Object Detection

Tomasz Stańczyk
tomasz.stanczyk@inria.fr

With slides from Andrew Ng and other sources (referenced)

Inria
3iA Côte d'Azur
Interdisciplinary Institute for Artificial Intelligence

Today's Agenda

Object detection fundamentals - based on DeepLearningAI materials by Andrew Ng

+ references for more information/self-study if desired

YOLOX object detection algorithm

+ references for more information/self-study if desired

Later today: YOLOX-based practical assignment

DeepLearning.AI

LearnOpenCV.com

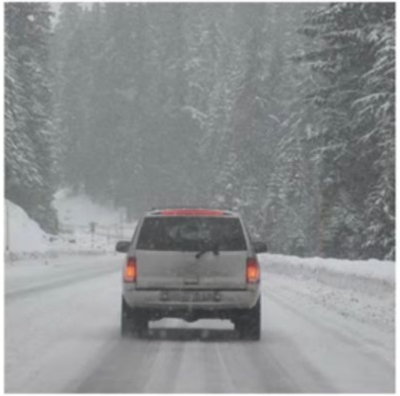
EXPERT AI BLOG

Object Detection Fundamentals

Selected slides by Andrew Ng

What are localization and detection?

Image classification



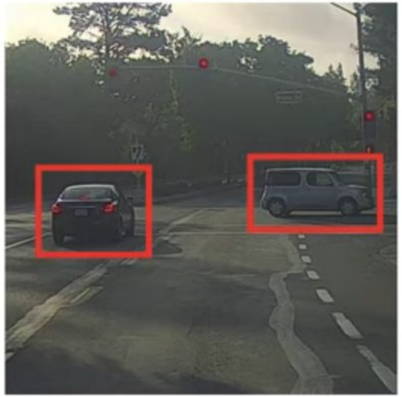
"Car"

Classification with localization

"Car"

1 object

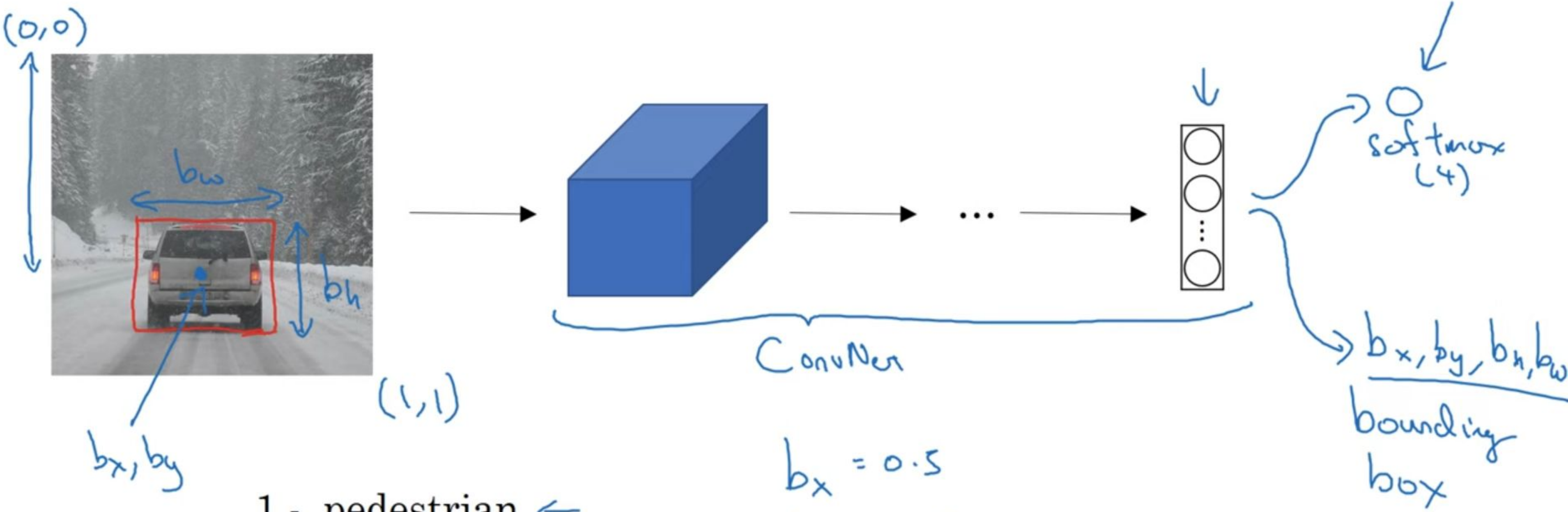
Detection



multiple objects

Classification with localization

Object Localization



1 - pedestrian

2 - car

3 - motorcycle

4 - background

$b_x = 0.5$

$b_y = 0.7$

$b_h = 0.3$

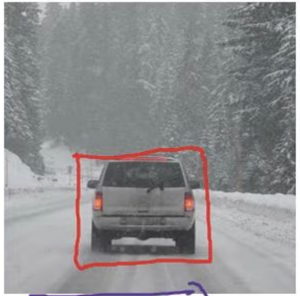
$b_w = 0.4$

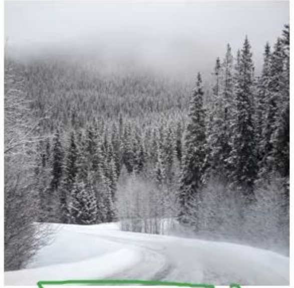
Defining the target label y

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

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

x =





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

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

p_c: is there any object?

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

p_c

"don't care"

(x, y)

Car detection example

Object Detection

Training set:

