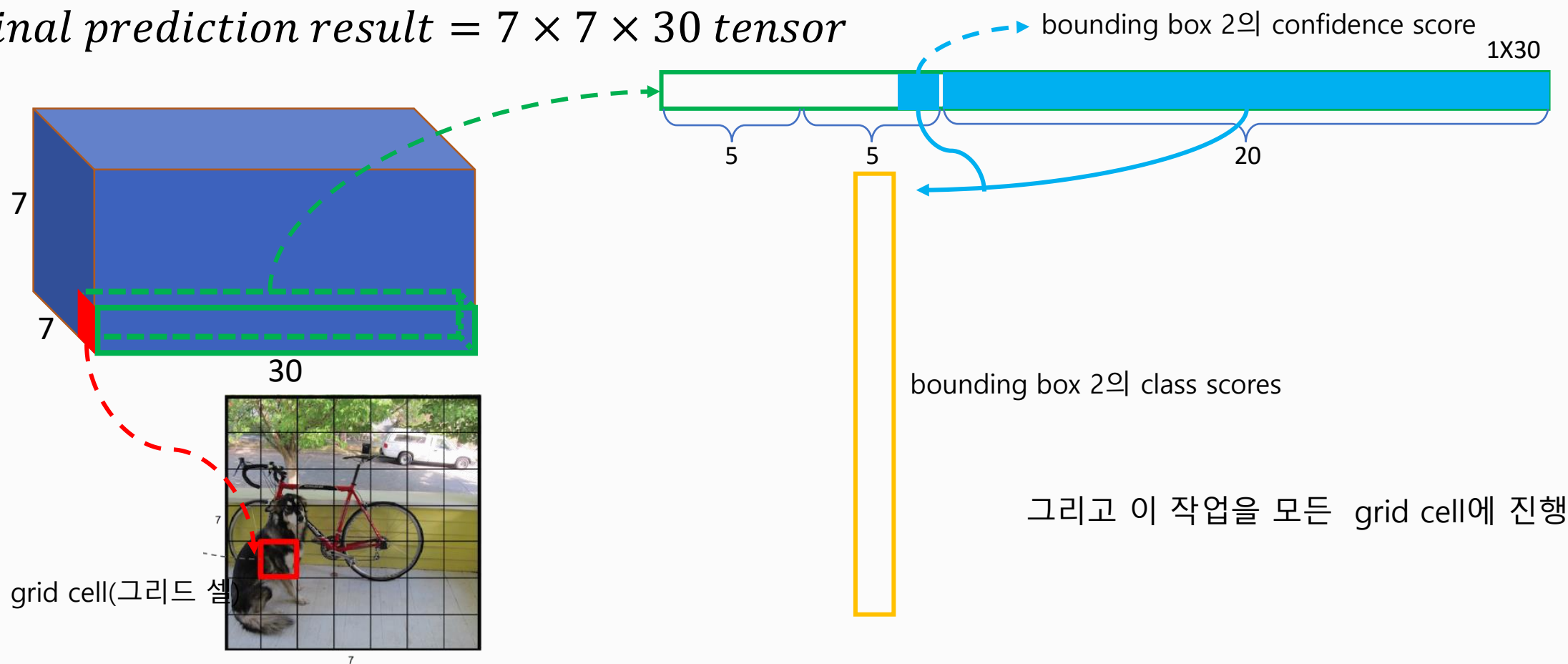
*Final prediction result =  $7 \times 7 \times 30$  tensor*



# YOLO란?

## YOLO 작동 원리

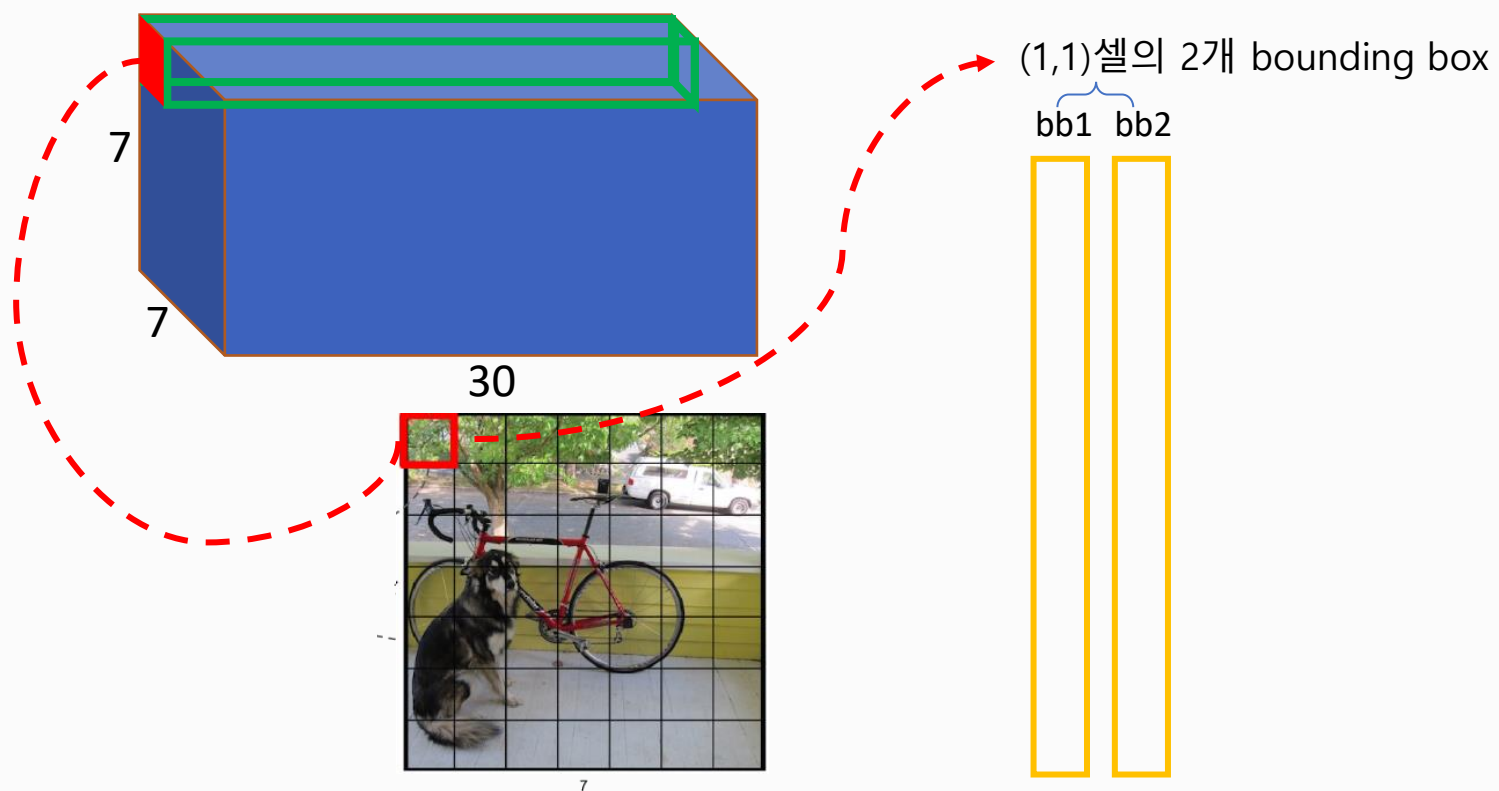
*Final prediction result =  $7 \times 7 \times 30$  tensor*



