Chapter 5: Object Detection

deeplearning.ai

Object Detection

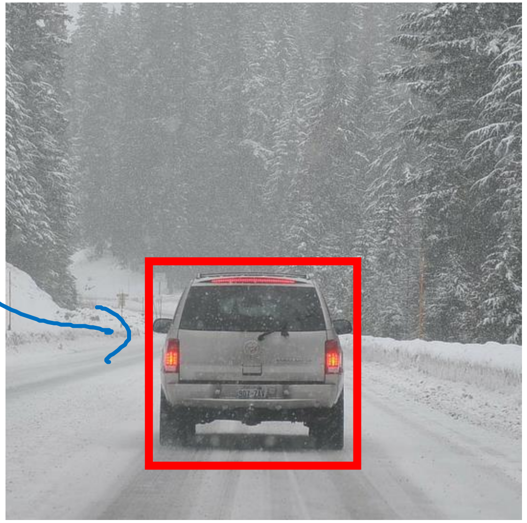
Object localization

What are localization and detection?

Image classification

"Car"

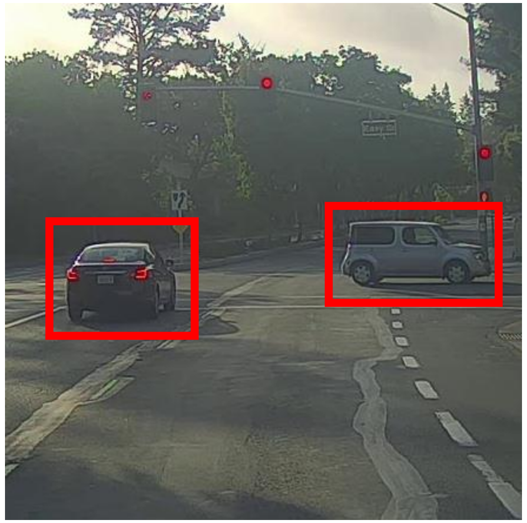
Classification with localization



"Car"

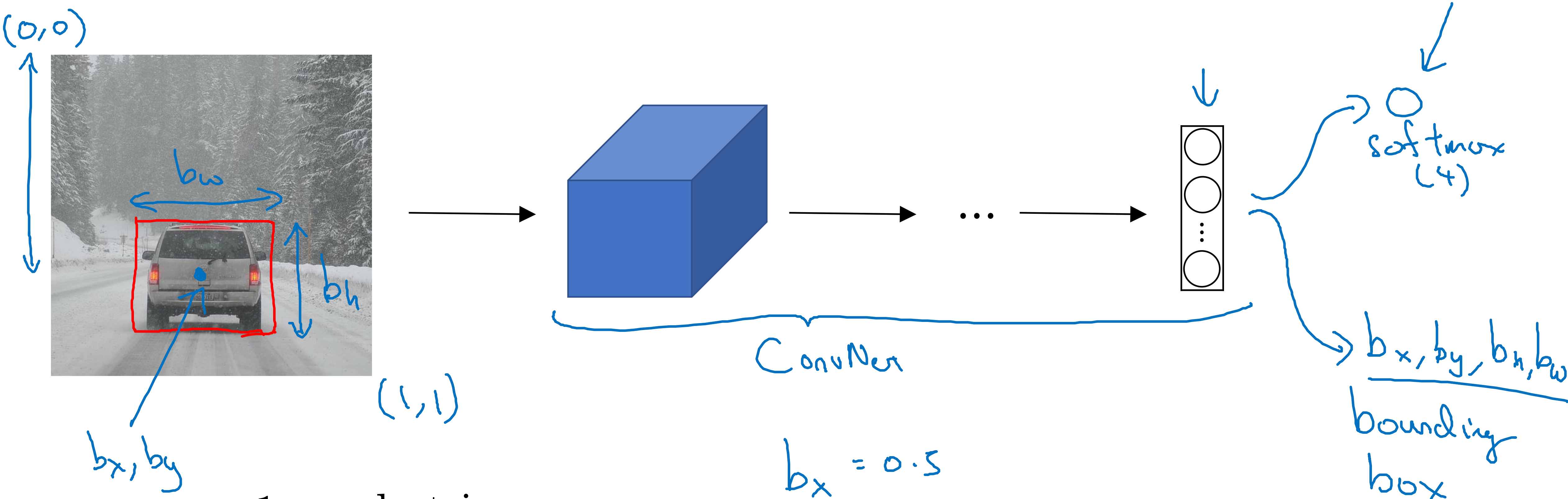
1 object

Detection



multiple objects

Classification with localization



1 - pedestrian ←

2 - car ←

3 - motorcycle ←

4 - background

$b_x = 0.5$

$b_y = 0.7$

$b_h = 0.3$

$b_w = 0.4$

Defining the target label y

1 - pedestrian

2 - car

3 - motorcycle

4 - background

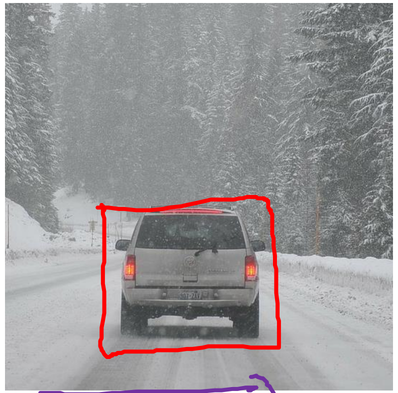
Need to output b_x, b_y, b_h, b_w, class label (1-4)

