deeplearning.ai

Object Detection

Object localization

Hello and welcome back.

What are localization and detection?

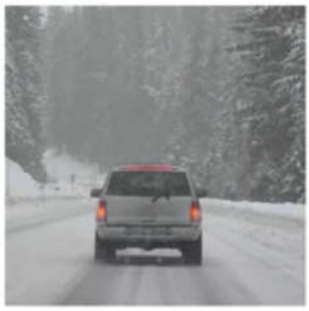
Image classification

You're already familiar with the image classification task where an algorithm

Andrew Ng

What are localization and detection?

Image classification



"Car"

Classification with localization

"Car"

or drawing a red rectangle around the position of the car in the image.

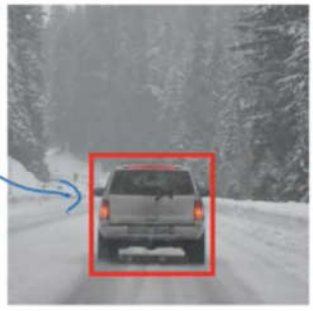
Andrew Ng

What are localization and detection?

Image classification

"Car"

Classification with localization



"Car"

Where the term localization refers to figuring out where in the picture

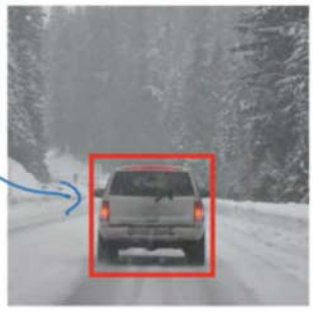
Andrew Ng

What are localization and detection?

Image classification

"Car"

Classification with localization



"Car"

Detection

you have to detect them all and
and localized them all.

Andrew Ng

What are localization and detection?

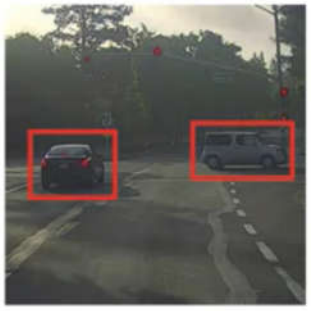
Image classification

"Car"

Classification with localization

"Car"

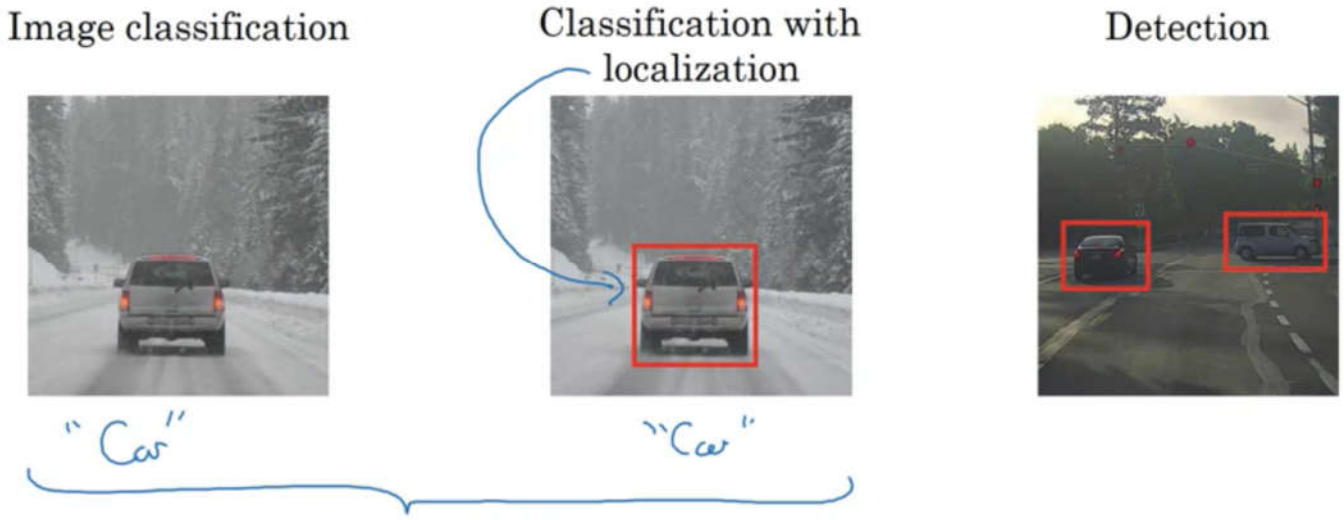
Detection



but maybe other pedestrians and
motorcycles and maybe even other objects.

Andrew Ng

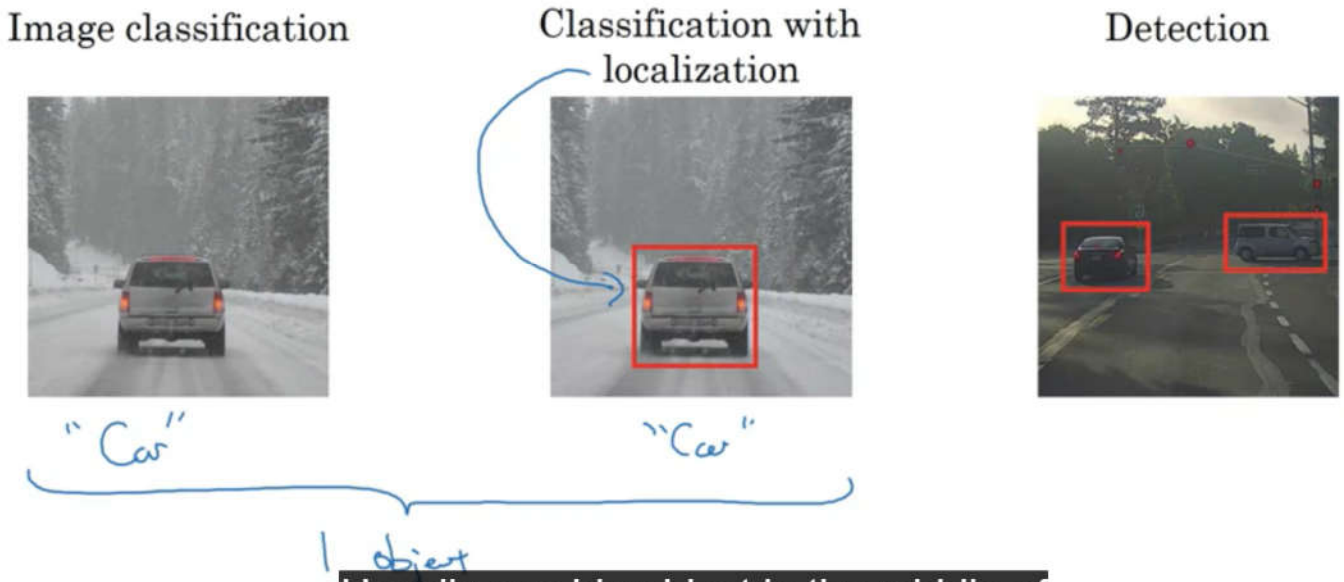
What are localization and detection?



the classification of localization problems usually have one object.

Andrew Ng

What are localization and detection?



Usually one big object in the middle of the image that you're trying to recognize

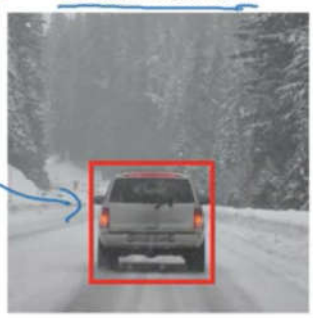
Andrew Ng

What are localization and detection?

Image classification

"Car"

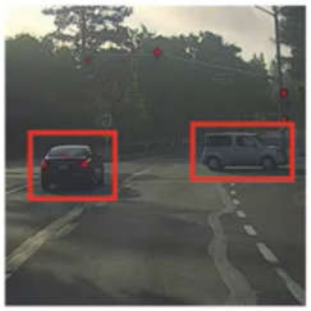
Classification with localization



"Car"

1 object

Detection

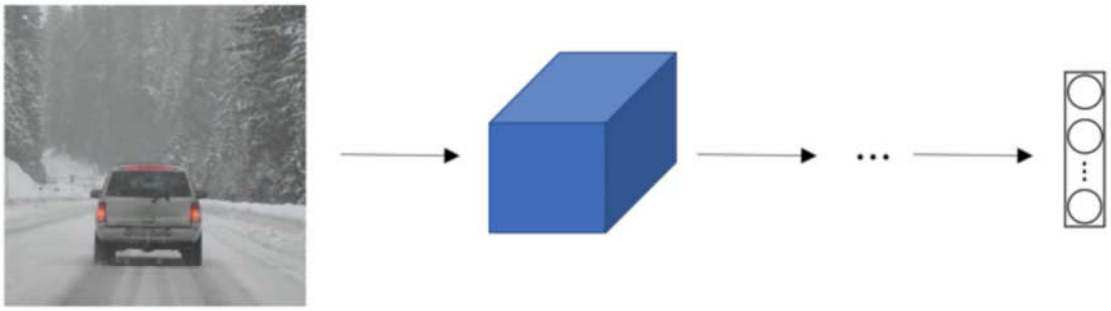


multiple objects

localization will then turn out to be useful for detection.

Andrew Ng

Classification with localization

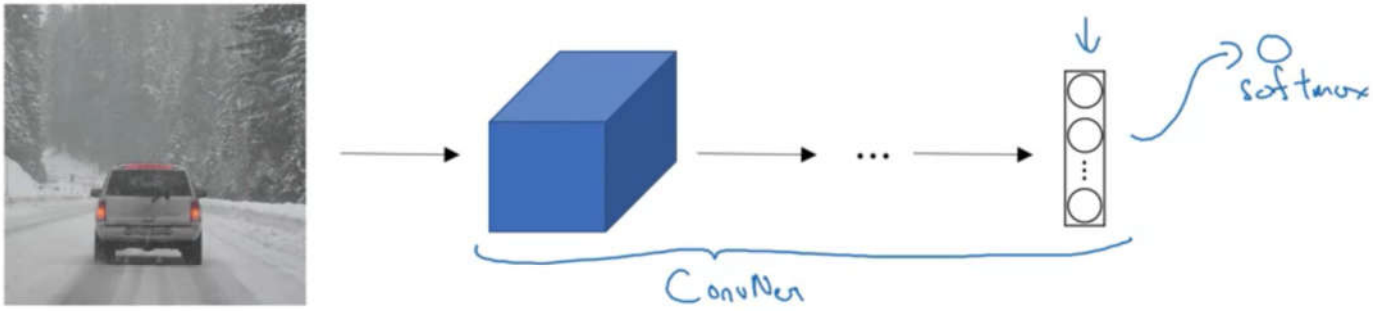


You're already familiar with the image classification problem, in which you might

2:16 / 11:53

Andrew Ng

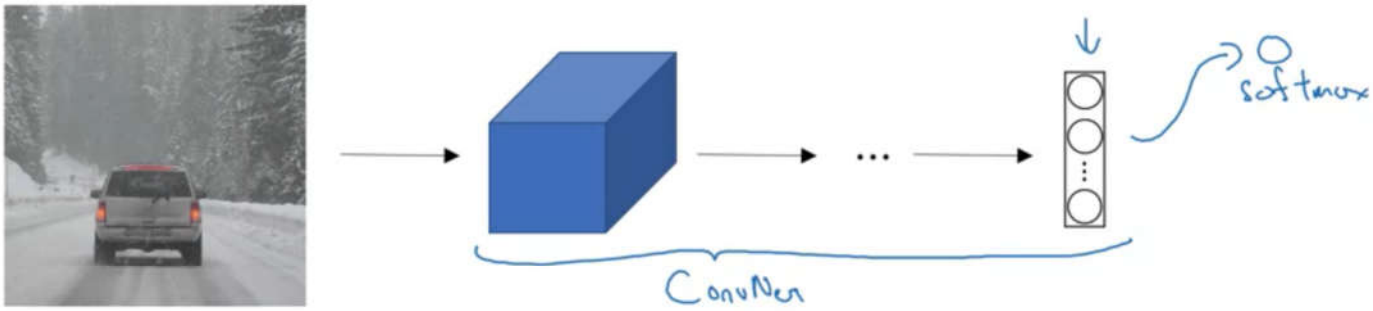
Classification with localization



1 - pedestrian
2 - car
3 - motorcycle
4 - backgro This means none of the above.

Andrew Ng

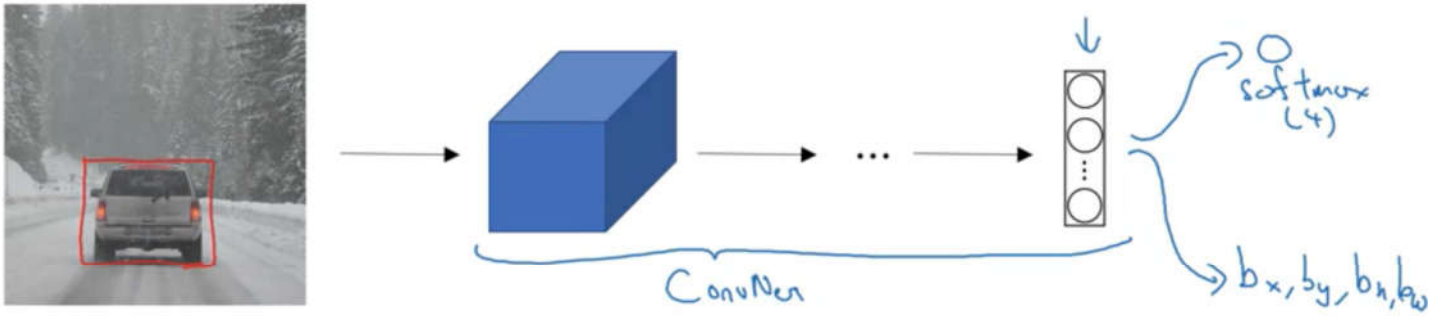
Classification with localization



1 - pedestrian
2 - car
3 - motorcycle
4 - background no car, no motorcycle,
then you might have an output background.

Andrew Ng

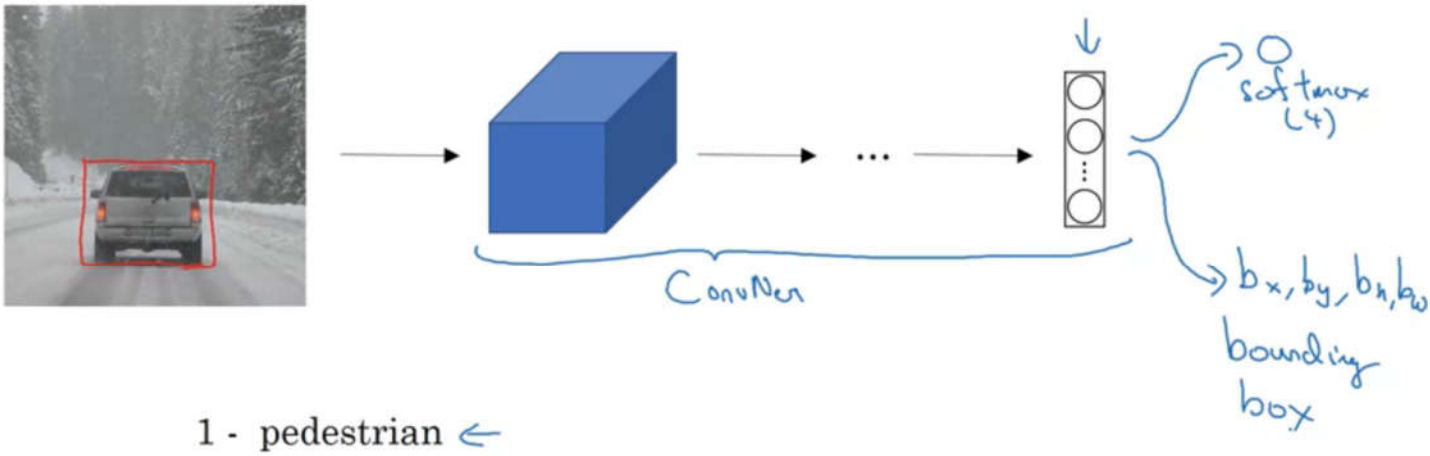
Classification with localization



1 - pedestrian
2 - car
3 - motorcycle
4 - background more numbers, and
I'm going to call them bx, by, bh, and bw.

Andrew Ng

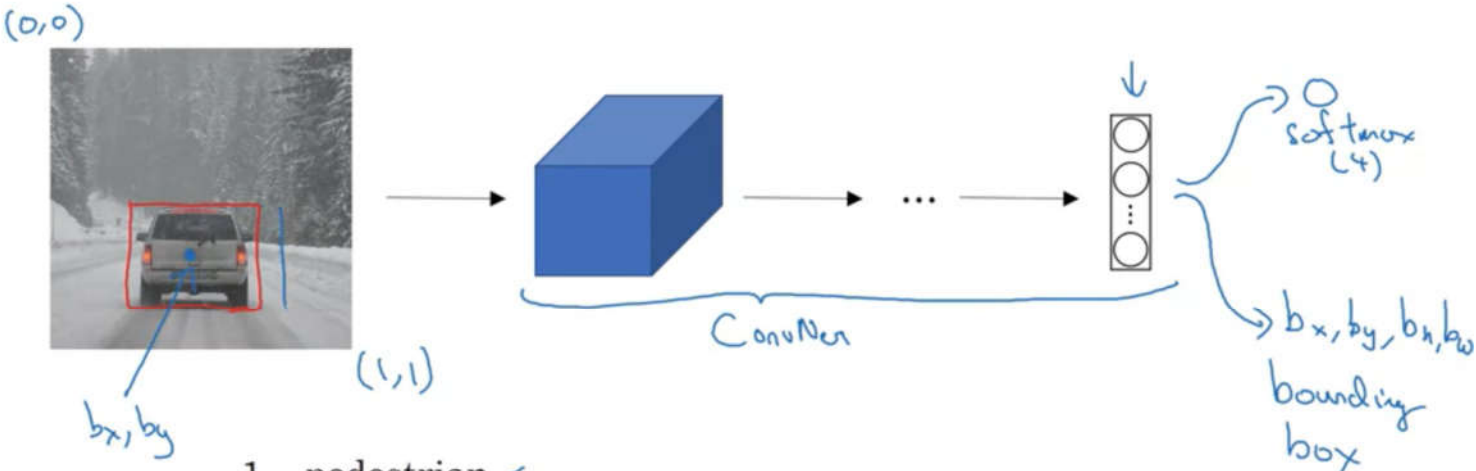
Classification with localization



1 - pedestrian
2 - car
3 - motorcycle
4 - bac And these four numbers parameterized
the bounding box of the detected object.

Andrew Ng

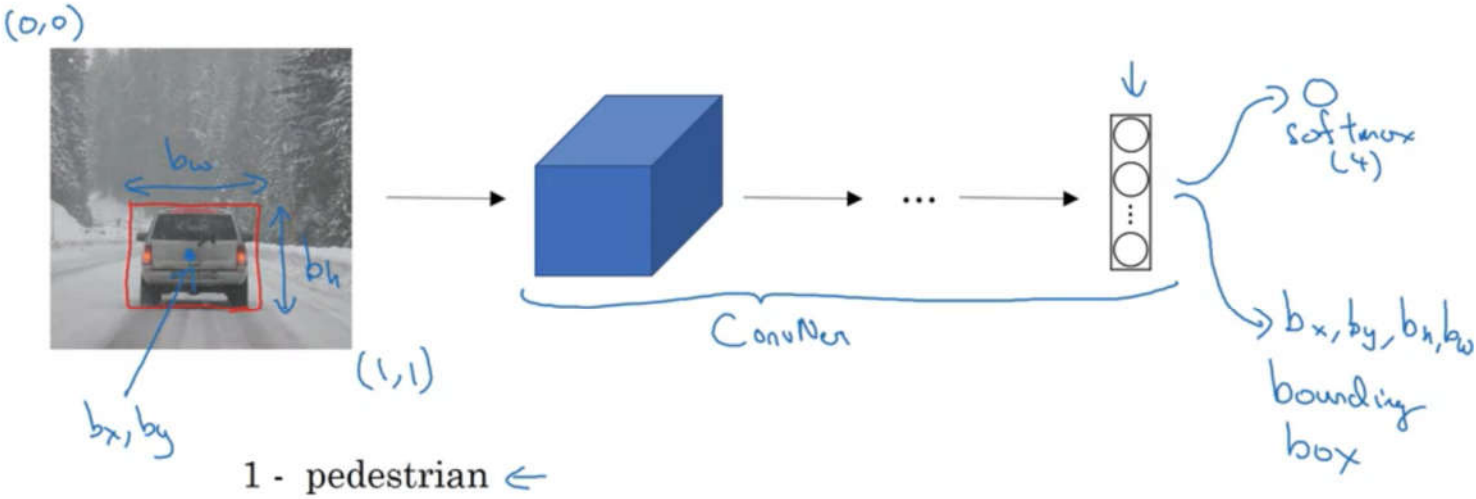
Classification with localization



1 - pedestrian
2 - car
3 - motorcycle
4 - background by as well as the height, that would be bh,

Andrew Ng

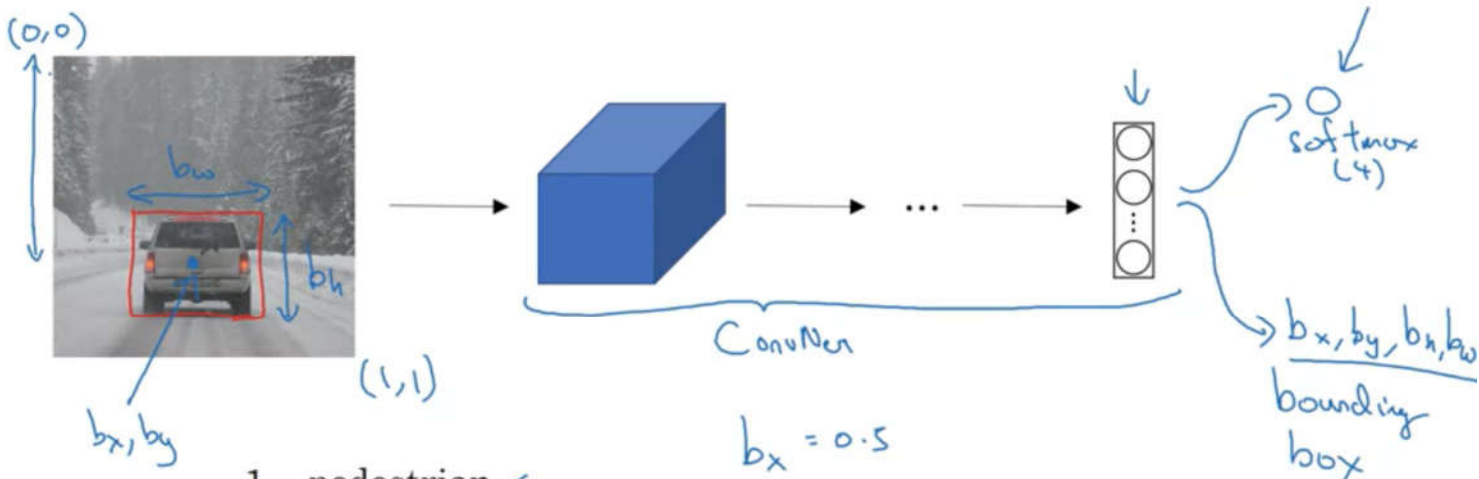
Classification with localization



1 - pedestrian
2 - car
3 - motorcycle
4 - backg So now if your training set contains not just the object cross label,

Andrew Ng

Classification with localization



$b_x = 0.5$

$b_y = 0.7$

1 - pedestrian ←

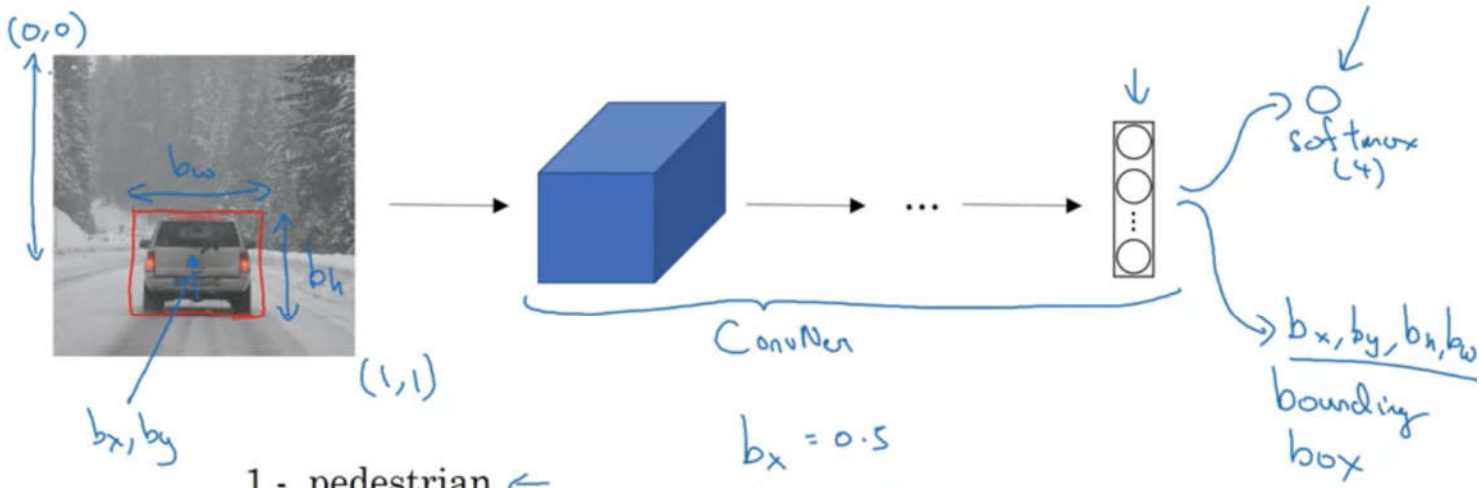
2 - car ←

3 - motorcycle ←

4 - back by might be about 0.7 since it's about maybe 70% to the way down to the image.

Andrew Ng

Classification with localization



$b_x = 0.5$

$b_y = 0.7$

1 - pedestrian ←

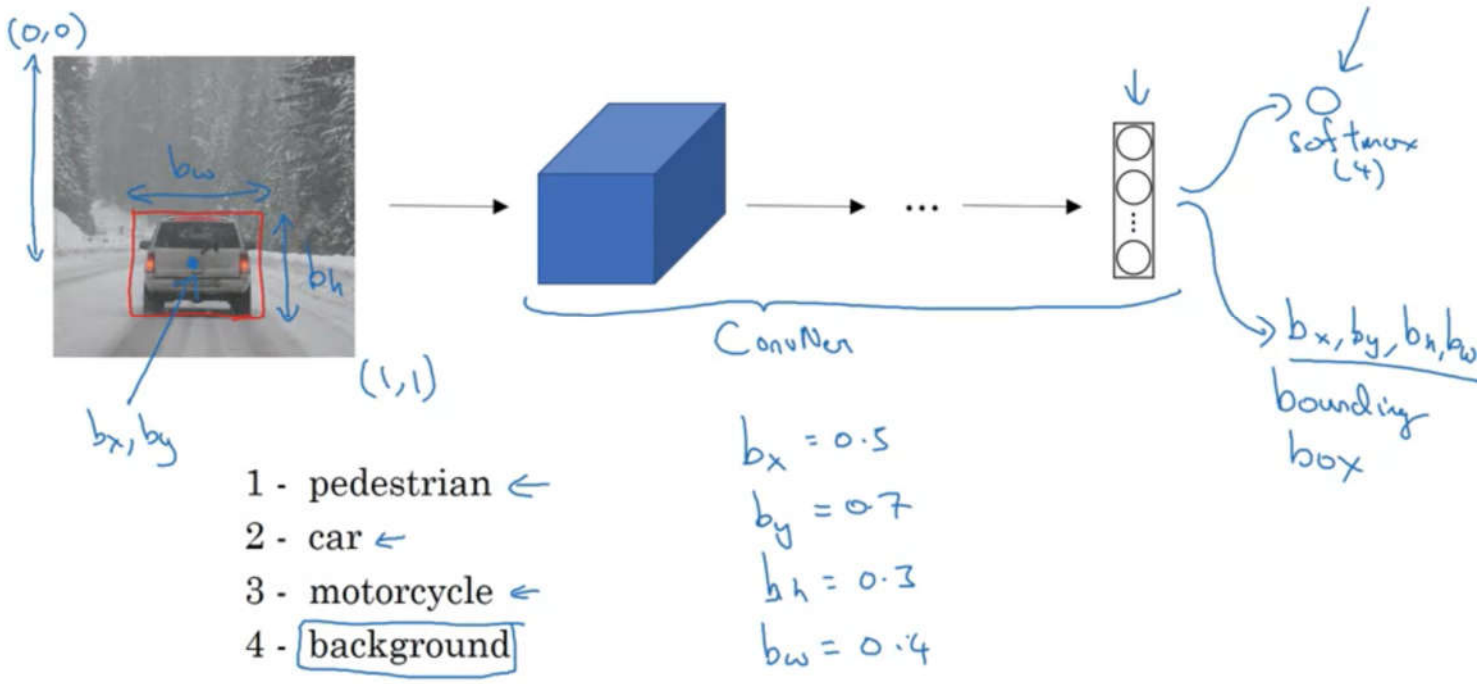
2 - car ←

3 - motorcycle ←

4 - back by might be about 0.7 since it's about maybe 70% to the way down to the image.

Andrew Ng

Classification with localization



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)

y for this as a supervised learning task.

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)

So just as a reminder these are our four classes, and

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)

So, let's define the target label y as follows.

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)

$y = \begin{bmatrix} p_c \\ \\ \\ \end{bmatrix}$ is there an object?

So, if the object is, classes 1, 2 or 3, pc will be equal to 1.

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)

$y = \begin{bmatrix} p_c \\ \\ \\ \end{bmatrix}$ is there an object?

And if it's the background class, so

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)

$$y = \begin{bmatrix} p_c \\ \\ \\ \\ \end{bmatrix}$$

is there an object?

if it's none of the objects you're trying to detect, then pc will be 0.

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)

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

is there an object?

c3 which tells us is it the class 1, class 2 or class 3.

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)

$x =$

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

is there an object?

If this is a training set image, so if that is x, then y will be

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)

$x =$

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

is there an object?

← "don't care"

which of the three objects, c1, c2, c3 it thinks it is.

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)

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

is there an object?

"don't care"

So given a set of label training examples, this is how you will construct x,

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)

$\mathcal{L}(\hat{y}$

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

is there an object?

"don't care"

Finally, next let's describe the loss function

Andrew Ng

Defining the target label y

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

$\mathcal{L}(\hat{y}, y)$

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

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

is there an object?

← "don't care"

(x, y)

you use to train the neural network.

Andrew Ng

Defining the target label y

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

$\mathcal{L}(\hat{y}, y)$

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

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

is there an object?

← "don't care"

So the ground true label was y and the neural network outputs some yhat.

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)

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

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ \vdots \end{bmatrix} \quad \text{is there an object?} \qquad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ \vdots \end{bmatrix} \qquad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \leftarrow \text{"don't care"}$$

So that goes from sum of the squares of the difference of the elements.

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)

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

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ \vdots \end{bmatrix} \quad \text{is there an object?} \qquad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ \vdots \end{bmatrix} \qquad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \leftarrow \text{"don't care"}$$

because in that second case, all of the rest of the components are don't care us.

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)

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

$$x = \quad y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ \vdots \end{bmatrix} \text{ is there an object?} \qquad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ \vdots \end{bmatrix} \qquad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \leftarrow \text{"don't care"}$$

Whereas if y1 = 0, then the second to the eighth components I don't care.

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)

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

$$x = \quad y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ \vdots \end{bmatrix} \text{ is there an object?} \qquad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ \vdots \end{bmatrix} \qquad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \leftarrow \text{"don't care"}$$

p_c

(x, y)

estimating y1, which is equal to pc.

Andrew Ng

Press Esc to exit full screen
deeplearning.ai
Object Detection
Landmark detection
In the previous video,
0:00 / 5:56

Landmark detection

b_x, b_y, b_h, b_w

Let's say you're building a face recognition application and for some reason,

Andrew Ng

Landmark detection

b_x, b_y, b_h, b_w

you want the algorithm to tell you where is the corner of someone's eye.

Andrew Ng

Landmark detection

b_x, b_y, b_h, b_w

$l_{1x}, l_{1y},$
$l_{2x}, l_{2y},$
$l_{3x}, l_{3y},$

the estimated position of all those four points of the person's face.

