deeplearning.ai

Object Detection

Object localization

What are localization and detection?

Image classification

"Car"

Classification with localization

"Car"

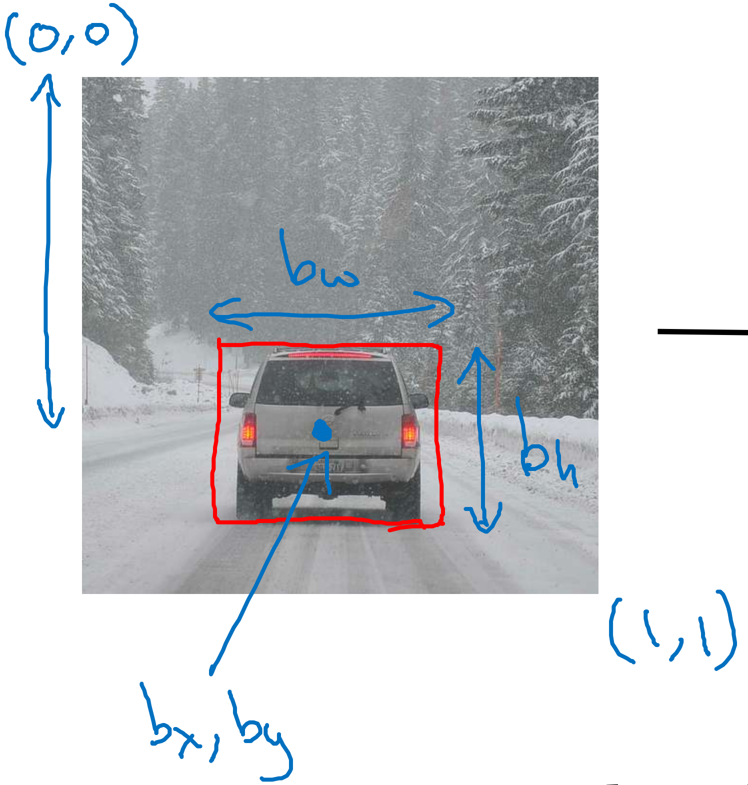
1 object

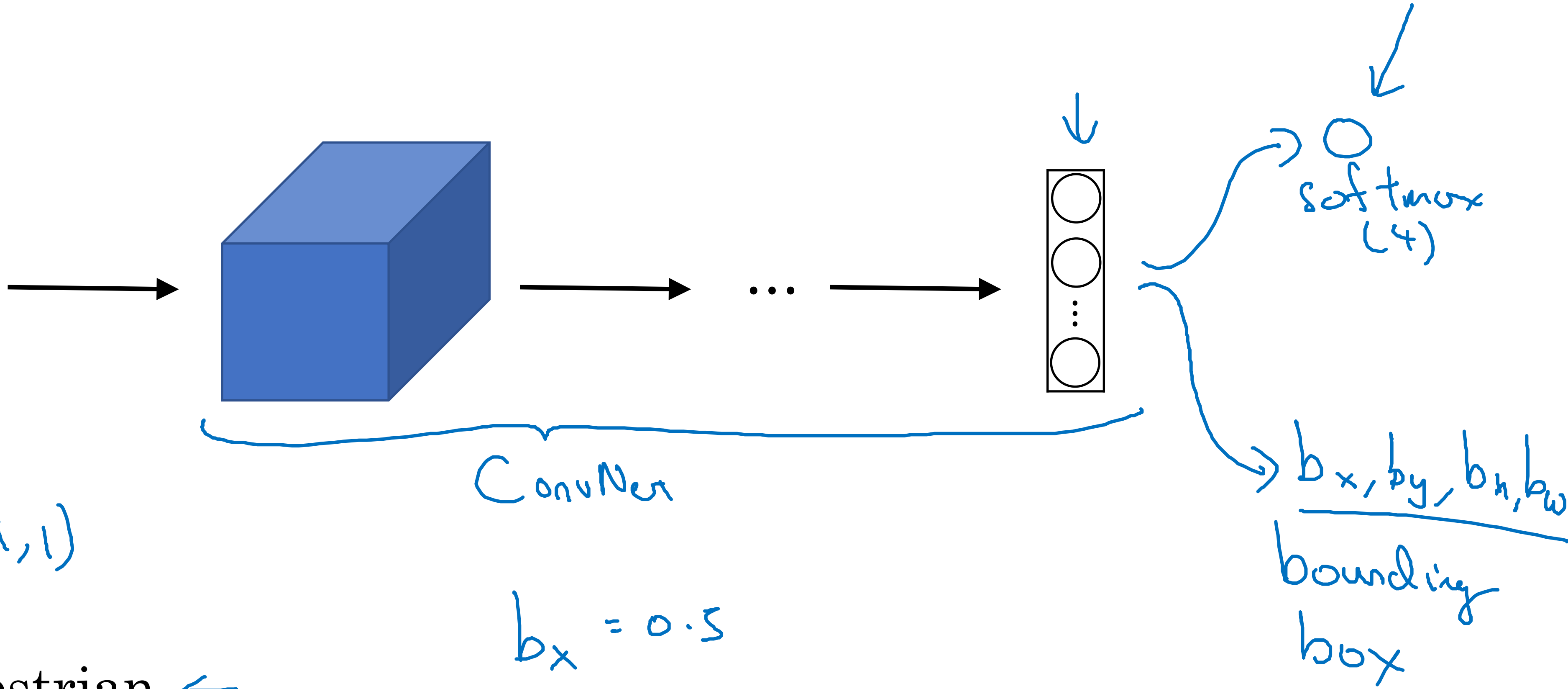
Detection

multiple objects

Andrew Ng

Classification with localization





1 - pedestrian ←
2 - car ←
3 - motorcycle ←
4 - $\boxed{\text{background}}$

$b_x = 0.5$
$b_y = 0.7$
$b_h = 0.3$
$b_w = 0.4$

Defining the target label y

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

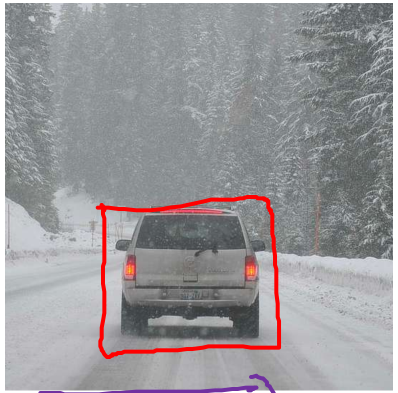
Need to output b_x, b_y, b_h, b_w, class label (1-4)

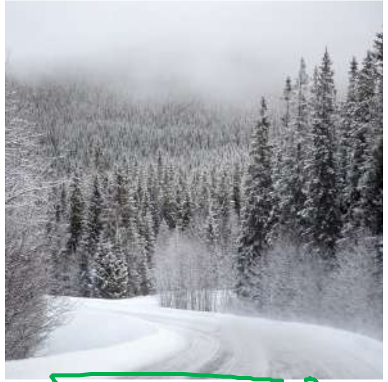
$$\mathcal{L}(\hat{y}, y) = \begin{cases} (\hat{y}_1-y_1)^2 + (\hat{y}_2-y_2)^2 + \dots + (\hat{y}_8-y_8)^2 & \text{if } y_1=1 \\ (\hat{y}_1-y_1)^2 & \text{if } y_1 = 0 \end{cases}$$

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

is there any object?

$x =$





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

p_c

"don't care"

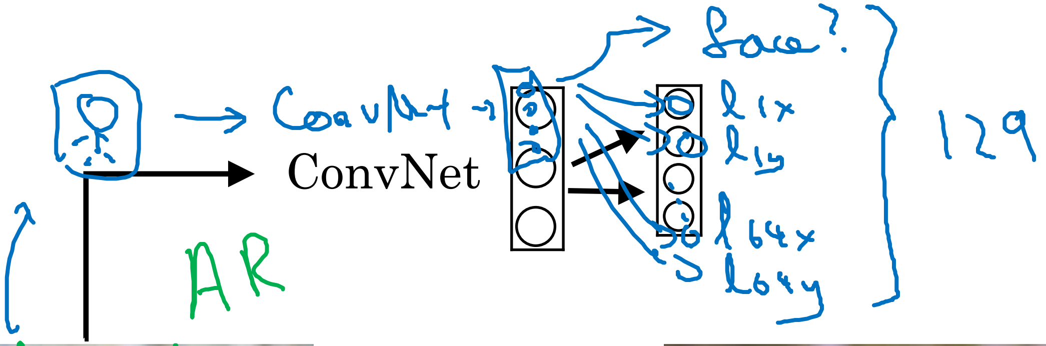
(x, y)

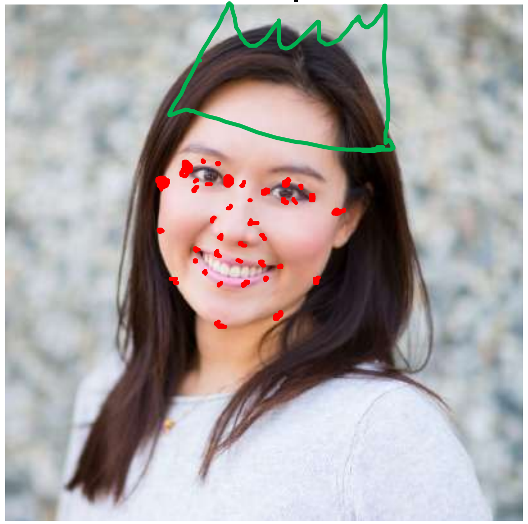
deeplearning.ai

Object Detection

Landmark detection

Landmark detection





b_x, b_y, b_h, b_w

$l_{1x}, l_{1y},$
$l_{2x}, l_{2y},$
$l_{3x}, l_{3y},$
l_{4x}, l_{4y}
$\vdots$
l_{64}, l_{64y}

