deeplearning.ai

Object Detection

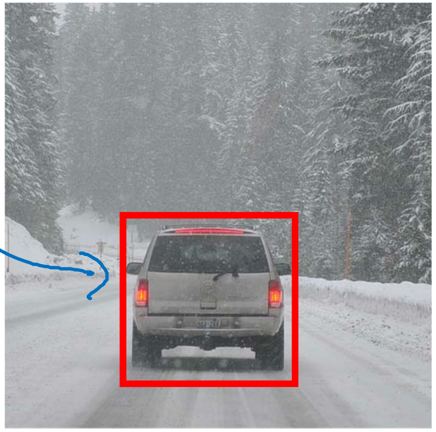
Object localization

What are localization and detection?

Image classification

" Car "

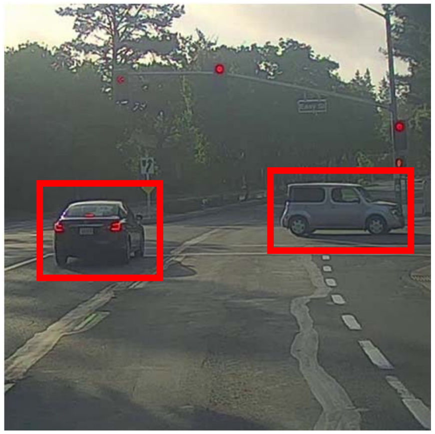
Classification with localization



"Car "

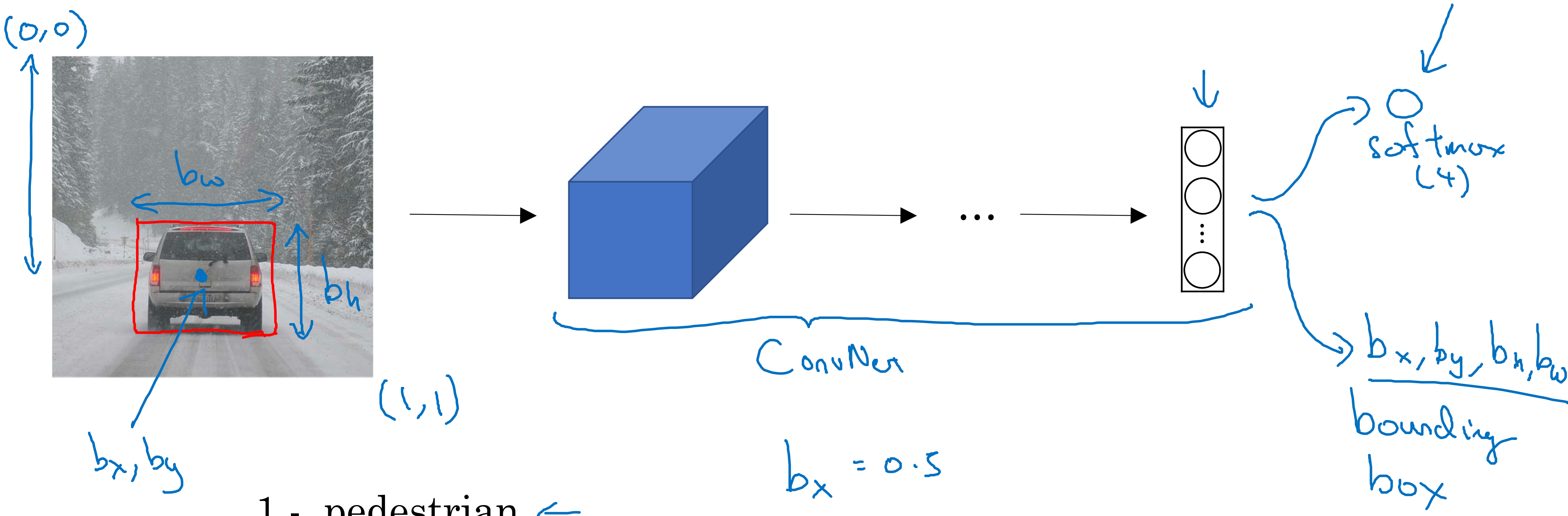
1 object

Detection



multiple objects

Classification with localization



1 - pedestrian ←

2 - car ←

3 - motorcycle ←

4 - background

$b_x = 0.5$

$b_y = 0.7$

$b_h = 0.3$

$b_w = 0.4$

Andrew Ng

Defining the target label y

1 - pedestrian
2 - car ←
3 - motorcycle
4 - background ←

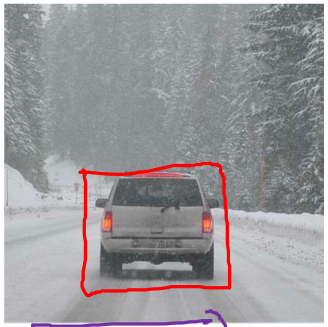
Need to output b_x, b_y, b_h, b_w, class label (1-4)

$$\mathcal{L}(\hat{y}, y) = \begin{cases} (\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \cdots + (\hat{y}_8 - y_8)^2 & \text{if } y_1 = 1 \\ (\hat{y}_1 - y_1)^2 & \text{if } y_1 = 0 \end{cases}$$

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

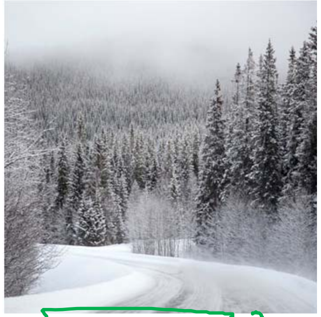
is there any object?

