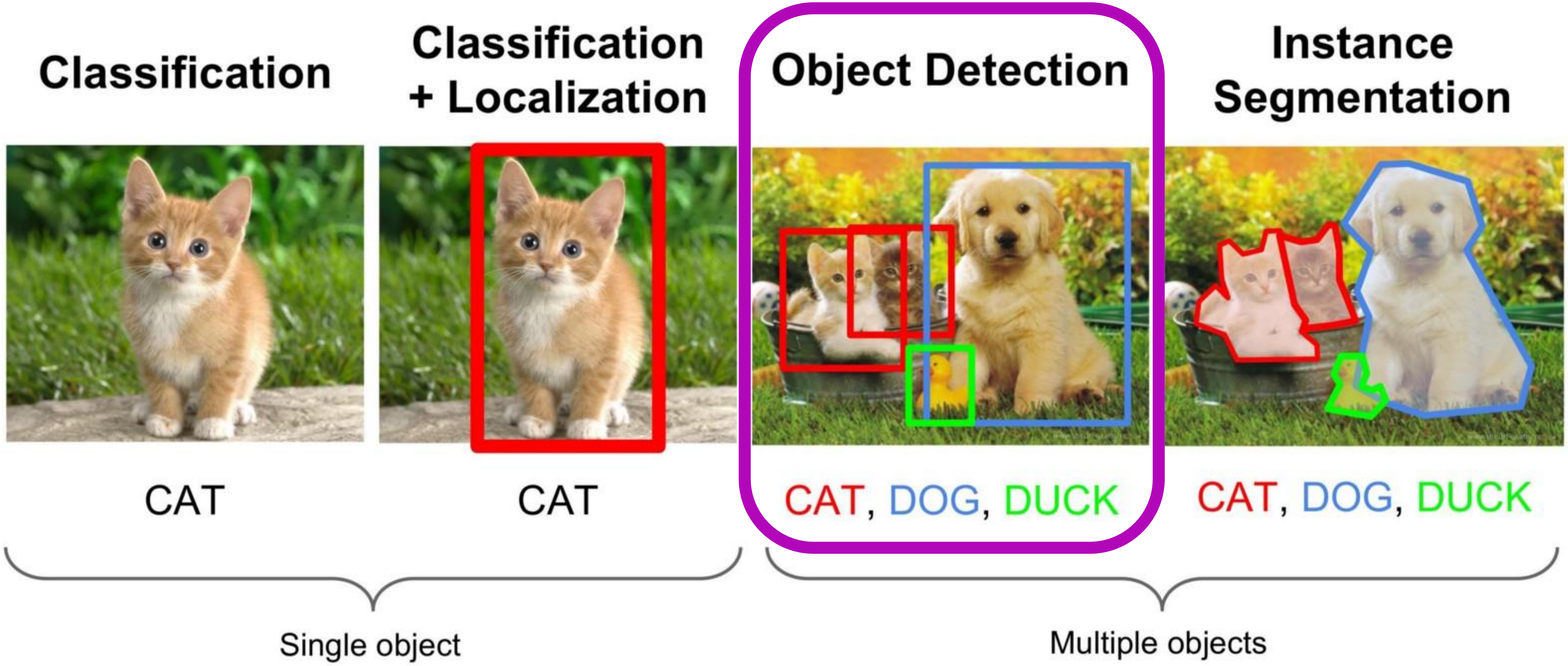
You Only Look Once: Unified, Real-Time Object Detection (YOLO v1)

Joseph Redmon Santosh Divvala Ross Girshick Ali Farhadi

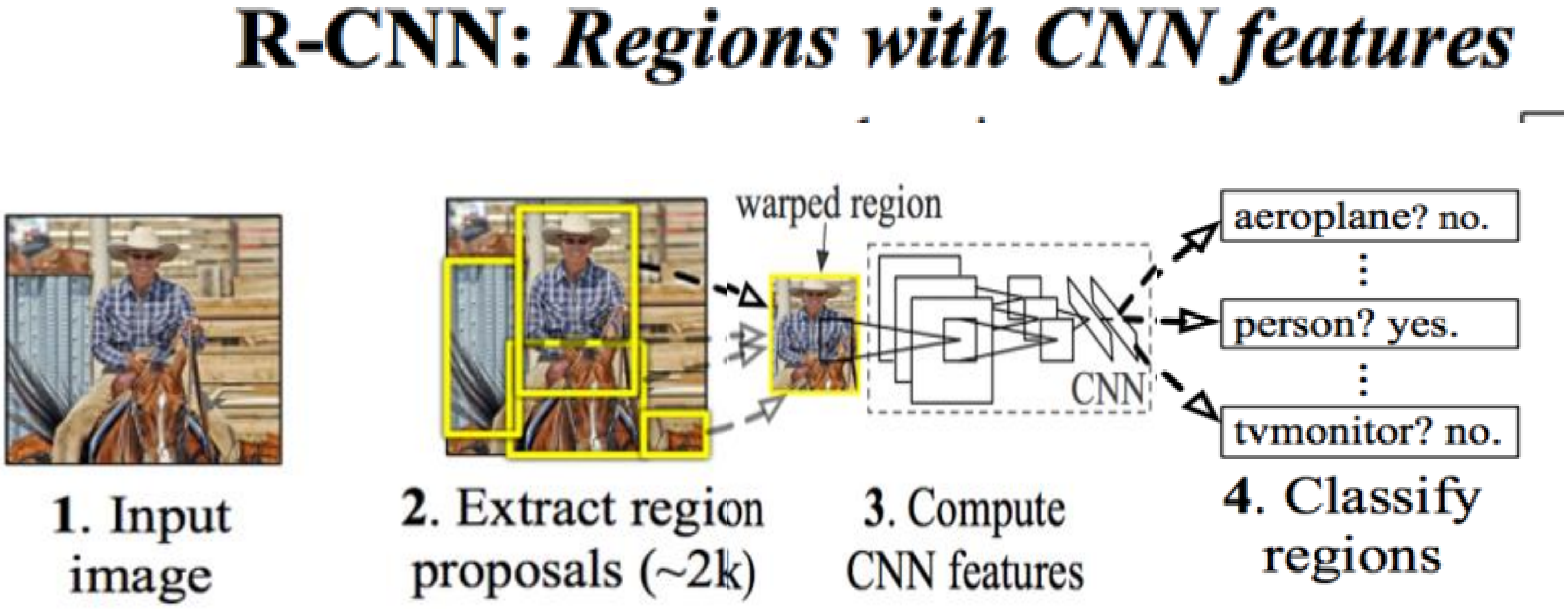
University of Washington, Allen Institute for AI, Facebook AI Research

