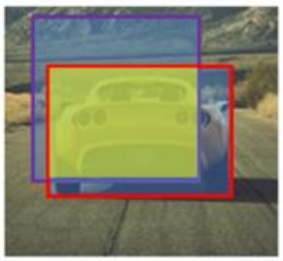
Today lecture:

1. **Intersect over union**
2. **Non maxima suppression**
3. **Anchor box**
4. **Complete execution of Yolo algorithm**
5. **Mean average precision -mAP**
6. **Transfer learning**

Evaluation

- Intersection over union



Intersection over union (IoU)

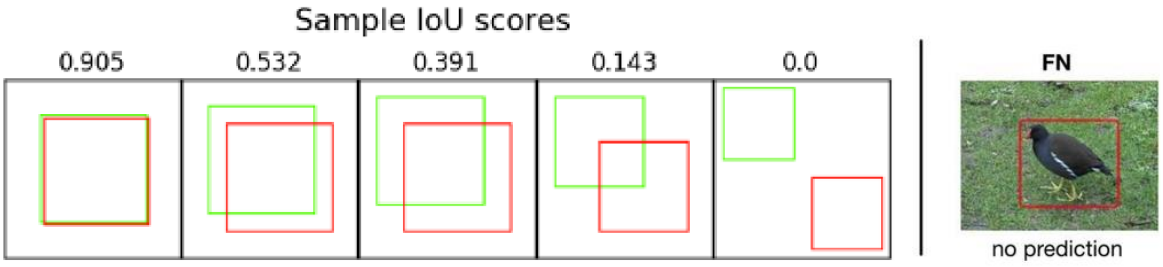
$$= \frac{\text{size of } \square}{\text{size of } \square}$$

General Criteria:

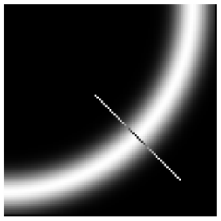
"Correct" if IoU $\geq$ **0**.5

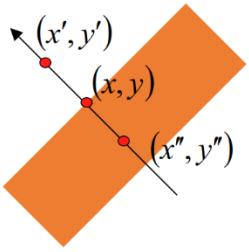
Evaluation

Sample IoU



Non-maximum suppression





$$M(x,y)=\begin{cases}|\nabla S|(x,y) & \text{if } |\nabla S|(x,y)>|\Delta S|(x',y') \\ & \&\ |\Delta S|(x,y)>|\Delta S|(x'',y'') \\ 0 & \text{otherwise}\end{cases}$$

x' and **x**'' are the neighbors of **x** along normal direction to an edge

Non-maxima suppression (NMS)

- Iterate over all detections
  - Pick the highest scoring box
- Find overlap
  - with all other boxes
- Remove boxes with high overlap
  - Threshold, usually 0.5

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Shape
Description automatically generated

Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU)
(Also called "Jaccard similarity" or "Jaccard index"):

$$\frac{\textit{Area of Intersection}}{\textit{Area of Union}}$$

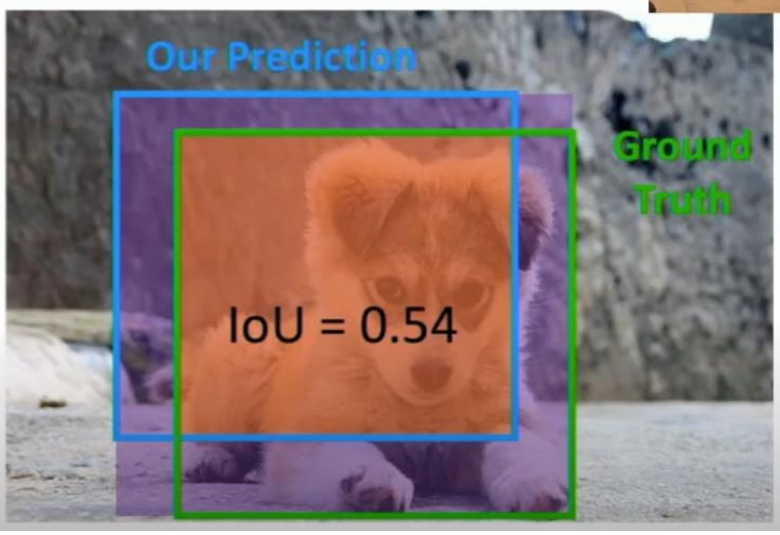
Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU)
(Also called "Jaccard similarity" or "Jaccard index"):

$$\frac{\textit{Area of Intersection}}{\textit{Area of Union}}$$

IoU > 0.5 is "decent"



Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU)
(Also called "Jaccard similarity" or "Jaccard index"):

$$\frac{\textit{Area of Intersection}}{\textit{Area of Union}}$$

IoU > 0.5 is "decent",
IoU > 0.7 is "pretty good",

Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

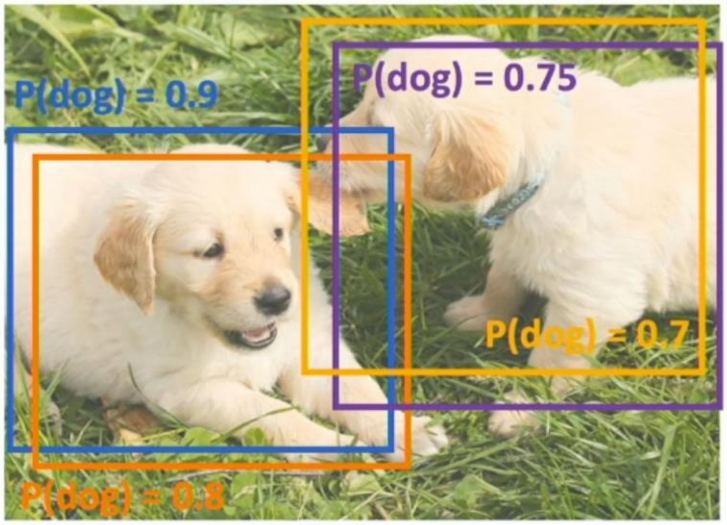
Intersection over Union (IoU)
(Also called "Jaccard similarity" or "Jaccard index"):

$$\frac{\textit{Area of Intersection}}{\textit{Area of Union}}$$

IoU > 0.5 is "decent",
IoU > 0.7 is "pretty good",
IoU > 0.9 is "almost perfect"

Overlapping Boxes

Problem: Object detectors often output many overlapping detections:



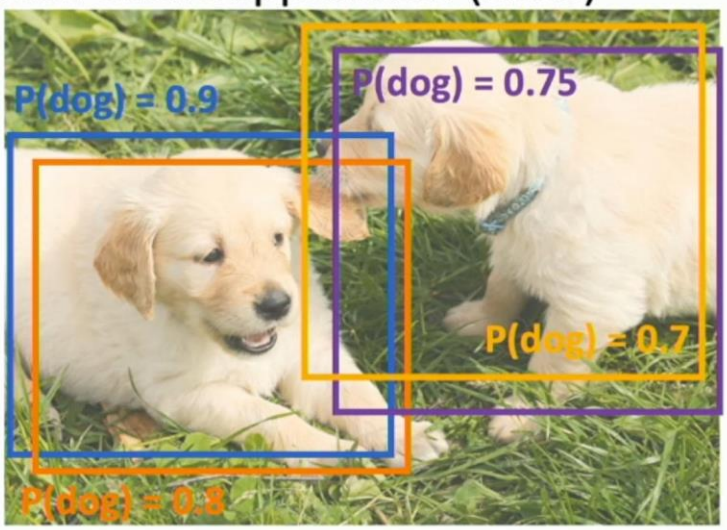
Puppy image is CC0 Public Domain

Overlapping Boxes: Non-Max Suppression (NMS)

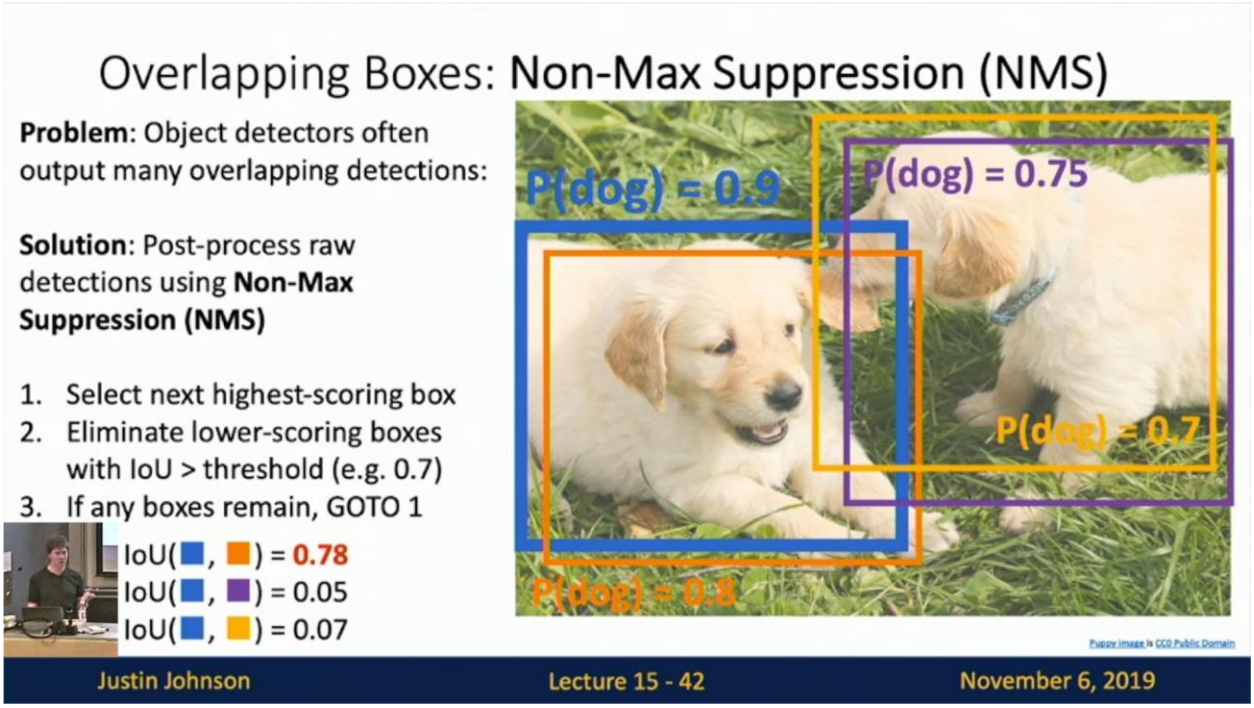
Problem: Object detectors often output many overlapping detections:

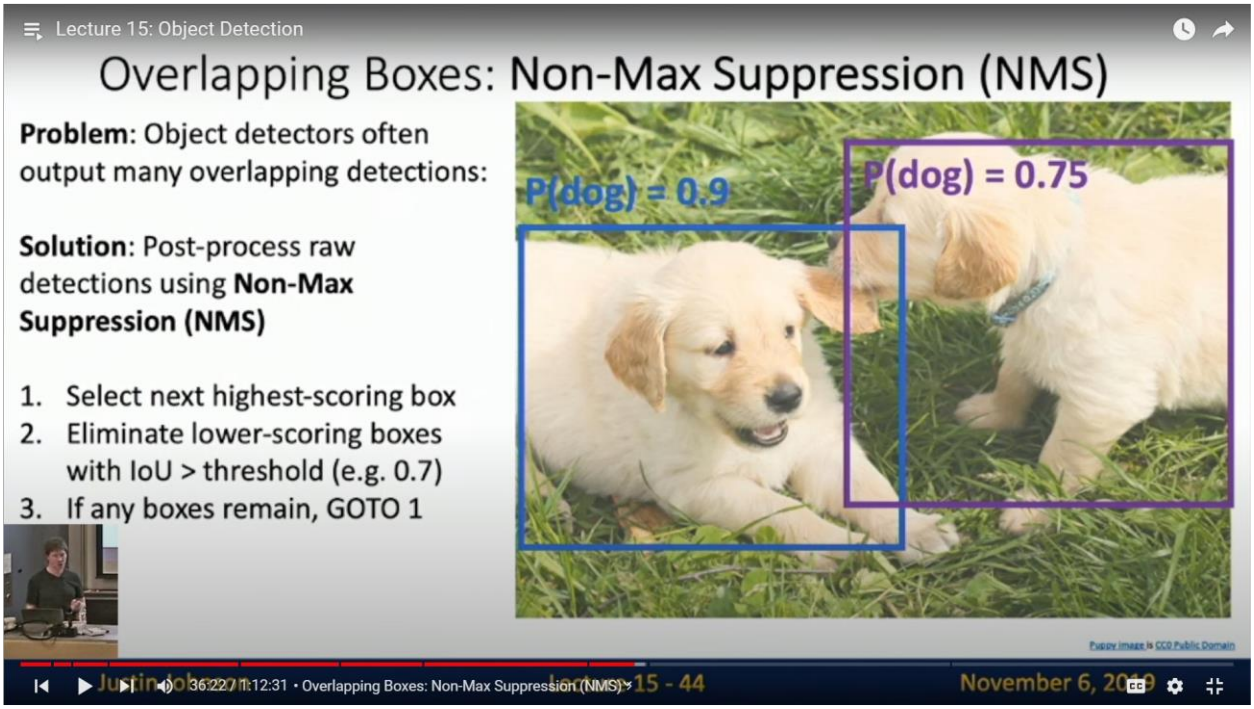
Solution: Post-process raw detections using **Non-Max Suppression (NMS)**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1



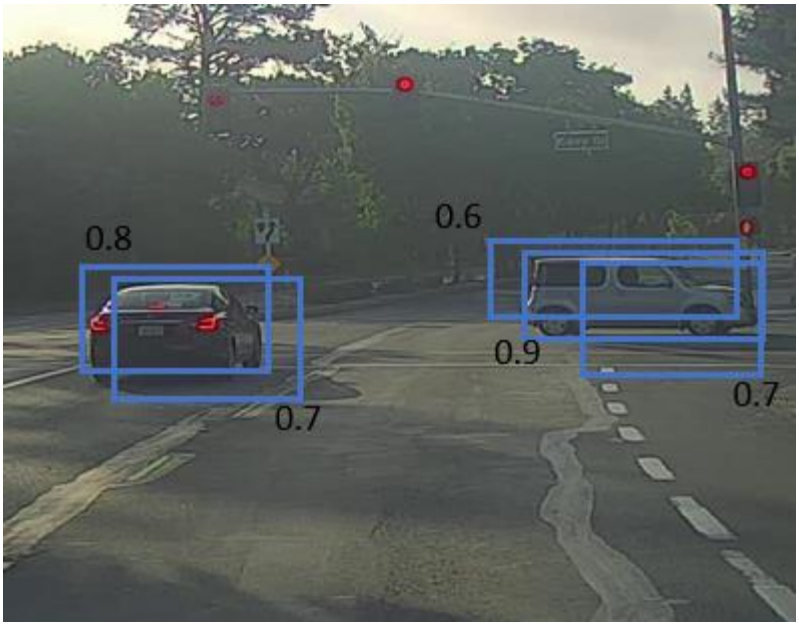
Puppy image is CC0 Public Domain



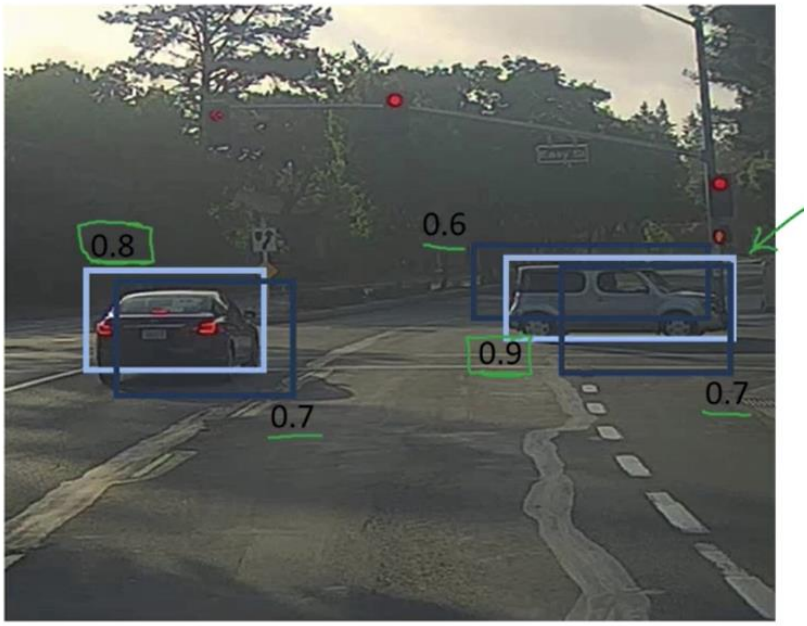


- One of the problems we have addressed in YOLO is that **it can detect an object multiple times**

- Non-max Suppression is a way to make sure that YOLO detects the object just once.
- For example:



i) Discard Box with low Pc
ii) For a specific object class
   - Output 0.9 box,
   - discard overlapping boxes 0.6 and 0.7, as have IOU >=0.5
   - Output 0.8 box
   - discard overlapping box 0.7, as have IOU>=0.
   - We left with 2 boxes

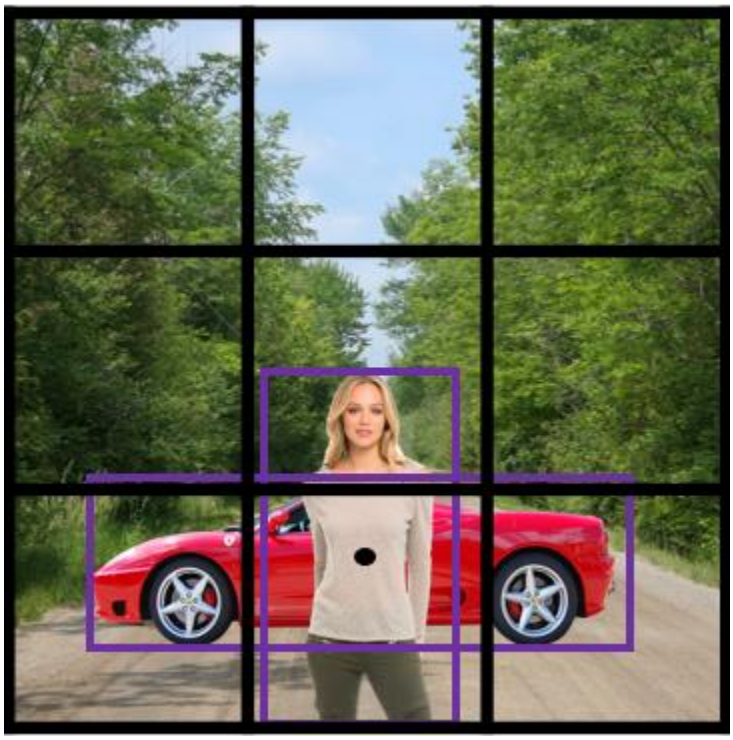


- Non-max suppression algorithm:
  - Lets assume that we are targeting **one class as an output class**.
  - Y shape should be [Pc, bx, by, bh, hw] Where Pc is the probability if that object occurs.
  - **Discard** all boxes with **Pc <= 0.6 (from prediction volume)**
  - While there are any remaining boxes:

    a. Pick the box with the **largest Pc** Output that as a prediction.
    b. **Discard** any remaining box with **IoU > 0.5** with that box output in the previous step i.e any box with high overlap (greater than overlap threshold of 0.5).
- If there are **multiple classes/object types** c you want to detect, you should run the Non-max suppression **c** times, once for every output class.

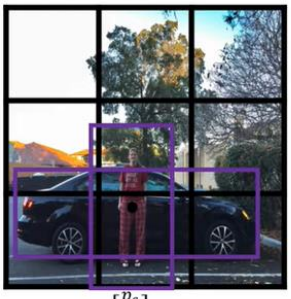
Anchor Boxes

- In YOLO, a grid only detects one object. What if a grid cell wants to detect multiple object?

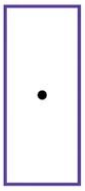


    - Car and person rectangle center belongs to same region.
    - In practice this happens rarely.
- Overlapping objects:

Overlapping objects:



Anchor box 1:



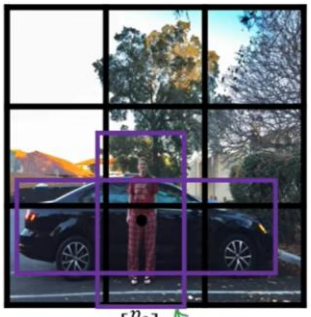
Anchor box 2:

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

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

Andrew Ng

Overlapping objects:



Anchor box 1:

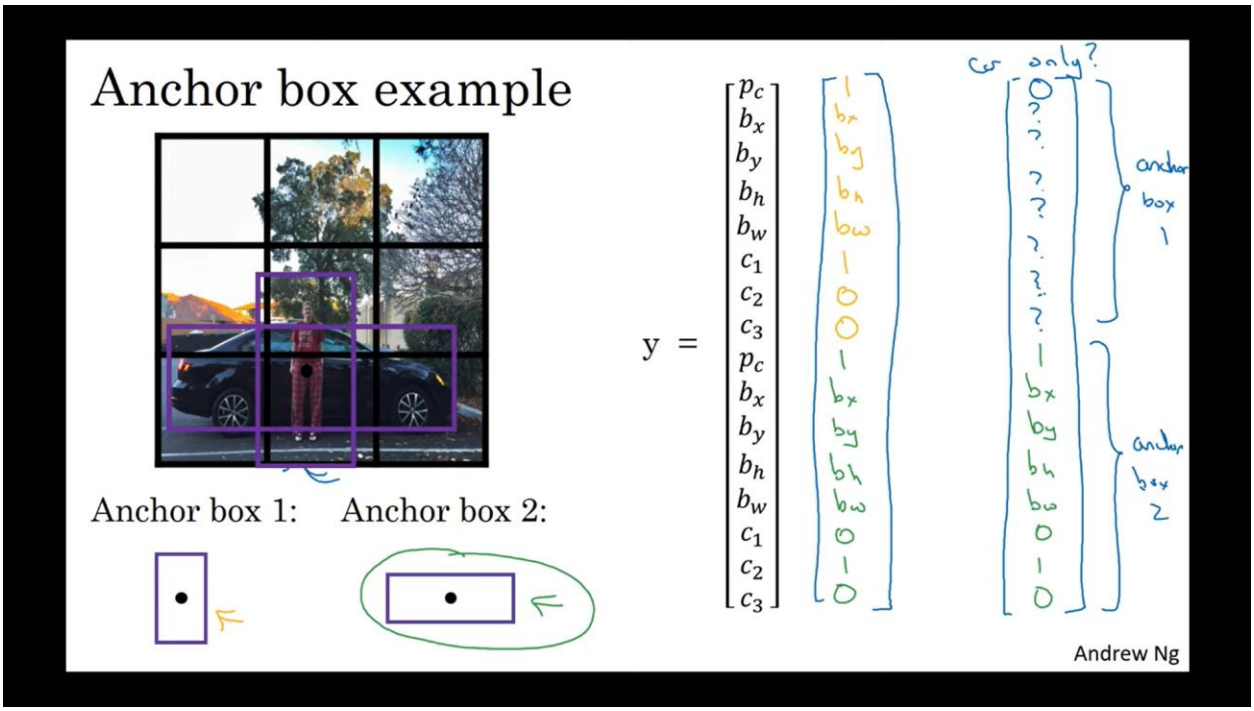
Anchor box 2:

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

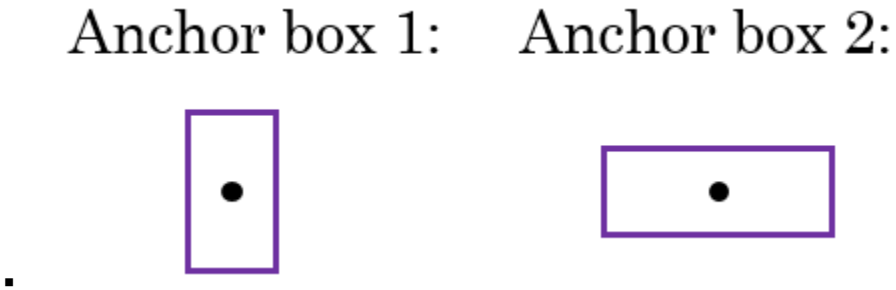
$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ \vdots \\ c_3 \end{bmatrix}$ Anchor box 1 Anchor box 2

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

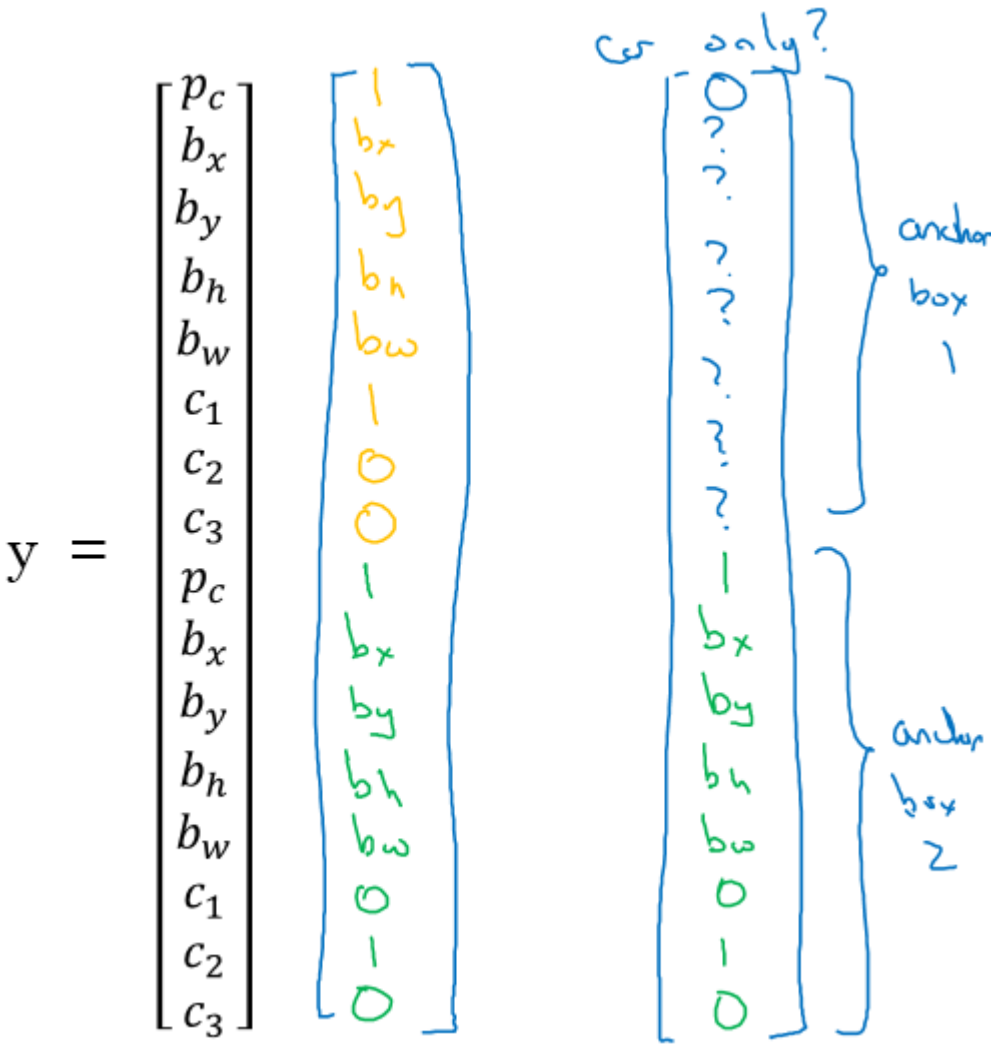
Andrew Ng



- The idea of Anchor boxes helps us solving this issue.
- If Y = [Pc, bx, by, bh, bw, c1, c2, c3] Then to use two anchor boxes like this:
  - Y = [Pc, bx, by, bh, bw, c1, c2, c3, Pc, bx, by, bh, bw, c1, c2, c3] We simply have repeated the one anchor Y.
  - The two anchor boxes you choose should be known as a shape:



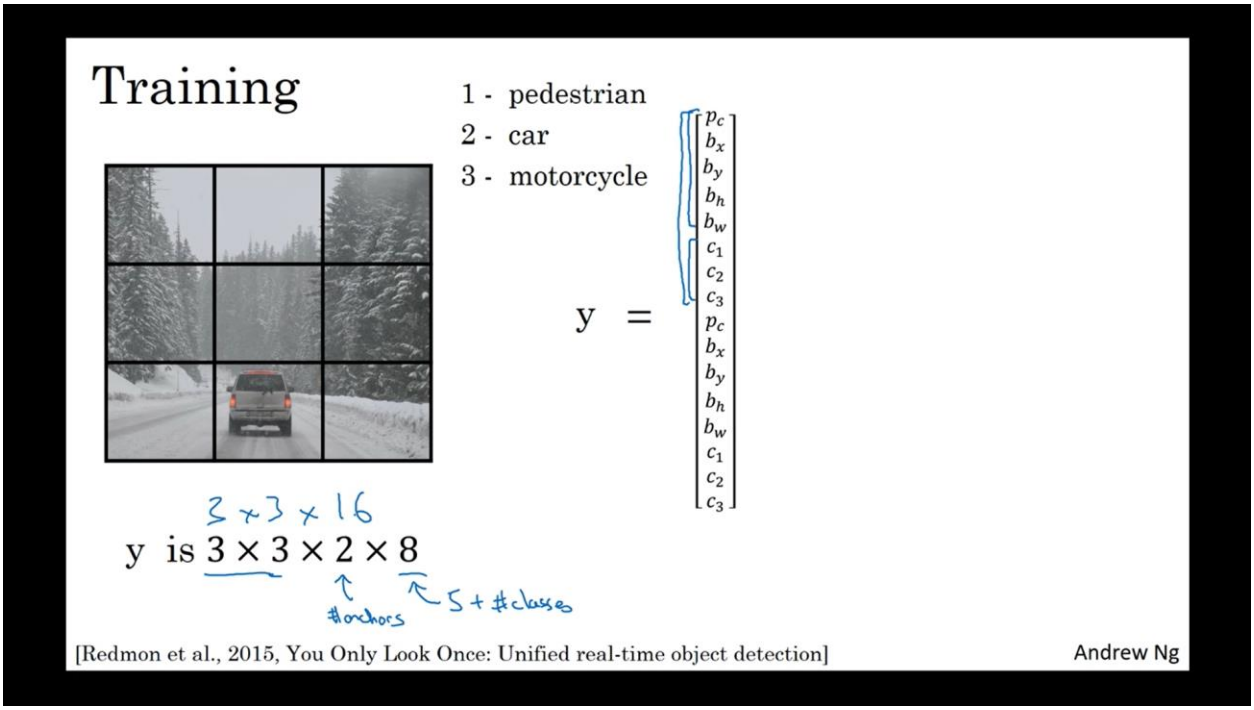
- So Previously, each object in training image is assigned to grid cell that contains that object's midpoint.
- With two anchor boxes, Each object in training image is assigned to grid cell that contains object's midpoint and anchor box for the grid cell with highest IoU. You have to check where your object should be based on its rectangle closest to which anchor box.
- Example of data:



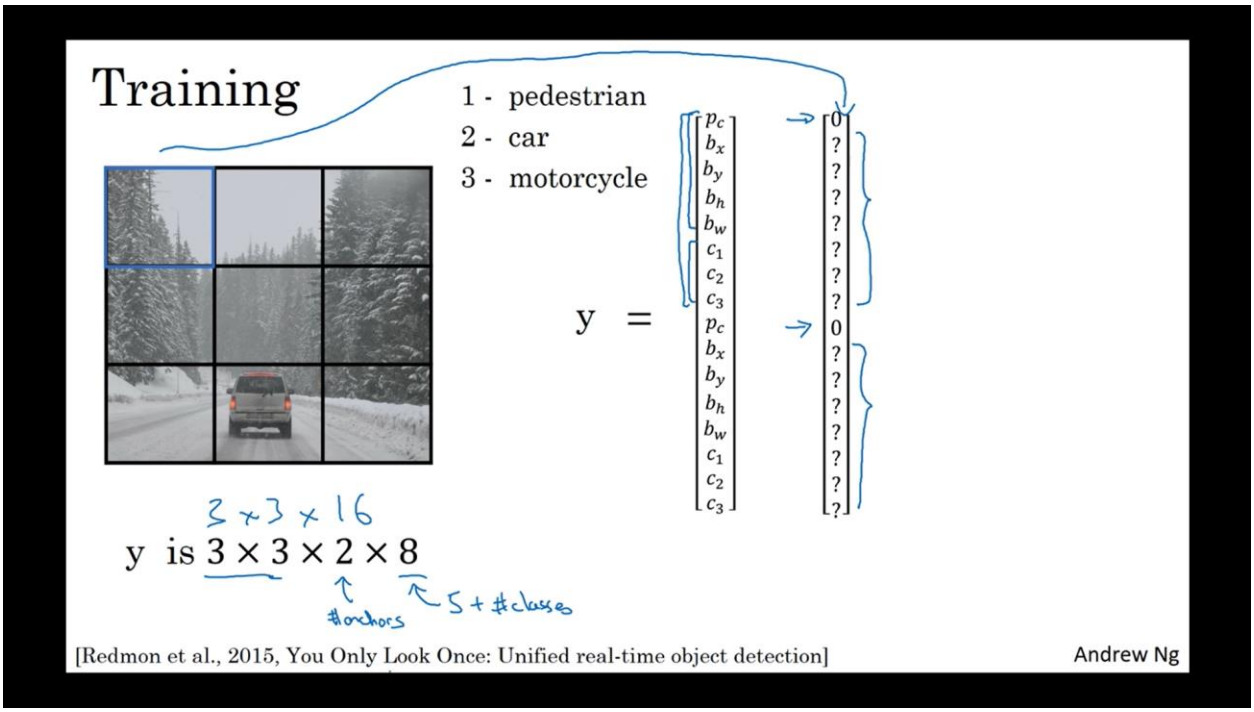
-
  - Where the car was near the anchor 2 than anchor 1.
- You may have two or more anchor boxes but you should know their shapes.
  - how do you choose the anchor boxes and people used to just choose them by hand. Maybe five or ten anchor box shapes that spans a variety of shapes that cover the types of objects you seem to detect frequently.
  - You may also use a k-means algorithm on your dataset to specify that.
- Anchor boxes allows your algorithm to specialize, means in our case to easily detect wider images or taller ones.

- **YOLO Algorithm**

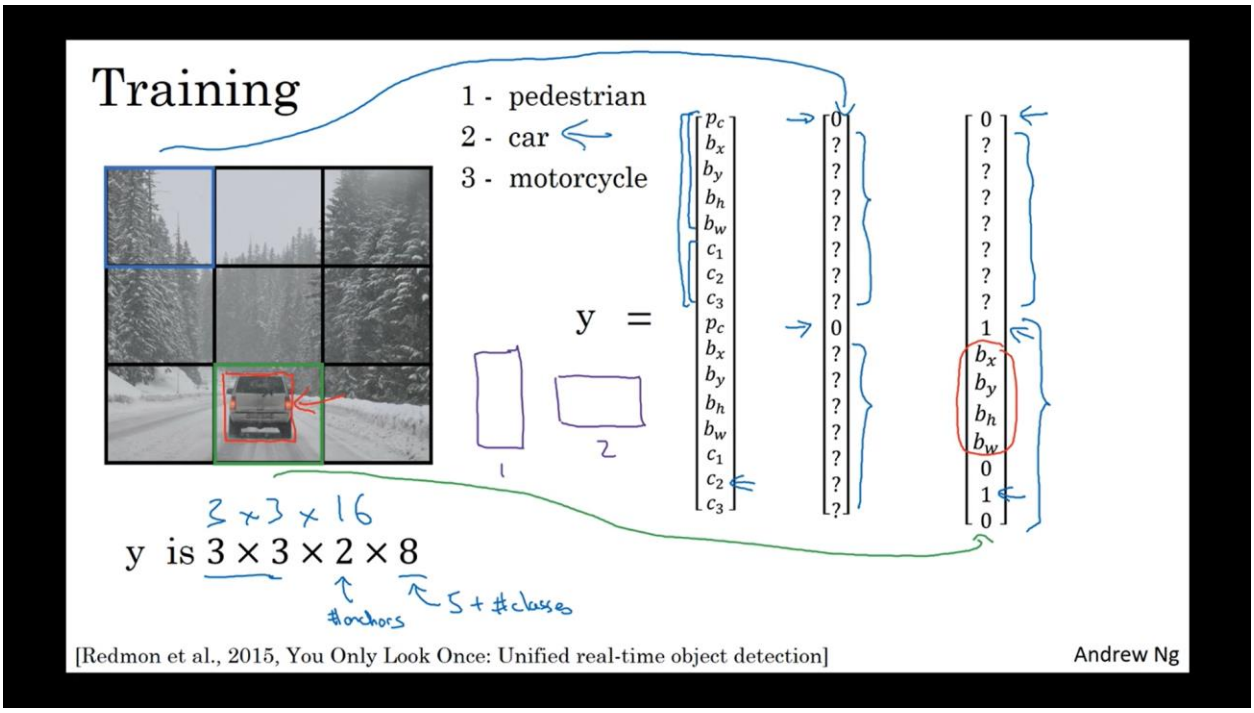
Training Dataset : x is input image of size 19x19x3, y is output of size 3x3x16=3x3x2x8



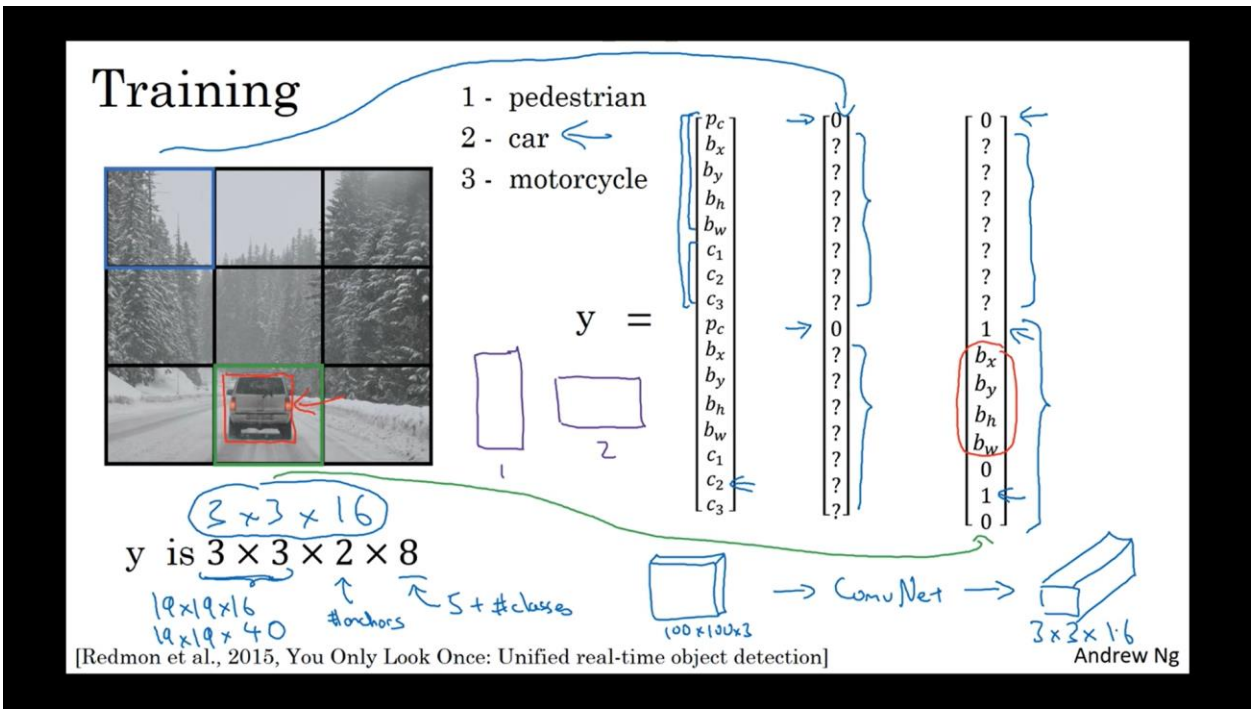
- If a 14x14x3 grid does not contain any object, the y vector [0, ?,?,?,?,?,?,?,?,0,?,?,?,?,?,?]

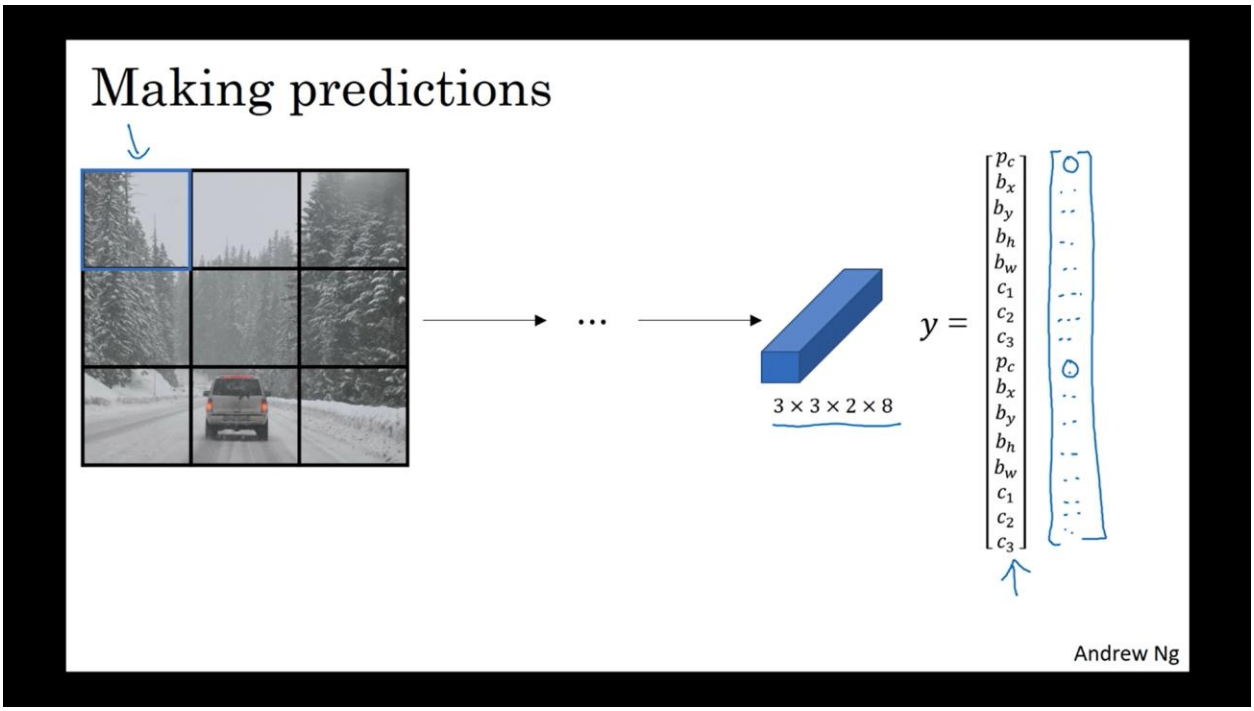


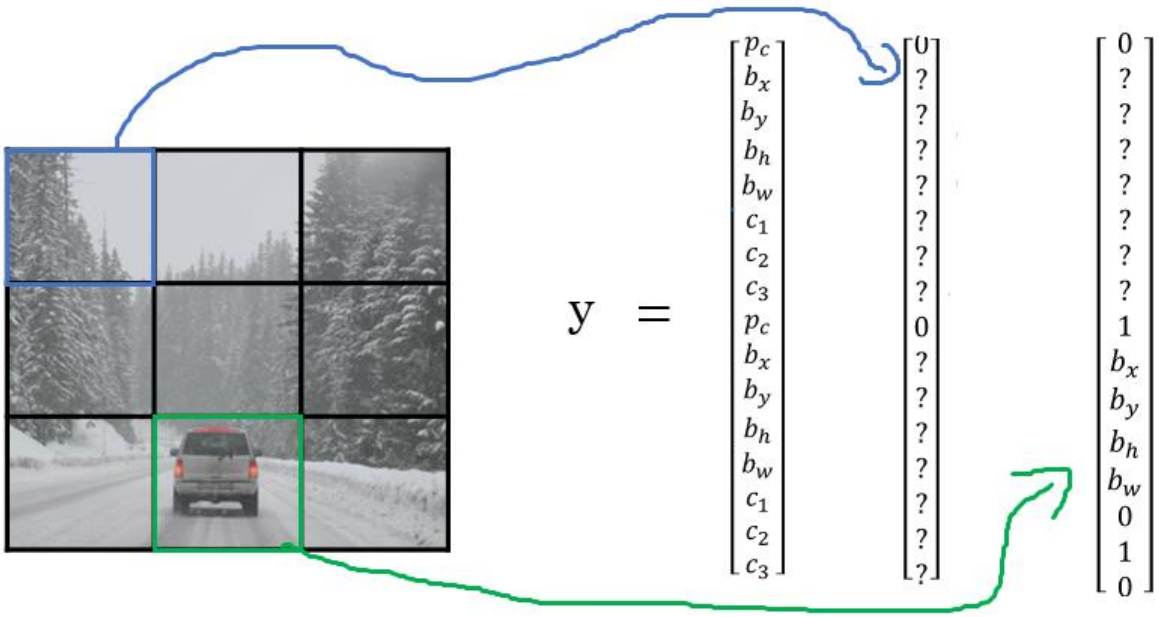
If a grid of 14x14 contains the object then y = [0, ???????,1,bx,by,bw,bh,0,1,0]



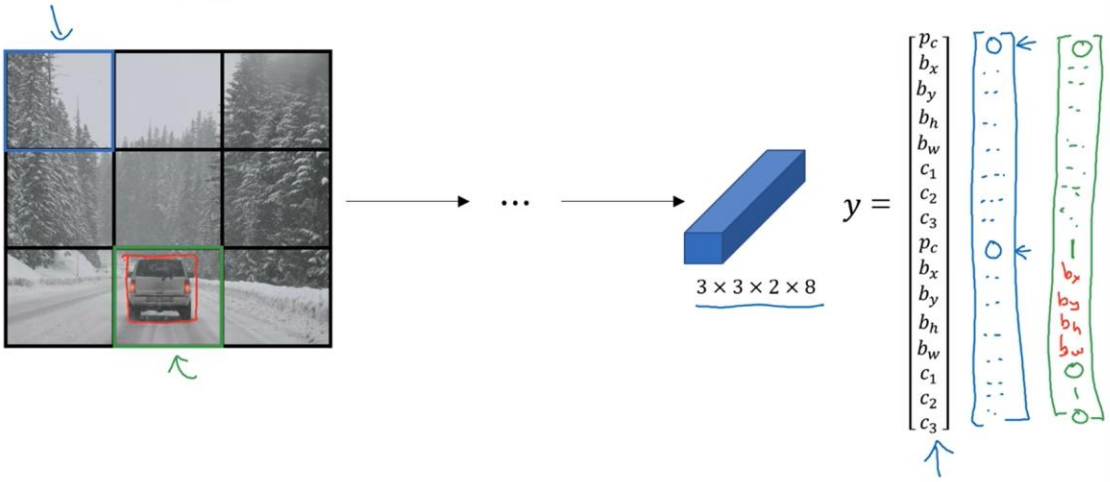
See the relationship between input image X(100x100x3) and ouput y(19x19x) dimensions




Making predictions
...
$3 \times 3 \times 2 \times 8$
$y =$
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
Andrew Ng


y =
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
0 ? ? ? ? ? ? ? 0 ? ? ? ? ? ? ?
0 ? ? ? ? ? ? ? 1 b_x b_y b_h b_w 0 1 0

Making predictions



Andrew Ng

Outputting the non-max supressed outputs

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

Andrew Ng

Outputting the non-max supressed outputs

- For each grid call, get 2 predicted bounding boxes.
- Get rid of low probability predictions.

Andrew Ng

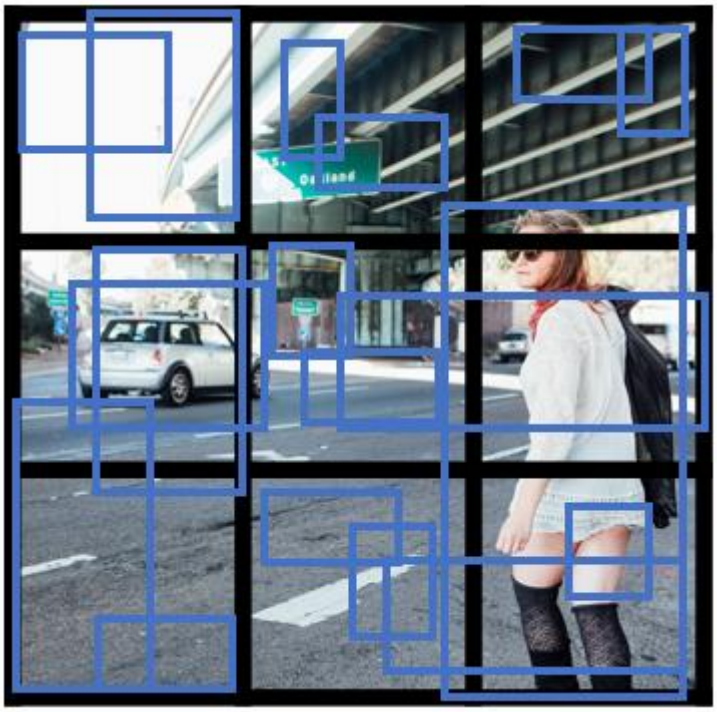
Outputting the non-max supressed outputs

- 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.

Andrew Ng

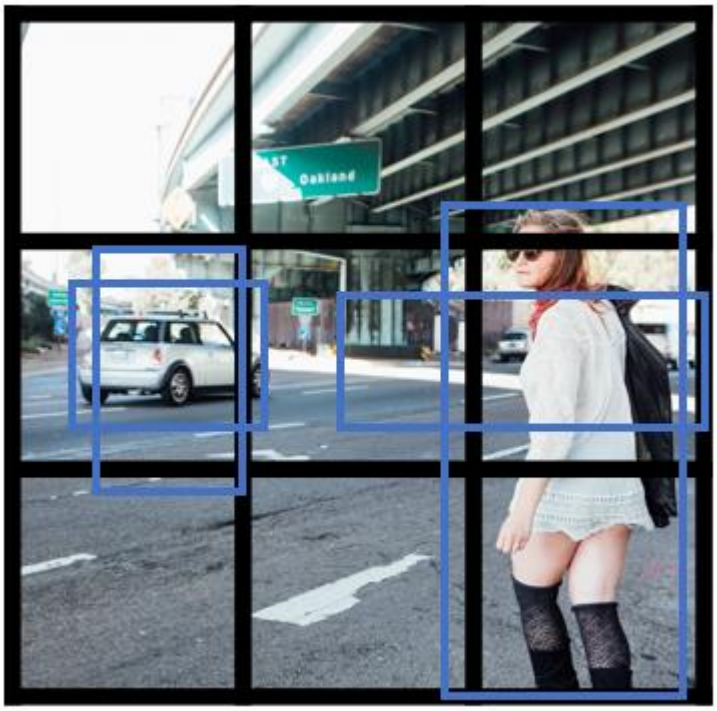
....
,,

- YOLO is a state-of-the-art object detection model that is fast and accurate
- Lets sum up and introduce the whole YOLO algorithm given an example.
- Suppose we need to do object detection for our autonomous driver system.It needs to identify three classes:
    i. Pedestrian (Walks on ground).
    ii. Car.
    iii. Motorcycle.
- We decided to choose two anchor boxes, a taller one and a wide one.
    i. Like we said in practice they use five or more anchor boxes hand made or generated using k-means.
- Our labeled Y shape will be [Ny, HeightOfGrid, WidthOfGrid, 16], where Ny is number of instances and each row (of size 16) is as follows:
    i. [Pc, bx, by, bh, bw, c1, c2, c3, Pc, bx, by, bh, bw, c1, c2, c3]
- Your dataset could be an image with a multiple labels and a rectangle for each label, we should go to your dataset and make the shape and values of Y like we agreed.
    i. An example:
        ▪
    ii. We first initialize all of them to zeros and ?, then for each label and rectangle choose its closest grid point then the shape to fill it and then the best anchor point based on the IOU. so that the shape of Y for one image should be [HeightOfGrid, WidthOfGrid,16]
- Train the labeled images on a Conv net. you should receive an output of [HeightOfGrid, WidthOfGrid,16] for our case.
- To make predictions, run the Conv net on an image and run Non-max suppression algorithm for each class you have in our case there are 3 classes.
    i. You could get something like that:



- Total number of generated boxes are grid_width * grid_height * no_of_anchors = 3 x 3 x 2

ii. By removing the low probability predictions you should have:



iii. Then get the best probability followed by the IOU filtering:

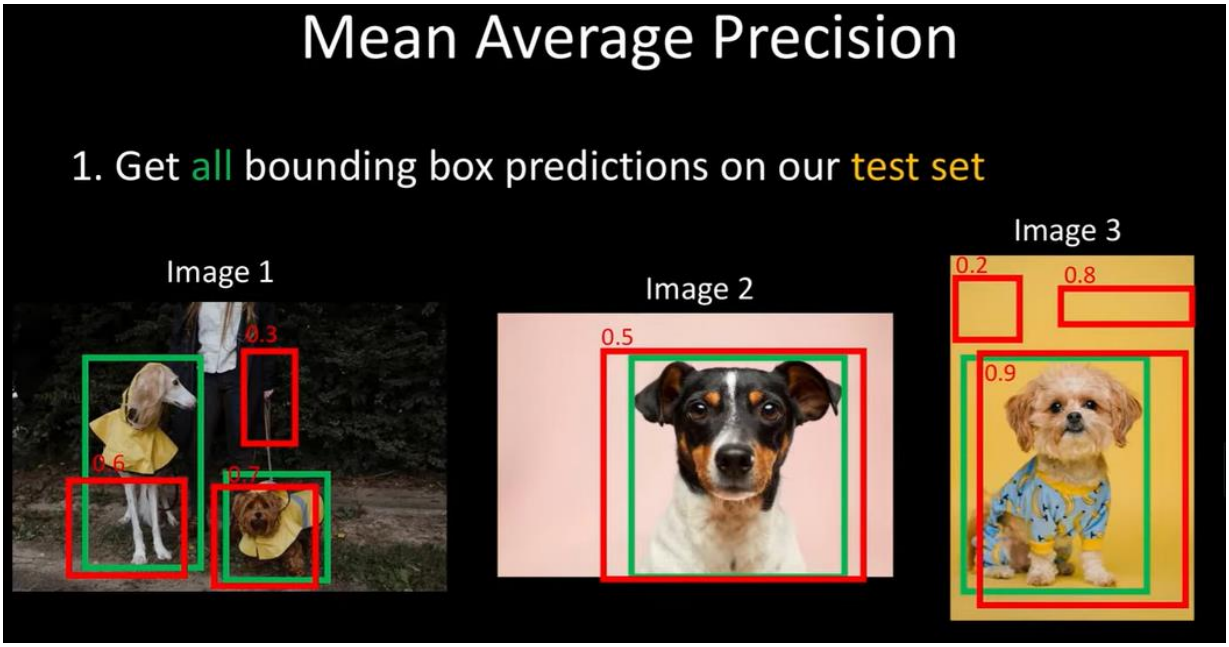
- 

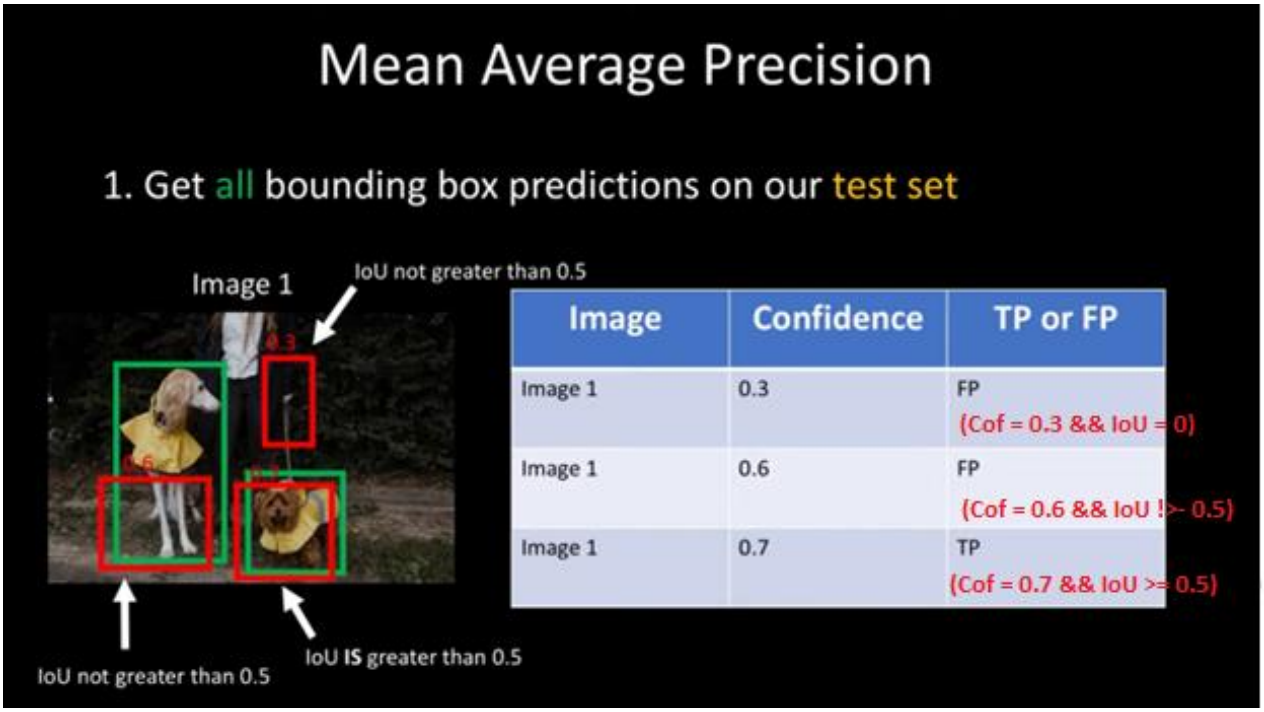
- YOLO are not good at detecting smaller object.
- [YOLO9000 Better, faster, stronger]

;;;;;;;;;;;;;;;

Mean Average Precsion: MAP is a key evaluation metric in object detection that assesses a model's accuracy across all classes by considering both classification and localization performance

https://www.youtube.com/watch?v=FppOzcDvaDI





Mean Average Precision

1. Get all bounding box predictions on our test set

Image 2

Image	Confidence	TP or FP
Image 2	0.5	TP

As (Confidence >=0.5 && IoU => 0.5) so it is TP

If (Confidence >=0.5 && IoU =>0.5) then it is TP, else it is FP

Mean Average Precision

1. Get all bounding box predictions on our test set

Image 3

Image	Confidence	TP or FP
Image 3	0.2	FP (As Conf = 0.2 && IOU= 0
Image 3	0.8	FP (As Conf = 0.8 && IOU= 0)
Image 3	0.9	TP (As Conf = 0.9 && IOU>=0.5)

1. Get all bounding box predictions on our test set

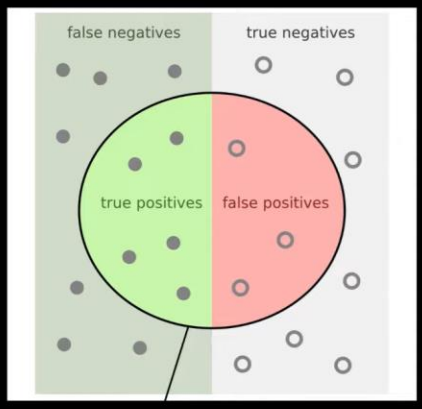
Image	Confidence	TP or FP
Image 1	0.3	FP
Image 1	0.6	FP
Image 1	0.7	TP
Image 2	0.5	TP
Image 3	0.2	FP
Image 3	0.8	FP
Image 3	0.9	TP

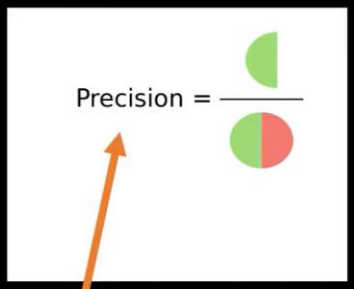
Mean Average Precision

2. Sort by descending **confidence** score

Image	Confidence	TP or FP
Image 3	0.9	TP
Image 3	0.8	FP
Image 1	0.7	TP
Image 1	0.6	FP
Image 2	0.5	TP
Image 1	0.3	FP
Image 3	0.2	FP

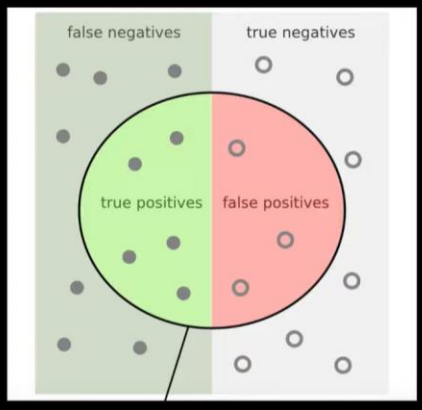
Precision and Recall

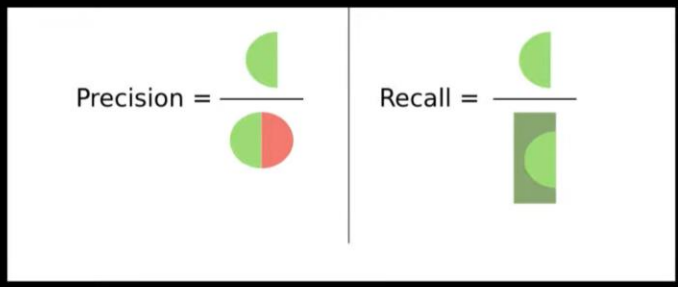




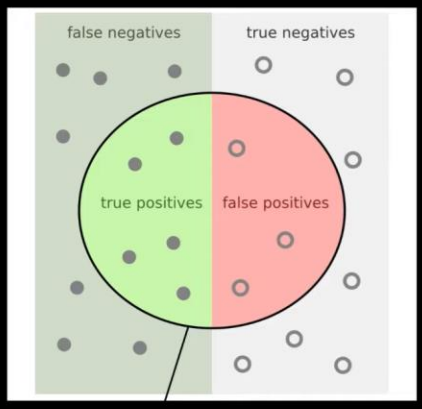
(Of **all** bounding box predictions, what fraction was actually correct?)

Precision and Recall





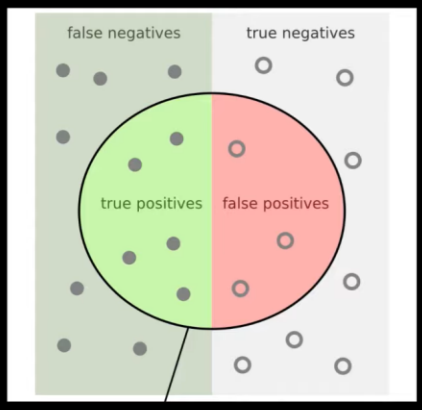
Precision and Recall



Precision = $\frac{\text{true positives}}{\text{true positives} + \text{false positives}}$ Recall = $\frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$

(Of **all** target bounding boxes, what fraction did we correctly detect?)

Precision and Recall



Precision = $\frac{\text{true positives}}{\text{true positives} + \text{false positives}}$ Recall = $\frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$

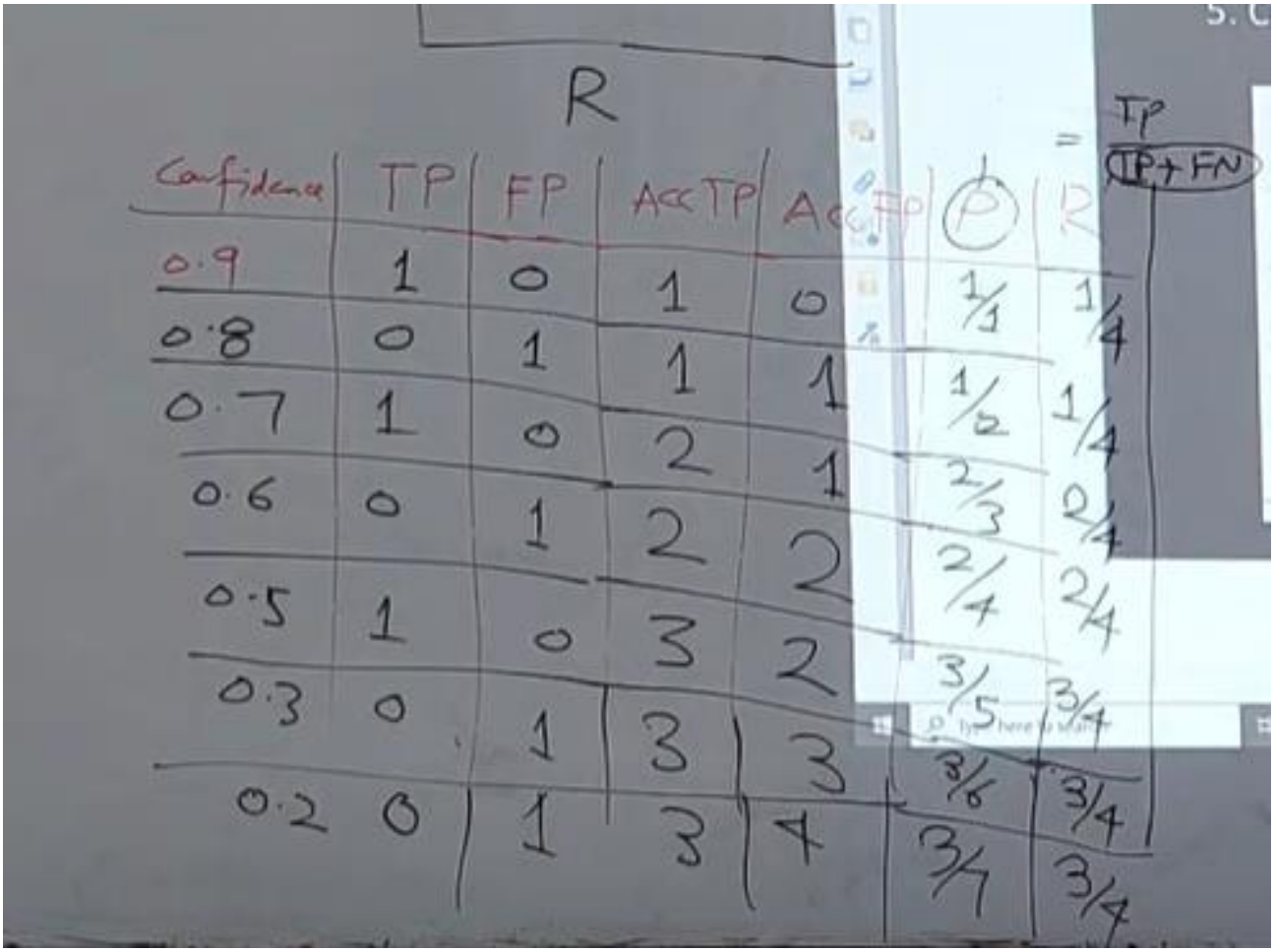
There's a battle between precision and recall! Different applications may prioritize recall and others precision

Not missing any person in prediction in self-driving car (Relax Pc and IOU threshold)

Mean Average Precision

3. Calculate the Precision and Recall as we go through all outputs

Image	Confidence	TP or FP	Precision	Recall
Image 3	0.9	TP		
Image 3	0.8	FP		
Image 1	0.7	TP		
Image 1	0.6	FP		
Image 2	0.5	TP		
Image 1	0.3	FP		
Image 3	0.2	FP		



Mean Average Precision

3. Calculate the Precision and Recall as we go through all outputs

Image	Confidence	TP or FP	Precision	Recall
Image 3	0.9	TP	1 / 1	1 / 4
Image 3	0.8	FP	1 / 2	1 / 4
Image 1	0.7	TP	2 / 3	2 / 4
Image 1	0.6	FP	2 / 4	2 / 4
Image 2	0.5	TP	3 / 5	3 / 4
Image 1	0.3	FP	3 / 6	3 / 4
Image 3	0.2	FP	3 / 7	3 / 4

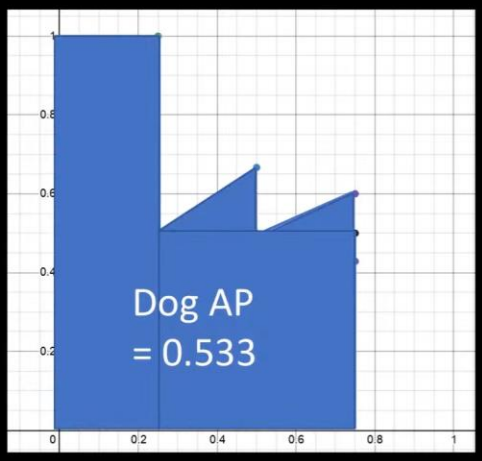
Mean Average Precision

4. Plot the Precision-Recall graph

Precision	Recall
1 / 1	1 / 4
1 / 2	1 / 4
2 / 3	2 / 4
2 / 4	2 / 4
3 / 5	3 / 4
3 / 6	3 / 4
3 / 7	3 / 4

Mean Average Precision

5. Calculate Area under PR curve



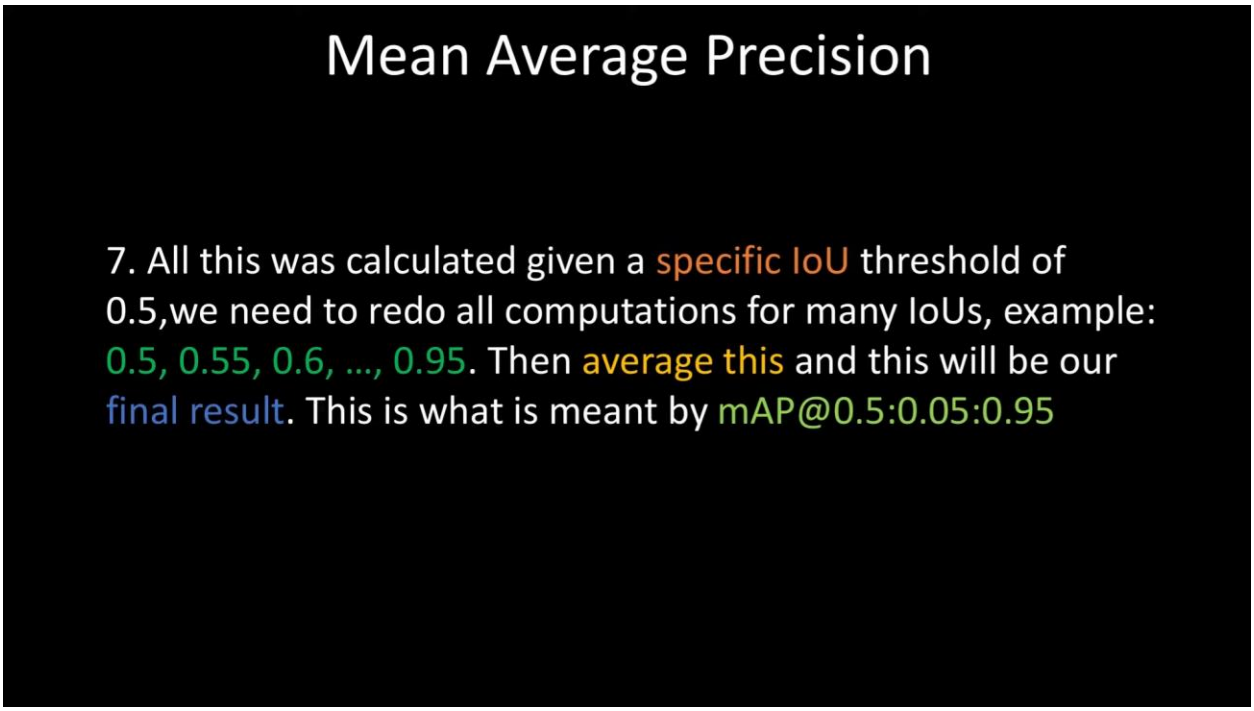
Precision	Recall
1 / 1	1 / 4
1 / 2	1 / 4
2 / 3	2 / 4
2 / 4	2 / 4
3 / 5	3 / 4
3 / 6	3 / 4
3 / 7	3 / 4

Mean Average Precision

6. This was only for dog class, we need to calculate for all classes. Let's say we do this for cats and dogs

- Cat AP = 0.74
- Dog AP = 0.533

mAP = (0.533+0.74)/2 = 0.6365



While F1 score can be considered for evaluating object detection, mAP generally offers more comprehensive and informative insights