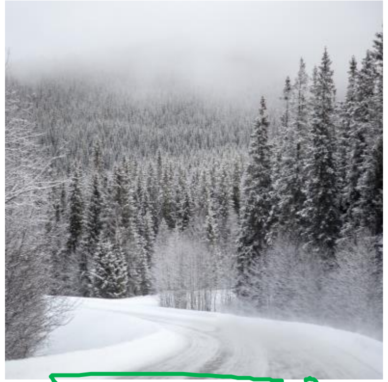
$$\mathcal{L}(\hat{y},y) = \begin{cases} (\hat{y}_1-y_1)^2+(\hat{y}_2-y_2)^2 + \dots + (\hat{y}_8-y_8)^2 & \text{if } y_1=1 \\ (\hat{y}_1-y_1)^2 & \text{if } y_1=0 \end{cases}$$

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

p_c: is there any object?

x =





$$(x, y) \qquad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

p_c

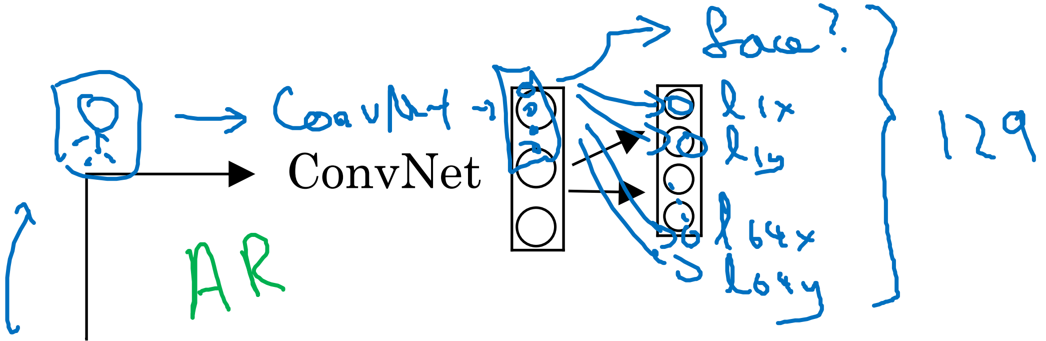
"don't care"