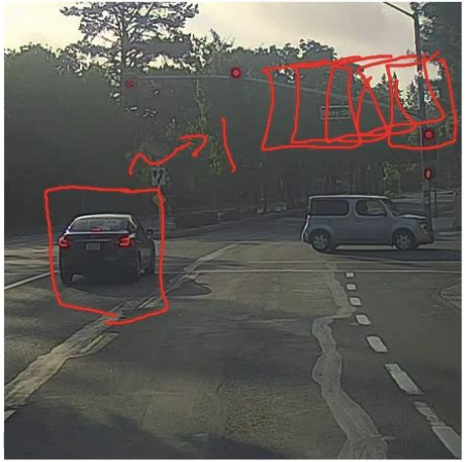
x	y
	1
	1
	1
	0
	0

 → ConvNet → y

Sliding windows detection

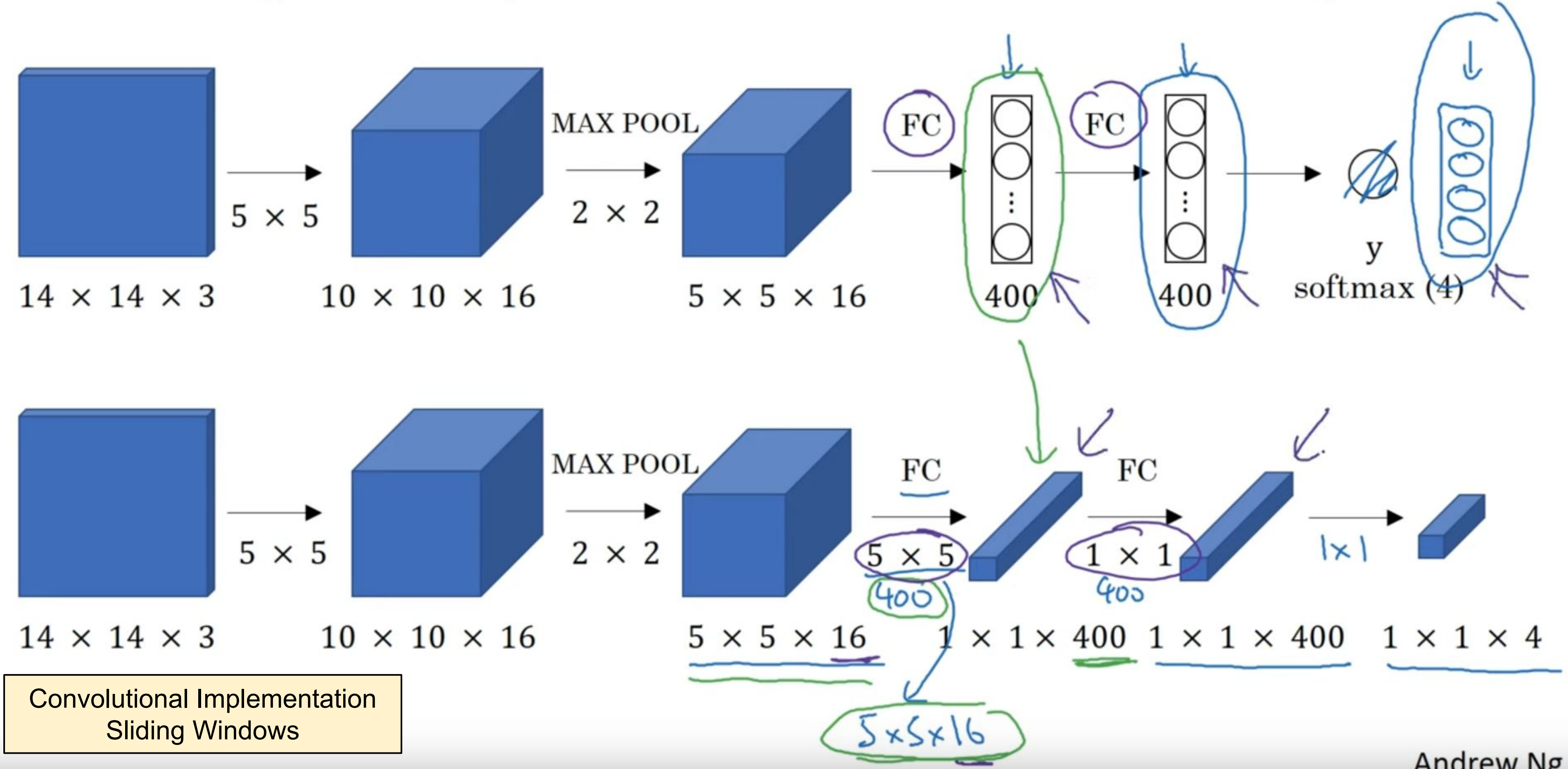
→ ConvNet → 0

→ ConvNet



Computation cost

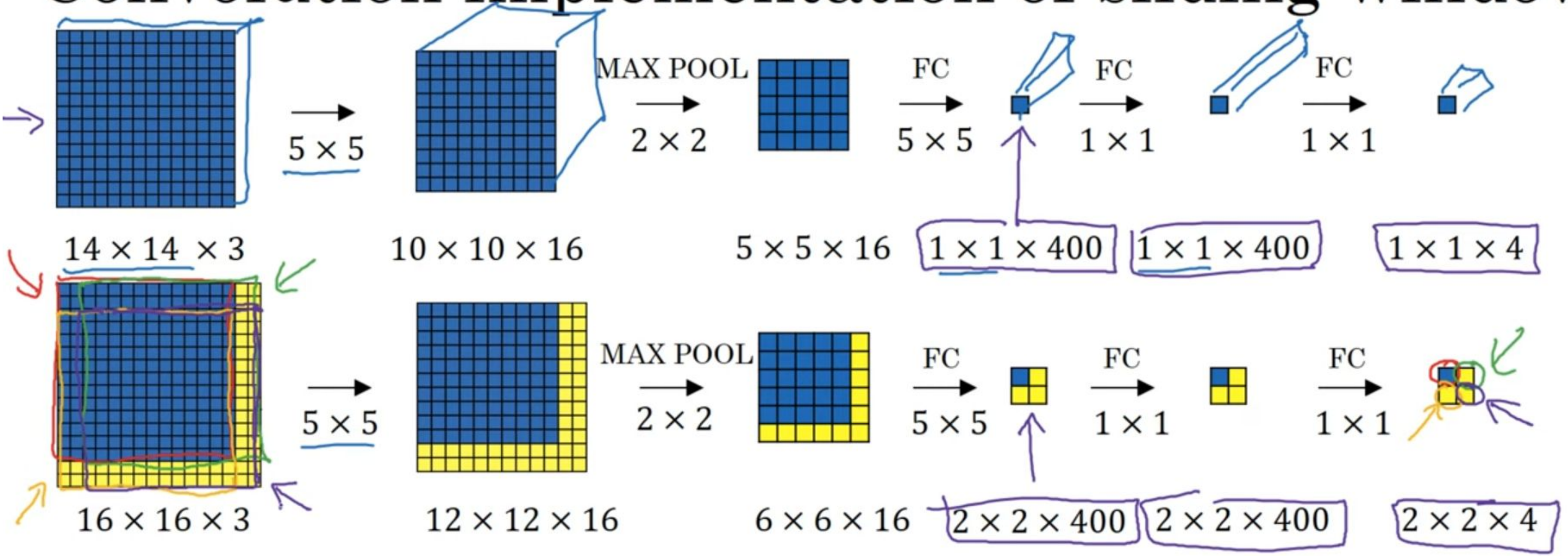
Turning FC layer into convolutional layers



Convolutional Implementation
Sliding Windows

Andrew Ng

Convolution implementation of sliding windows

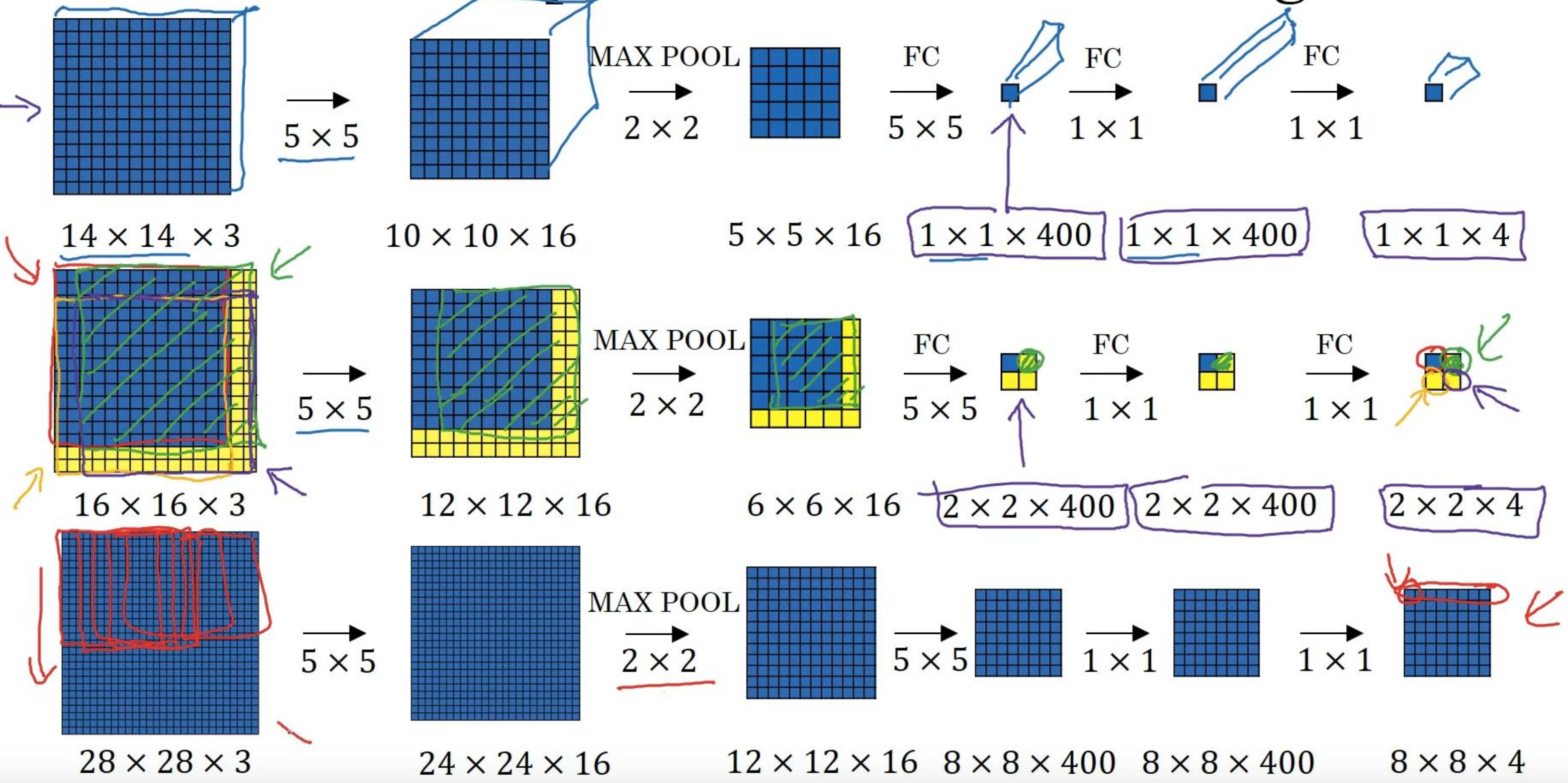


Convolutional Implementation
Sliding Windows

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

Andrew Ng

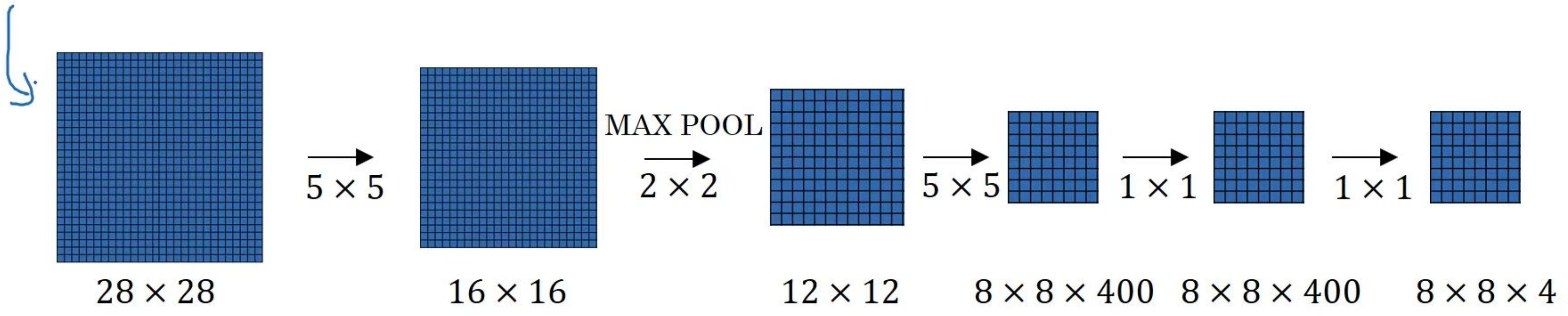
Convolution implementation of sliding windows

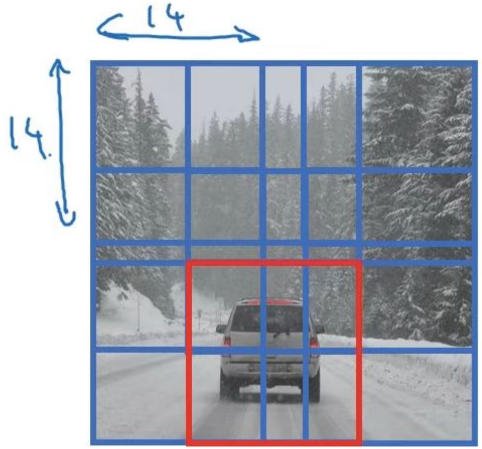


[Sermanet et al., 2014, OverFeat: Integrated recognition, localization and detection using convolutional networks]

Andrew Ng

Convolution implementation of sliding windows



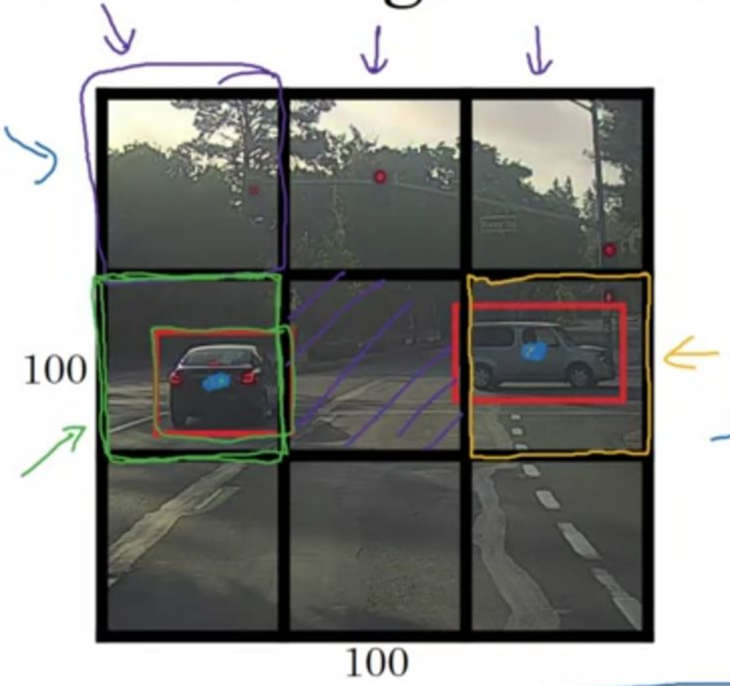


Convolutional Implementation
Sliding Windows

Andrew Ng

YOLO algorithm

Bounding Box Predictions



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:

$3 \times 3 \times 8$

$\underline{x}$ → CONV → MAXPOOL → .... → y

$100 \times 100 \times 3$ $\qquad$ $3 \times 3 \times 8$

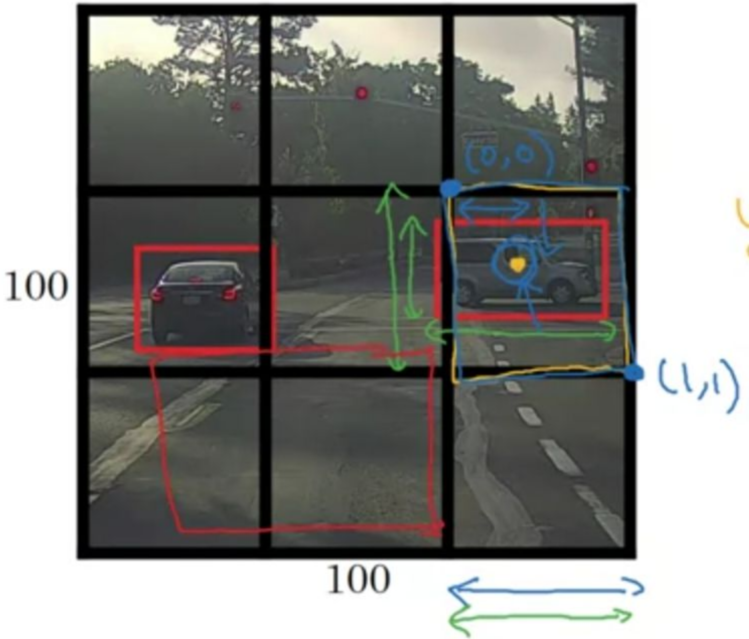
→ $19 \times 19 \times 8$

361

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

Andrew Ng

Specify the bounding boxes



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

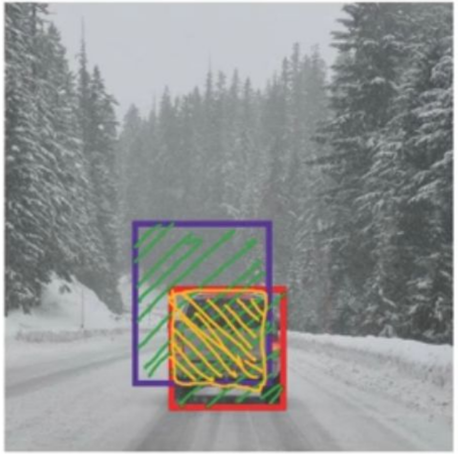
$\left.\begin{matrix} 0.4 \\ 0.3 \end{matrix}\right\}$ between 0 and 1

$\left.\begin{matrix} 0.9 \\ 0.5 \end{matrix}\right\}$ could be > 1

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

Evaluating object localization

Intersection
Over Union



Intersection over Union (IoU)

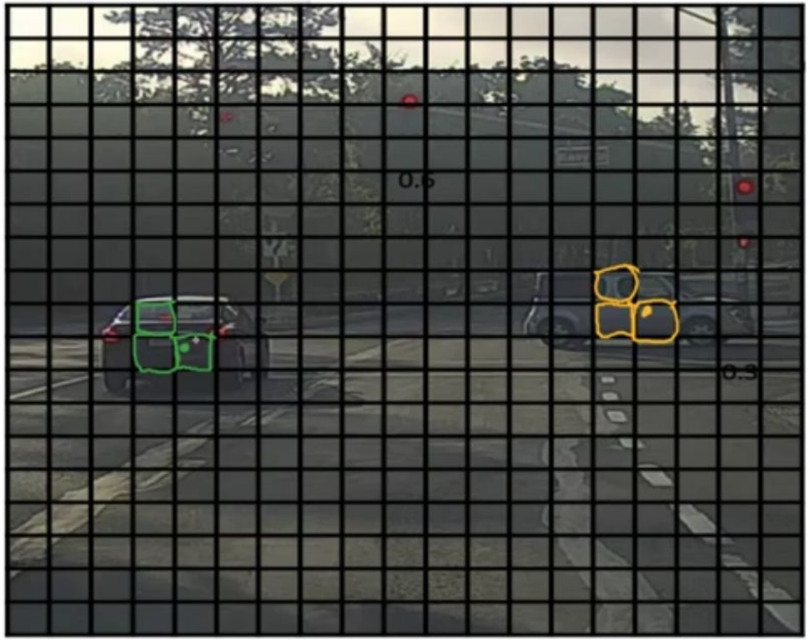
$$= \frac{\text{size of}}{\text{size of}}$$

"Correct" if IoU ≥ $\boxed{0.5}$

0.6

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

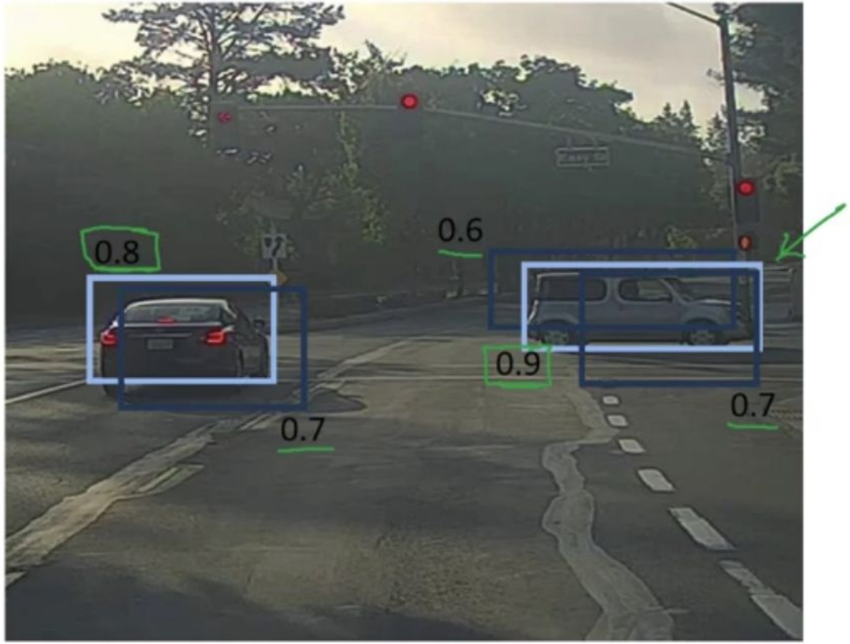
Non-max suppression example



19x19

Non-max suppression example

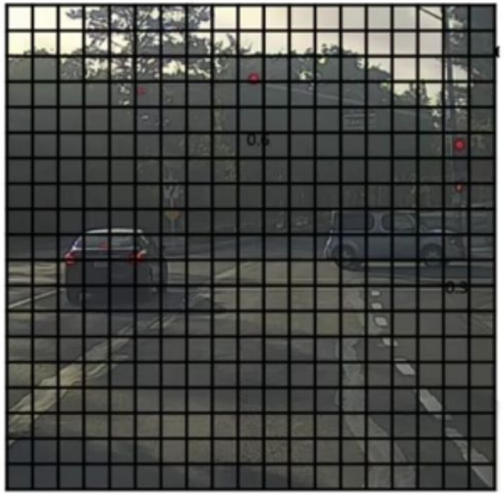
Non-max Suppression



p_c

Non-max suppression algorithm

Non-max Suppression



19× 19

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

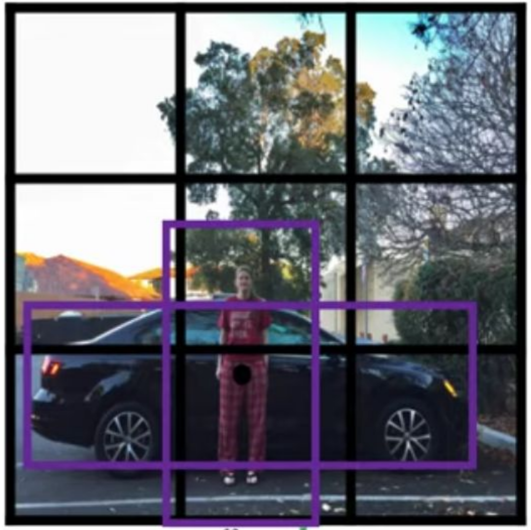
Discard all boxes with $p_c \leq 0.6$

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

Overlapping objects:

Anchor Boxes



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

Anchor box 1:

Anchor box 2:

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

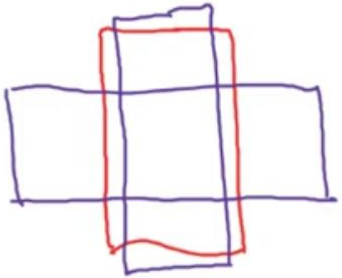
Anchor box algorithm

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.