# YOLO란?

## YOLO 작동 원리

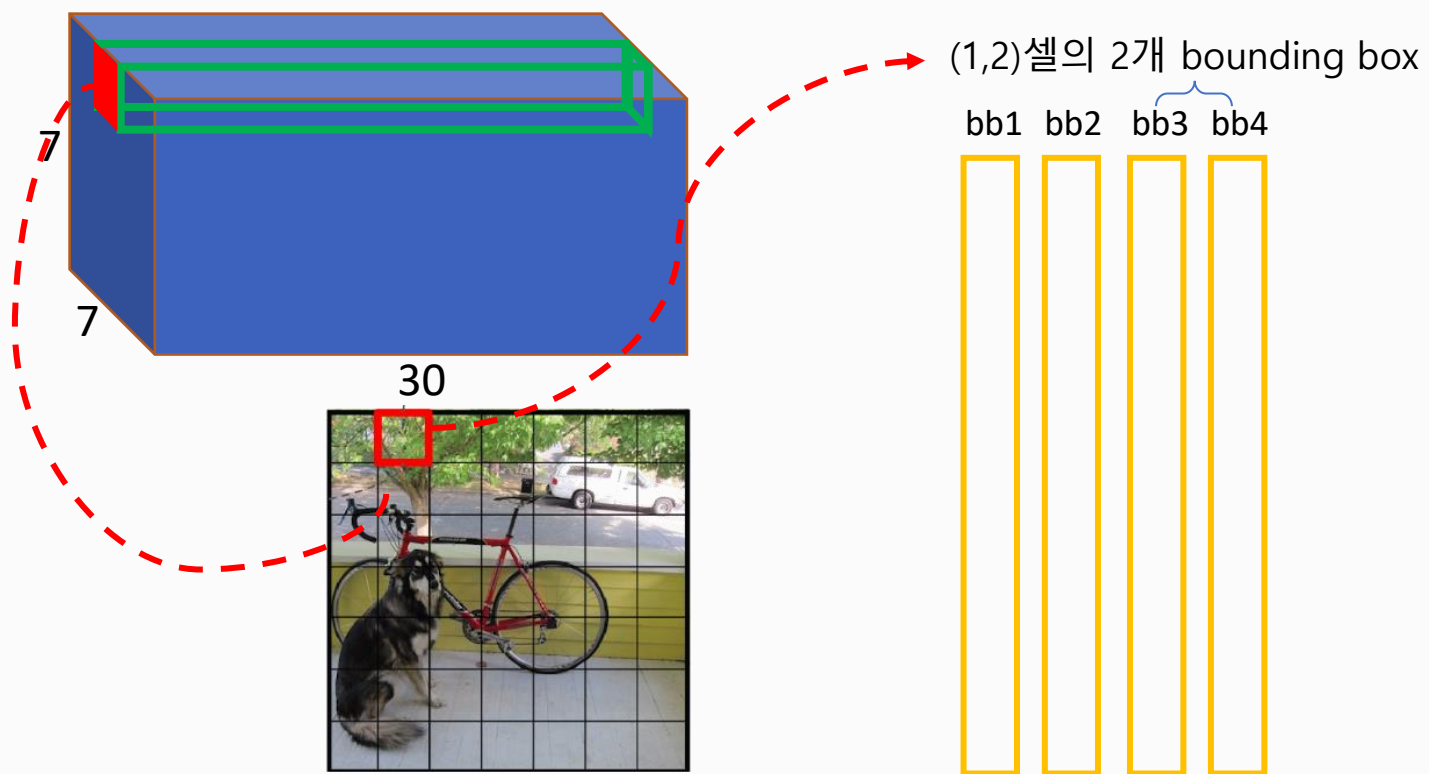
*Final prediction result =  $7 \times 7 \times 30$  tensor*



# YOLO란?

## YOLO 작동 원리

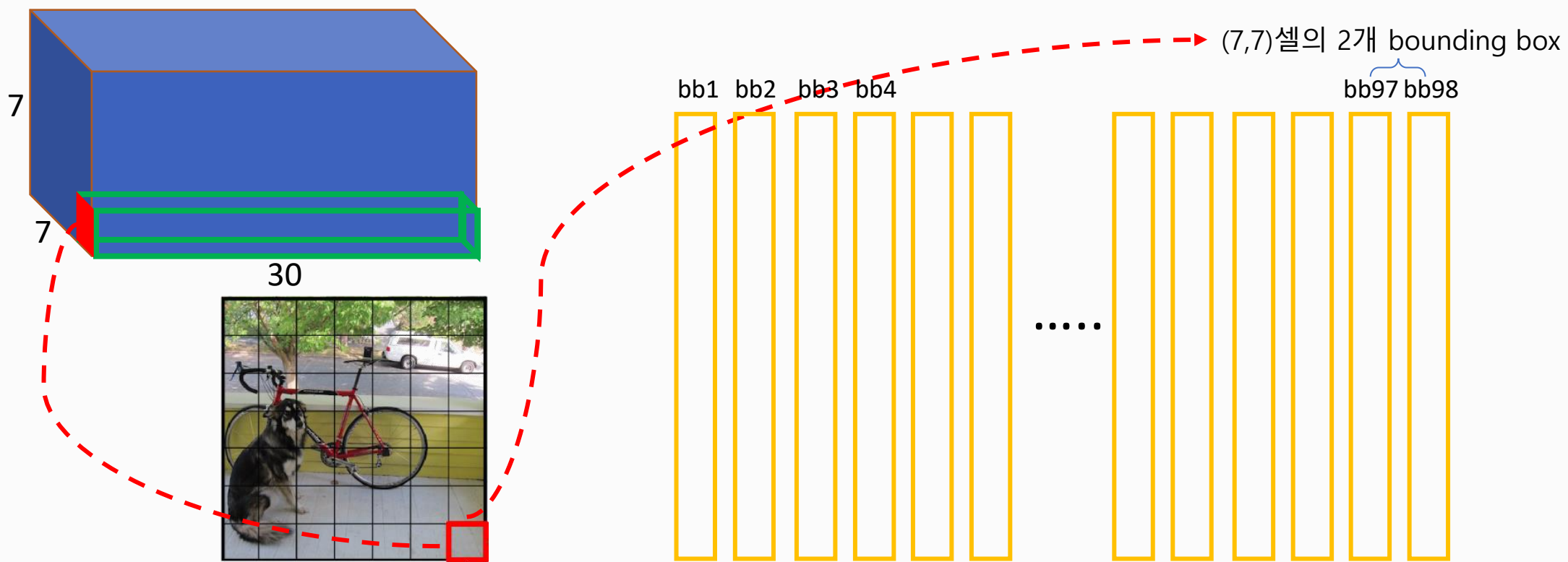
*Final prediction result =  $7 \times 7 \times 30$  tensor*



# YOLO란?

## YOLO 작동 원리

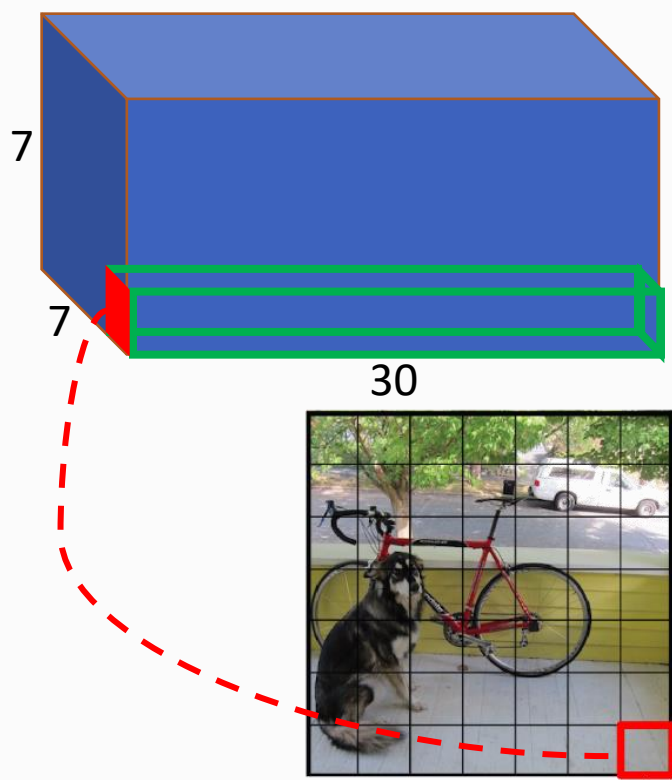
*Final prediction result =  $7 \times 7 \times 30$  tensor*



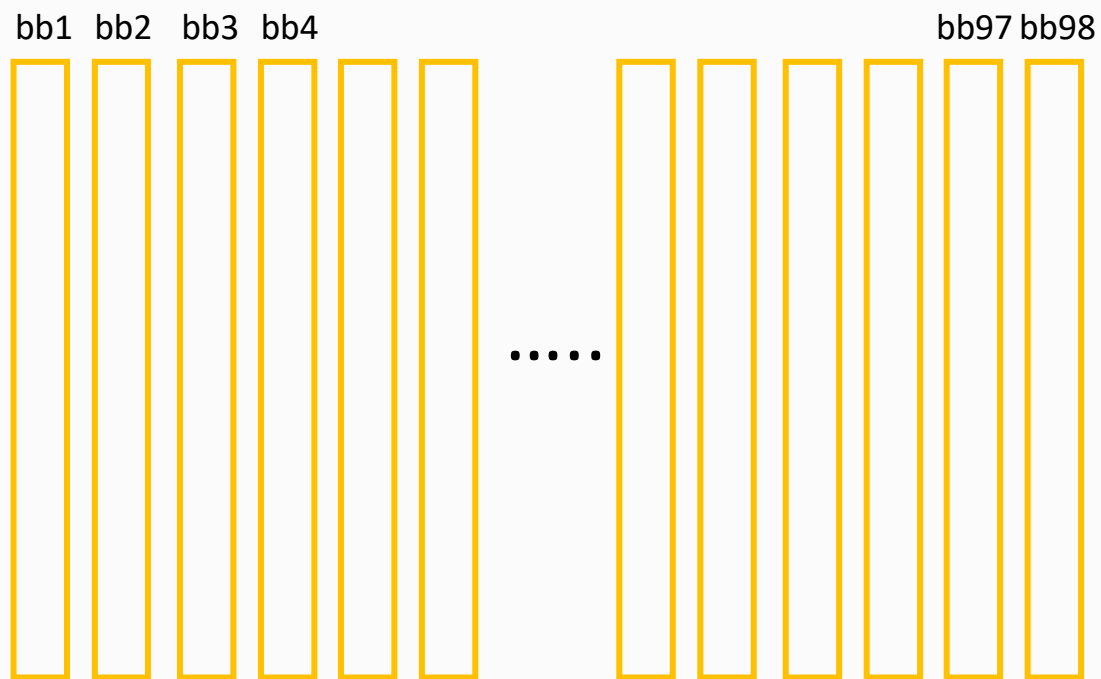
# YOLO란?

