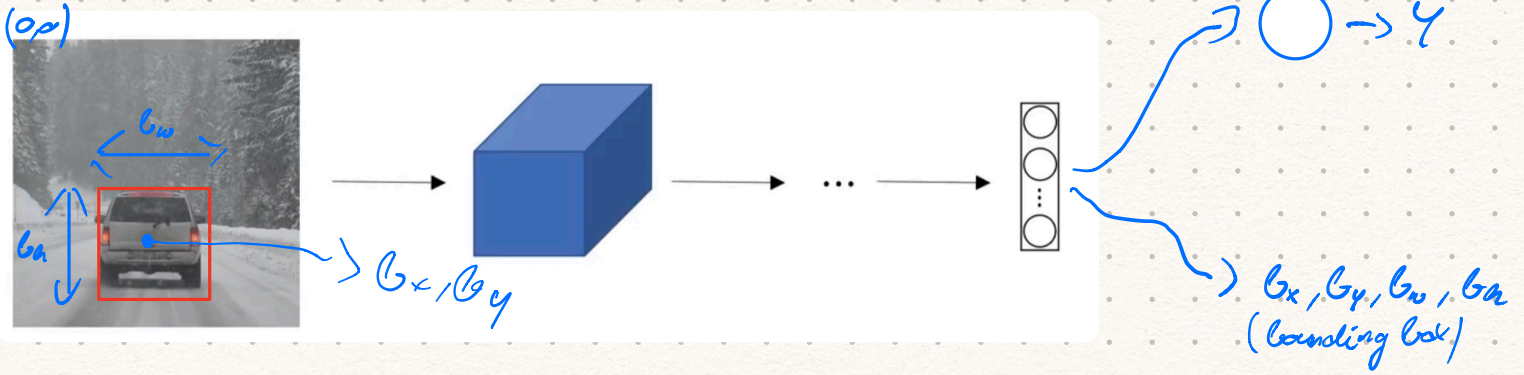


# Classification with localization



Softmax  $\Rightarrow$

- 1: pedestrian
- 2: car
- 3: bike
- 4: else

## Target label $y$

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

$\rightarrow$  is there any objects?

Bounding box

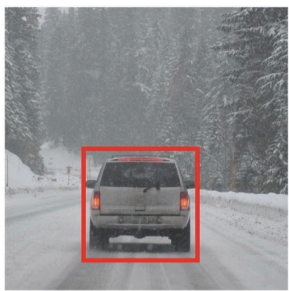
$c_1 \rightarrow$  class 1  
 $c_2 \rightarrow$  class 2  
 $c_3 \rightarrow$  class 3

$$L(\bar{y}, y) = (\bar{y}_1 - y_1)^2 + \dots + (\bar{y}_8 - y_8)^2 \quad \text{if } p_c = 1$$

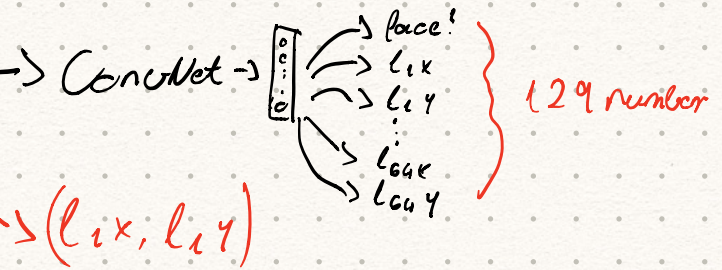
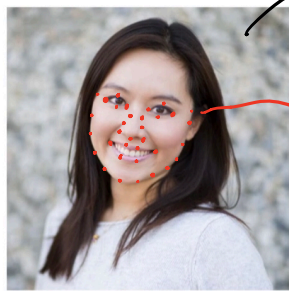
$$= (\bar{y}_1 - y_1)^2 \quad \text{if } p_c = 0$$

(You can use different loss for the different components)

