

# 深度學習 Pytorch手把手實作 物件偵測

黃志勝

義隆電子人工智慧研發部

國立陽明交通大學 AI學院 合聘助理教授

國立台北科技大學 電資學院 合聘助理教授



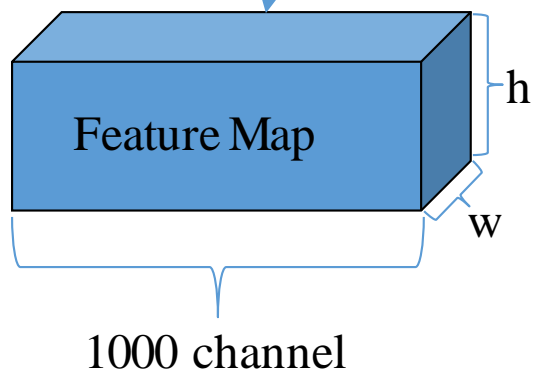
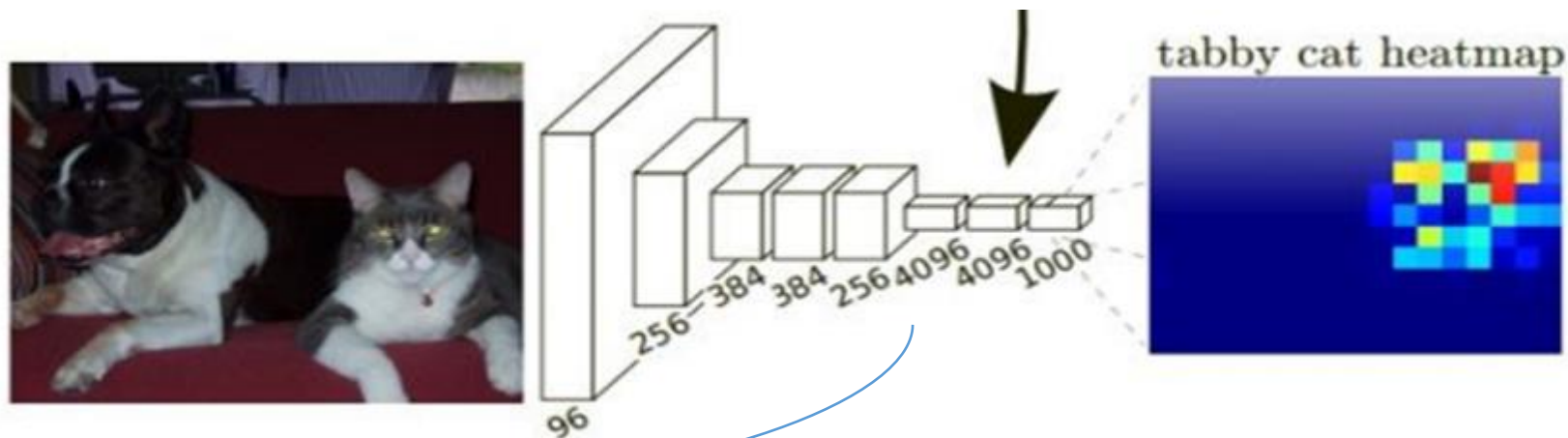
# Introduction