Andrew Ng

Landmark detection

b_x, b_y, b_h, b_w

$l_{1x}, l_{1y},$
$l_{2x}, l_{2y},$
$l_{3x}, l_{3y},$

and generating a label training sets that contains all of these landmarks,

l_{64}, l_{64y}

Andrew Ng

Landmark detection

b_x, b_y, b_h, b_w

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

have it go through a convnet and have a convnet,

Andrew Ng

Landmark detection

b_x, b_y, b_h, b_w

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

that's sixty-four times two,

Andrew Ng

Landmark detection

b_x, b_y, b_h, b_w

We have a set of images as well as labels Y where people,

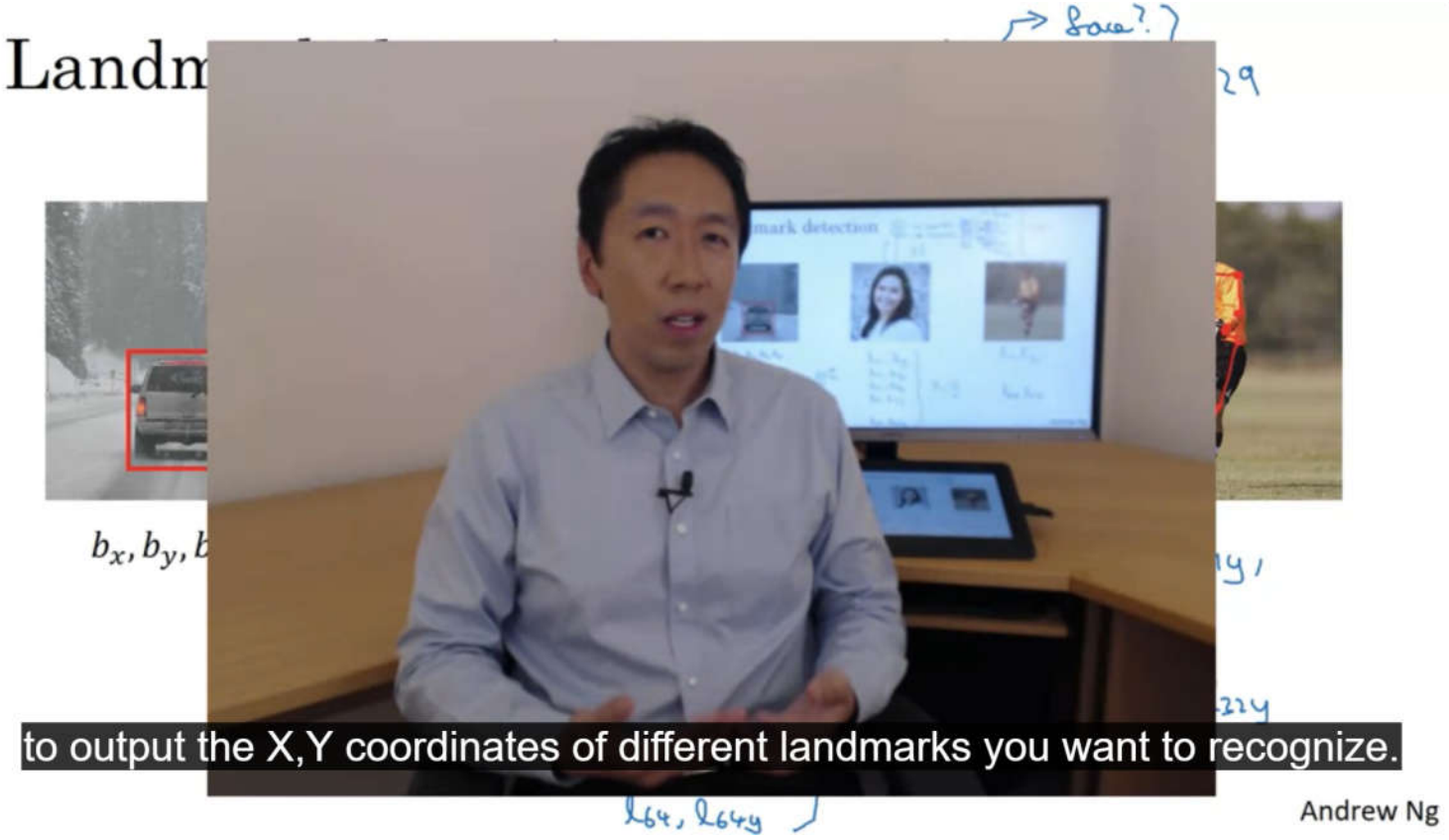
Andrew Ng

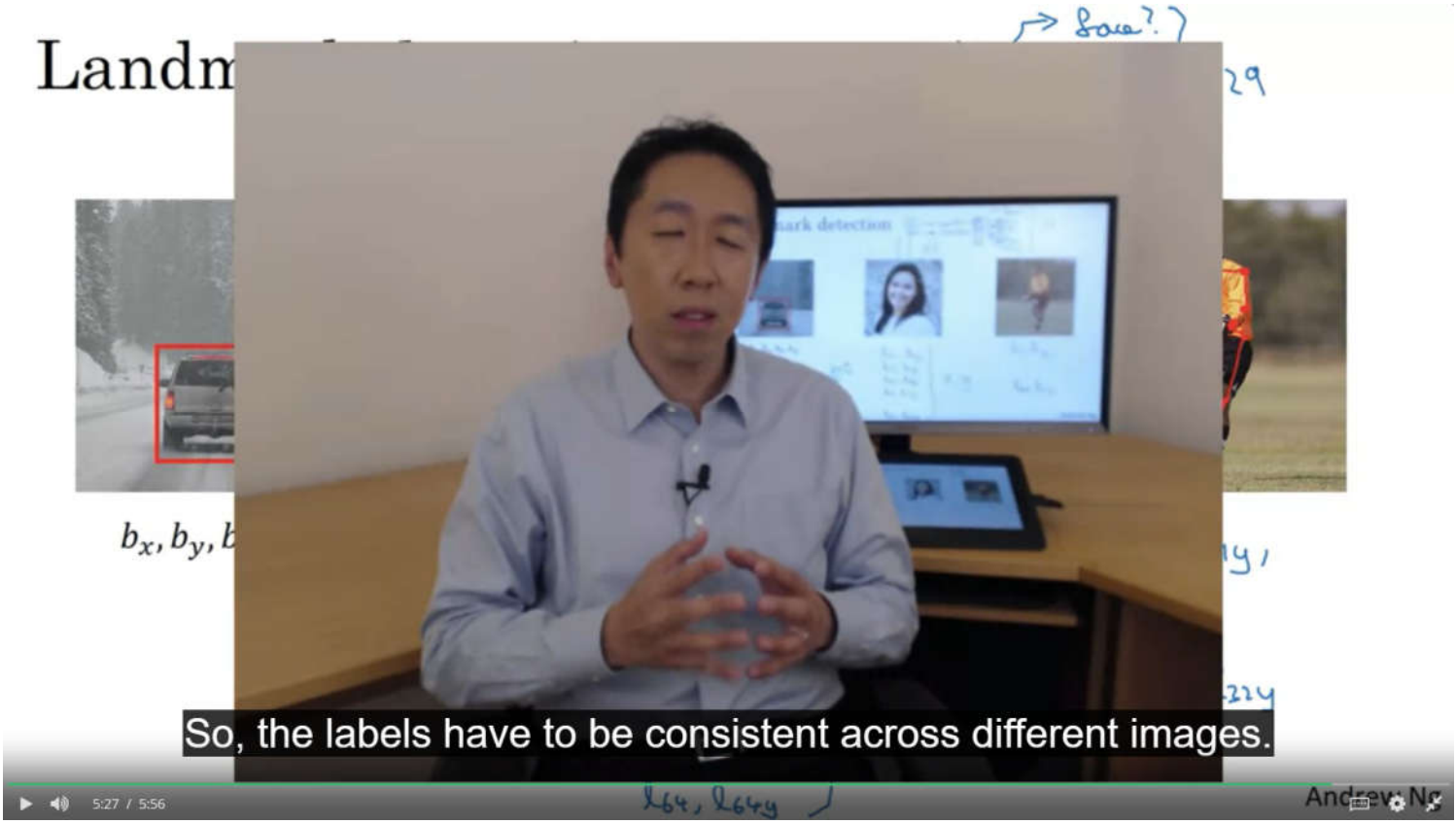
Landmark detection

b_x, b_y, b_h, b_w

you could also define a few key positions like the midpoint of the chest,

Andrew Ng


Landm
b_x, b_y, b
to output the X,Y coordinates of different landmarks you want to recognize.
Andrew Ng


Landm
b_x, b_y, b
So, the labels have to be consistent across different images.
5:27 / 5:56
Andrew Ng

deeplearning.ai
Object Detection
Object detection
You've learned about Object Localization as well as Landmark Detection.
0:00 / 5:48

deeplearning.ai
Object Detection
Object detection
object detection using something called the Sliding Windows Detection Algorithm.

Car detection example

Let's say you want to build a car detection algorithm.

Andrew Ng

Car detection example

Training set:

x	y
	1
	1

So, this is image x has a positive example, there's a car,

Andrew Ng

Car detection example

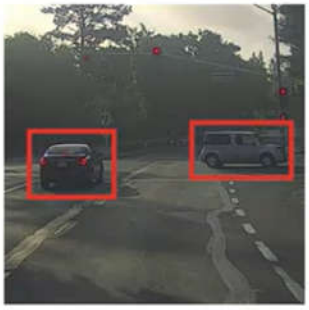
Training set:

x	y
	1
	1
	1
	0

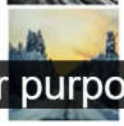
and then there's not a car, there's not a car.

Andrew Ng

Car detection example



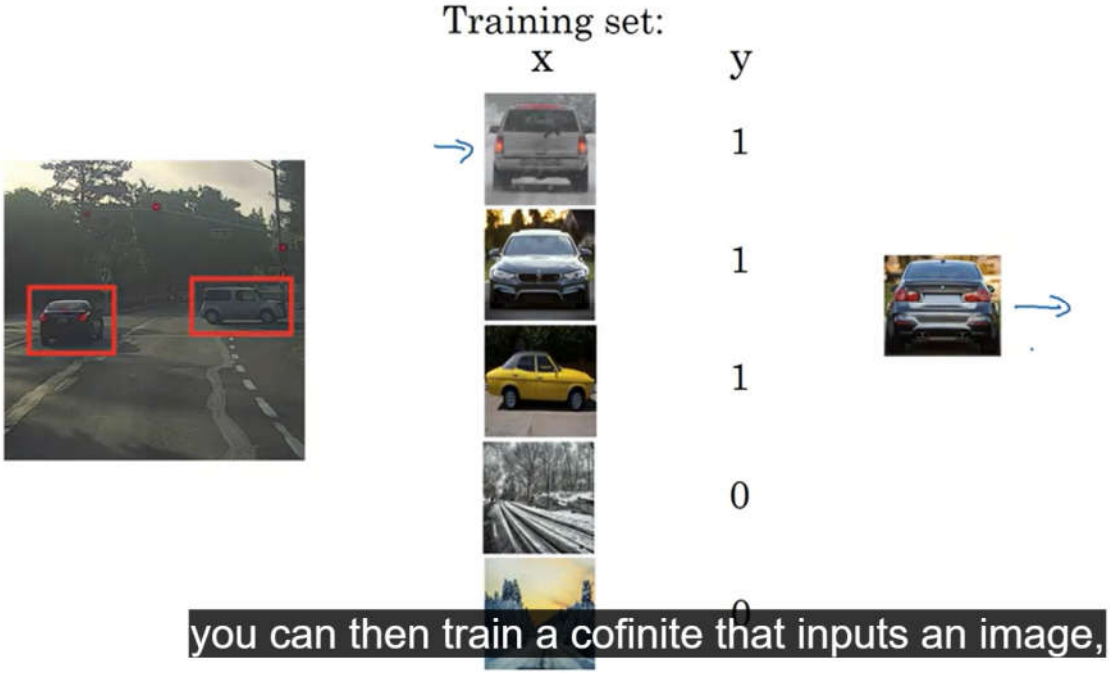
Training set:

x	y
	1
	1
	1
	0
	0

And for our purposes in this training set,

Andrew Ng

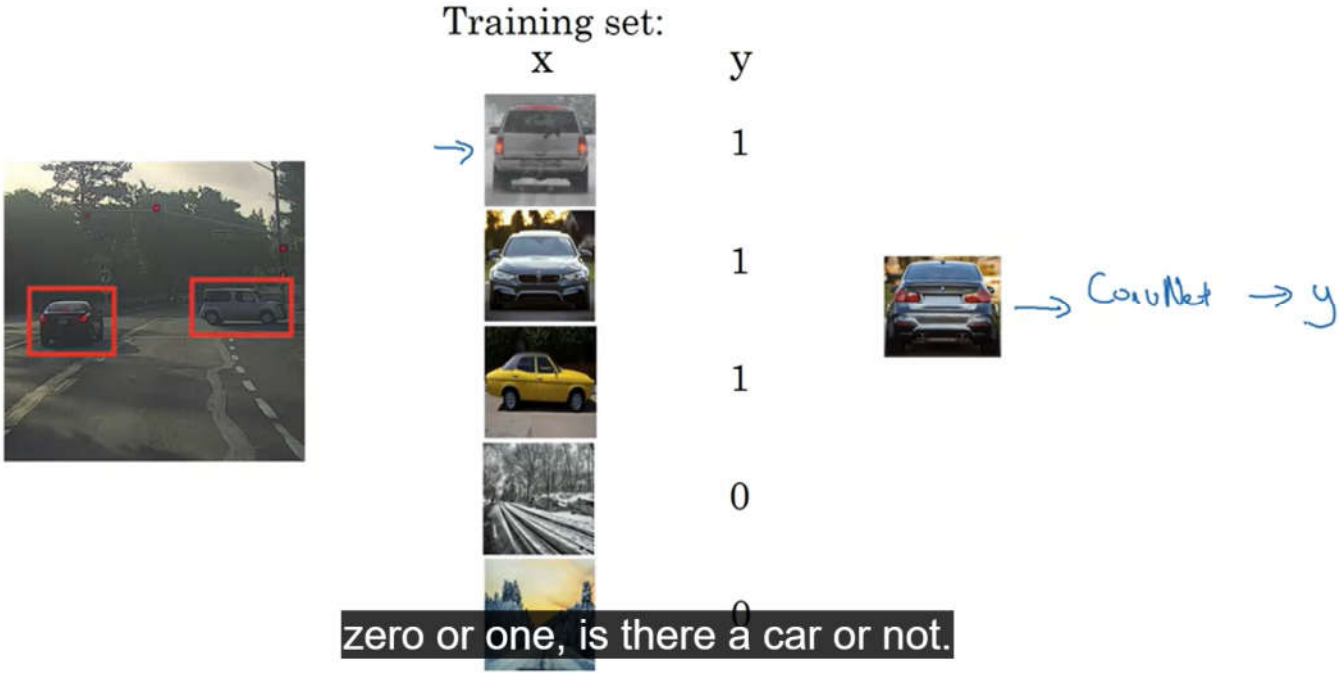
Car detection example



you can then train a cofinite that inputs an image,

Andrew Ng

Car detection example



zero or one, is there a car or not.

Andrew Ng

Sliding windows detection

if you have a test image like this what you do is you

Andrew Ng

Sliding windows detection

start by picking a certain window size, shown down there.

Andrew Ng

Sliding windows detection

And then you would input into this cofinite a small rectangular region.

Andrew Ng

Sliding windows detection

→ ConvNet → 0

So, you're feeding just the region of the image

Andrew Ng

Sliding windows detection

And you keep going until you've slid the window across every position in the image.

Andrew Ng

Sliding windows detection

So, now you take a slightly larger region and run that region.

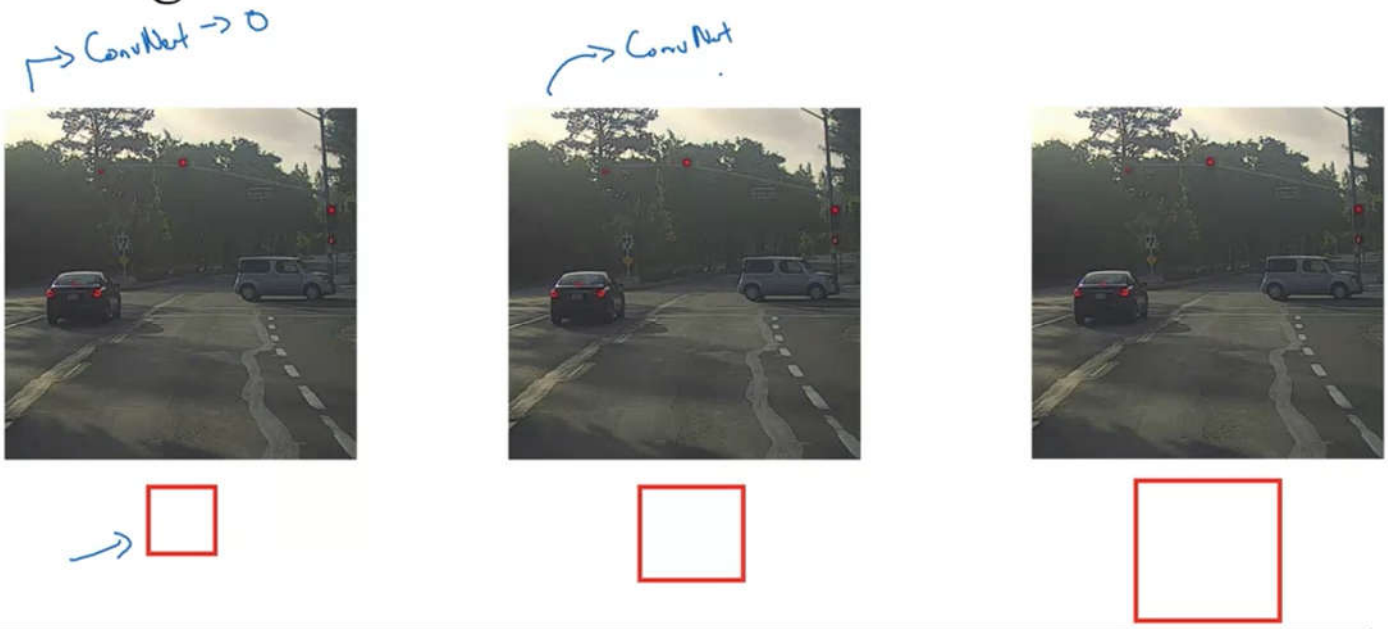
Andrew Ng

Sliding windows detection

And you run that throughout your entire image until you get to the end.

Andrew Ng

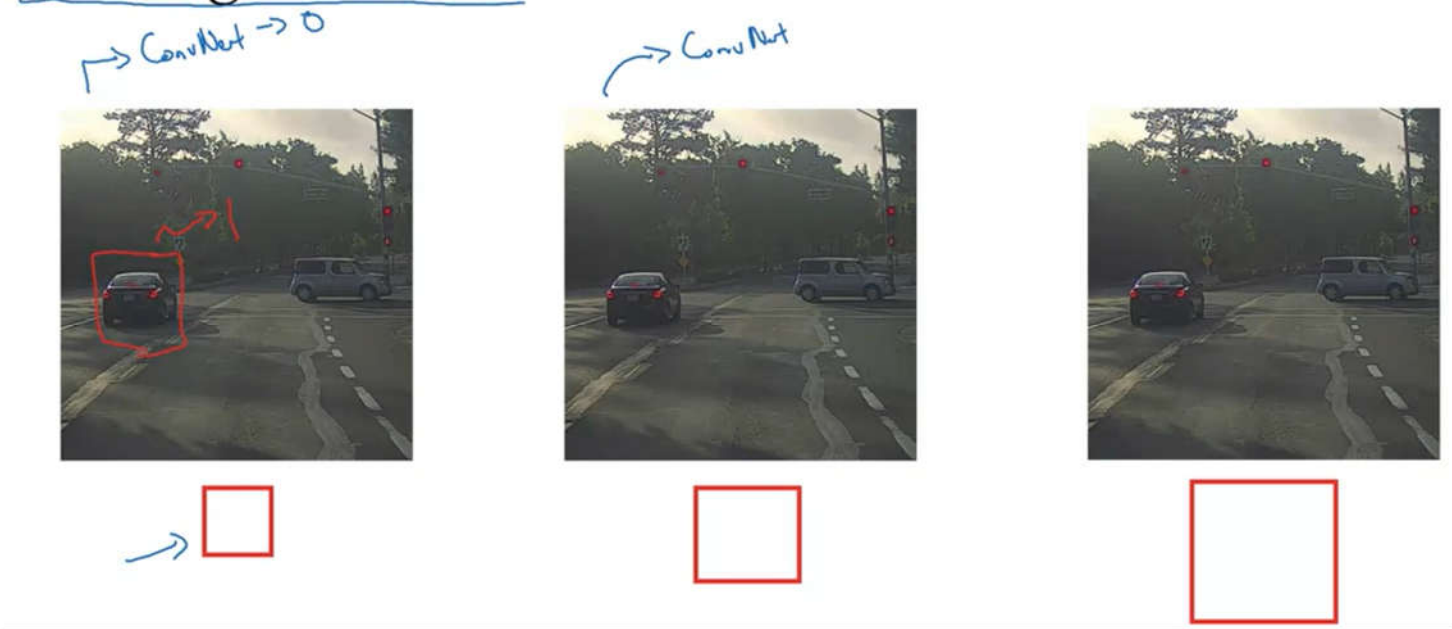
Sliding windows detection



And then you might do the third time using even larger windows and so on.

Andrew Ng

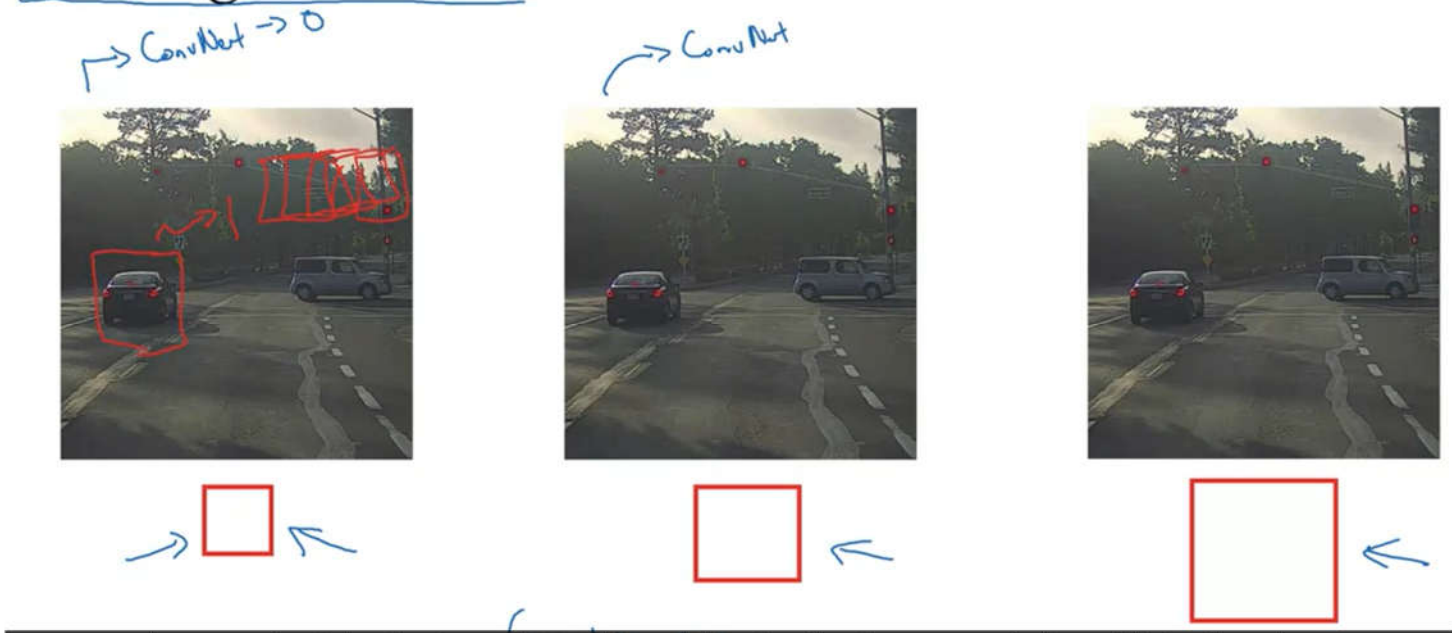
Sliding windows detection



So this algorithm is called Sliding Windows Detection because you take these windows,

Andrew Ng

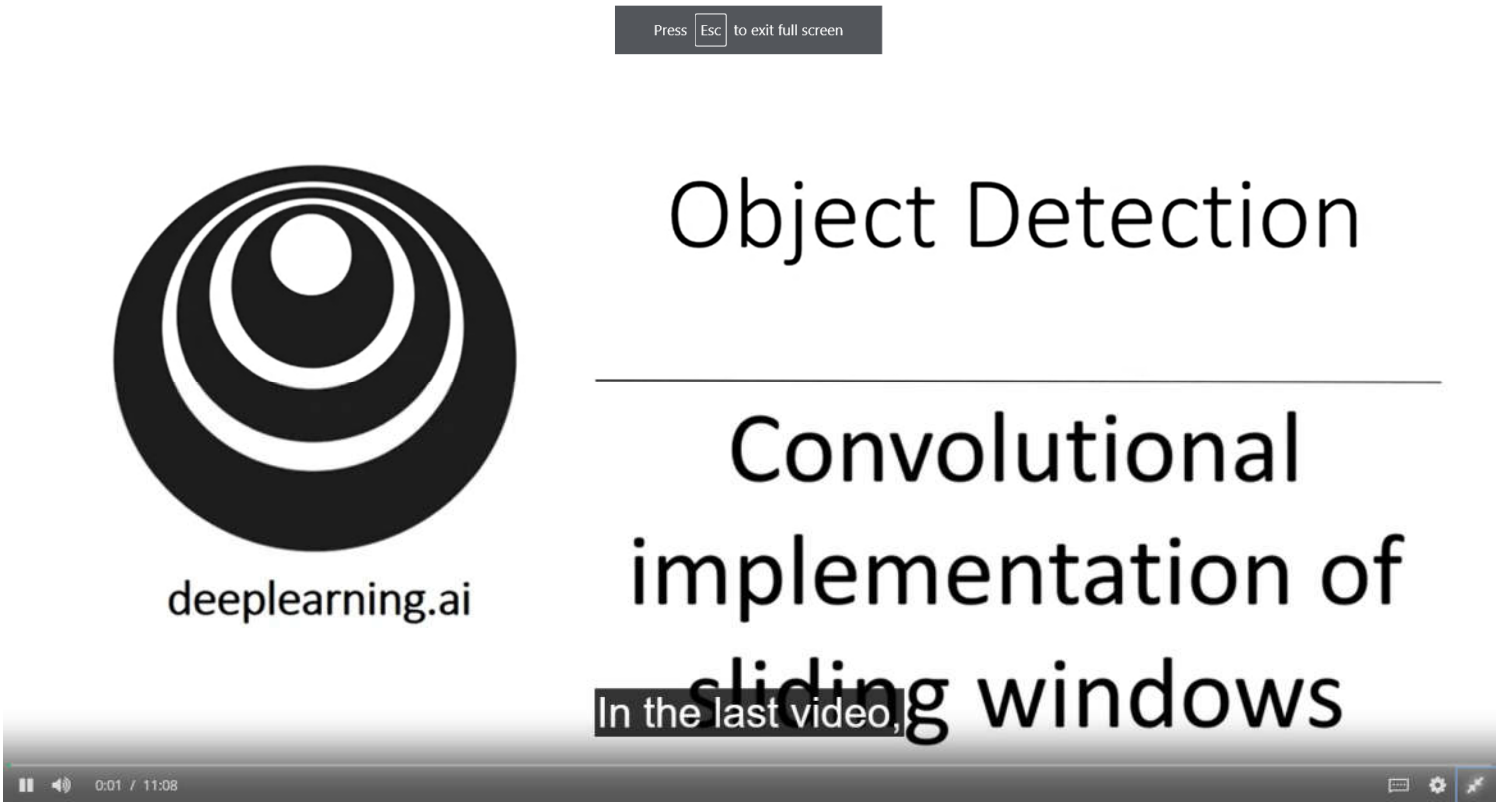
Sliding windows detection



you end up not able to localize the objects that accurately within the image as well.

5:26 / 5:48

Andrew Ng



Turning FC layer into convolutional layers

To build up towards the convolutional implementation of sliding windows let's first see

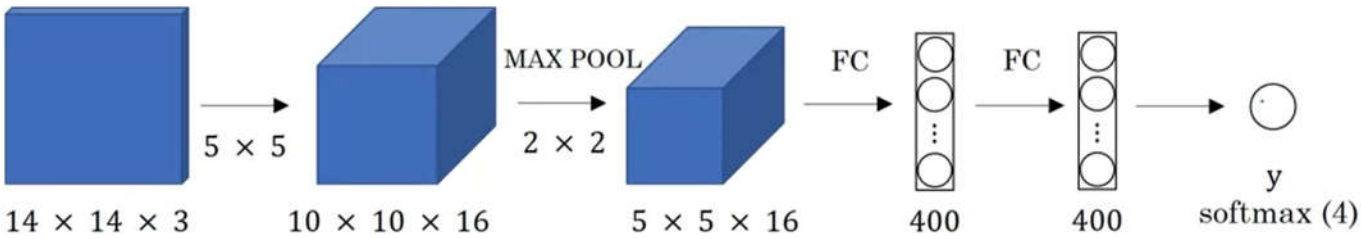
Andrew Ng

Turning FC layer into convolutional layers

how you can turn fully connected layers in neural network into convolutional layers.

Andrew Ng

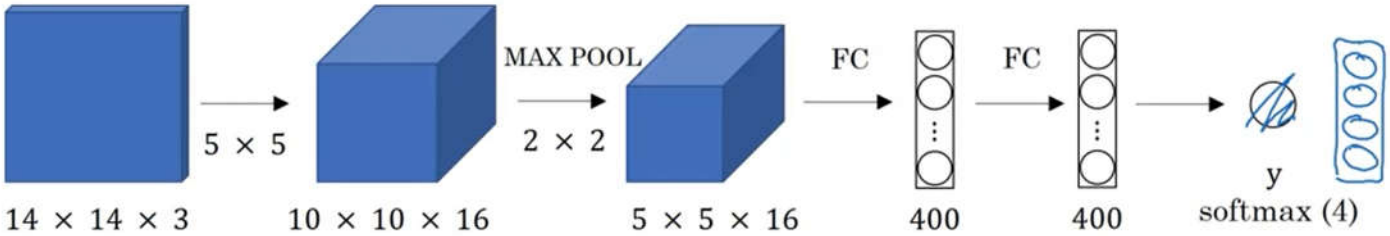
Turning FC layer into convolutional layers



In order to make the change we'll need to in a second,

Andrew Ng

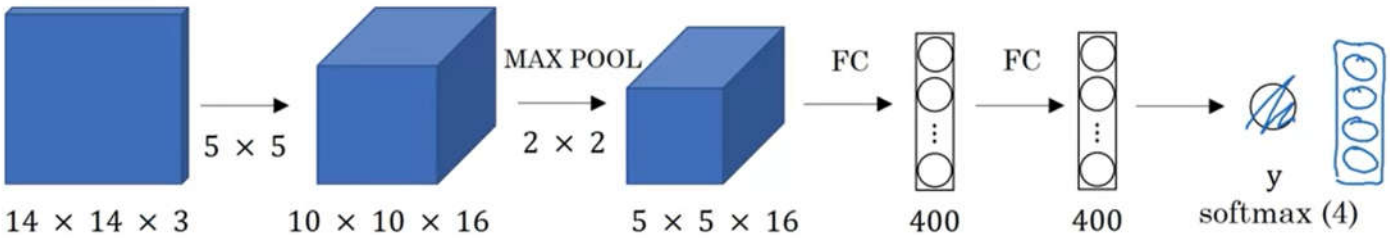
Turning FC layer into convolutional layers



corresponding to the cause probabilities of

Andrew Ng

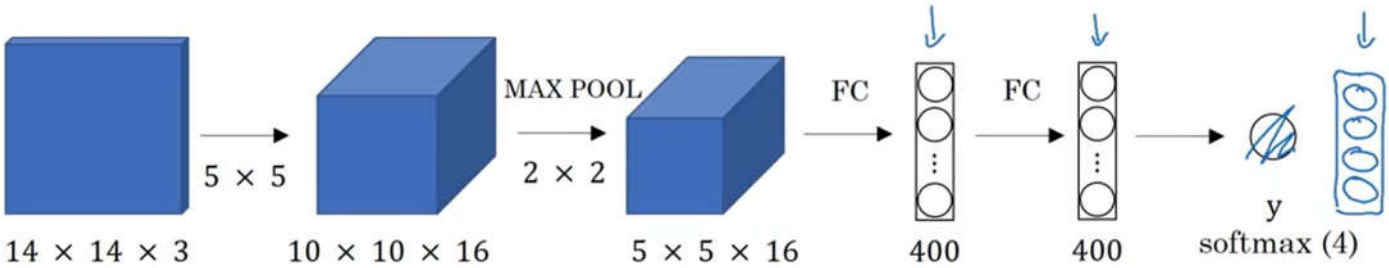
Turning FC layer into convolutional layers

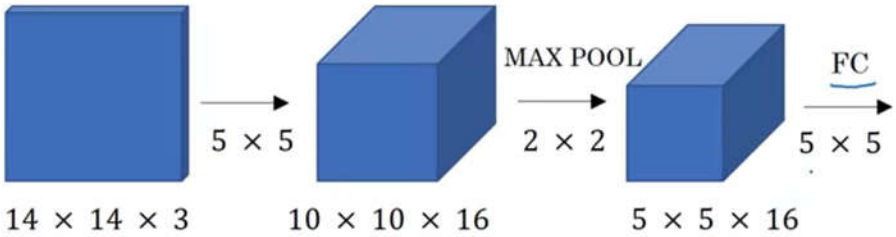


the four causes that softmax units is classified amongst.

Andrew Ng

Turning FC layer into convolutional layers

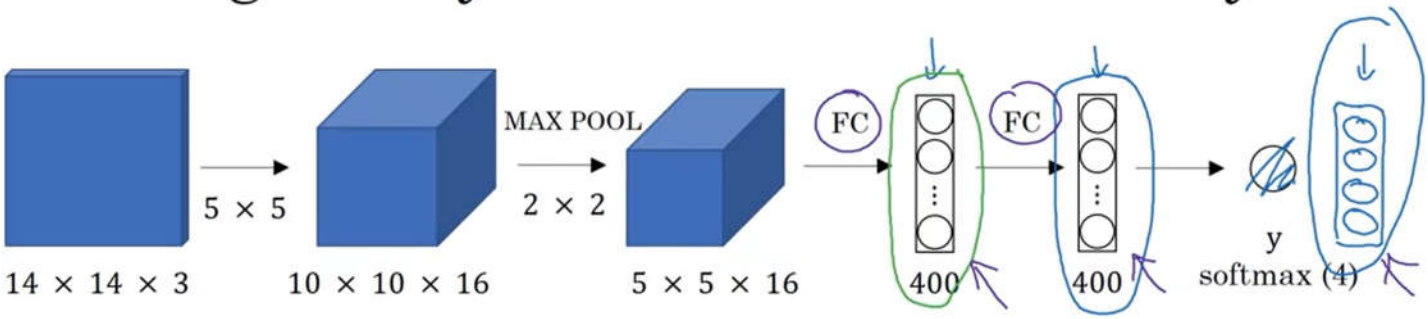


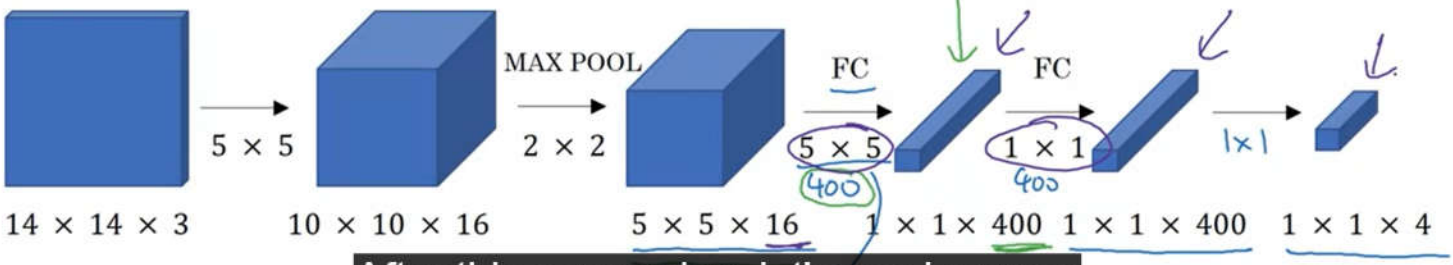


5 filter and let's use 400 5 by 5 filters.

Andrew Ng

Turning FC layer into convolutional layers





After this conversion, let's see how you

Andrew Ng

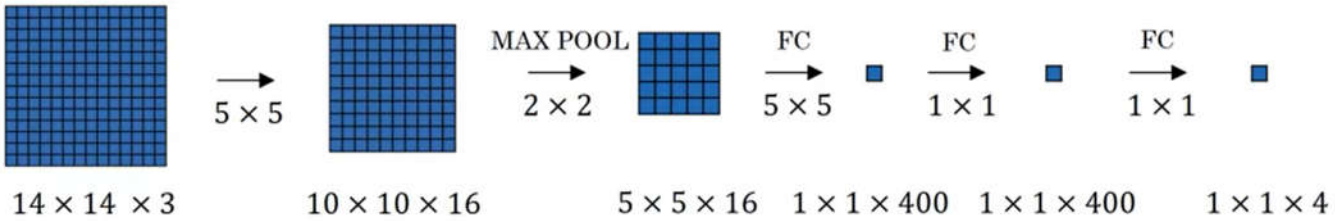
Convolution implementation of sliding windows

Michael Mathieu, Robert Fergus and Yann Lecun.

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

Andrew Ng

Convolution implementation of sliding windows

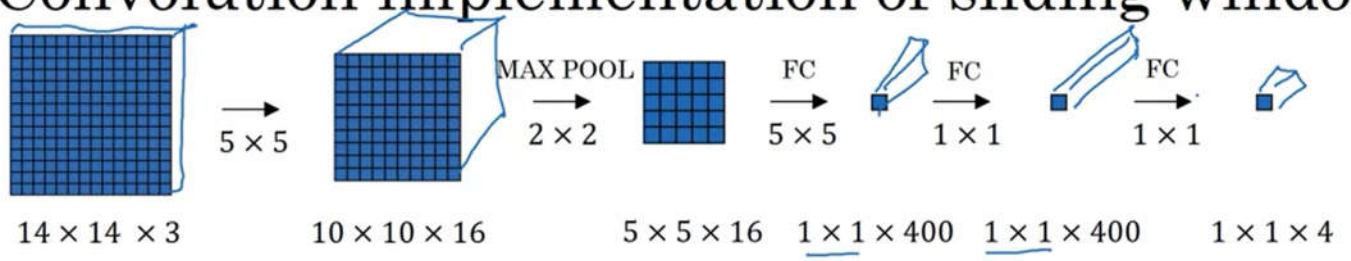


follows that eventually outputs a 1 by 1 by 4 volume,

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

Andrew Ng

Convolution implementation of sliding windows

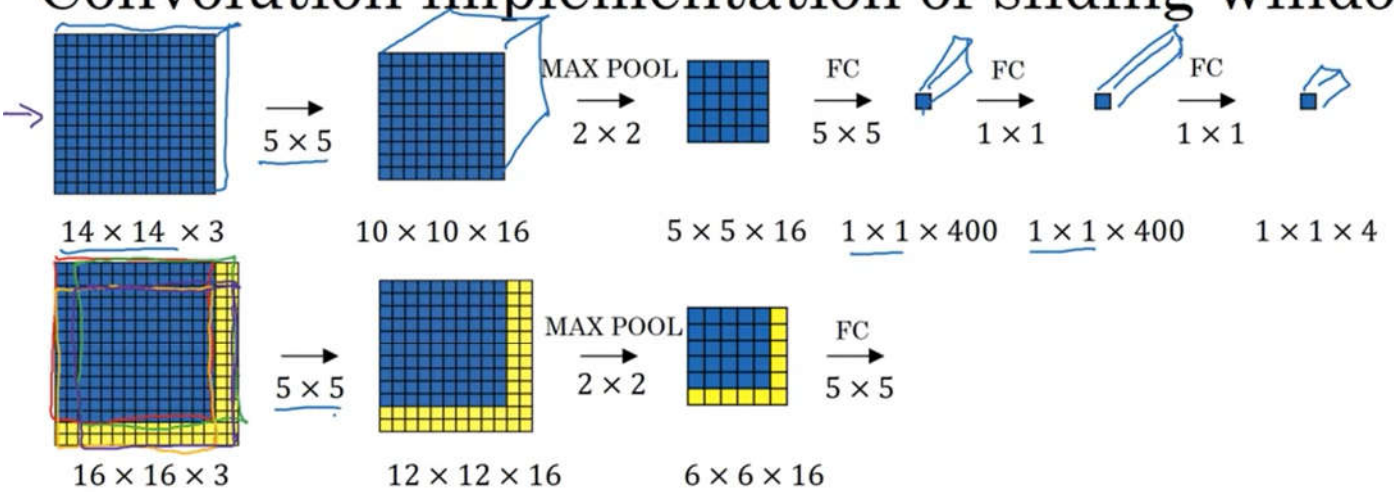


So just dropped the three components of these drawings, just for this slide.