$x =$



$$\begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

(x, y)



$$\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

p_c

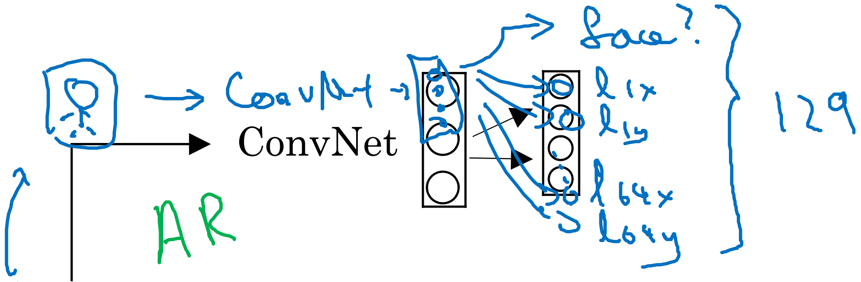
← "don't care"

deeplearning.ai

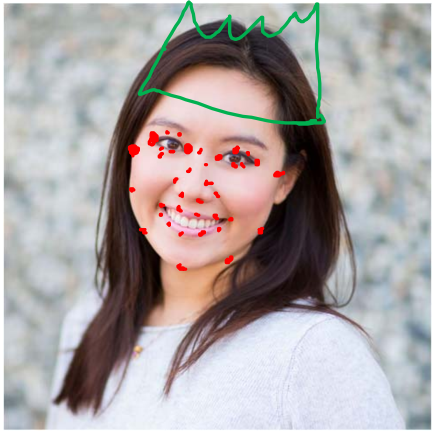
Object Detection

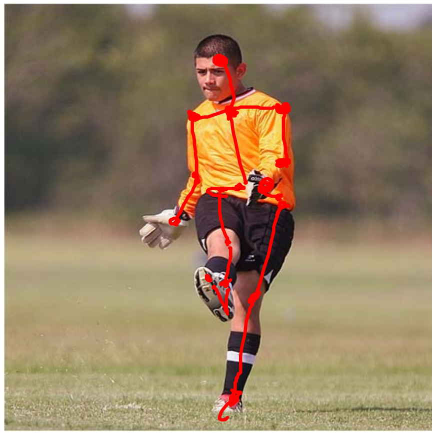
Landmark detection

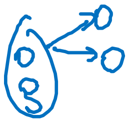
Landmark detection



b_x, b_y, b_h, b_w







$$\left.\begin{array}{l} l_{1x},\ l_{1y}, \\ l_{2x},\ l_{2y}, \\ l_{3x},\ l_{3y}, \\ l_{4x},\ l_{4,y} \\ \vdots \\ l_{64},\ l_{64y} \end{array}\right\} X, y$$

$$\begin{array}{l} l_{1x}, l_{1y}, \\ \vdots \\ l_{32x}\ l_{32y} \end{array}$$

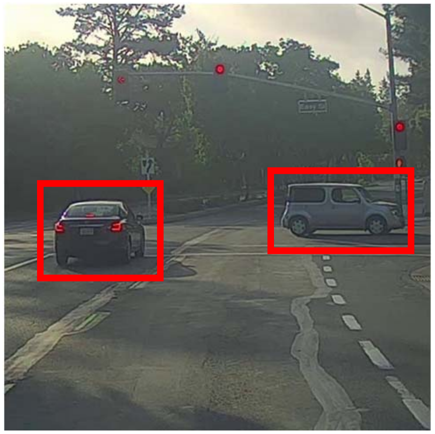
Andrew Ng

deeplearning.ai

Object Detection

Object detection

Car detection example



Training set:

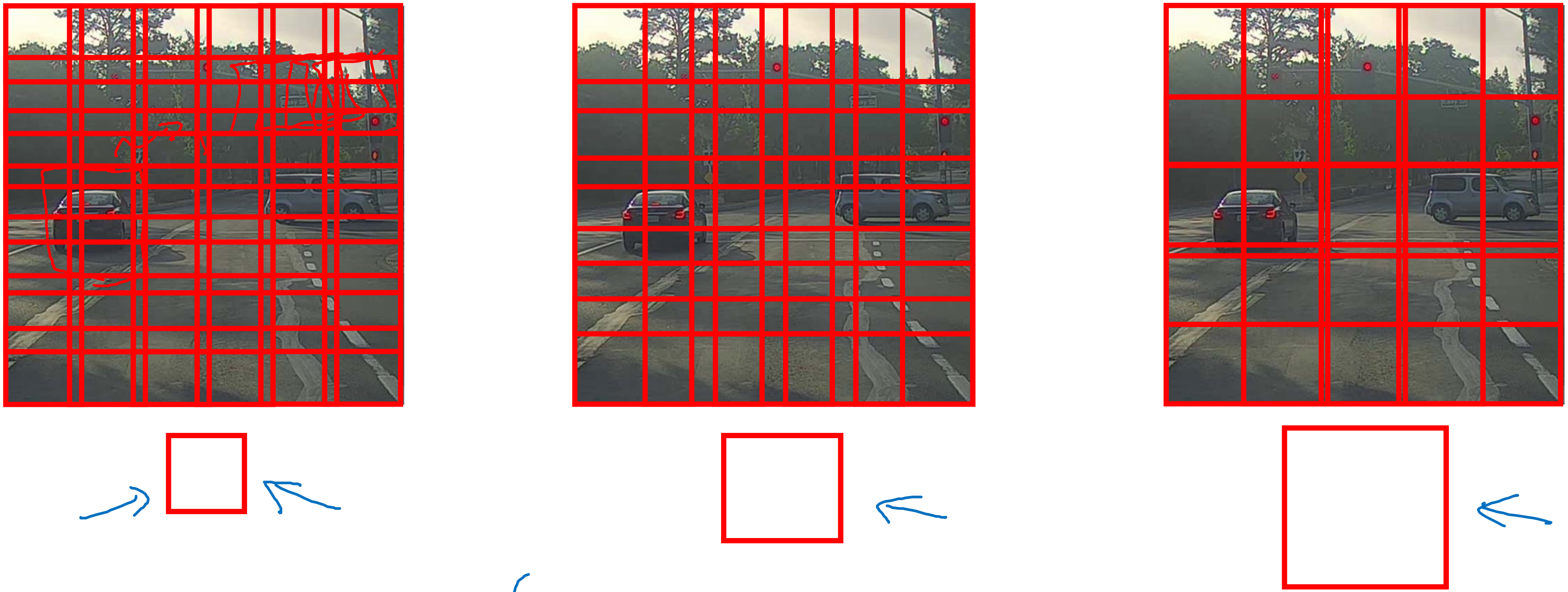
x	y
	1
	1
	1
	0
	0

→ ConvNet → y

Sliding windows detection

→ ConvNet → 0

→ ConvNet



Computation cost

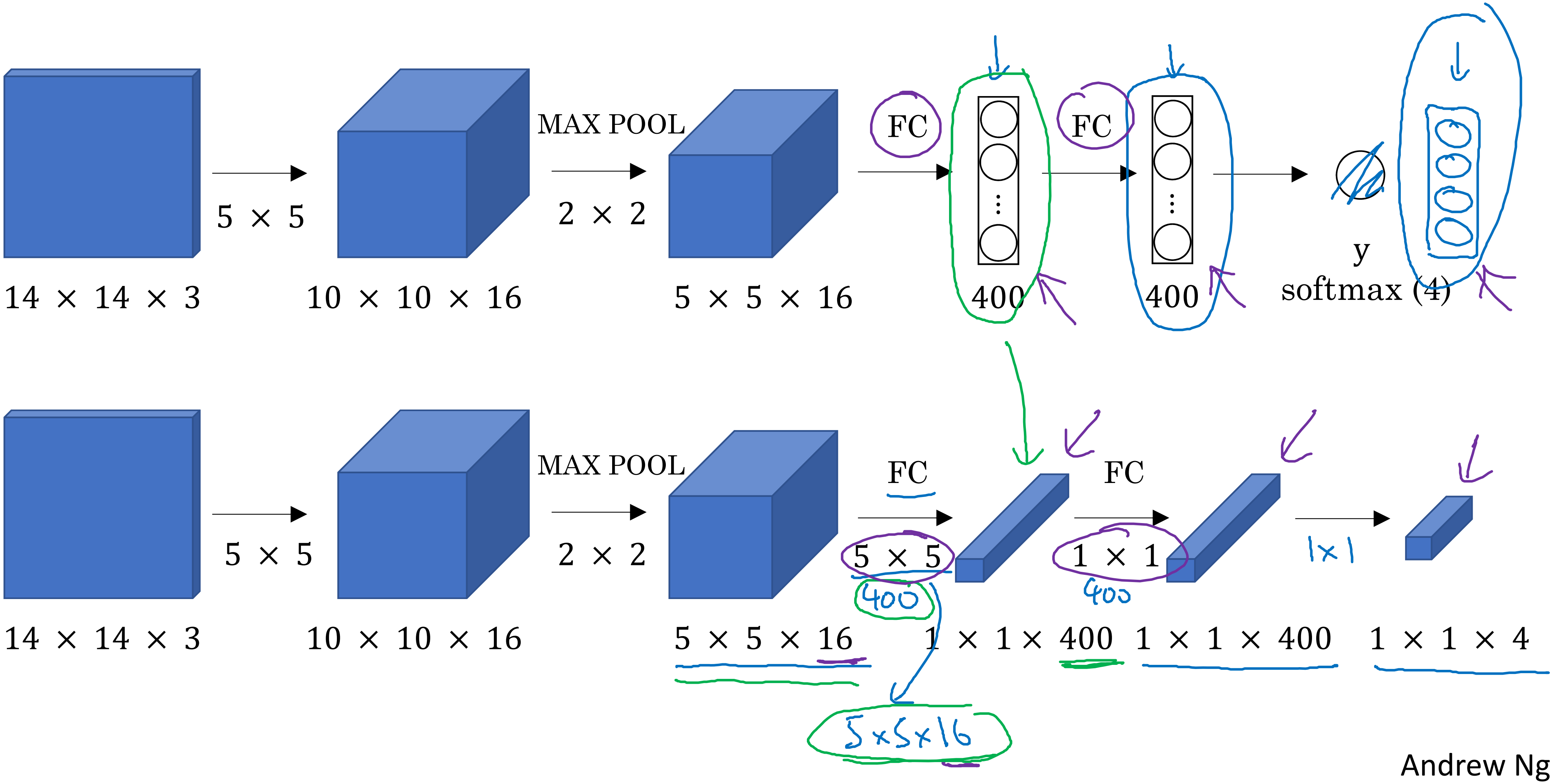
Andrew Ng

deeplearning.ai

Object Detection

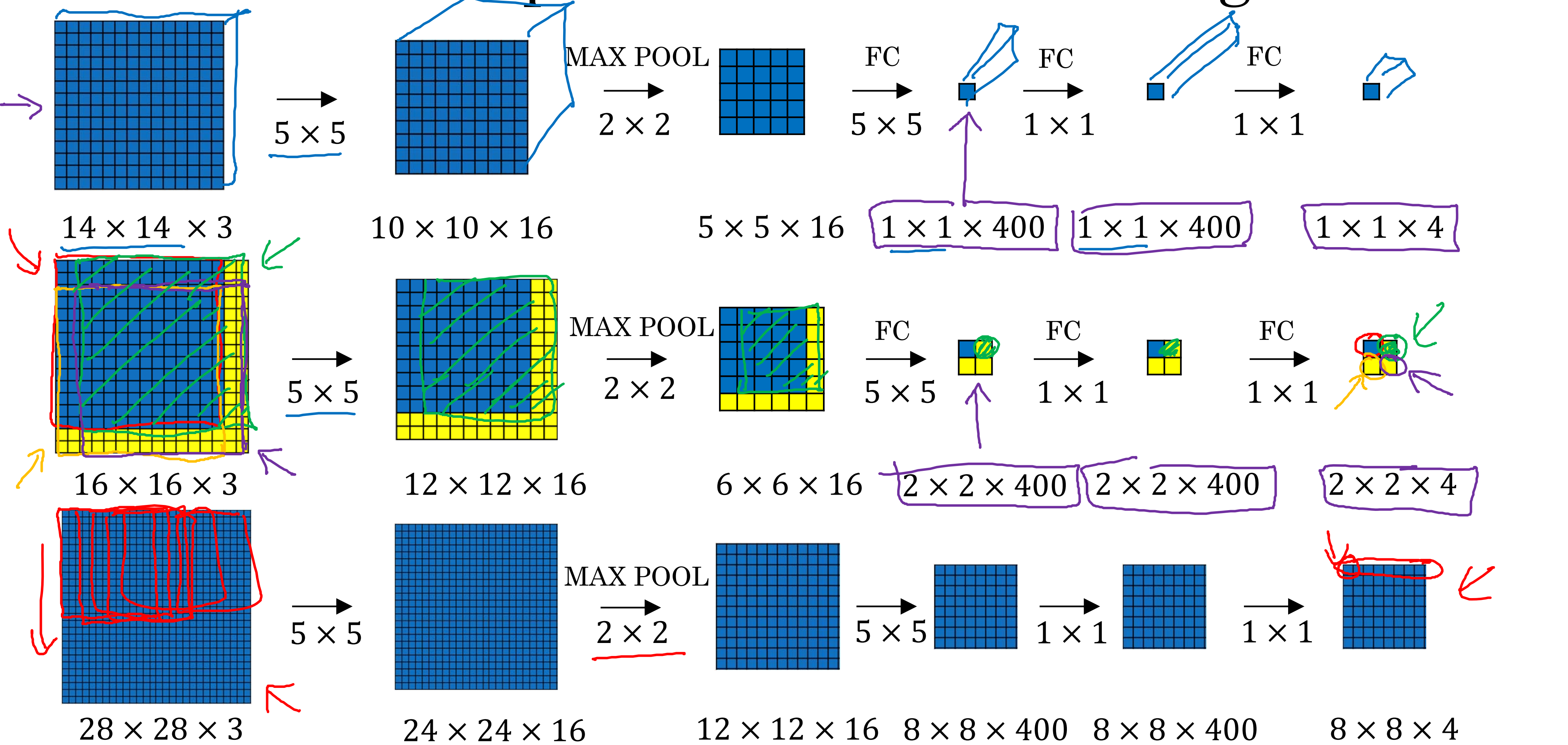
Convolutional implementation of sliding windows

Turning FC layer into convolutional layers



Andrew Ng

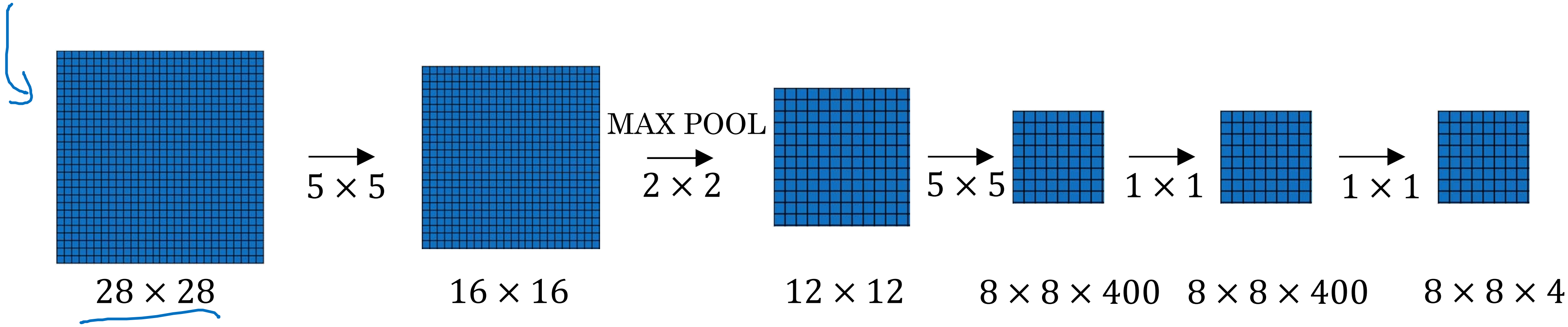
Convolution implementation of sliding windows