- Brief Introduction
- 直覺的方式建立自己的物件偵測模型
- YOLOv1



# Deep Learning: Feature Extractor

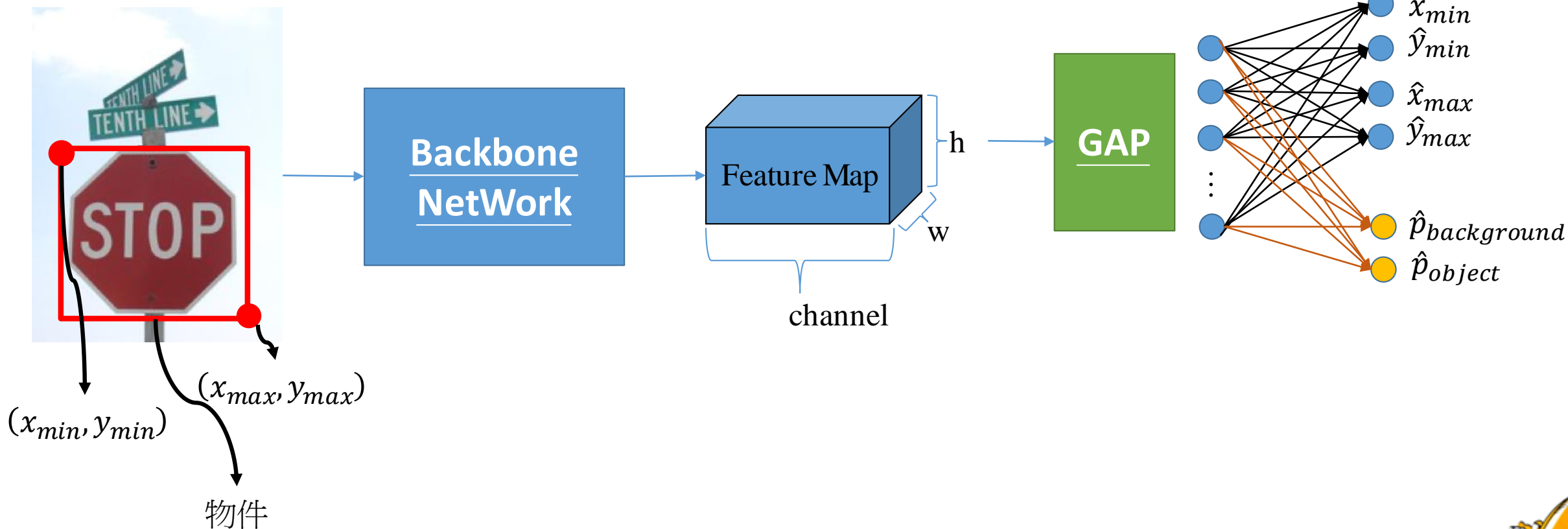
- Feature map可以做什麼？



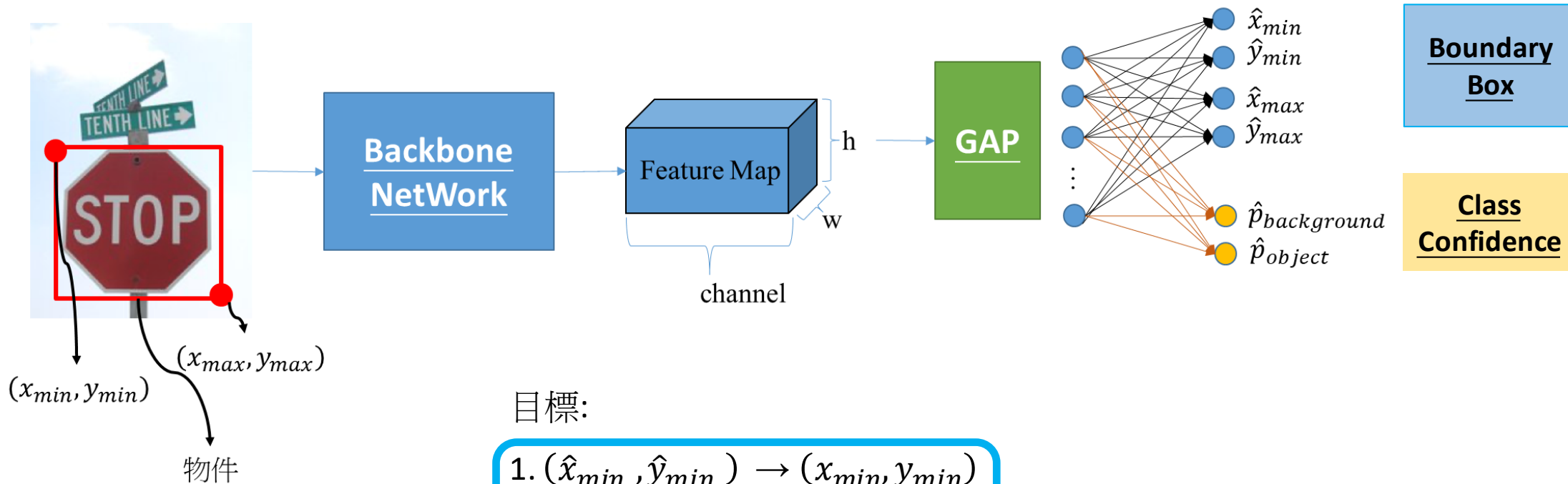
- 這張圖是貓還是狗 (Classification)
- 框出圖片內有貓和狗的位置 (Object detection)
- 從框出的物件(貓和狗)中的描繪出實際物件的輪廓 (Instance Segmentation)
- 把圖片描繪出物件(貓和狗)的輪廓 (Semantic Segmentation)



