

EE655: Computer Vision & Deep Learning

Lecture 11

Koteswar Rao Jerripothula, PhD
Department of Electrical Engineering
IIT Kanpur

Credit: Andrew Ng

Lecture Outline

Object Localization: Classification with Localization



Detection multiple objects of same class

Sliding Window via CNN itself

You Only Look Once (YOLO) Object Detection

Classification with Localization

Assumption: Only one object is present.

Given an image we want to learn:

- the class of the image
- where is the object located in the image.

We need to detect **a class** and **a rectangle** of where that object is.

Image Classification



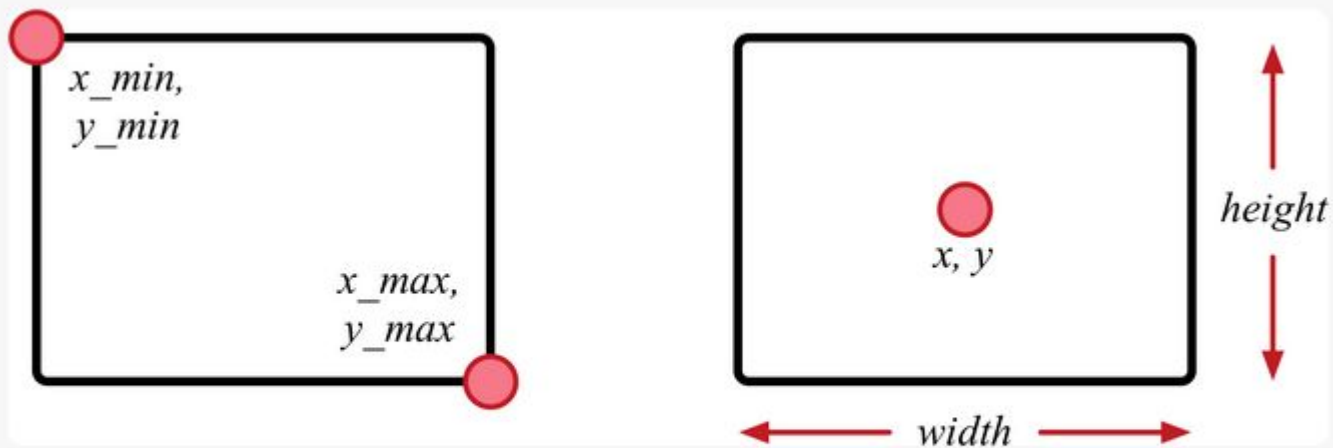
Cat

Classification with Localization

Only for
single
objects



Cat + Locations



The two types of bounding boxes

Classification with localization



Target label

- Pc Probability of object presence
- Bx X-coordinate of center of bounding box
- By Y-coordinate of center of bounding box
- Bw Width of the bounding box
- Bh Height of the bounding box
- C1 Probability of object belonging to class 1
- C2 Probability of object belonging to class 2
- C3 Probability of object belonging to class 3

Target Labels & Loss Function

- 1 - pedestrian
- 2 - car
- 3 - motorcycle



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

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

$$\begin{cases} (\hat{y}_1 - y_1)^2 & \text{if } y_1 = 0 \end{cases}$$

Squared
Error

1
Bx
By
Bh
Bw
0
1
0

0
?
?
?
?
?
?
?

?: Don't Care

Better Loss:

For Pc: Logistic Regression Loss

For Bx,By,Bh,Bw: Mean Squared Error Loss

For C1-C3: Cross Entropy Loss

For Pc=1

Sum of all Losses

For Pc=0

Only loss for Pc

Lecture Outline

Object Localization: Classification with Localization

Detection multiple objects of same class 

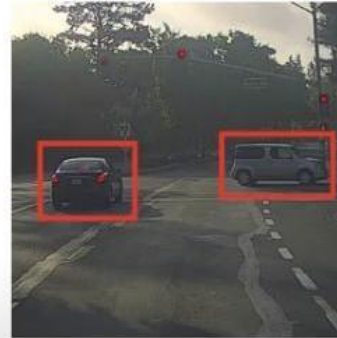
Sliding Window via CNN itself






You Only Look Once (YOLO) Object Detection

Object Detection through sliding window concept

Car detection example

Let's try to detect multiple cars.