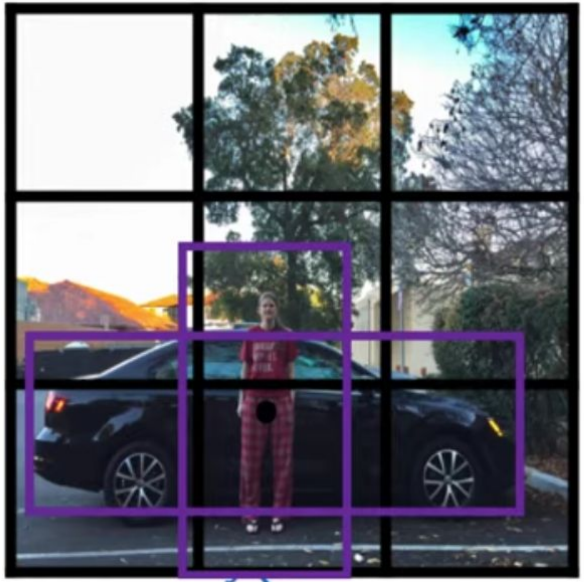
(grid cell, anchor box)

Output y:

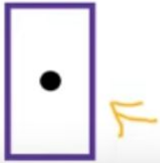
$3 \times 3 \times 16$

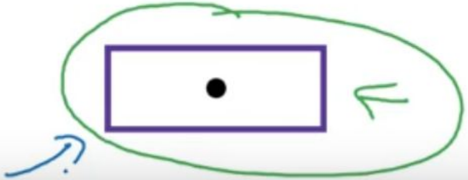
$3 \times 3 \times 2 \times 8$

Anchor box example



Anchor box 1: Anchor box 2:



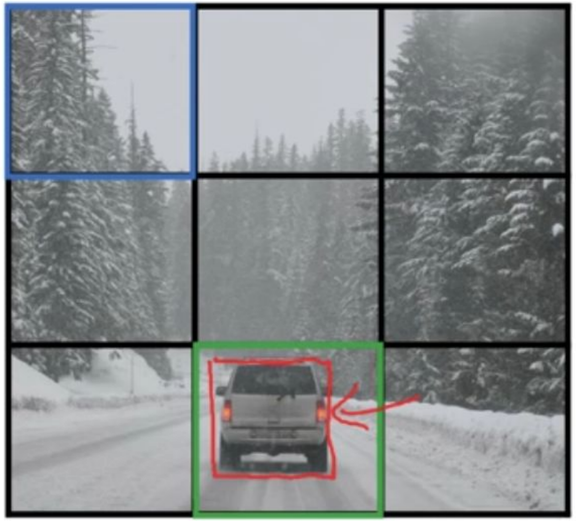


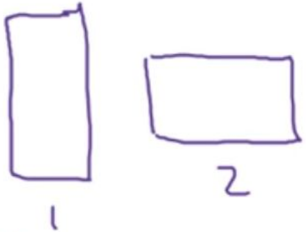
$$
y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}
\quad
\begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 1 \\ 0 \\ 0 \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}
\quad
\overset{\text{car only?}}{\begin{bmatrix}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

Training

YOLO Algorithm

1 - pedestrian
2 - car
3 - motorcycle





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

3 × 3 × 16

y is 3 × 3 × 2 × 8

19×19×16
19×19×40

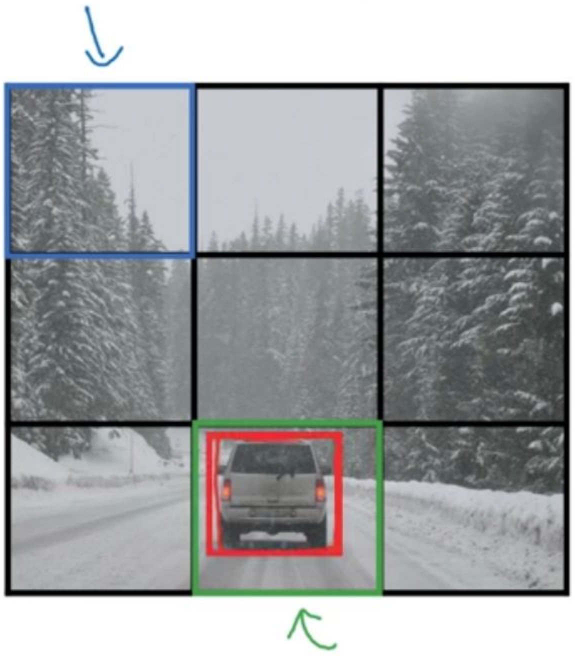
#anchors

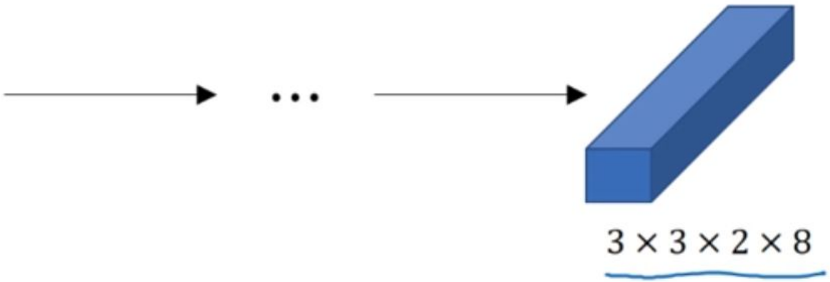
5 + #classes

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

Making predictions

YOLO Algorithm



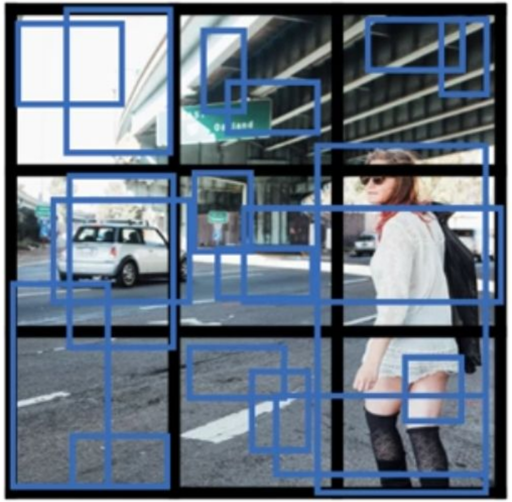


$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}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

Andrew Ng

Outputting the non-max supressed outputs

YOLO Algorithm



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

Outputting the non-max supressed outputs

YOLO Algorithm

- 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.

For more details…

Check the Andrew Ng's videos on object detection

Available on YouTube

See the following playlist: https://www.youtube.com/playlist?list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF

Videos: C4W3L**01**, C4W3L**03**, C4W3L**04**, C4W3L**06**, C4W3L**07**, C4W3L**08**, C4W3L**09**

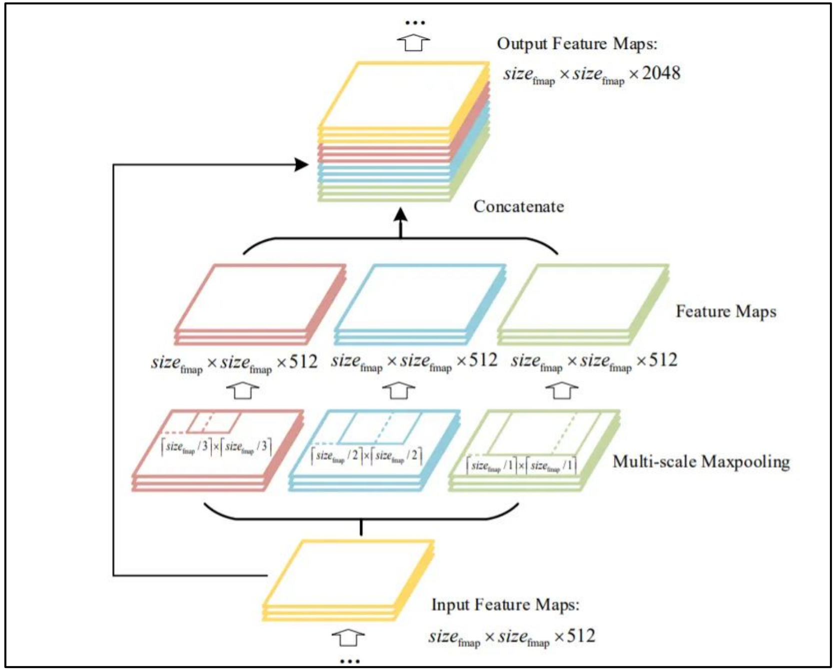
YOLOX
Object Detection Algorithm

Some information and figures
Courtesy of LearnOpenCV

YOLOX

Built on top of YOLOv3 with Darknet-53 backbone and SPP layer (YOLOv3-SPP)

Spatial Pyramid Pooling (SPP) layer →



Huang Z. et al. *DC-SPP-YOLO: Dense Connection and Spatial Pyramid Pooling Based YOLO for Object Detection.* 2019
https://arxiv.org/ftp/arxiv/papers/1903/1903.08589.pdf

YOLOX

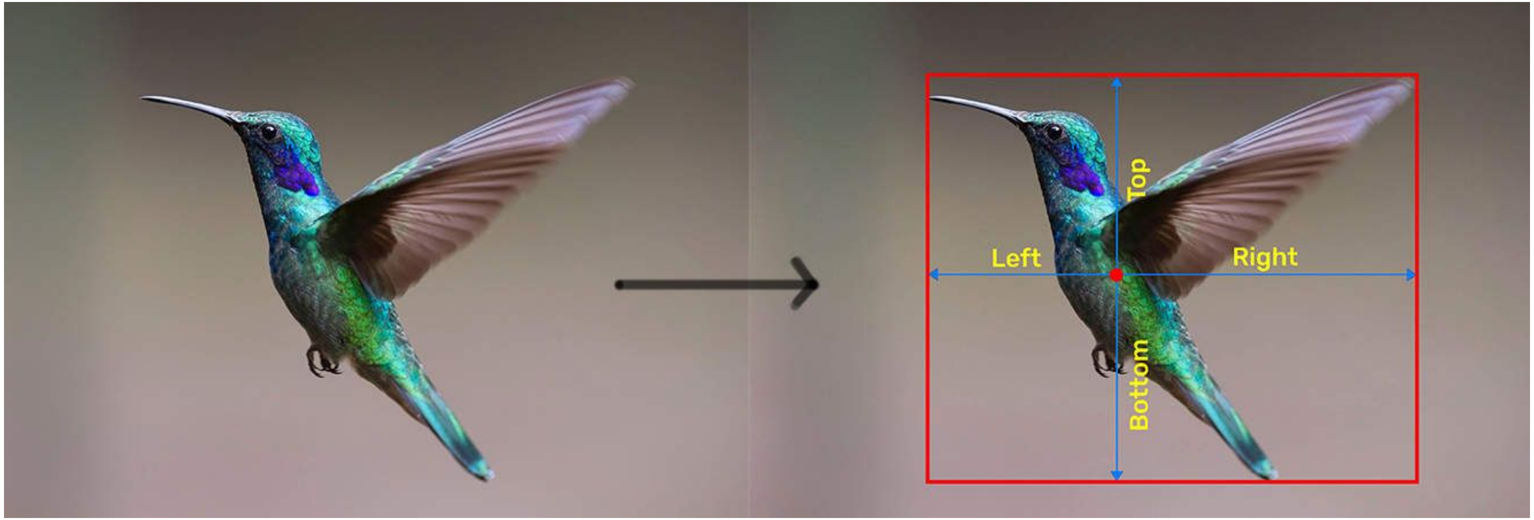
YOLOX' distinctive features:

- Anchor free design
- Decoupled head
- simOTA label assignment strategy
- Advanced Augmentations: Mixup and Mosaic

Center based detector

Find positive point in the center

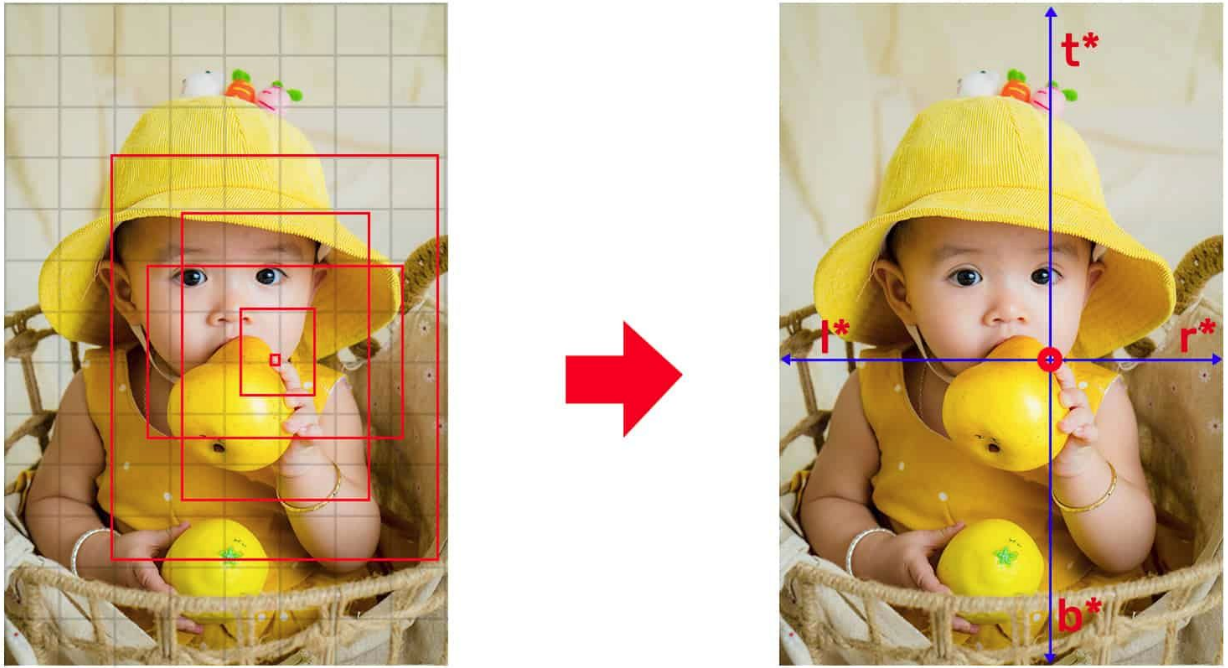
Predict four distances from the positive to the boundary



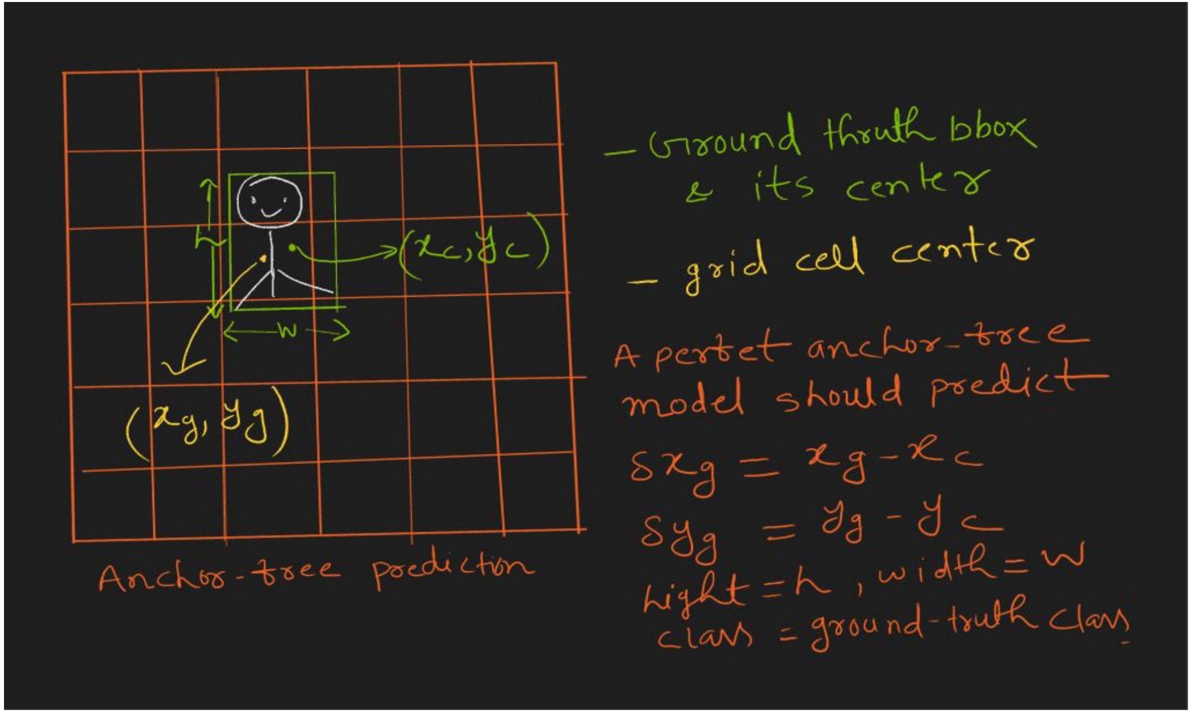
Anchor Free YOLOX

Anchor based

Anchor free



What is Anchor Free Object Detection

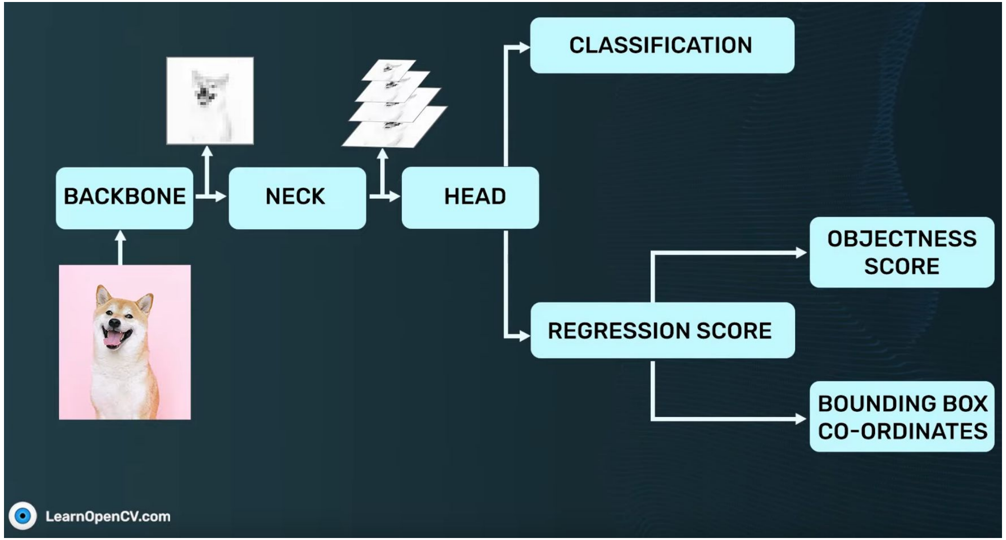


YOLO architecture

Backbone: extracts features of an image

Neck: producing feature maps with multiple scales

Head: outputs localization and classification scores



YOLOX - decoupled head

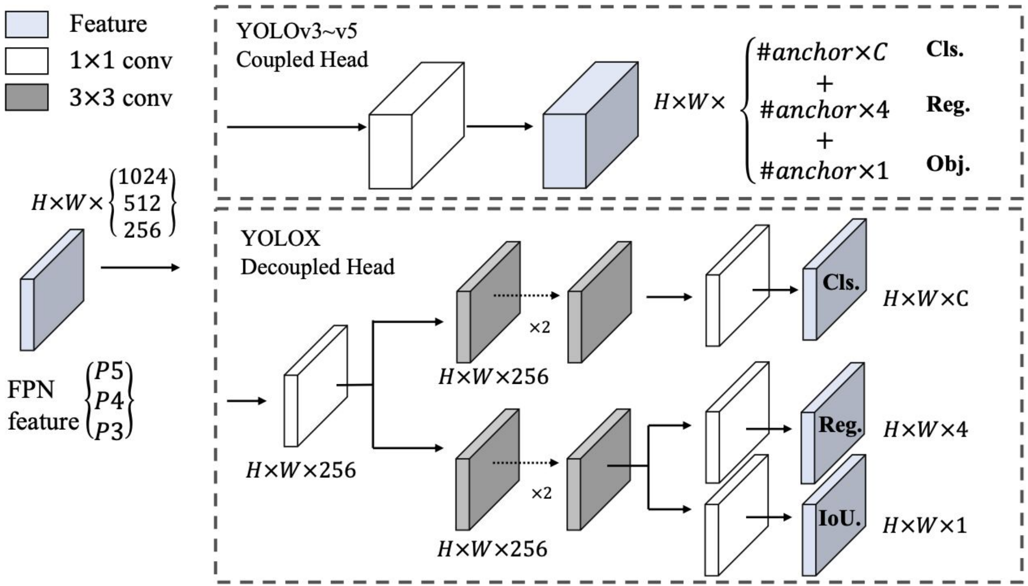
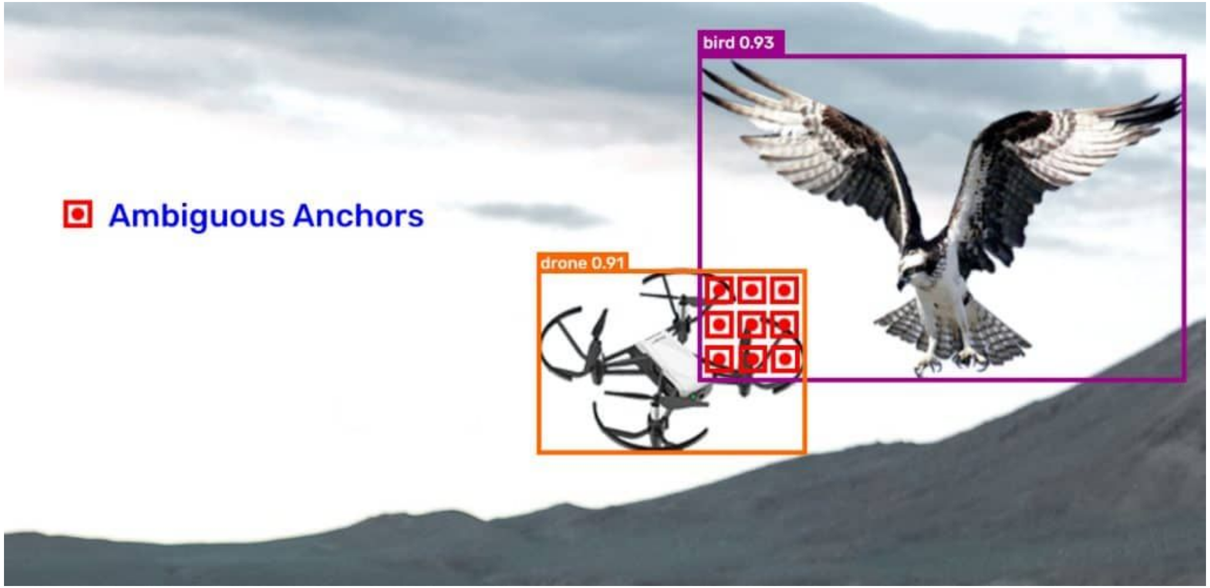


Figure 2: Illustration of the difference between YOLOv3 head and the proposed decoupled head. For each level of FPN feature, we first adopt a 1×1 conv layer to reduce the feature channel to 256 and then add two parallel branches with two 3×3 conv layers each for classification and regression tasks respectively. IoU branch is added on the regression branch.

SimOTA Advanced Label Assignment Strategy

OTA: Optimal Transport Assignment for Object Detection



SimOTA Advanced Label Assignment Strategy

Briefly explained by LearnOpenCV

What is simOTA in YOLOX?

Simplified OTA or simOTA is the redesigned Optimal Transport Assignment strategy. The training cost does not increase but average precision(AP) is definitely improved. It is shown with empirical evidence in the paper.

In simOTA, iteration is not performed for every positive label. A strategy called **Dynamic Top K** is used to estimate the approximate number of positive anchors for each ground truth. Here, only the top **K** number of positive labels are selected. This reduces the number of iterations by many folds.

The number of positive labels per ground truth (GT) varies due to the following factors.

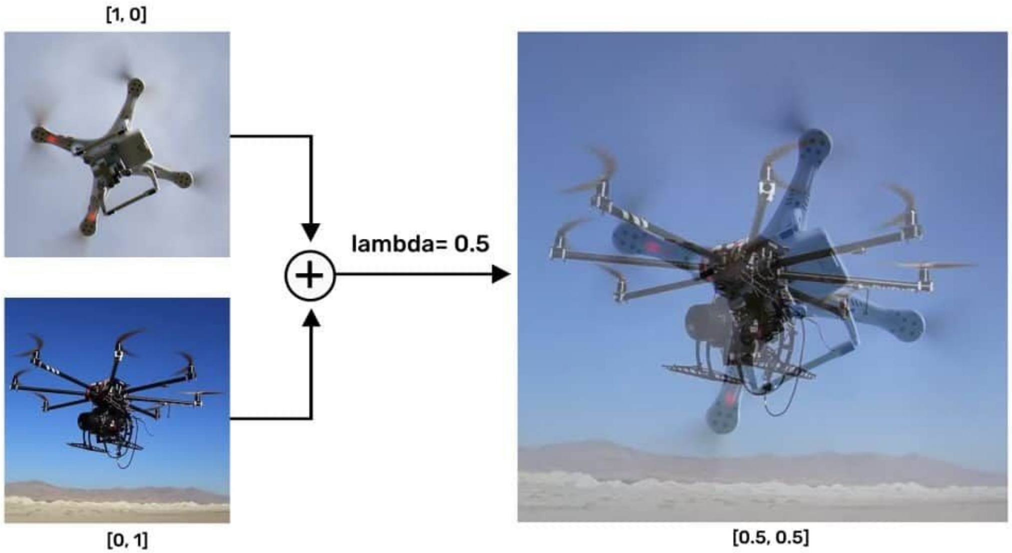
- Size
- Scale
- Occlusion conditions etc.

However, it is difficult to model a mapping function from these factors to the positive anchor number **K**. Hence it is done on the basis of IoU value. The IoU values of the **anchors** to the ground truth(GT) are summed up to represent the GT's estimated number of positive anchors.

The intuition is such that the number of positive anchors for a certain GT should be positively correlated with the number of anchors that have well regressed.

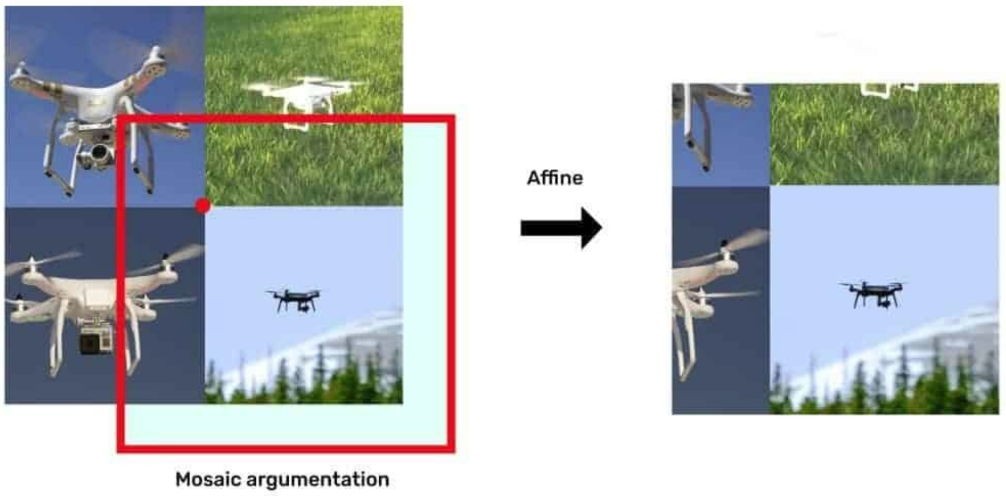
Strong Data Augmentation in YOLOX

Mixup Augmentation



Strong Data Augmentation in YOLOX

Mosaic Augmentation



Performance gain - step by step

Methods	AP (%)	Parameters	GFLOPs	Latency	FPS
YOLOv3-ultralytics[2]	44.3	63.00 M	157.3	10.5 ms	95.2
YOLOv3 baseline	38.5	63.00 M	157.3	10.5 ms	95.2
+decoupled head	39.6 (+1.1)	63.86 M	186.0	11.6 ms	86.2
+strong augmentation	42.0 (+2.4)	63.86 M	186.0	11.6 ms	86.2
+anchor-free	42.9 (+0.9)	63.72 M	185.3	11.1 ms	90.1
+multi positives	45.0 (+2.1)	63.72 M	185.3	11.1 ms	90.1
+SimOTA	**47.3** (+2.3)	63.72 M	185.3	11.1 ms	90.1
+NMS free (optional)	46.5 (-0.8)	67.27 M	205.1	13.5 ms	74.1

Table 2: Roadmap of YOLOX-Darknet53 in terms of AP (%) on COCO *val*. All the models are tested at 640×640 resolution, with FP16-precision and batch=1 on a Tesla V100. The latency and FPS in this table are measured without post-processing.

For more details…

Check the following:

YOLOX Object Detector Paper Explanation and Custom Training
https://learnopencv.com/yolox-object-detector-paper-explanation-and-custom-training/

CenterNet: Objects as Points – Anchor Free Object Detection Explained
https://learnopencv.com/centernet-anchor-free-object-detection-explained/

Paper Review: “YOLOX: Exceeding YOLO Series in 2021”
https://medium.com/mlearning-ai/paper-review-yolox-exceeding-yolo-series-in-2021-ffc1bd94a1f3

The YOLOX paper
https://arxiv.org/pdf/2107.08430.pdf

…more interesting blog posts of your choice! Plenty of stuff available online!

Questions and Answers