# 物件偵測



# 物件偵測

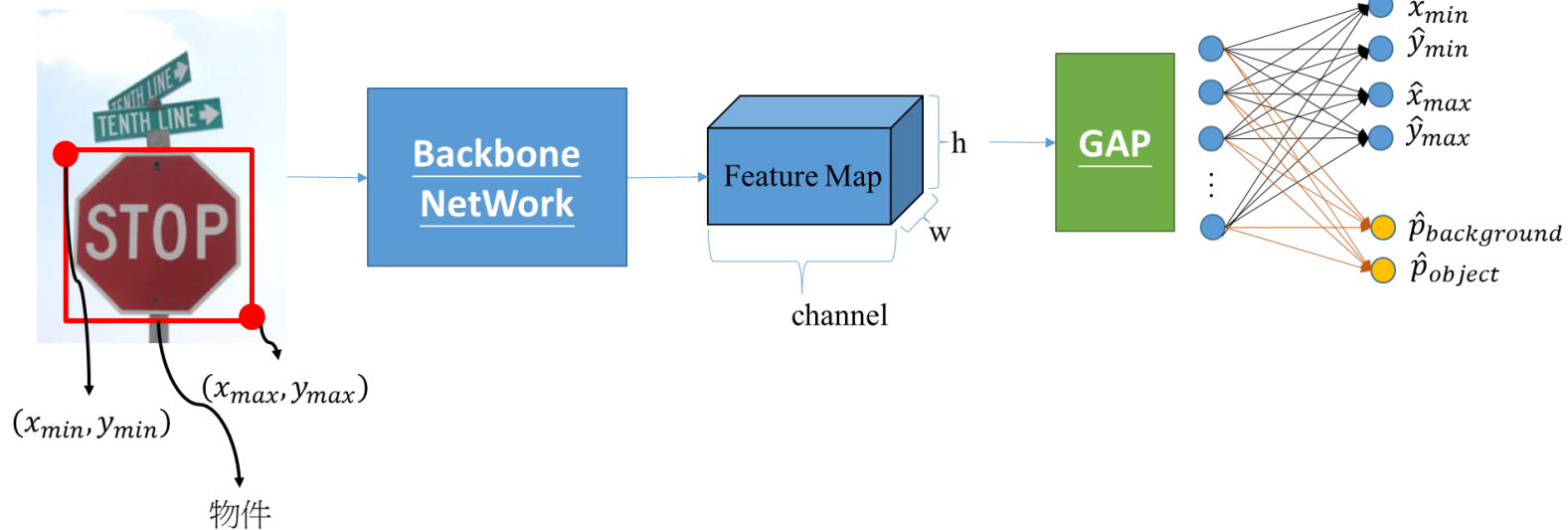


	Background	Object
$x$	$p_{background} = 0$	$p_{object} = 1$

最小化 {MSE (Regression)+Cross Entropy (Classification)}



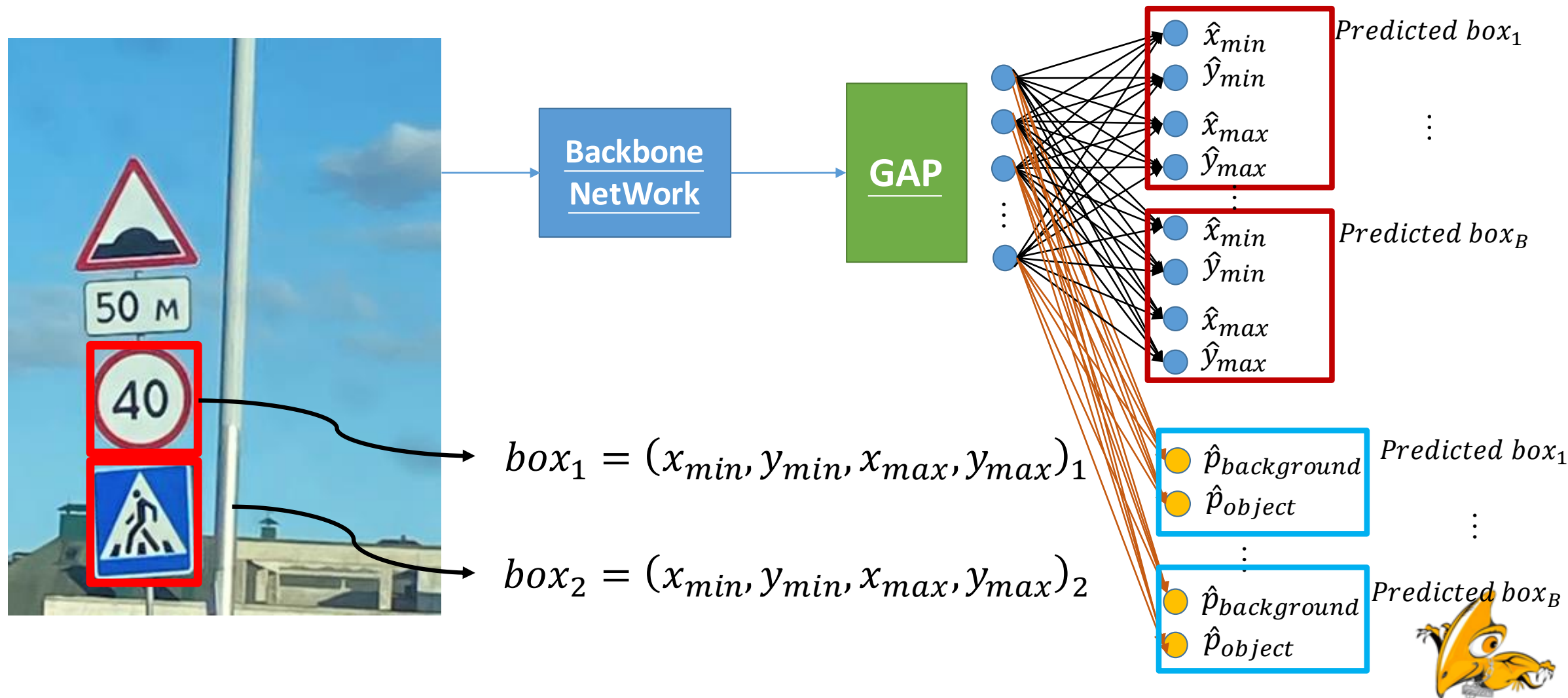
# 上述的架構會有問題



- 物件數超過1 ????

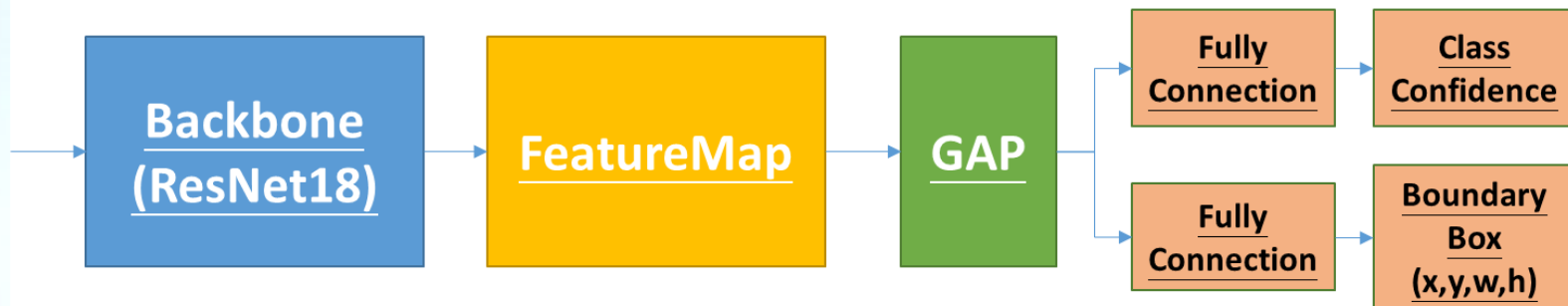