## YOLO 작동 원리

*Final prediction result =  $7 \times 7 \times 30$  tensor*



총 개수 :  $7 \times 7 \times 2 = 98$  bounding box

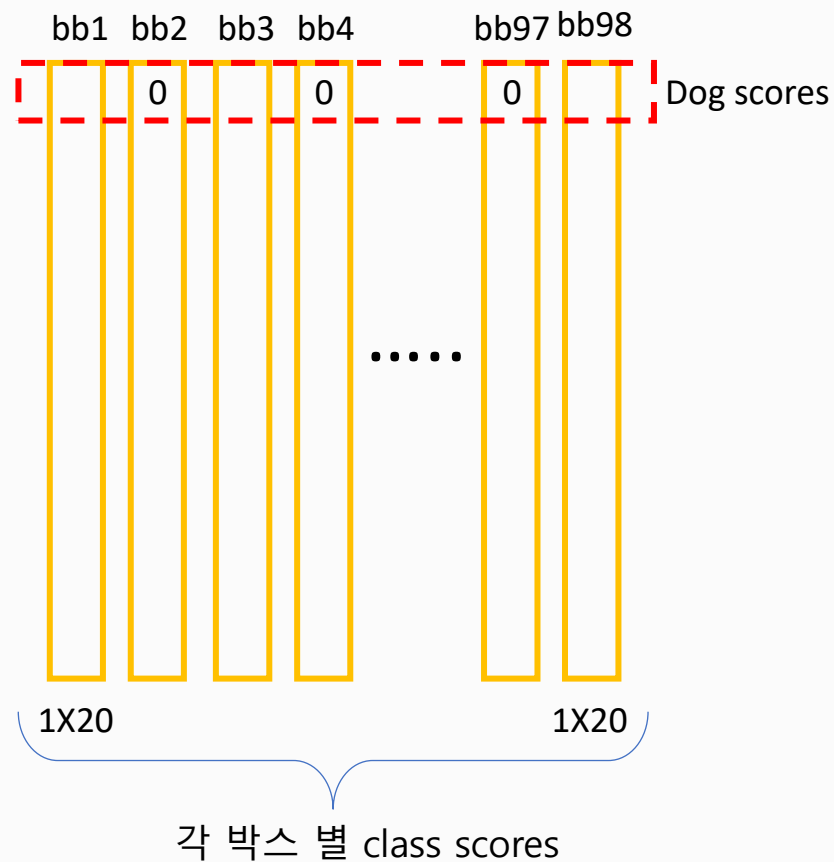




# YOLO란?

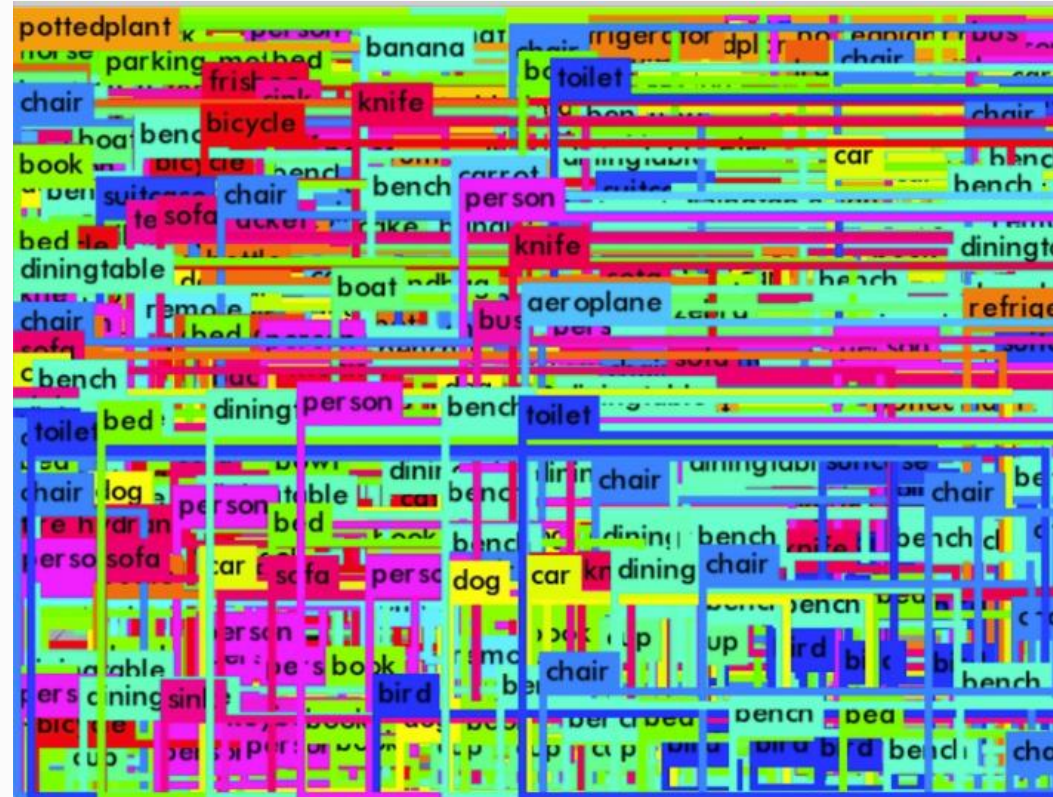
## YOLO 작동 원리

- ① Thresh holding (스레시 홀드)  
만약  $\text{score} < \text{thresh1} (0.2)$  라면  
→ 0으로 치환