X, y

$l_{1x}, l_{1y},$
$\vdots$
$l_{32x}\ l_{32y}$

Andrew Ng

deeplearning.ai

Object Detection

Object detection

Car detection example

Training set:

x	y
	1
	1
	1
	0
	0

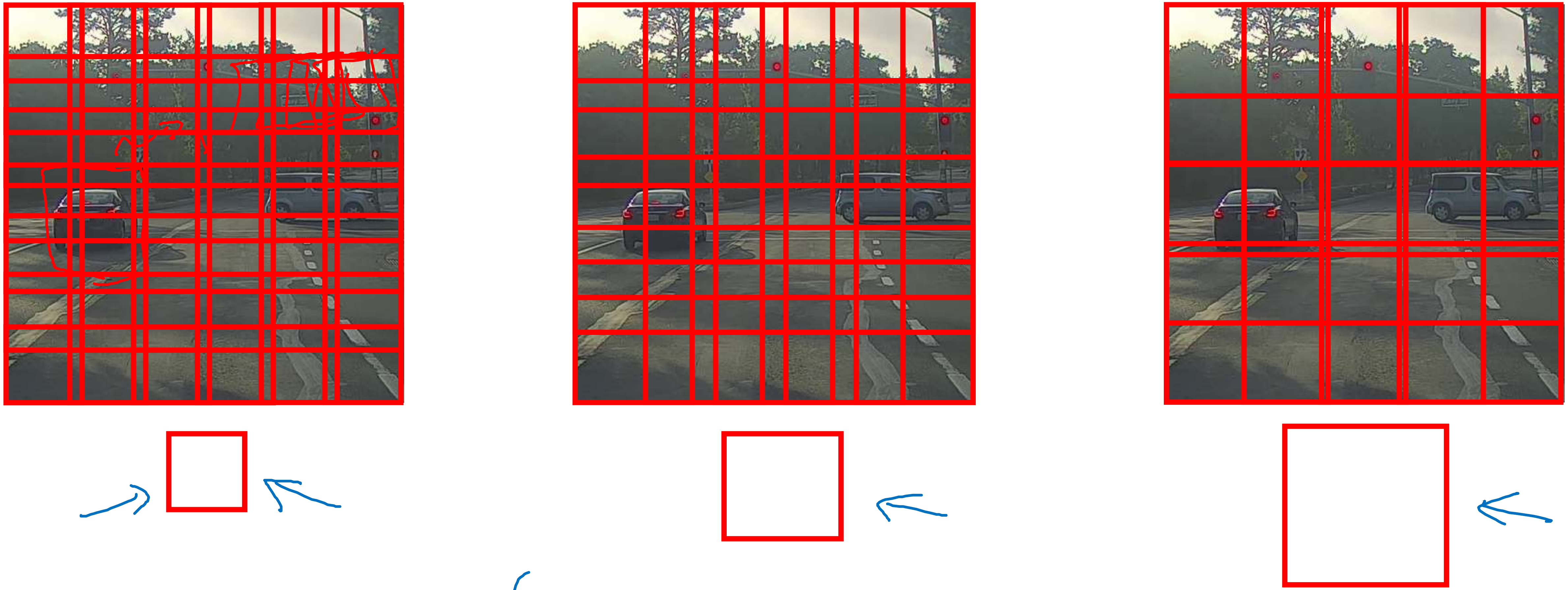
 → ConvNet → y

Andrew Ng

Sliding windows detection

→ ConvNet → 0

→ ConvNet



Computation cost

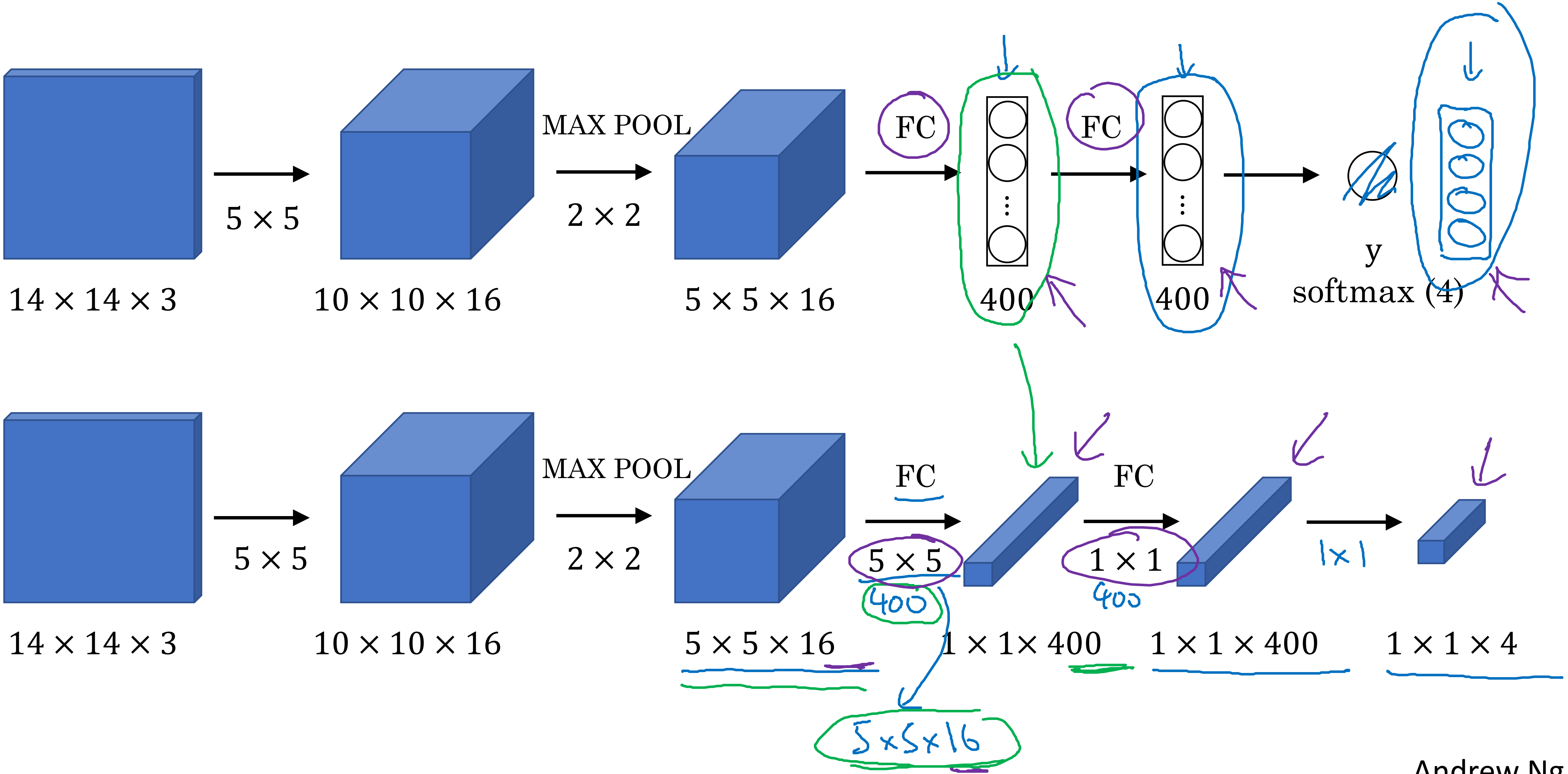
Andrew Ng

deeplearning.ai

Object Detection

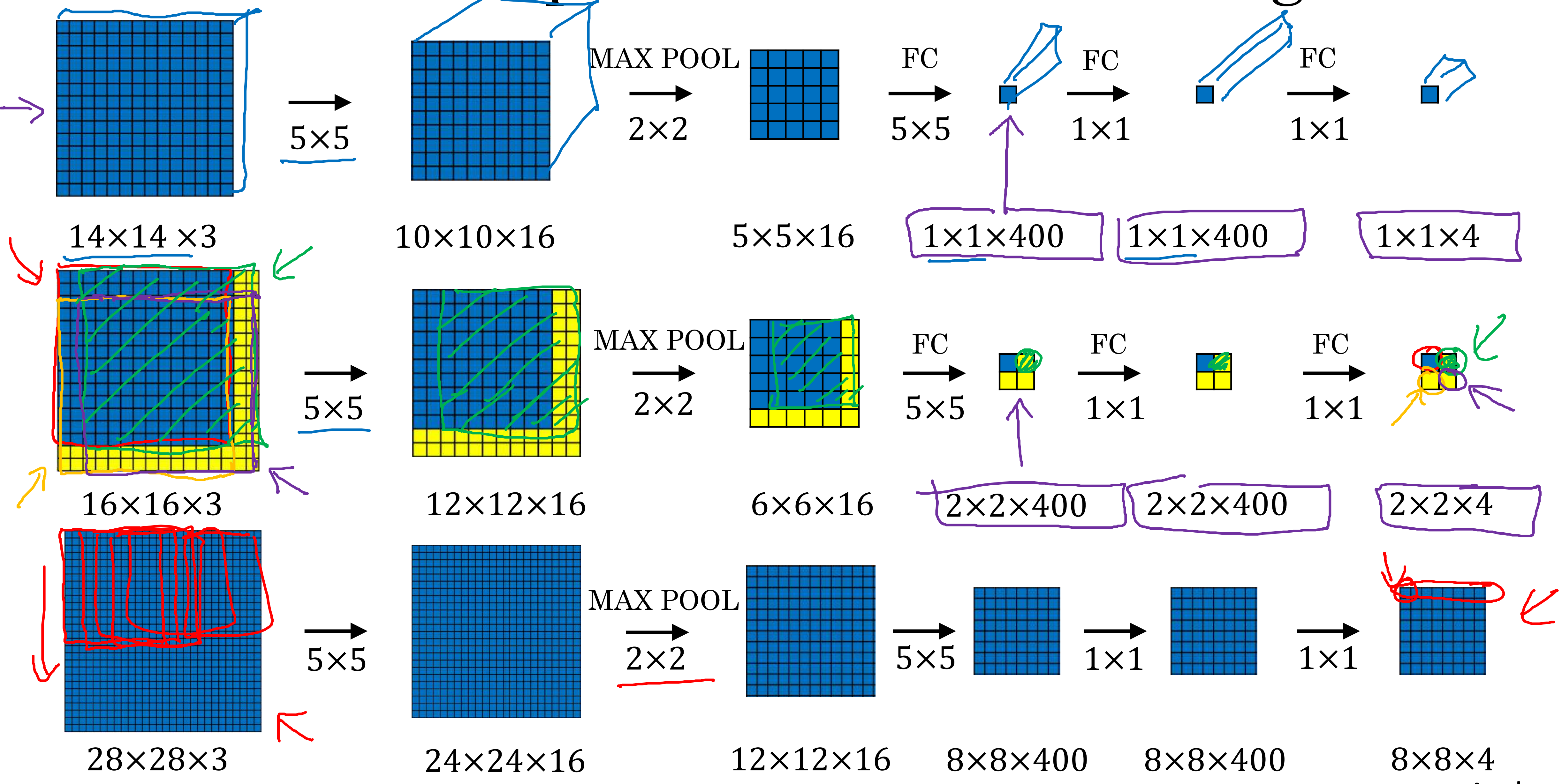
Convolutional implementation of sliding windows