## handmade detection



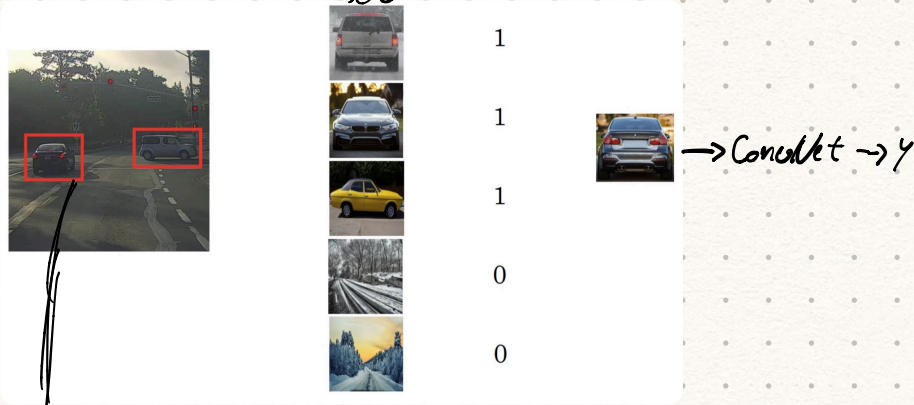
$b_x, b_y, b_h, b_w$



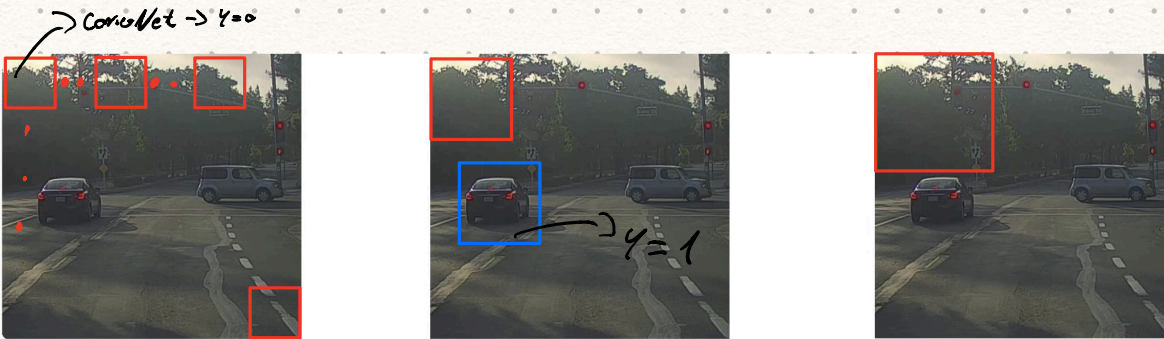


# Object detection

Training set  $y$

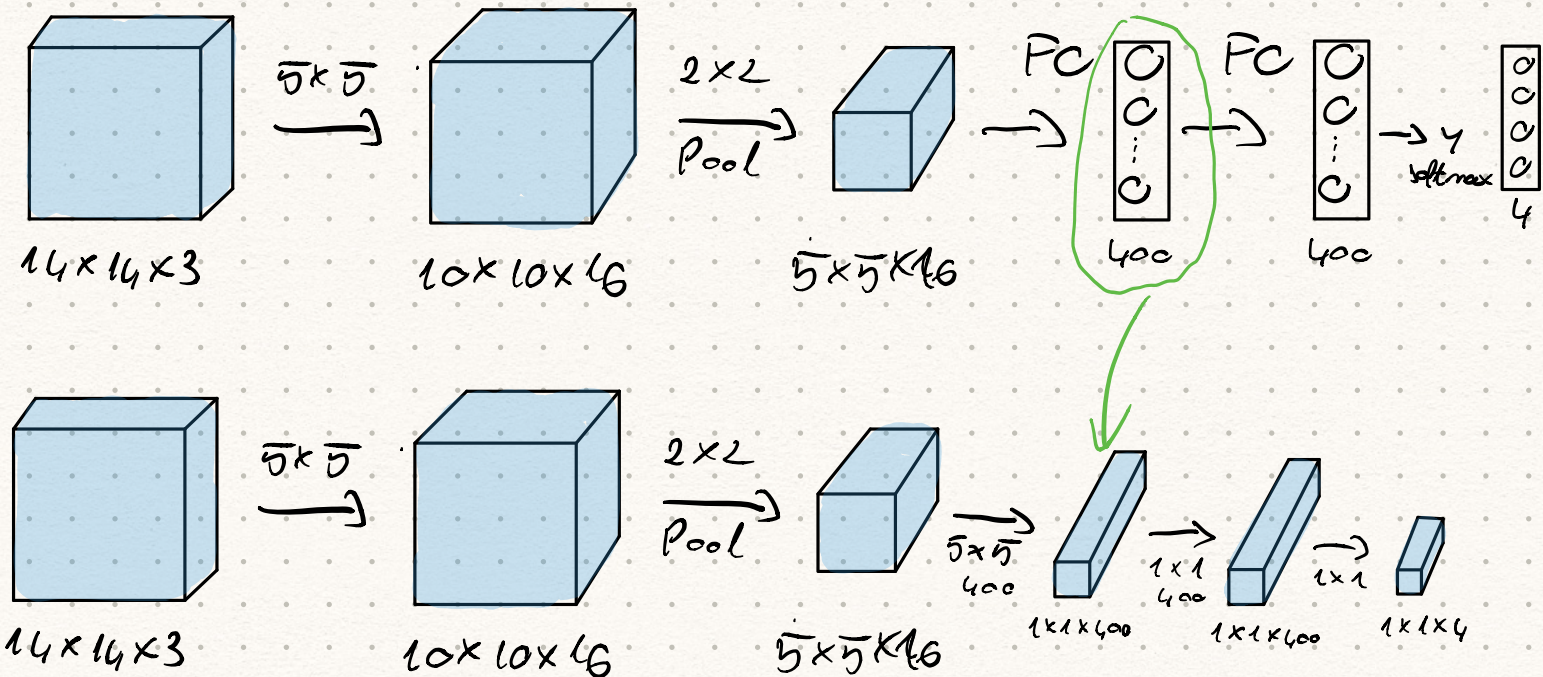


# Sliding Window detection



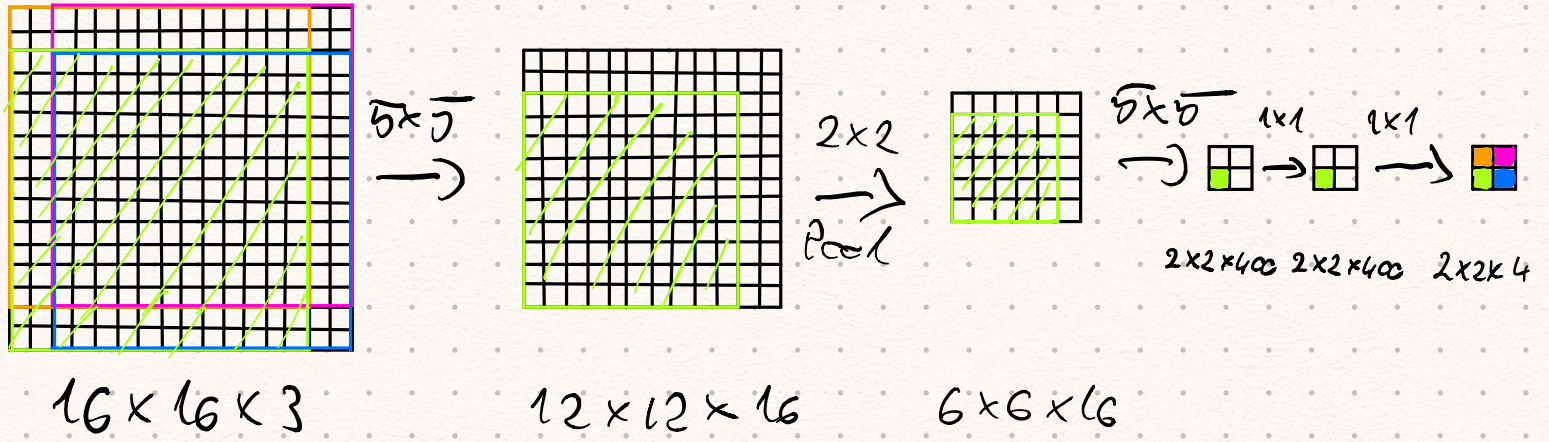
