# CS230: Lecture 5

# Advanced topics in Object Detection

Kian Katanforoosh, Andrew Ng
Stanford University

# Today's outline

We will learn:

- the **Intuition** behind object detection methods
- a series of computer vision **papers**

I. Object detection Intuition

II. R-CNN

III. Fast R-CNN

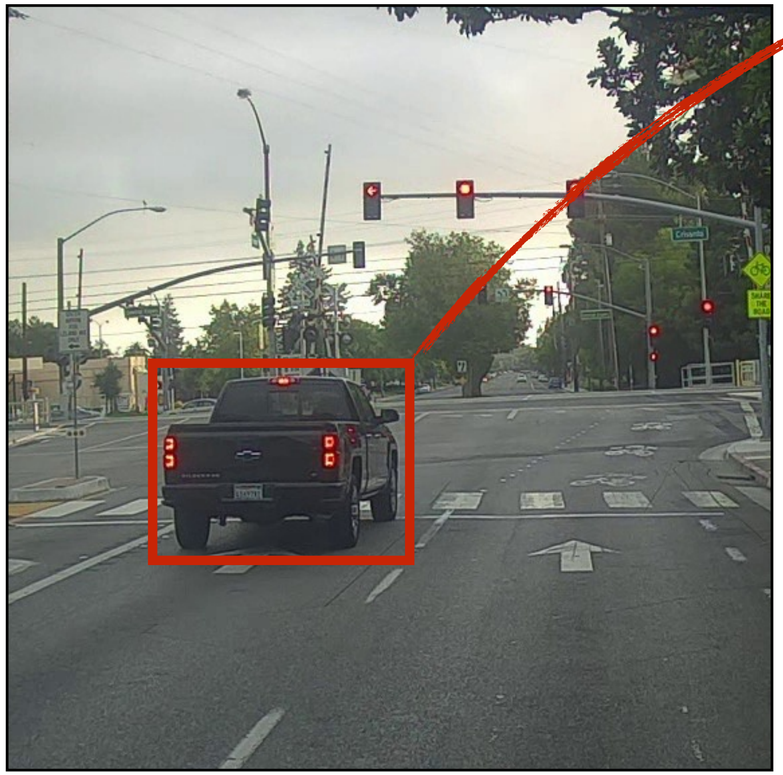
IV. Faster R-CNN

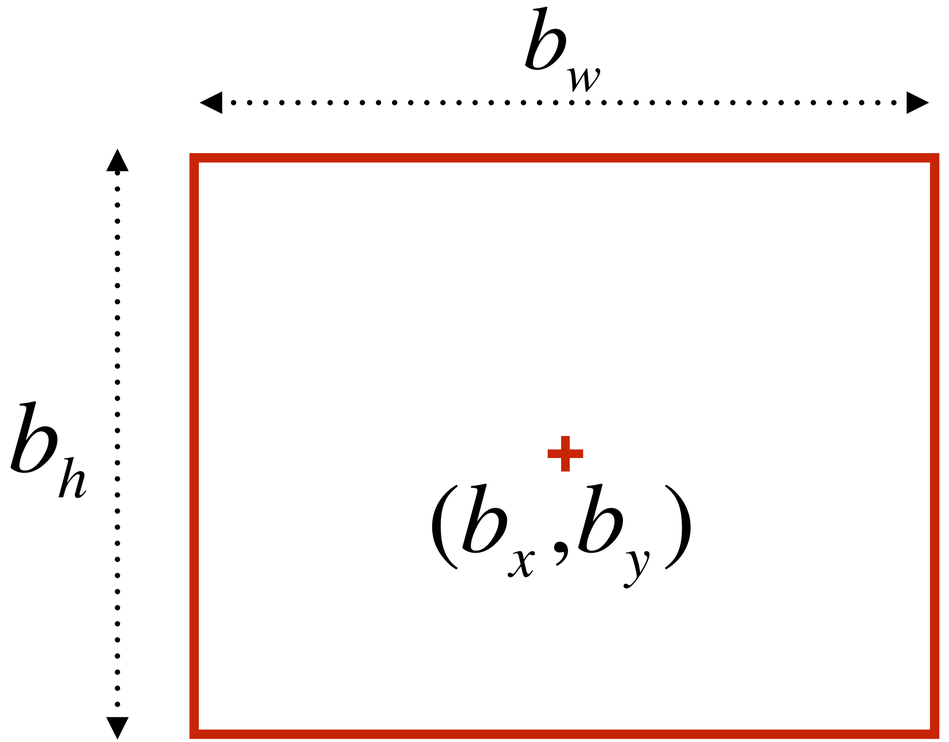
V. YOLO

Let's go

# Object Detection Intuition

$$y = (p_c, b_x, b_y, b_h, b_w, c)$$





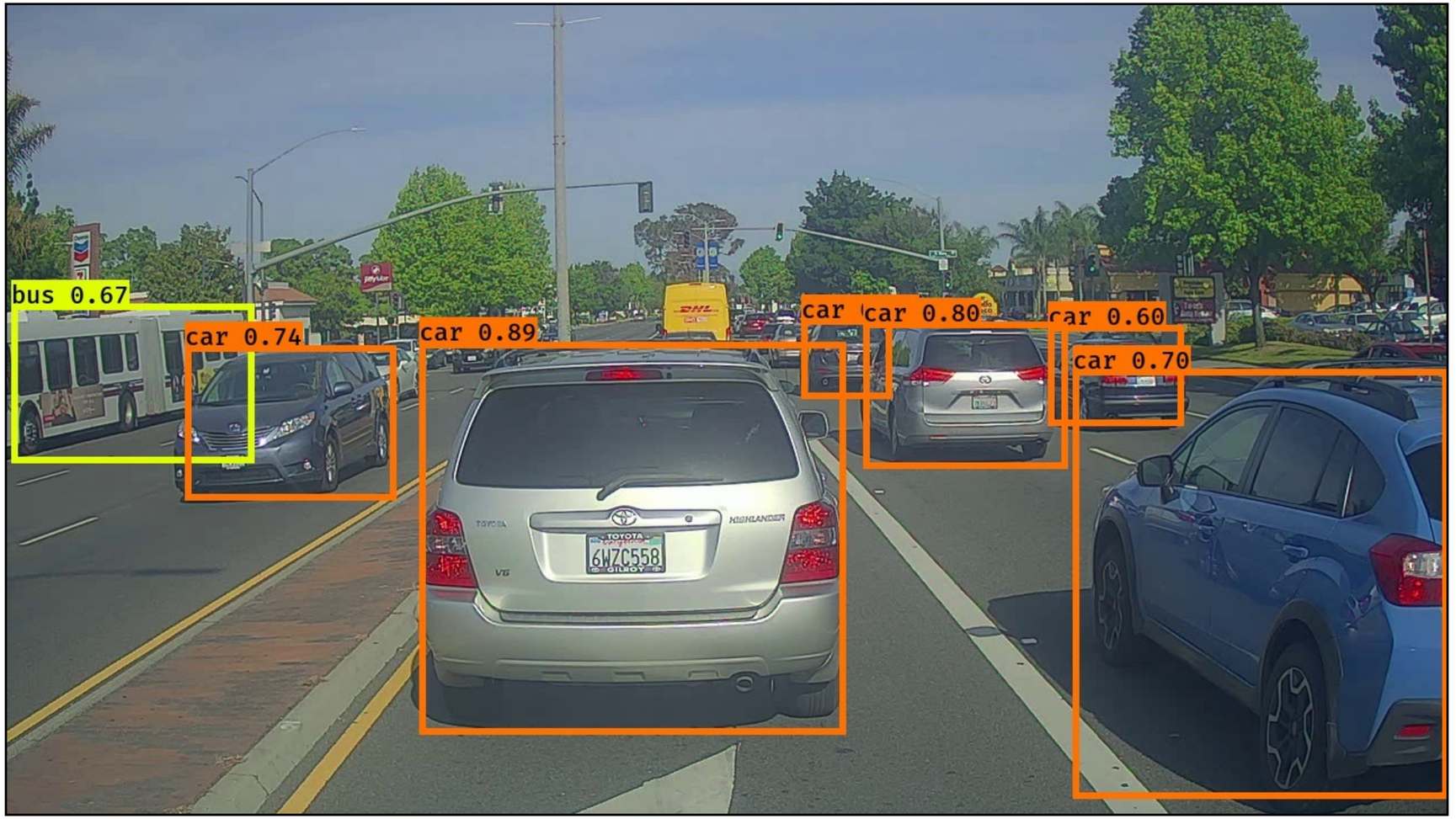

$p_c = 1$ : confidence of an object being present in the bounding box

$c = 0$ : class of the object being detected (here 0 for “car”)

Kian Katanforoosh, Andrew Ng

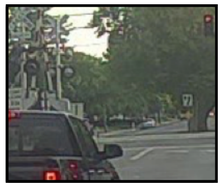
# Object Detection Intuition

$$y = (p_c, b_x, b_y, b_h, b_w, c)$$




Kian Katanforoosh, Andrew Ng

# Object Detection Intuition



Classifier

$$\begin{cases} y \\ p_c \end{cases}$$

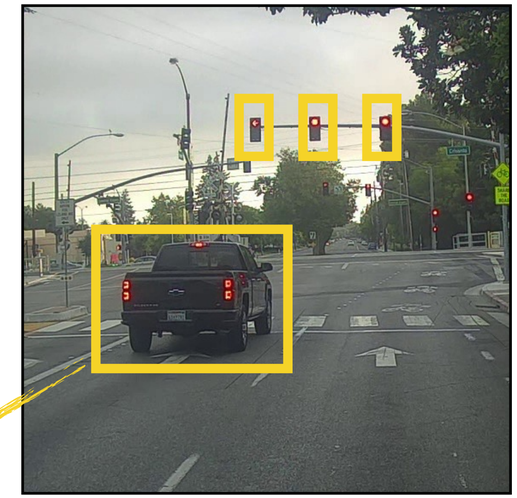
Far too many computations

Kian Katanforoosh, Andrew Ng

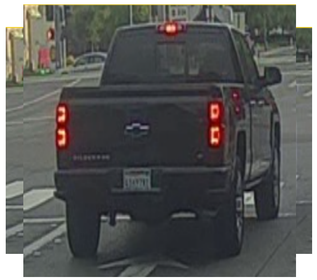
# R-CNN (Regions with CNN features)

input image

region proposal



How do you find these regions?



warped proposal

CNN

**4096-d**

$$\begin{pmatrix} 0.931 \\ 0.433 \\ 0.331 \\ \vdots \\ 0.942 \\ 0.158 \\ 0.039 \end{pmatrix}$$

| | |
|---|---|
| SVM (car) | “yes” |
| SVM (pedestrian) | “no” |
| … | … |
| SVM (traffic light) | “no” |

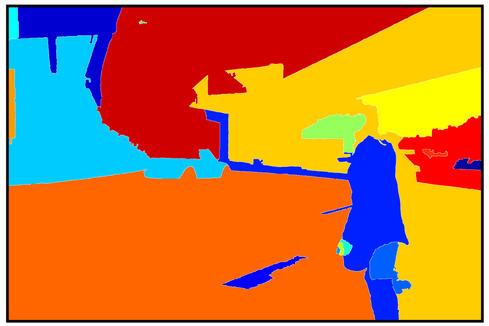
Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik: Rich feature hierarchies for accurate object detection and semantic segmentation (2014)

# R-CNN (Regions with CNN features)

input image

"segmentation"

segmentation labels



"proposals"

region proposals

At test time on a new image, ≈ 2000 regions proposals!

region

CNN

**4096-d**

$$\begin{pmatrix} 0.931 \\ 0.433 \\ 0.331 \\ \vdots \\ 0.942 \\ 0.158 \\ 0.039 \end{pmatrix}$$

| | |
|---|---|
| SVM (car) | "yes" |
| SVM (pedestrian) | "no" |
| … | … |
| SVM (traffic light) | "no" |

**x 2000**

But we only needed two boxes: 1 car + 1 pedestrian …

J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers , and A.W.M. Smeulders: Selective Search for Object Recognition (2012)

Kian Katanforoosh, Andrew Ng

# Non-max suppression

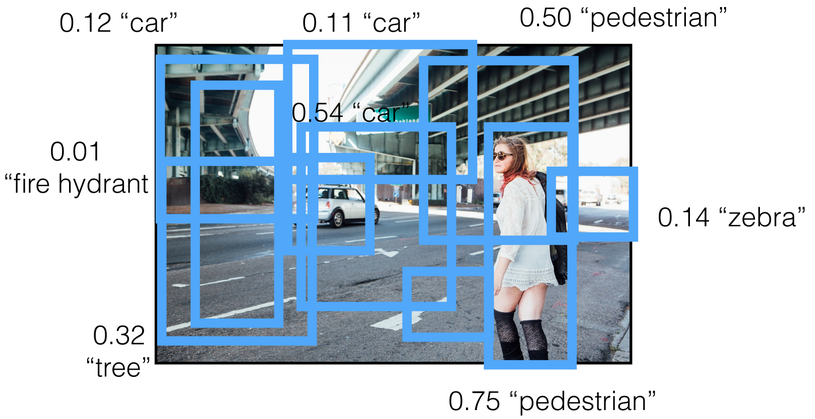
input image

selective search

region proposals

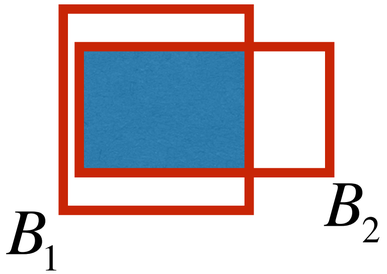
CNN + SVMs

region + scores + classes

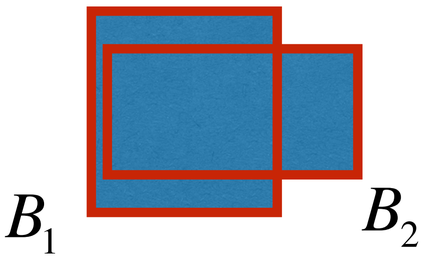



NMS + score threshold


**Intersection**




**Union**



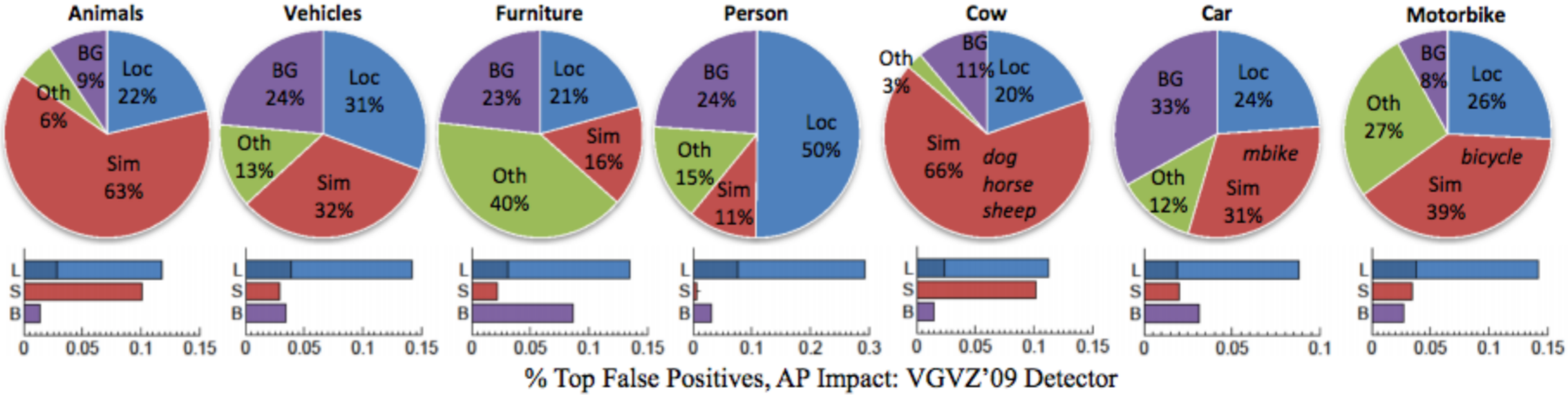

**Intersection over Union**

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{intersection}}{\text{union}}$$

Still slow…
And training is not unified: CNN and each SVM have to be trained separately!

Kian Katanforoosh, Andrew Ng

# Evaluate the performance of your model




% Top False Positives, AP Impact: VGVZ'09 Detector

- BG: background and non-labelled objects
- Loc: localization error (IoU)
- Sim: Confusion with similar objects
- Oth: Confusion with dissimilar objects

Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai: Diagnosing Error in Object Detectors

Kian Katanforoosh, Andrew Ng

# Fast R-CNN

R-CNN drawbacks:

- Training is a multi-stage pipeline
- Training is expensive in space and time
- Testing is slow

Fast R-CNN solutions:

- Share computations for the CNN (not per-region anymore)
- Make training single-stage with multi-task loss
- No disk storage

Kian Katanforoosh, Andrew Ng

# Fast R-CNN

input image + RoIs

CNN

feature map

RoI Pooling layer

FCs

4096-d

$$\begin{pmatrix} 0.931 \\ 0.433 \\ 0.331 \\ \vdots \\ 0.942 \\ 0.158 \\ 0.039 \end{pmatrix}$$

FCs (Softmax)

#classes

$$\begin{pmatrix} 0.31 \\ 0.13 \\ 0.47 \\ \vdots \\ 0.43 \\ 0.81 \\ 0.09 \end{pmatrix}$$

How do you train this?

FCs (Bbox regressor)

$$\begin{pmatrix} 0.24 \\ 0.32 \\ 0.18 \\ 0.14 \end{pmatrix} \cdots \begin{pmatrix} 0.14 \\ 0.54 \\ 0.19 \\ 0.34 \end{pmatrix}$$

#classes

$$L\left(p, u, t^{u}, v\right) = L_{cls}\left(p, u\right) + \lambda \mathrm{I}_{[u \geq 1]} L_{loc}\left(t^{u}, v\right)$$

Ross Girshick: Fast R-CNN (2015)

Kian Katanforoosh, Andrew Ng

Let’s make it even faster

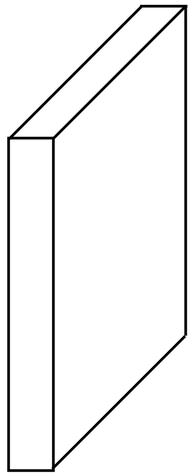
Kian Katanforoosh, Andrew Ng

# Faster R-CNN

input image (no RoI)

CNN

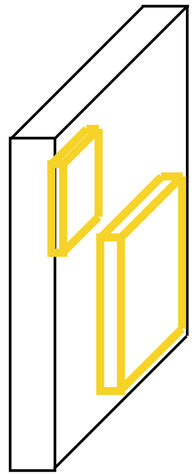
feature map



“attention”

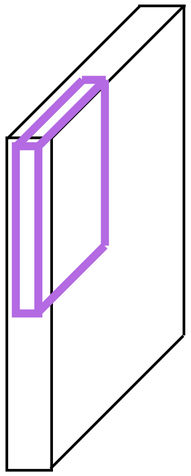
Region Proposal Network

feature map with RoIs

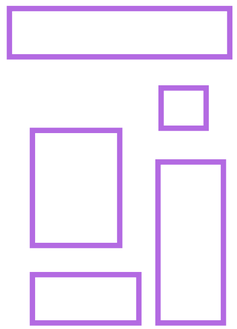


Fast R-CNN

feature map



k anchors per window



FCs (Softmax)

**k scores**

$$\begin{pmatrix} 0.31 \\ 0.13 \\ 0.47 \\ \vdots \\ 0.43 \\ 0.81 \\ 0.09 \end{pmatrix}$$

$$t_x = (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a,$$
$$t_w = \log(w/w_a), \quad t_h = \log(h/h_a),$$
$$t_x^* = (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a,$$
$$t_w^* = \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a),$$

FCs (Bbox regressor)

$$\begin{pmatrix} 0.24 \\ 0.32 \\ 0.18 \\ 0.14 \end{pmatrix} \cdots \begin{pmatrix} 0.14 \\ 0.54 \\ 0.19 \\ 0.34 \end{pmatrix}$$

**k**

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks (2016)

# YOLO

preprocessed image
(608, 608, 3)

Deep CNN

reduction
factor: 32

19

19

encoding
(19,19, 5, 85)

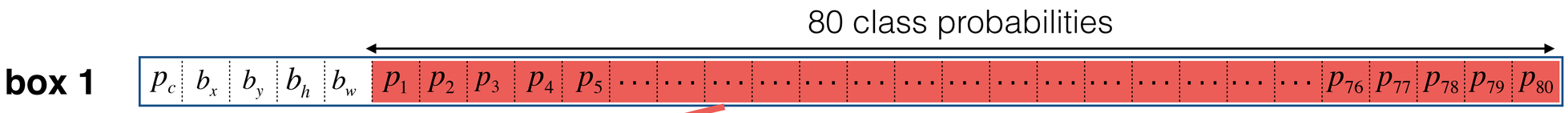
$p_c$ $b_x$ $b_y$ $b_h$ $b_w$ 80 class probabilities

box 1 …
box 2 …
box 3 …
box 4 …
box 5 …

Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi: You Only Look Once: Unified, Real-Time Object Detection (2015)
Joseph Redmon, Ali Farhadi: YOLO9000: Better, Faster, Stronger (2016)

Kian Katanforoosh, Andrew Ng

# YOLO




$$scores = p_c * \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_{78} \\ p_{79} \\ p_{80} \end{pmatrix} = \begin{pmatrix} p_c p_1 \\ p_c p_2 \\ p_c p_3 \\ \vdots \\ p_c p_{78} \\ p_c p_{79} \\ p_c p_{80} \end{pmatrix} = \begin{pmatrix} 0.12 \\ 0.23 \\ 0.44 \\ \vdots \\ 0.07 \\ 0.11 \\ 0.09 \end{pmatrix}$$

find the max

**score**: 0.44
**box**: $(b_x, b_y, b_h, b_w)$
**class**: c = 3 (“car”)

the box $(b_x, b_y, b_h, b_w)$ has detected c = 3 (“car”) with probability score: 0.44

Kian Katanforoosh, Andrew Ng

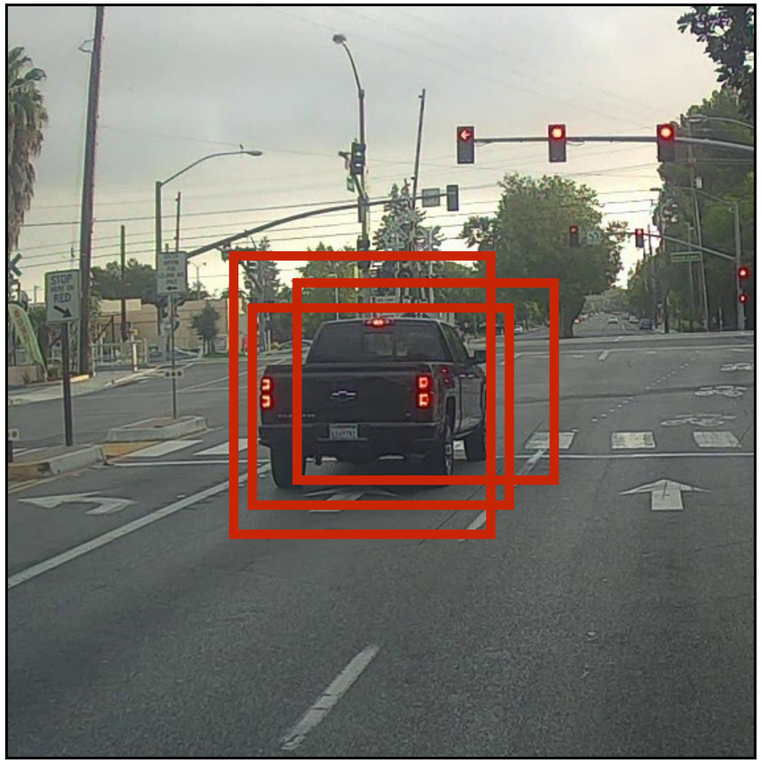
# YOLO

Need to filter:
- Score thresholding
- NMS

Kian Katanforoosh, Andrew Ng

# YOLO

Before non-max suppression



**Non-Max Suppression**

After non-max suppression

Kian Katanforoosh, Andrew Ng

# YOLO

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

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

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

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

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

demo

Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi: You Only Look Once: Unified, Real-Time Object Detection

Kian Katanforoosh, Andrew Ng