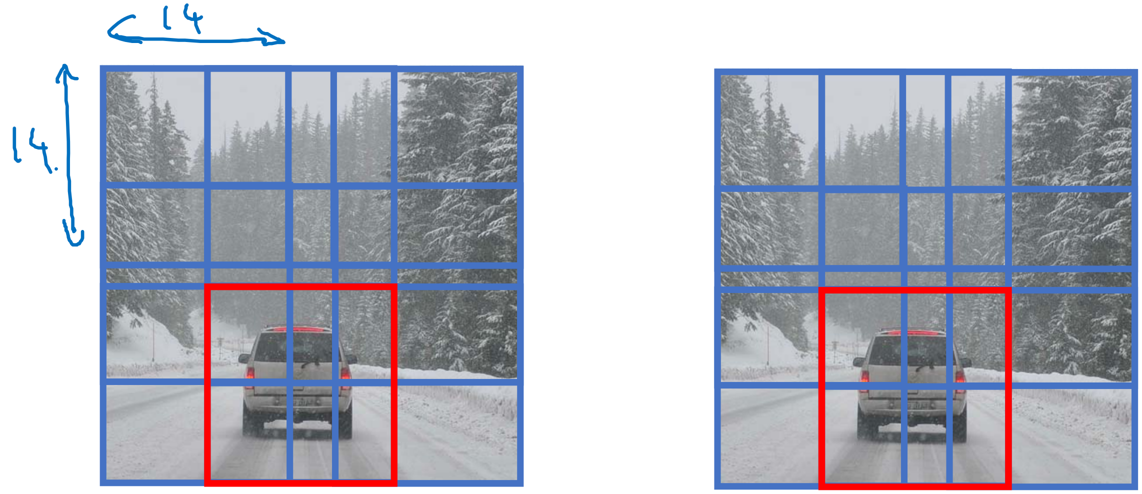
[Sermanet et al., 2014, OverFeat: Integrated recognition, localization and detection using convolutional networks]

Andrew Ng

Convolution implementation of sliding windows



28 × 28 $\xrightarrow{5 \times 5}$ 16 × 16 $\xrightarrow[2 \times 2]{\text{MAX POOL}}$ 12 × 12 $\xrightarrow{5 \times 5}$ 8 × 8 × 400 $\xrightarrow{1 \times 1}$ 8 × 8 × 400 $\xrightarrow{1 \times 1}$ 8 × 8 × 4