# 預測框增加到B個





# 物件偵測



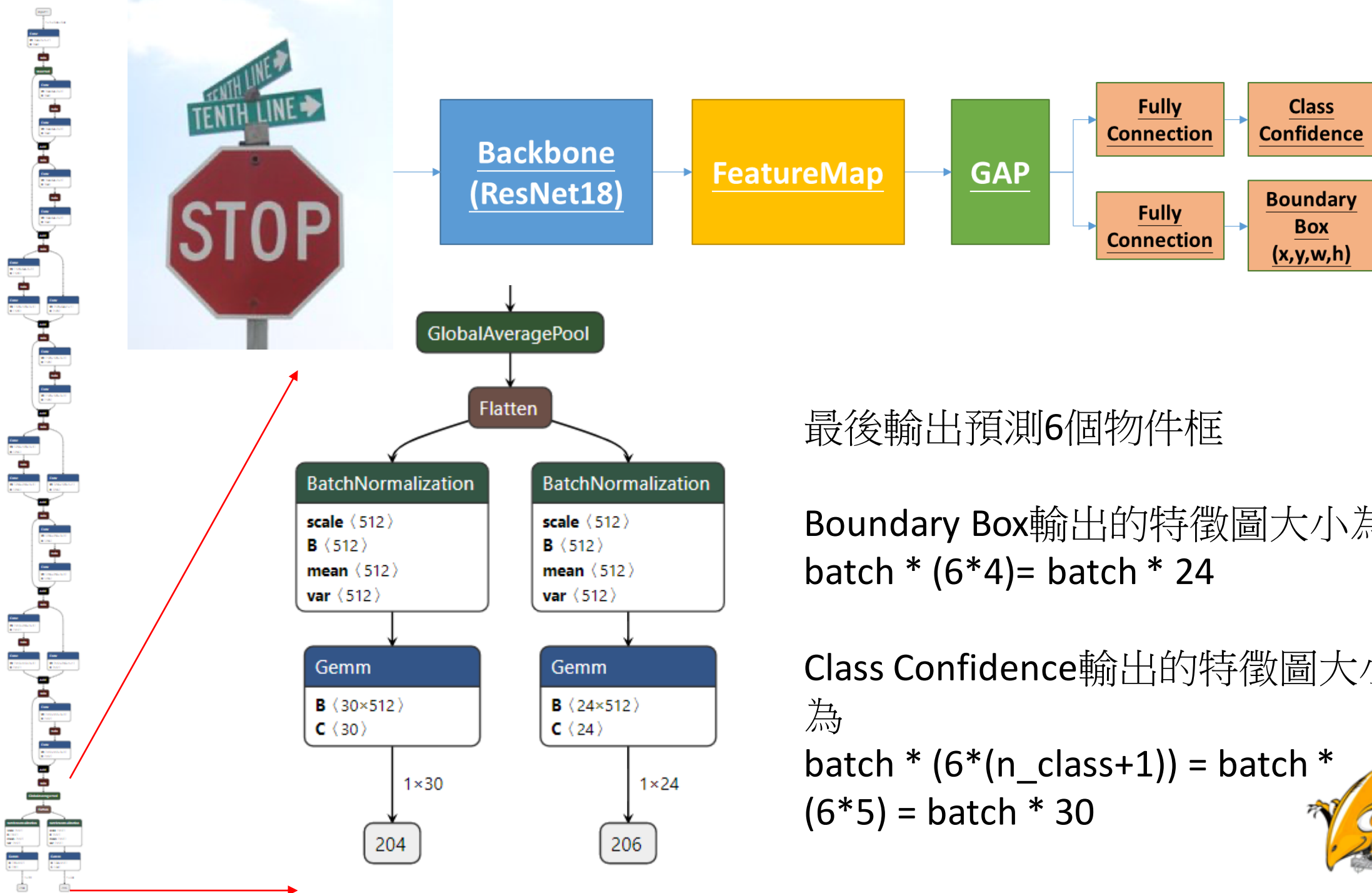
最後輸出預測6個物件框

Boundary Box輸出的特徵圖大小為  
 $\text{batch} * (6 * 4) = \text{batch} * 24$

Class Confidence輸出的特徵圖大小為  
 $\text{batch} * (6 * (n\_class + 1)) = \text{batch} * (6 * 5) = \text{batch} * 30$