스레시 홀드  
→ 여러 값을 어떤 임계점을 기준으로 두 가지 부류로 나누는 방법

## YOLO 작동 원리

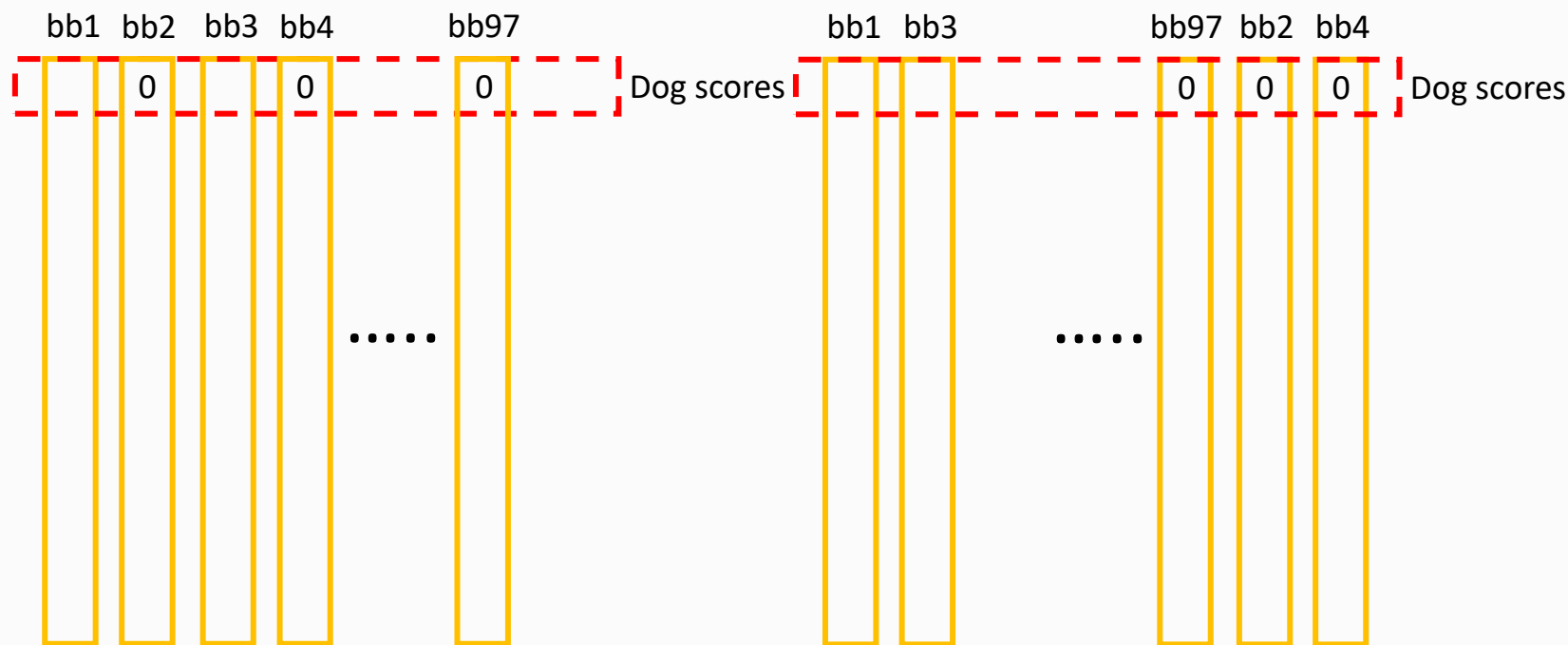


클래스 20개의 점수가 0.2보다 작으면 그건 해당 클래스가 아니라고 판단

# YOLO란?

## YOLO 작동 원리

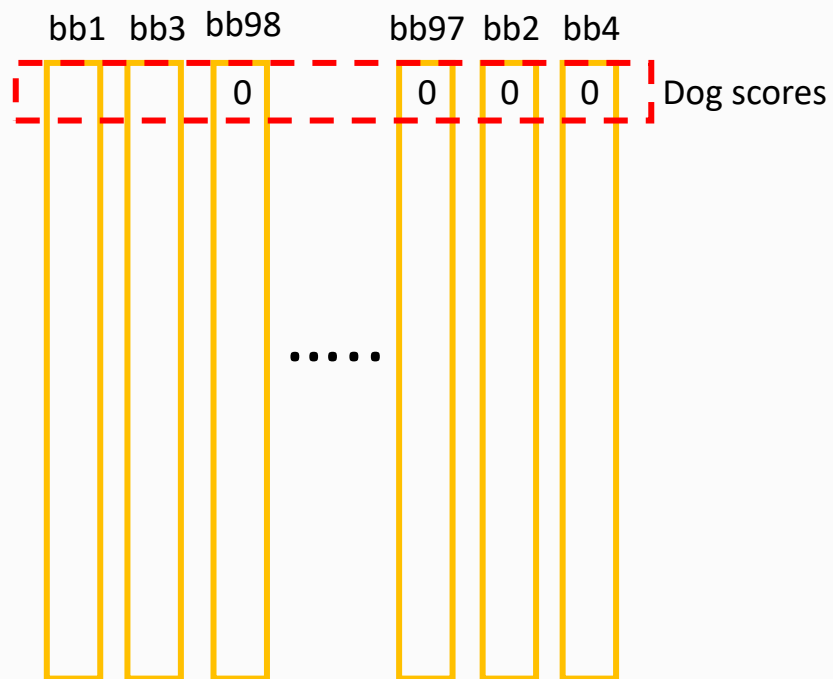
- ② Sort descending(내림차순)  
높은 score부터 차례대로 정렬



# YOLO란?

## YOLO 작동 원리

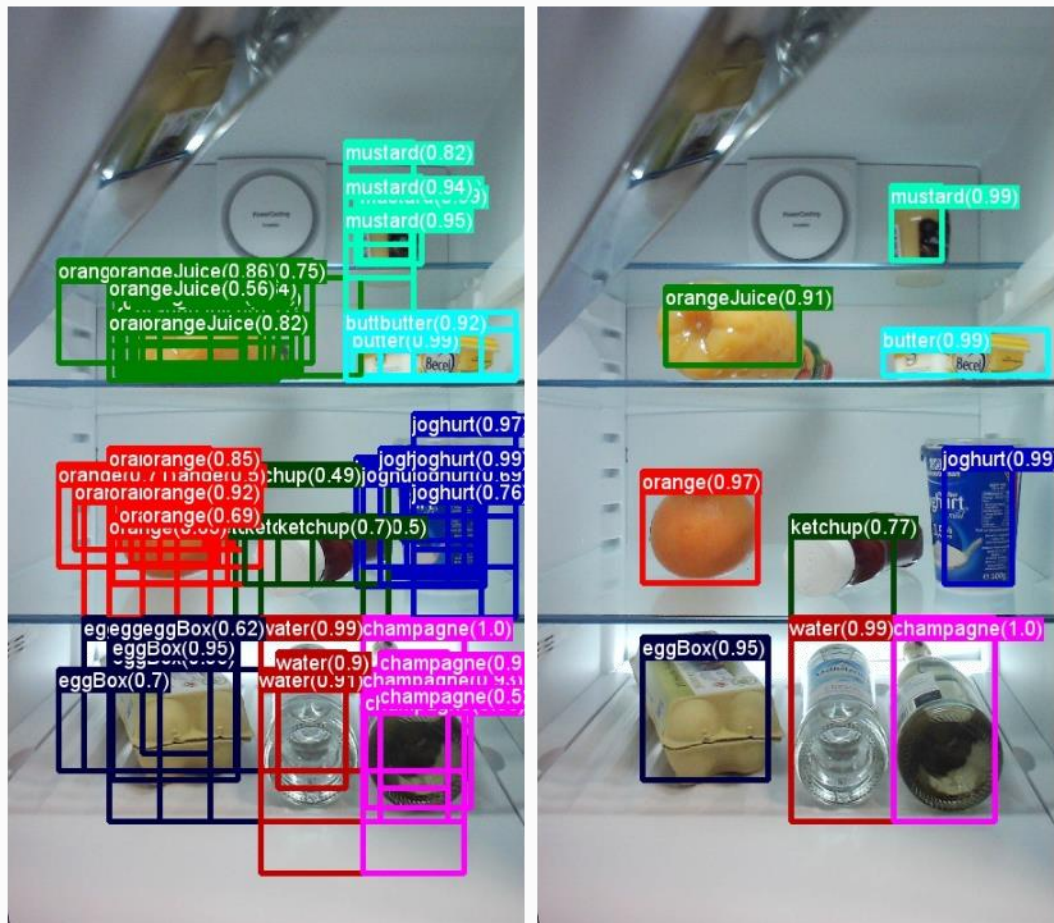
③ NMS 알고리즘  
NMS알고리즘에 따라서  
0으로 치환



HOW IT WORKS??

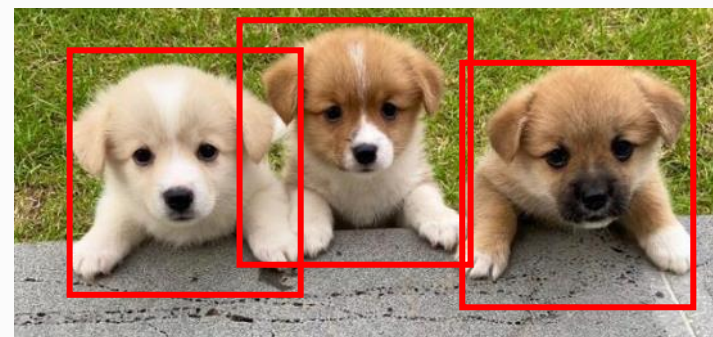
# YOLO란?