Sunwoo Kim
swkim@dongguk.edu

Computer Vision Task

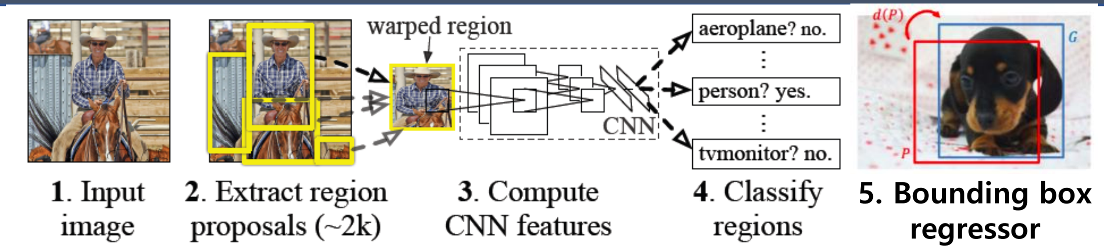


Structure of R-CNN

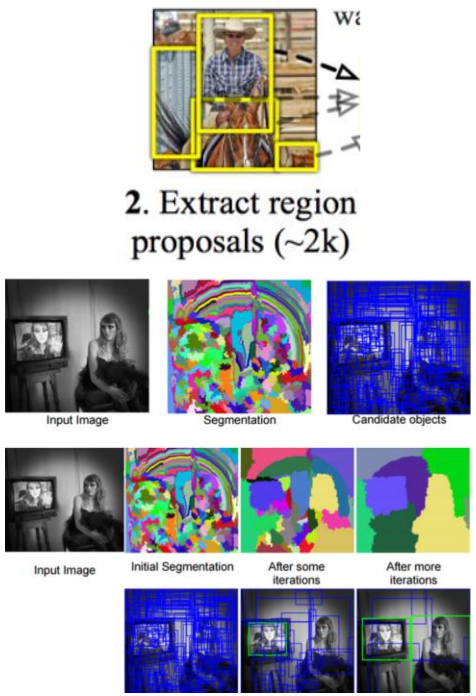


< 1step>   < 2step >   < 3step >

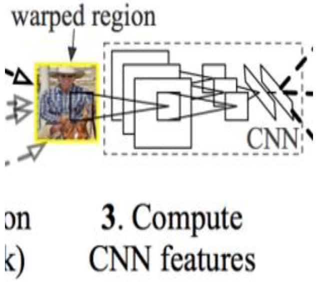
Structure of R-CNN

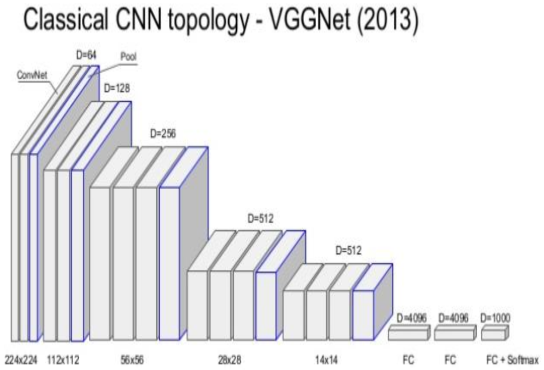


<Selective Search>

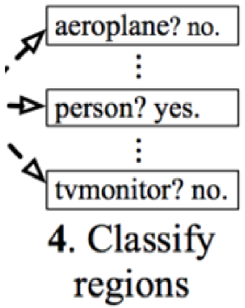


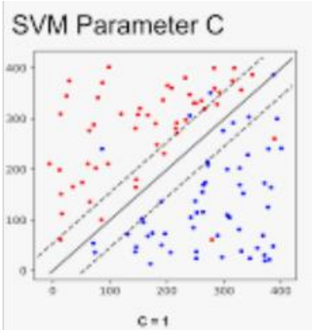
< CNN >





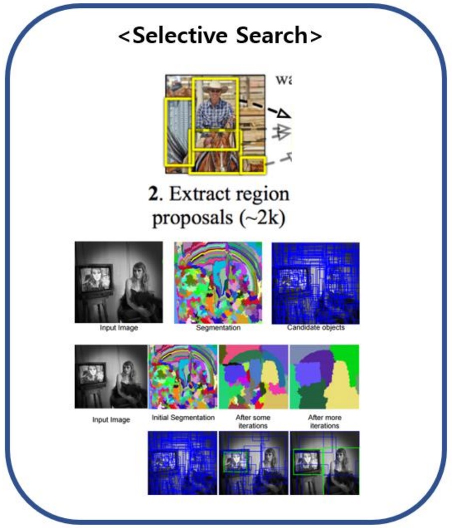
< SVM & regression >

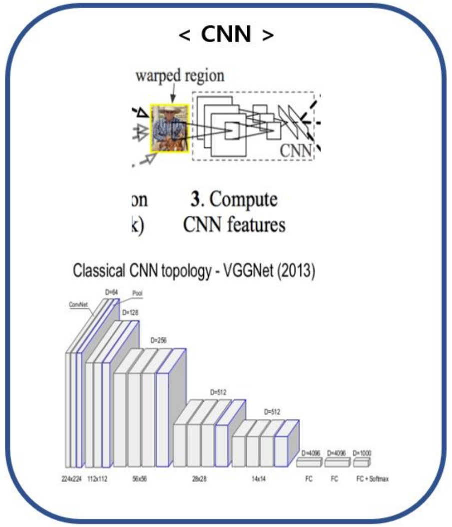


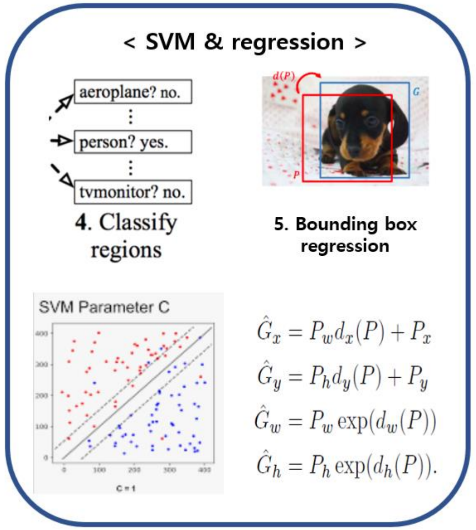


$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp(d_w(P))$$
$$\hat{G}_h = P_h \exp(d_h(P)).$$

Problem of R-CNN series







These complex pipelines are **slow** and **hard** to optimize
Because each individual component must be trained separately