Turning FC layer into convolutional layers



Andrew Ng

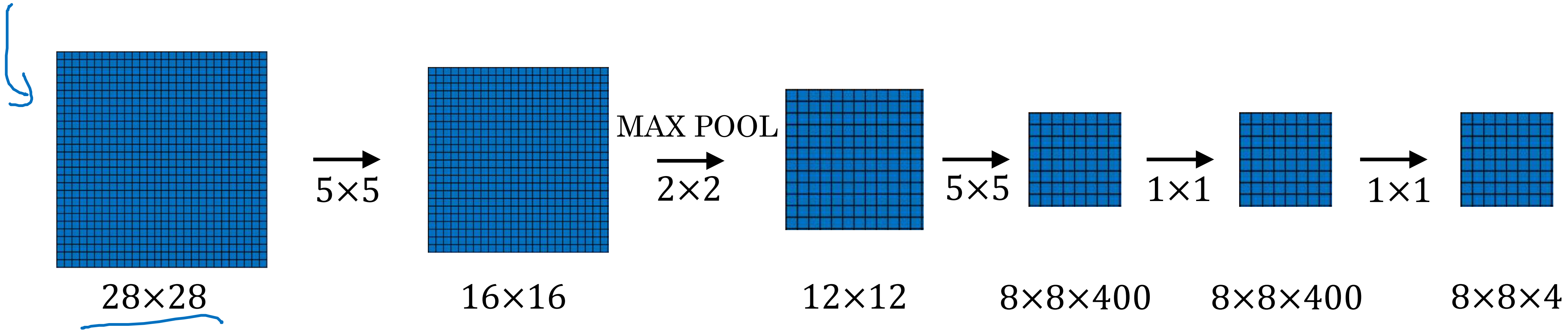
Convolution implementation of sliding windows

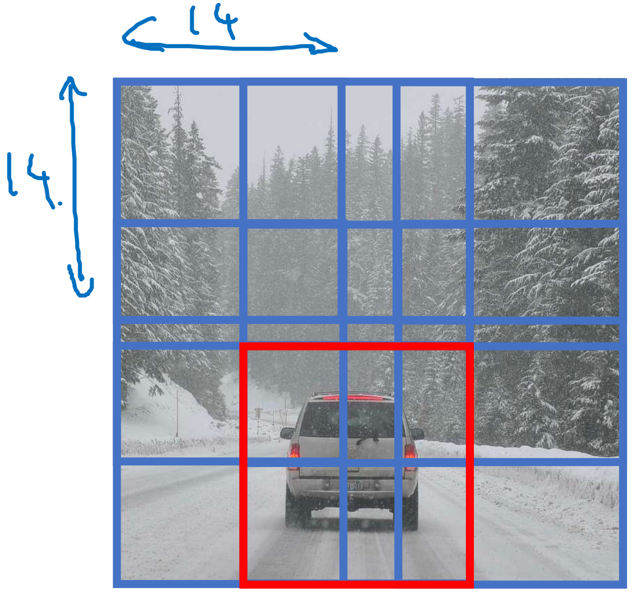


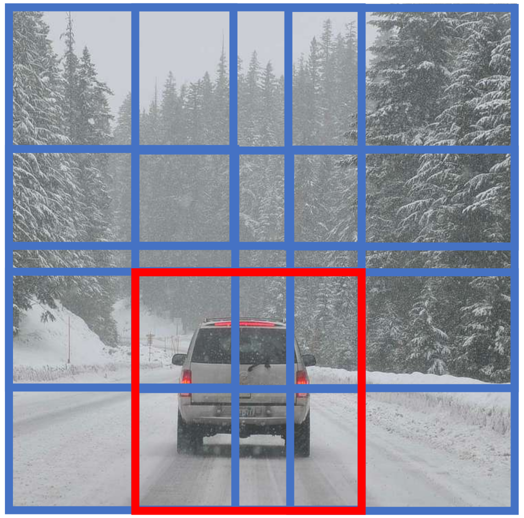
[Sermanet et al., 2014, OverFeat: Integrated recognition, localization and detection using convolutional networks]

Andrew Ng

Convolution implementation of sliding windows







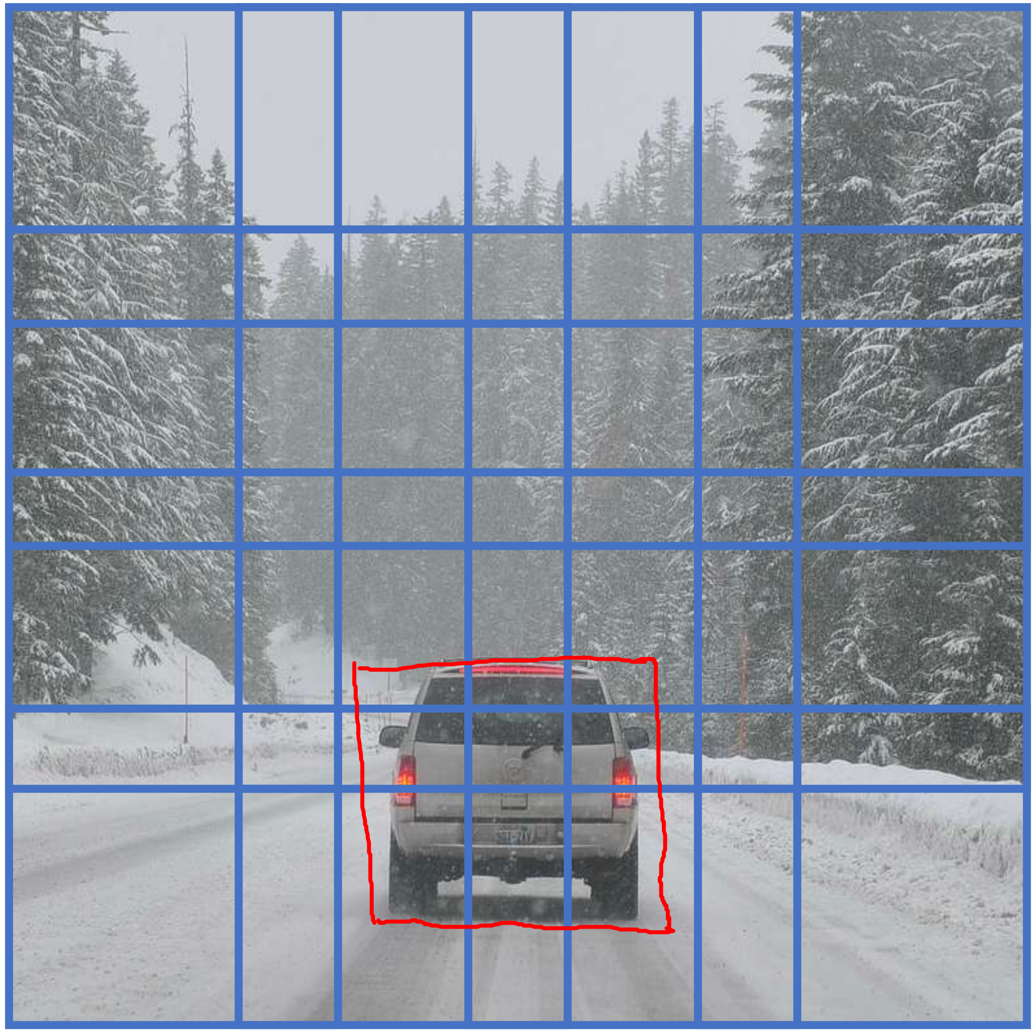
Andrew Ng

Object Detection

deeplearning.ai

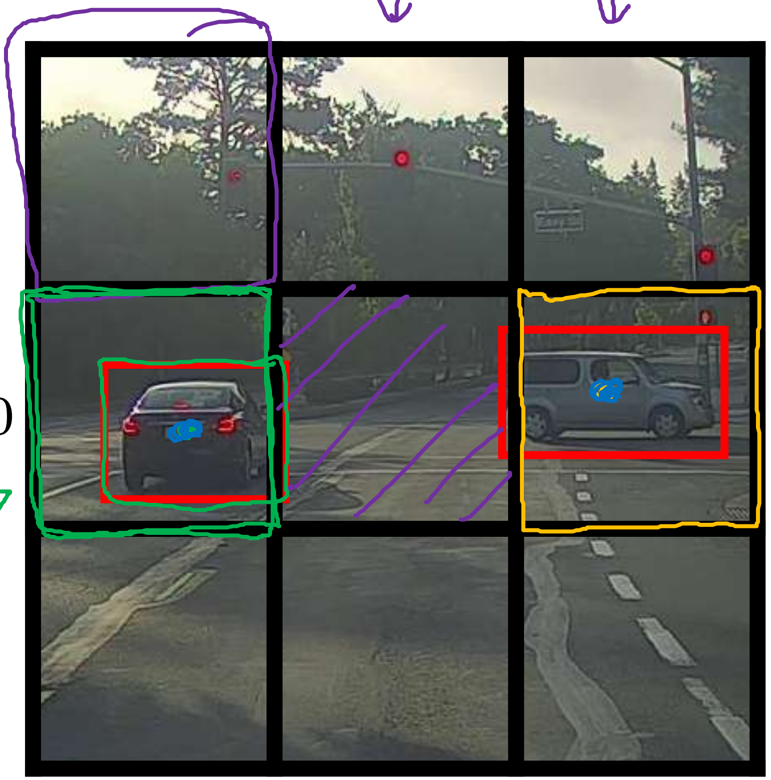
Bounding box predictions

Output accurate bounding boxes



Andrew Ng

YOLO algorithm



100

100

Labels for training

For each grid cell:

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

Target output: $3 \times 3 \times 8$

X ($100 \times 100 \times 3$) → CONV → MAXPOOL → .... → y ($3 \times 3 \times 8$)