## NMS 알고리즘



### Non – Maximum Suppression : intuition

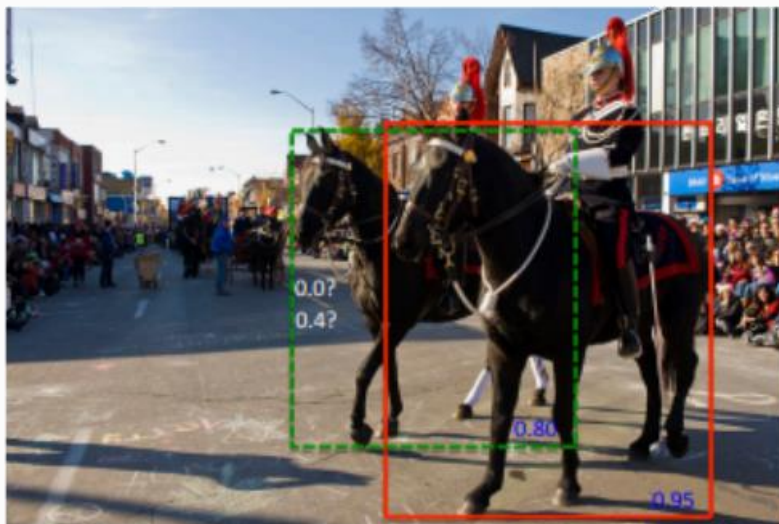
- NMS 알고리즘의 목적  
- 이미지 내에서 중복되는 오브젝트를 출력해주기 위함!
- 중복이 되는 경계 박스들을 제거
- 여러 경계 박스가 겹쳐있을 때 최대 값을 갖는 오브젝트를 제외하고 모두 지움
- 테스트 결과 정확도가 2~3%상승



# YOLO란?

## NMS 알고리즘

- 하나의 임계치  $N$ 에 대해서 수행하기 때문에  $N$ 을 어떻게 선택하는가에 따라 성능 평가 결과가 달라짐
- Bounding box 수의 비해 물체의 수가 현저히 적으므로 최종 average precision 이 낮아짐
- 같은 클래스의 물체가 겹쳐 있을 경우 각 물체에 대해 bounding box를 잡아야 함에도 불구하고 NMS에 의해 작은 score을 가진 bounding box가 무시된다





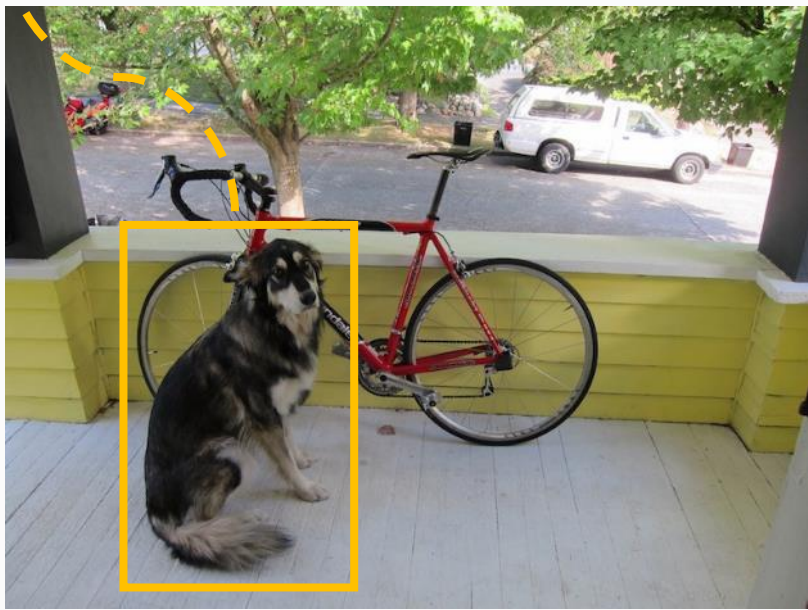
# YOLO란?

## NMS 알고리즘

### Non – Maximum Suppression : intuition

각 bounding box에서 class(Dogs)의 score

class: dog	bb47 bb20 bb15 bb7										bb1 bb4 bb8 bb98				1x98
	0.5	0.3	0.2	0.1							0	0	0	0	



## NMS 알고리즘

## Non – Maximum Suppression : intuition

1

bb47 bb20 bb15 bb7										bb1	bb4	bb8	bb98
0.5	0.3	0.2	0.1							0	0	0	0





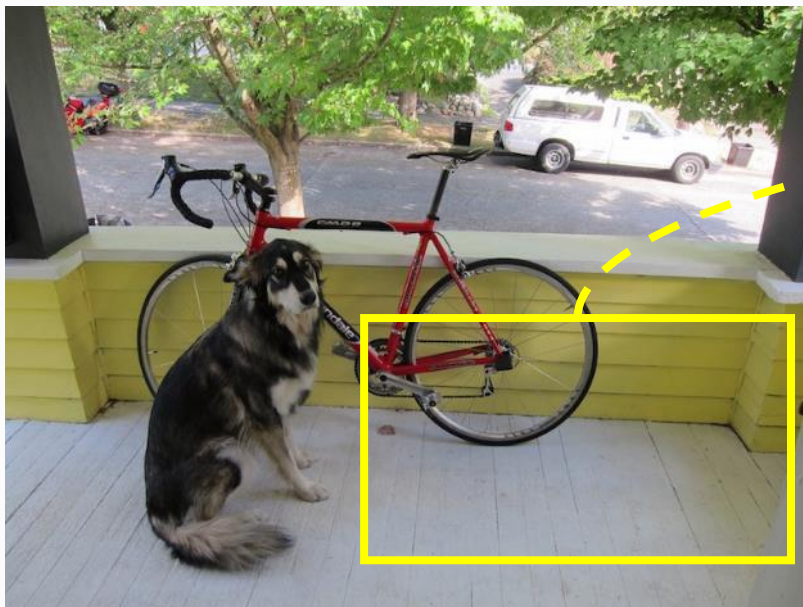
# YOLO란?

## NMS 알고리즘

### Non – Maximum Suppression : intuition

각 bounding box에서 class(Dogs)의 score

class: dog	bb47 bb20 bb15 bb7																				bb1 bb4 bb8 bb98				1x98
	0.5	0.3	0.2	0.1													0	0	0	0					

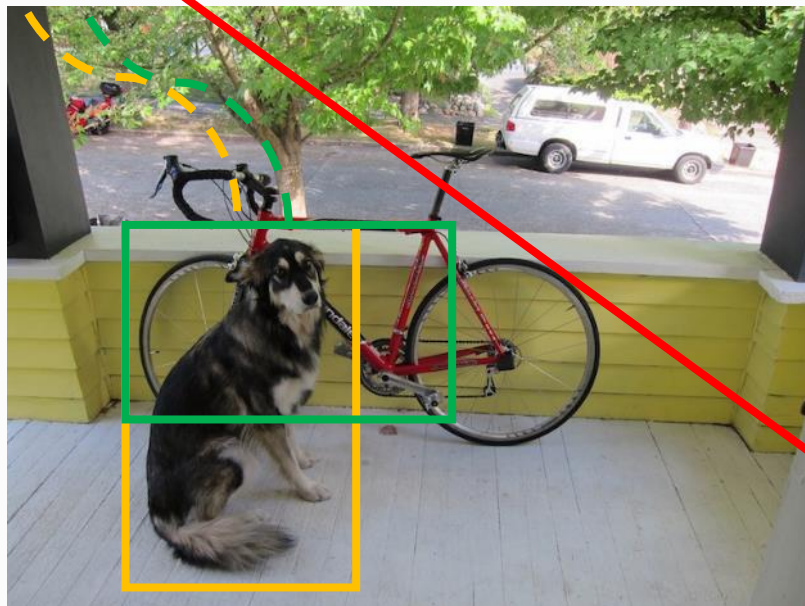


## NMS 알고리즘

## Non – Maximum Suppression : intuition

각 bounding box에서 class(Dogs)의 score

	bb47	bb20	bb15	bb7					bb1	bb4	bb8	bb98
class: dog	0.5	0.1	0.2	0.1					0	0	0	0



가장 높은 score을 가진 boundingbox

→ **bbox\_max**

0이 아닌 score를 가지는 다른 boundingbox(비교대상)

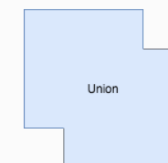
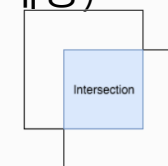
→ **bbox** **cur**

만약  $\text{IOU}(\text{bbox\_max}, \text{bbox\_cur}) > 0.5$  이라면

→ **bbox** **cur** 을 0으로 치환

$$\text{IoU} =$$

IOU > 0.5 이므로 0으로 치환



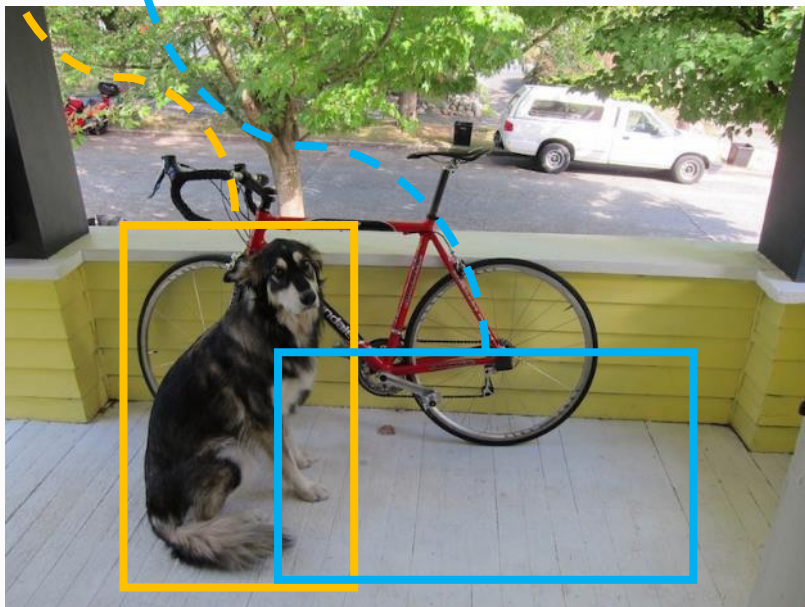
# YOLO란?