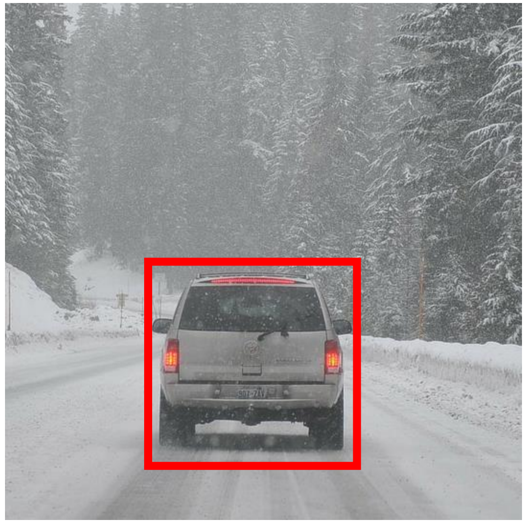
deeplearning.ai

Object Detection

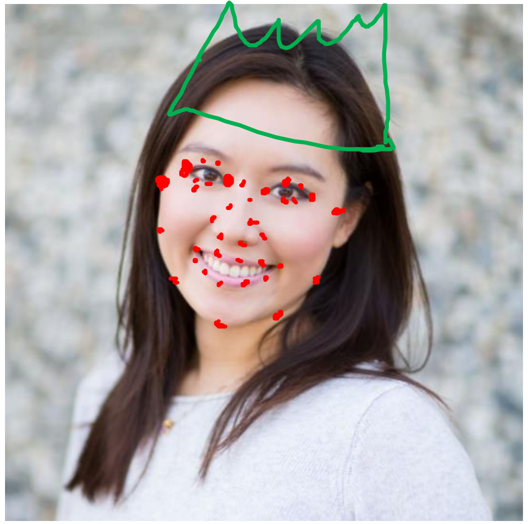
Landmark detection

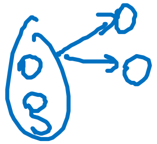
Landmark detection



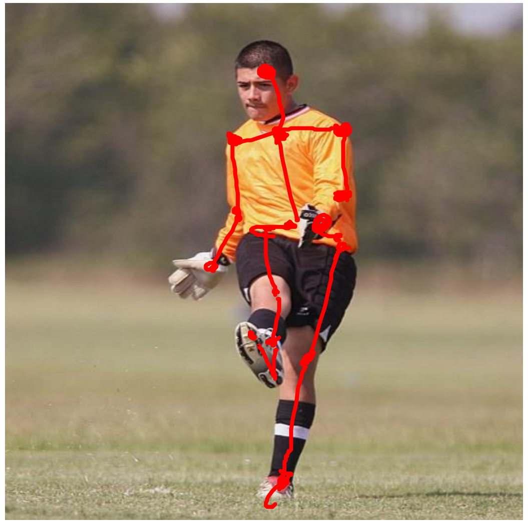


b_x, b_y, b_h, b_w





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



$l_{1x}, l_{1y},$

$\vdots$

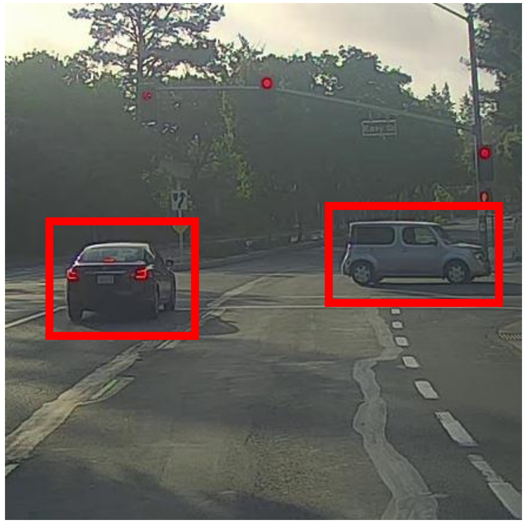
$l_{32x}\ l_{32y}$

deeplearning.ai

Object Detection

Object detection

Car detection example



Training set:

x	y
	1
	1
	1
	0
	0

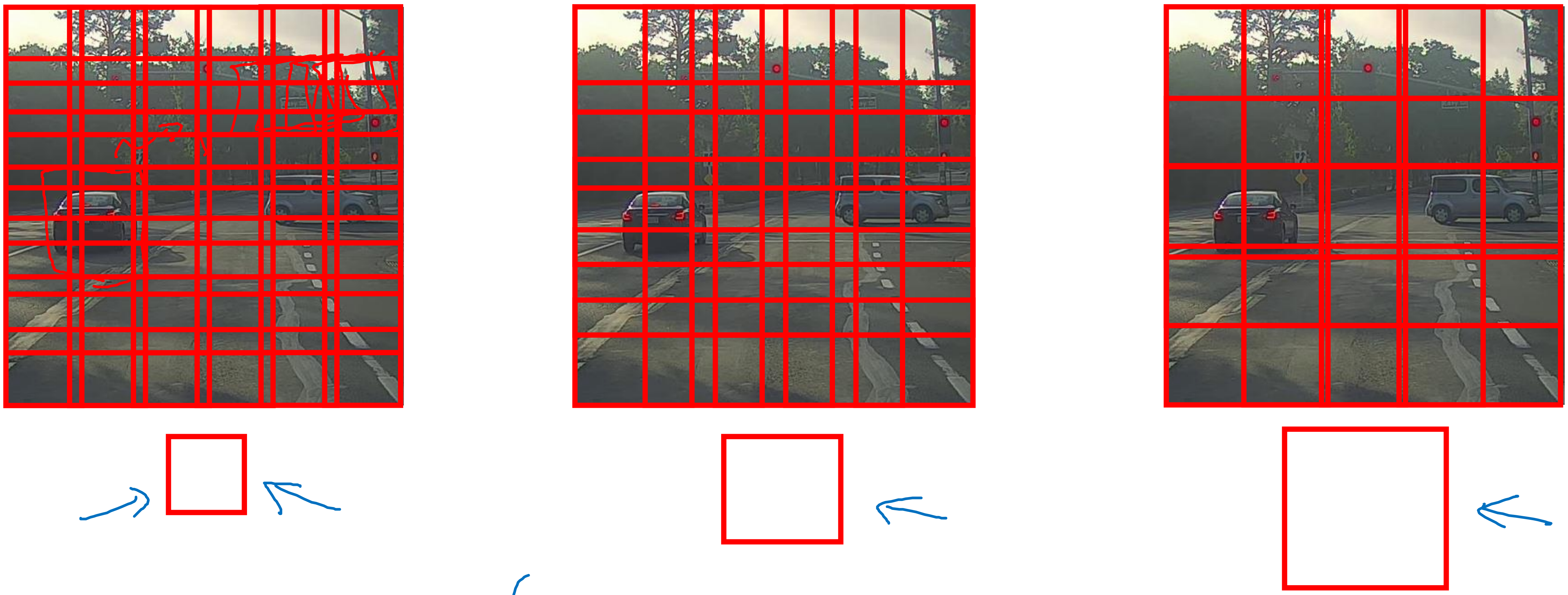
→ ConvNet → y

Andrew Ng

Sliding windows detection

→ ConvNet → 0

→ ConvNet



Computation cost

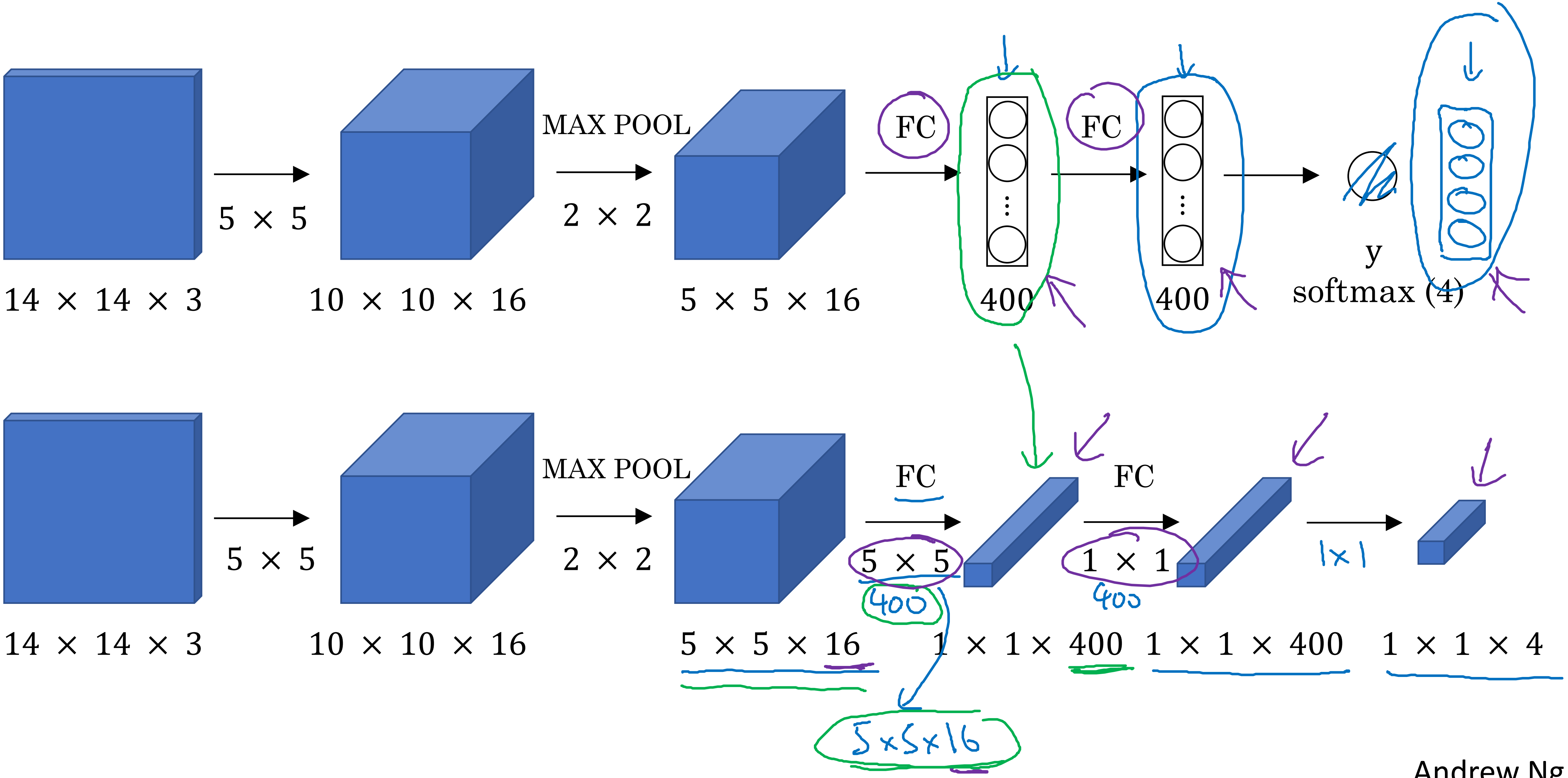
Andrew Ng

deeplearning.ai

Object Detection

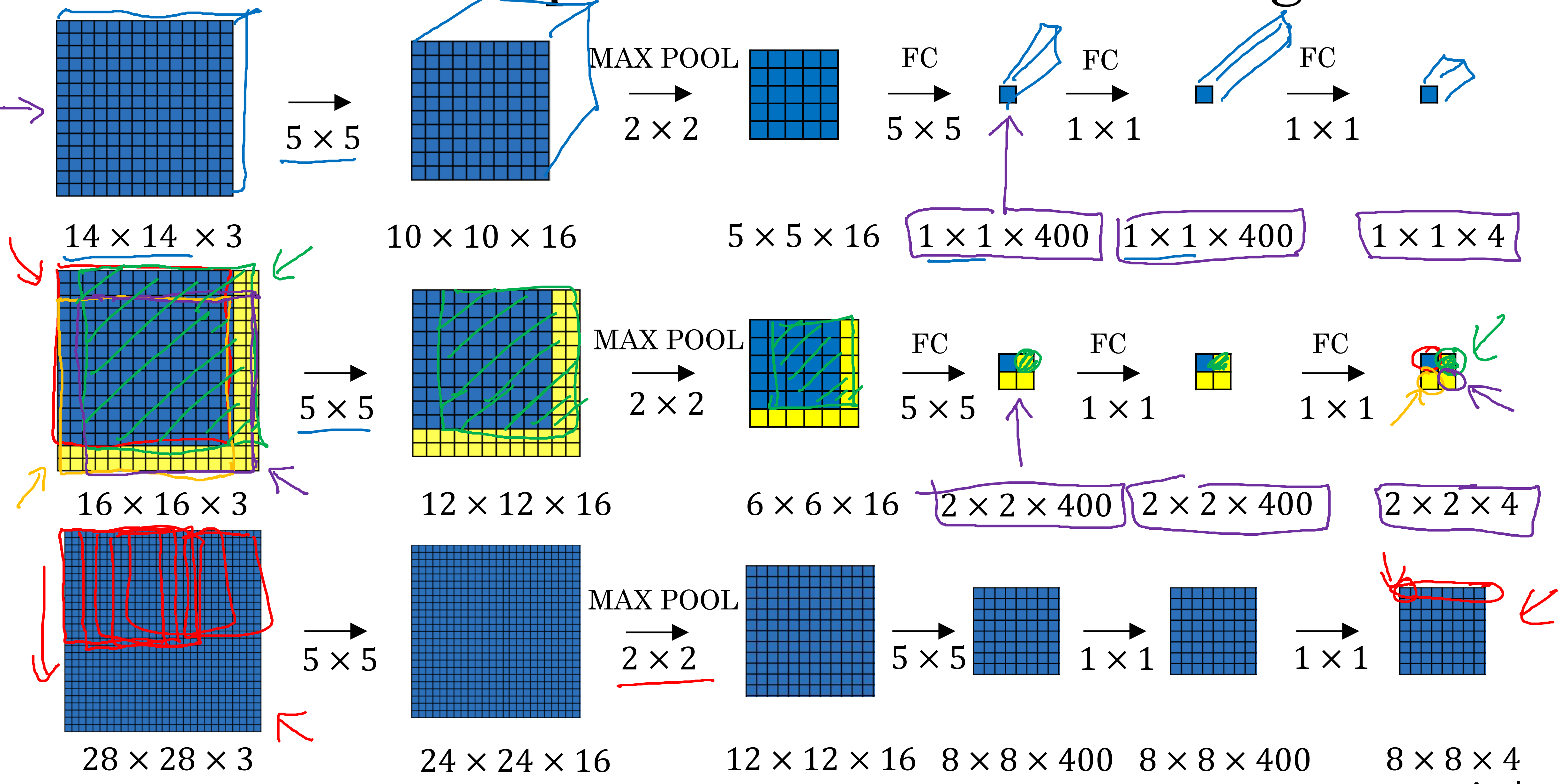
Convolutional implementation of sliding windows