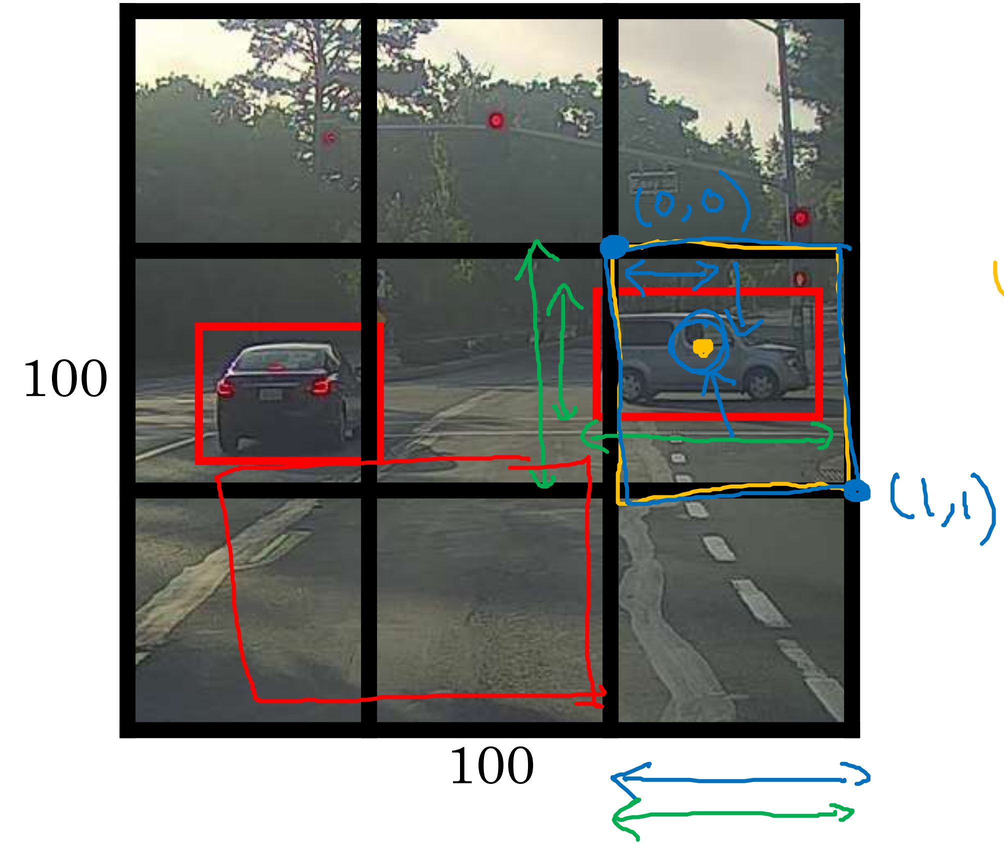
→ $19 \times 19 \times 8$

361

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

Andrew Ng

Specify the bounding boxes



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

0.4
0.3 } between 0 and 1

0.9
0.5 } could be > 1

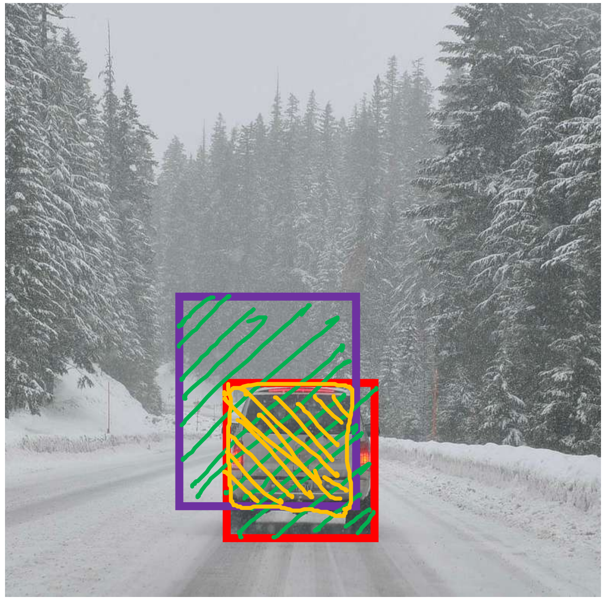
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]

deeplearning.ai

Object Detection

Intersection over union

Evaluating object localization



Intersection over Union (IoU)

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

"Correct" if IoU ≥ $\boxed{0.5}$

0.6

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

deeplearning.ai

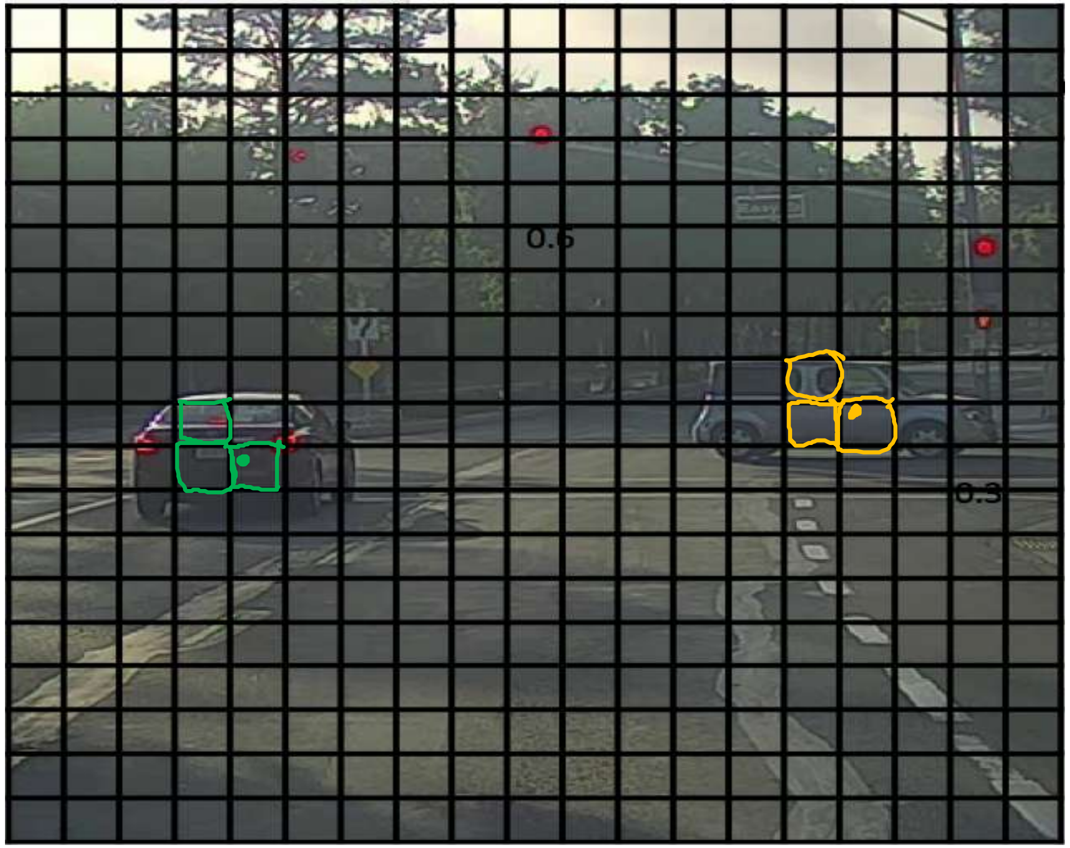
Object Detection

Non-max suppression

Non-max suppression example

Andrew Ng

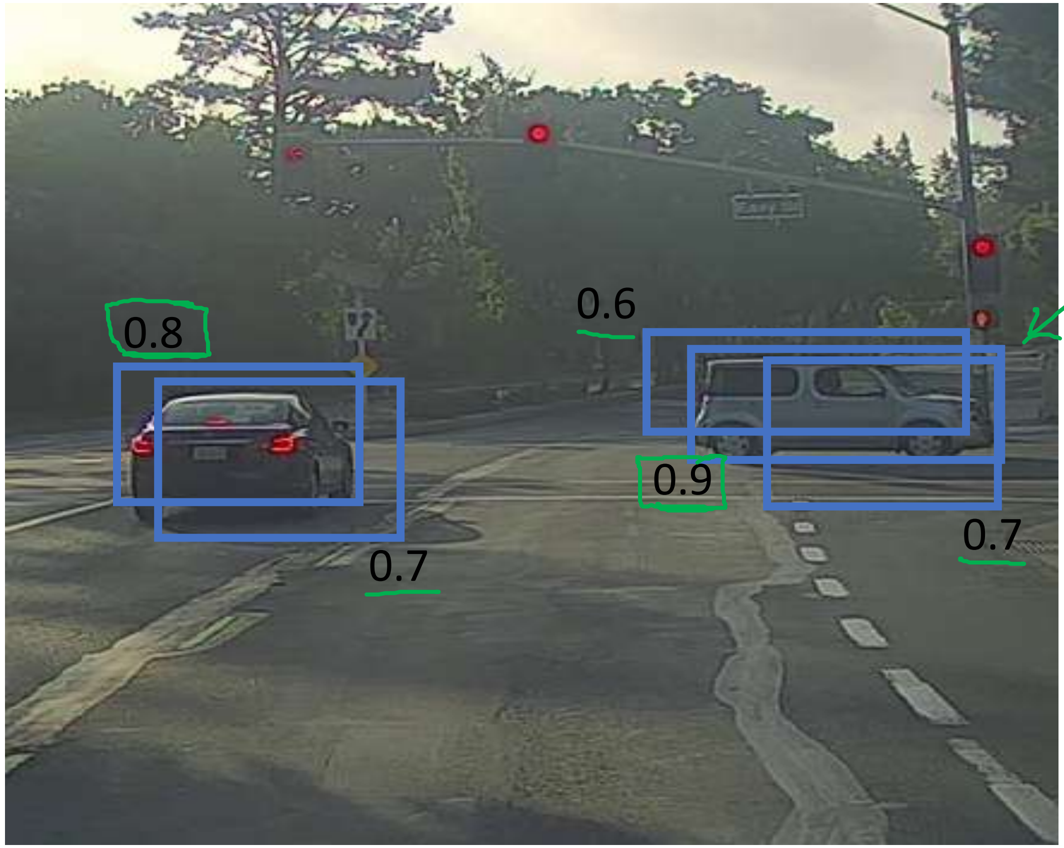
Non-max suppression example



19x19

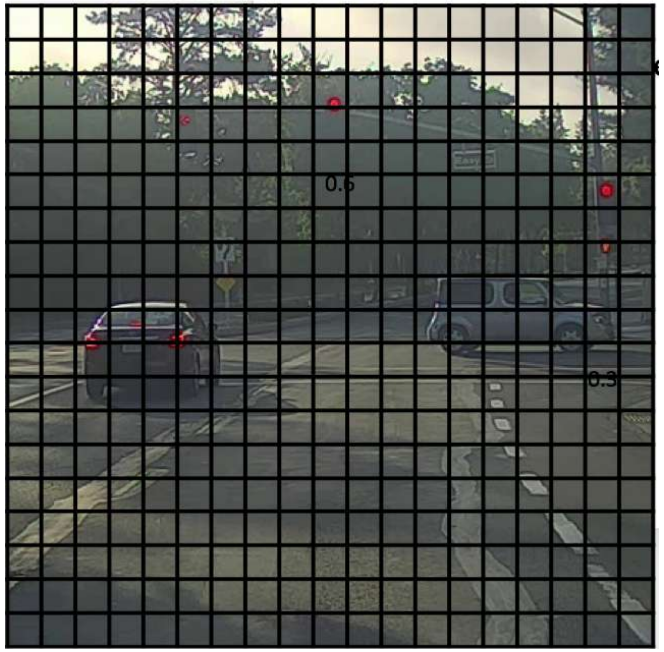
Andrew Ng

Non-max suppression example



P_c

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 ≥ 0.5 with the box output in the previous step