## NMS 알고리즘

### Non – Maximum Suppression : intuition

각 bounding box에서 class(Dogs)의 score

										bb1	bb4	bb8	bb98		
class: dog	bb47	bb20	bb15	bb7						0	0	0	0	1x98	
	0.5	0.1	0.2	0.1											



가장 높은 score를 가진 boundingbox

→ **bb47**

0이 아닌 score를 가지는 다른 boundingbox(비교대상)

→ **bb20**

만약  $\text{IOU}(\text{bb47}, \text{bb20}) > 0.5$  이라면

→ **bb20** 을 0으로 치환

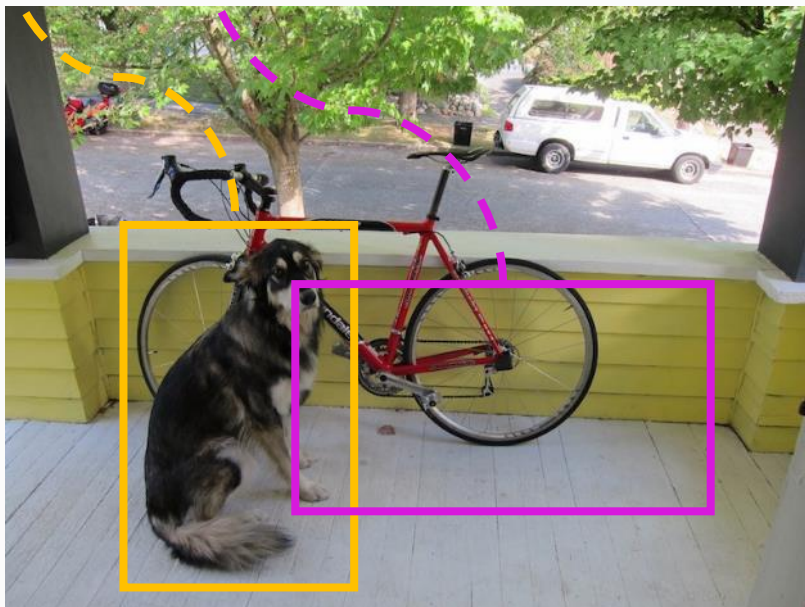
IOU < 0.5 이므로 값을 그대로 둠

## NMS 알고리즘

## Non – Maximum Suppression : intuition

각 bounding box에서 class(Dogs)의 score

	bb47	bb20	bb15	bb7					bb1	bb4	bb8	bb98
class: dog	0.5	0.1	0.2	0.1					0	0	0	0



가장 높은 score을 가진 boundingbox

→ **bbox\_max**

0이 아닌 score를 가지는 다른 boundingbox (비교대상)

→ **bbox\_cur**

만약  $\text{IOU}(\text{bbox\_max}, \text{bbox\_cur}) > 0.5$  이라면

→ **bbox\_cur** 을 0으로 치환

IOU < 0.5 이므로 값을 그대로 둠

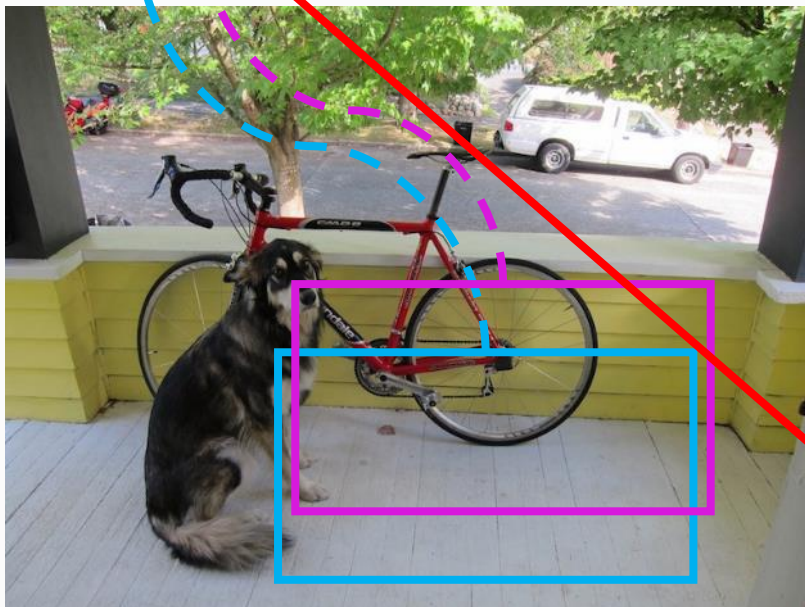
## NMS 알고리즘

## Non – Maximum Suppression : intuition

각 bounding box에서 class(Dogs)의 score

	bb47	bb20	bb15	bb7									bb1	bb4	bb8	bb98
class: dog	0.5	0.	0.2	0.									0	0	0	0

1x98



그 다음으로 가장 높은 score을 가진 boundingbox  
→ **bbox\_max**

0이 아닌 score를 가지는 다른 boundingbox(비교대상)  
→ **bbox\_cur**

만약  $\text{IOU}(\text{bbox\_max}, \text{bbox\_cur}) > 0.5$  이라면  
 →  $\text{bbox\_cur}$  을 0으로 치환

IOU > 0.5 이므로 0으로 치환

**이 모든 과정을 더 이상 할 수 없을 때까지 반복**



## NMS 알고리즘

## Non – Maximum Suppression : intuition

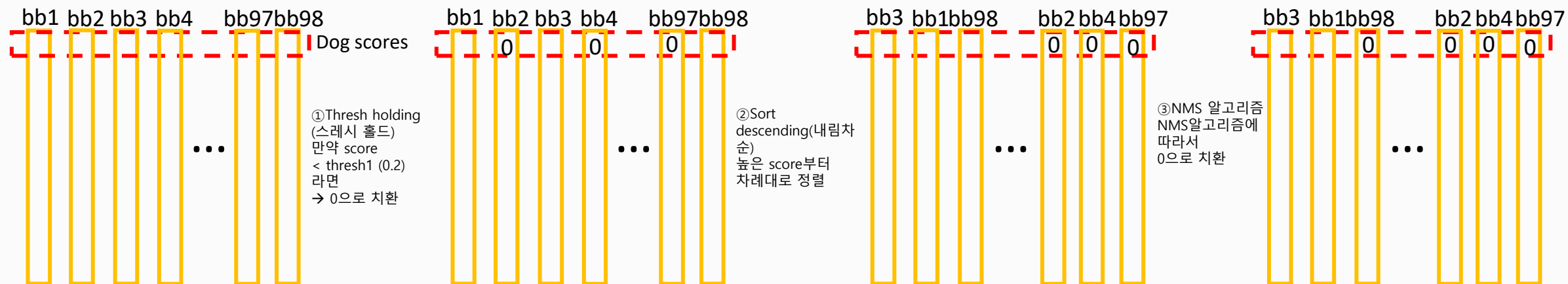
각 bounding box에서 class(Dogs)의 score

[illegible]