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

Andrew Ng

Convolution implementation of sliding windows

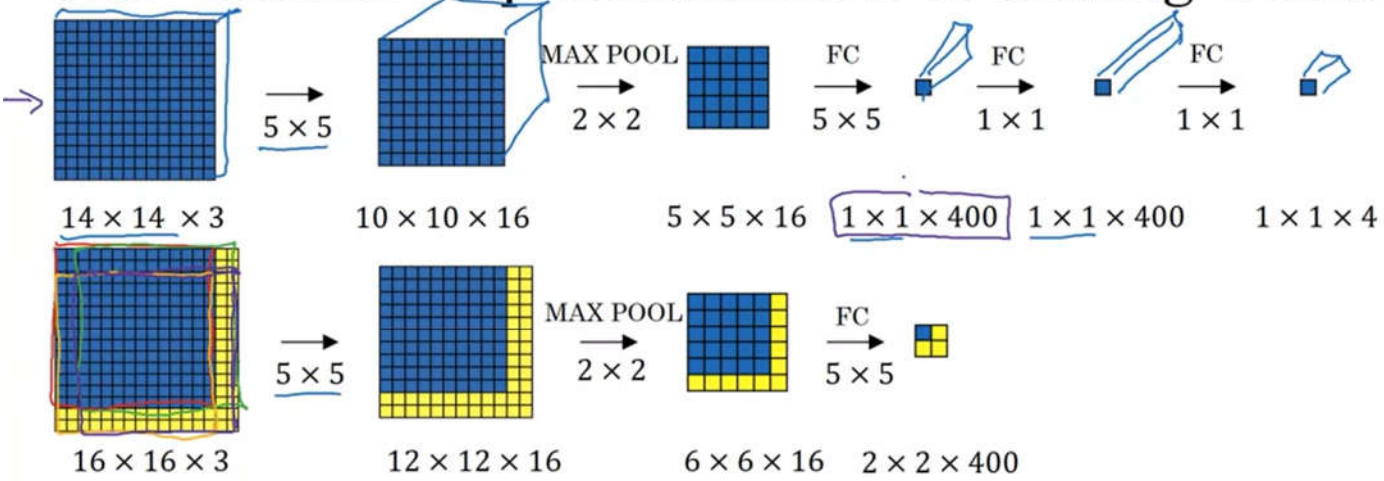


Now you have a 6 by 6 by 16,

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

Andrew Ng

Convolution implementation of sliding windows

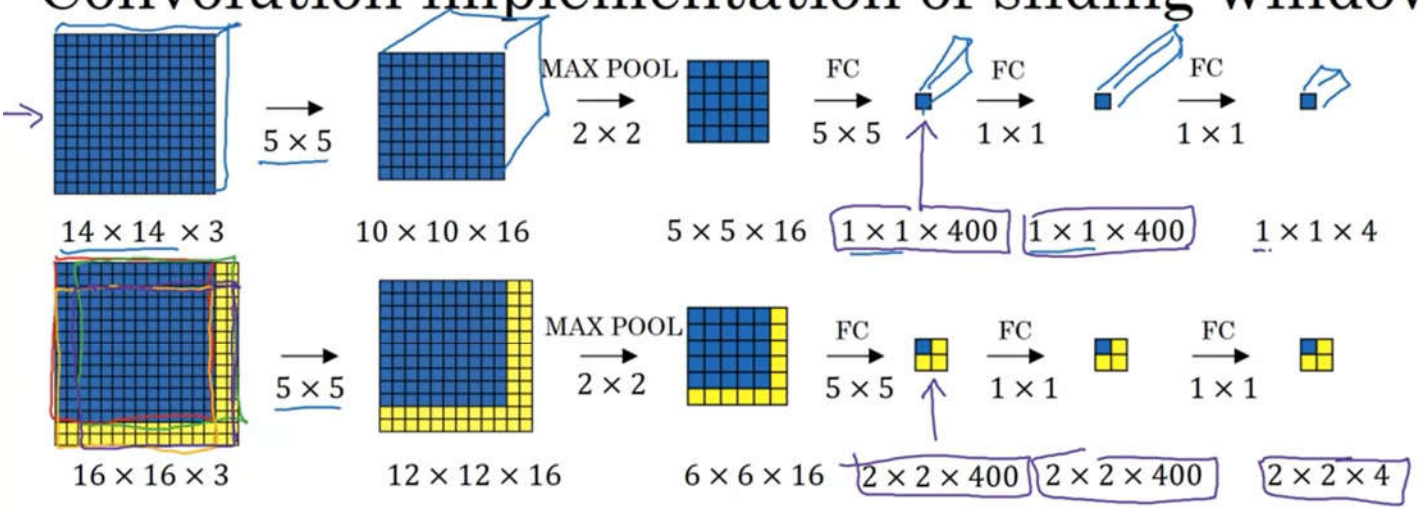


So now instead of a 1 by 1 by 400 volume,

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

Andrew Ng

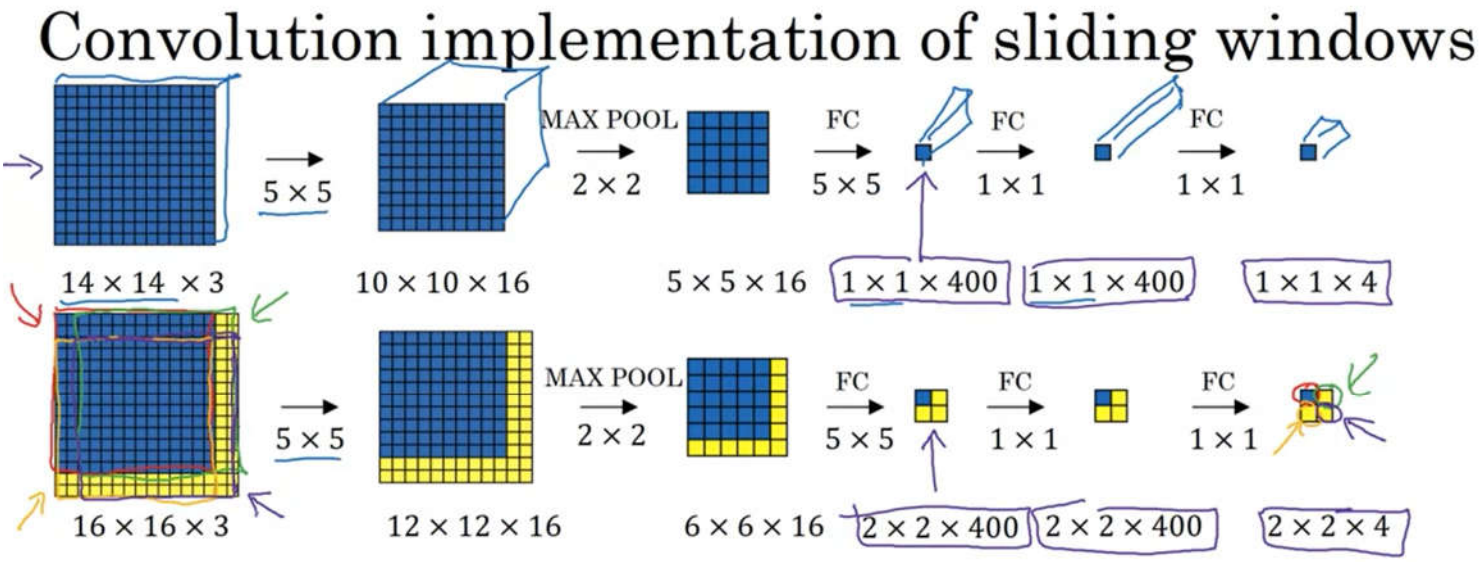
Convolution implementation of sliding windows



2 by 2 by 4 output volume instead of 1 by 1 by 4.

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

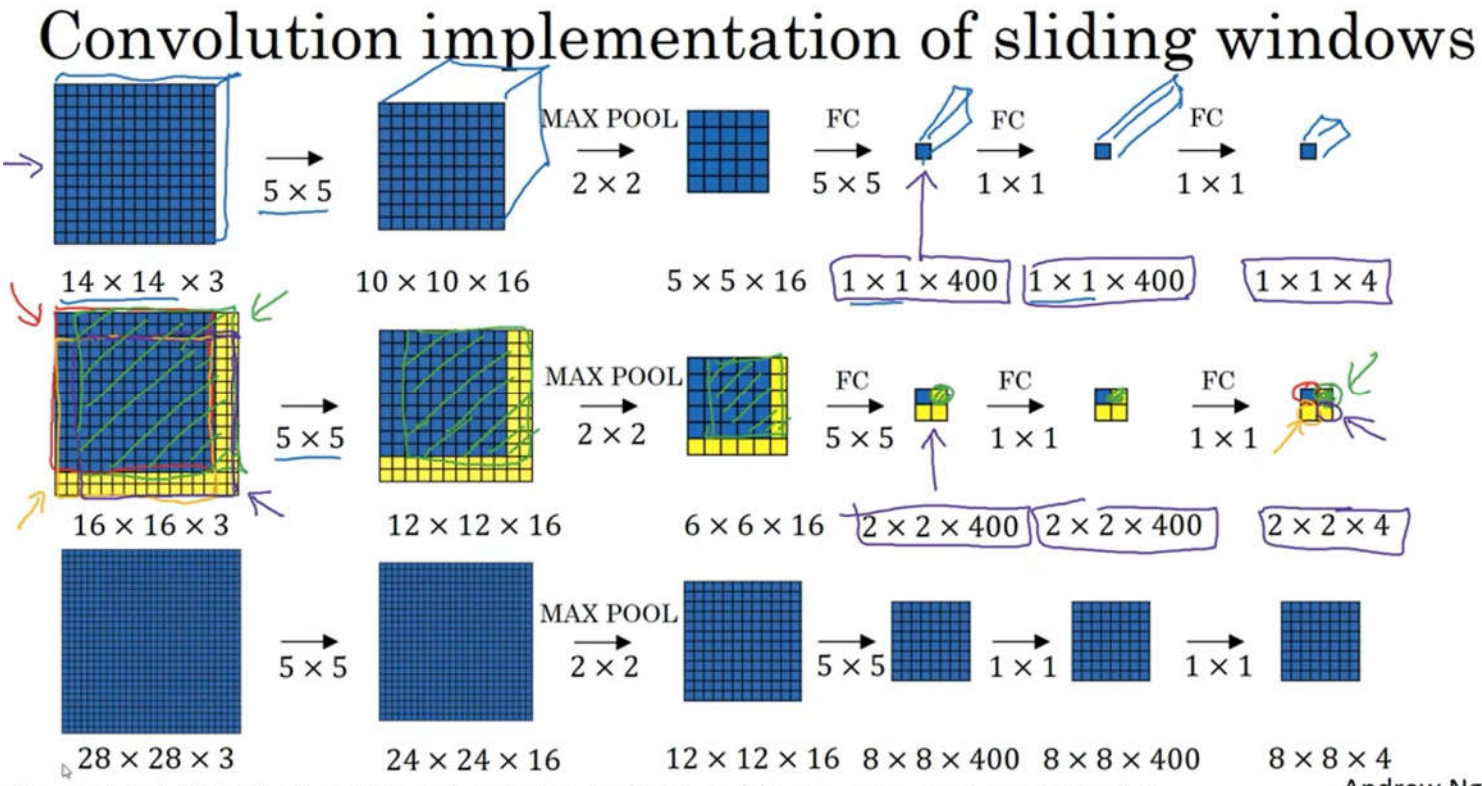
Andrew Ng



And if you step through all the steps of the calculation,

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

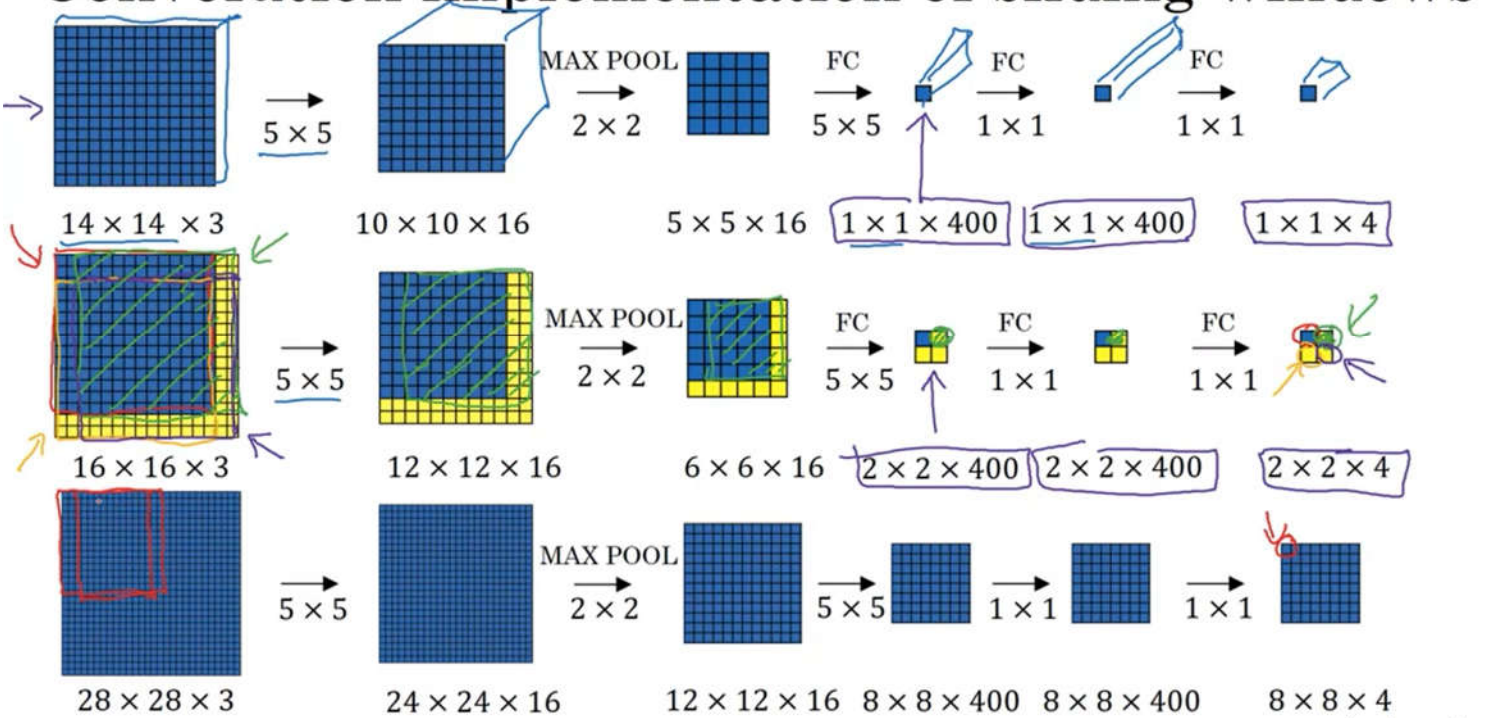
Andrew Ng



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

Andrew Ng

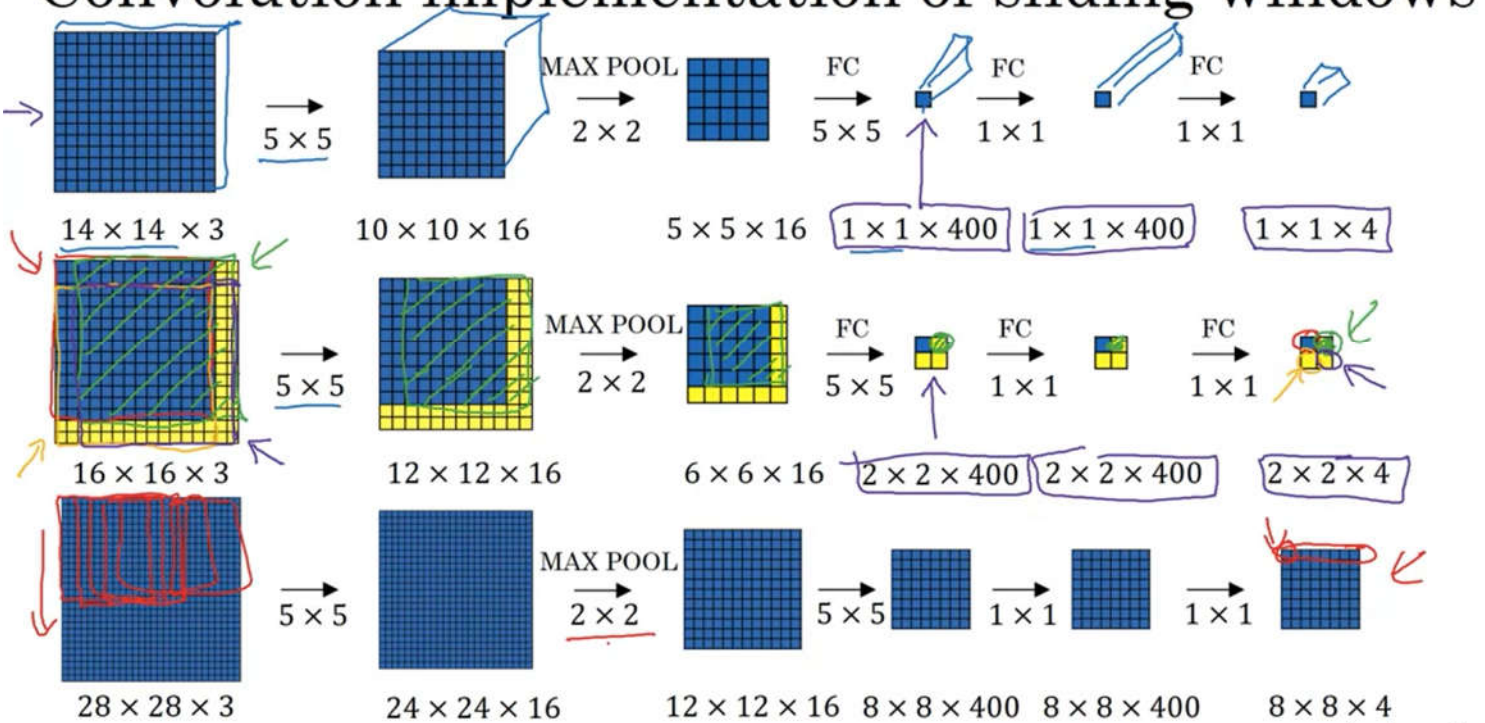
Convolution implementation of sliding windows



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

Andrew Ng

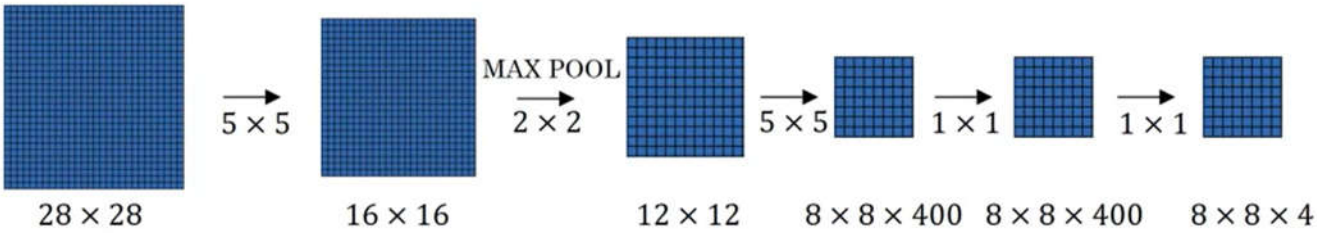
Convolution implementation of sliding windows



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

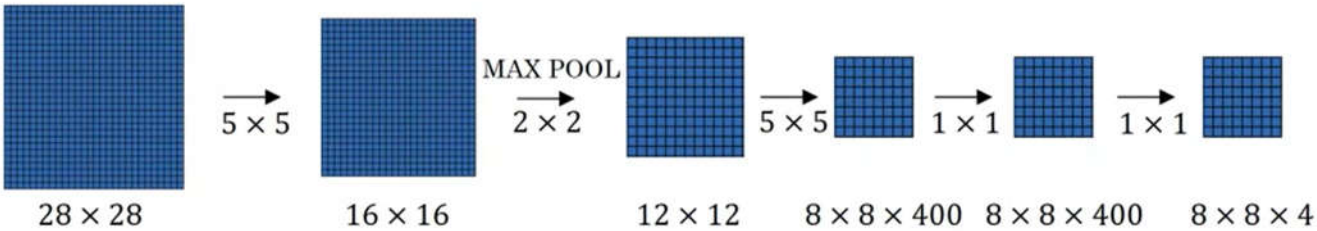
Andrew Ng

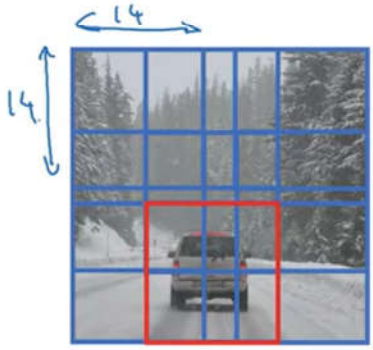
Convolution implementation of sliding windows



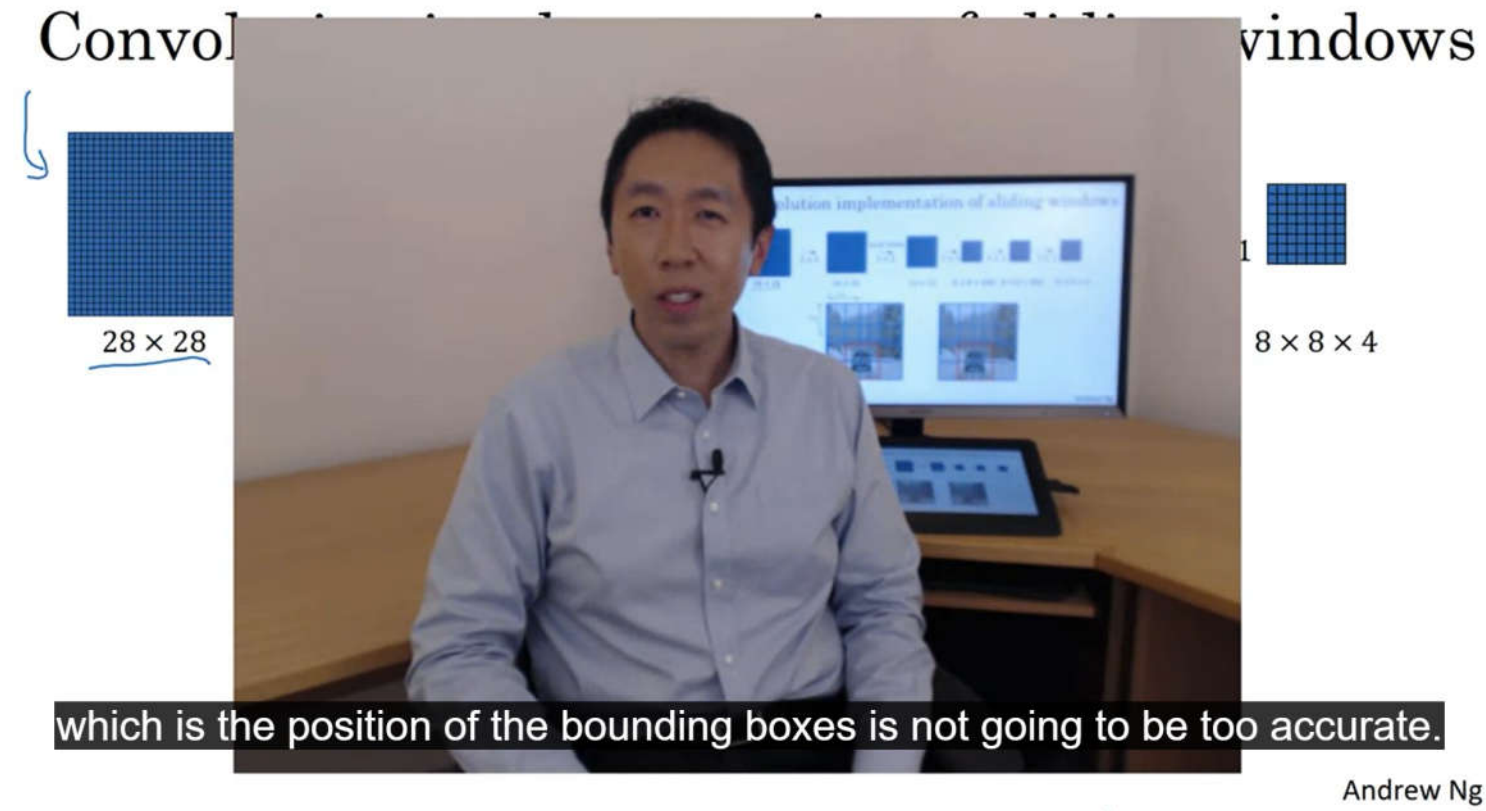
Andrew Ng

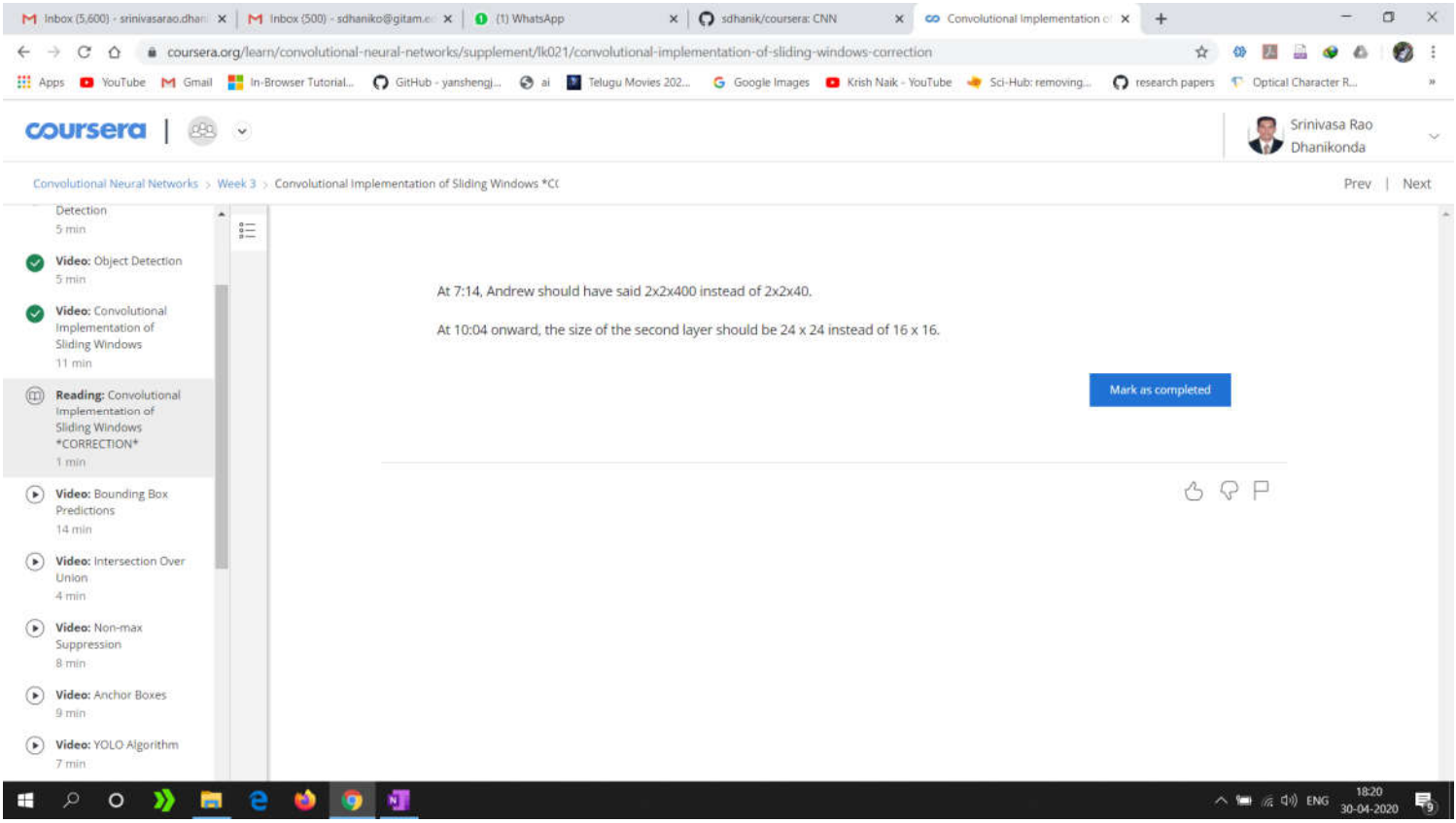
Convolution implementation of sliding windows





Andrew Ng


Convol
vindows
28 × 28
8 × 8 × 4
which is the position of the bounding boxes is not going to be too accurate.
Andrew Ng


coursera.org/learn/convolutional-neural-networks/supplement/lk021/convolutional-implementation-of-sliding-windows-correction
coursera
Srinivasa Rao Dhanikonda
Convolutional Neural Networks > Week 3 > Convolutional Implementation of Sliding Windows *C(
Prev | Next
Detection 5 min
Video: Object Detection 5 min
Video: Convolutional Implementation of Sliding Windows 11 min
Reading: Convolutional Implementation of Sliding Windows *CORRECTION* 1 min
Video: Bounding Box Predictions 14 min
Video: Intersection Over Union 4 min
Video: Non-max Suppression 8 min
Video: Anchor Boxes 9 min
Video: YOLO Algorithm 7 min
At 7:14, Andrew should have said 2x2x400 instead of 2x2x40.
At 10:04 onward, the size of the second layer should be 24 x 24 instead of 16 x 16.
Mark as completed

Object Detection

Bounding box predictions

you learned how to use a convolutional implementation of sliding windows.

Object Detection

Bounding box predictions

but it still has a problem of not quite outputting the most accurate bounding boxes.

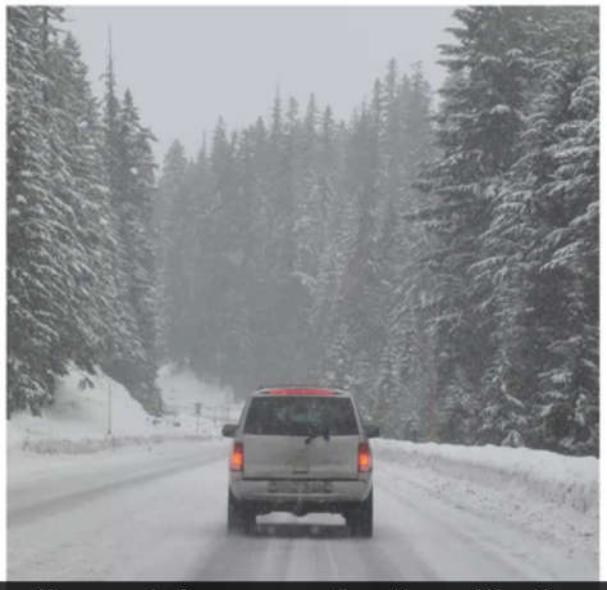
Output accurate bounding boxes

With sliding windows, you take

0:20 / 14:31

Andrew Ng

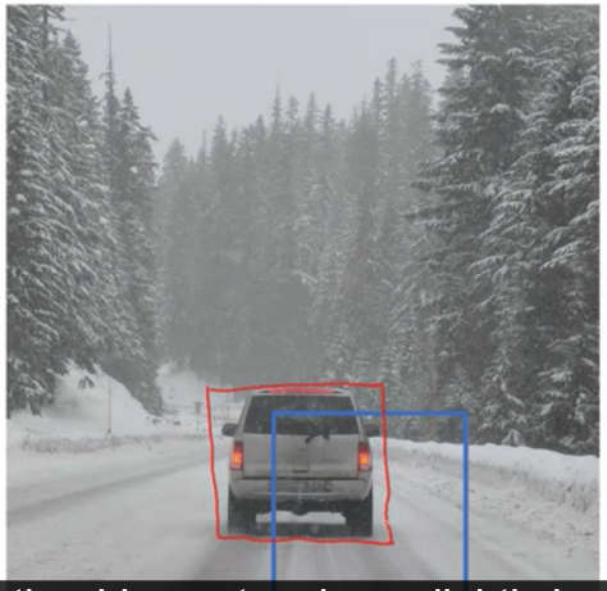
Output accurate bounding boxes



none of the boxes really match up perfectly with the position of the car.

Andrew Ng

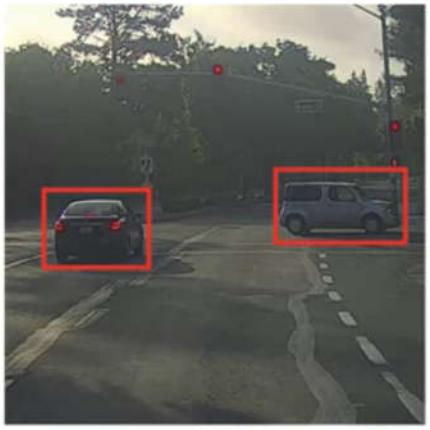
Output accurate bounding boxes



it's actually has a slightly wider rectangle or slightly horizontal aspect ratio.

Andrew Ng

YOLO algorithm



100

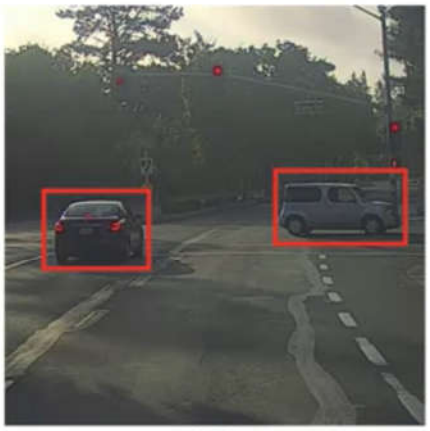
100

A good way to get this output more accurate bounding boxes is with the YOLO algorithm.

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

Andrew Ng

YOLO algorithm



100

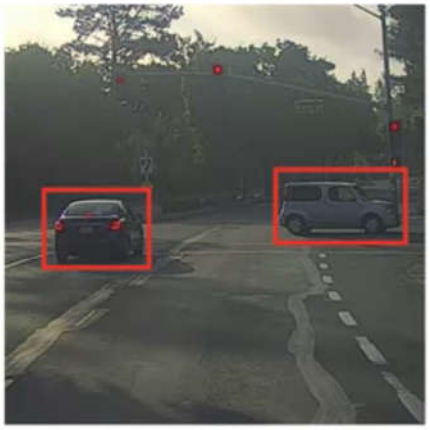
100

YOLO stands for, You Only Look Once.

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

Andrew Ng

YOLO algorithm



100

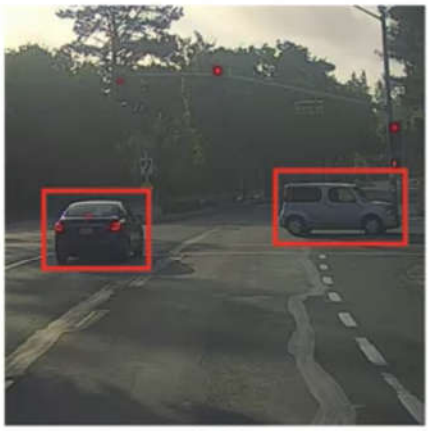
100

Let's say you have an input image at 100 by 100,

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

Andrew Ng

YOLO algorithm

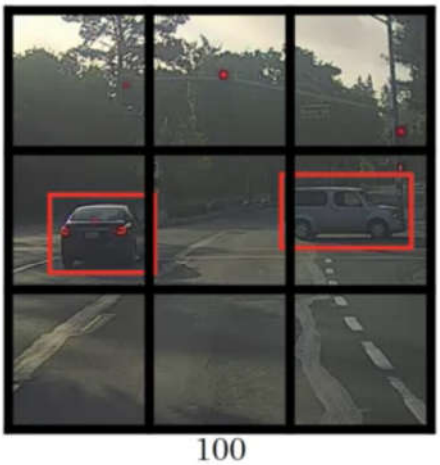


you're going to place down a grid on this image.