Turning FC layer into convolutional layers



Andrew Ng

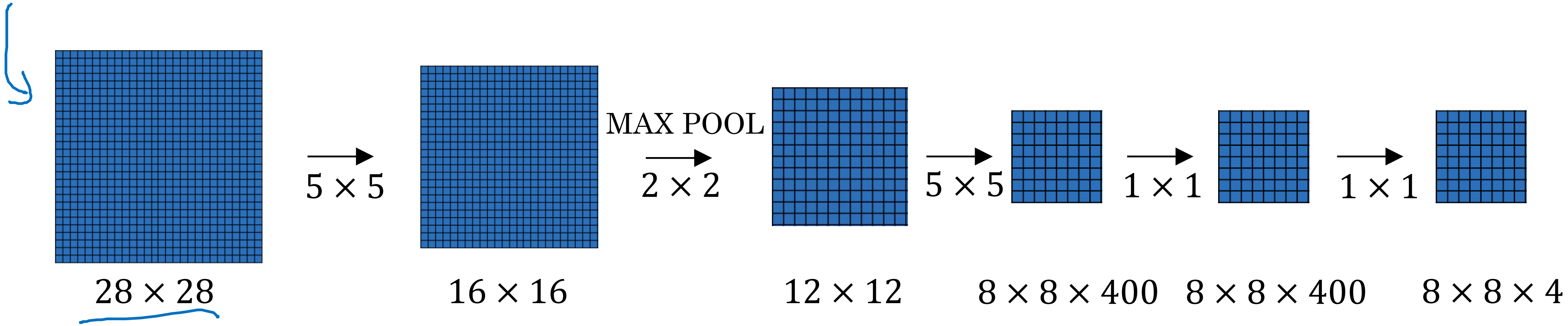
Convolution implementation of sliding windows



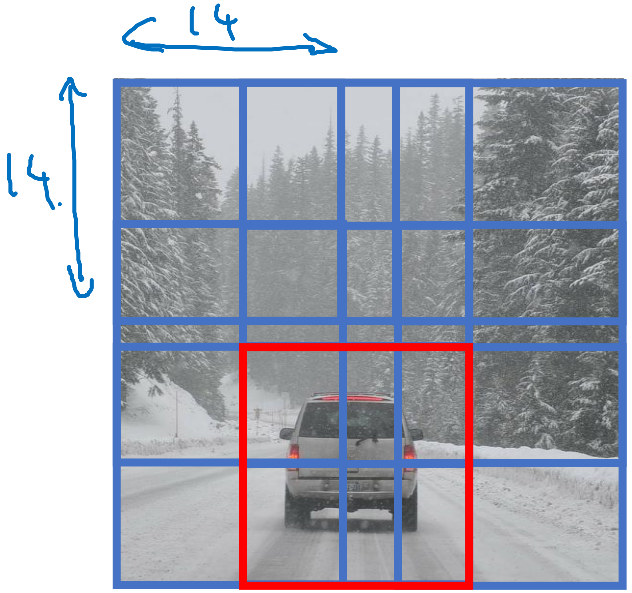
[Sermanet et al., 2014, OverFeat: Integrated recognition, localization and detection using convolutional networks]