• With small stride steps moving the window ⇒ very slow algorithm

# Turning FC layer into convolutional layers

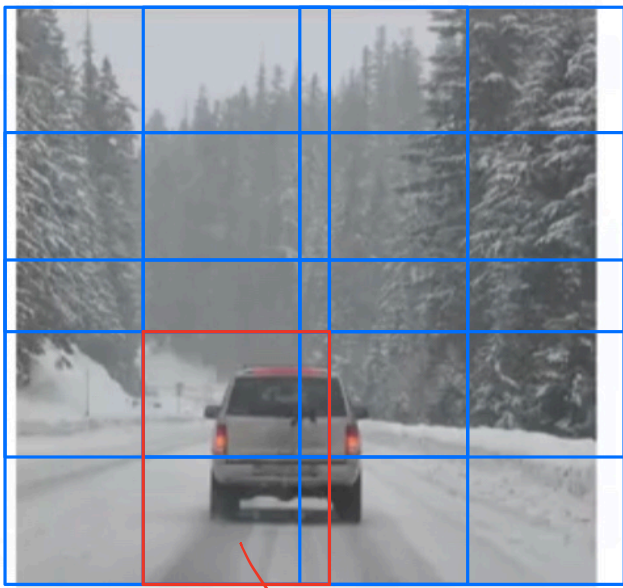




# Convolutional implementation of sliding window



- Shares a lot of computation in the regions of the image that are common.



$y=1$

# Evaluating object localization

↳ Intersection over Union (IoU)

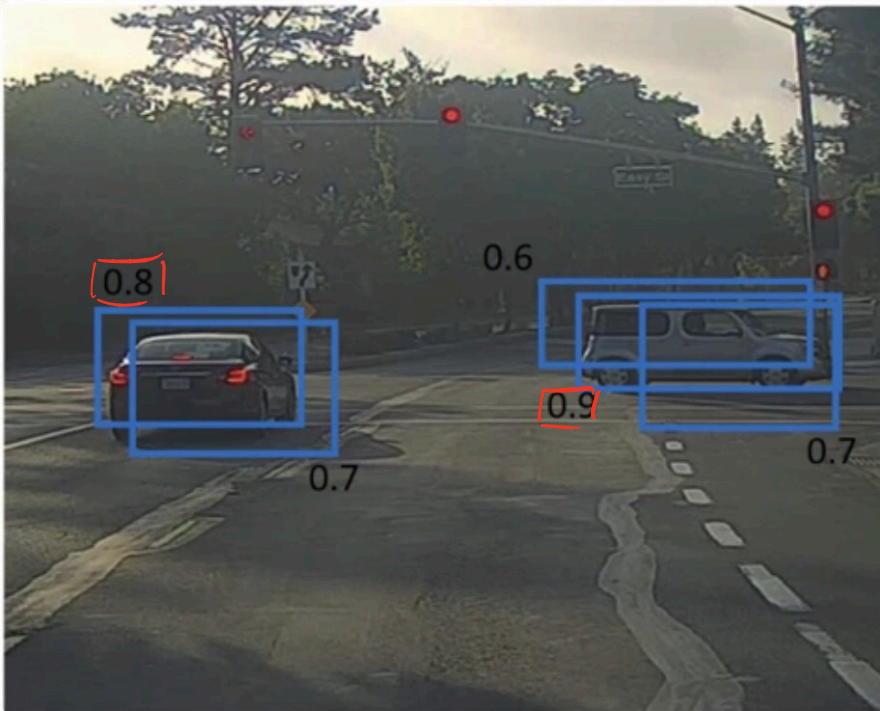


$$IoU = \frac{\text{Size of } \begin{array}{|c|} \hline \text{orange} \\ \hline \end{array}}{\text{Size of } \begin{array}{|c|} \hline \text{green} \\ \hline \end{array}}$$



## Non-max suppression

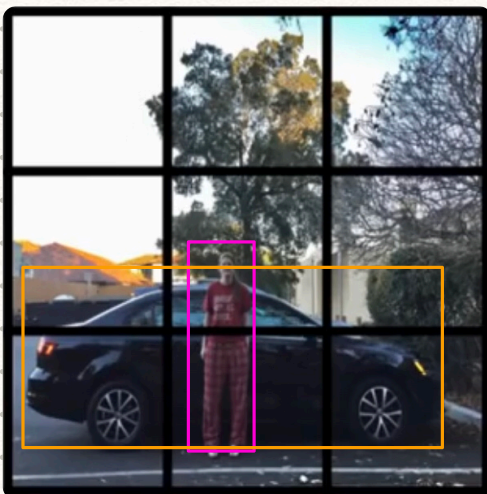
↳ an object is detected multiple times → problem