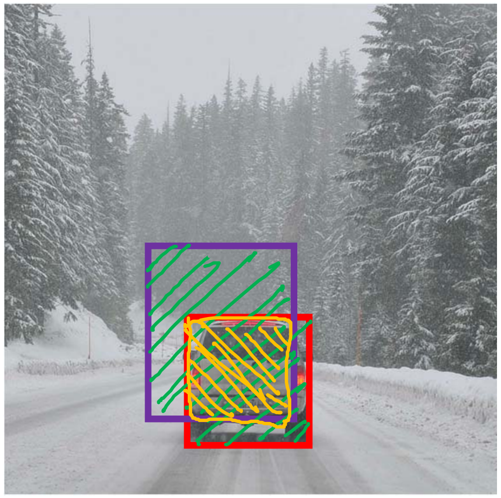


deeplearning.ai

Object Detection

Intersection over union

Evaluating object localization



Intersection over Union (IoU)

= size of 🟨 / size of 🟩

“Correct” if IoU ≥ $\boxed{0.5}$ ←

0.6 ←

More generally, IoU is a measure of the overlap between two bounding boxes.

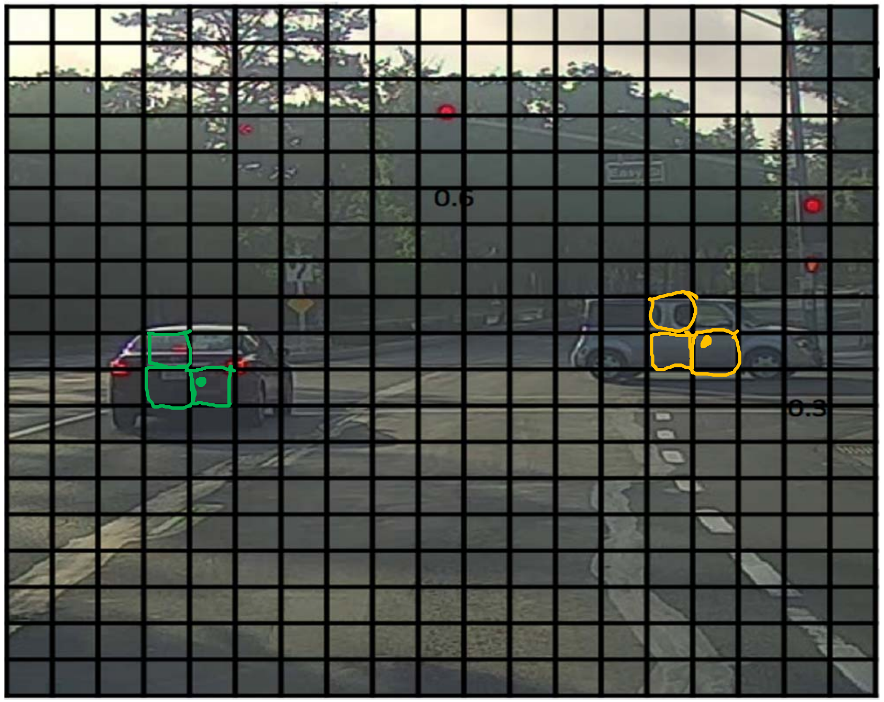
deeplearning.ai

Object Detection

Non-max suppression

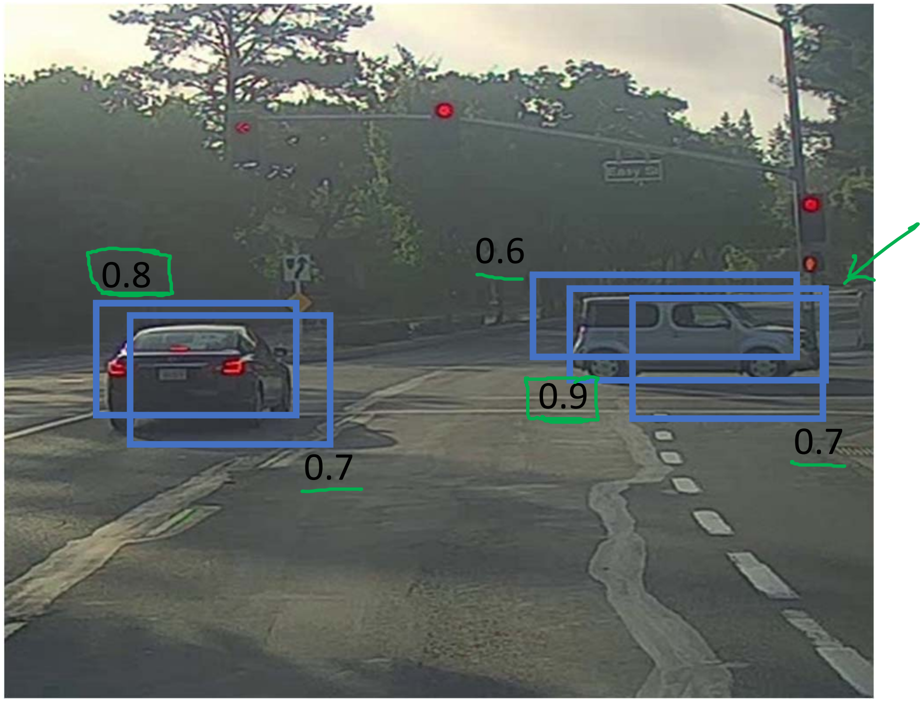
Non-max suppression example

Non-max suppression example



19×19

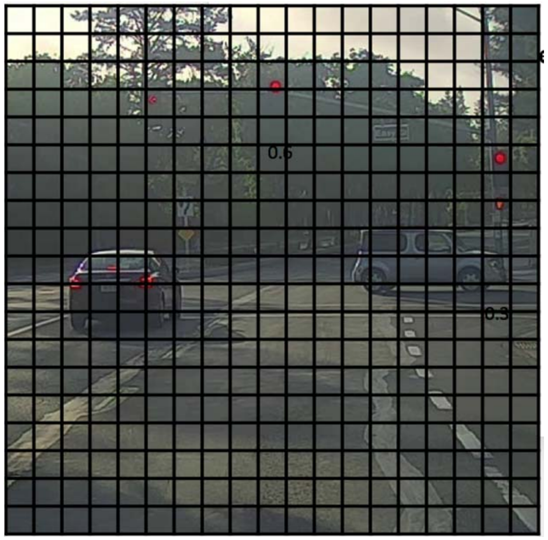
Non-max suppression example



P_c

Andrew Ng

Non-max suppression algorithm



$\underline{19 \times 19}$

Each output prediction is: $\begin{bmatrix} \boxed{p_c} \\ \boxed{\begin{matrix} b_x \\ b_y \\ b_h \\ b_w \end{matrix}} \end{bmatrix}$ $\boxed{\begin{matrix} \to c_1 \\ \to c_2 \\ \to c_3 \end{matrix}}$

Discard all boxes with $\underline{p_c \leq 0.6}$

While there are any remaining boxes:

- Pick the box with the largest $\underline{p_c}$ Output that as a $\underline{\text{prediction}}$.
- Discard any $\underline{\text{remaining box}}$ with $\underline{\text{IoU} \geq 0.5}$ with the box output in the previous step

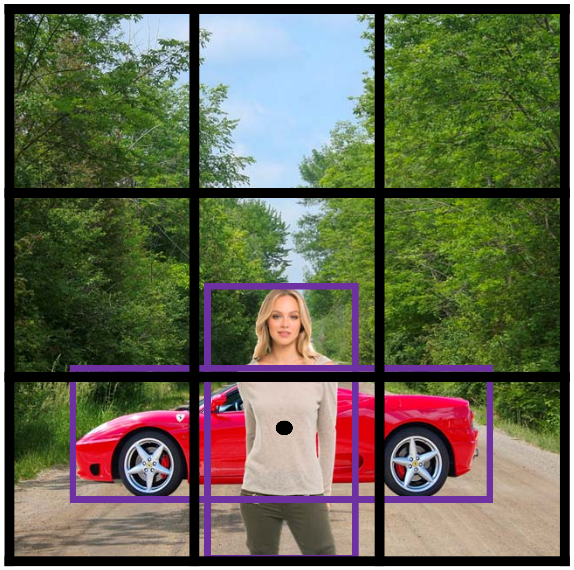
Andrew Ng

deeplearning.ai

Object Detection