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

Andrew Ng

YOLO algorithm

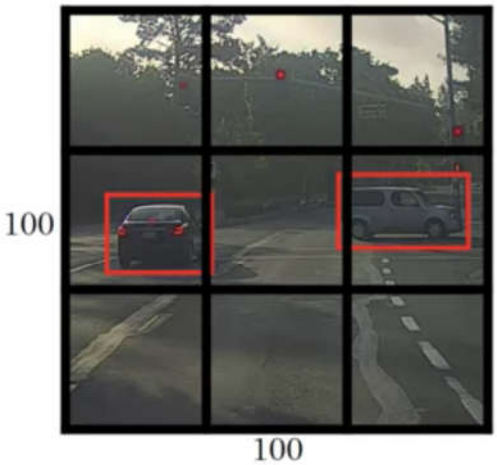


the image classification and localization algorithm that you saw a few videos back.

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

Andrew Ng

YOLO algorithm

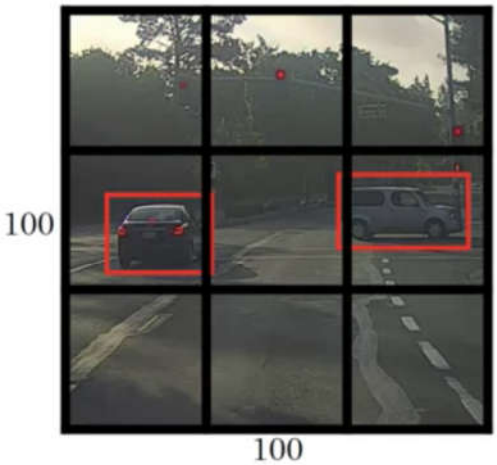


and apply it to each of the nine grids.

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

Andrew Ng

YOLO algorithm

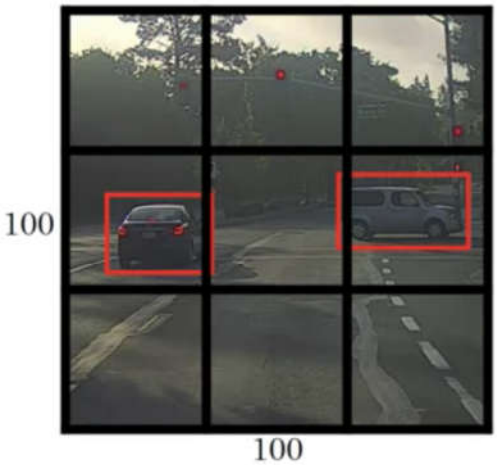


And the basic idea is you're going to take the image classification and

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

Andrew Ng

YOLO algorithm



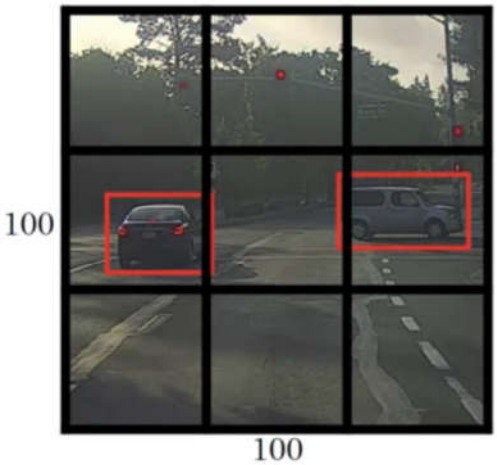
Labels for training

here's how you define the labels you use for training.

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

Andrew Ng

YOLO algorithm



Labels for training

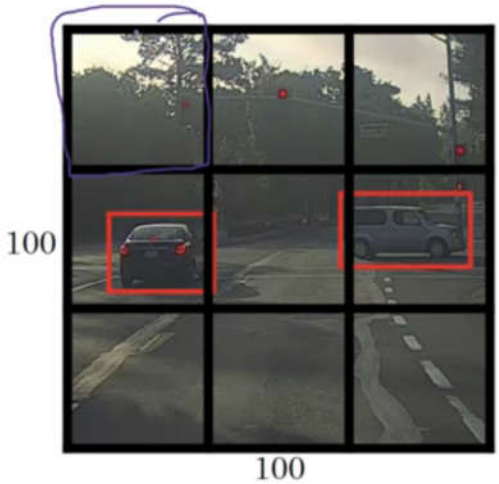
For each grid cell:

So for each of the nine grid cells, you specify a label Y,

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

Andrew Ng

YOLO algorithm



Labels for training

For each grid cell:

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

this one up here.

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

Andrew Ng

YOLO algorithm

Labels for training

For each grid cell:

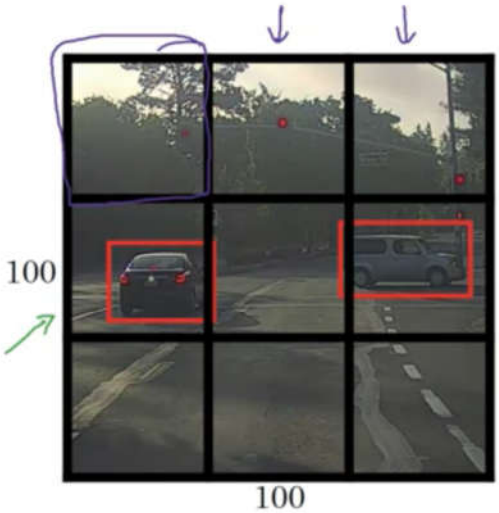
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

So, the label vector Y for the upper left grid cell would be zero,

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

Andrew Ng

YOLO algorithm



Labels for training

For each grid cell:

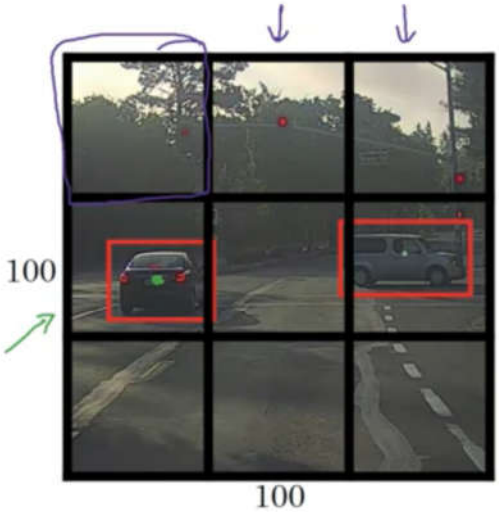
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

this image has two objects.

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

Andrew Ng

YOLO algorithm



Labels for training

For each grid cell:

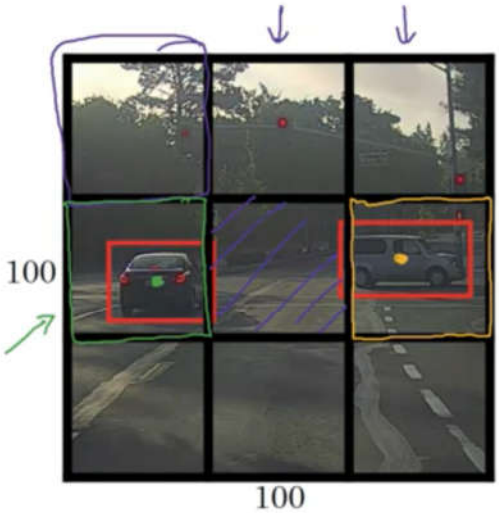
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

And what the YOLO algorithm does is it takes the midpoint of reach of

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

Andrew Ng

YOLO algorithm



Labels for training

For each grid cell:

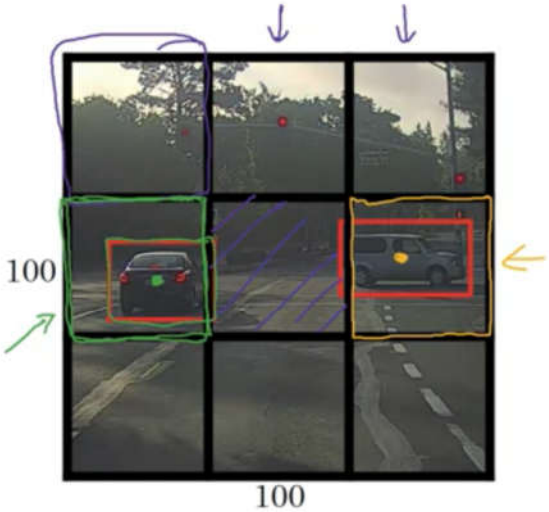
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

and so the first component PC,

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

Andrew Ng

YOLO algorithm



Labels for training

For each grid cell:

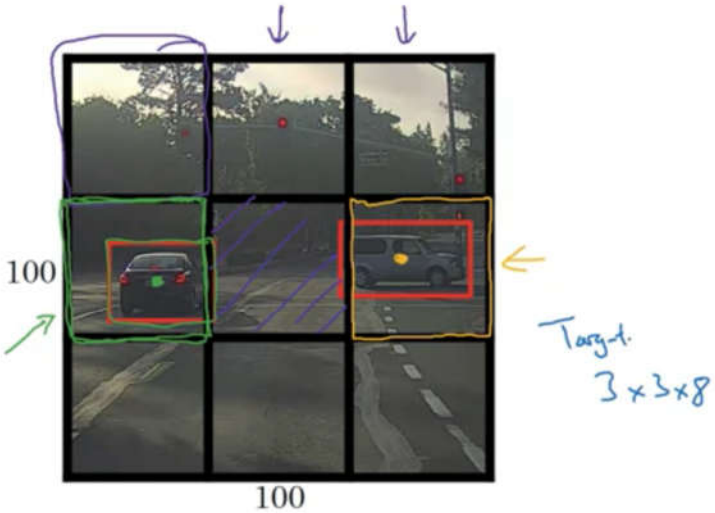
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w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

So, for each of these nine grid cells,

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

Andrew Ng

YOLO algorithm



Labels for training

For each grid cell:

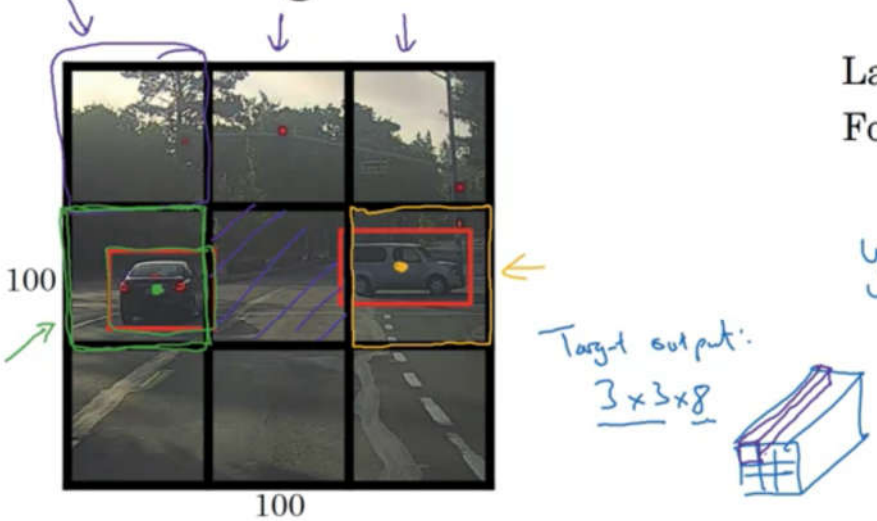
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w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

So the target output is going to be 3 by 3 by 8 because you have 3 by 3 grid cells.

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

Andrew Ng

YOLO algorithm



Labels for training

For each grid cell:

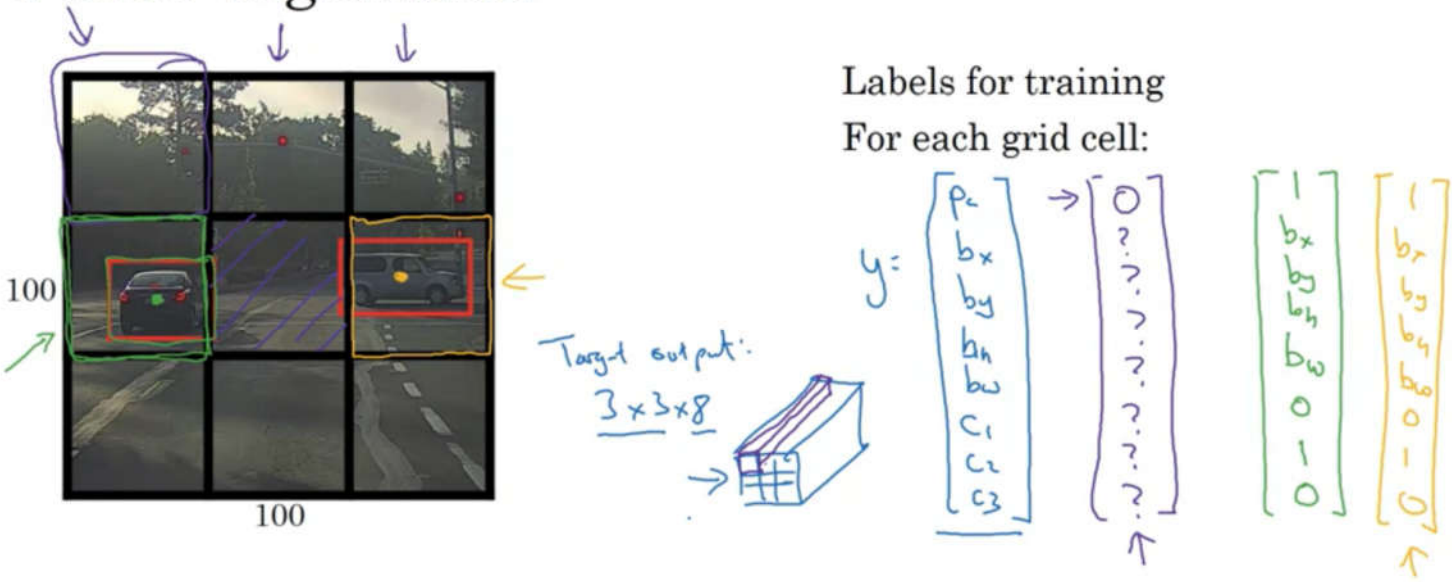
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w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

does it correspond in eight dimensional target vector Y that you want to the output.

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

Andrew Ng

YOLO algorithm

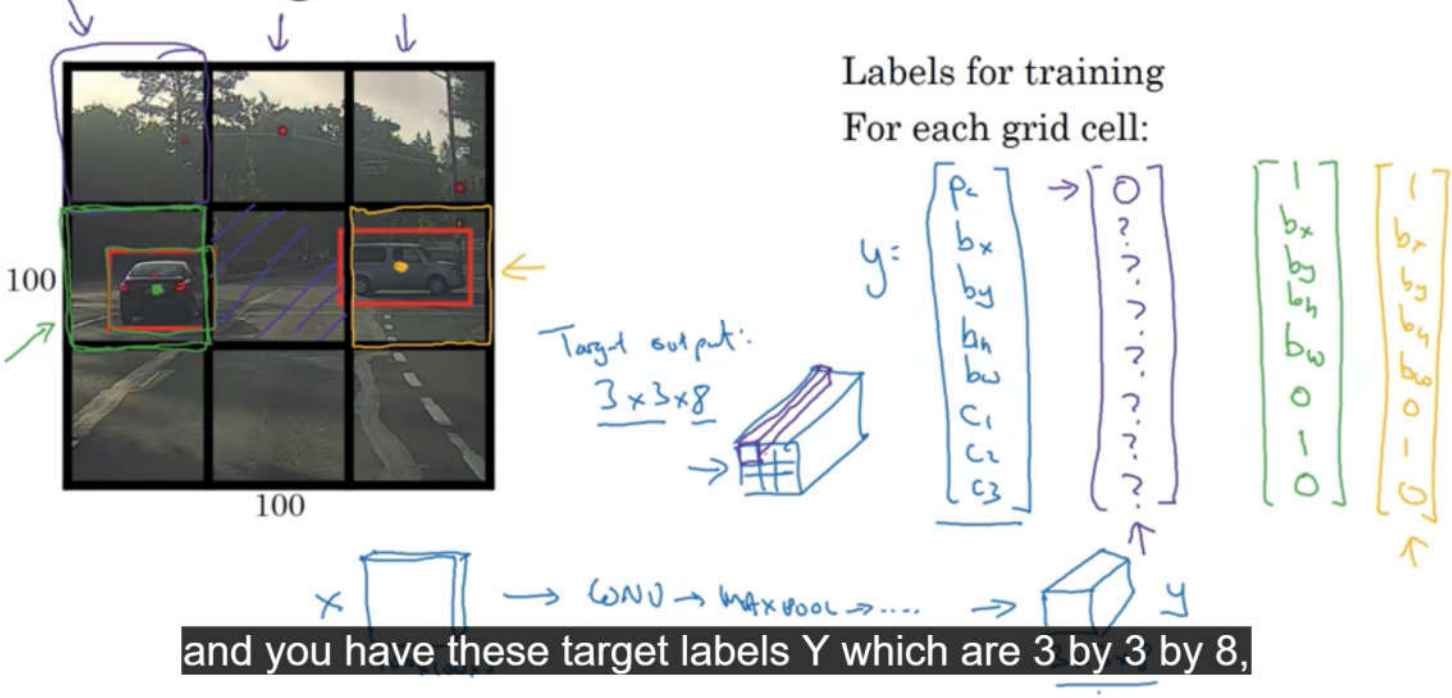


the output label for this image is now itself a 3 by 3 by 8 volume.

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

Andrew Ng

YOLO algorithm

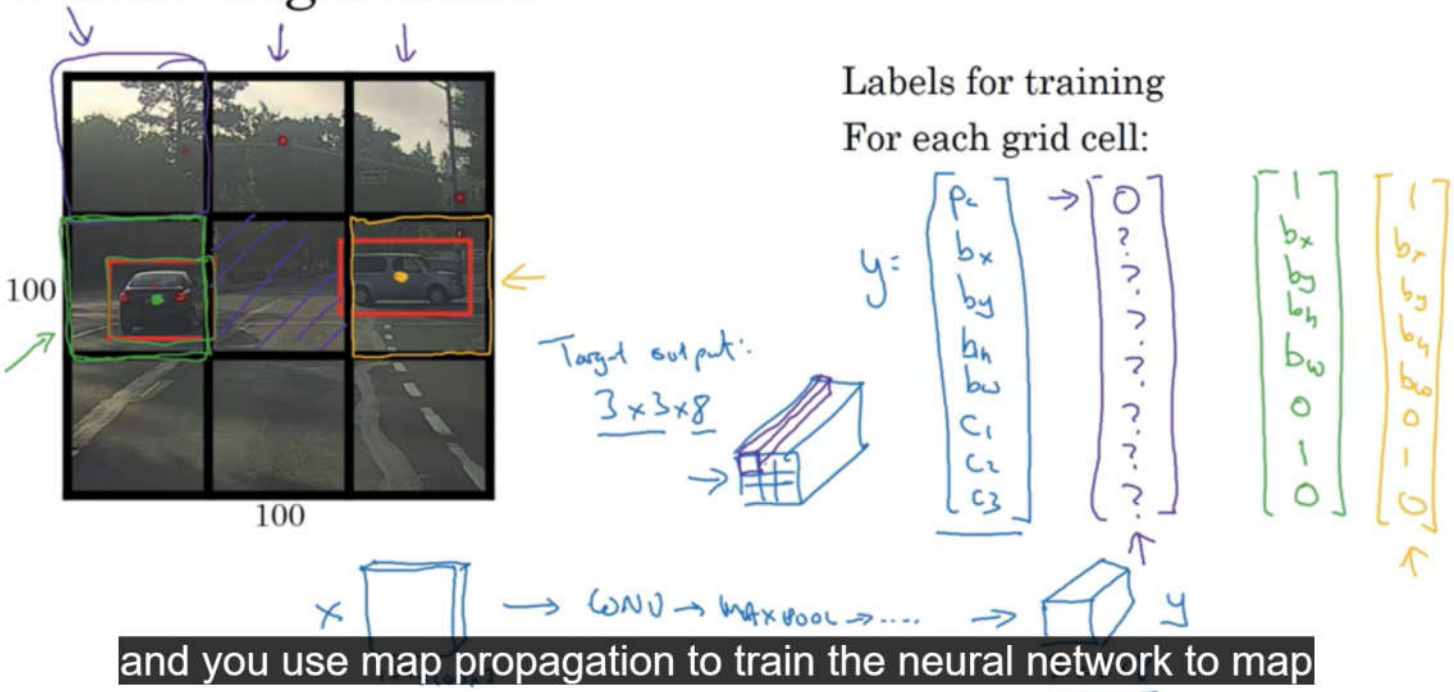


and you have these target labels Y which are 3 by 3 by 8,

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

Andrew Ng

YOLO algorithm

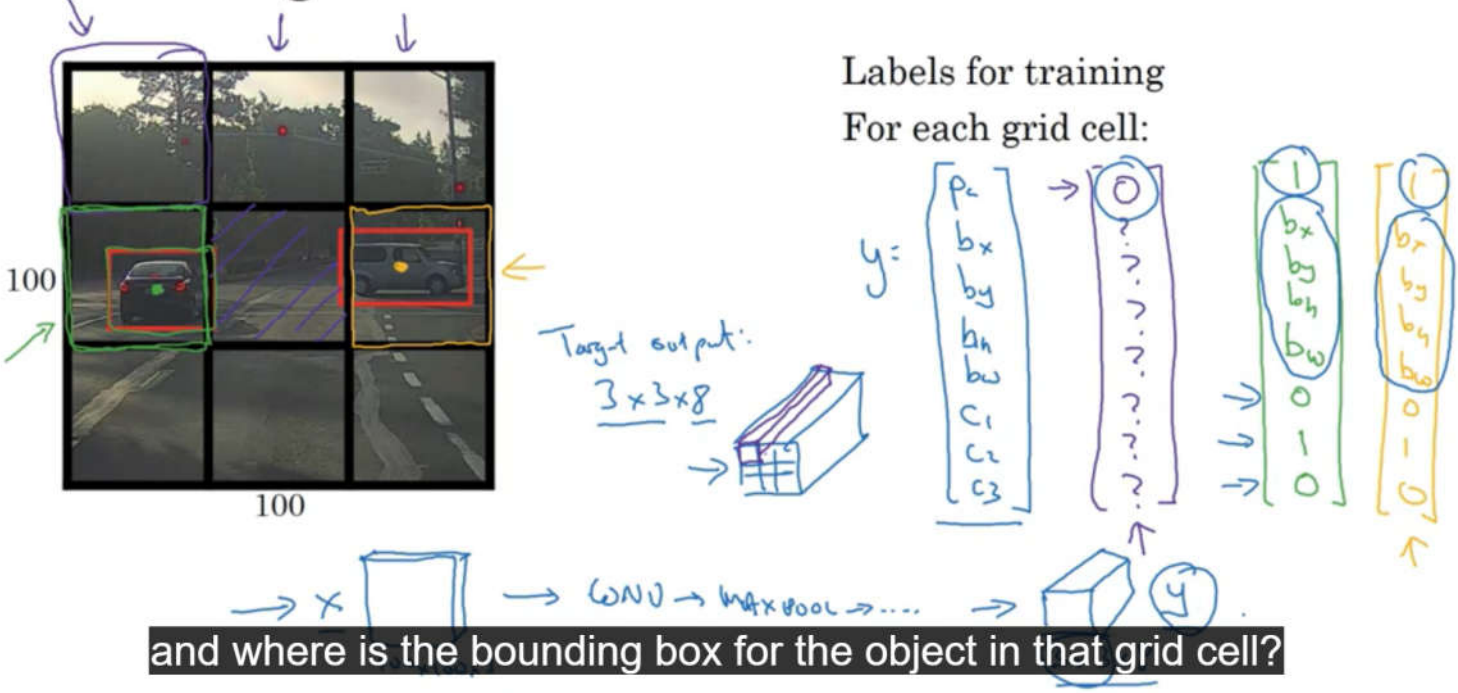


and you use map propagation to train the neural network to map

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

Andrew Ng

YOLO algorithm

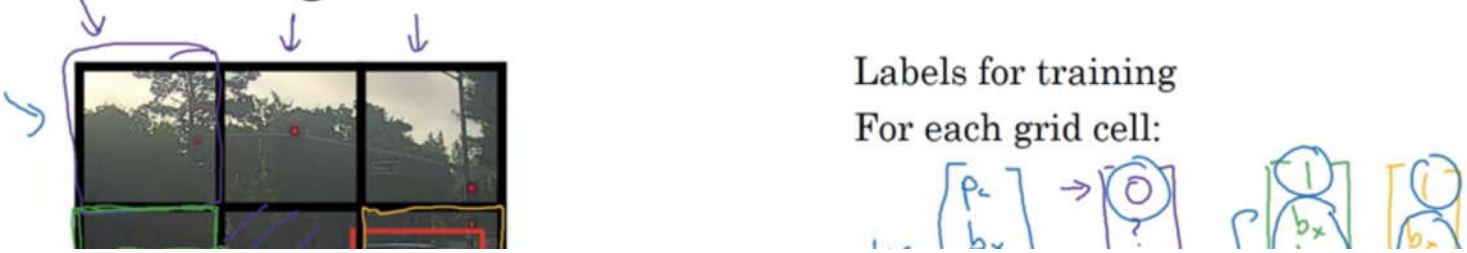


and where is the bounding box for the object in that grid cell?

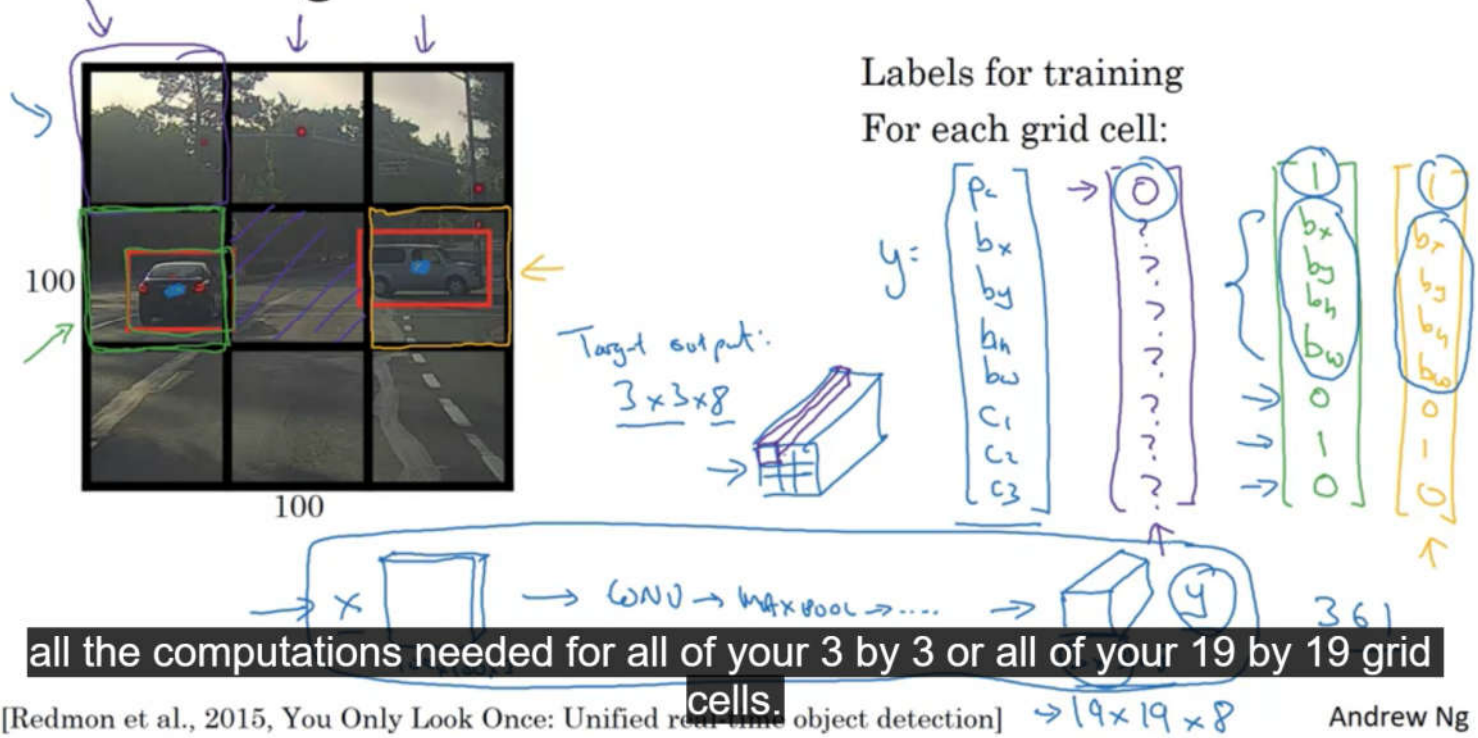
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]

Andrew Ng

YOLO algorithm



YOLO algorithm

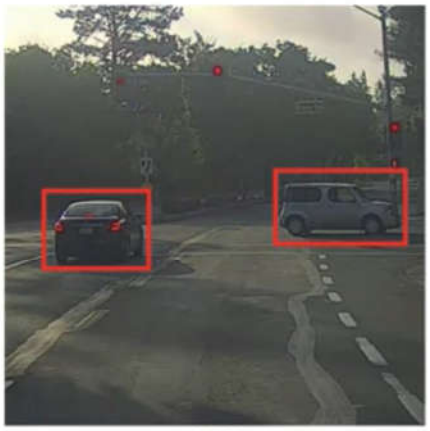


all the computations needed for all of your 3 by 3 or all of your 19 by 19 grid cells.

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

Andrew Ng

Specify the bounding boxes



100

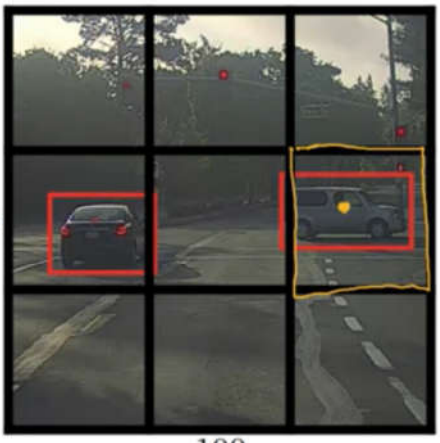
100

So, given these two cars,

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

Andrew Ng

Specify the bounding boxes



100

100

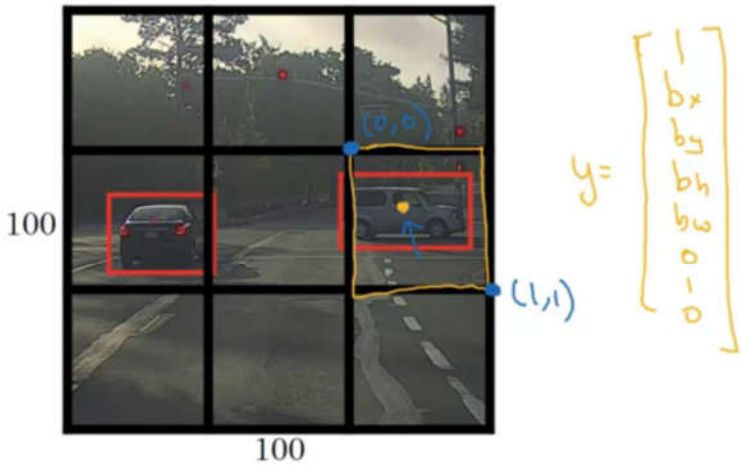
$$y = \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

So, how do you specify the bounding box?

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

Andrew Ng

Specify the bounding boxes

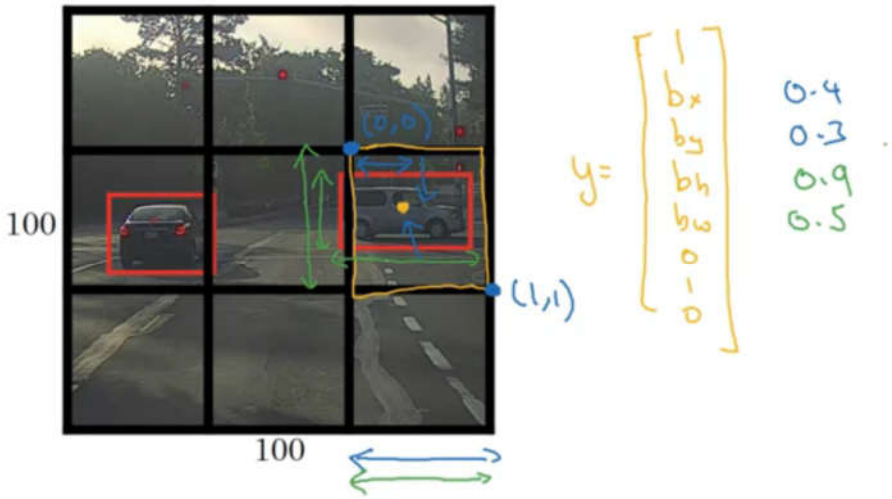


So to specify the position of that midpoint,

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

Andrew Ng

Specify the bounding boxes

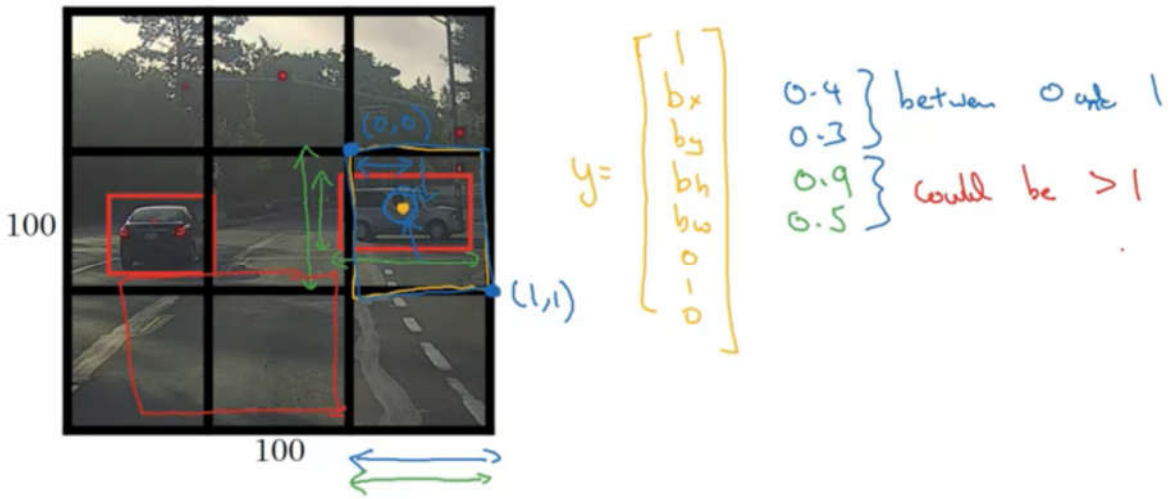


So, in other words, this bx, by, BH,

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

Andrew Ng

Specify the bounding boxes



So, there are multiple ways of specifying the bounding boxes,

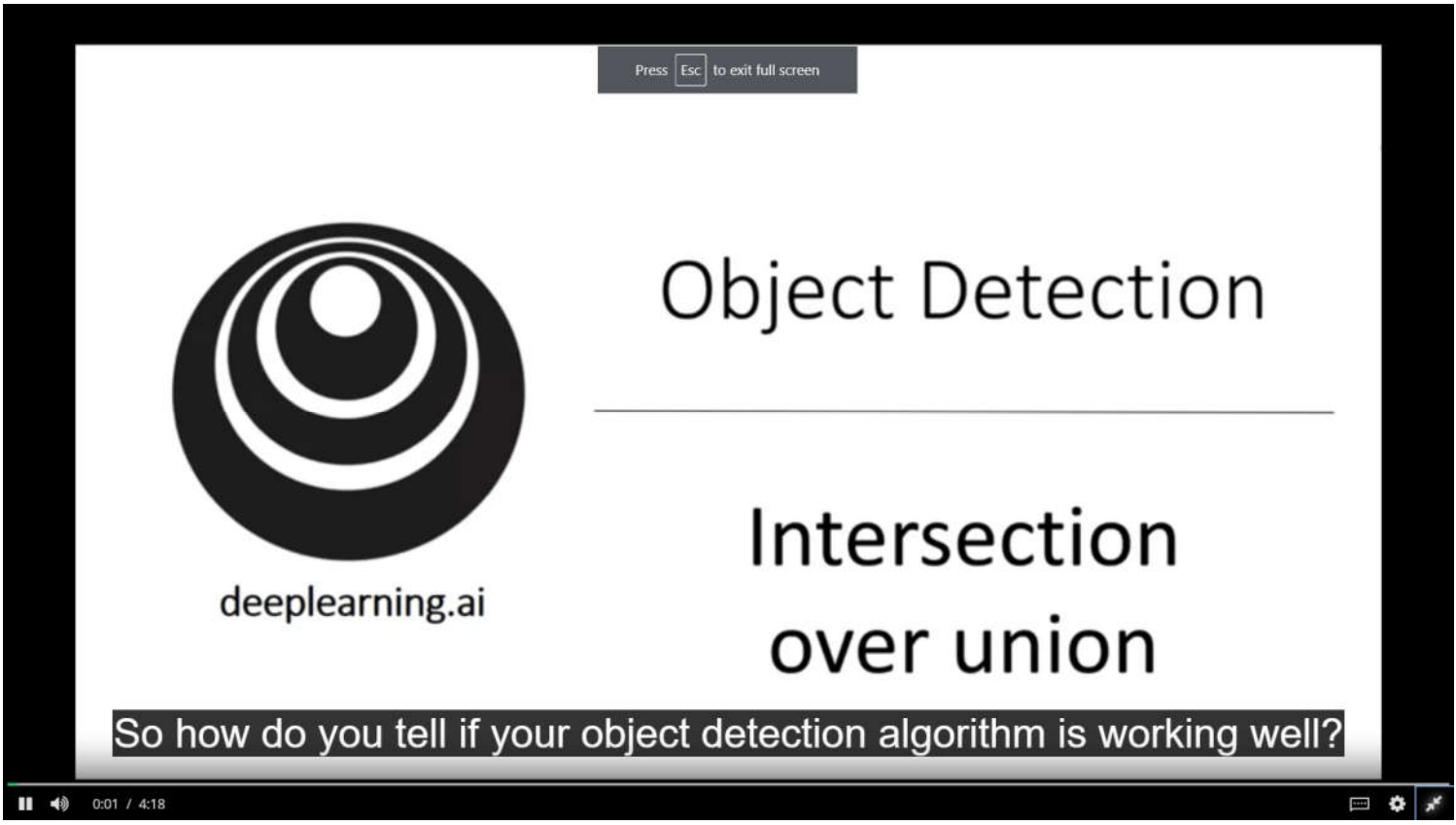
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]