↳ highlight the rectangle with the highest probability and

↳ remove the remaining rectangles with high IOU

## Overlapping objects → Anchor boxes



Anchor 1



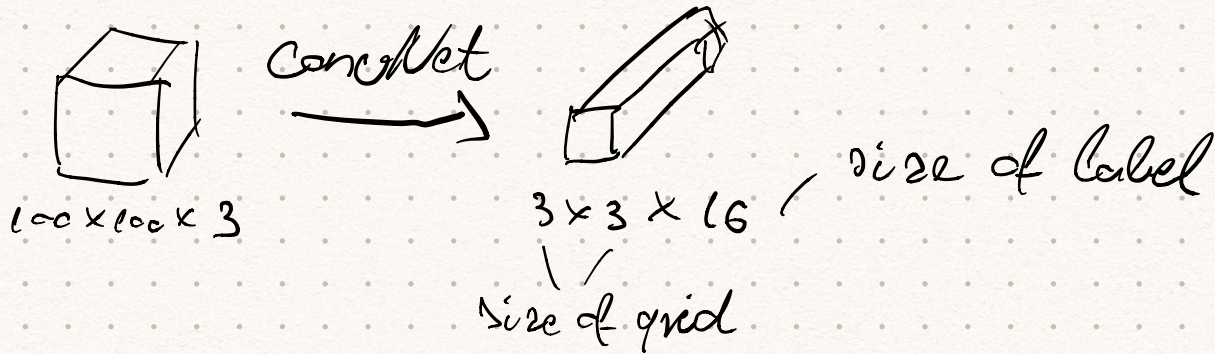
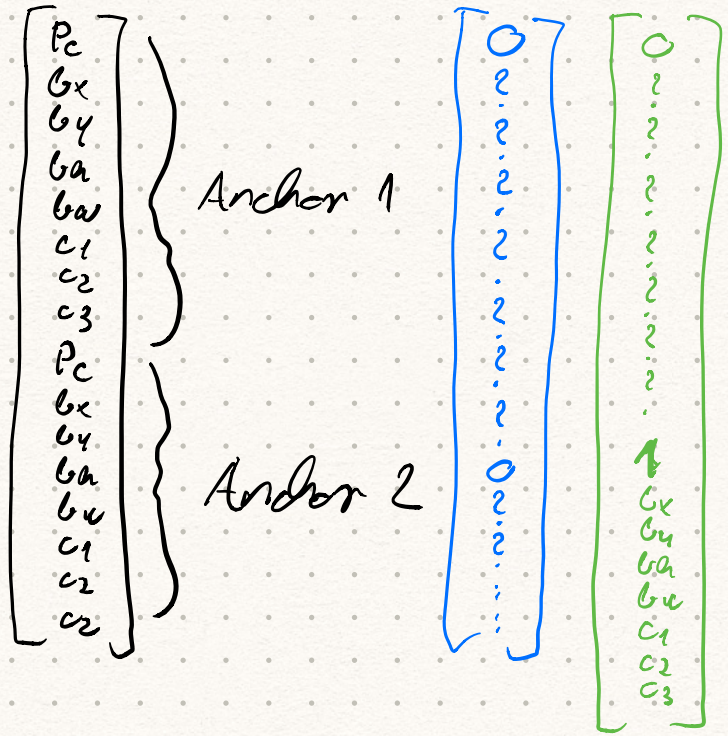
Anchor 2



Multiple anchor boxes can be assigned to the same grid cell.

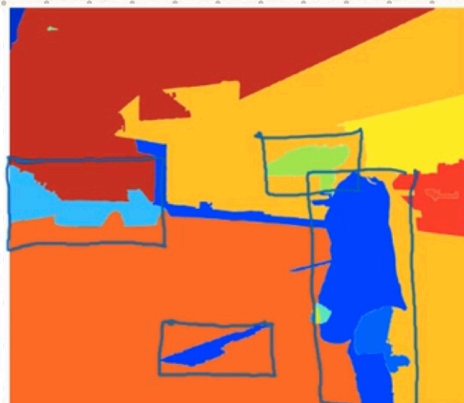
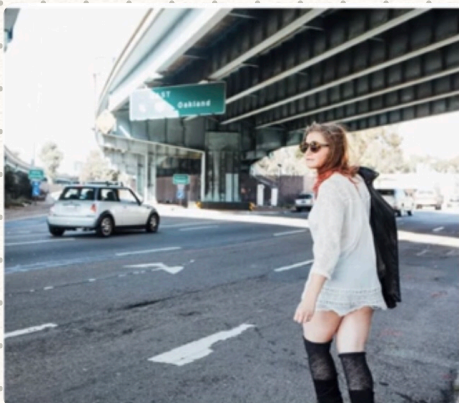


# YOLO algorithm



# Region proposal - R-CNN (YOLO is faster)

- Rather than run sliding windows try to give only few windows for further processing



segmentation

R-CNN → slow  
 ↳ Fast R-CNN  
 use convolutional implementation of sliding window  
 ↳ Faster R-CNN  
 use convNet to propose regions