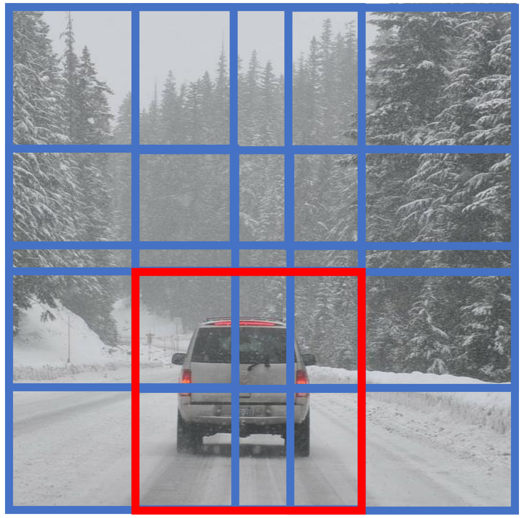
Andrew Ng

Convolution implementation of sliding windows



28 × 28 → 5 × 5 → 16 × 16 → MAX POOL 2 × 2 → 12 × 12 → 5 × 5 → 8 × 8 × 400 → 1 × 1 → 8 × 8 × 400 → 1 × 1 → 8 × 8 × 4



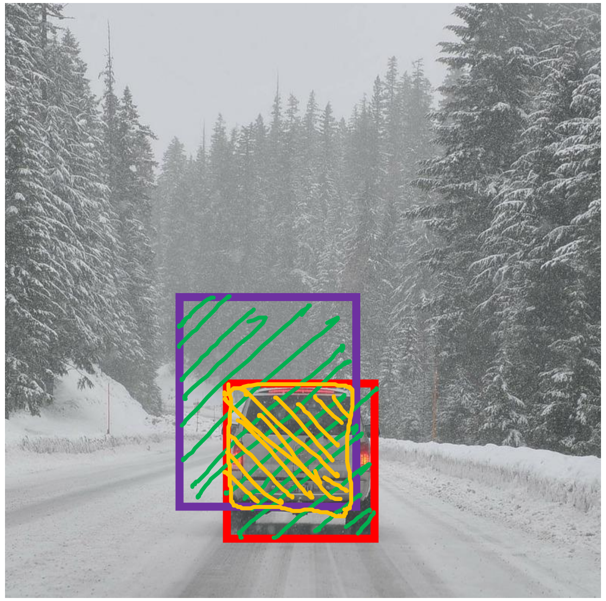


deeplearning.ai

Object Detection

Intersection over union

Evaluating object localization



Intersection over Union (IoU)