Problem Solving

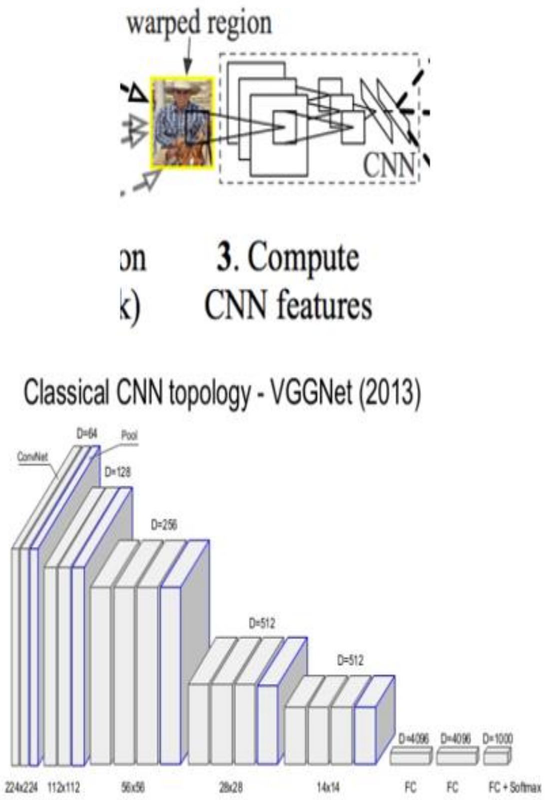
How do we make detection algorithm
fast and **simple** ?

Think as a **single regression problem**

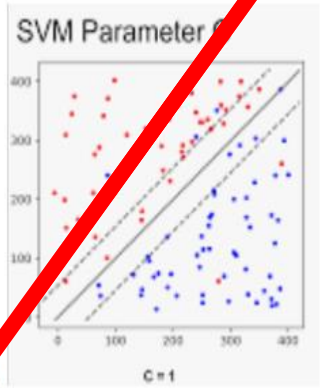
You Only Look Once

<Selective Search>

< CNN >



< SVM & regression >



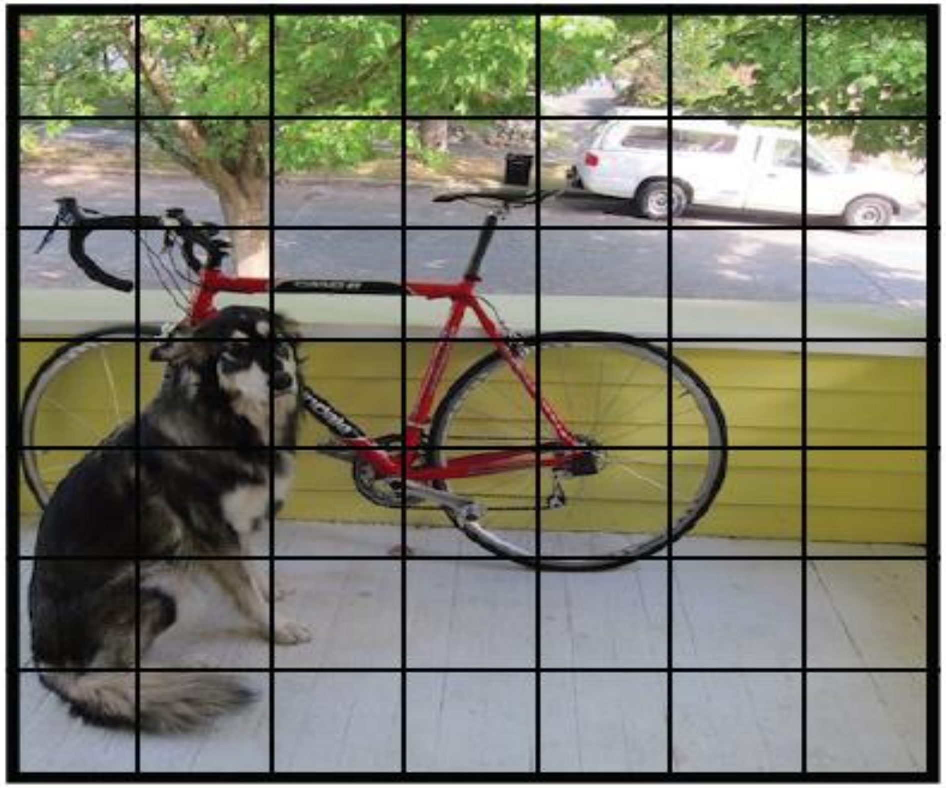
$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp(d_w(P))$$
$$\hat{G}_h = P_h \exp(d_h(P)).$$