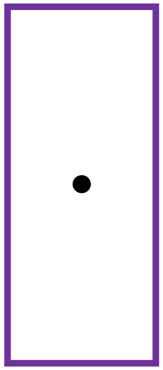
Anchor boxes

Overlapping objects:

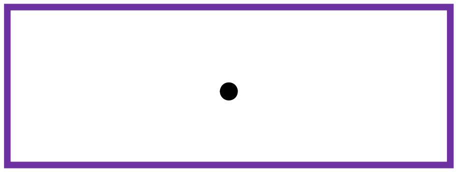


$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ \left.\begin{matrix} c_1 \\ c_2 \\ c_3 \end{matrix}\right\} \end{bmatrix}$$

Anchor box 1:



Anchor box 2:



$$y = \left[\begin{array}{l} \left.\begin{matrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{matrix}\right\} \text{anchor box 1} \\ \left.\begin{matrix} p_c \\ b_x \\ \vdots \\ c_3 \end{matrix}\right\} \text{anchor box 2} \end{array}\right]$$

[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]

Andrew Ng

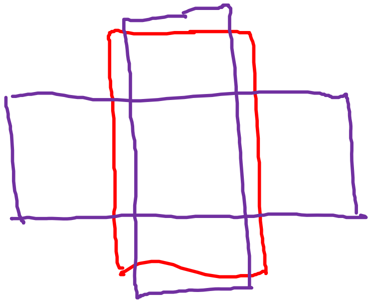
Anchor box algorithm

Previously:

Each object in training image is assigned to grid cell that contains that object's midpoint.

Output y:

$\underline{3\times 3}\times \underline{8}$



With two anchor boxes:

Each object in training image is assigned to grid cell that contains object's midpoint and anchor box for the grid cell with highest IoU.

(grid cell, anchor box)

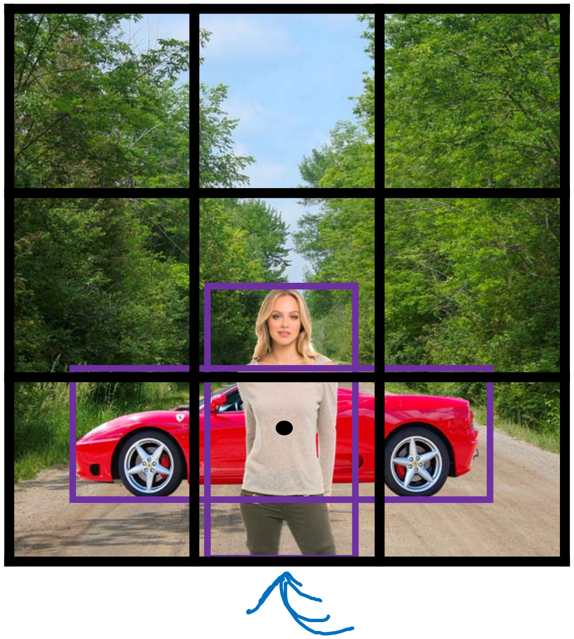
Output y:

$3\times 3\times \underline{16}$

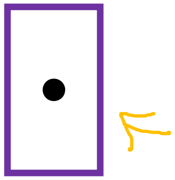
$3\times 3\times \underline{2}\times \underline{8}$

Andrew Ng

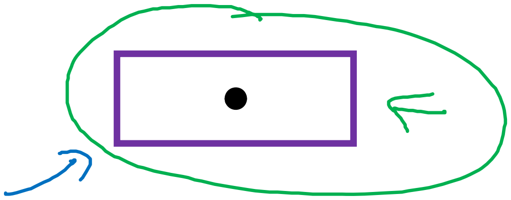
Anchor box example



Anchor box 1:



Anchor box 2:



$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 1 \\ 0 \\ 0 \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

car only?

$$\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{matrix}}\right\} \text{anchor box 1} \\ \left.\vphantom{\begin{matrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{matrix}}\right\} \text{anchor box 2} \end{matrix}$$

Andrew Ng

deeplearning.ai

Object Detection

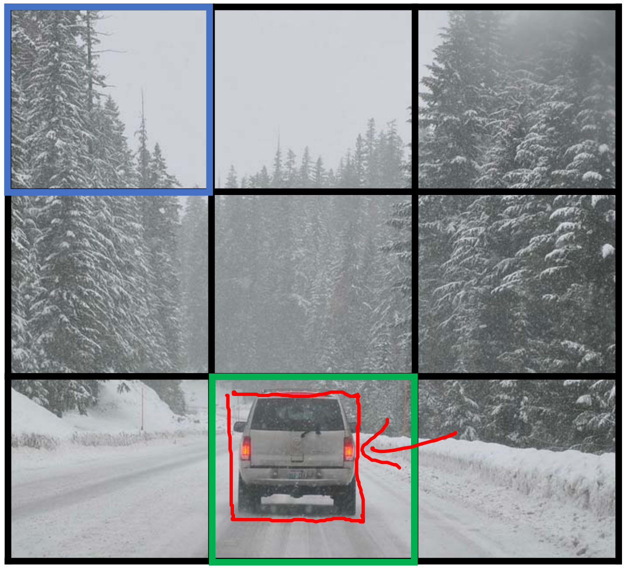
Putting it together: YOLO algorithm

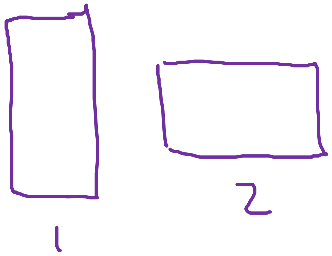
Training

1 - pedestrian

2 - car

3 - motorcycle





$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \qquad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