Training set:		To develop a classifier
Cropped Images	x	y
		1
		1
		1
		0
		0

Sliding Windows Detection

The idea is to slide a window to crop an image portion out to check it through a CNN to see if there is a car inside the window.

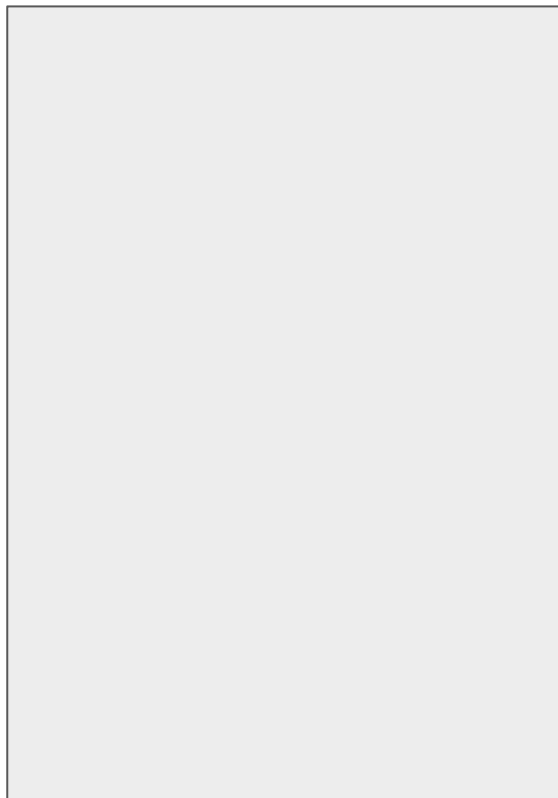
A window with a car becomes the required bounding box.