Sunwoo Kim

March 22, 2019

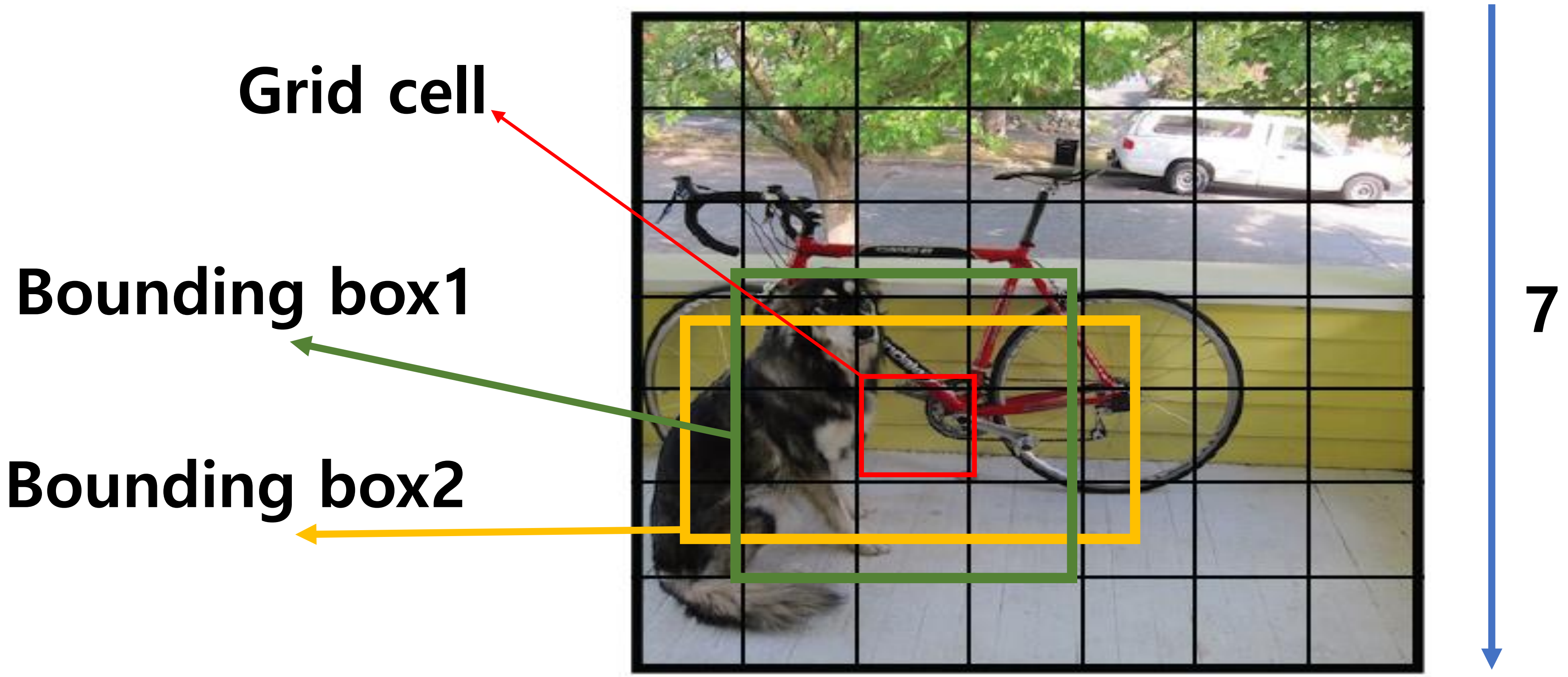
Unified Detection

1. Divide the input image into S X S grid



S × S grid on input

2. Each grid cell has the number of B bounding box

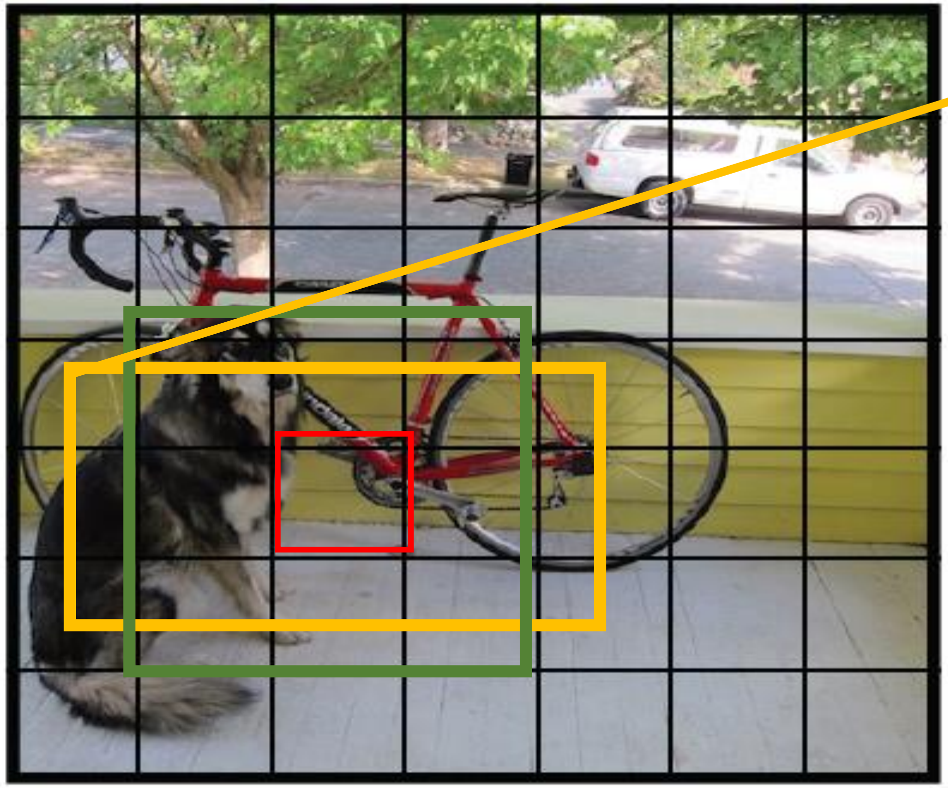


S × S grid on input

S = 7
B = 2

Unified Detection

3. Calculate (x, y, w, h, confidence score) for each bounding box



S × S grid on input

Bounding box2

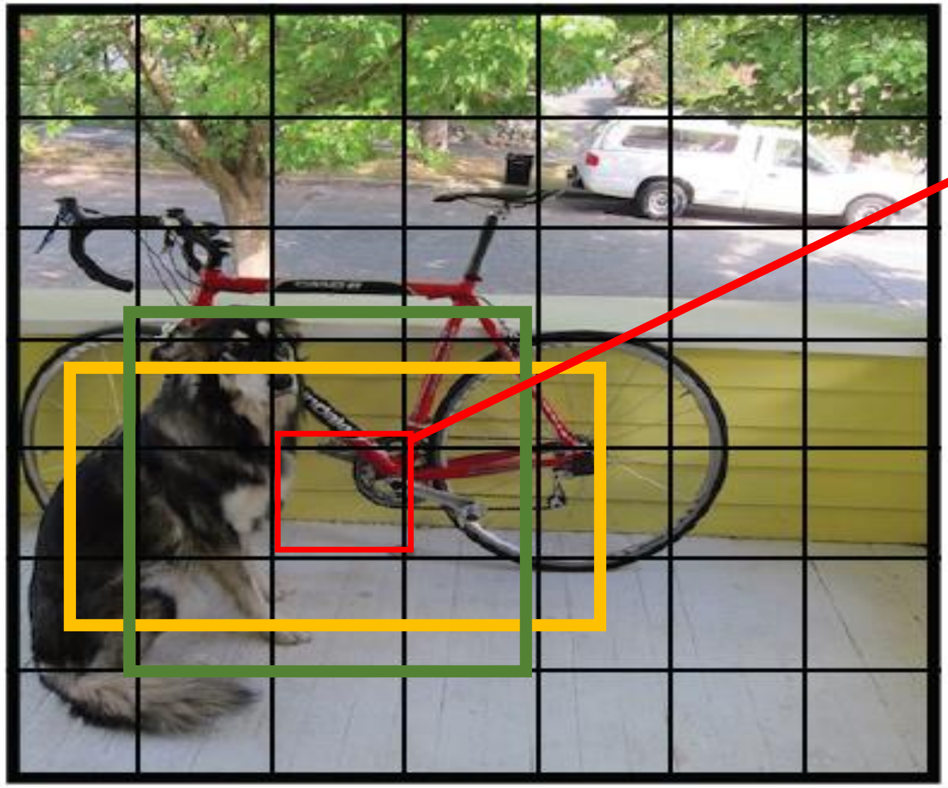
X : x center of bb
Y : y center of bb
W : relative width of the Img
H : relative height of the Img