$$= \frac{\text{size of } 🟨}{\text{size of } 🟩}$$

"Correct" if $\underline{\text{IoU} \geq \boxed{0.5}}$ ←

$\underline{0.6}$ ←

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

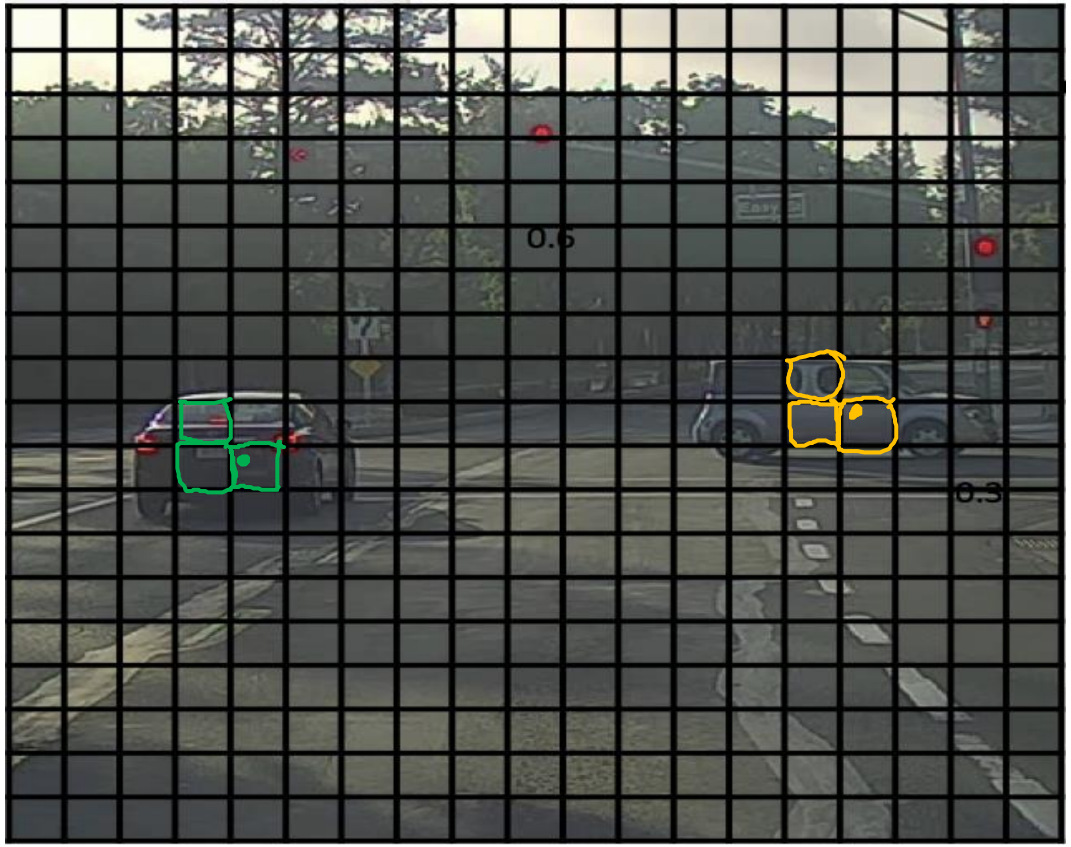
deeplearning.ai

Object Detection

Non-max suppression

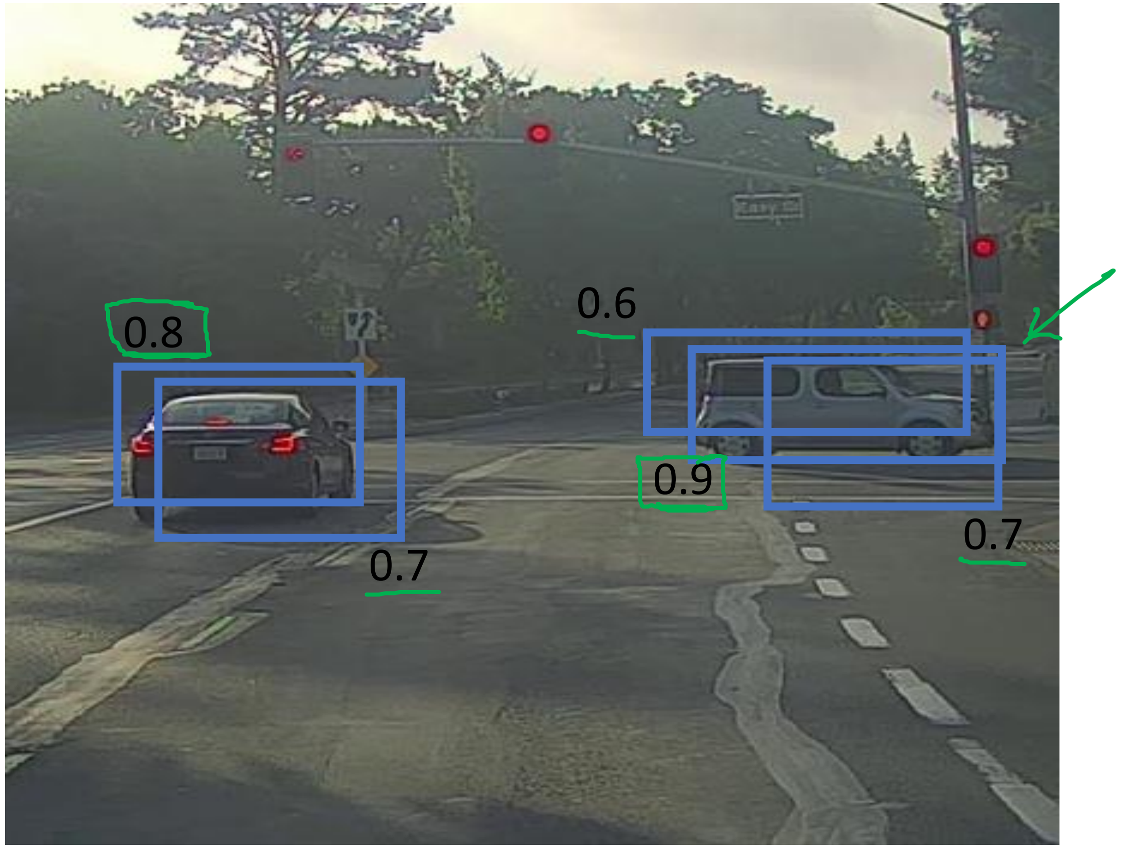
Non-max suppression example

Non-max suppression example



19x19

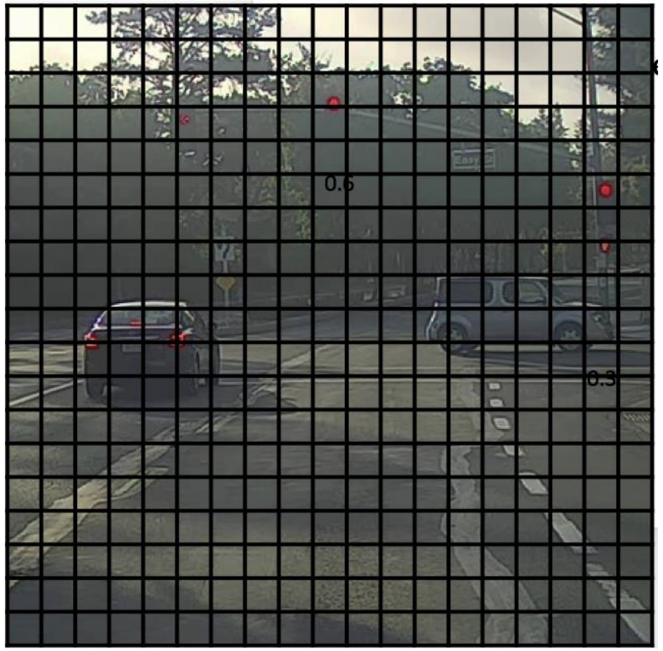
Non-max suppression example



p_c

Andrew Ng

Non-max suppression algorithm



19× 19

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

Discard all boxes with $p_c \leq 0.6$

While there are any remaining boxes:

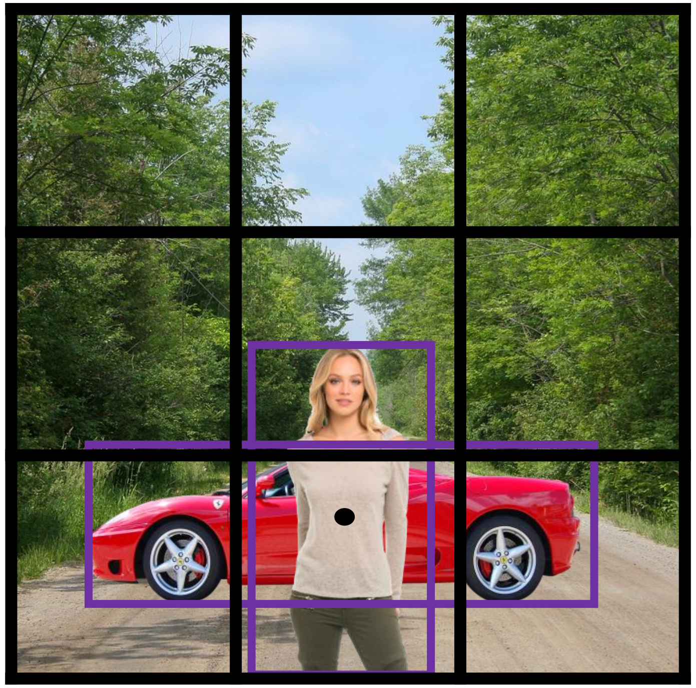
- Pick the box with the largest p_c Output that as a prediction.
- Discard any remaining box with IoU $\geq$ 0.5 with the box output in the previous step