최종적으로 2개의 bounding box만 남음

# YOLO란?

## YOLO 순서

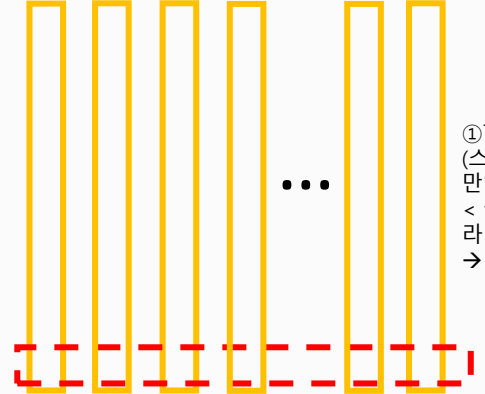


이 과정을 다른 모든 class 에 맞춰 진행

# YOLO란?

## YOLO 순서

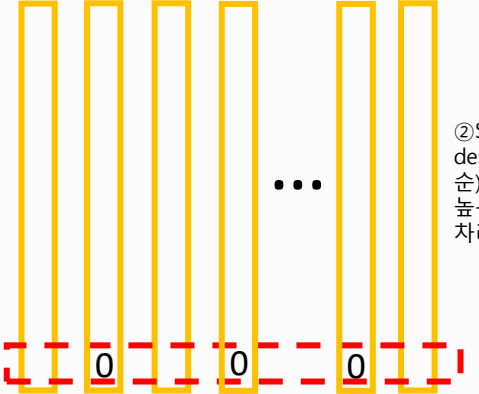
bb1 bb2 bb3 bb4 ... bb97 bb98



human scores

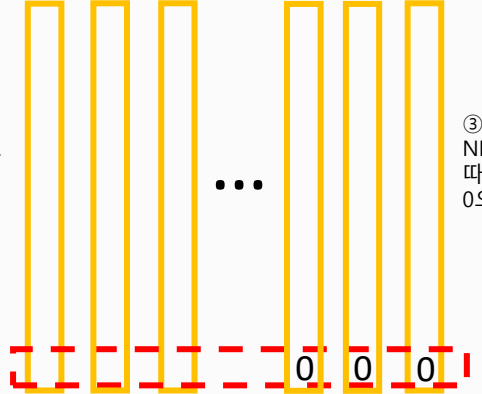
①Thresh holding  
(스레시 홀드)  
만약 score  
< thresh1 (0.2)  
라면  
→ 0으로 치환

bb1 bb2 bb3 bb4 ... bb97 bb98



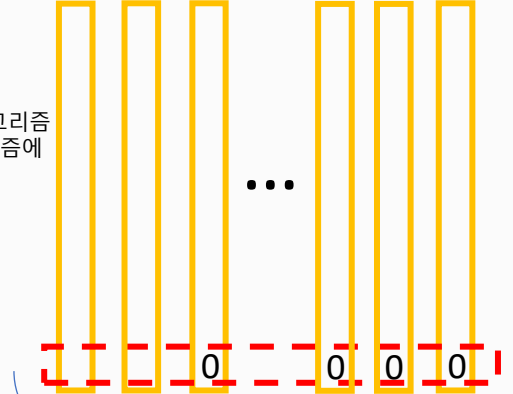
②Sort  
descending(내림차  
순)  
높은 score부터  
차례대로 정렬

bb3 bb1 bb98 ... bb2 bb4 bb97



③NMS 알고리즘  
NMS알고리즘에  
따라서  
0으로 치환

bb3 bb1 bb98 ... bb2 bb4 bb97

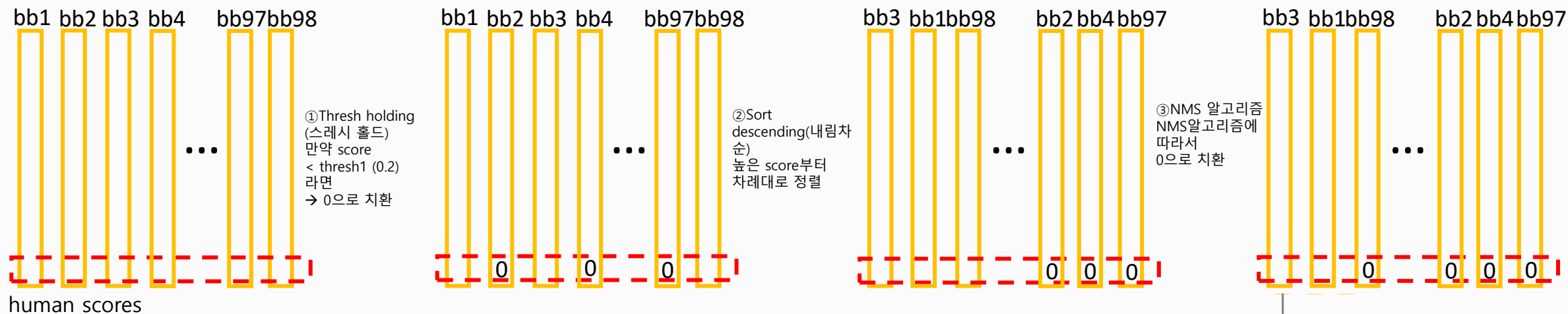


모든 클래스에 대해 진행하고 나면 무수히 많은 0이 있게 됨



# YOLO란?

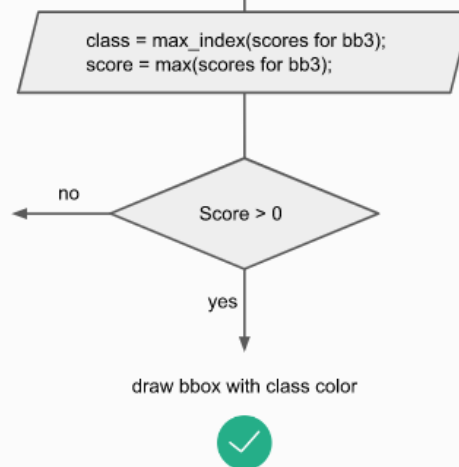
## YOLO 순서



class score에 따라서 나타낼 bbox를 선택함

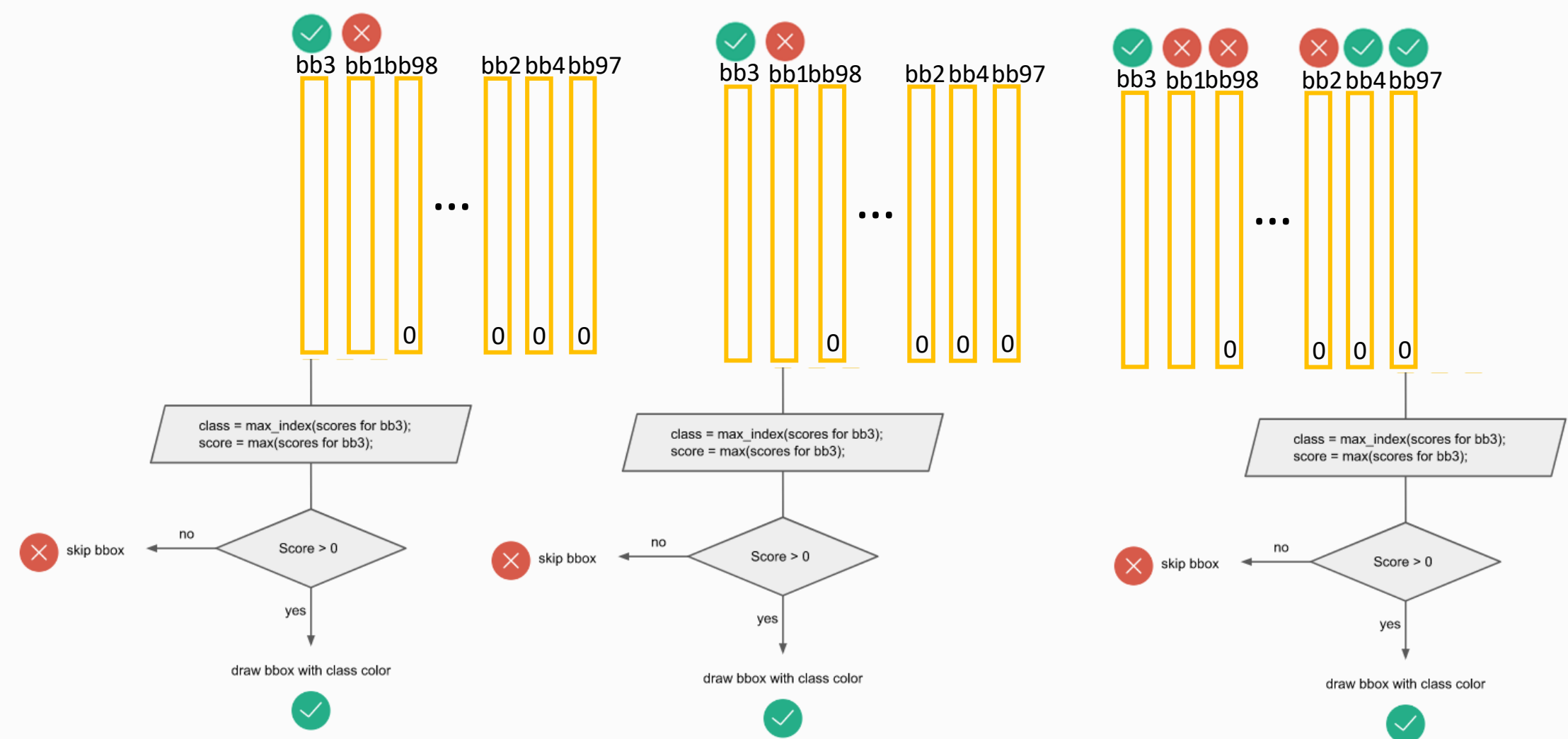
class score이 0보다 크다면 → 나타내기  
class score이 0이라면 → 삭제