Confidence score

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

Unified Detection

4. Calculate 'C' conditional class probability for each grid cell



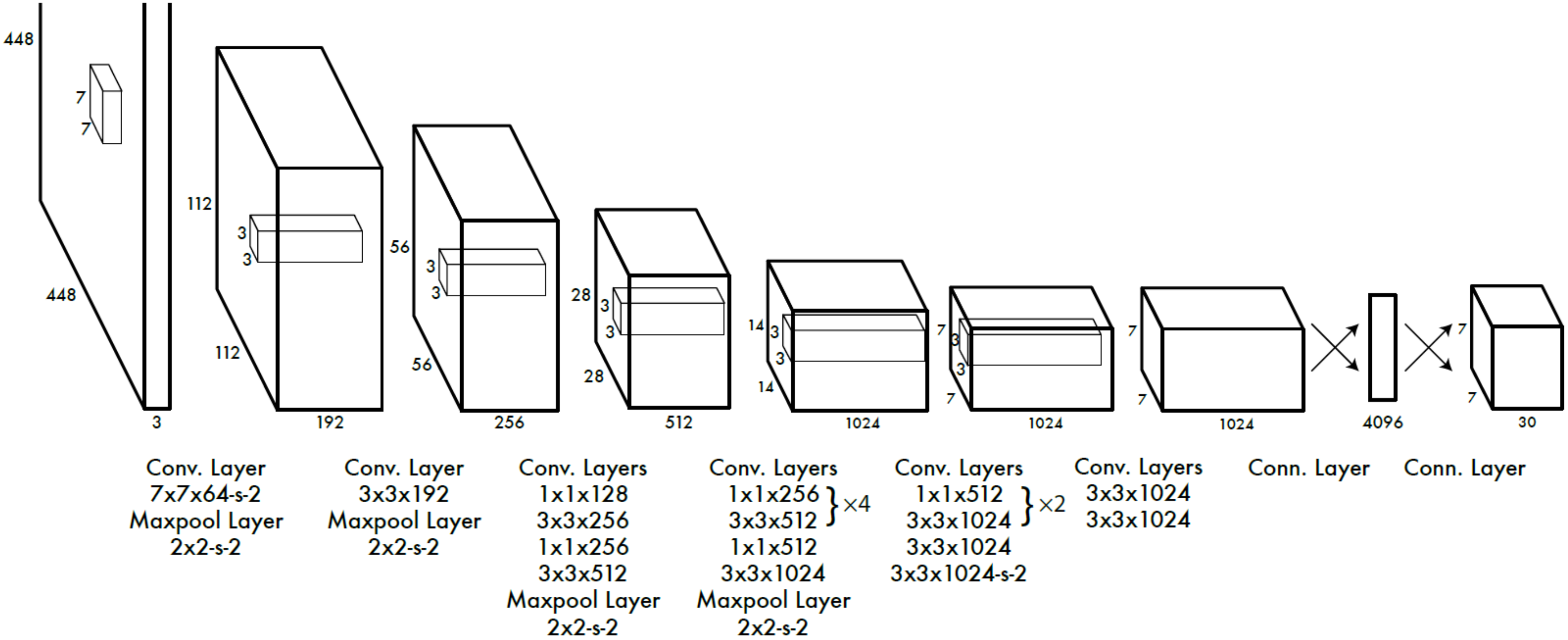
$S \times S$ grid on input

Grid cell

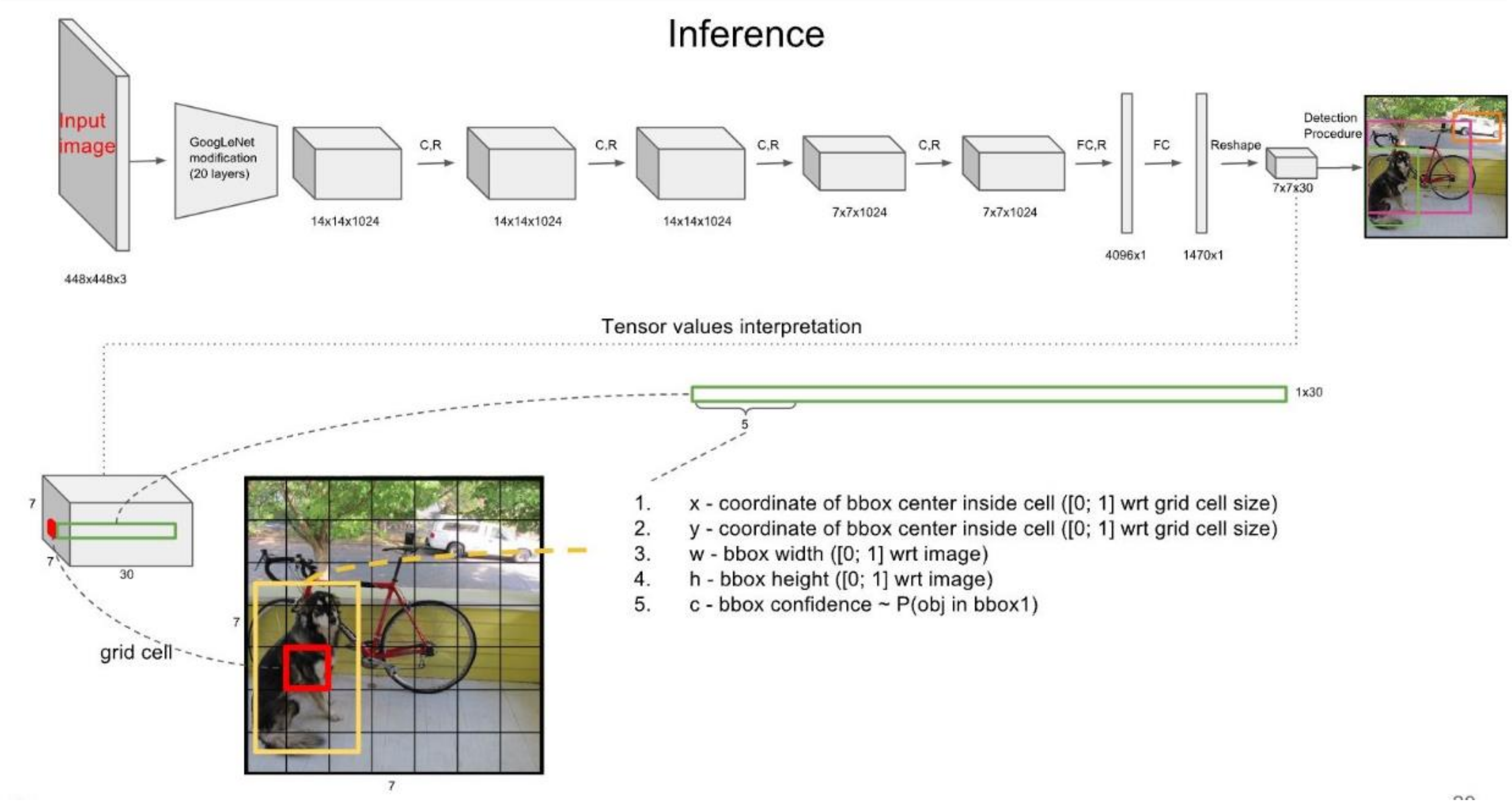
Conditional Class Probability

$$Pr(Class_i|Object)$$

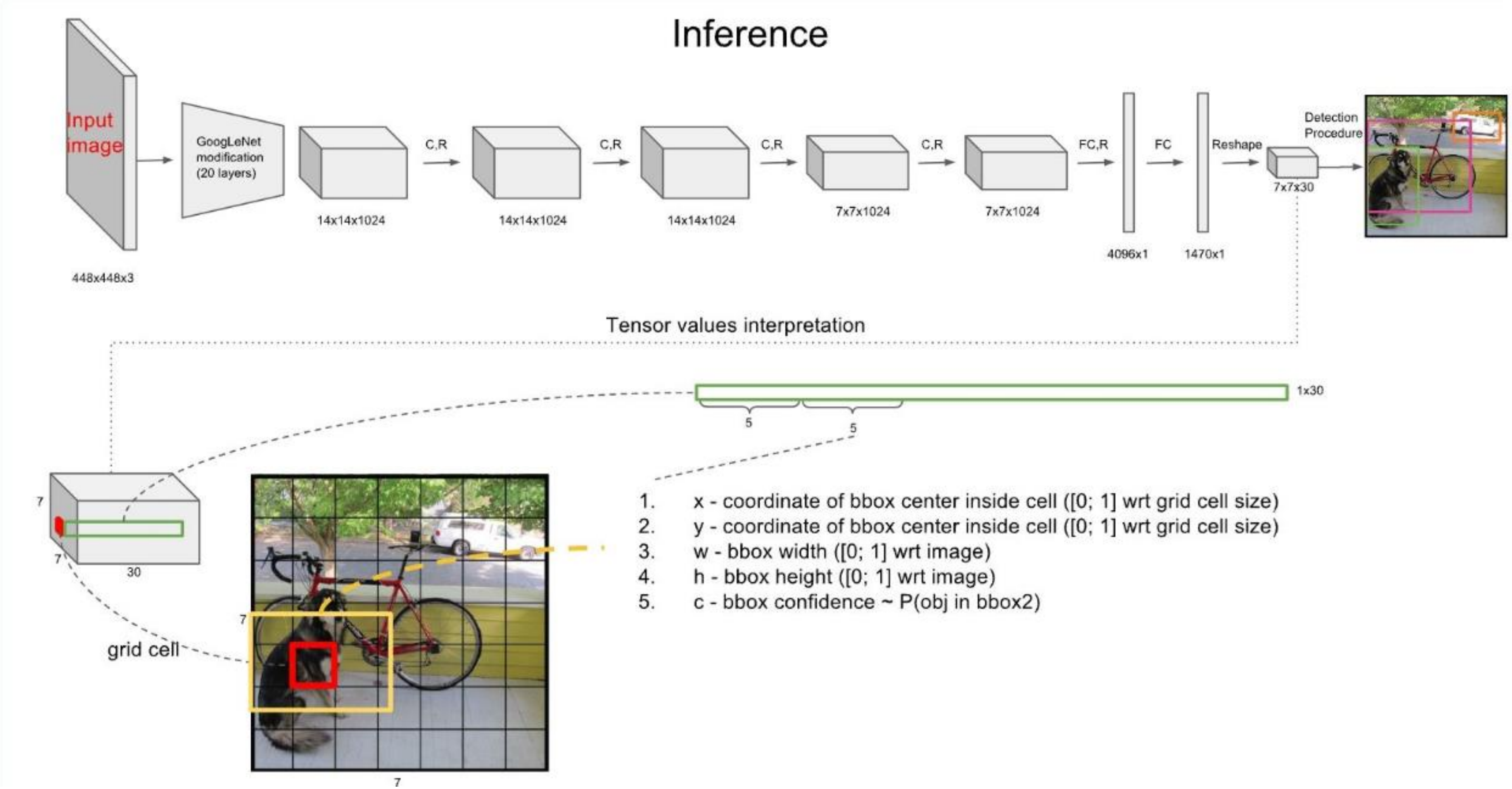
Network Design



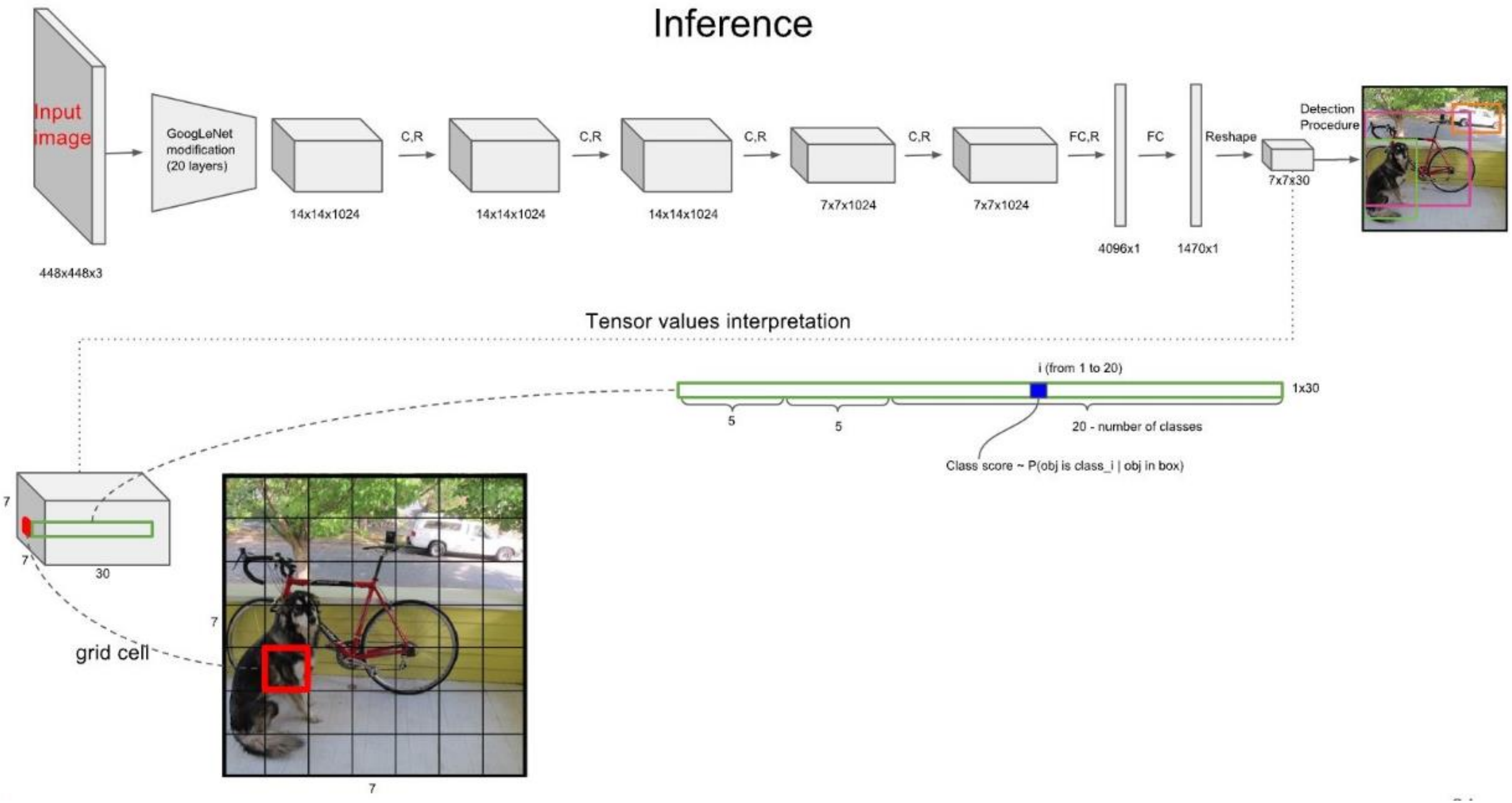
Network Design



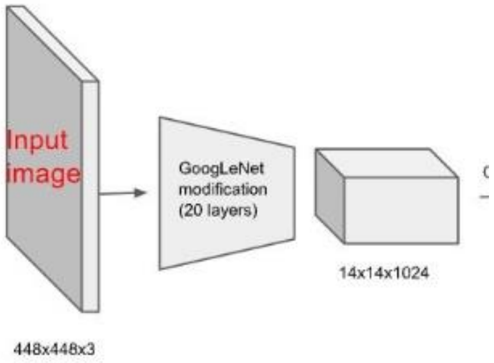
Network Design



Network Design

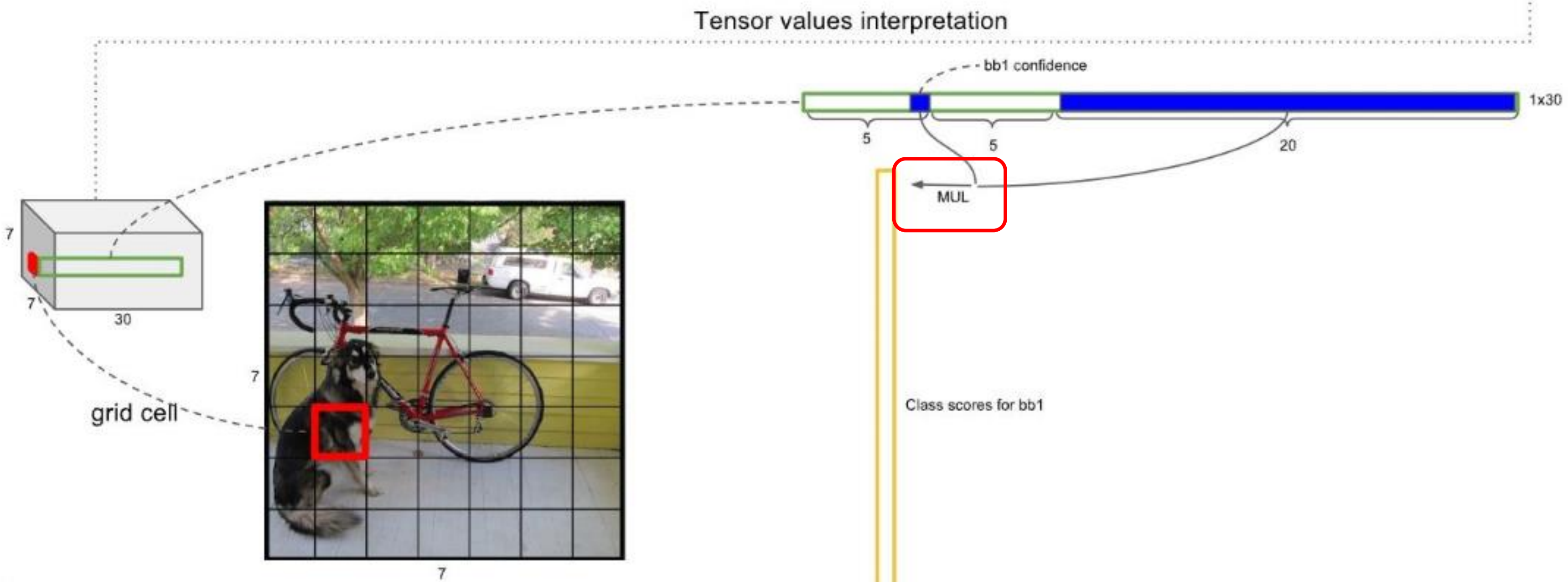


Network Design

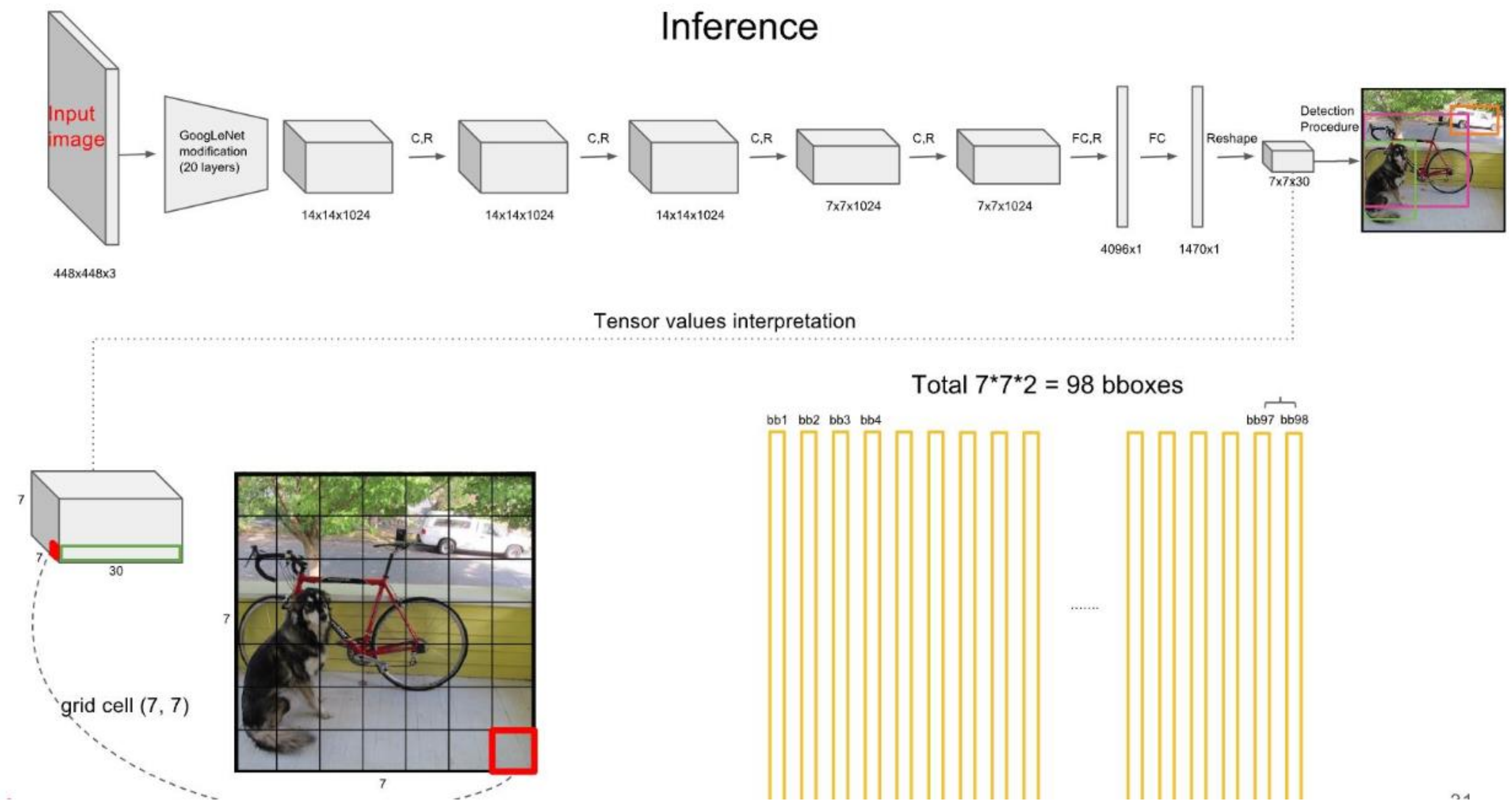


Inference

$$
\begin{aligned}
ClassSpecificConfidenceScore &= ConditionalClassProbability * ConfidenceScore \\
&= Pr(Classi|Object) * Pr(Object) * IOU^{truth}_{pred} \\
&= Pr(Classi) * IOU^{truth}_{pred}
\end{aligned}
$$



Network Design



Non-Maximum Suppression

https://docs.google.com/presentation/d/1aeRvtKG21KHdD5lg6Hgyhx5rPq_ZOsGjG5rJ1HP7BbA/pub?start=false&loop=false&delayms=3000&slide=id.g137784ab86_4_5544

Loss Function

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

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

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \qquad (3)$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \qquad (4)$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \qquad (5)$$

$\lambda_{\textbf{coord}}$ = 5

λ_{noobj} = 0.5

S = grid

B = bb

C = confidence score

c = class

Training

Pre-training : ImageNet 1000-class dataset, 20 conv layer
Add layer : 4conv layer + 2 FC layer
Batch size : 64
Momentum : 0.9
Decay : 0.0005
Dropout rate : 0.5
Activation function : leaky relu

Limitation

1. One grid cell can predict one class
-> It makes to difficult to predict when objects are dense

2. Bounding boxes are learned from data (x,y,w,h,CS)
-> It struggles to objects in new or unusual aspect ratio

3. Coarse features & don't address error in box size
-> It makes localization relatively incorrect

Results

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	**155**
YOLO	2007+2012	**63.4**	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

Table 1: Real-Time Systems on PASCAL VOC 2007. Comparing the performance and speed of fast detectors. Fast YOLO is the fastest detector on record for PASCAL VOC detection and is still twice as accurate as any other real-time detector. YOLO is 10 mAP more accurate than the fast version while still well above real-time in speed.

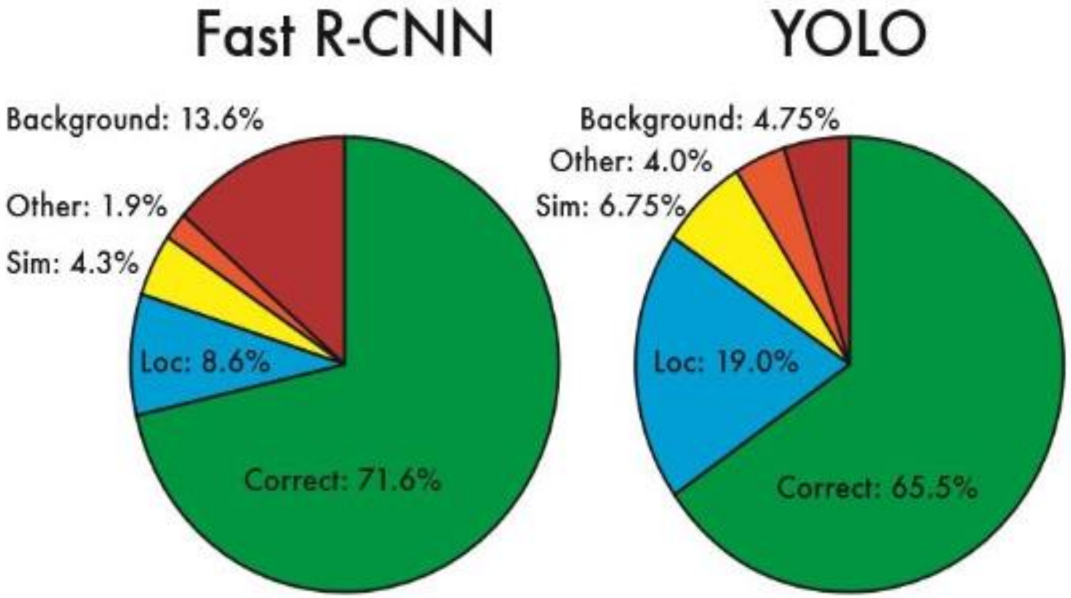


Figure 4: Error Analysis: Fast R-CNN vs. YOLO These charts show the percentage of localization and background errors in the top N detections for various categories (N = # objects in that category).

Thanks