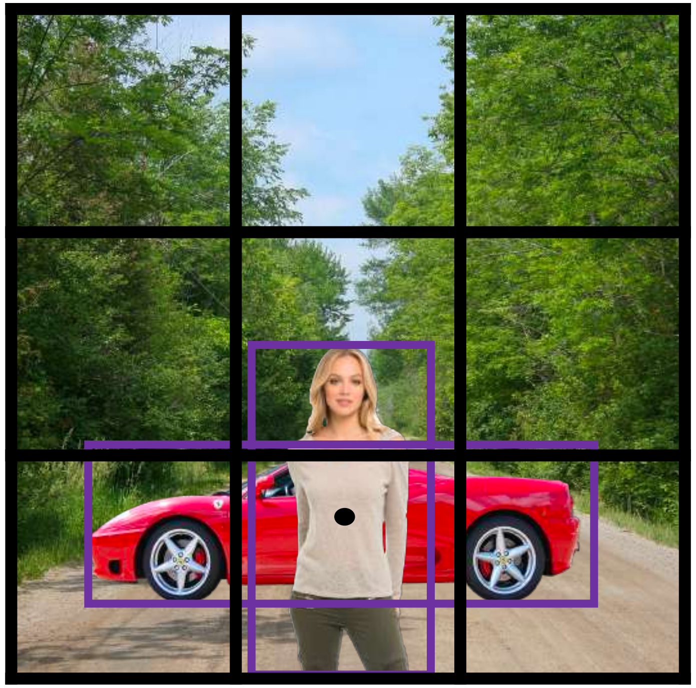
Andrew Ng

Object Detection

Anchor boxes

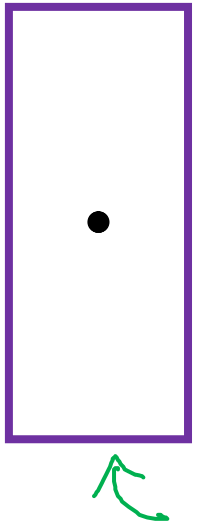
deeplearning.ai

Overlapping objects:

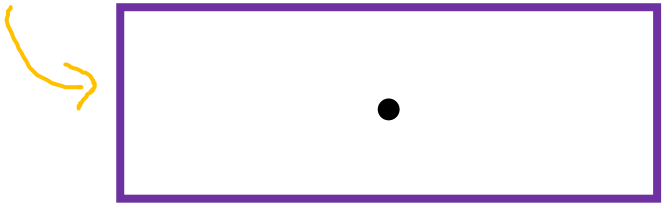


$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Anchor box 1:



Anchor box 2:



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

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

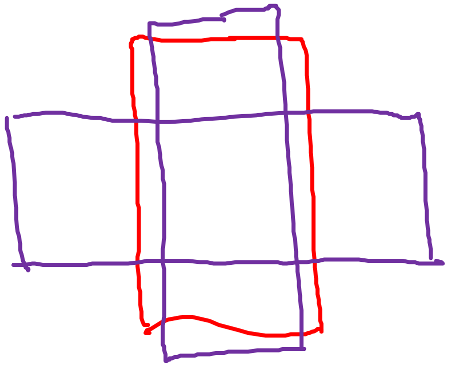
Anchor box algorithm

Previously:

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

Output y:

$3 \times 3 \times 8$



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)

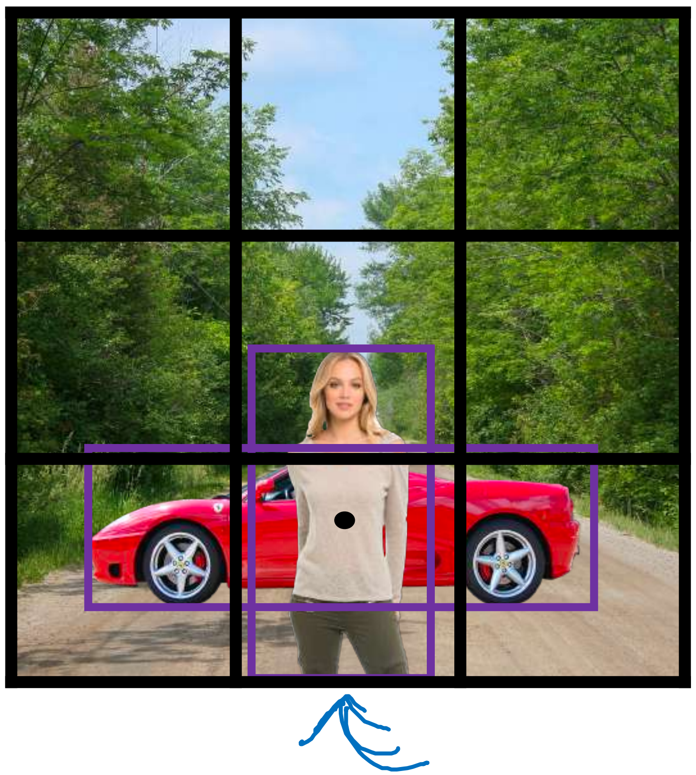
Output y:

$3 \times 3 \times 16$

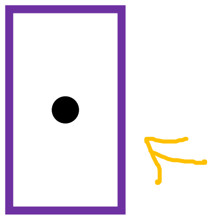
$3 \times 3 \times 2 \times 8$

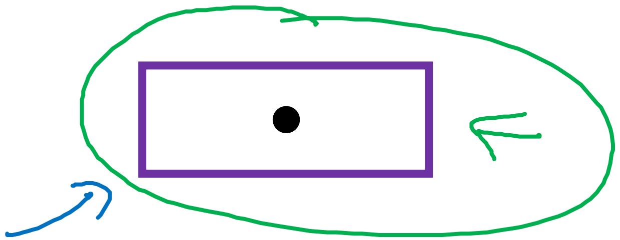
Andrew Ng

Anchor box example



Anchor box 1: Anchor box 2:





car only?

$$
y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}
\qquad
\begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 1 \\ 0 \\ 0 \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}
\qquad
\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}
\begin{array}{l} \left.\vphantom{\begin{matrix}0\\?\\?\\?\\?\\?\\?\\?\end{matrix}}\right\} \text{anchor box 1} \\ \left.\vphantom{\begin{matrix}1\\b\\b\\b\\b\\0\\1\\0\end{matrix}}\right\} \text{anchor box 2} \end{array}
$$