skip bbox



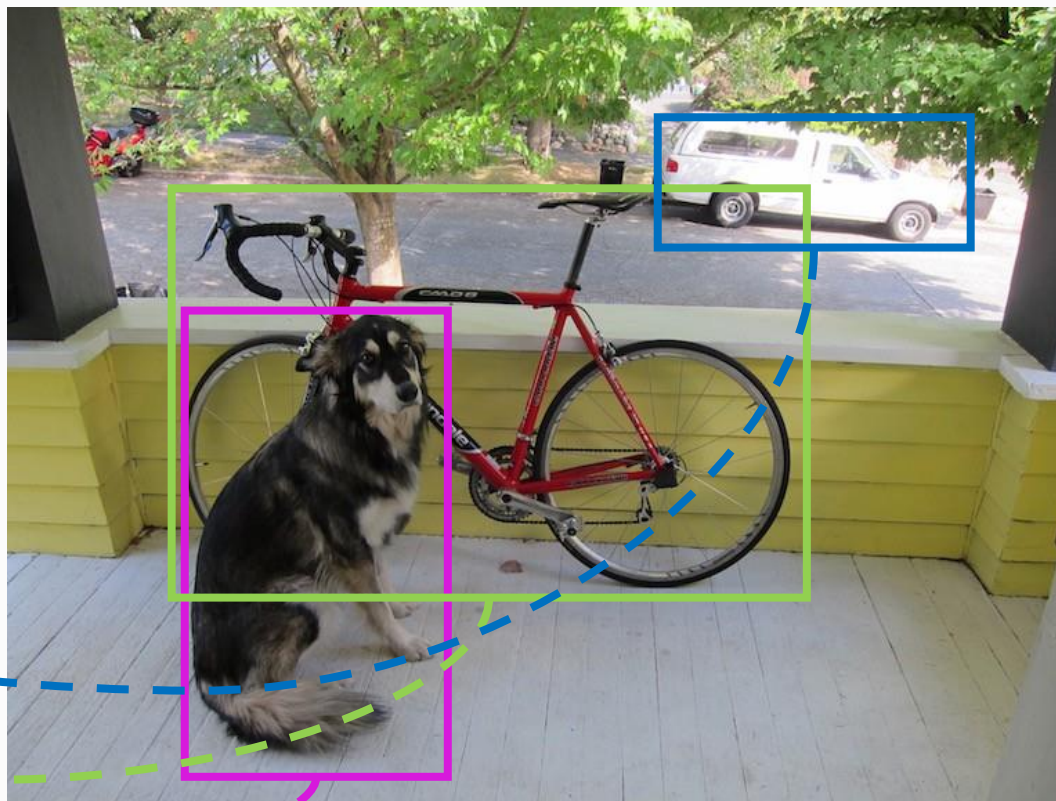
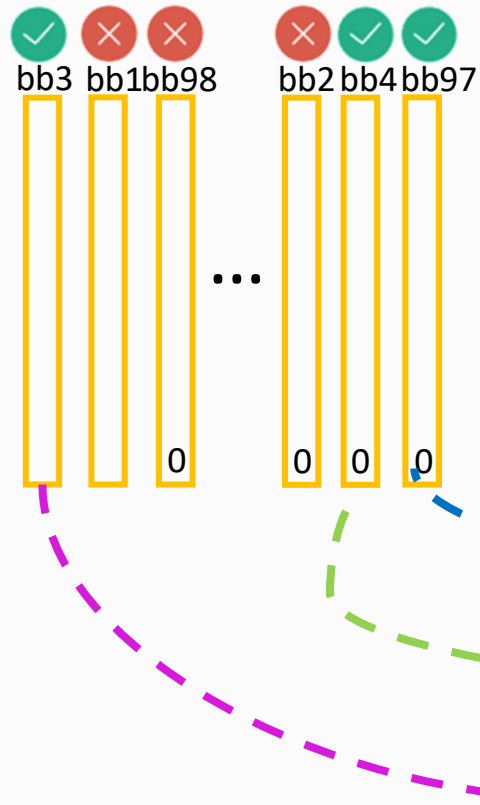
# YOLO란?

## YOLO 순서



# YOLO란?

## YOLO 순서





## 국방데이터에 적용



YOLO v5

스마트 지뢰탐지 시스템

# 국방데이터에 적용

## YOLO v5실습 : 적용해보기

```

▶ git clone https://github.com/ultralytics/yolov5 # clone
  %cd yolov5
  %pip install -qr requirements.txt # install 설치 완료 실행할 수 있는 위치가 다른듯

Cloning into 'yolov5'...
remote: Enumerating objects: 12190, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (60/60), done.
remote: Total 12190 (delta 31), reused 26 (delta 6), pack-reused 12124
Receiving objects: 100% (12190/12190), 12.63 MiB | 23.57 MiB/s, done.
Resolving deltas: 100% (8376/8376), done.
/content/yolov5
| 1.6 MB 5.0 MB/s

```

```

▶ import torch

# Model
model = torch.hub.load('ultralytics/yolov5', 'yolov5s') # 불러오기 or yolov5n - yolov5x6, custom

```

```

▶ import torch
import utils
display = utils.notebook_init() # checks

YOLOv5 🚀 v6.2-94-g1aea74c Python=3.7.13 torch=1.12.1+cu113 CUDA:0 (Tesla T4, 15110MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 37.4/166.8 GB disk)

```

```

▶ !python detect.py --weights yolov5s.pt --source 'https://www.youtube.com/watch?v=0mn-Yf6U6j4'... ## 손흥민 vs 칠레전

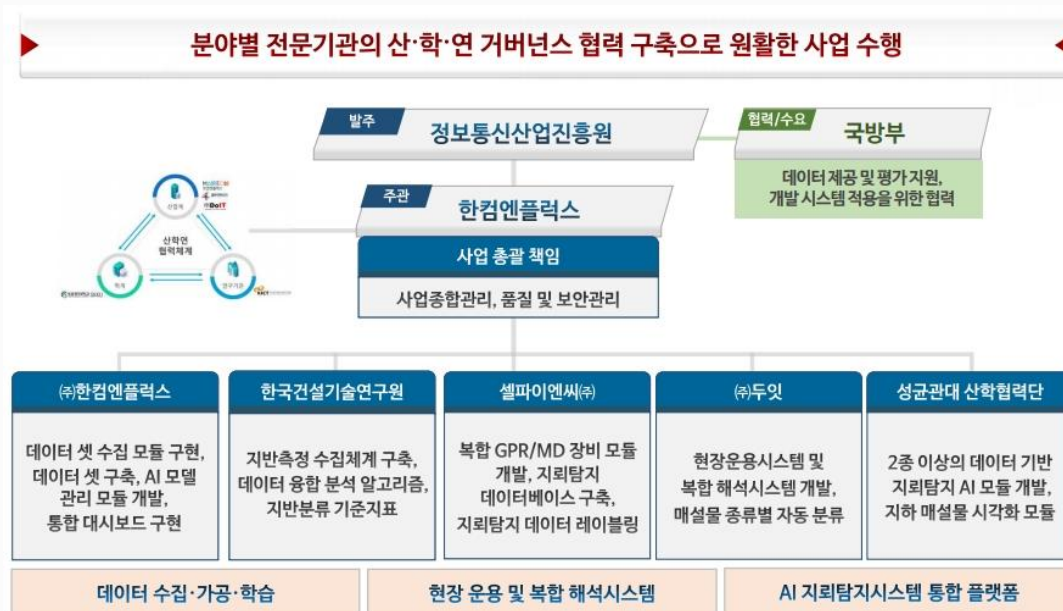
0: 384x640 21 persons, 1 sports ball, Done. (0.009s)
0: 384x640 21 persons, 1 sports ball, Done. (0.009s)
0: 384x640 21 persons, 1 sports ball, Done. (0.013s)
0: 384x640 21 persons, 1 sports ball, Done. (0.009s)
0: 384x640 19 persons, 1 sports ball, Done. (0.010s)

```

# 스마트 지뢰탐지 시스템

## 일반현황

- 사업명 : AI융합 지뢰탐지 시스템
- 사업기간 : '21. 7. ~ '23. 12
- 사업금액 : 66억원
- 주관기관 : NIPA
- 수행기관 : (주)한컴엔플렉스 – 셀파이엔씨 – Do It



# 발표 들어주셔서 감사합니다 :)

Q&A

Date

2022.09.06

Presenter

박정완 센터장