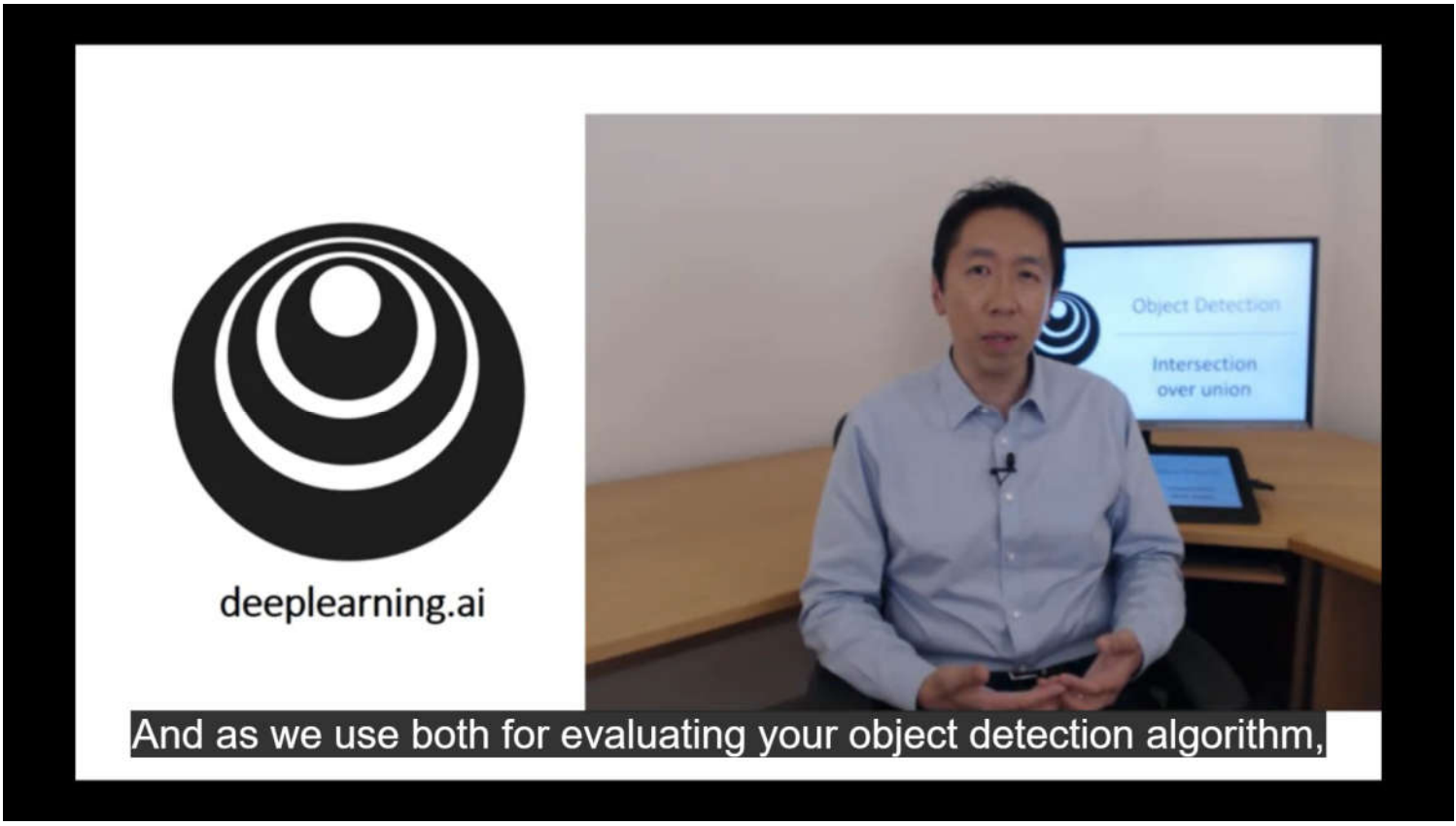
Andrew Ng

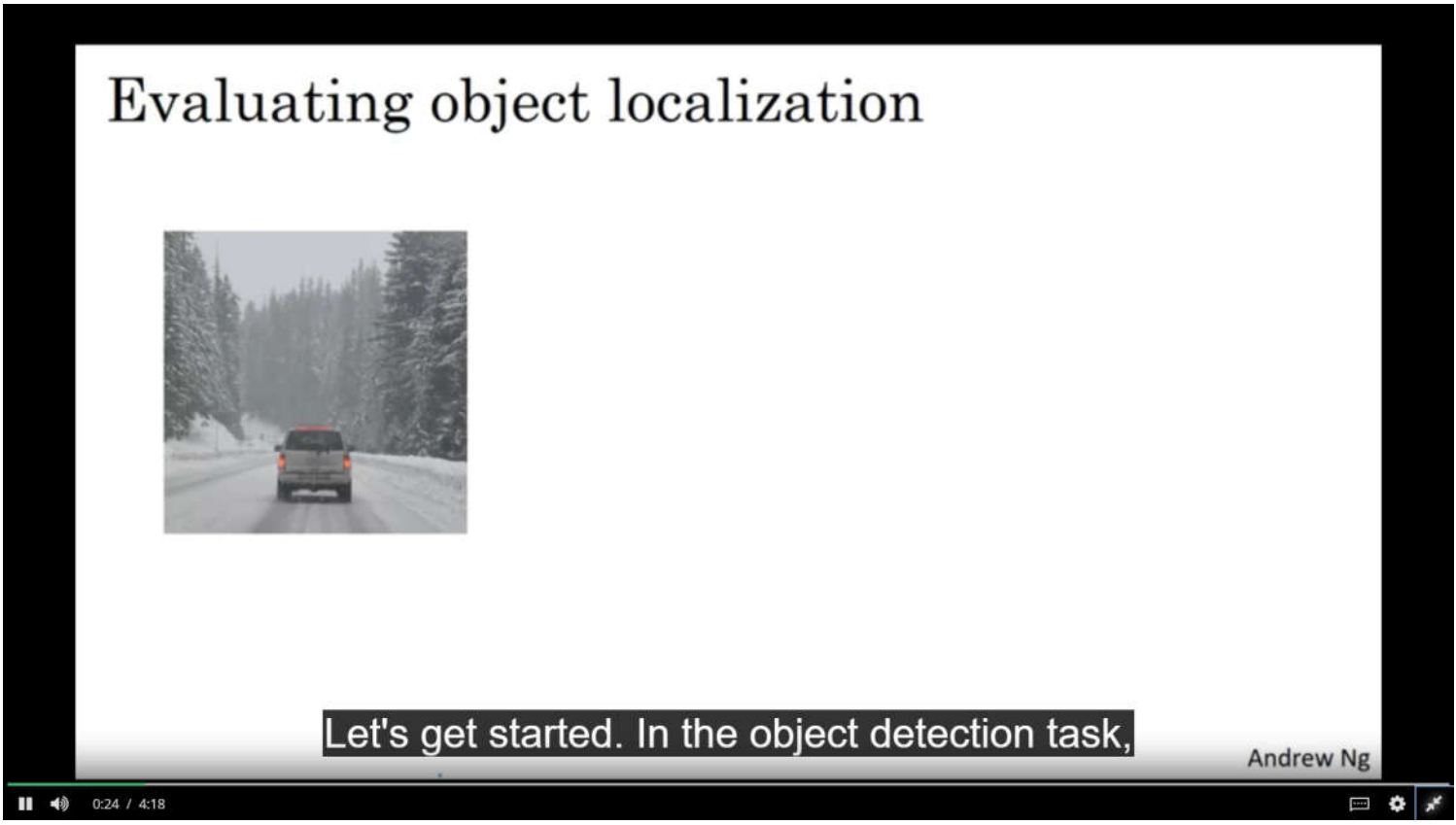
YOLO paper reference at the bottom of these past couple slides I use.

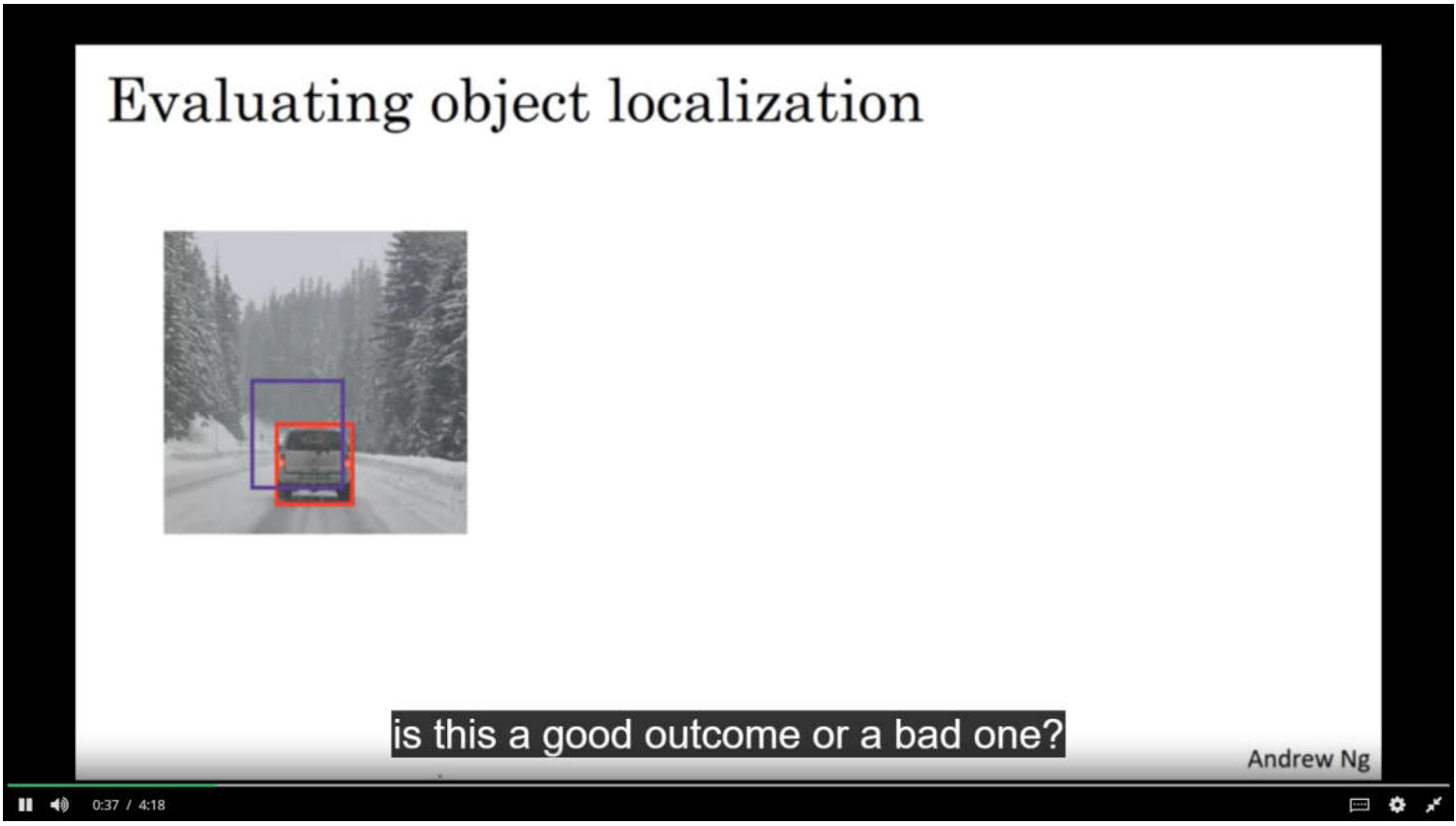
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]

Andrew Ng

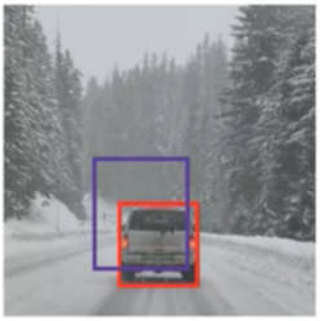

Press Esc to exit full screen
deeplearning.ai
Object Detection
Intersection over union
So how do you tell if your object detection algorithm is working well?
0:01 / 4:18


deeplearning.ai
Object Detection
Intersection over union
And as we use both for evaluating your object detection algorithm,


Evaluating object localization
Let's get started. In the object detection task,
Andrew Ng
0:24 / 4:18


Evaluating object localization
is this a good outcome or a bad one?
Andrew Ng
0:37 / 4:18

Evaluating object localization

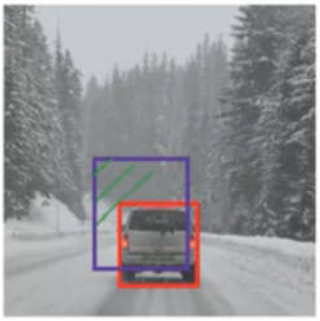


Intersection over Union (IoU)

or IoU does, is it computes the intersection over union of these two bounding boxes.

Andrew Ng

Evaluating object localization

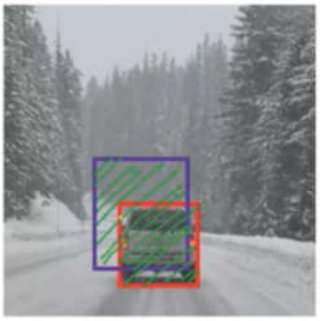


Intersection over Union (IoU)

So, the union of these two bounding boxes is this area,

Andrew Ng

Evaluating object localization



Intersection over Union (IoU)

whereas the intersection is this smaller region here.

Andrew Ng

Evaluating object localization

Intersection over Union (IoU)

So what the intersection of a union does is it computes the size of the intersection.

Andrew Ng

Evaluating object localization

Intersection over Union (IoU)

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

"Correct" if IoU $\geq$ 0.5

judge that your answer is correct if the IoU is greater than 0.5.

Andrew Ng

Evaluating object localization

Intersection over Union (IoU)

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

"Correct" if IoU $\geq$ 0.5

And if the predicted and the ground-truth bounding boxes overlapped perfectly,

Andrew Ng

v

Evaluating object localization

Intersection over Union (IoU)

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

"Correct" if IoU $\geq$ 0.5

This is just a convention.

Andrew Ng

Evaluating object localization

Intersection over Union (IoU)

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

"Correct" if IoU $\geq$ 0.5

a threshold to judge as whether the predicted bounding box is correct or not.

Andrew Ng

Evaluating object localization

Intersection over Union (IoU)

Evaluating object localization

Intersection over Union (IoU)

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

"Correct" if IoU ≥ 0.5

0.6

But the higher the IoUs,

Andrew Ng

Evaluating object localization

Intersection over Union (IoU)

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

"Correct" if IoU ≥ 0.5

0.6

the more accurate the bounding the box.

Andrew Ng

Evaluating object localization

Intersection over Union (IoU)

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

"Correct" if IoU ≥ 0.5

0.6

I sometimes see people use more stringent criteria like 0.6 or maybe 0.7.

Andrew Ng

Evaluating object localization

Intersection over Union (IoU)

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

"Correct" if IoU ≥ 0.5

0.6

your object localization algorithm is accurate or not.

Andrew Ng

Evaluating object localization

Intersection over Union (IoU)

Evaluating object localization

Intersection over Union (IoU)

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

"Correct" if IoU ≥ 0.5 ←

0.6 ←

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

Andrew Ng

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

Press Esc to exit full screen

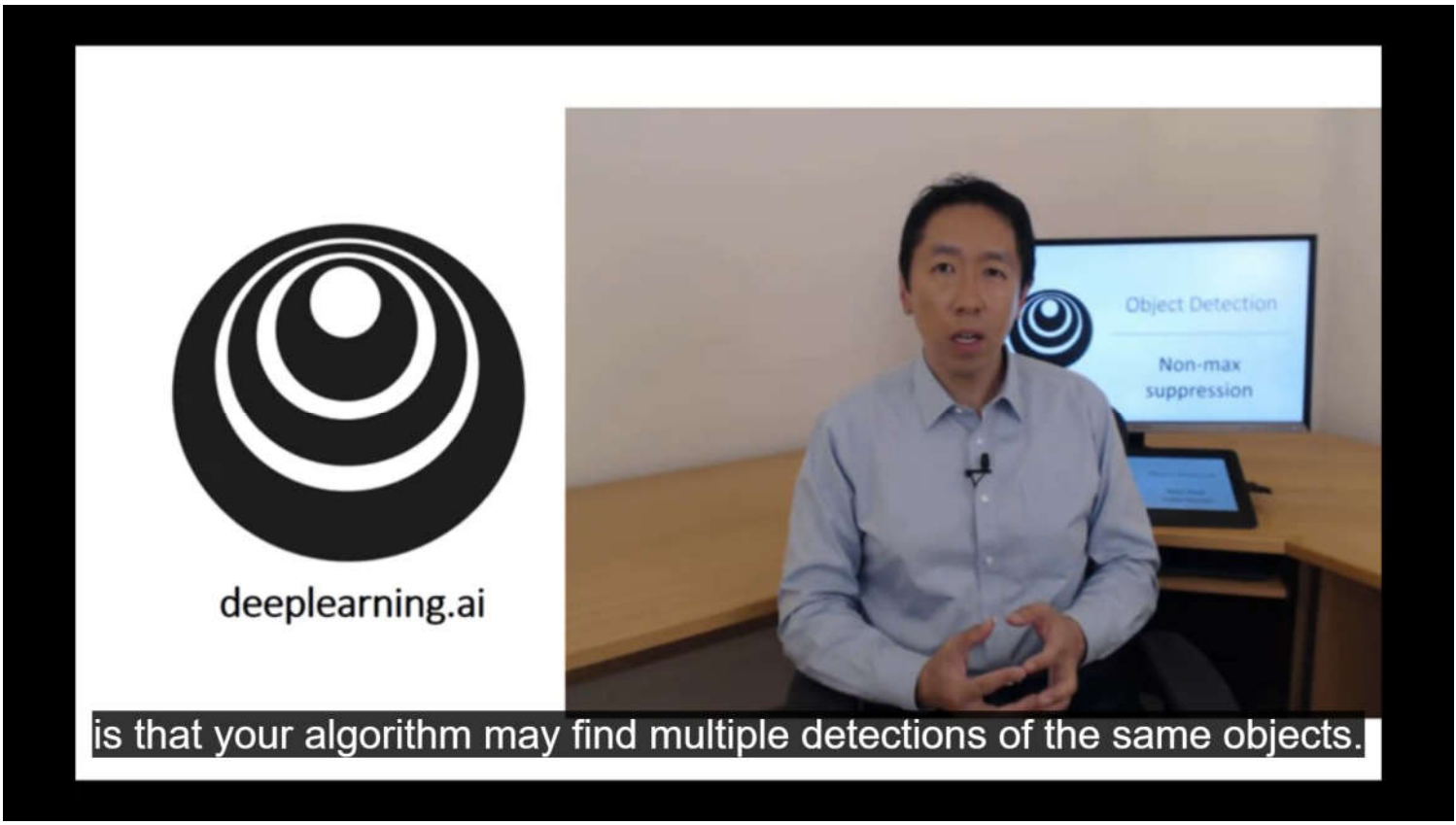
deeplearning.ai

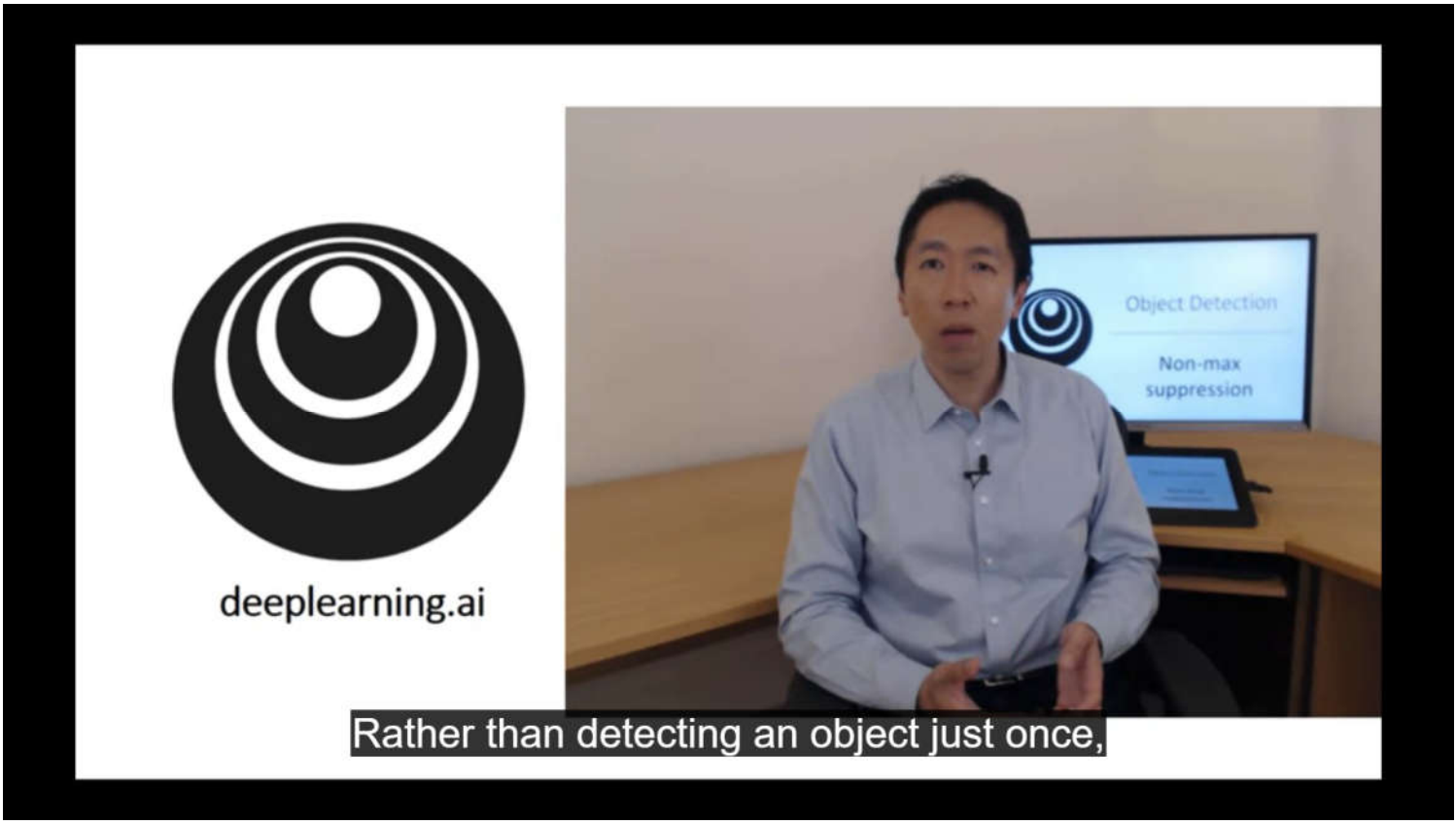
Object Detection

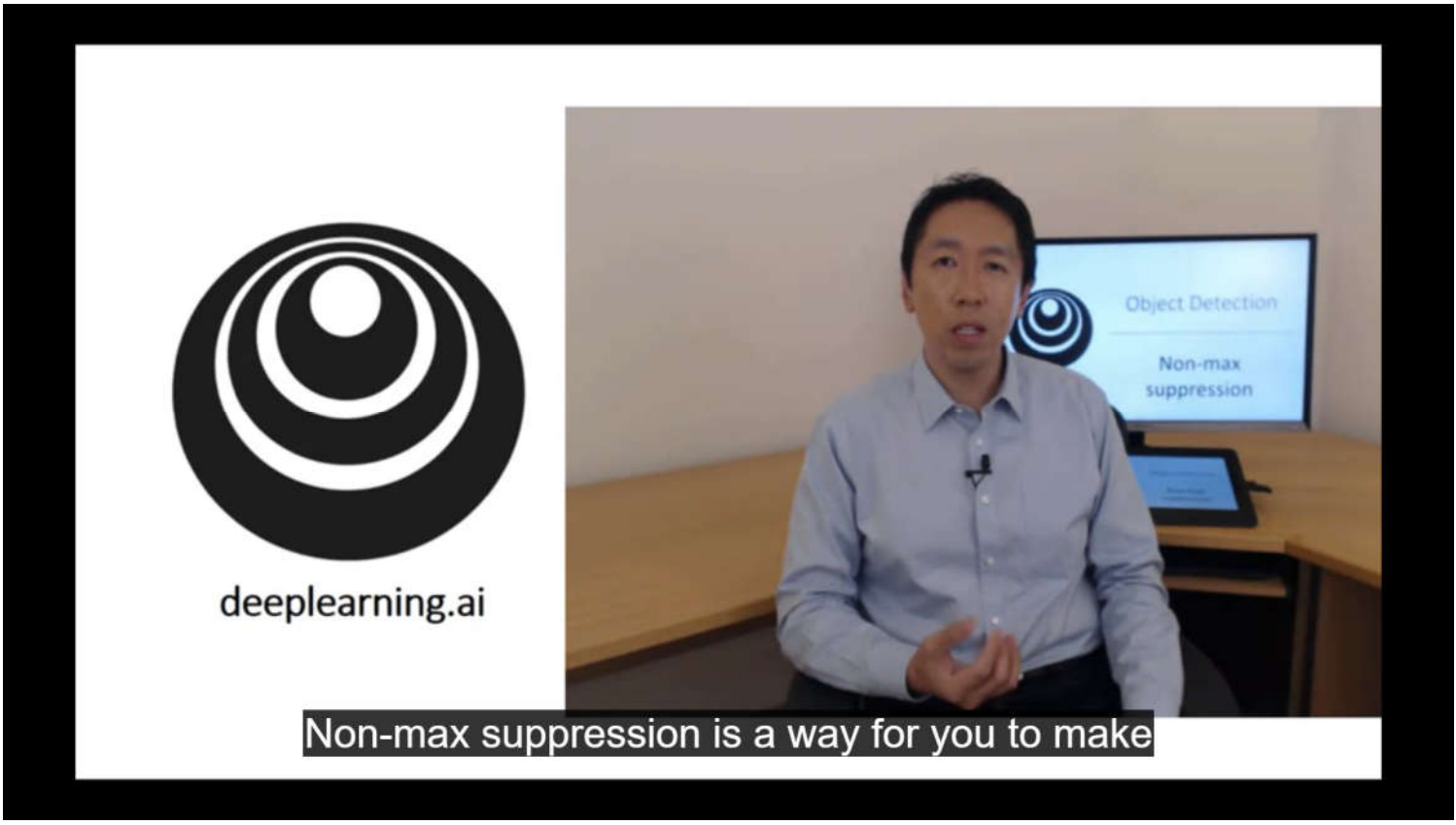
Non-max suppression

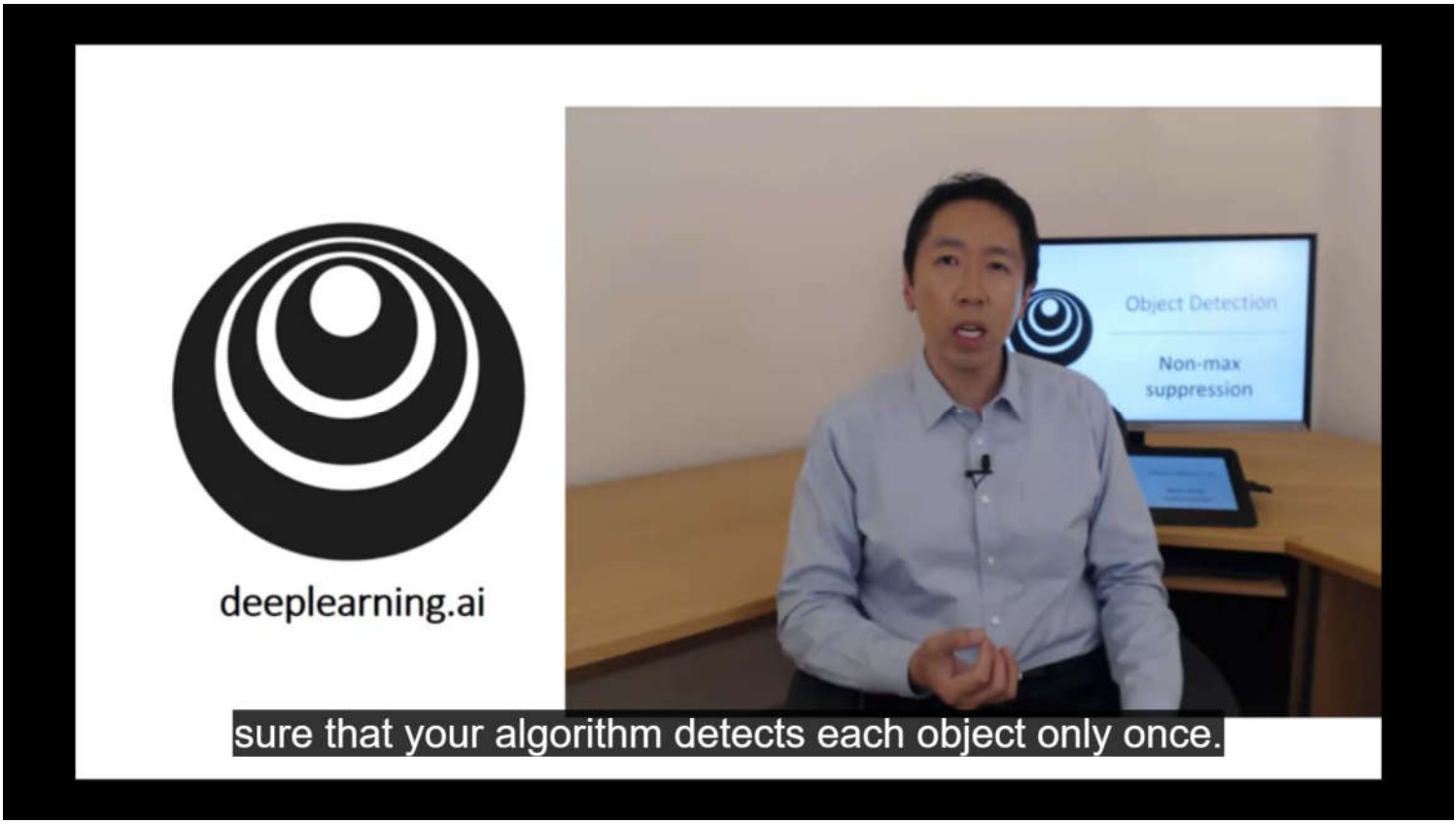
One of the problems of Object Detection as you've learned about this so far,

0:00 / 8:02


deeplearning.ai
Object Detection
Non-max suppression
is that your algorithm may find multiple detections of the same objects.


deeplearning.ai
Object Detection
Non-max suppression
Rather than detecting an object just once,


deeplearning.ai
Object Detection
Non-max suppression
Non-max suppression is a way for you to make

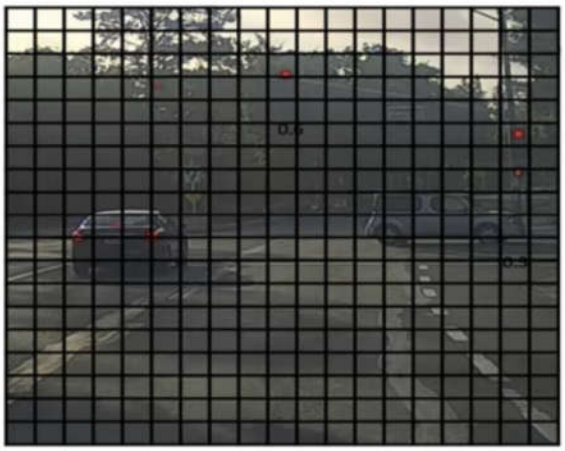

deeplearning.ai
Object Detection
Non-max suppression
sure that your algorithm detects each object only once.

Non-max suppression example

cars, and motorcycles in this image.

Andrew Ng

Non-max suppression example

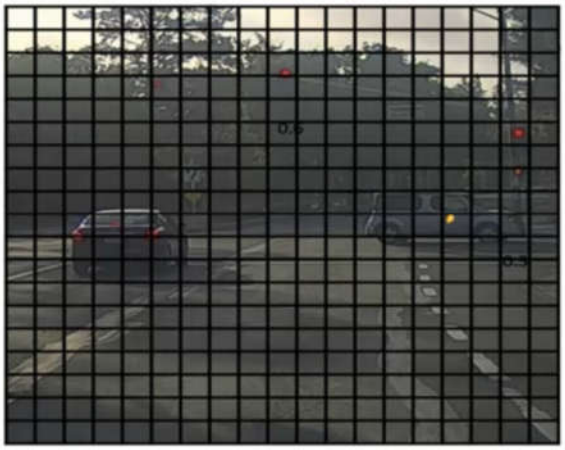


10

and this is a 19 by 19 grid.

Andrew Ng

Non-max suppression example

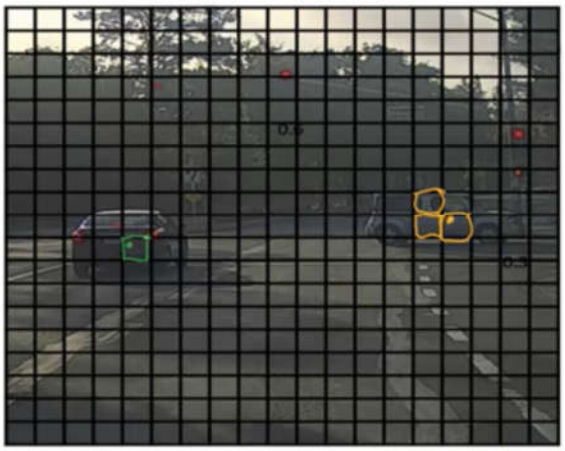


19x19

Now, while technically this car has just one midpoint,

Andrew Ng

Non-max suppression example



19x19

not only that box might decide things that's on the car,

Andrew Ng

Non-max suppression example

Non-max suppression example

the image classification and localization algorithm on every grid cell,

Andrew Ng

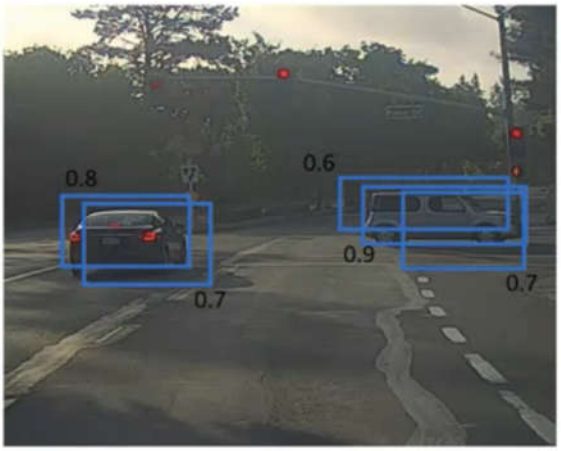
Non-max suppression example

p_c

"My Pc, my chance of thinking I have an object in it is large."

Andrew Ng

Non-max suppression example



p_c

you might end up with multiple detections of each object.

Andrew Ng

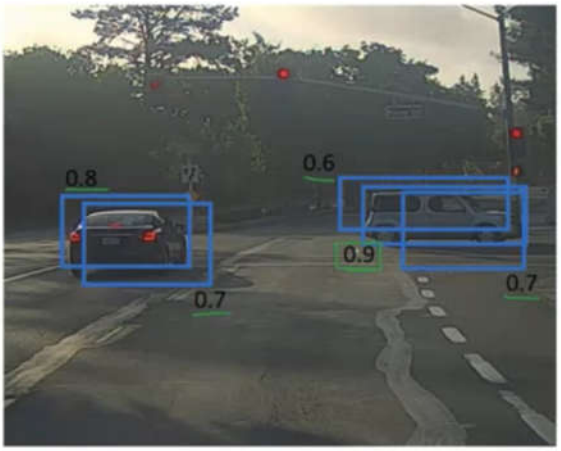
Non-max suppression example

p_c

So they end up with just one detection per car,

Andrew Ng

Non-max suppression example

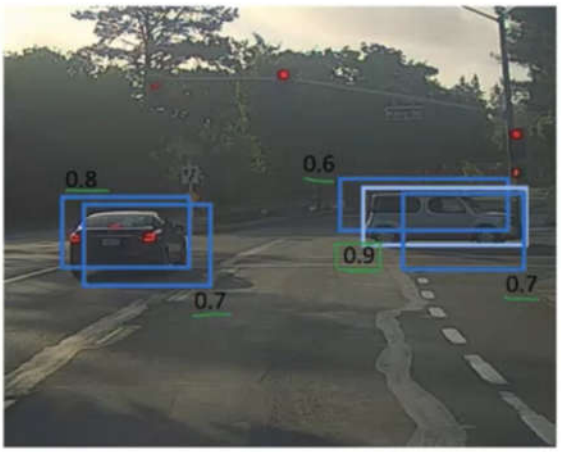


p_c

which in this case is 0.9 and says,

Andrew Ng

Non-max suppression example

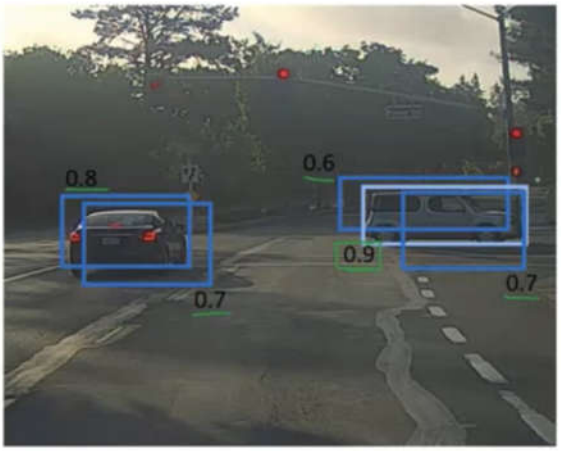


p_c

Having done that the non-max suppression part then looks at all of

Andrew Ng

Non-max suppression example

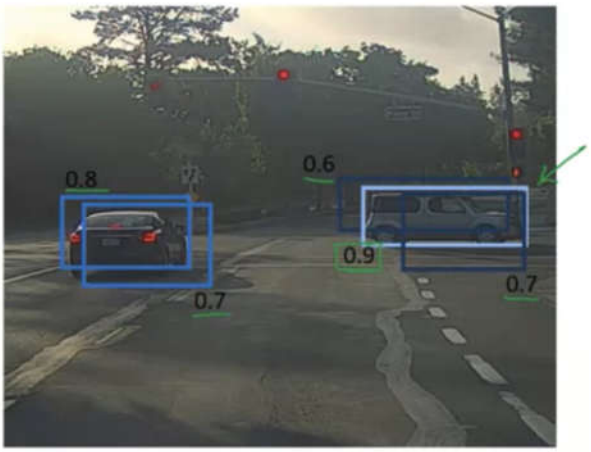


p_c

with this one that you've just output will get suppressed.

Andrew Ng

Non-max suppression example



p_c

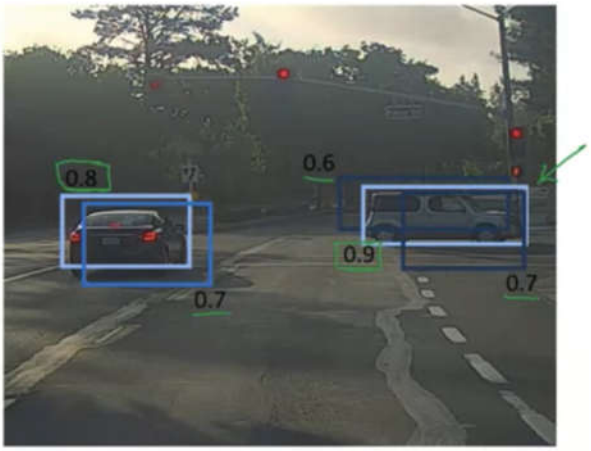
Both of those overlap a lot with the light blue rectangle.

Andrew Ng

Non-max suppression example

p_c

Non-max suppression example

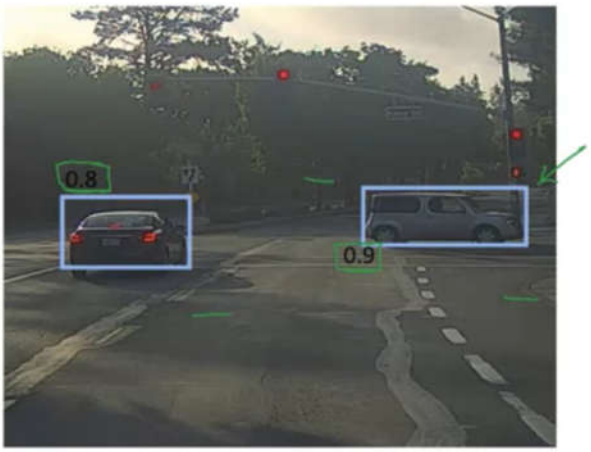


p_c

So let's commit to that and just say,

Andrew Ng

Non-max suppression example

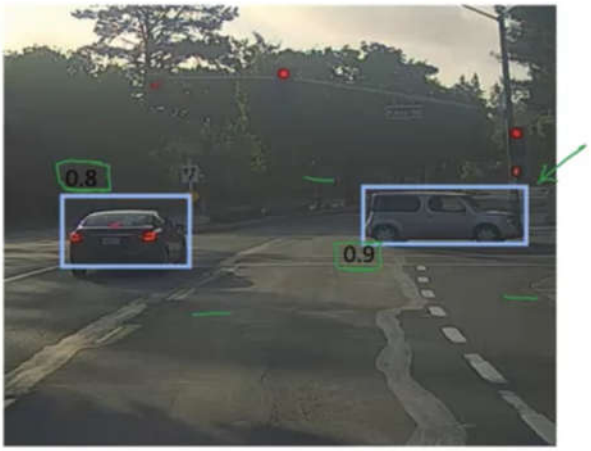


p_c

and these are your two final predictions.

Andrew Ng

Non-max suppression example

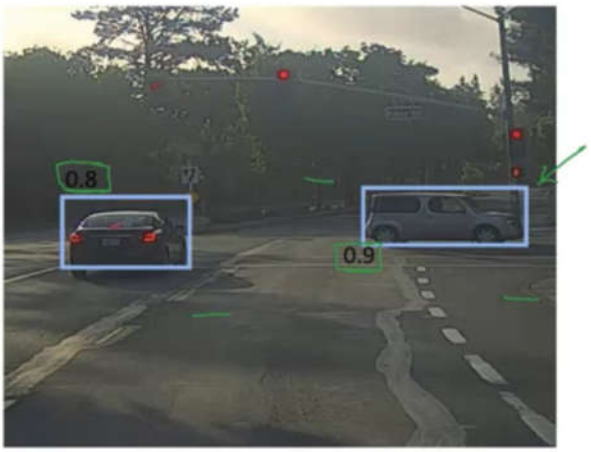


p_c

but suppress the close-by ones that are non-maximal.

Andrew Ng

Non-max suppression example

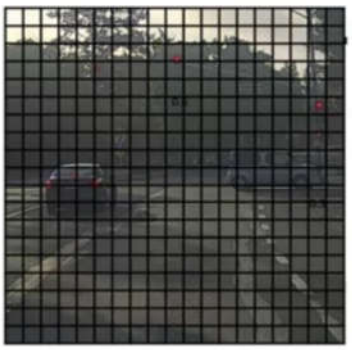


p_c

your maximal probabilities classifications

Andrew Ng

Non-max suppression algorithm

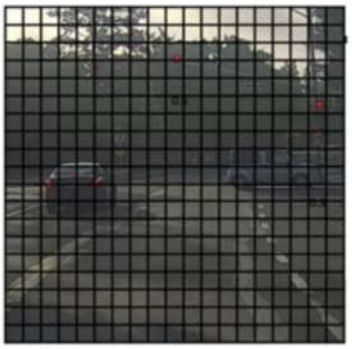


19× 19

First, on this 19 by 19 grid,

Andrew Ng

Non-max suppression algorithm



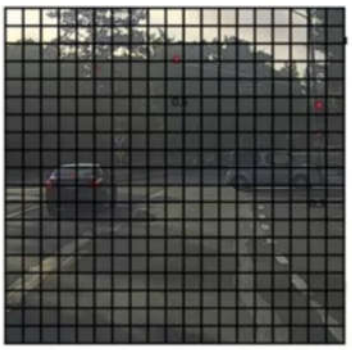
19× 19

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

C3, and pretend for this line,

Andrew Ng

Non-max suppression algorithm



19× 19

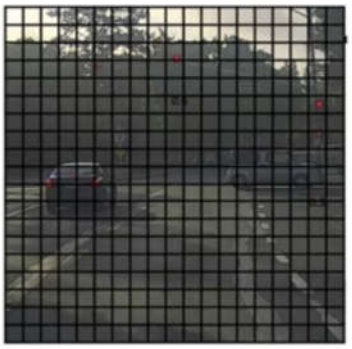
Each output prediction is: $\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \end{bmatrix}$ $\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$

Discard all boxes with $p_c \leq 0.6$

the first thing you can do is discard all the boxes,

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:

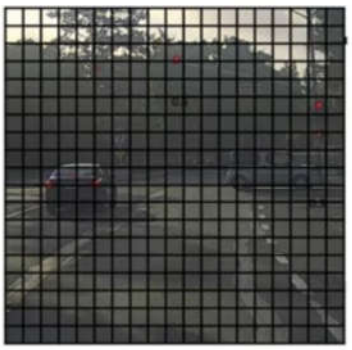
Next, while there are

Andrew Ng

Non-max suppression algorithm

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

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 $p_c \leq 0.6$

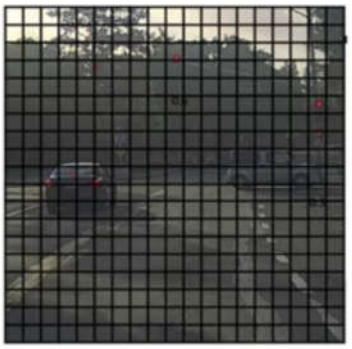
While there are any remaining boxes:

- Pick the box with the largest p_c
  Output that as a prediction.

with the highest Pc,

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 $p_c \leq 0.6$

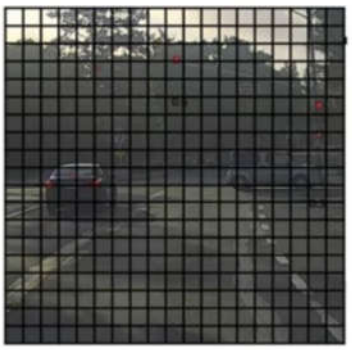
While there are any remaining boxes:

- Pick the box with the largest p_c
  Output that as a prediction.
- Discard any remaining box with
  IoU ≥ 0.5 with the box output

and that was not previously discarded.

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 ≥ 0.5 with the box output in previous step

with a high IOU,

Andrew Ng

Press Esc to exit full screen

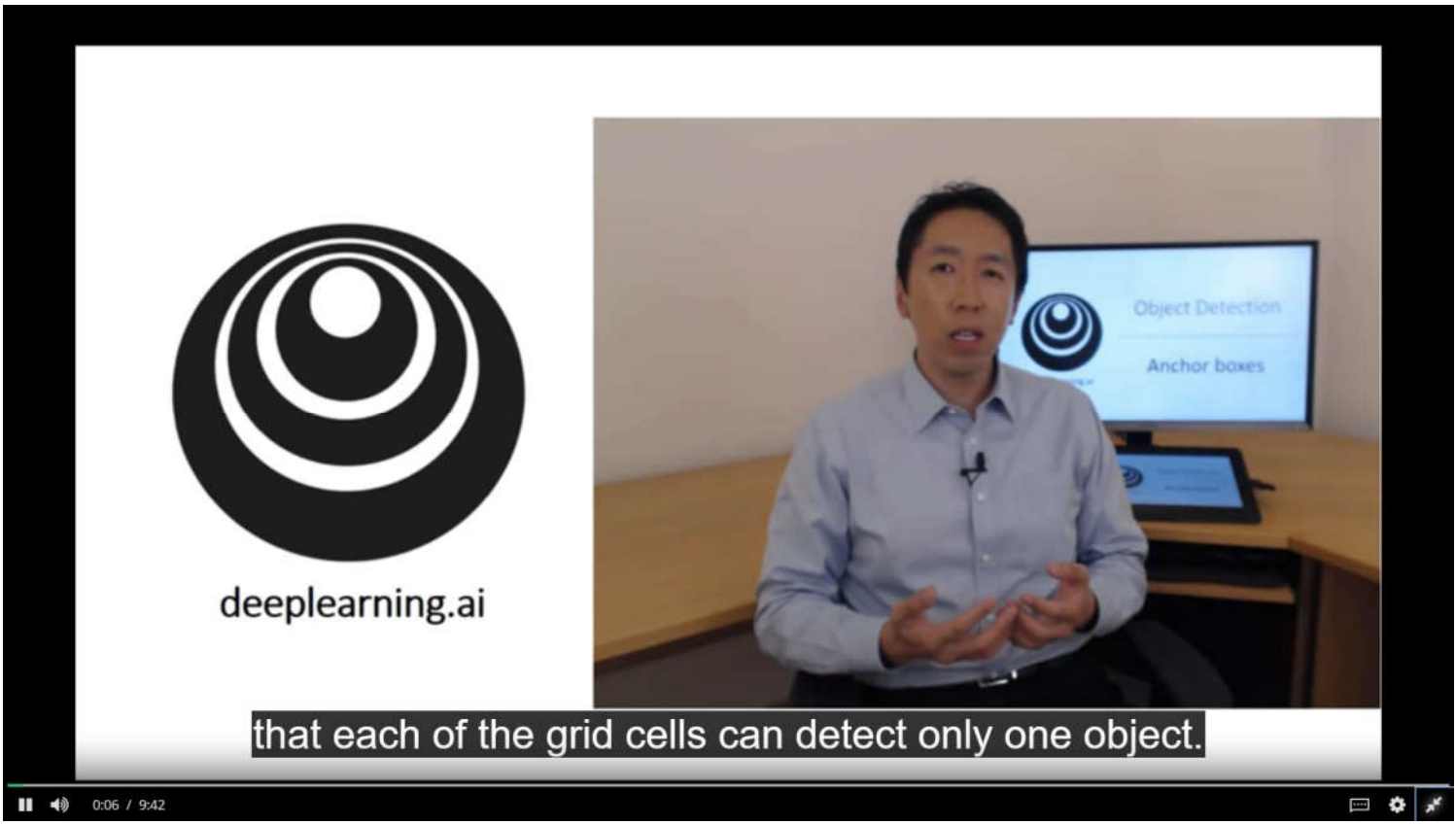
deeplearning.ai

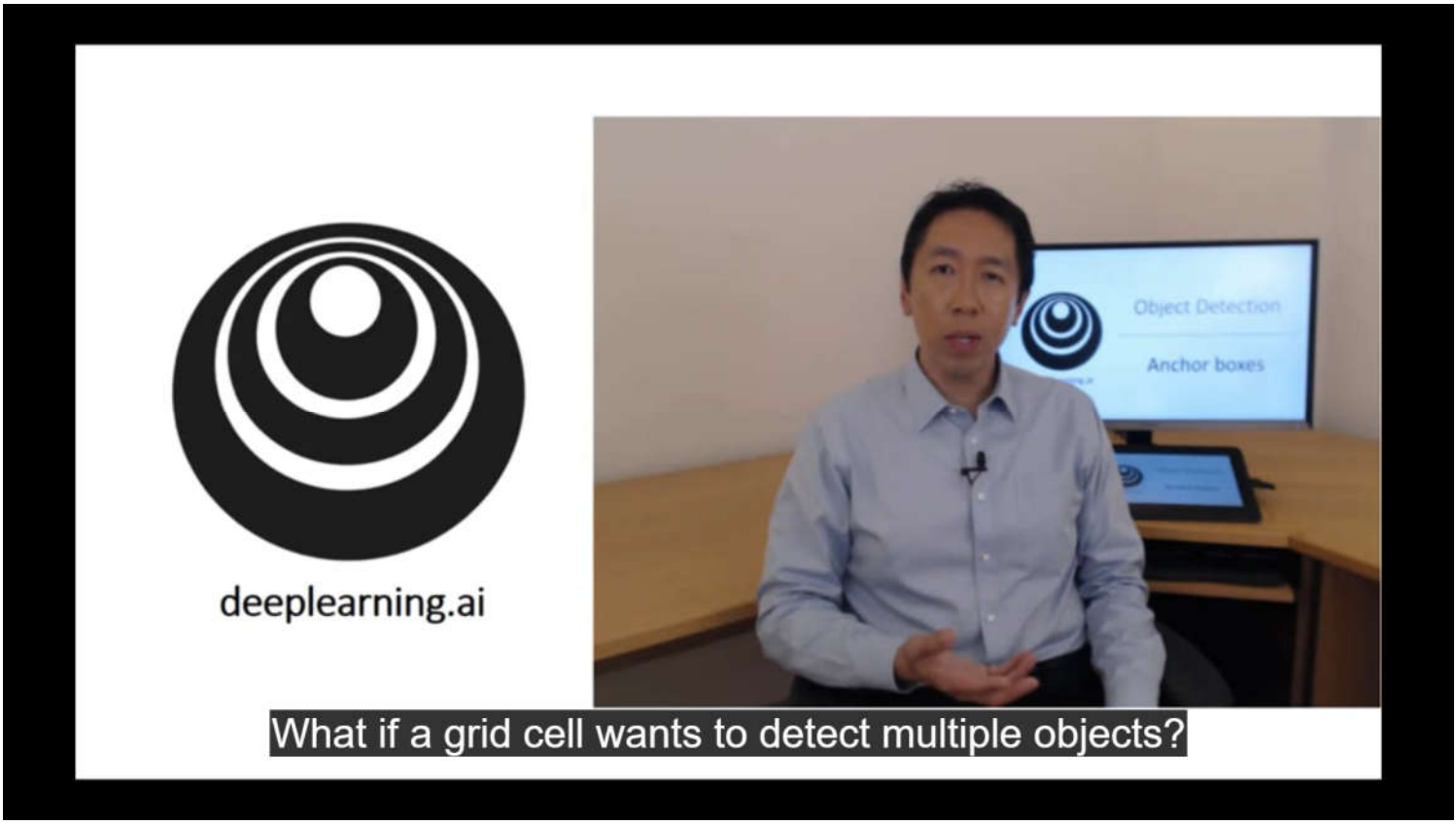
Object Detection

Anchor boxes

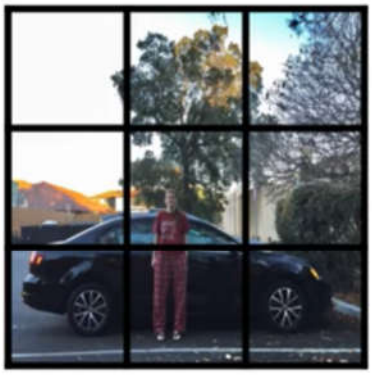
One of the problems with object detection as you have seen it so far is

0:02 / 9:42


deeplearning.ai
Object Detection
Anchor boxes
that each of the grid cells can detect only one object.
0:06 / 9:42


deeplearning.ai
Object Detection
Anchor boxes
What if a grid cell wants to detect multiple objects?

Overlapping objects:

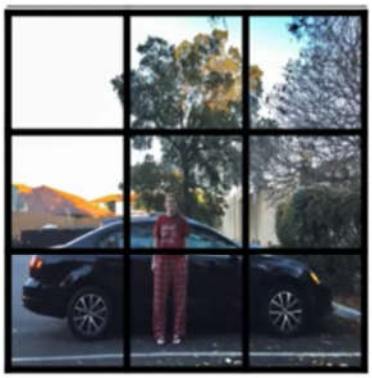


Let's say you have an image like this.

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

Andrew Ng

Overlapping objects:



I am going to continue to use a 3 by 3 grid.

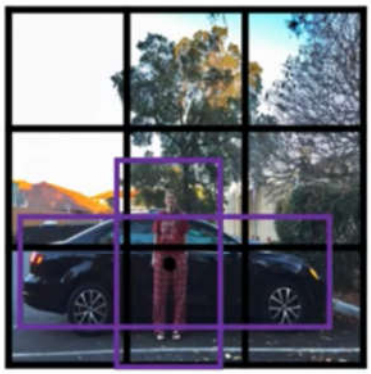
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]

Andrew Ng

v

Overlapping objects:

Overlapping objects:

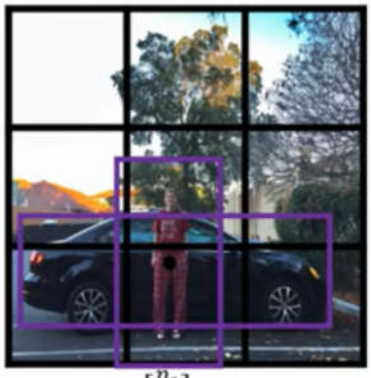


in almost the same place and both of them fall into the same grid cell.

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

Andrew Ng

Overlapping objects:

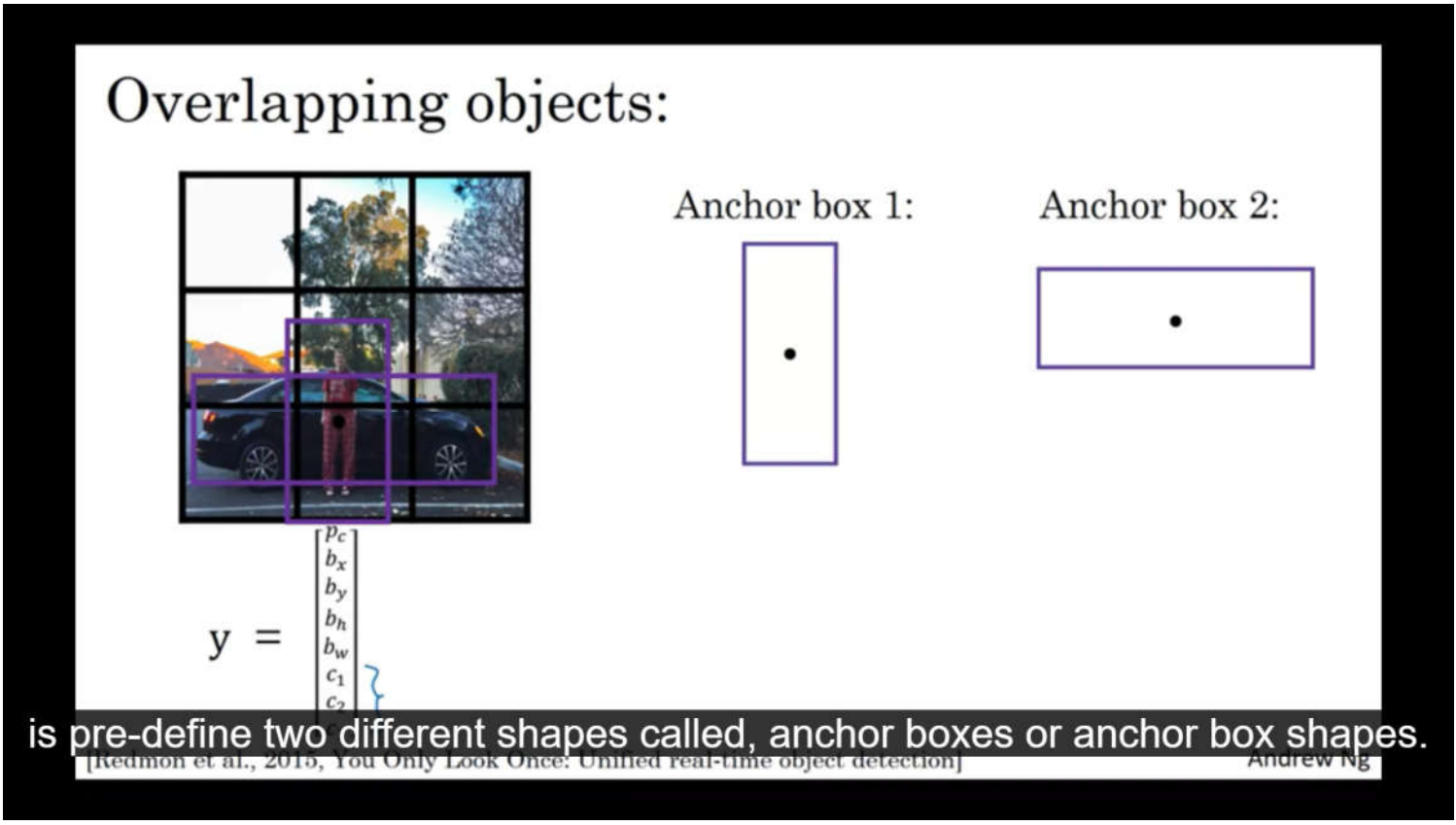


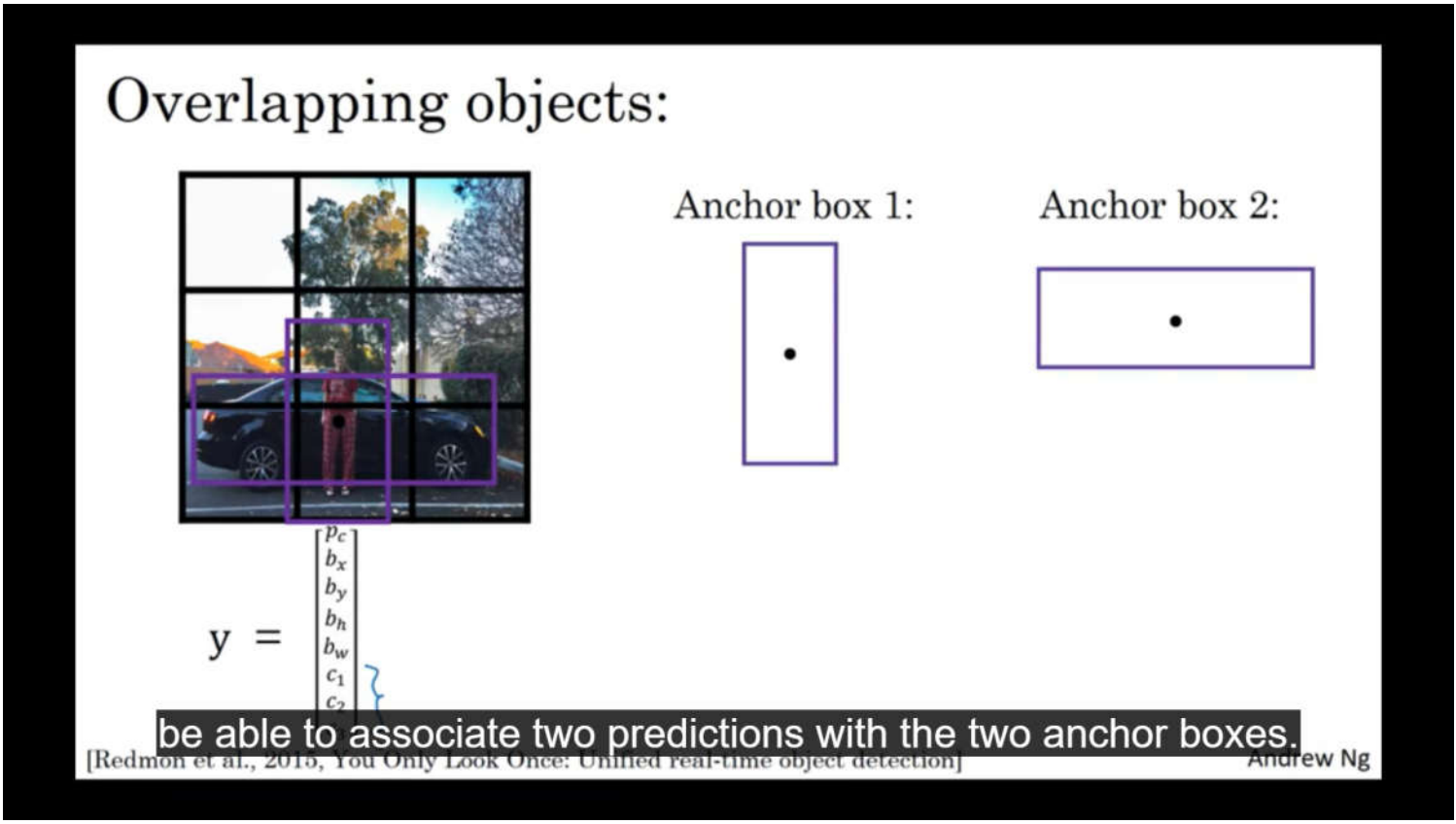
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \end{bmatrix}$$

So I have to pick one of the two detections to output.

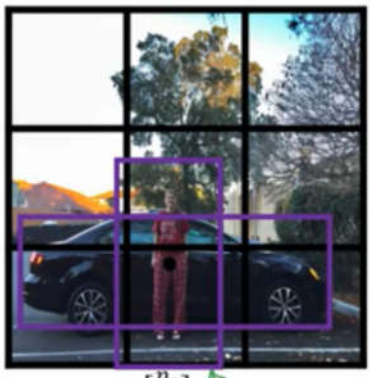
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]

Andrew Ng


Overlapping objects:
Anchor box 1:
Anchor box 2:
y =
p_c
b_x
b_y
b_h
b_w
c_1
c_2
is pre-define two different shapes called, anchor boxes or anchor box shapes.
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Andrew Ng


Overlapping objects:
Anchor box 1:
Anchor box 2:
y =
p_c
b_x
b_y
b_h
b_w
c_1
c_2
be able to associate two predictions with the two anchor boxes.
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Andrew Ng

Overlapping objects:



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

Anchor box 1:

Anchor box 2:

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

Anchor box 1

Anchor box 2

the pedestrian is more similar to the shape of anchor box 1 and anchor box 2,

[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.

So to summarize, previously,

Andrew Ng

Anchor box algorithm

Previously:

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

Output y:

3 × 3 × 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.

you now do that following.

Andrew Ng

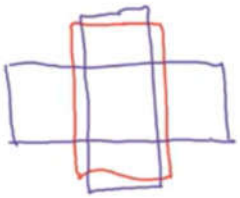
Anchor box algorithm

Previously:

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

Output y:

3 × 3 × 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.

and then you see which of the two anchor boxes has a higher IoU,

Andrew Ng

Anchor box algorithm

Previously:

Each object in training image is assigned to grid

With two anchor boxes:

Each object in training image is assigned to grid

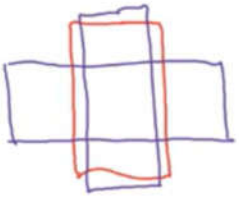
Anchor box algorithm

Previously:

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

Output y:

3 × 3 × 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)

It gets assigned to grid cell comma anchor box pair.

Andrew Ng

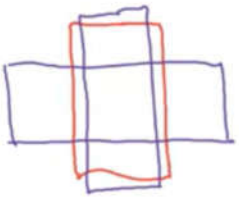
Anchor box algorithm

Previously:

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

Output y:

3 × 3 × 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.

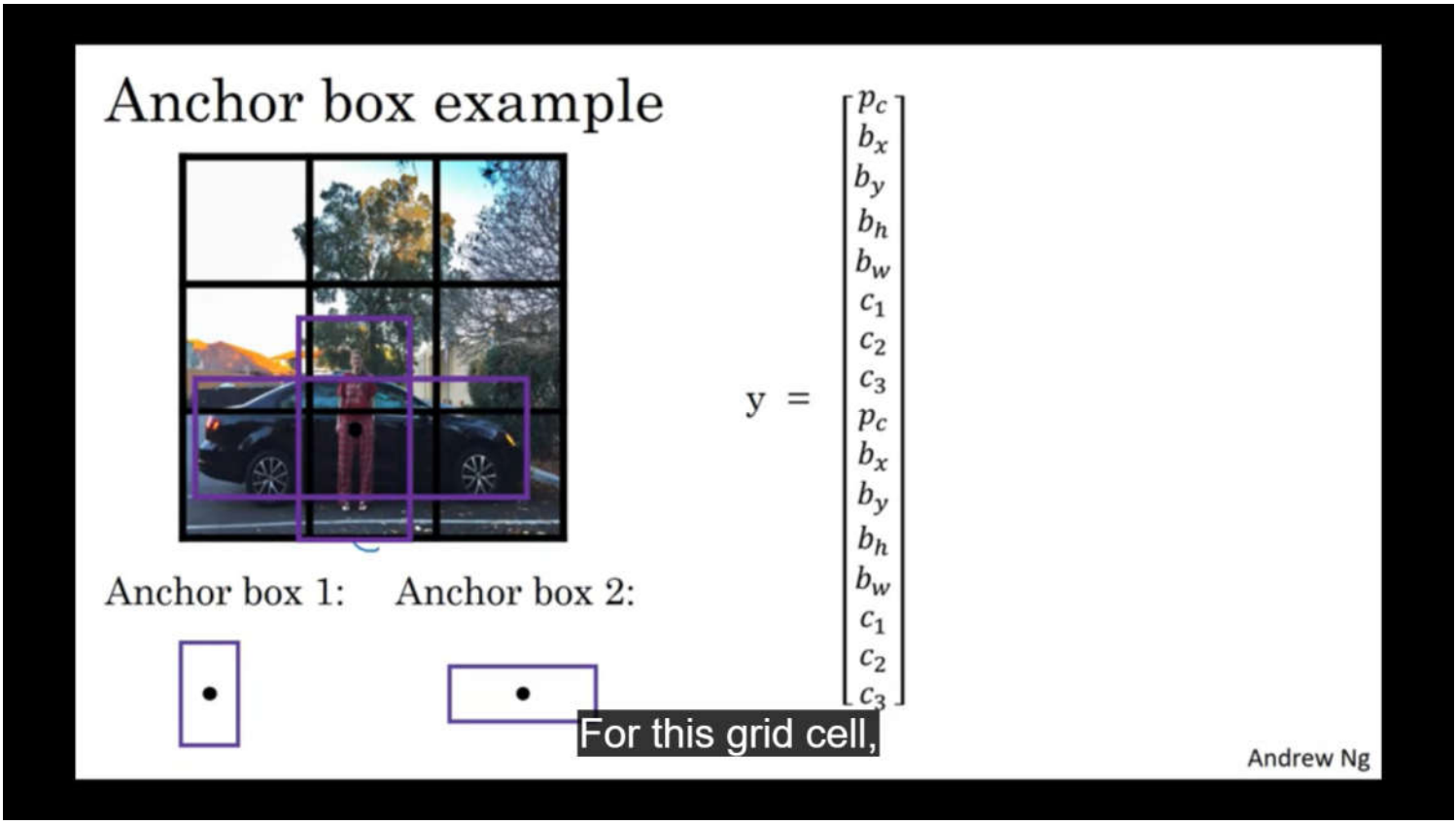
(grid cell, anchor box)

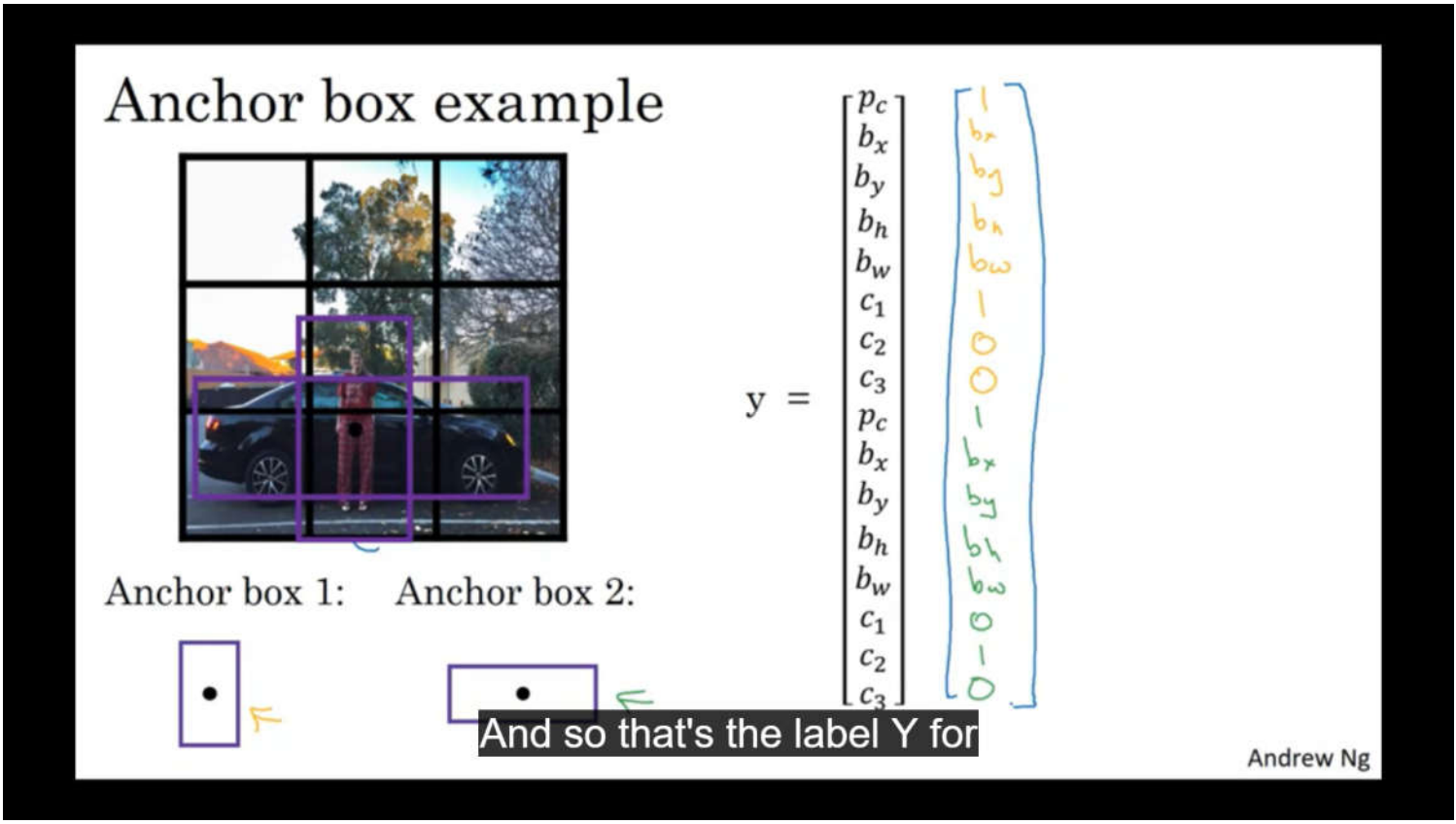
Output y:

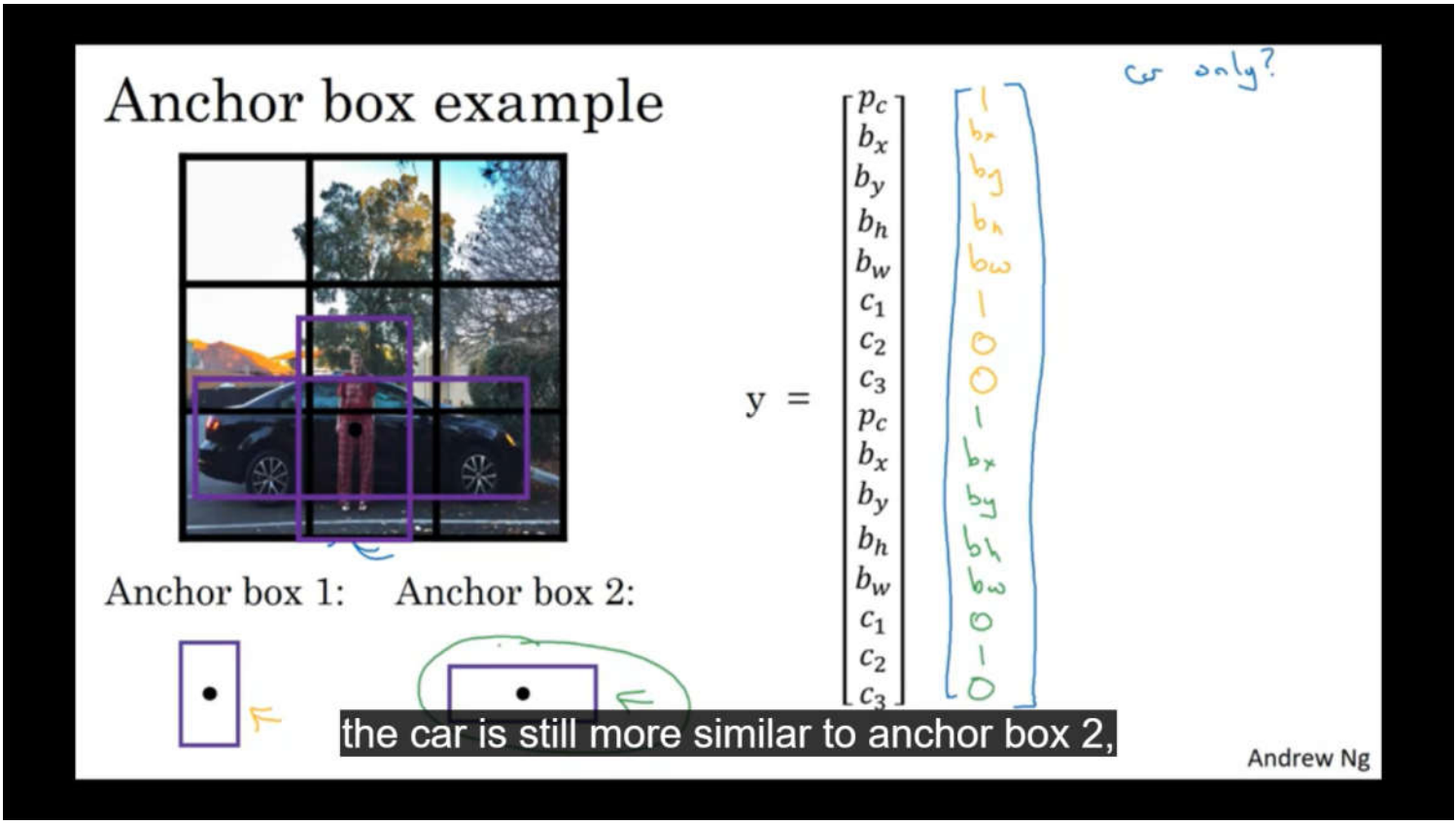
3 × 3 × 16

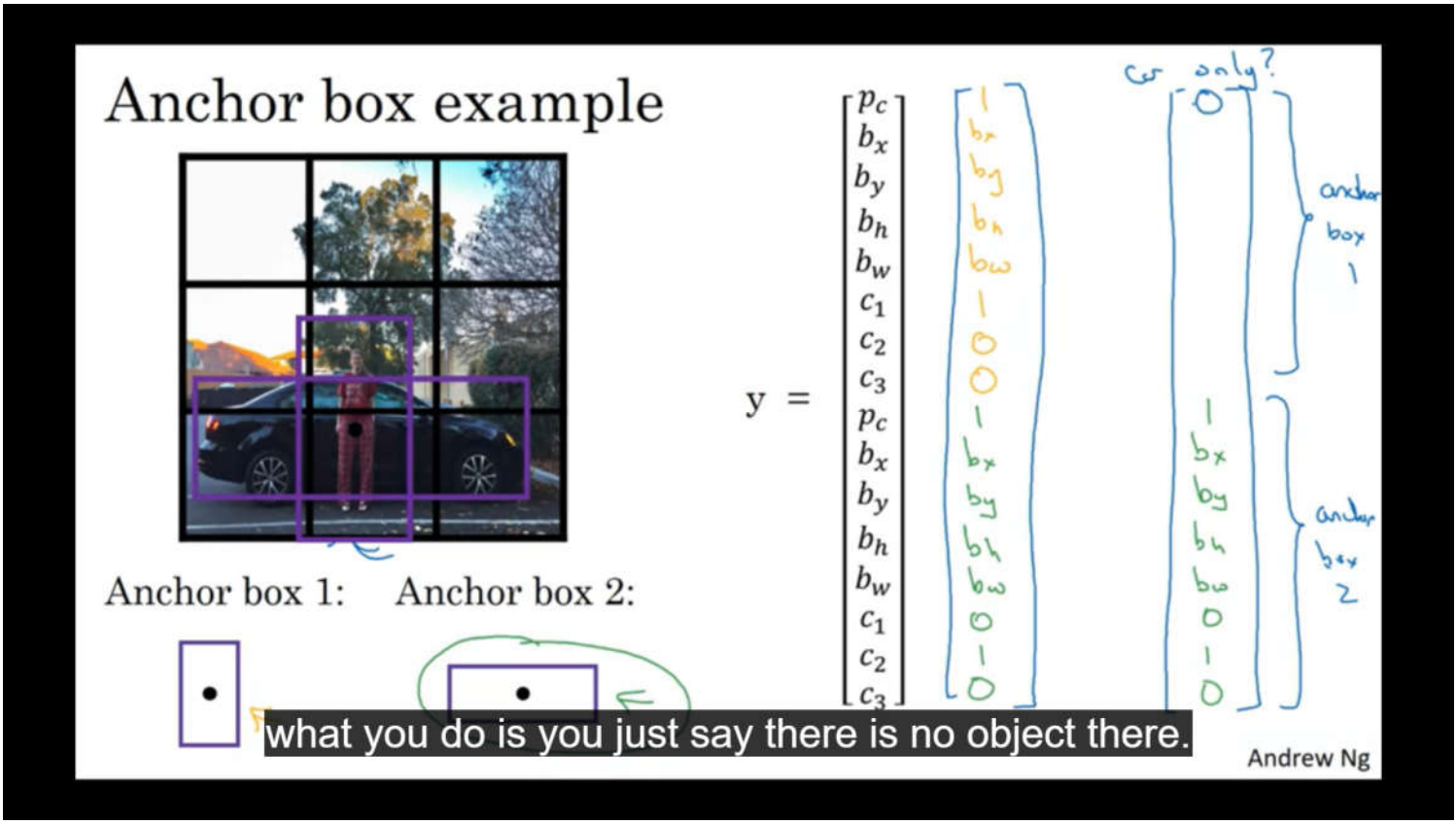
Because as you saw on the previous slide,

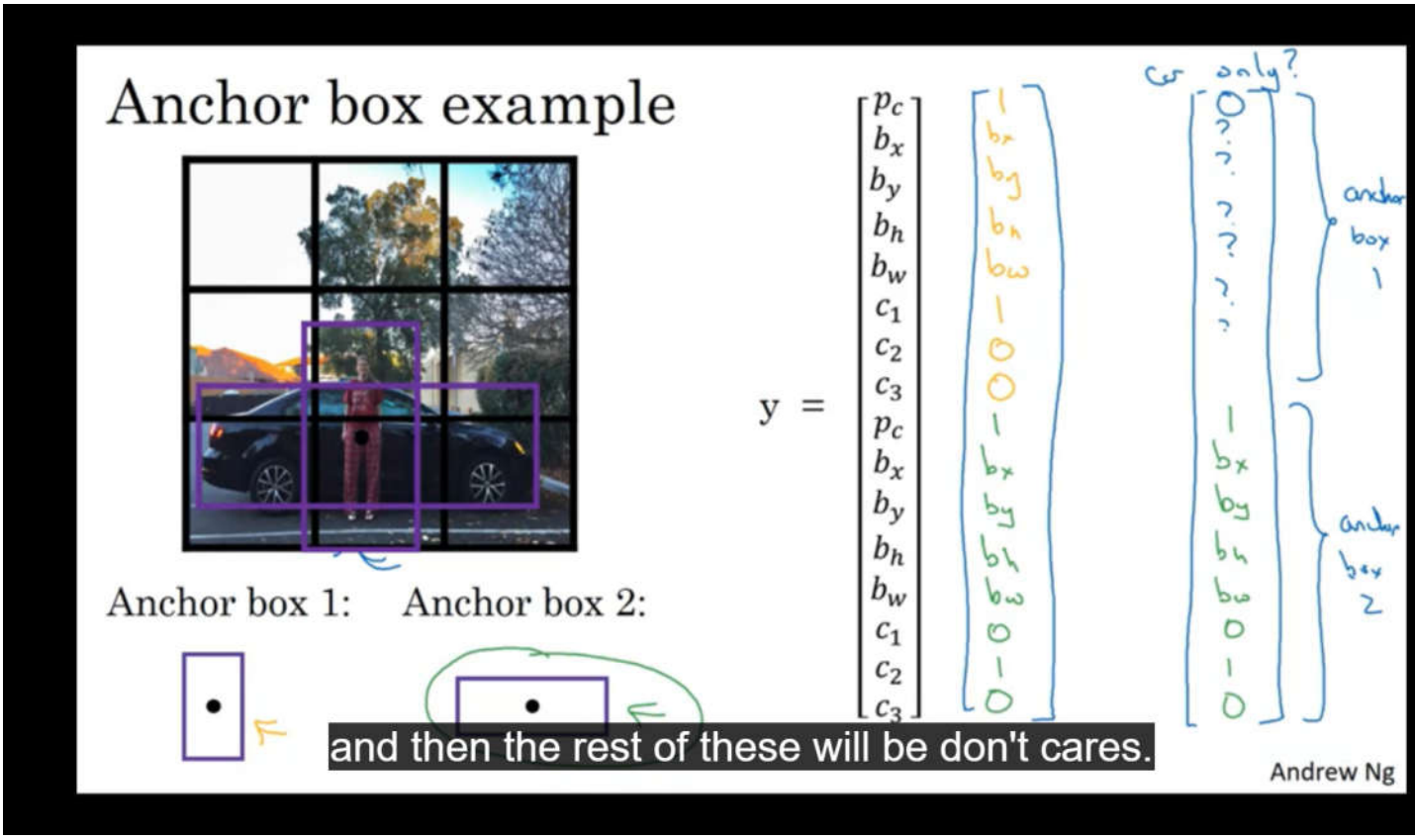
Andrew Ng

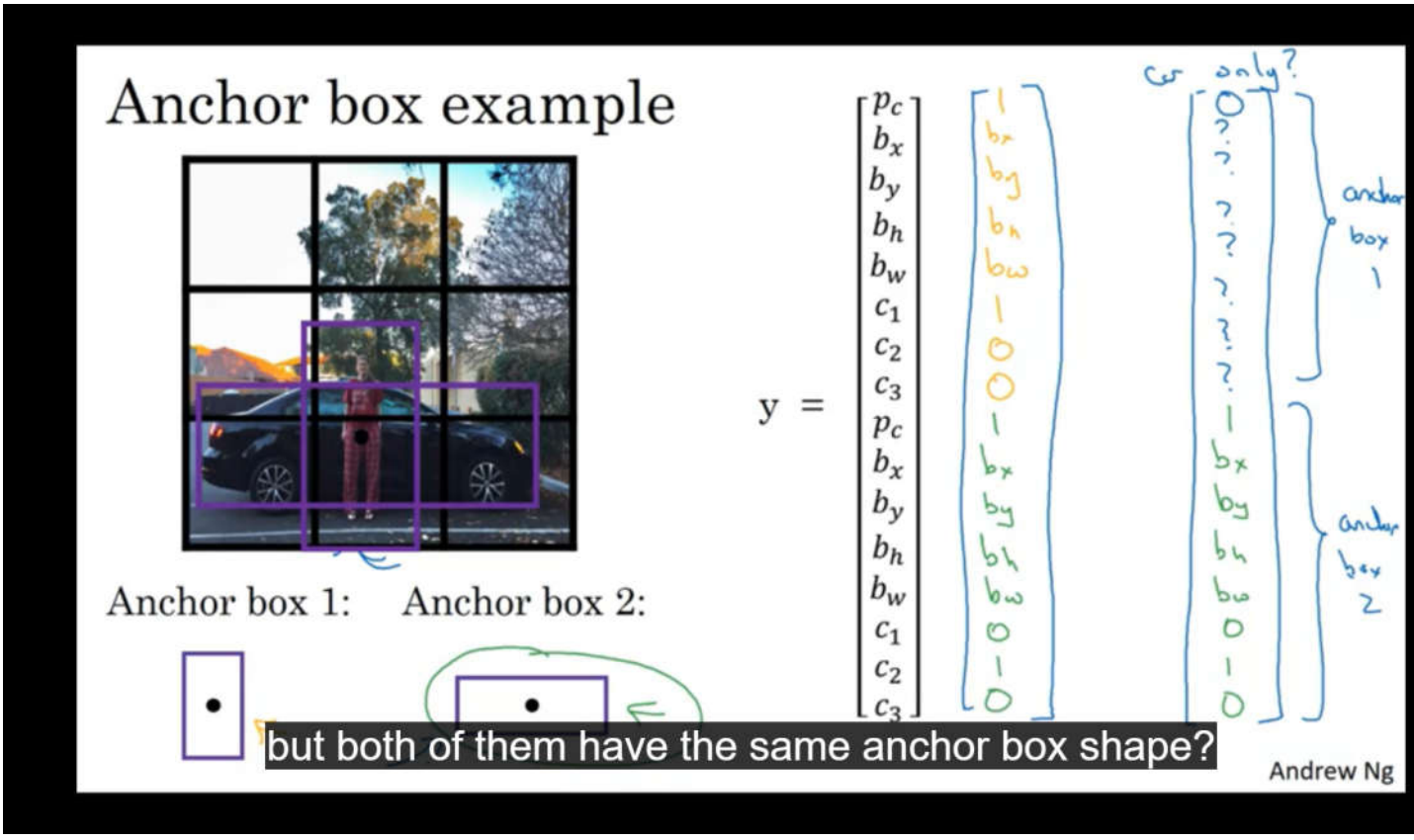

Anchor box example
y =
p_c b_x b_y b_h b_w c_1 c_2 c_3 p_c b_x b_y b_h b_w c_1 c_2 c_3
Anchor box 1:
Anchor box 2:
For this grid cell,
Andrew Ng


Anchor box example
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
1 bx by bh bw 1 0 0 1 bx by bh bw 0 1 0
Anchor box 1:
Anchor box 2:
And so that's the label Y for
Andrew Ng


Anchor box example
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
car only?
Anchor box 1:
Anchor box 2:
the car is still more similar to anchor box 2,
Andrew Ng


Anchor box example
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
car only?
anchor box 1
anchor box 2
Anchor box 1:
Anchor box 2:
what you do is you just say there is no object there.
Andrew Ng


Anchor box example
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
Anchor box 1:
Anchor box 2:
and then the rest of these will be don't cares.
Andrew Ng


Anchor box example
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
Anchor box 1:
Anchor box 2:
but both of them have the same anchor box shape?
Andrew Ng

deeplearning.ai

Object Detection

Putting it together: YOLO algorithm

You've already seen most of the components of object detection.

Training

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

First, let's see how you construct your training set.

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

Andrew Ng

0:11 / 7:01

Training

1 - pedestrian
2 - car
3 - motorcycle

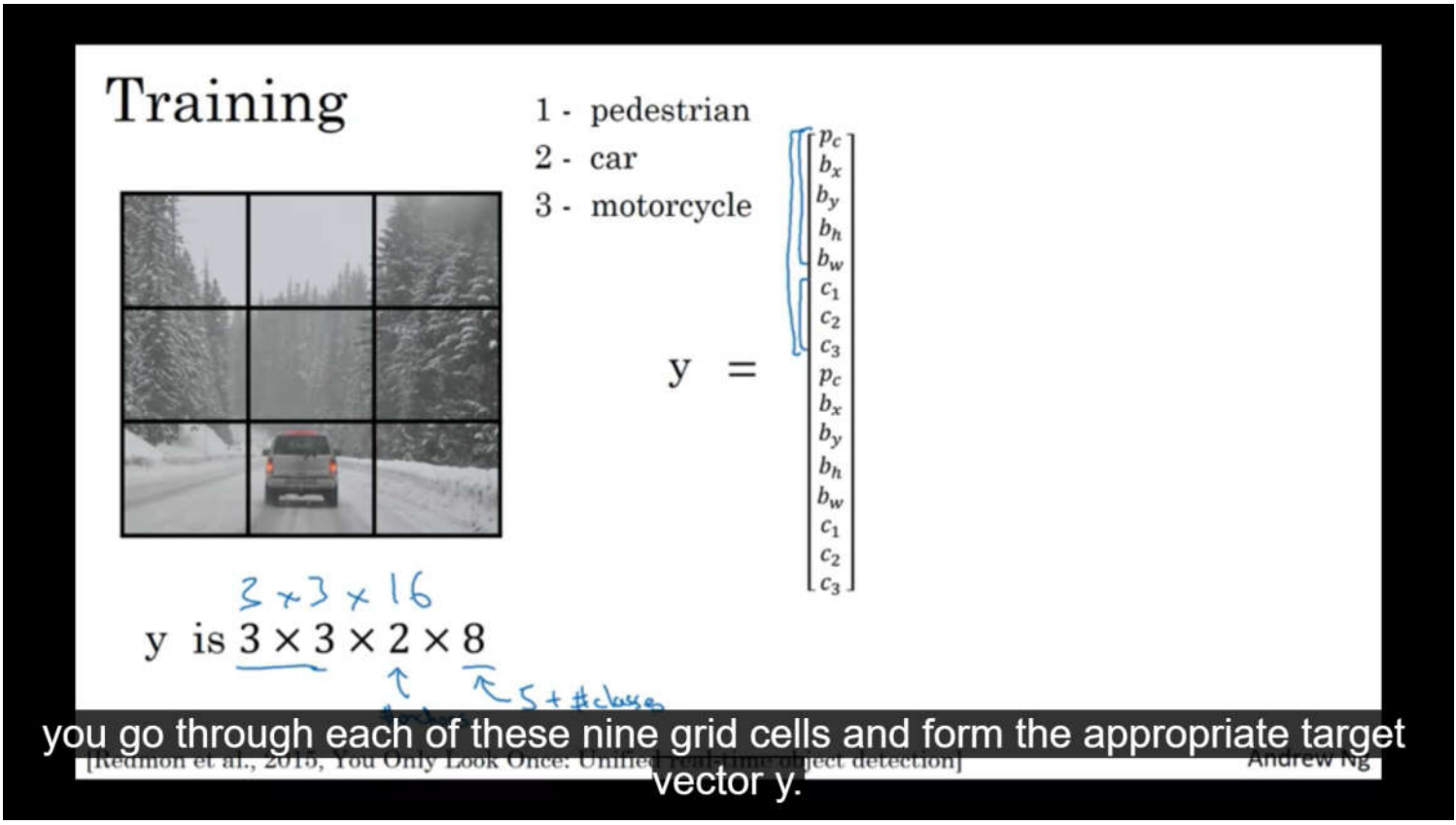
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

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

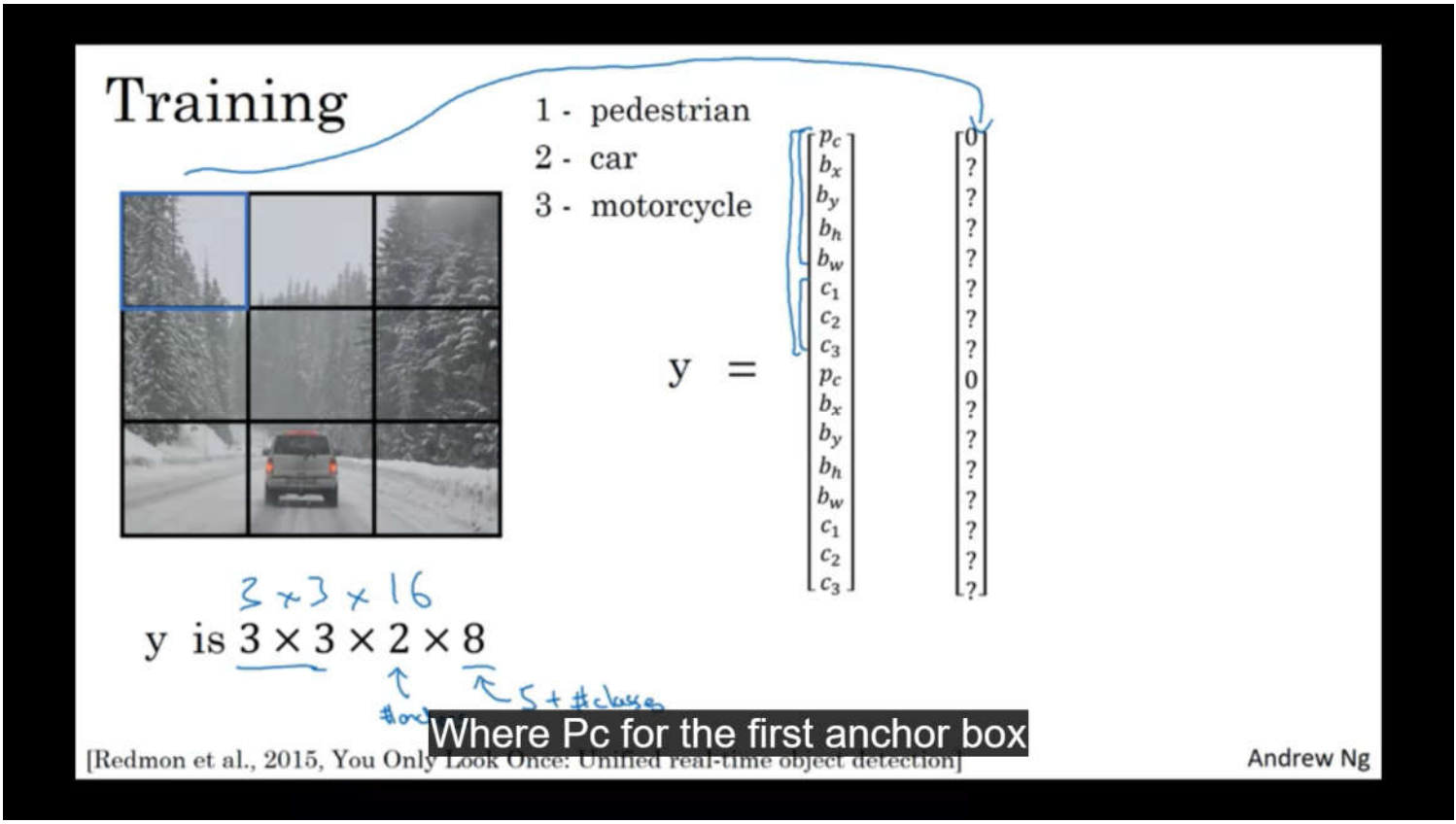
by two, this is the number of anchors,

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

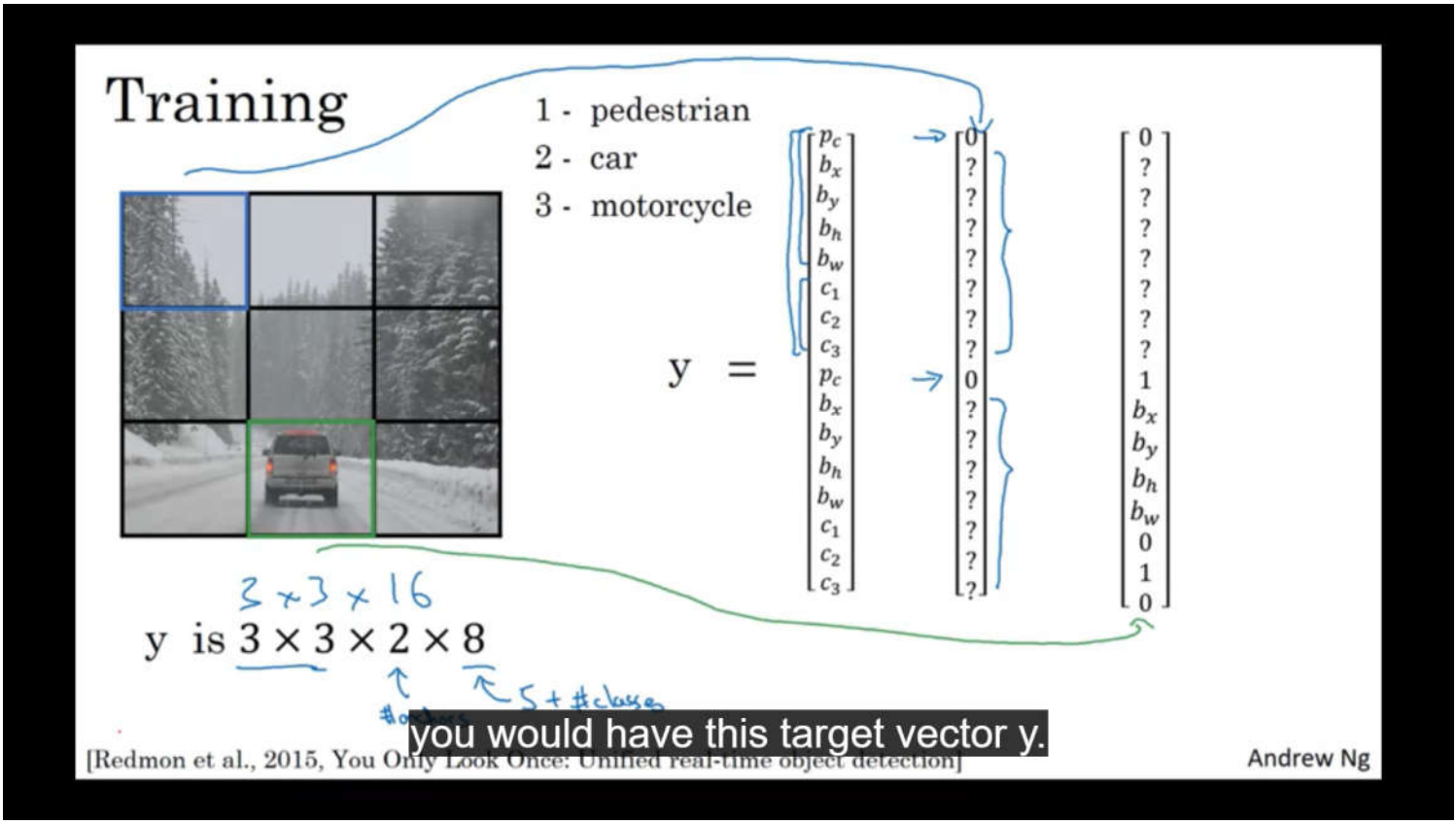
Andrew Ng

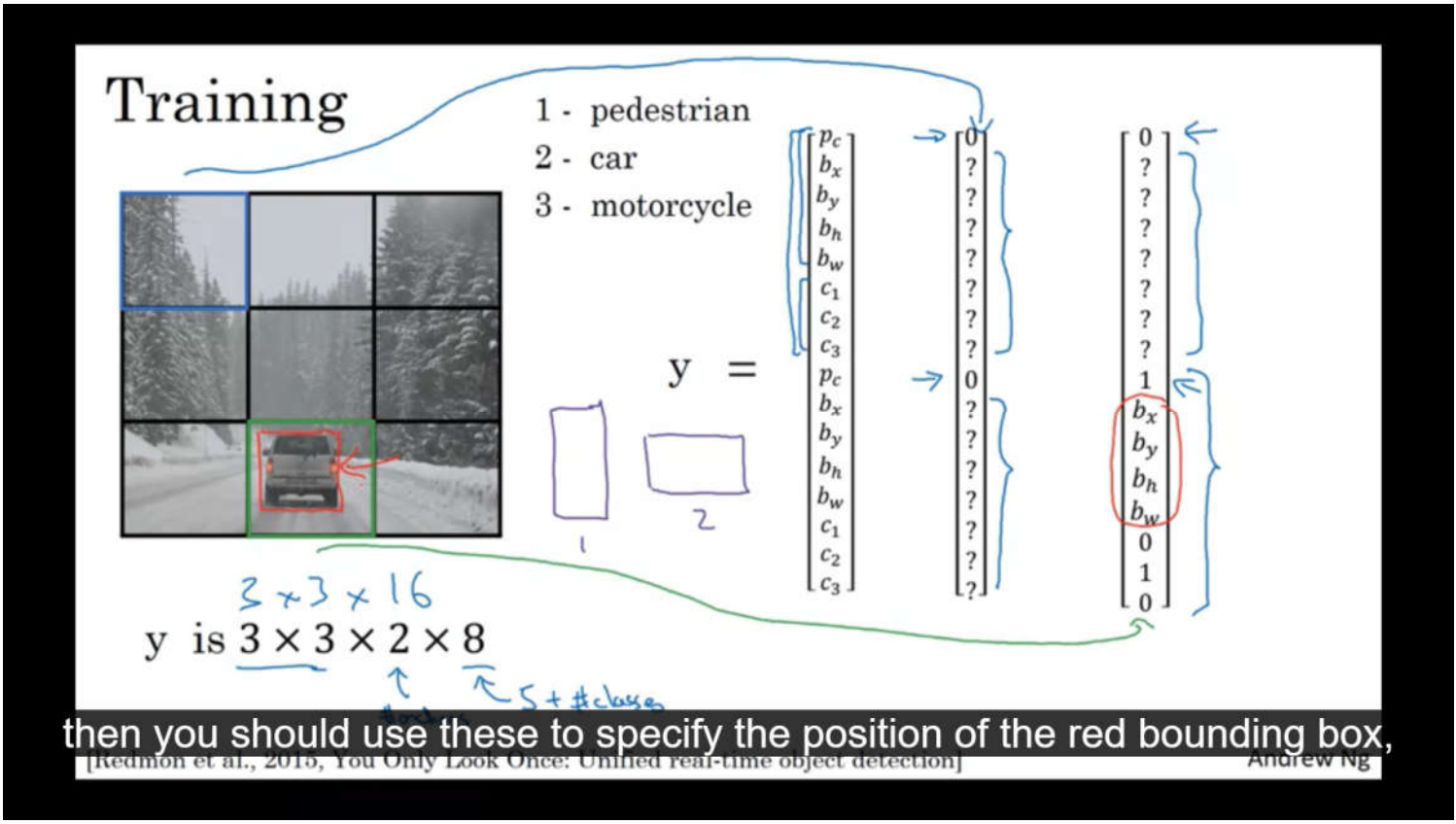


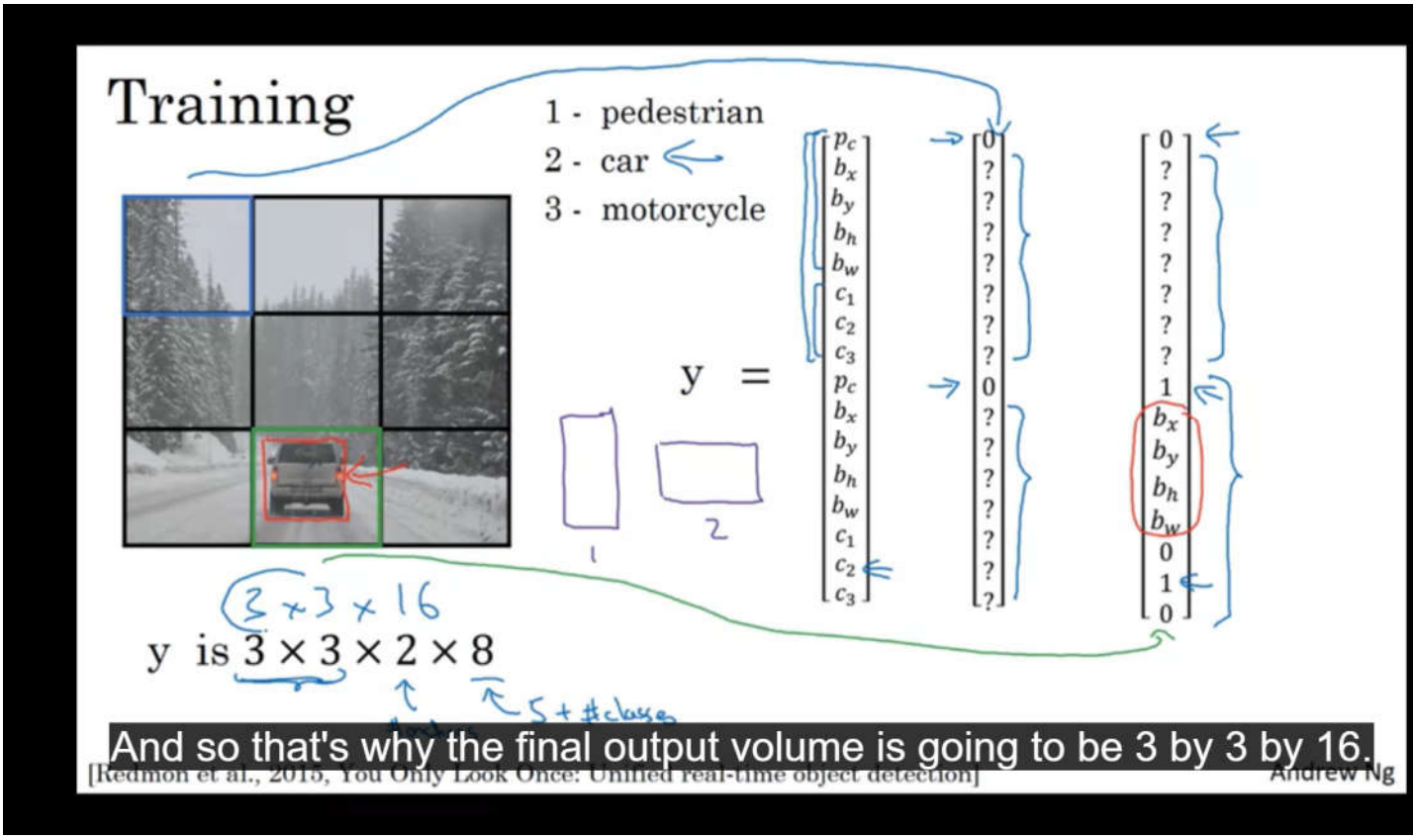
you go through each of these nine grid cells and form the appropriate target vector y.