Andrew Ng

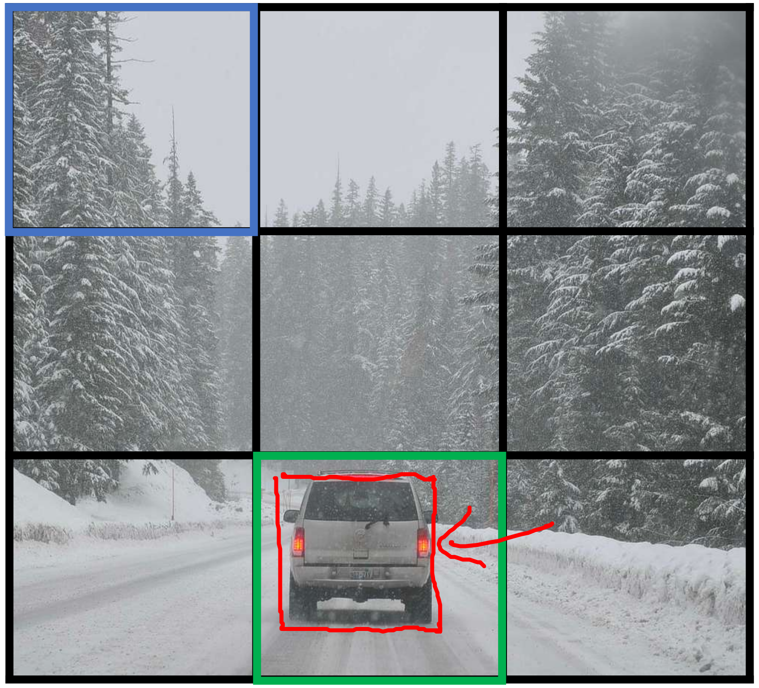
deeplearning.ai

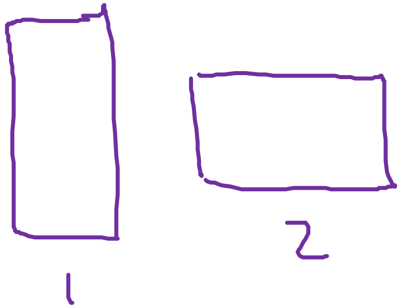
Object Detection

Putting it together: YOLO algorithm

Training

1 - pedestrian
2 - car
3 - motorcycle





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

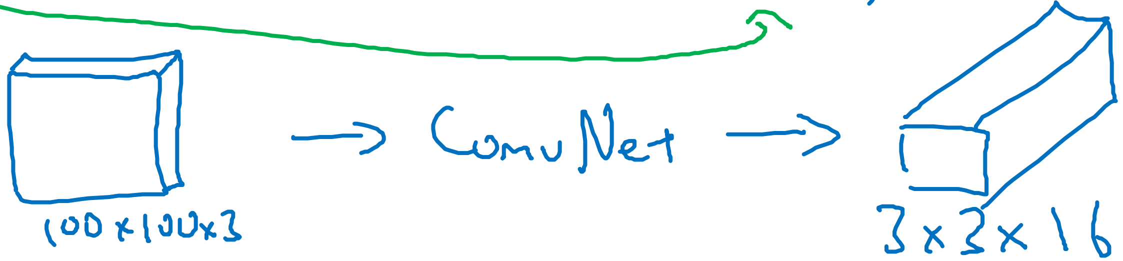
3 × 3 × 16

y is 3×3×2×8

19×19×16
19×19×40

#anchors

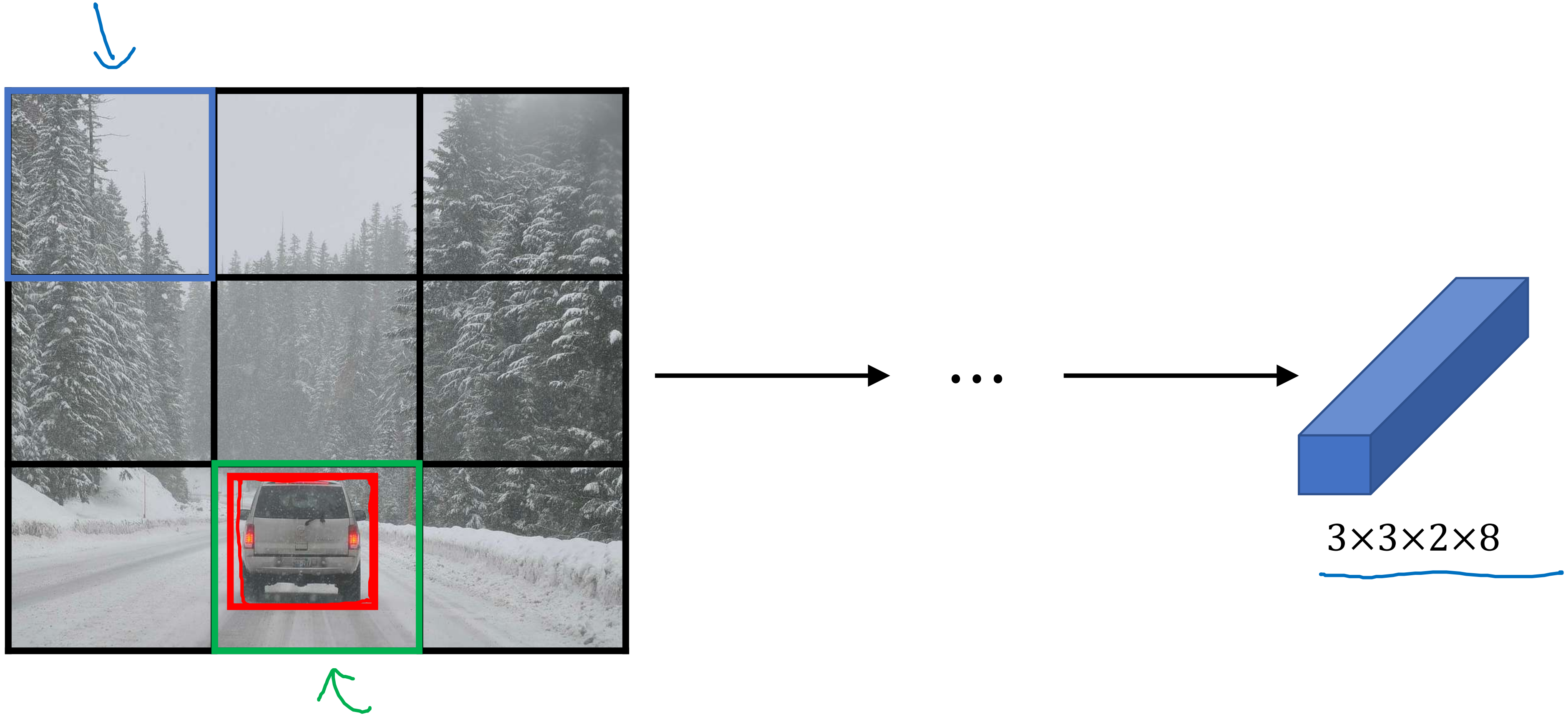
5 + #classes



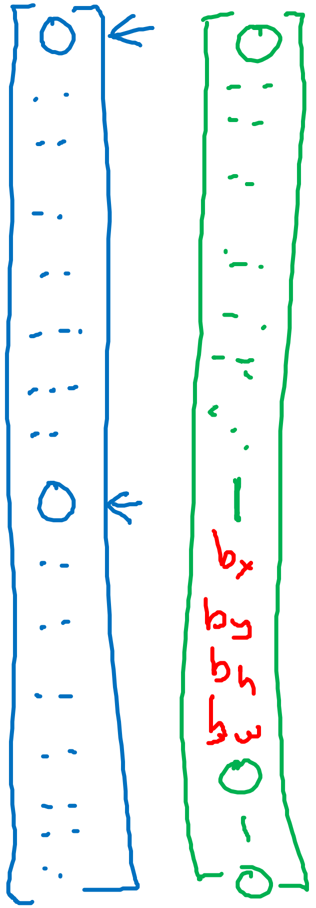
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]

Andrew Ng

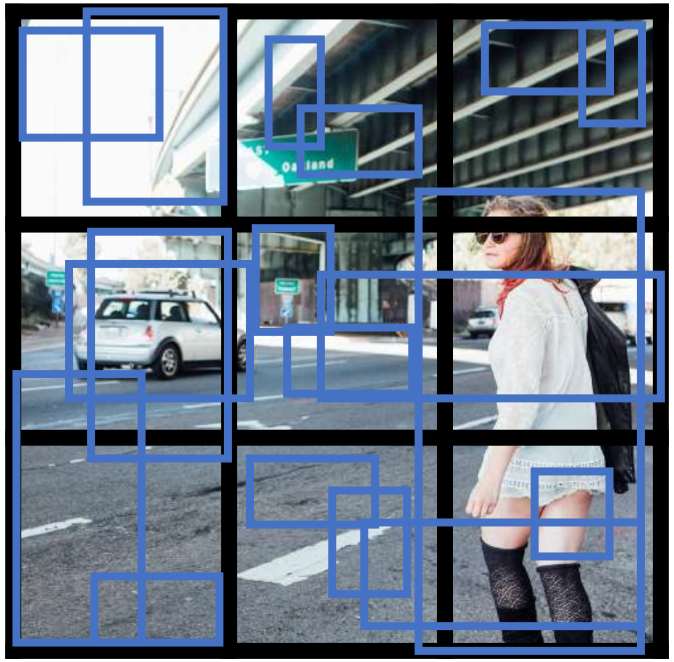
Making predictions



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



Outputting the non-max supressed outputs



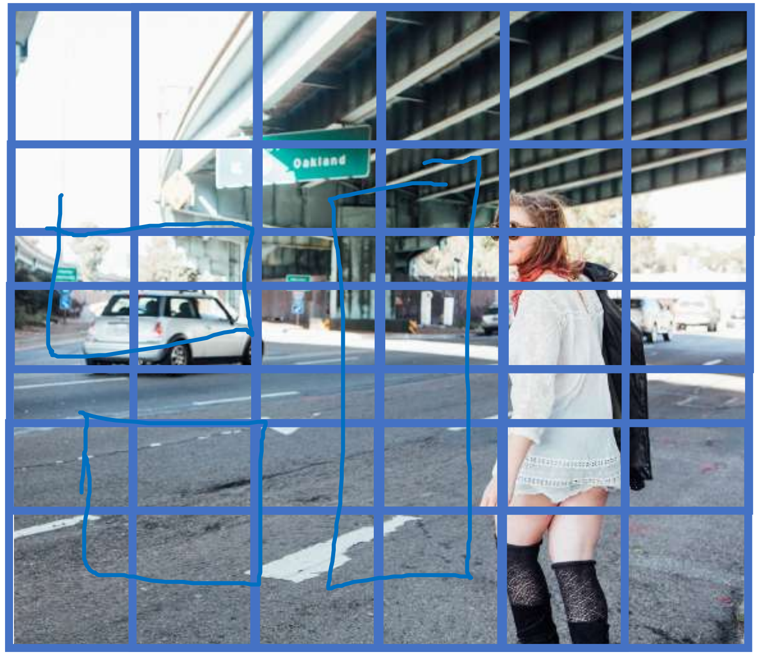
- For each grid call, get 2 predicted bounding boxes.
- Get rid of low probability predictions.
- For each class (pedestrian, car, motorcycle) use non-max suppression to generate final predictions.

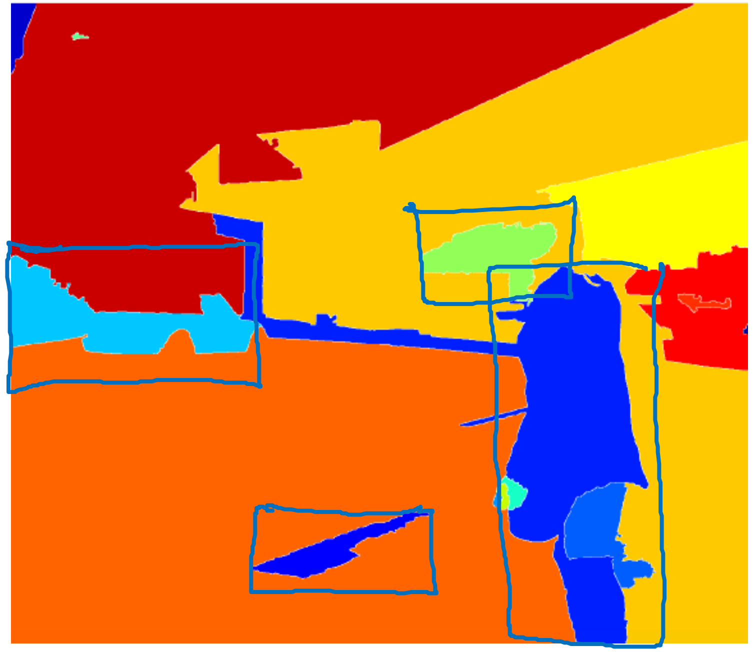
deeplearning.ai

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] Andrew Ng

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]