3 × 3 × 16

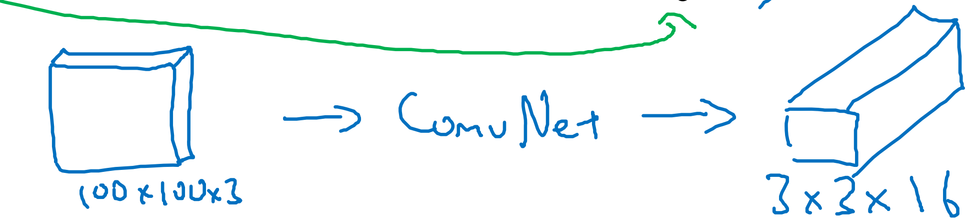
y is $3 \times 3 \times 2 \times 8$

19×19×16

19×19×40

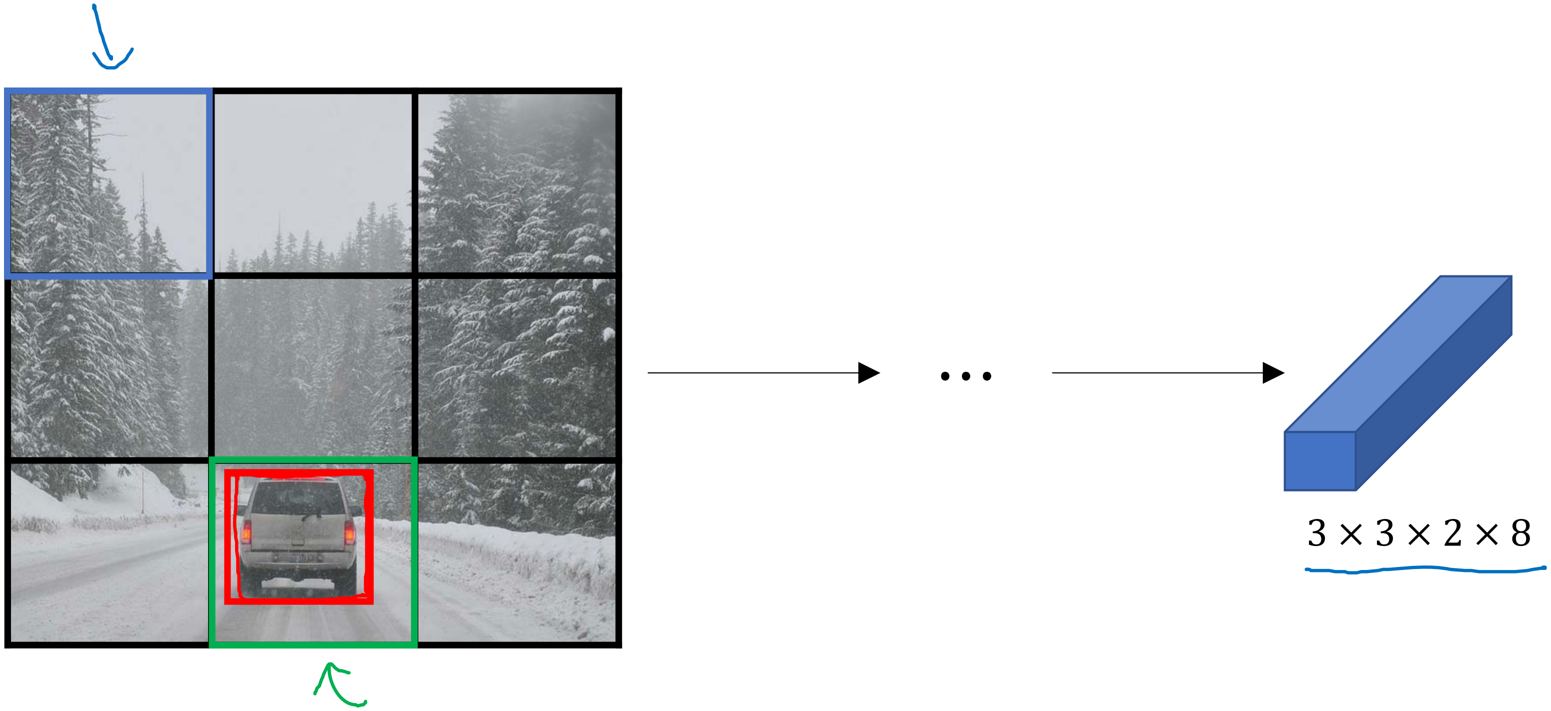
#anchors

5 + #classes

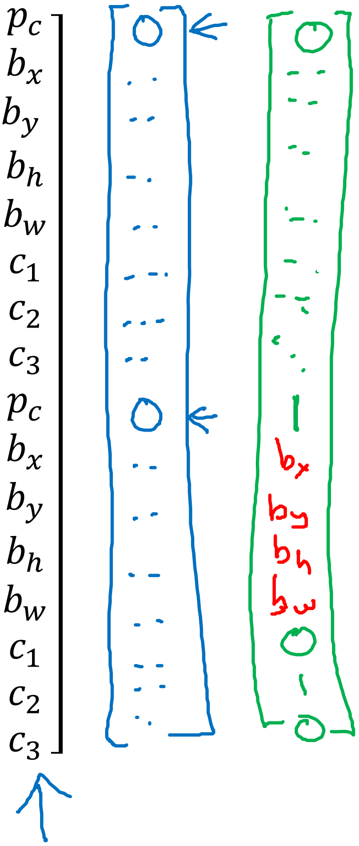


[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]

Making predictions

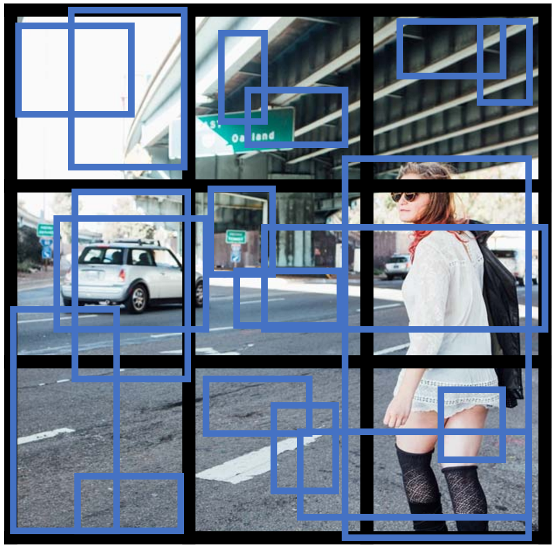


$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$



Andrew Ng

Outputting the non-max supressed outputs



- For each grid call, get 2 predicted bounding boxes.

- Get rid of low probability predictions.

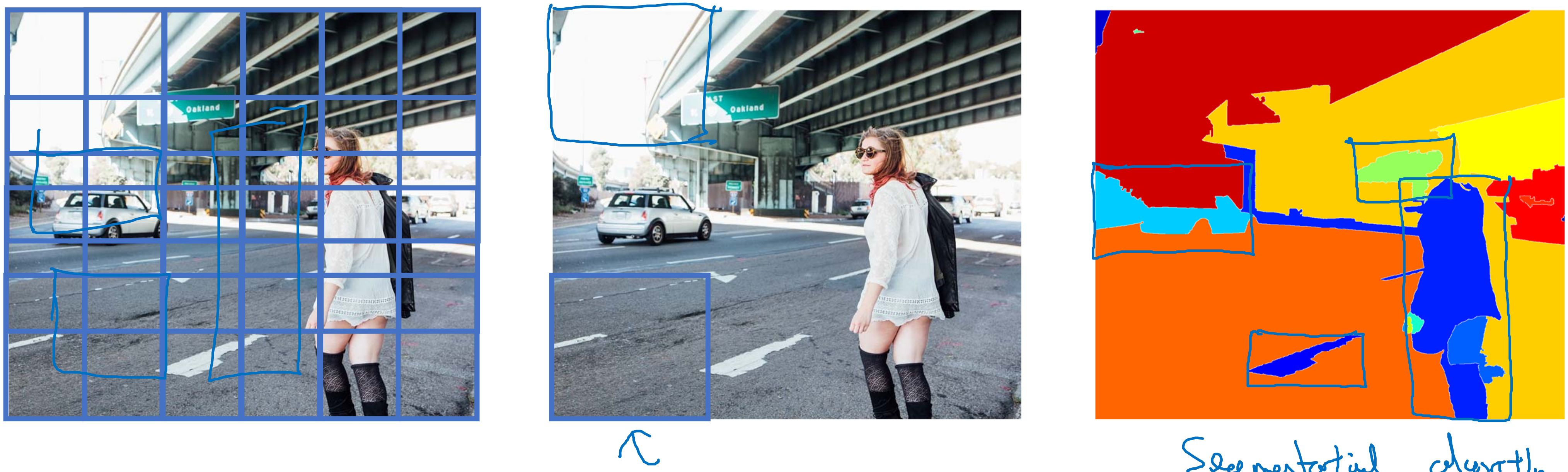
- For each class (pedestrian, car, motorcycle) use non-max suppression to generate final predictions.

deeplearning.ai

Object Detection

Region proposals (Optional)

Region proposal: R-CNN



Segmentation algorithm

~2,000

[Girshik et. al, 2013, Rich feature hierarchies for accurate object detection and semantic segmentation]

Faster algorithms

R-CNN: Propose regions. Classify proposed regions one at a time. Output label + bounding box.

Fast R-CNN: Propose regions. Use convolution implementation of sliding windows to classify all the proposed regions.

Faster R-CNN: Use convolutional network to propose regions.

[Girshik et. al, 2013. Rich feature hierarchies for accurate object detection and semantic segmentation]
[Girshik, 2015. Fast R-CNN]
[Ren et. al, 2016. Faster R-CNN: Towards real-time object detection with region proposal networks]

Andrew Ng