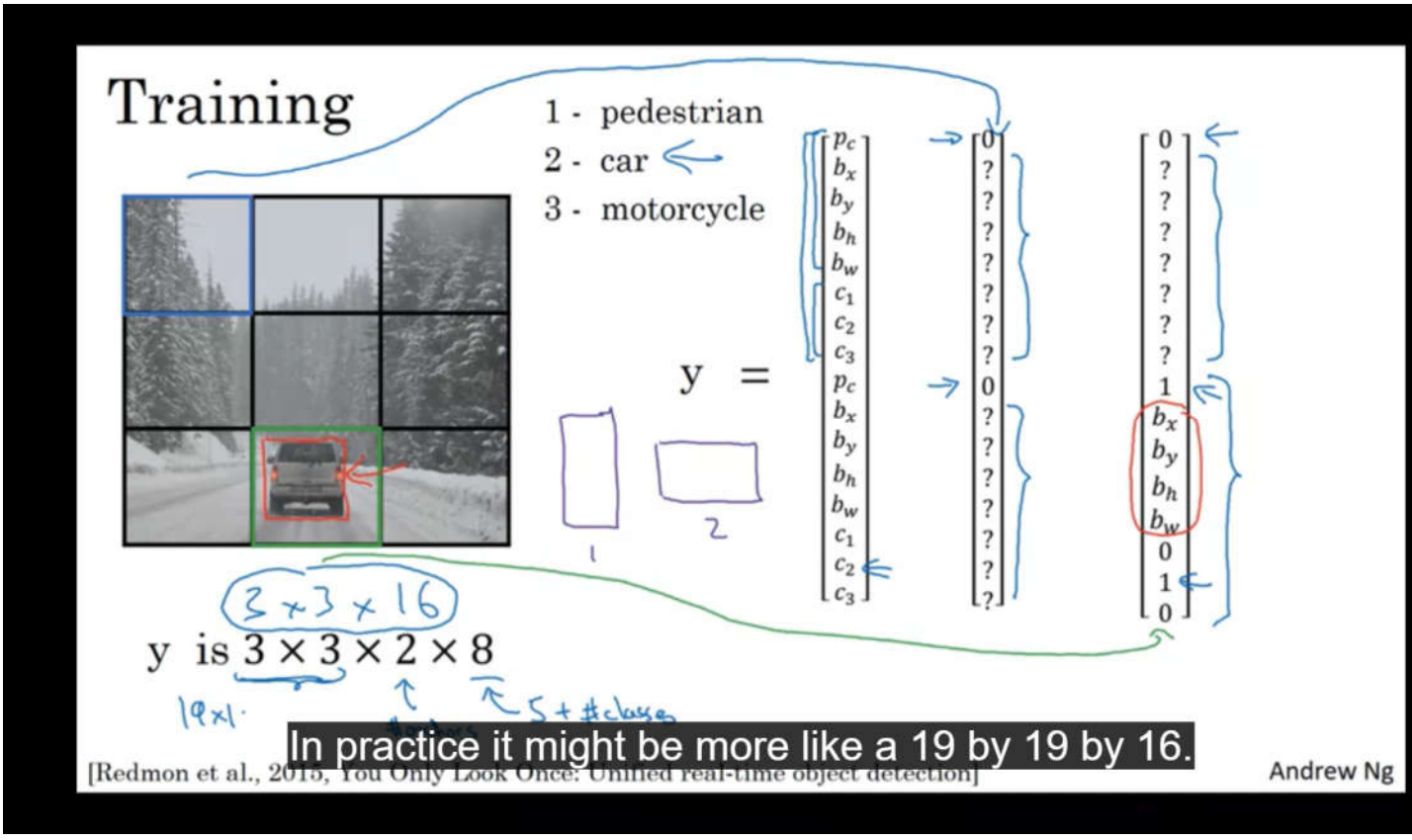


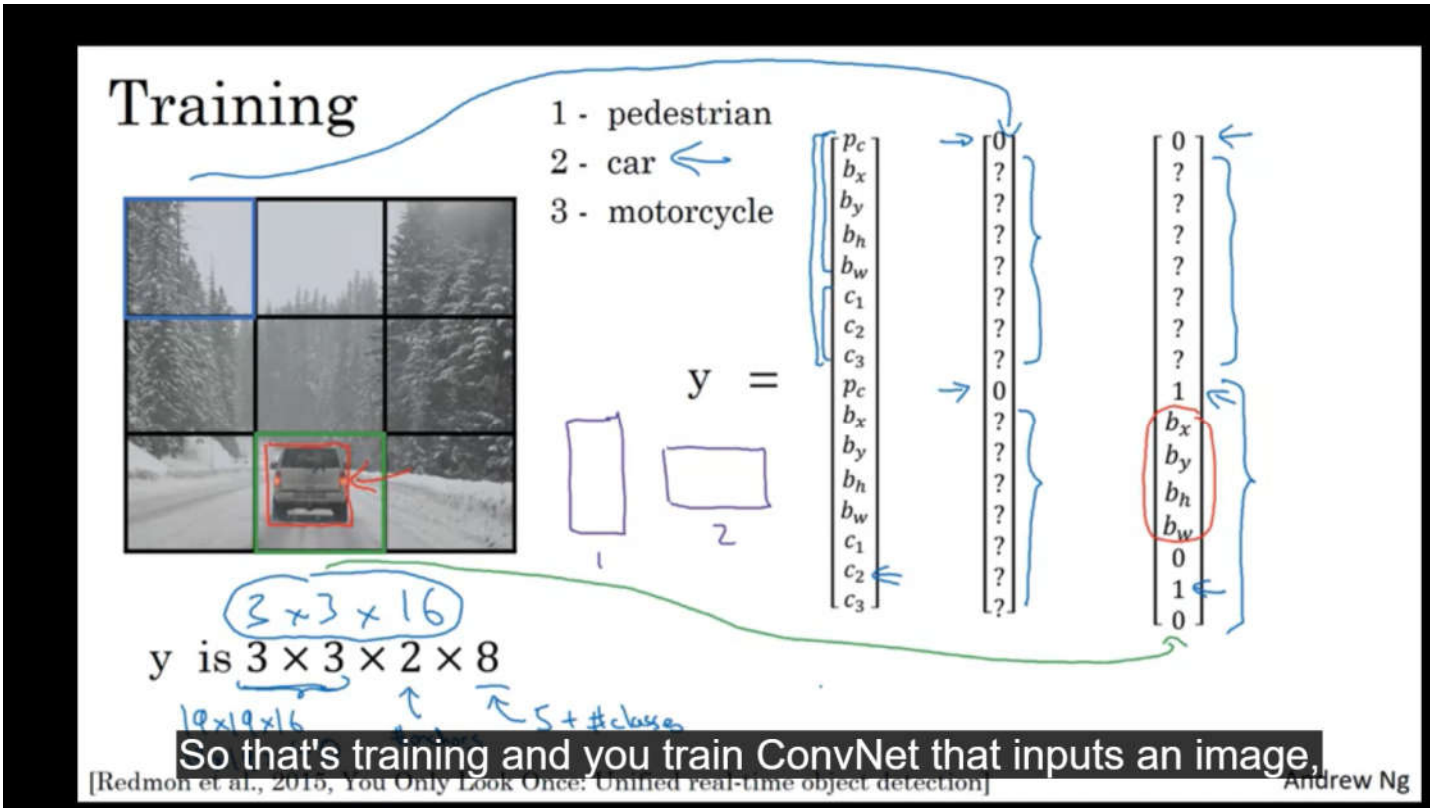
Where Pc for the first anchor box

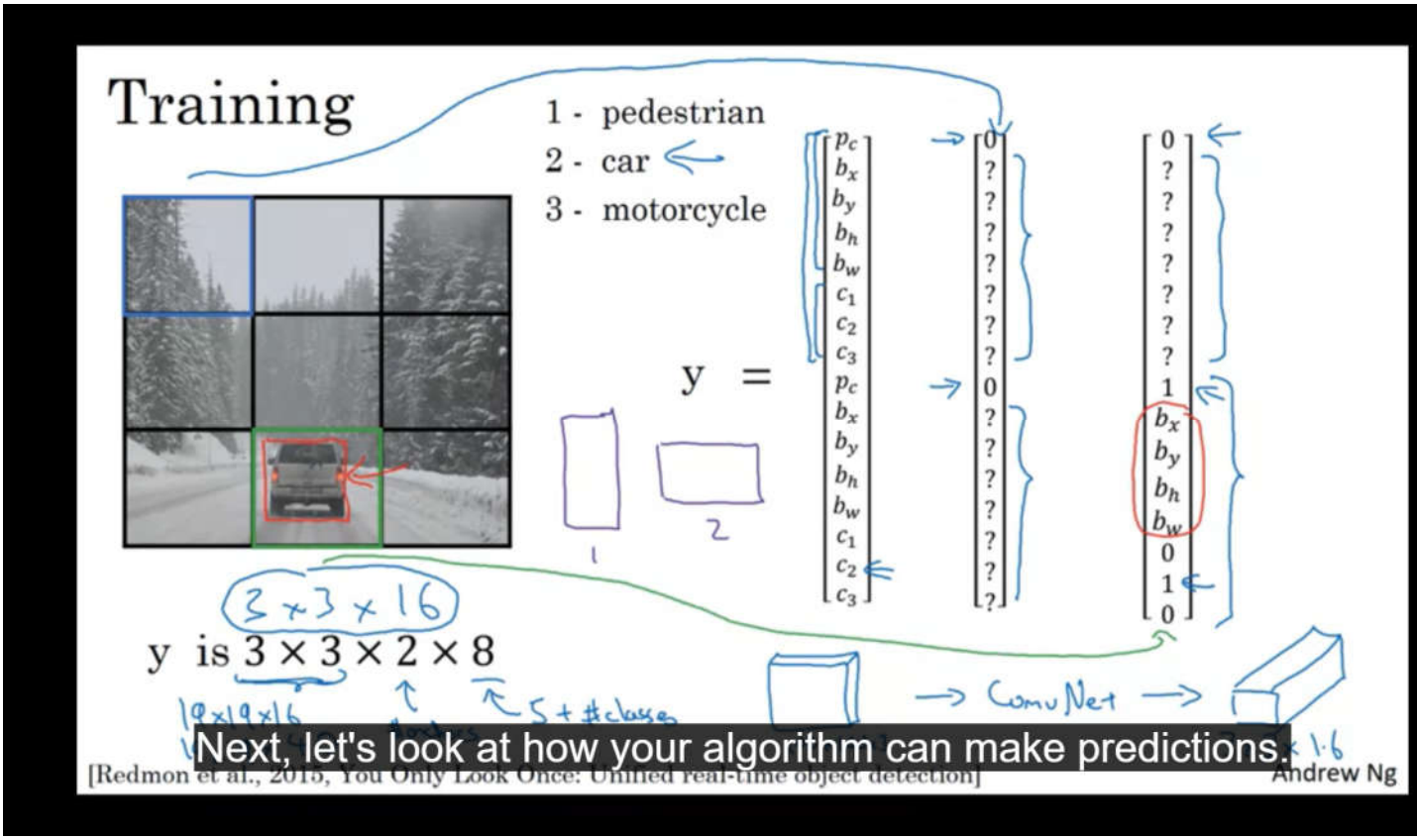

Training
1 - pedestrian
2 - car
3 - motorcycle
y =
y is 3 × 3 × 2 × 8
3 × 3 × 16
5 + #classes
you would have this target vector y.
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Andrew Ng

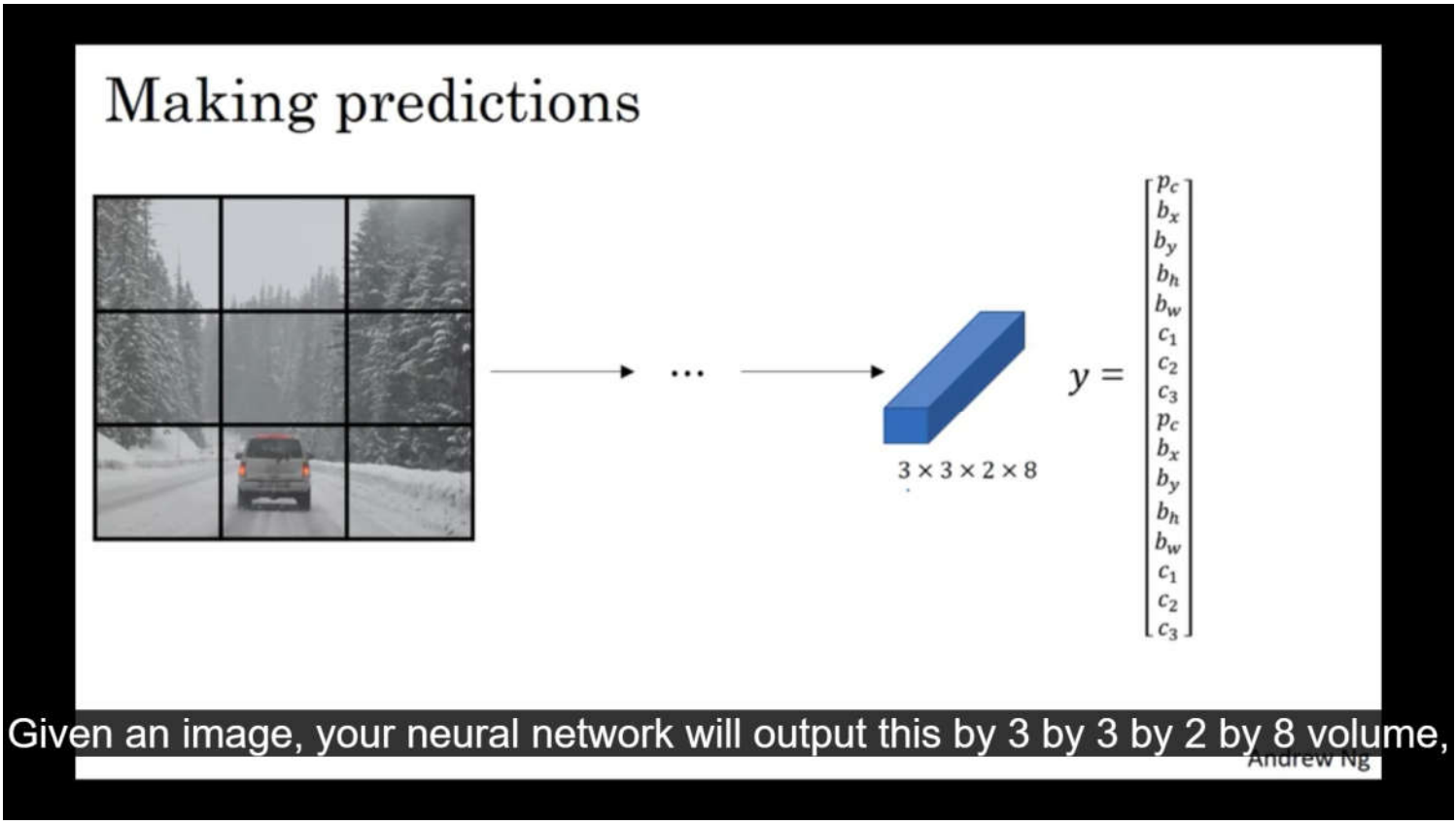

Training
1 - pedestrian
2 - car
3 - motorcycle
y =
y is 3 × 3 × 2 × 8
3 × 3 × 16
5 + #classes
then you should use these to specify the position of the red bounding box,
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Andrew Ng


Training
1 - pedestrian
2 - car
3 - motorcycle
y =
y is 3 × 3 × 2 × 8
3 × 3 × 16
5 + #classes
And so that's why the final output volume is going to be 3 by 3 by 16.
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Andrew Ng

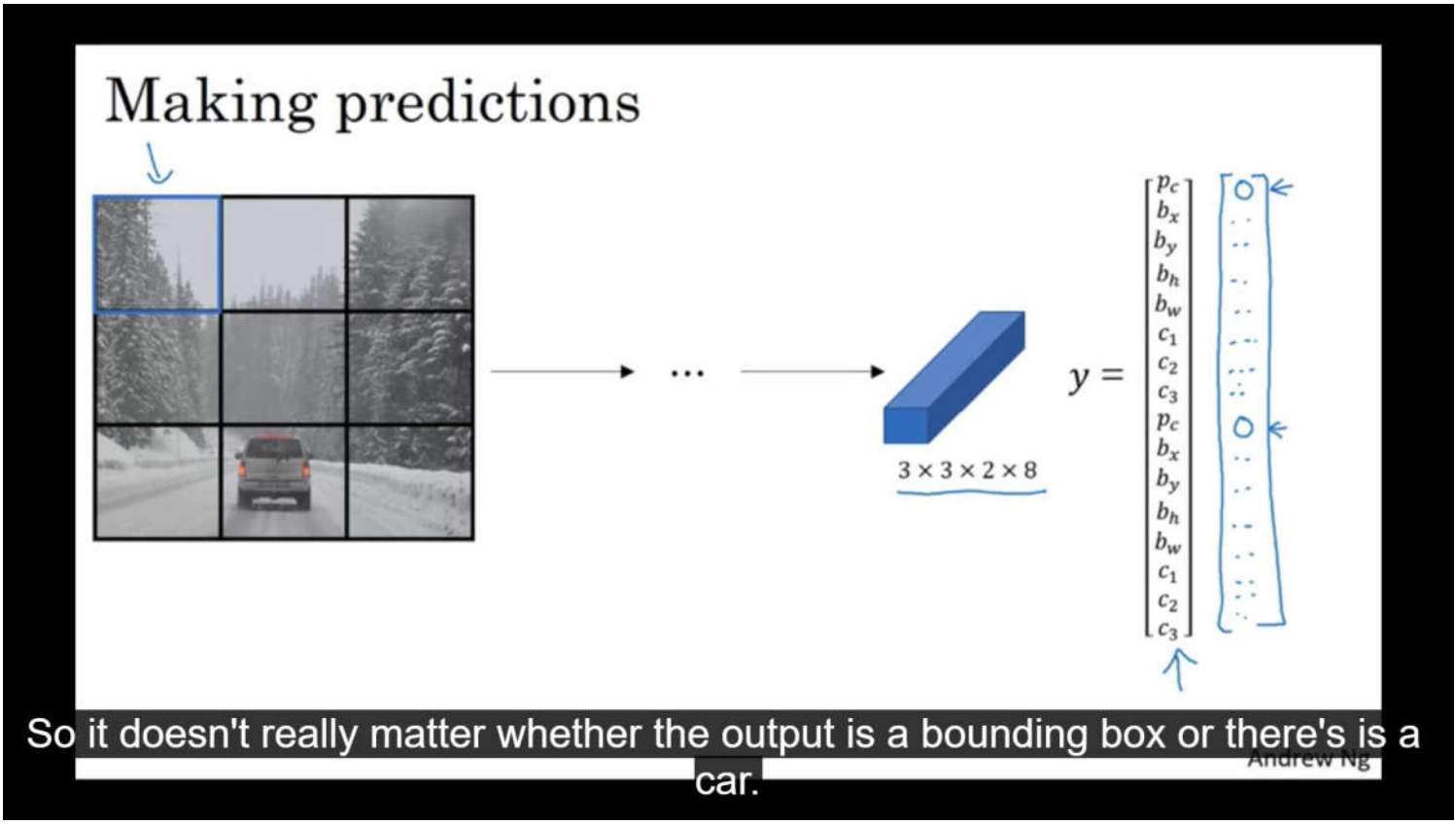

Training
1 - pedestrian
2 - car
3 - motorcycle
y =
y is 3 × 3 × 2 × 8
3 × 3 × 16
19×1
5 + #classes
In practice it might be more like a 19 by 19 by 16.
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Andrew Ng


Training
1 - pedestrian
2 - car
3 - motorcycle
y =
y is 3 × 3 × 2 × 8
3 × 3 × 16
19×19×16
5 + #classes
So that's training and you train ConvNet that inputs an image,
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Andrew Ng

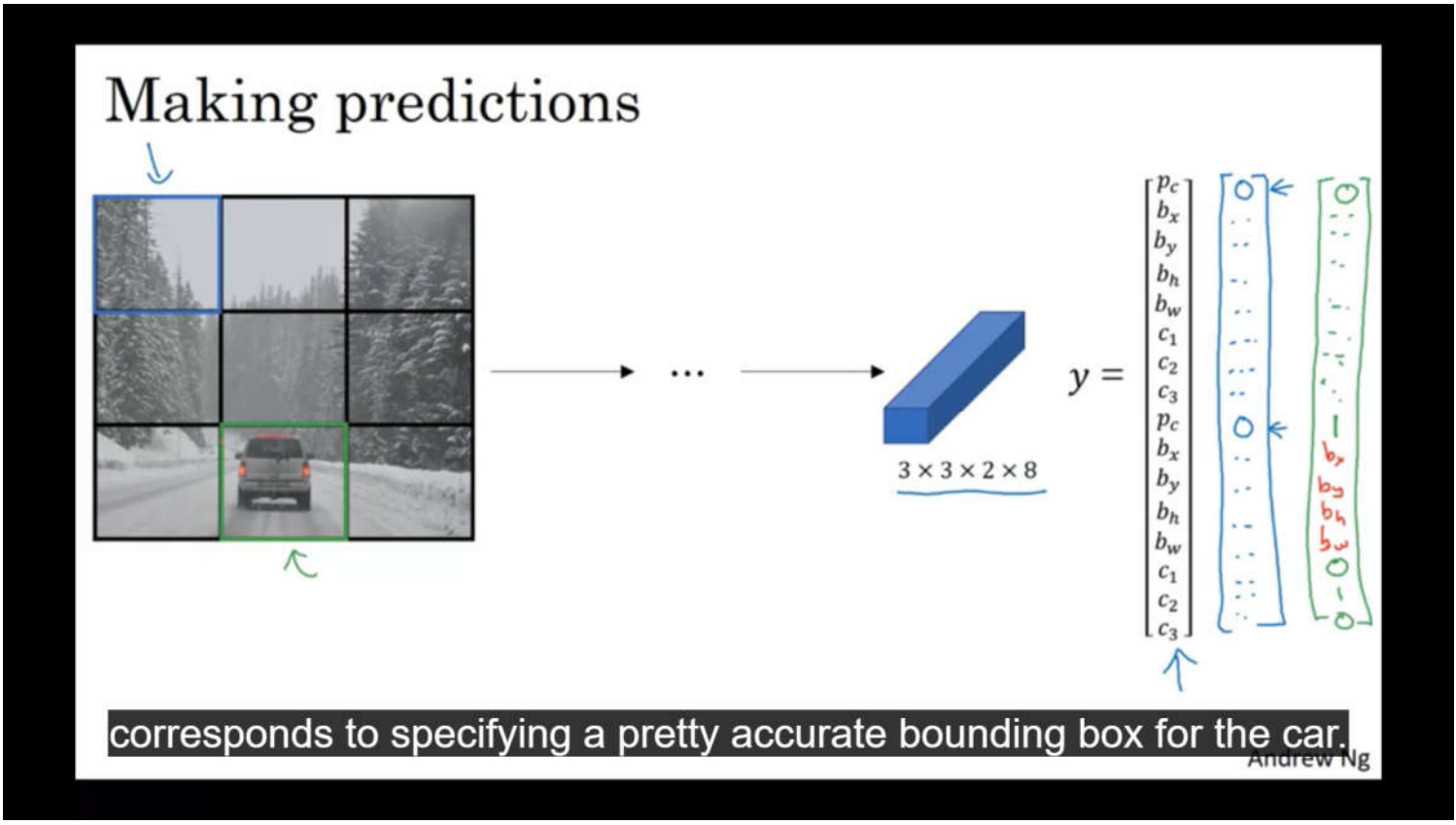

Training
1 - pedestrian
2 - car
3 - motorcycle
y =
y is 3 × 3 × 2 × 8
3 × 3 × 16
19×19×16
5 + #classes
ConvNet
Next, let's look at how your algorithm can make predictions.
[Redmon et al., 2015, You Only Look Once: Unified real-time object detection]
Andrew Ng

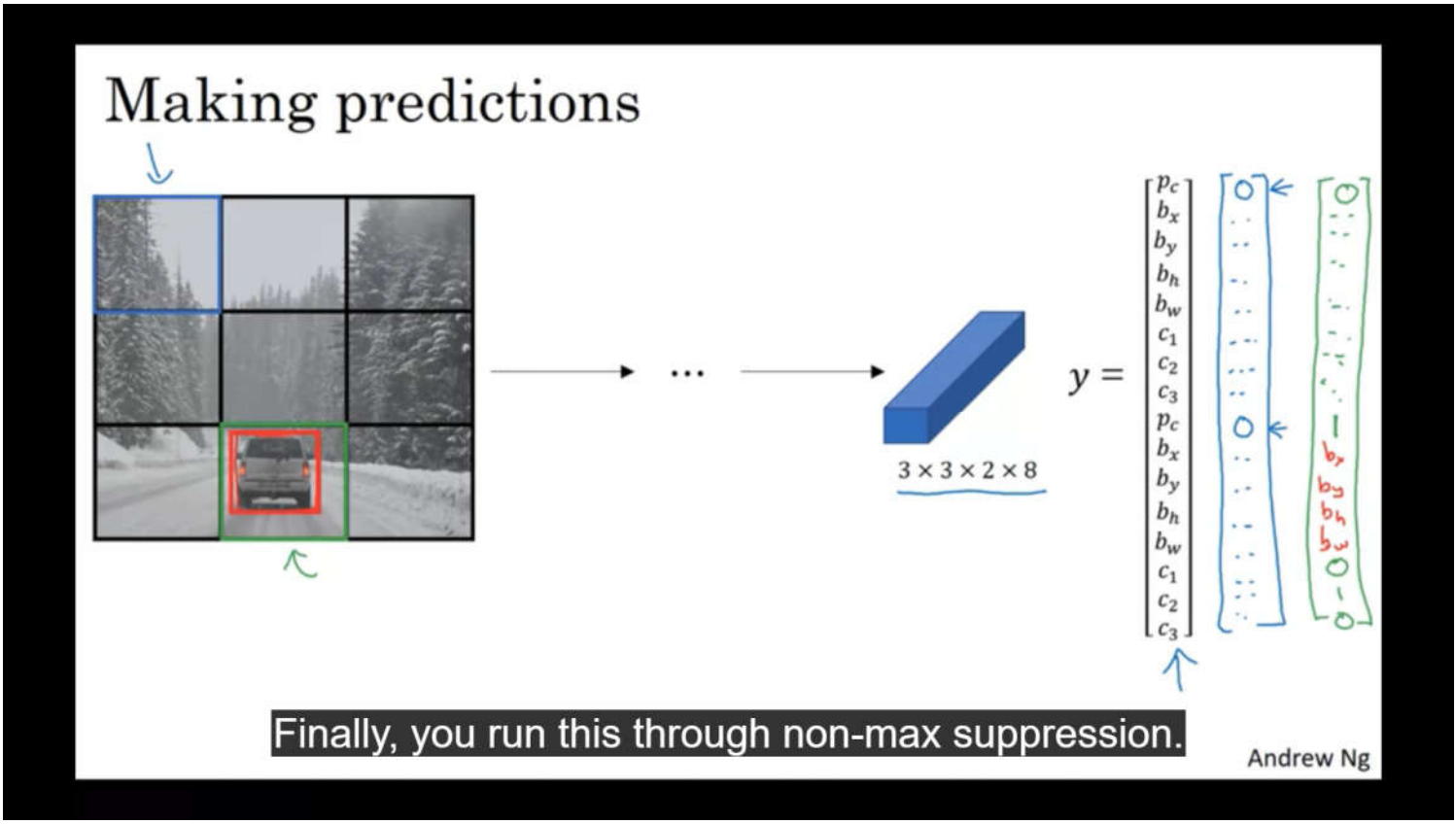


Given an image, your neural network will output this by 3 by 3 by 2 by 8 volume,



So it doesn't really matter whether the output is a bounding box or there's is a car.


Making predictions
...
3 × 3 × 2 × 8
y =
corresponds to specifying a pretty accurate bounding box for the car.
Andrew Ng


Making predictions
...
3 × 3 × 2 × 8
y =
Finally, you run this through non-max suppression.
Andrew Ng

Outputting the non-max supressed outputs

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

then for each of the non-grid cells,

Andrew Ng

Outputting the non-max supressed outputs

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

So let's say, those are the bounding boxes you get.

Andrew Ng

Outputting the non-max supressed outputs

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

Next, you then get rid of the 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.

And then finally if you have three classes you're trying to detect,

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.

What you do is, for each of the three classes,

Andrew Ng

coursera.org/learn/convolutional-neural-networks/supplement/l4SFt/yolo-algorithm-correction

coursera | Srinivasa Rao Dhanikonda

Convolutional Neural Networks > Week 3 > YOLO algorithm *CORRECTION*

Prev | Next

- Video: Object Detection 5 min
- Video: Convolutional Implementation of Sliding Windows 11 min
- Reading: Convolutional Implementation of Sliding Windows *CORRECTION* 1 min
- Video: Bounding Box Predictions 14 min
- Video: Intersection Over Union 4 min
- Video: Non-max Suppression 8 min
- Video: Anchor Boxes 9 min
- Video: YOLO Algorithm 7 min
- Reading: YOLO algorithm

At time 5:00, for the slide titled "Outputting the non-max suppressed output" the text should read "For each grid **cell**" instead of "For each grid call".

Mark as completed

coursera.org/learn/convolutional-neural-networks/exam/laJVI/detection-algorithms/attempt

Detection algorithms

Graded Quiz • 30 min

Due May 18, 11:59 AM +05

1. You are building a 3-class object classification and localization algorithm. The classes are: pedestrian (c=1), car (c=2), motorcycle (c=3). What would be the label for the following image? Recall $y = [p_c, b_x, b_y, b_h, b_w, c_1, c_2, c_3]$

1 point

- (selected) $y = [1, 0.3, 0.7, 0.3, 0.3, 0, 1, 0]$
- $y = [1, 0.7, 0.5, 0.3, 0.3, 0, 1, 0]$
- $y = [1, 0.3, 0.7, 0.5, 0.5, 0, 1, 0]$
- $y = [1, 0.3, 0.7, 0.5, 0.5, 1, 0, 0]$
- $y = [0, 0.2, 0.4, 0.5, 0.5, 0, 1, 0]$

coursera.org/learn/convolutional-neural-networks/exam/laJVI/detection-algorithms/attempt

Detection algorithms

Graded Quiz • 30 min

Due May 18, 11:59 AM +05

2. Continuing from the previous problem, what should y be for the image below? Remember that "?" means "don't care", which means that the neural network loss function won't care what the neural network gives for that component of the output. As before, $y = [p_c, b_x, b_y, b_h, b_w, c_1, c_2, c_3]$.

1 point

- (selected) $y = [0, ?, ?, ?, ?, ?, ?, ?]$
- $y = [1, ?, ?, ?, ?, 0, 0, 0]$
- $y = [?, ?, ?, ?, ?, ?, ?, ?]$
- $y = [1, ?, ?, ?, ?, ?, ?, ?]$
- $y = [0, ?, ?, ?, ?, 0, 0, 0]$

coursera.org/learn/convolutional-neural-networks/exam/laJVI/detection-algorithms/attempt

Detection algorithms

Graded Quiz • 30 min

Due May 18, 11:59 AM +05

3. You are working on a factory automation task. Your system will see a can of soft-drink coming down a conveyor belt, and you want it to take a picture and decide whether (i) there is a soft-drink can in the image, and if so (ii) its bounding box. Since the soft-drink can is round, the bounding box is always square, and the soft drink can always appears as the same size in the image. There is at most one soft drink can in each image. Here're some typical images in your training set:

1 point

What is the most appropriate set of output units for your neural network?

- ○ Logistic unit (for classifying if there is a soft-drink can in the image)
- ◉ Logistic unit, b_x and b_y
- ○ Logistic unit, b_x, b_y, b_h (since $b_w = b_h$)
- ○ Logistic unit, b_x, b_y, b_h, b_w

coursera.org/learn/convolutional-neural-networks/exam/laJVI/detection-algorithms/attempt

Detection algorithms

Graded Quiz • 30 min

Due May 18, 11:59 AM +05

- ○ Logistic unit, b_x, b_y, b_h, b_w

4. If you build a neural network that inputs a picture of a person's face and outputs N landmarks on the face (assume the input image always contains exactly one face), how many output units will the network have?

1 point

- ◉ N — N wrong
- ○ 2N — answer is 2N
- ○ 3N
- ○ N^2

5. When training one of the object detection systems described in lecture, you need a training set that contains many pictures of the object(s) you wish to detect. However, bounding boxes do not need to be provided in the training set, since the algorithm can learn to detect the objects by itself.

1 point

- ○ True
- ◉ False

6. Suppose you are applying a sliding windows classifier (non-convolutional implementation). Increasing the stride would tend to increase accuracy, but decrease computational cost.

1 point

coursera.org/learn/convolutional-neural-networks/exam/laJVI/detection-algorithms/attempt

Detection algorithms

Graded Quiz • 30 min

Due May 18, 11:59 AM +05

- ○ N^2

5. When training one of the object detection systems described in lecture, you need a training set that contains many pictures of the object(s) you wish to detect. However, bounding boxes do not need to be provided in the training set, since the algorithm can learn to detect the objects by itself. (1 point)

   - ○ True
   - ◉ False

6. Suppose you are applying a sliding windows classifier (non-convolutional implementation). Increasing the stride would tend to increase accuracy, but decrease computational cost. (1 point)

   - ○ True
   - ◉ False

7. In the YOLO algorithm, at training time, only one cell ---the one containing the center/midpoint of an object--- is responsible for detecting this object. (1 point)

   - ◉ True
   - ○ False

coursera.org/learn/convolutional-neural-networks/exam/laJVI/detection-algorithms/attempt

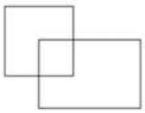
Detection algorithms

Graded Quiz • 30 min

Due May 18, 11:59 AM +05

- ○ False

8. What is the IoU between these two boxes? The upper-left box is 2x2, and the lower-right box is 2x3. The overlapping region is 1x1. (1 point)



   - ○ 1/6
   - ◉ 1/9
   - ○ 1/10
   - ○ None of the above

9. Suppose you run non-max suppression on the predicted boxes above. The parameters you use for non-max suppression are that boxes with probability ≤ 0.4 are discarded, and the IoU threshold for deciding if two boxes overlap is 0.5. How many boxes will remain after non-max suppression? (1 point)

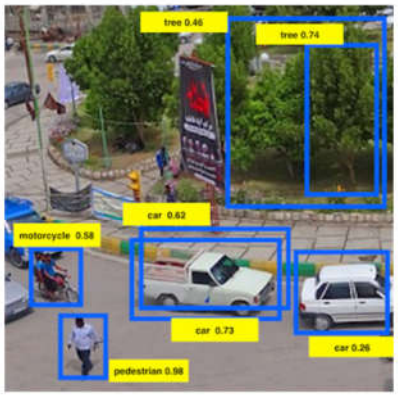
coursera.org/learn/convolutional-neural-networks/exam/laJVI/detection-algorithms/attempt

Detection algorithms

Graded Quiz • 30 min

Due May 18, 11:59 AM +05

two boxes overlap is 0.5. How many boxes will remain after non-max suppression?



- ○ 3
- ○ 4
- ◉ 5
- ○ 6

coursera.org/learn/convolutional-neural-networks/exam/laJVI/detection-algorithms/attempt

Detection algorithms

Graded Quiz • 30 min

Due May 18, 11:59 AM +05

- ○ 3
- ○ 4
- ◉ 5
- ○ 6
- ○ 7

10. Suppose you are using YOLO on a 19x19 grid, on a detection problem with 20 classes, and with 5 anchor boxes. During training, for each image you will need to construct an output volume y as the target value for the neural network; this corresponds to the last layer of the neural network. (y may include some "?", or "don't cares"). What is the dimension of this output volume?

1 point

- ○ 19x19x(5x20)
- ◉ 19x19x(5x25)
- ○ 19x19x(20x25)
- ○ 19x19x(25x20)