Drawback:
Computationally
Heavy



→ ConvNet → y

Sliding Windows at multiple sizes and multiple aspect ratios



Lecture Outline

Object Localization: Classification with Localization

Detection multiple objects of same class

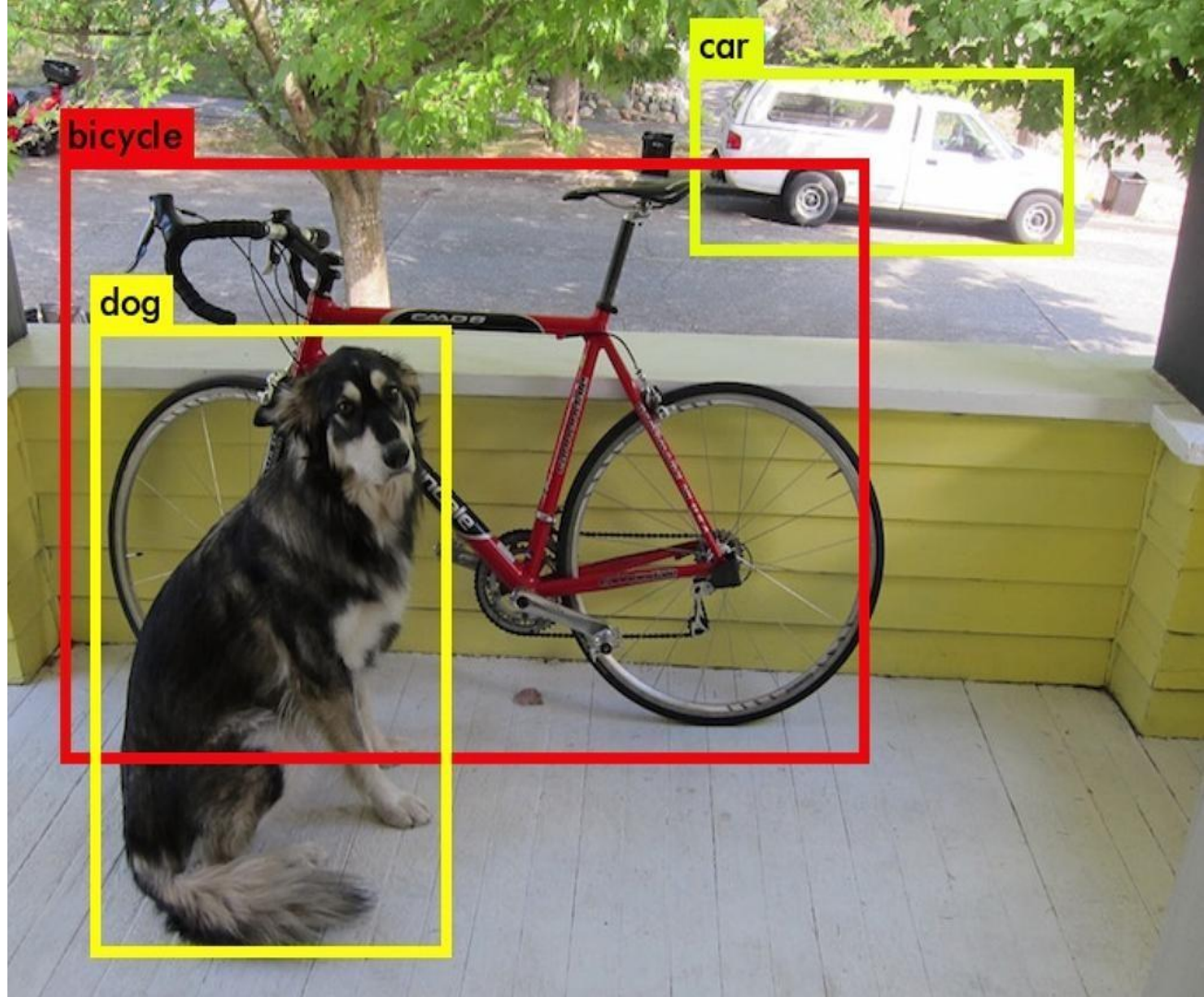
Sliding Window via CNN itself 

You Only Look Once (YOLO) Object Detection

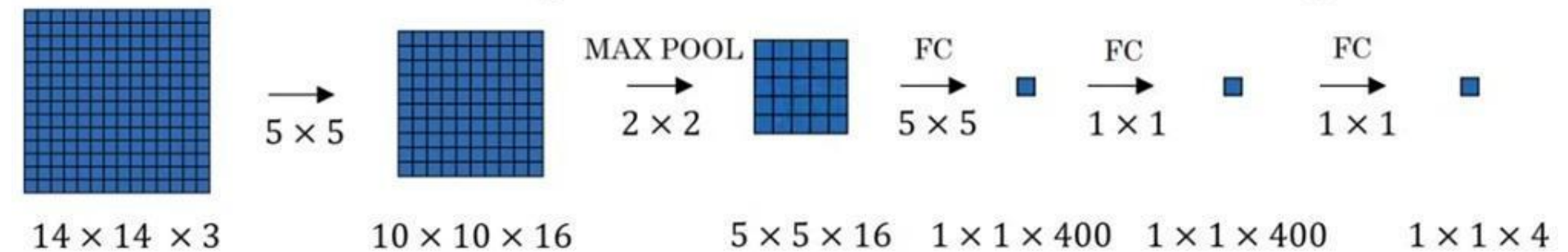
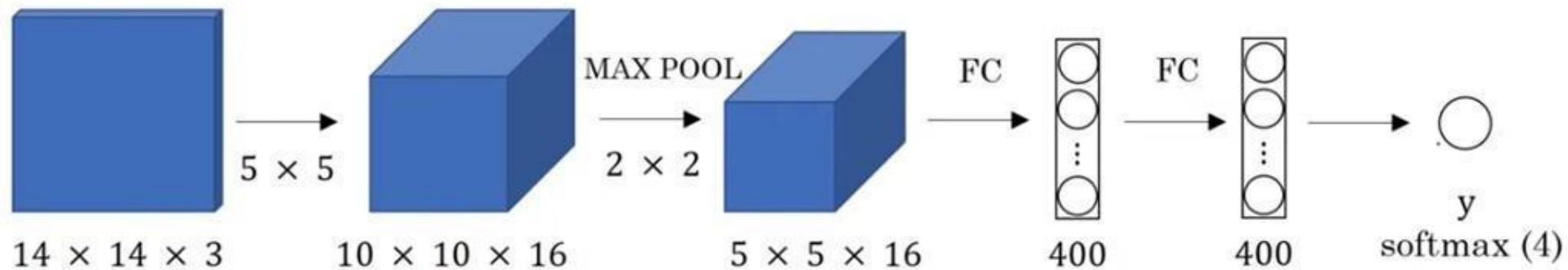
Object Detection