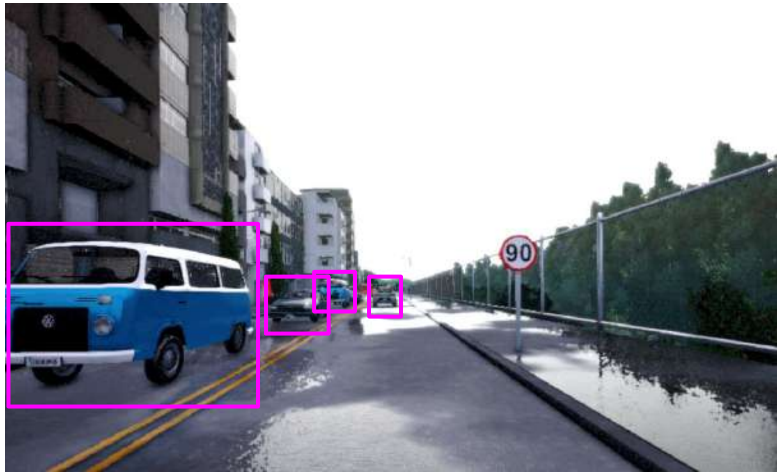
deeplearning.ai

Convolutional Neural Networks

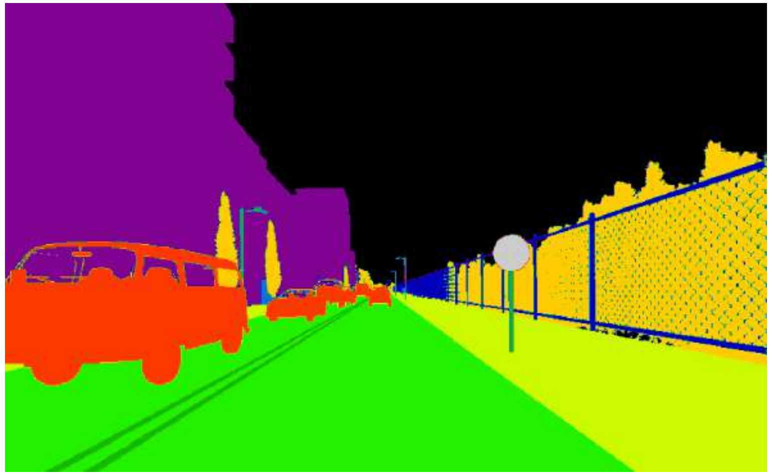
Semantic segmentation with U-Net

Object Detection vs. Semantic Segmentation

Input image

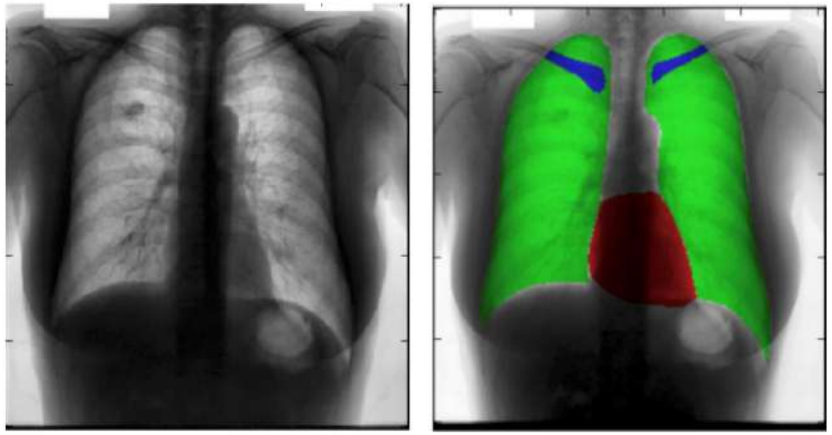


Object Detection

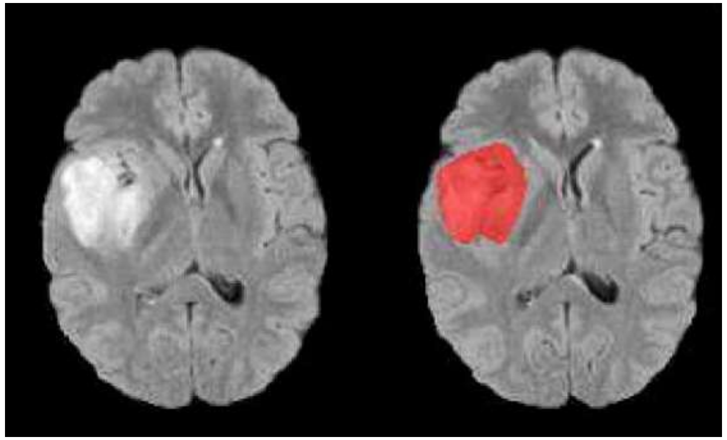


Semantic Segmentation

Motivation for U-Net



Chest X-Ray



Brain MRI

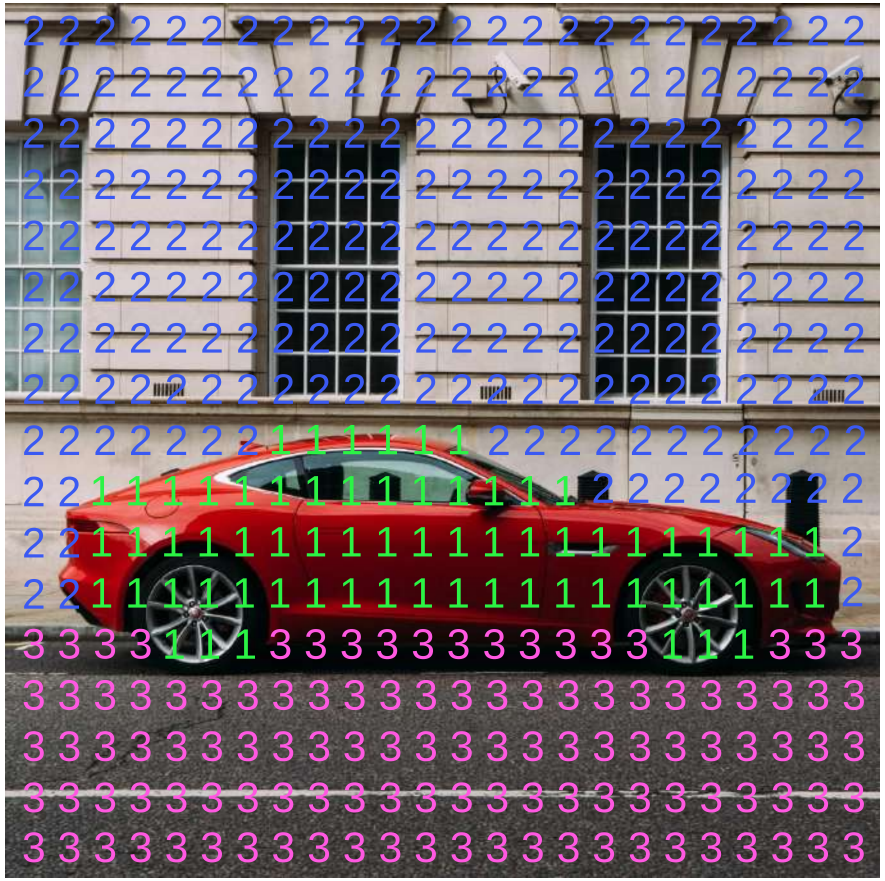
[Novikov et al., 2017, Fully Convolutional Architectures for Multi-Class Segmentation in Chest Radiographs]
[Dong et al., 2017, Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks]

Per-pixel class labels

1. Car
0. Not Car

Andrew Ng

Per-pixel class labels



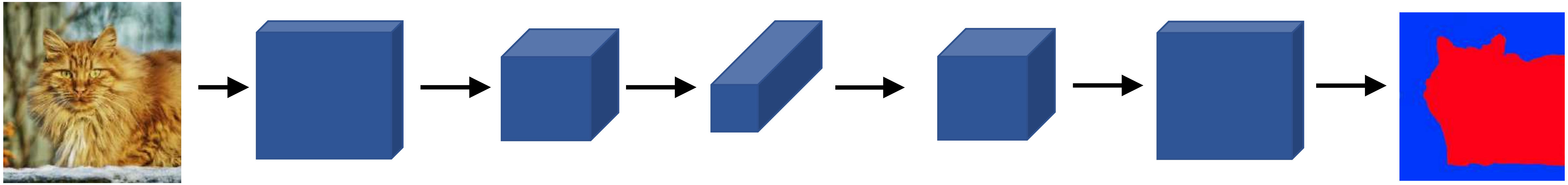
1. Car
2. Building
3. Road

2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
3	3	3	3	1	1	1	3	3	3	3	3	3	3	3	3	3	3	1	1	1	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Segmentation Map