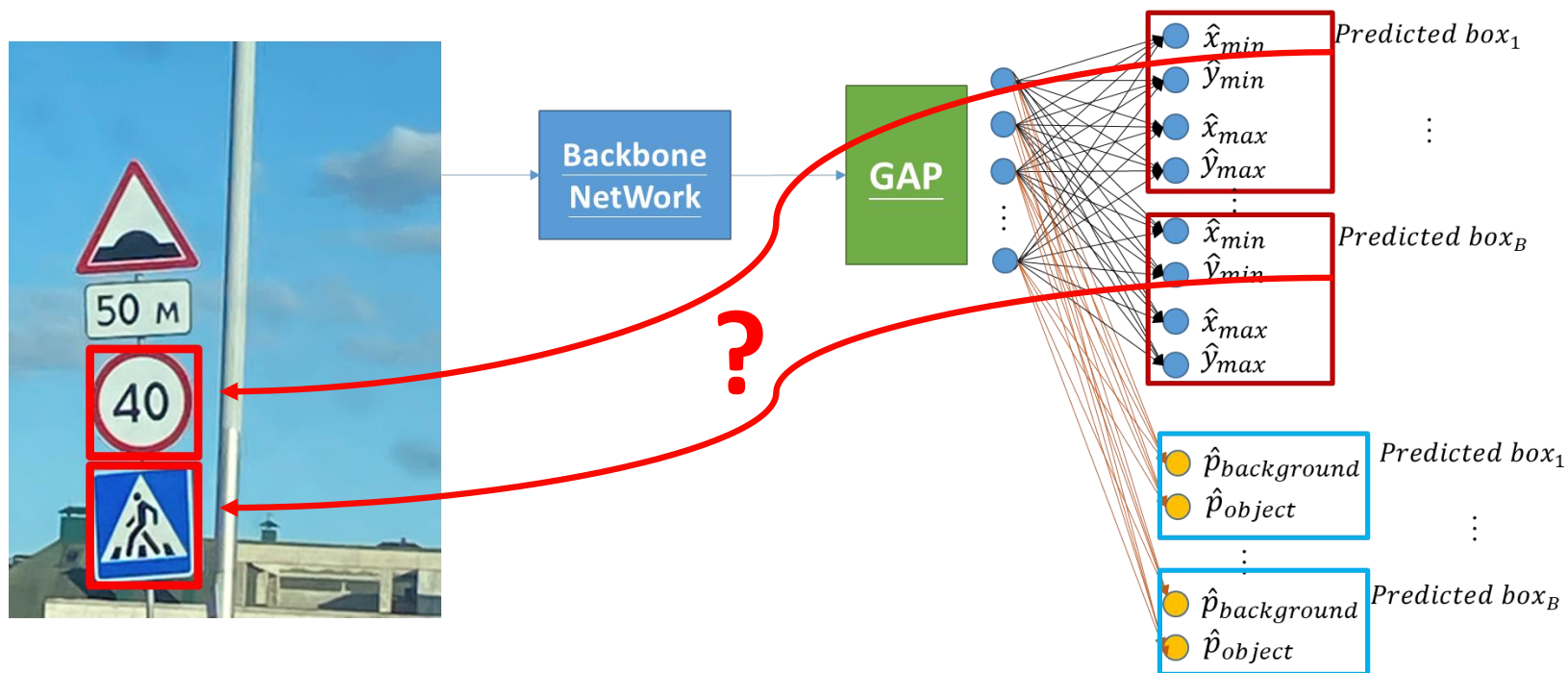
# Objection Detection

Pytorch 操作

Jupyter Notebook



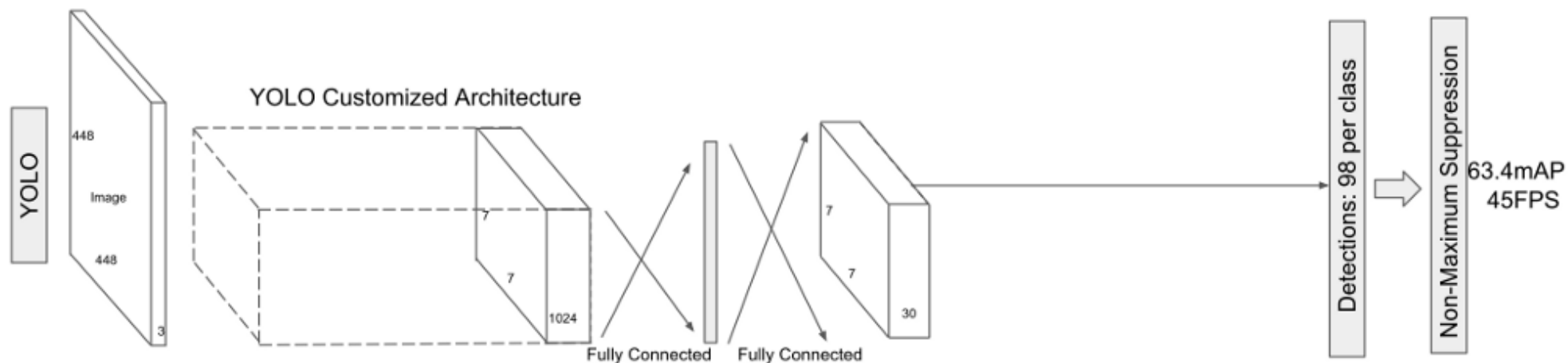
# 問題



學習的時候， $Predicted\ box_i, i = 1, 2, \dots, B$ 該預測哪一個物件框。  
造成混亂學習，所以會學不起來。



# YOLOv1的detector怎麼處理



最後一層是 $7 \times 7 \times 30$

$7 \times 7$ 就是的grid cell



$S \times S$  grid on input

$$30 = 2 \times 5 + 20$$

2: 2個Boundarybox

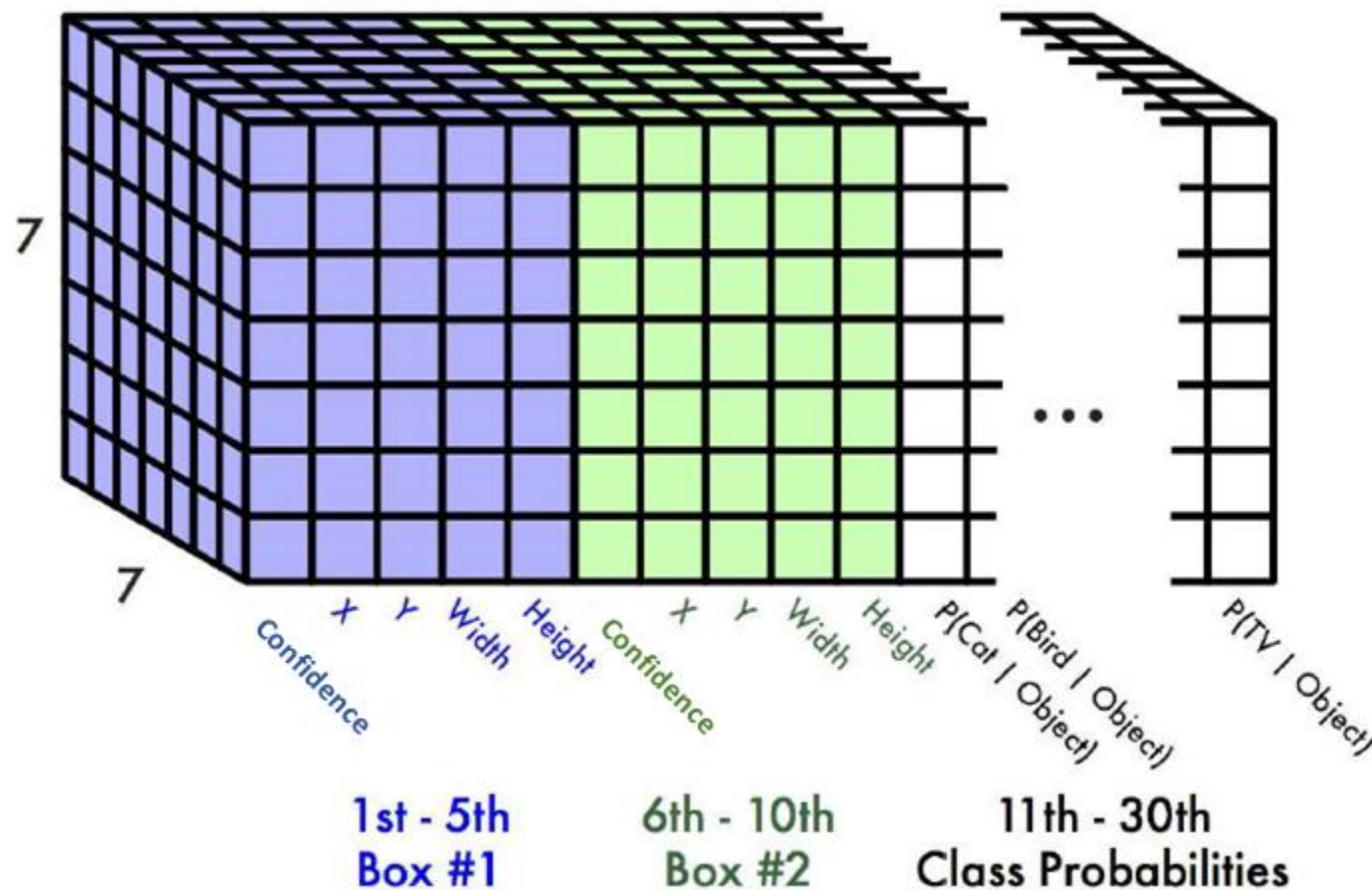