Multiple objects

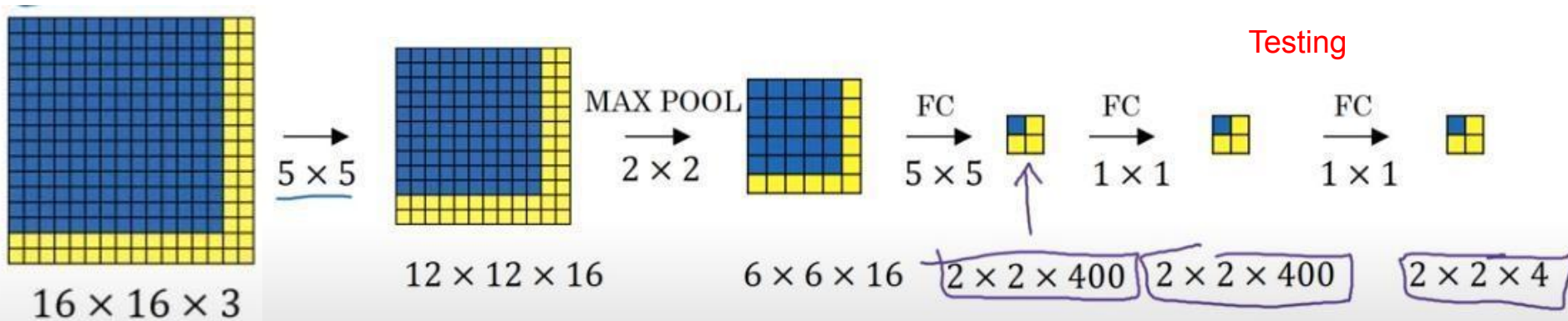
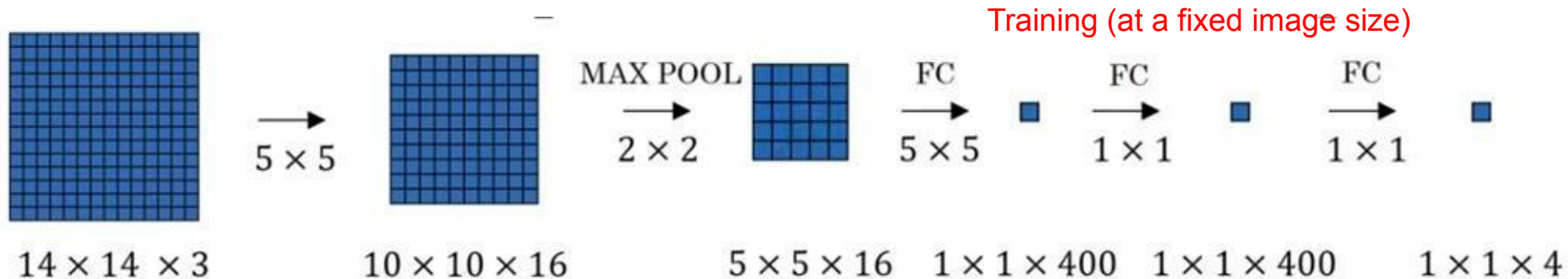
Multiple object classes