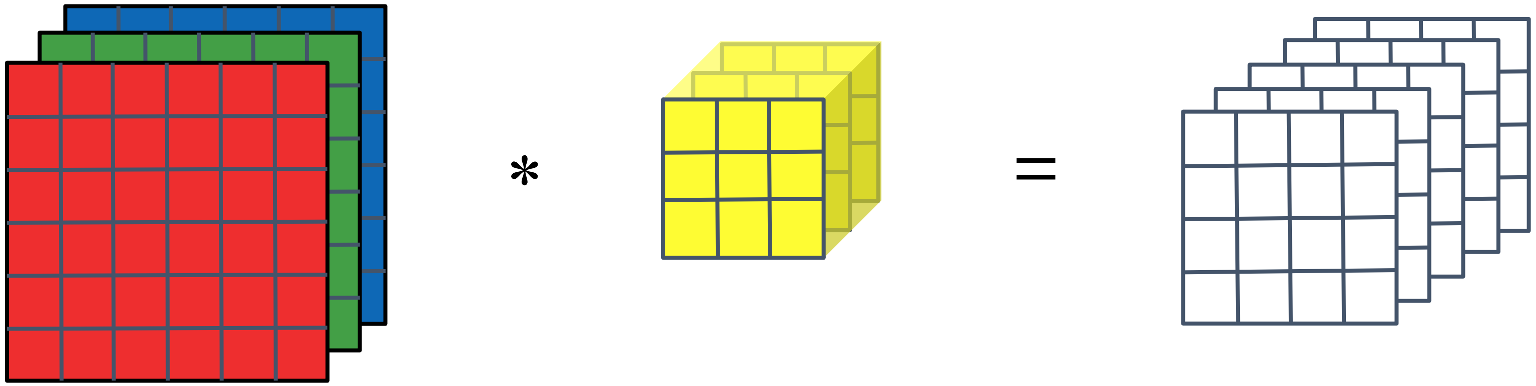
Andrew Ng

Deep Learning for Semantic Segmentation

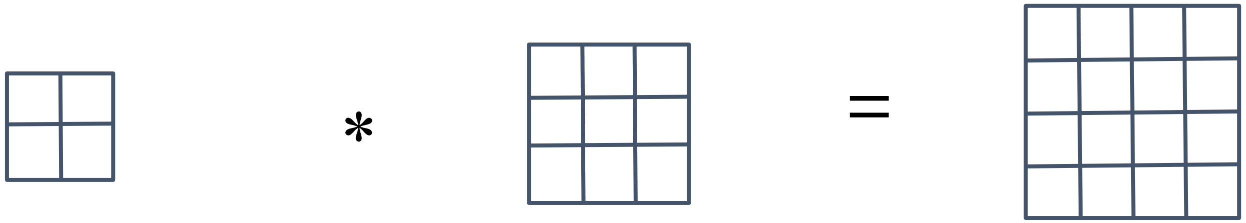


Transpose Convolution

Normal Convolution

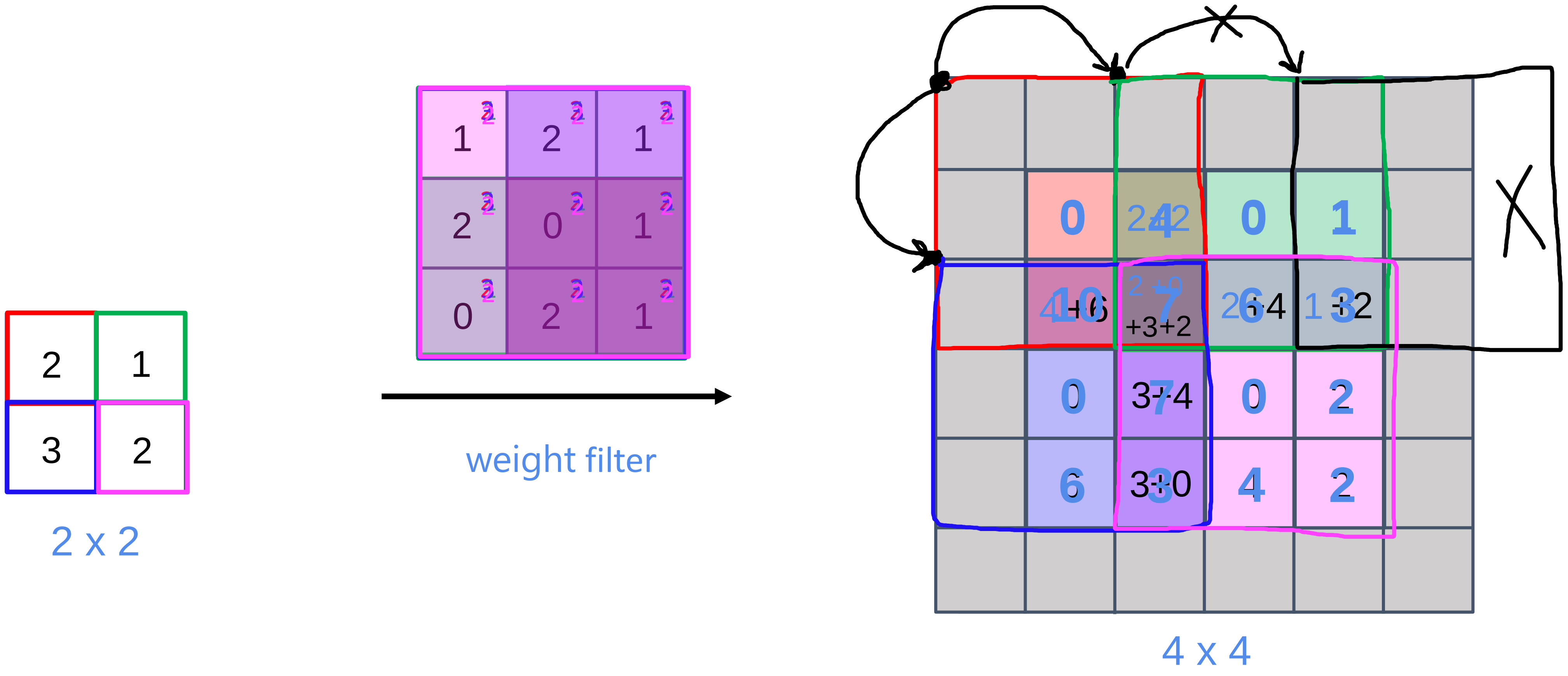


Transpose Convolution



Andrew Ng

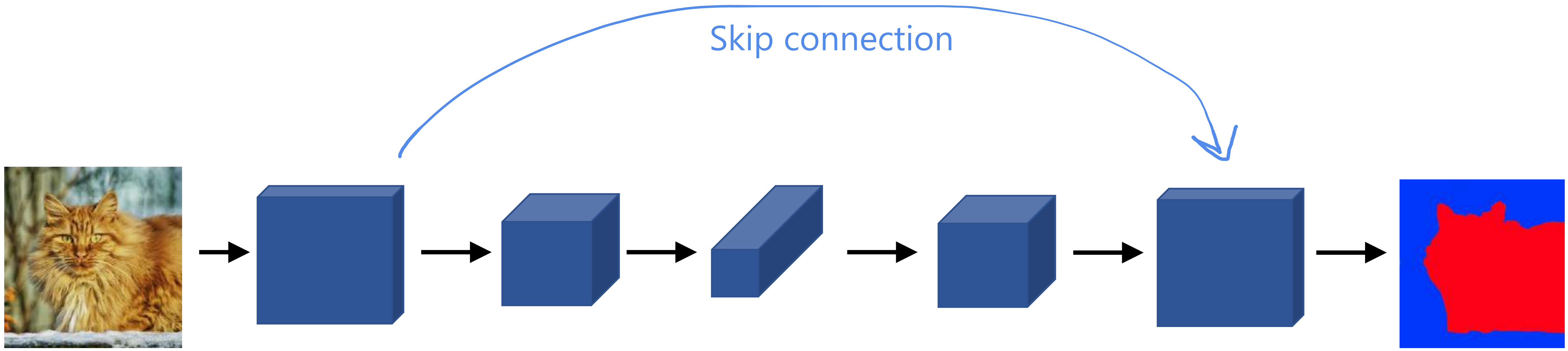
Transpose Convolution



filter f x f = 3 x 3 padding p = 1 stride s = 2

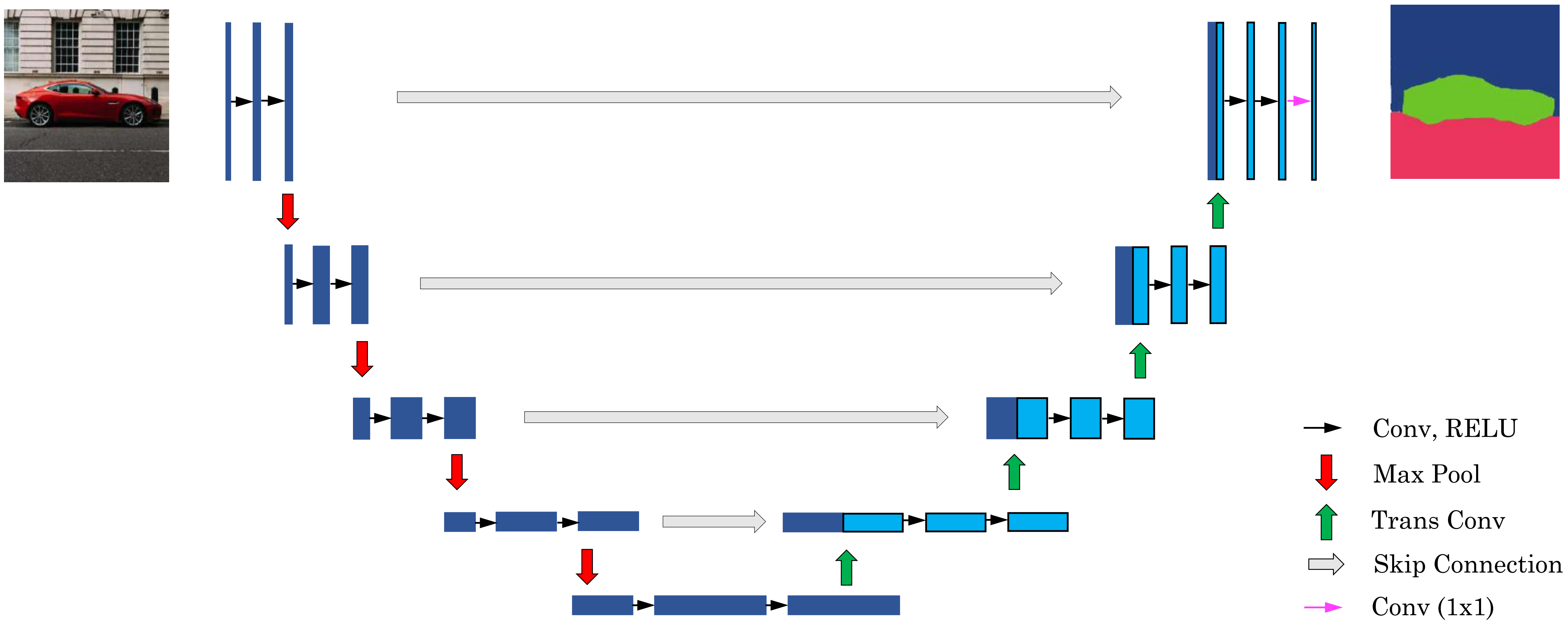
Andrew Ng

Deep Learning for Semantic Segmentation



Andrew Ng

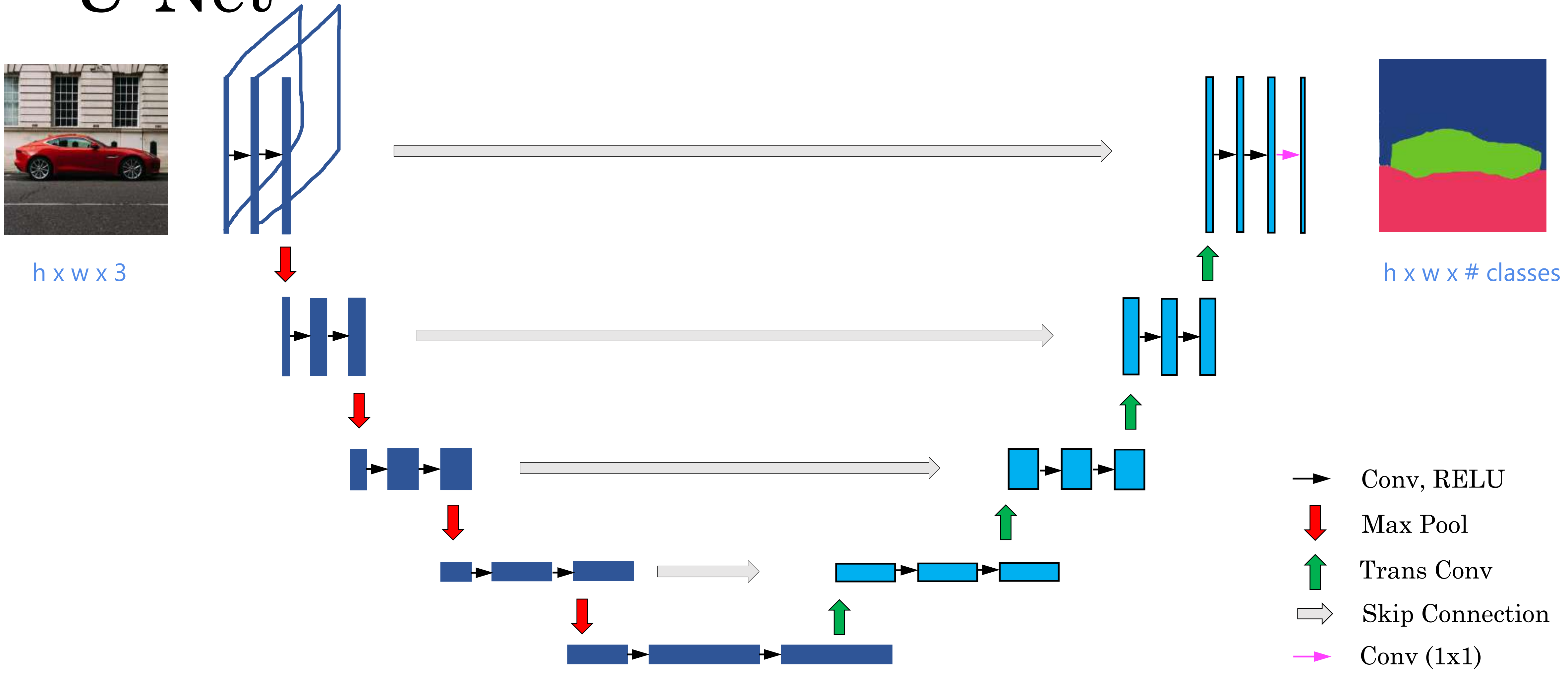
U-Net



[Ronneberger et al., 2015, U-Net: Convolutional Networks for Biomedical Image Segmentation]

Andrew Ng

U-Net



[Ronneberger et al., 2015, U-Net: Convolutional Networks for Biomedical Image Segmentation]

Andrew Ng