5: 每個Boundarybox (x, y, w, h, confidence)

20: 屬於20個類別的機率。



# YOLOv1的detector怎麼處理

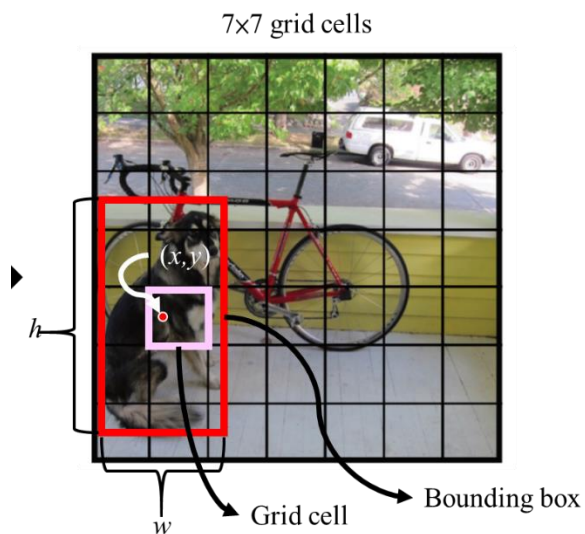
- 最後一層是 $7*7*30$



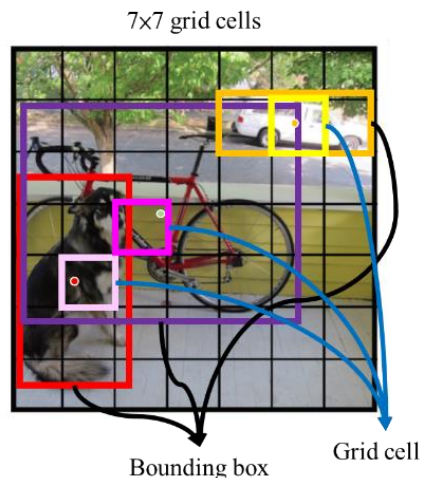
# YOLOv1的detector怎麼處理

5: 每個Boundary box ( $x, y, w, h$ , confidence)

20: 屬於20個類別的機率。



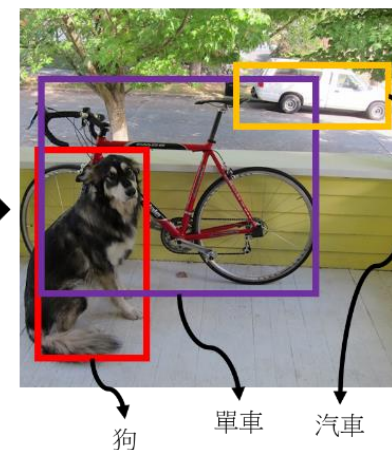
這個物件落在粉紅色框的grid cell  
座標為這個grid cell內紅色框這個  
Bounding box的中心 $(x, y)$ ，高寬為 $h, w$ 。



先利用閾值和NMS選出確定是物件的Bounding box



看選出物件所屬的grid cell屬於哪一類的機率最大。  
水藍色: 「狗」的機率最大  
黃色: 「單車」的機率最大  
粉紅: 「汽車」的機率最大  
橘色: 「地板」的機率最大





# YOLOv1 loss function

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

物件的中心座標( $x, y$ )和模型預測BBOX座標( $\hat{x}, \hat{y}$ )的均方差和

物件的長寬( $w, h$ )和模型預測BBOX長寬( $\hat{w}, \hat{h}$ )的均方差和

這個grid cell有沒有物件的信心度

這個物件被判斷成每個類別的機率

$1_i^{obj}$ : 是物件有出現在 grid cell  $i$ 。

$1_{ij}^{obj}$ : 是在第  $i$  個 grid cell 的第  $j$  個 Bounding Box 負責做預測。





# 是不是有物件

$1^{obj}$

$1^{noobj}$

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	1	1	0	0	0
0	0	1	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	0	0

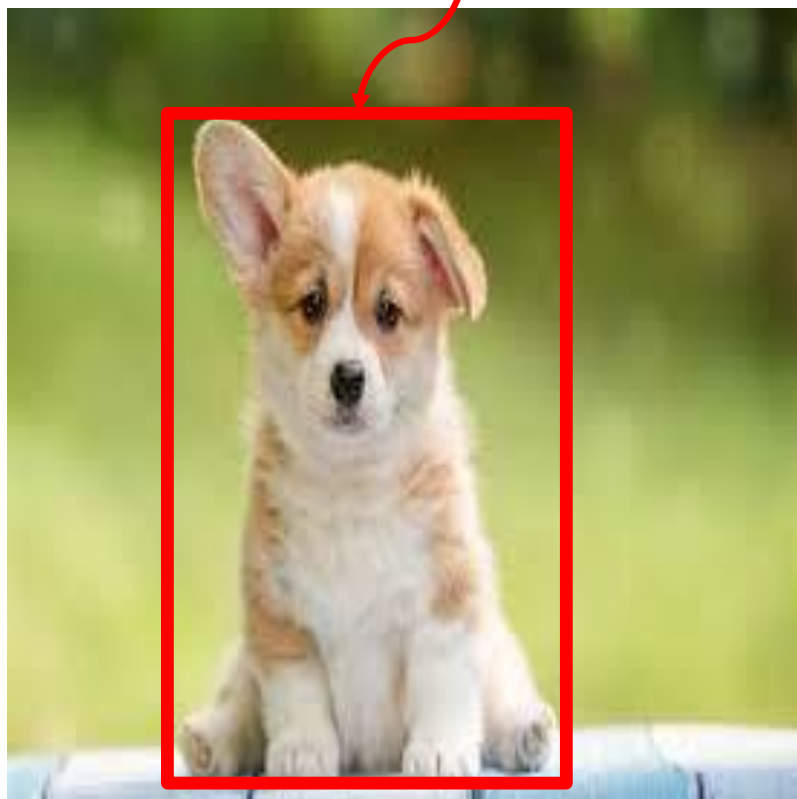
0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	1	1	0	0	0
0	0	1	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	0	0

1	1	1	1	1	1	1
1	0	0	0	1	1	1
1	1	0	0	0	1	1
1	1	0	0	1	1	1
1	1	0	0	1	1	1
1	1	0	0	0	1	1
1	0	0	0	0	1	1



# 是不是有物件

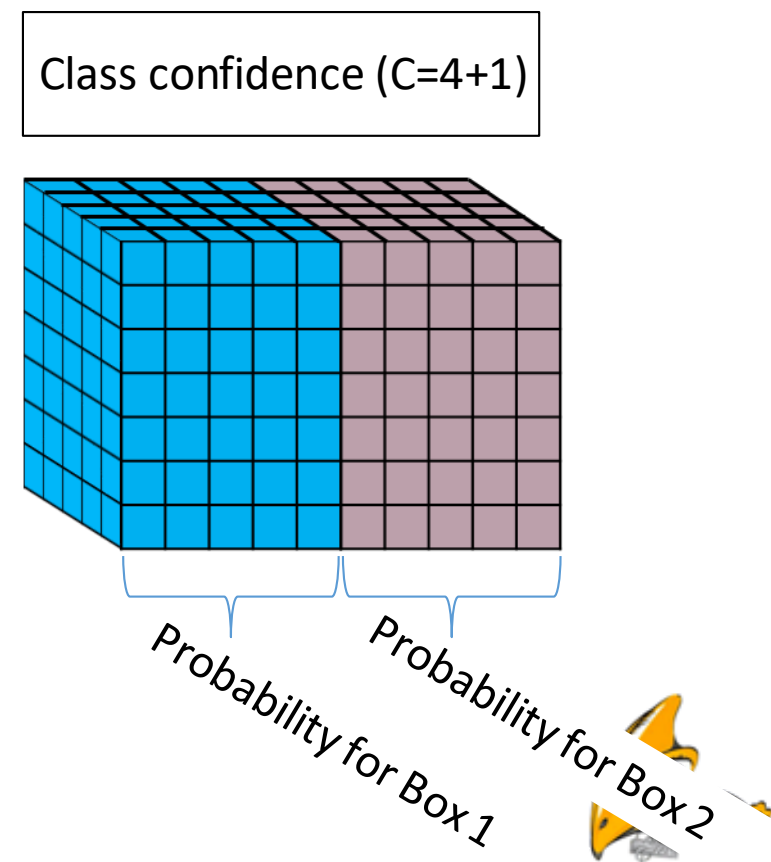
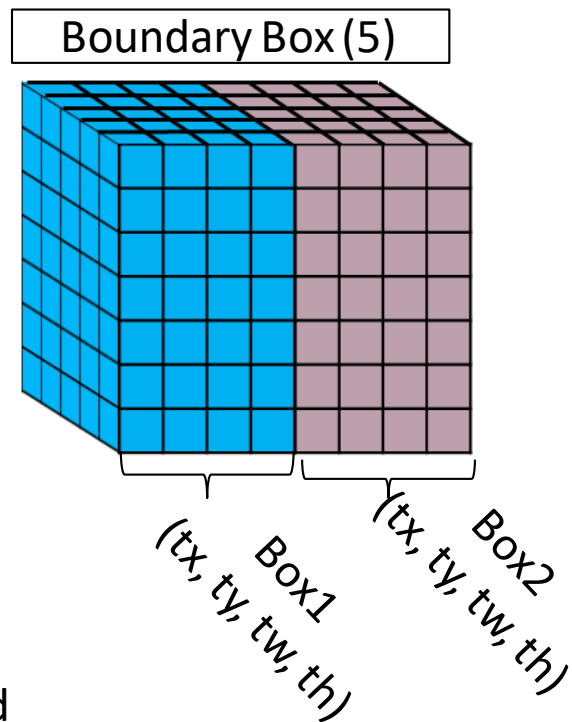
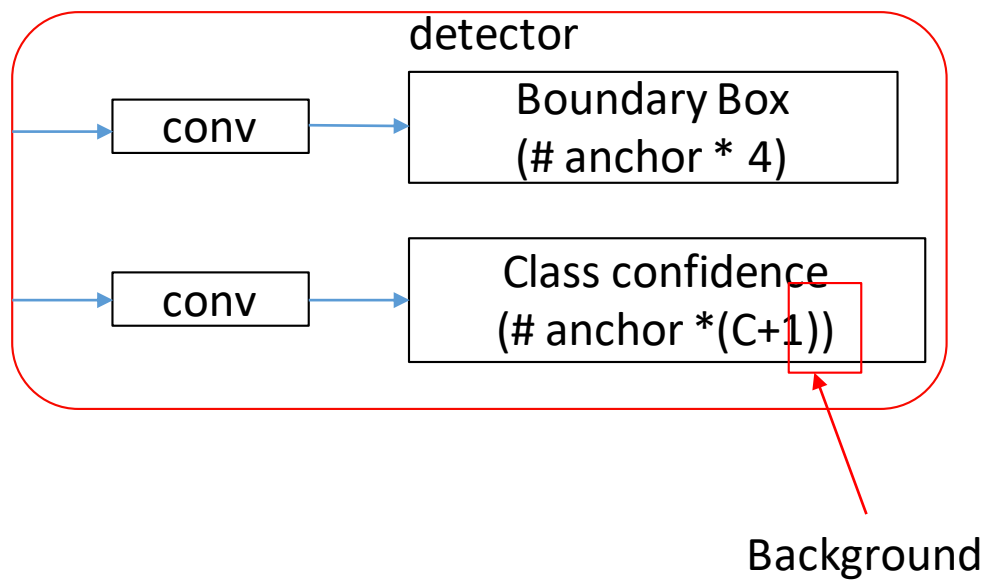
物件框