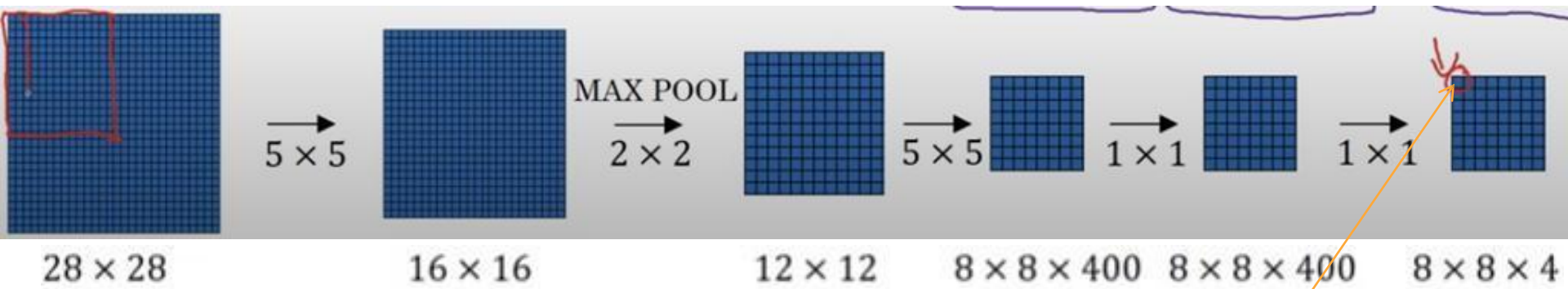


Converting a CNN into FCN (for an efficient sliding window implementation)

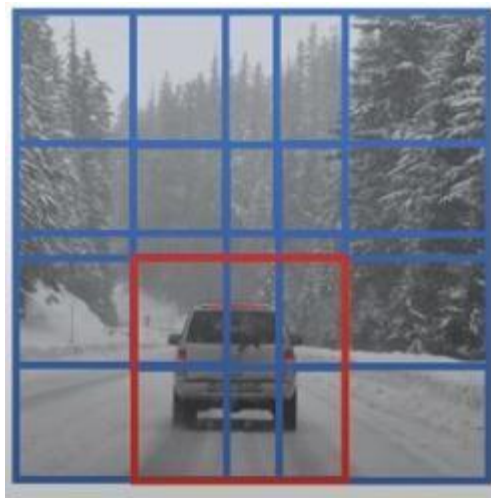


When we test with a larger image, we get multiple classification outputs, with each corresponding to a particular window instance of same size as the image size at which the training took place.

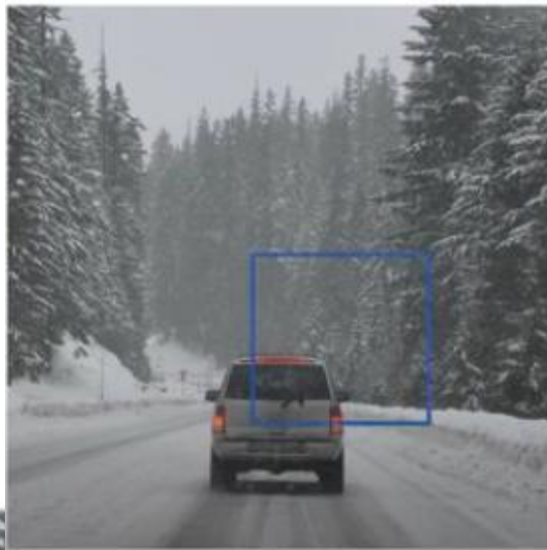




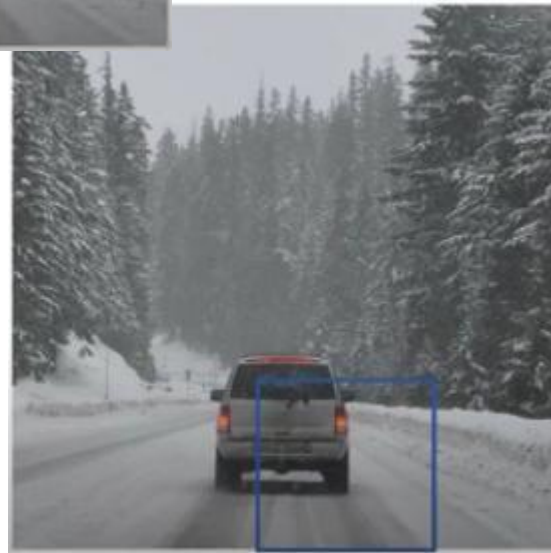
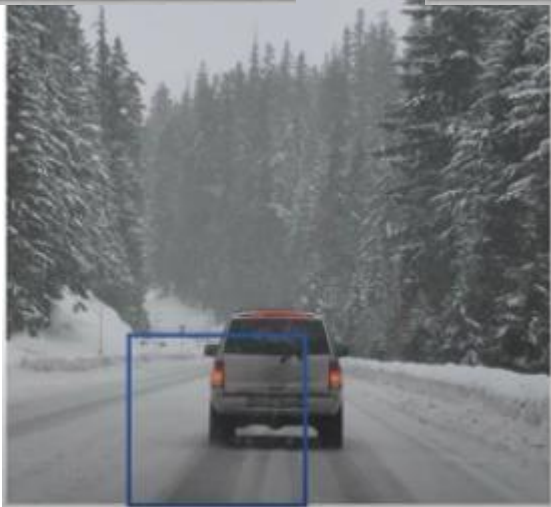
Each vector ($1 \times 1 \times 4$) at the output represents classification output of an individual sliding window instance.