deeplearning.ai

Object Detection

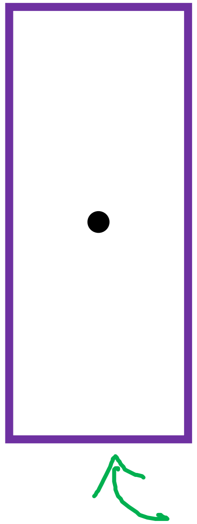
Anchor boxes

Overlapping objects:

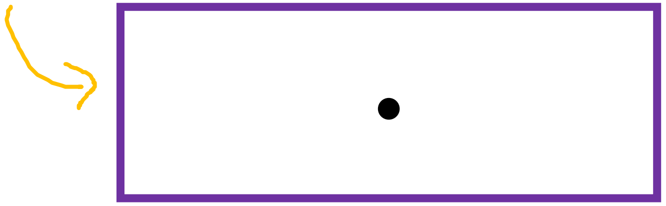


$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Anchor box 1:



Anchor box 2:



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

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

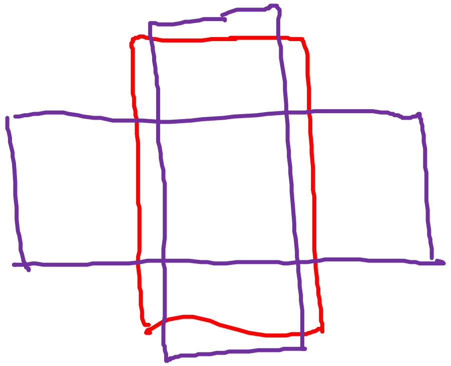
Anchor box algorithm

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)

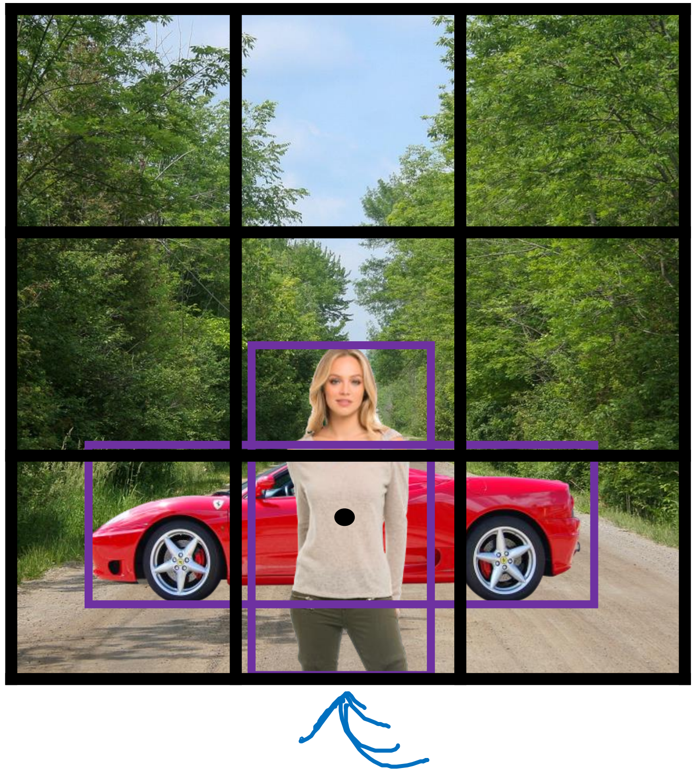
Output y:

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

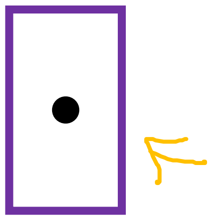
$3 \times 3 \times \underline{2} \times \underline{8}$

Andrew Ng

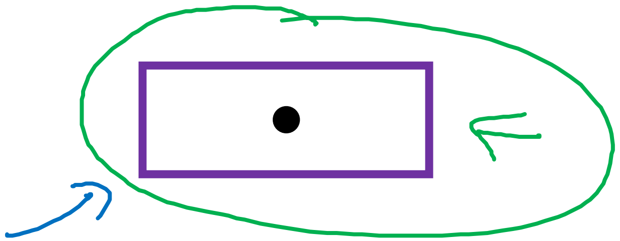
Anchor box example



Anchor box 1:



Anchor box 2:



$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 1 \\ 0 \\ 0 \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \overset{\text{car only?}}{\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}} \begin{matrix} \left.\vphantom{\begin{matrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{matrix}}\right\} \text{anchor box 1} \\ \left.\vphantom{\begin{matrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{matrix}}\right\} \text{anchor box 2} \end{matrix}$$

YOLO — You Only Look Once

YOLO — You Only Look Once

- Original paper (CVPR 2016. OpenCV People’s Choice Award) https://arxiv.org/pdf/1506.02640v5.pdf
- YOLOv2: https://arxiv.org/pdf/1612.08242v1.pdf
- YOLOv3: https://arxiv.org/pdf/1804.02767.pdf
- Open source: https://pjreddie.com/darknet/yolo/