0	0	0	0	0	0	0
0	1	1	1	1	0	0
0	1	1	1	1	0	0
0	1	1	1	1	0	0
0	1	1	1	1	0	0
0	1	1	1	1	0	0
0	1	1	1	1	0	0

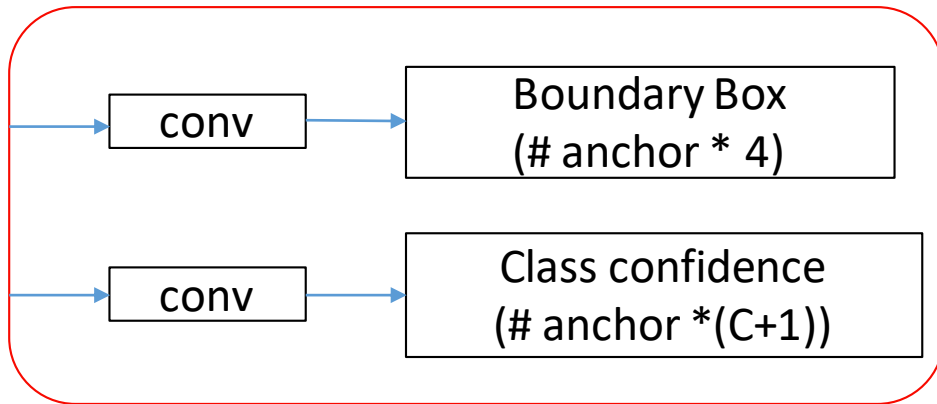


# SSD detector

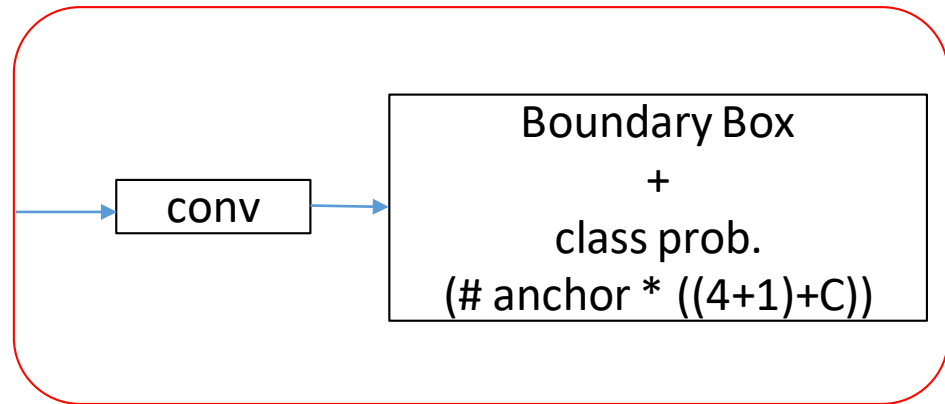


# SSD vs YOLO

SSD detector



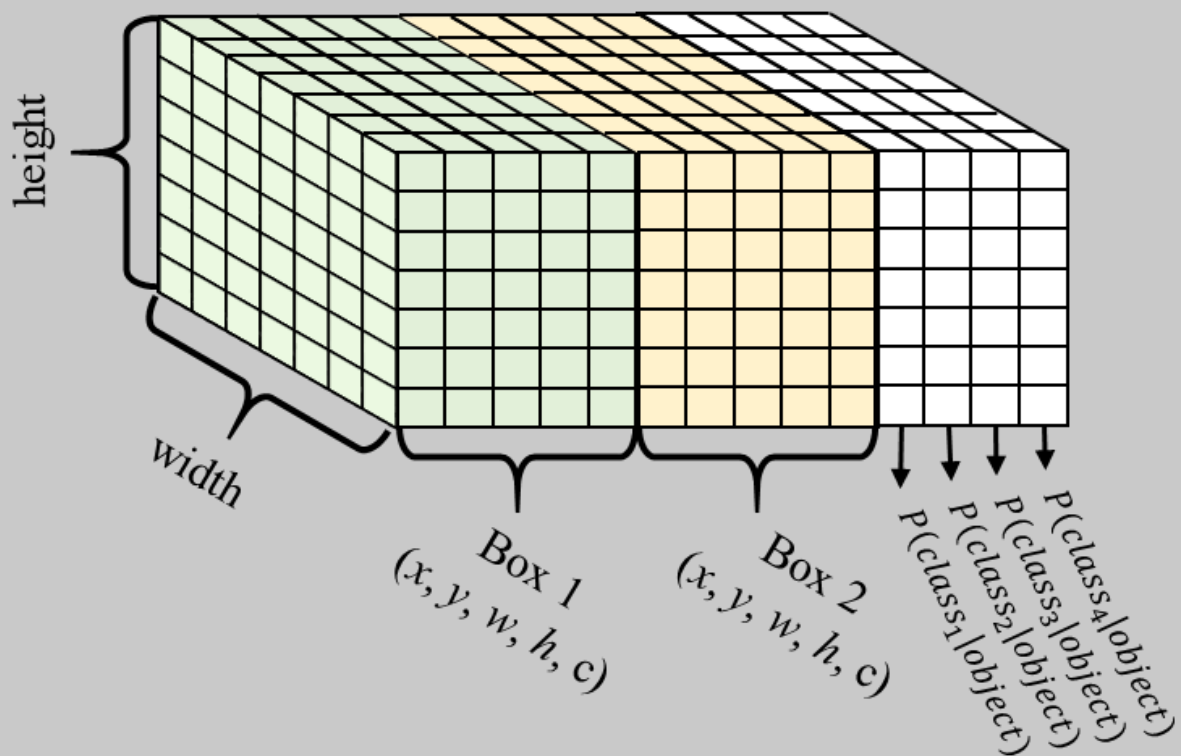
YOLO detector



# SSD vs YOLO

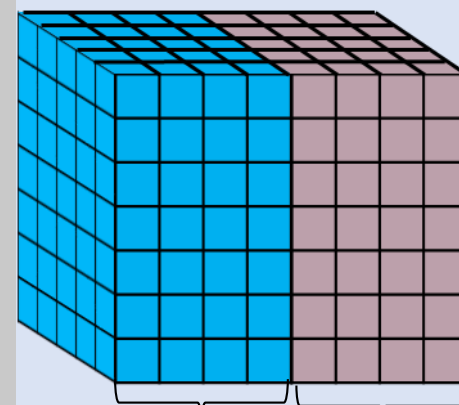
**YOLOv1**  
# class =4

- 最後一層是  $7*7*14$

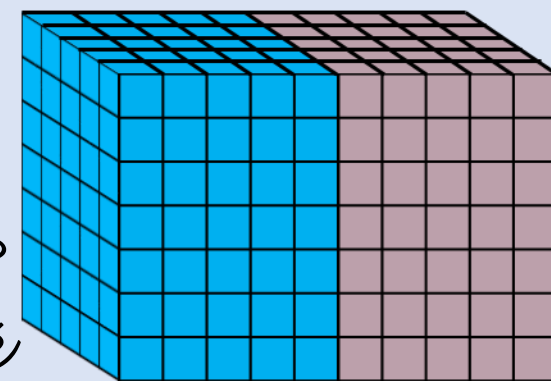


**SSD**  
# class =4

Boundary Box (5)



Class confidence (C=4+1)

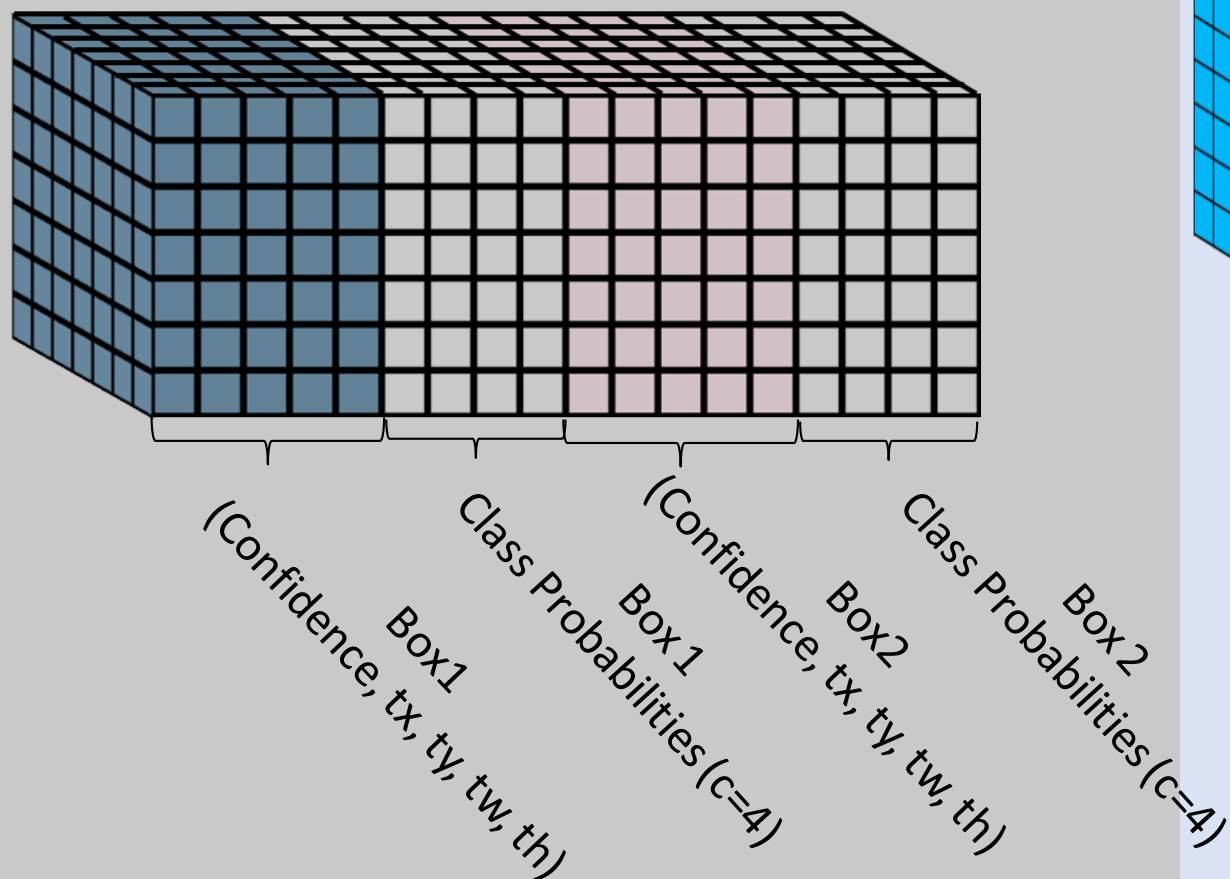


Probability for Box 1  
Probability for Box 2

# SSD vs YOLO

YOLOv2  
# class =4

#Anchor \* (5+4)



SSD  
# class =4

Boundary Box (5)

Class confidence (C=4+1)