In this way, we are able to get classification of all the sliding window instances at once.



We may not get
accurate bounding
boxes



Lecture Outline

Object Localization: Classification with Localization

Detection multiple objects of same class

Sliding Window via CNN itself

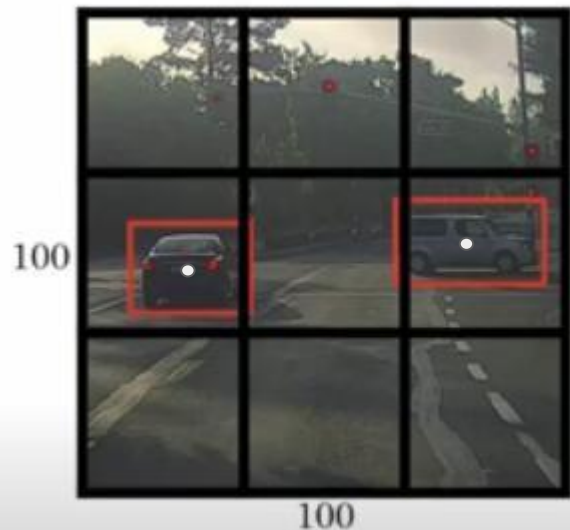
You Only Look Once (YOLO) Object Detection 



YOLO: You Only Look Once

Main Idea: Apply Image
Classification-with-Localization
Algorithm to each grid cell

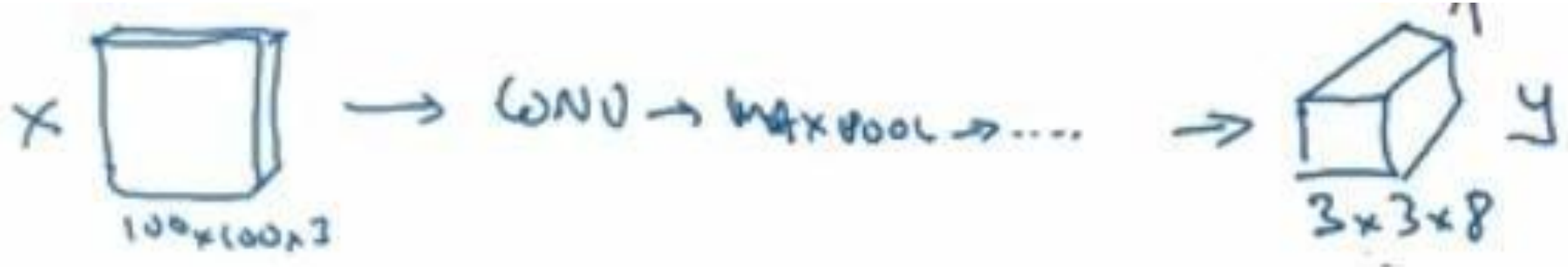
The groundtruth bounding box is assigned to that grid cell in which its centroid lies



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

Labels for training
For each grid cell:

$y =$
 p_c
 b_x
 b_y
 b_h
 b_w
 c_1
 c_2
 c_3

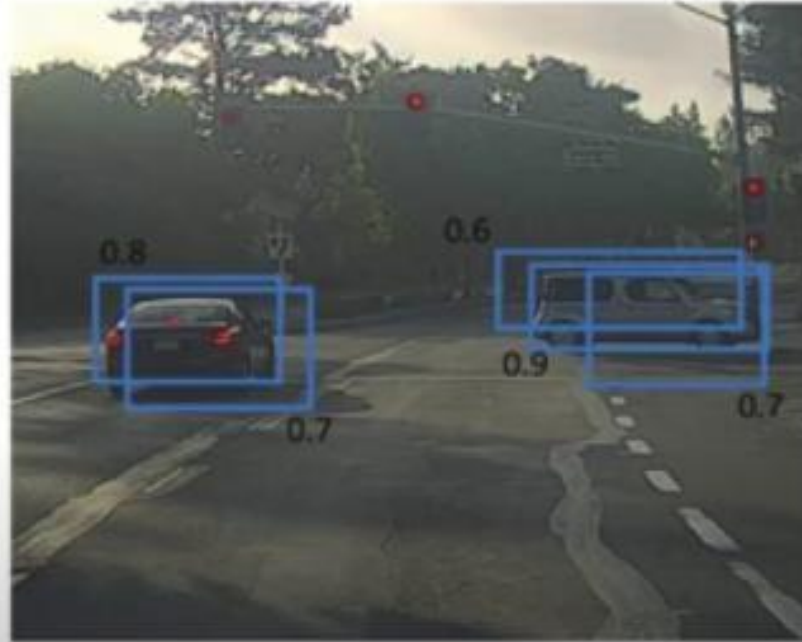


Since we need a fixed size output of FCN, we need to fix the input size also, which was not the case with the sliding window idea.

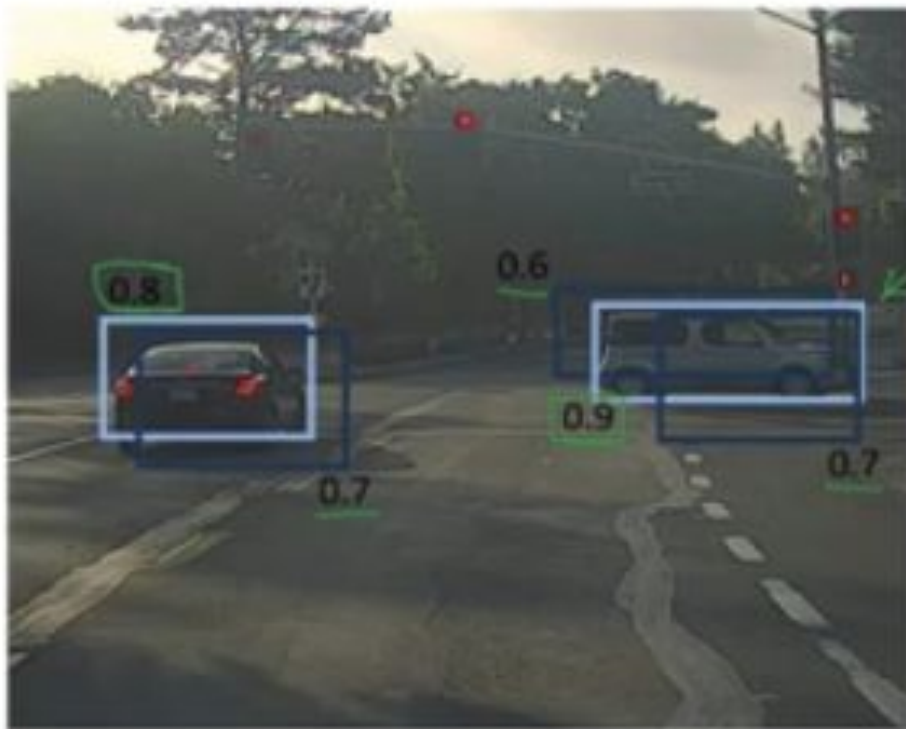


Ideally, we should have many grid cells to make our assumption of one object per grid-cell valid

Non-max suppression example



Many grid cells may get activated based on 'pc' value for the same object



The idea is to eliminate duplicate detections and select the most relevant (prominent) [bounding boxes](#) that correspond to the detected objects.

Discard all boxes with $p_c \leq 0.6$

The process can be broken down into the following steps:

1. Sort the bounding boxes based on their confidence scores.
2. Select the bounding box with the highest confidence score and save it as a detection.
3. Remove all the bounding boxes that have a significant overlap with the selected bounding box. The amount of overlap is typically determined by a predefined threshold value.
4. Repeat steps 2 and 3 until no more bounding boxes remain.

Do this for every class separately after observing whether class probabilities are getting activated in a grid cell based on certain threshold

Reference:

https://www.youtube.com/watch?v=ArPaAX_PhIs&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF