Deep Learning

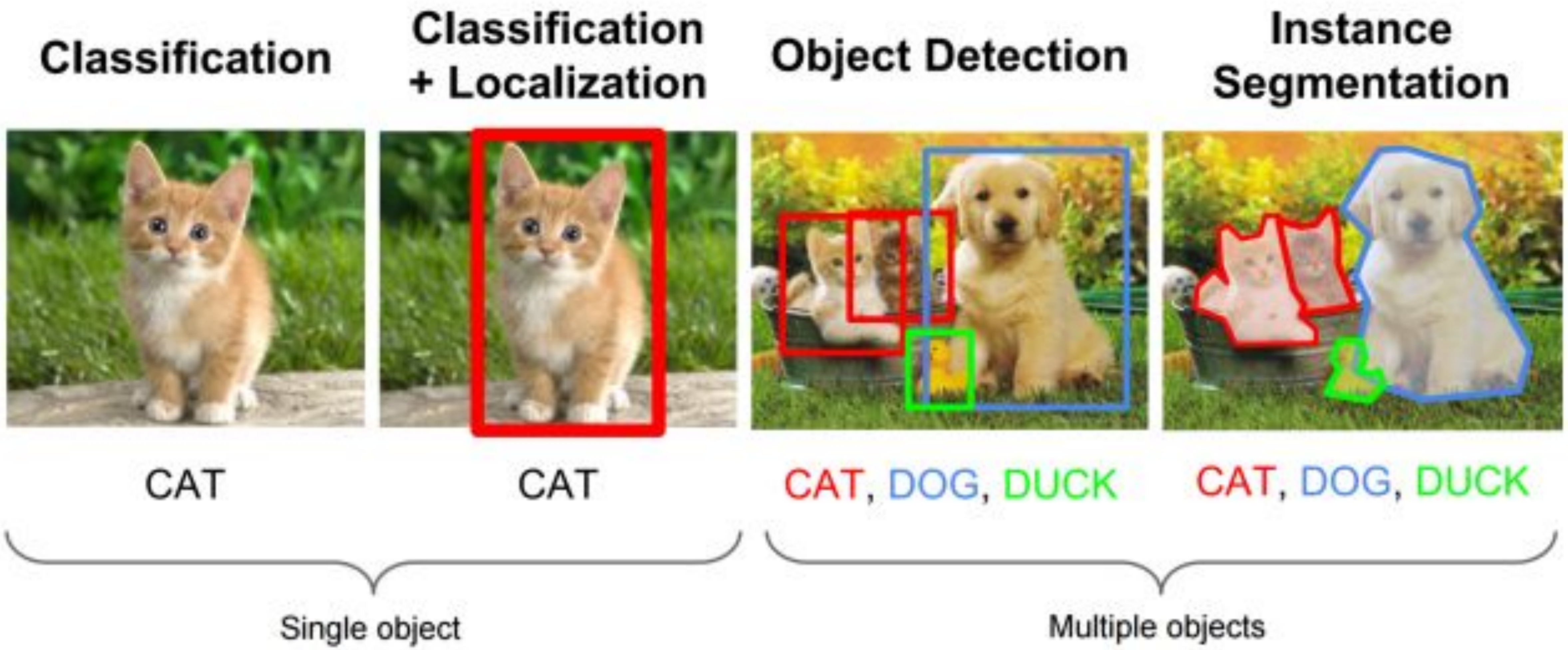
Big Data & Machine Learning Bootcamp - Keep Coding

Outline

1. Object localization
2. Landmark detection
3. Object detection
4. Convolutional implementation of sliding windows
5. Bounding box predictions (YOLO)
6. Intersection over union (IoU) (YOLO)
7. Non-max suppression (YOLO)
8. Anchor boxes (YOLO)
9. YOLO algorithm

1. Object localization

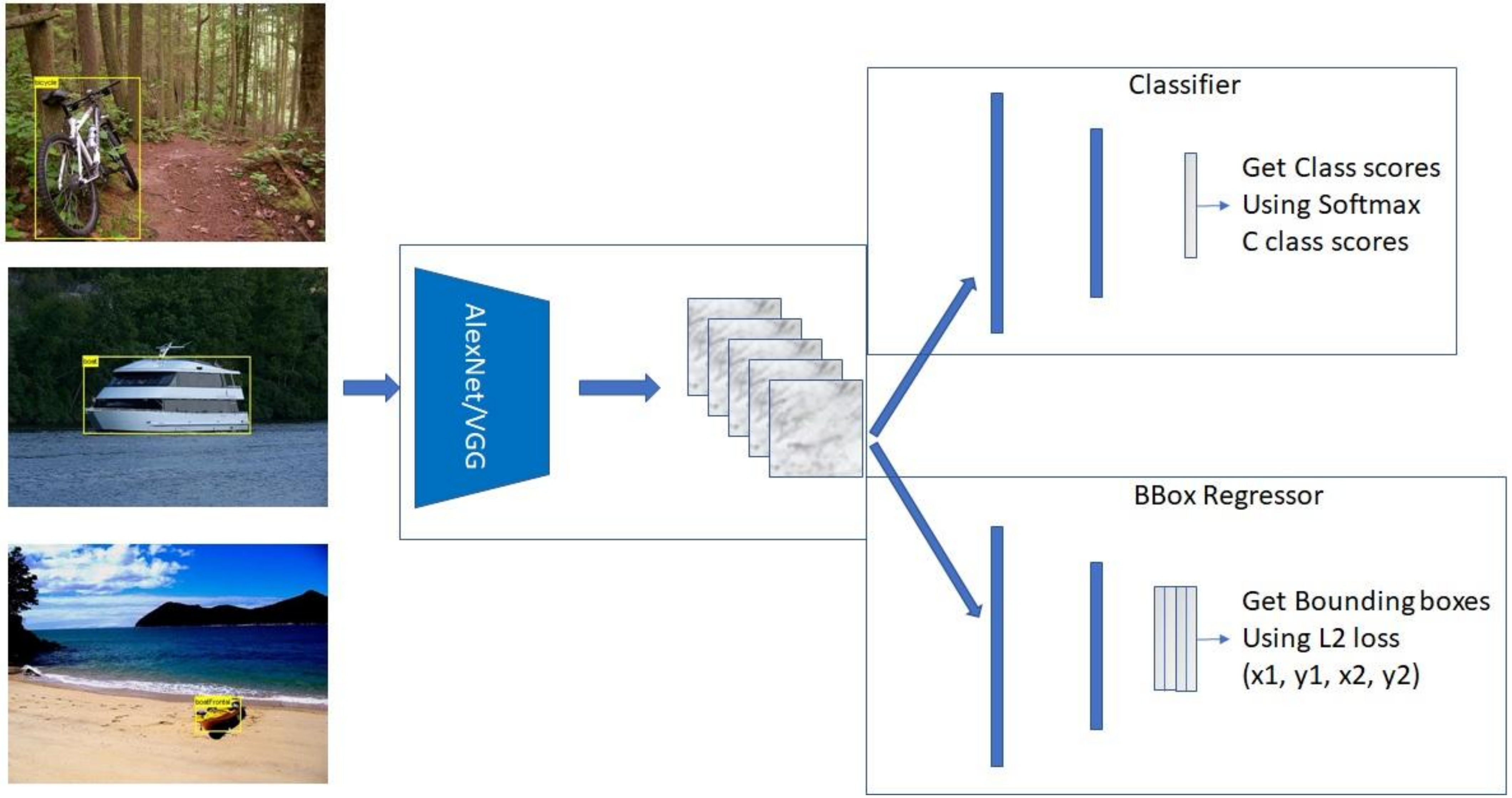
It is important to first distinguished **the difference tasks**. What is classification, classification with localization, object detection and instance segmentation. Depending on the task, the labels will change.



Sources:
- Coursera
- https://towardsdatascience.com/object-localization-in-overfeat-5bb2f7328b62

1. Object localization

Classification with localization



Remember that for classification only, the output was a class for each image?

For this task, the output is the ***class AND also the bounding box coordinates*** *(x1, y1, x2, y2)*

Sources:
- Coursera
- https://cogneethi.com/evodn/object_detection_intro/

1. Object localization

Labels for classification with localization:

Say for instance we have 4 different classes:

1. Pedestrian
2. Car
3. Motorcycle

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

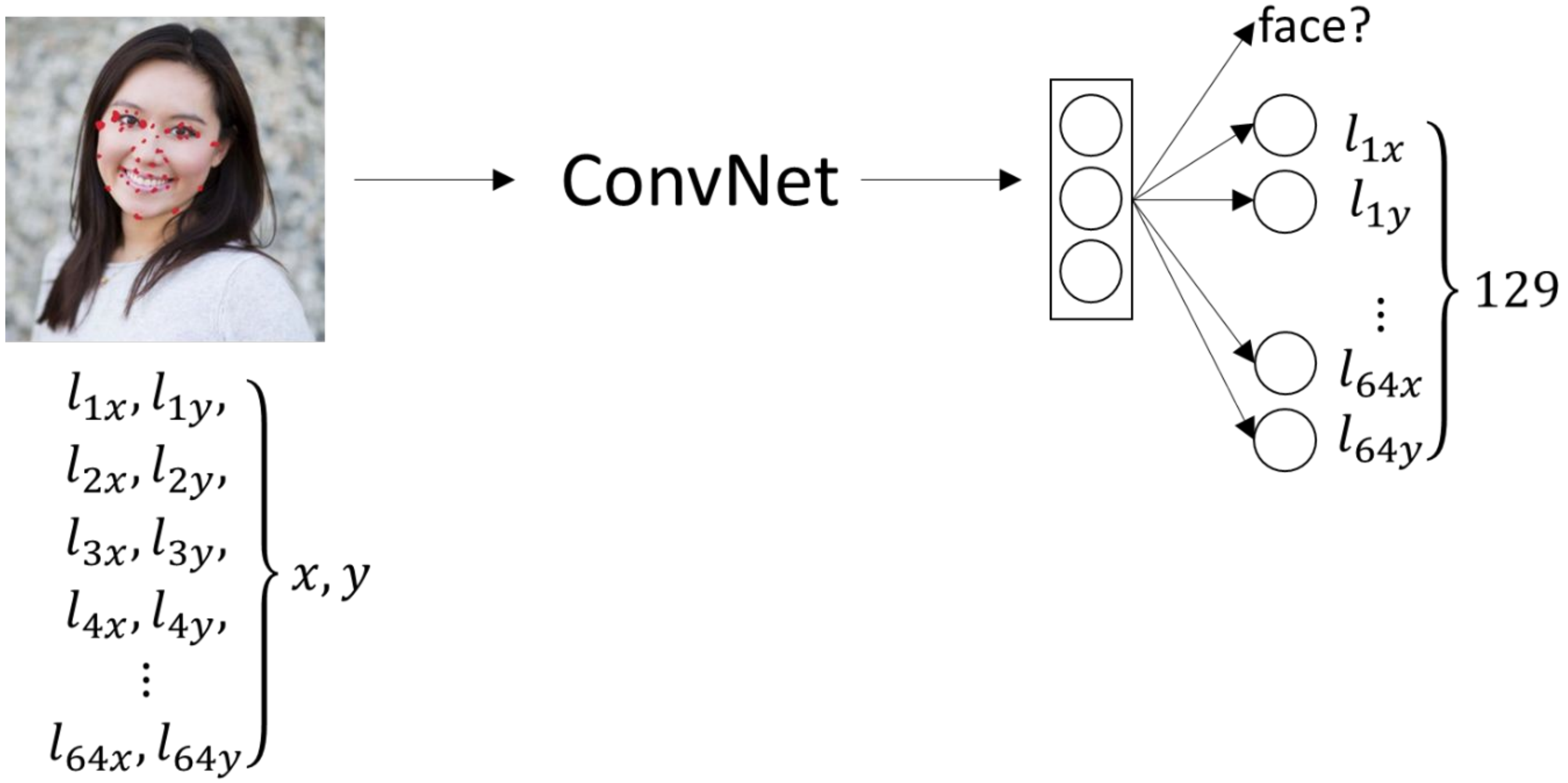
Then the label "y" will be a vector of 8 numbers. **The first component (p_c) will indicate if there is an object in the image, the next 4 components (b_x, b_y, b_h, b_w) will specify the bounding box and the last 3 (c_1, c_2, c_3) will represent the label for the detected object.**

If there is no object in the image, p_c = 0 and the other values won't matter

Sources:
- Coursera
- https://medium.com/machine-learning-bites/deeplearning-series-objection-detection-and-localization-yolo-algorithm-r-cnn-71d4dfd07d5f

2. Landmark detection

The idea of having multiple numbers as labels can also be applied to landmark detection.



The first number in the vector indicate if there is a face in the image. The other numbers represent landmark in the face is represented by (x, y) coordinate.

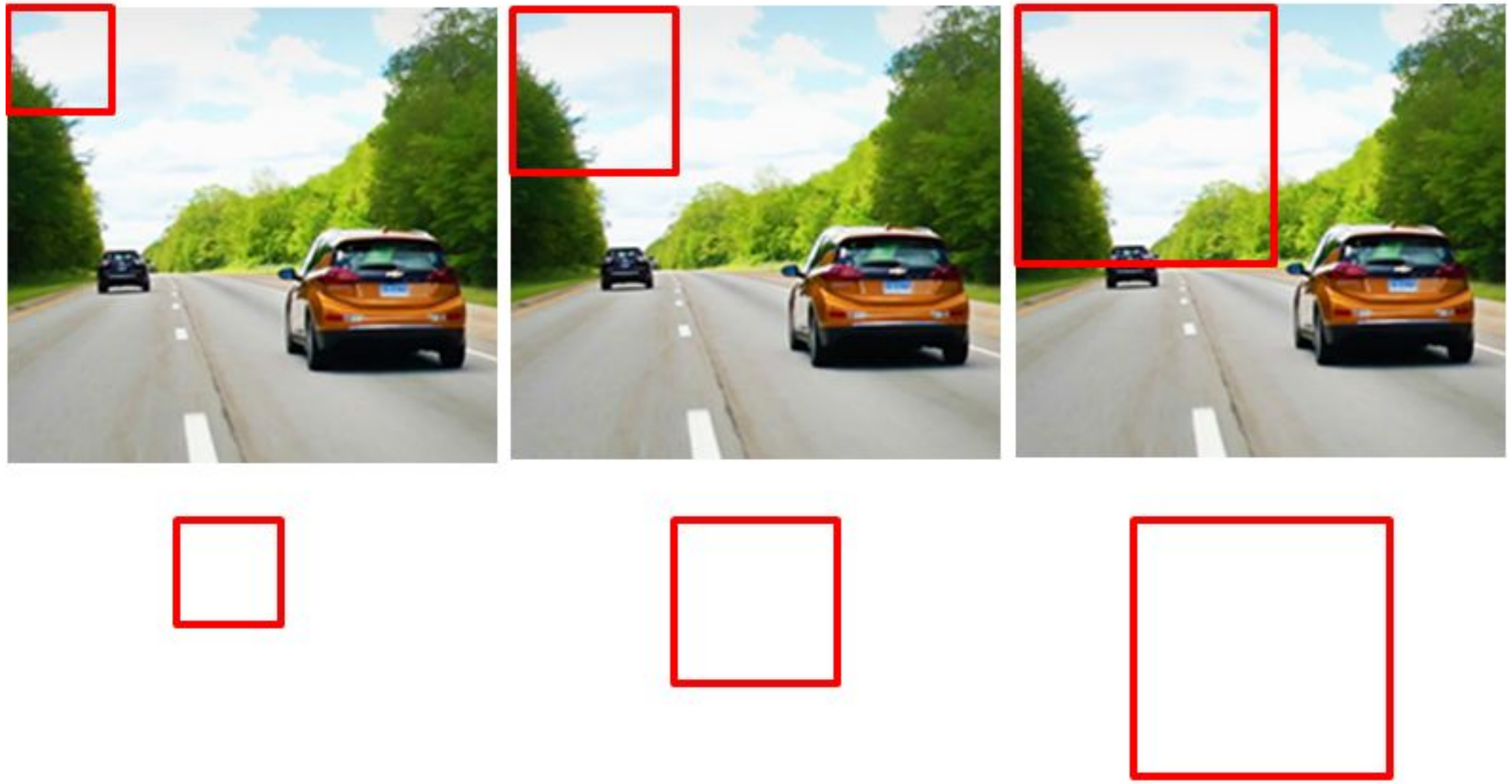
Applications of this are emotion detection, graphic effects and virtual reality!

Have you seen the instagram effects?

Sources:
- Coursera
- https://datahacker.rs/deep-learning-landmark-detection/

3. Object detection

Sliding window for object detection



Basically you choose a window size and pass it through the image with a defined stride. Each time classifying whether the object is in the window. Then, Increment the window and pass it again to the image.

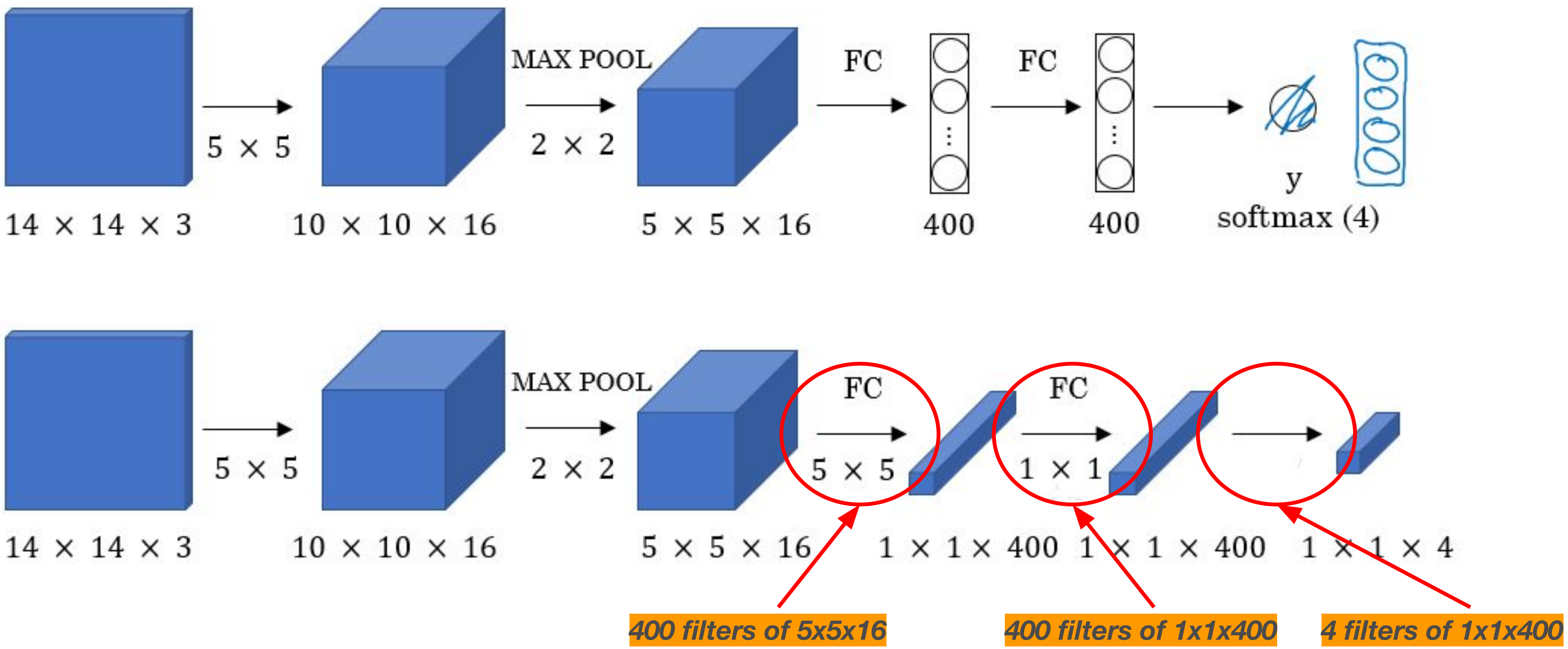
Huge disadvantage:
- *Computation cost*

Sources:
- Coursera
- http://datahacker.rs/deep-learning-object-detection/

4. Convolutional implementation of sliding windows

Sliding window implemented convolutionally.

Let's first see how we can turn a fully connected layer into a convolutional layer:

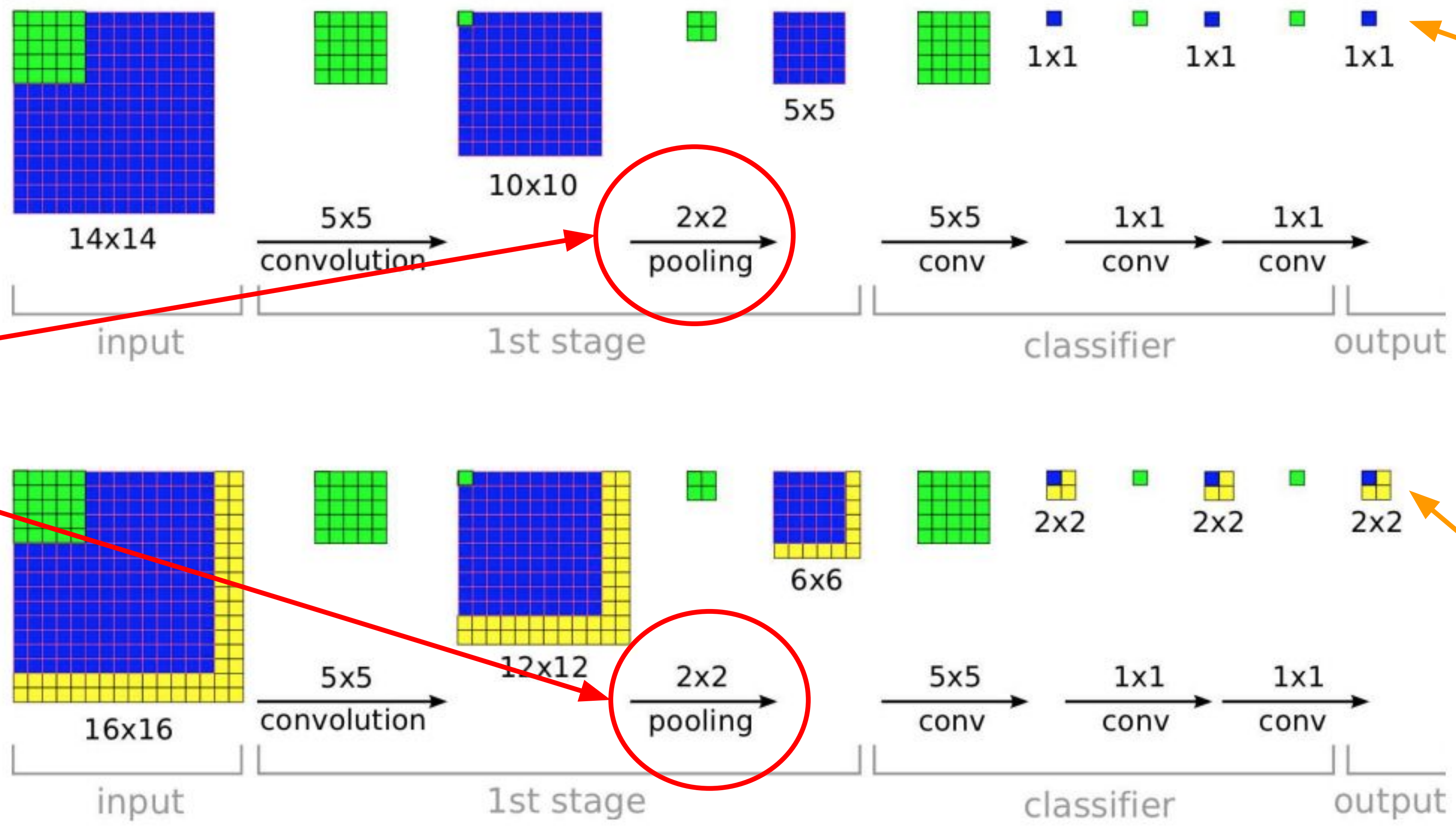


Sources:
- Coursera
- https://zhangruochi.com/Convolutional-Neural-Networks/2019/03/27/

4. Convolutional implementation of sliding windows

Sliding window implemented convolutionally.

Running a pooling layer of 2x2 corresponds to slide a window of 14x14 with stride of 2 in the original image



This is for one window

All of these computations performed at once!!

This is for 4 windows

Sermanet, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks." *arXiv preprint arXiv:1312.6229* (2013).

Sources:
- Coursera
- https://www.oreilly.com/library/view/deep-learning-for/9781788295628/e1f0c48f-e9bd-4972-b31c-a3d87a462e73.xhtml

5. Bounding box predictions (YOLO)

YOLO algorithm: You only look once!

The previous approach still have the problem that the bounding box is not always the size of the object we want to detect.

YOLO algorithm **first place a grid down to the image** and then apply the classification with localization algorithm.

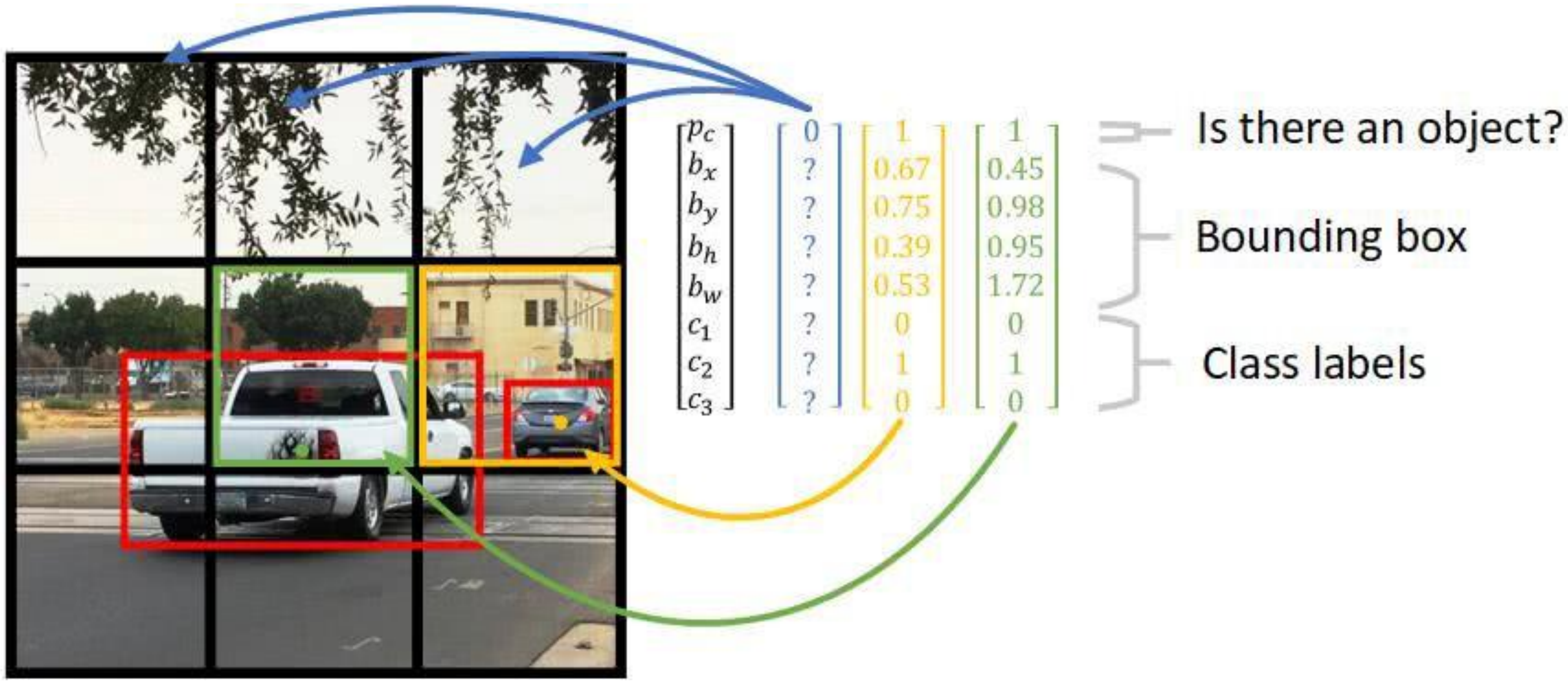
Let's see how the labels "y" are for this approach

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Sources:
- Coursera

5. Bounding box predictions (YOLO)

Labels for the YOLO algorithm



In this case, in which we placed a grid of 3x3, the labels will have a size of 3x3x8.

The label will be 1 when the cell contains the centre of the bounding box.

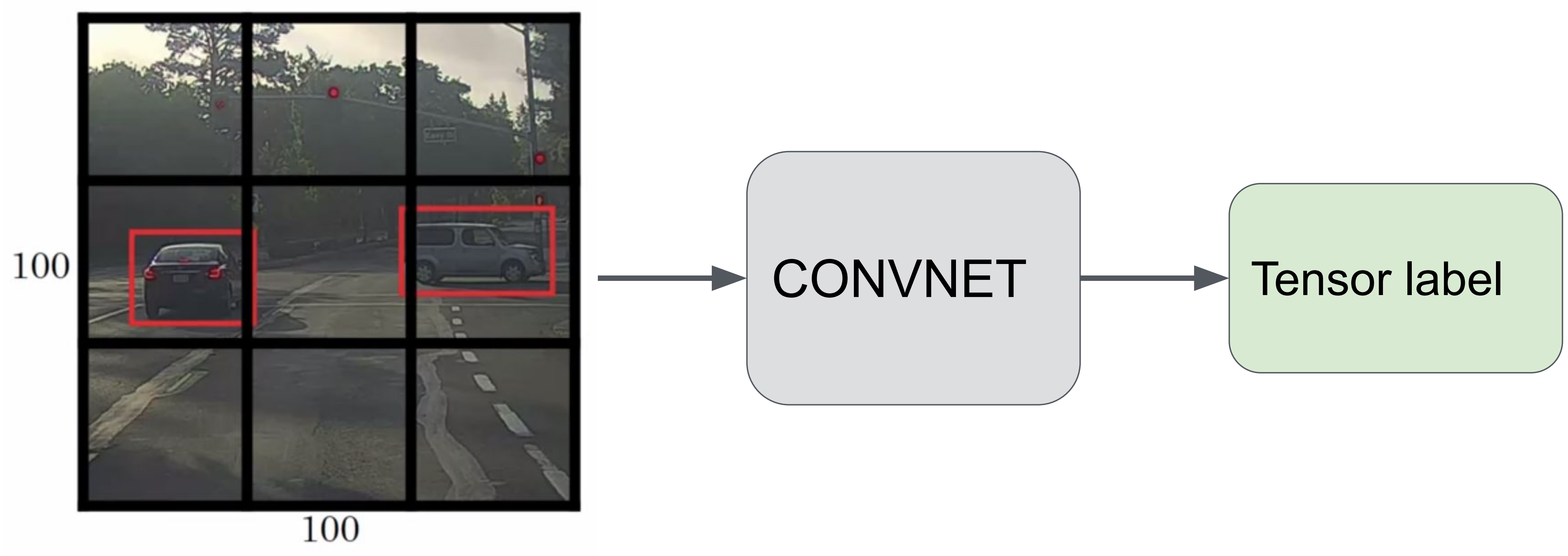
Usually, the grid is 19x19 or even more!

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Sources:
- Coursera
- https://stackoverflow.com/questions/50575301/yolo-object-detection-how-does-the-algorithm-predict-bounding-boxes-larger-than

5. Bounding box predictions (YOLO)

YOLO architecture in general



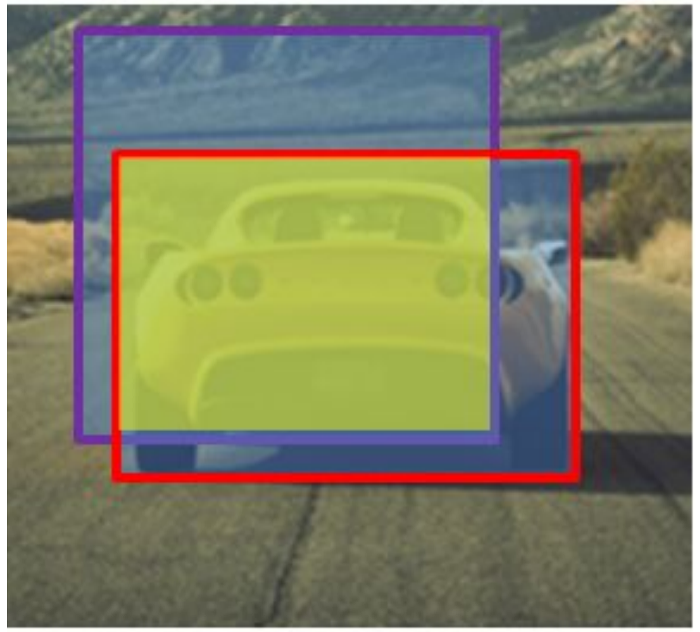
Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

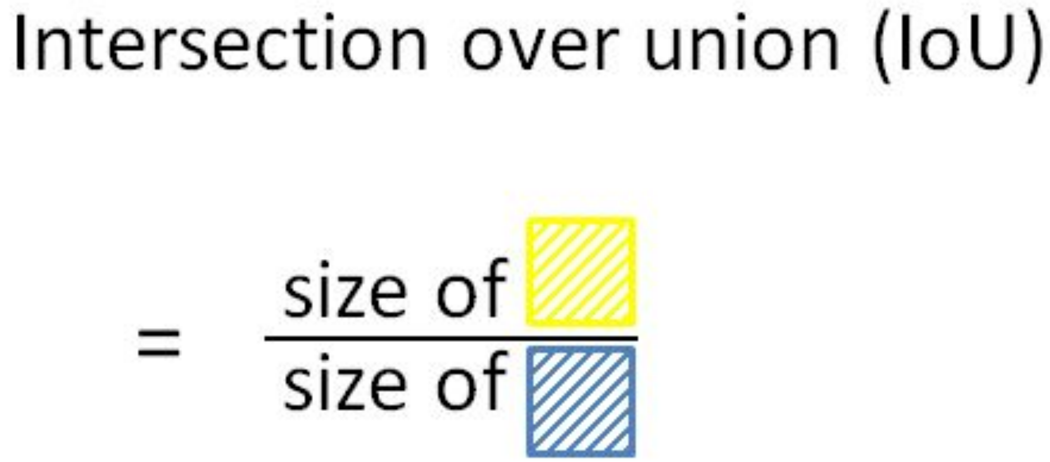
Sources:
- Coursera

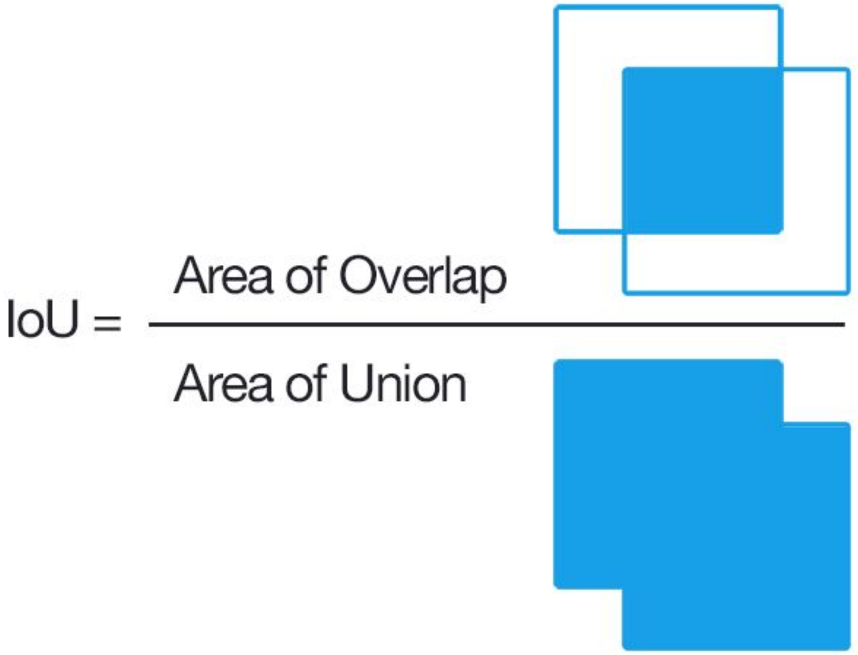
6. Intersection Over Union (YOLO)

Intersection Over Union (IoU) tells us if our object detection algorithm is doing well.

In general, this is a way to tell whether two boxes are similar







"Correct" if $\mathrm{IoU} \geq 0{,}5$

Sources:
- Coursera
- http://datahacker.rs/deep-learning-intersection-over-union/

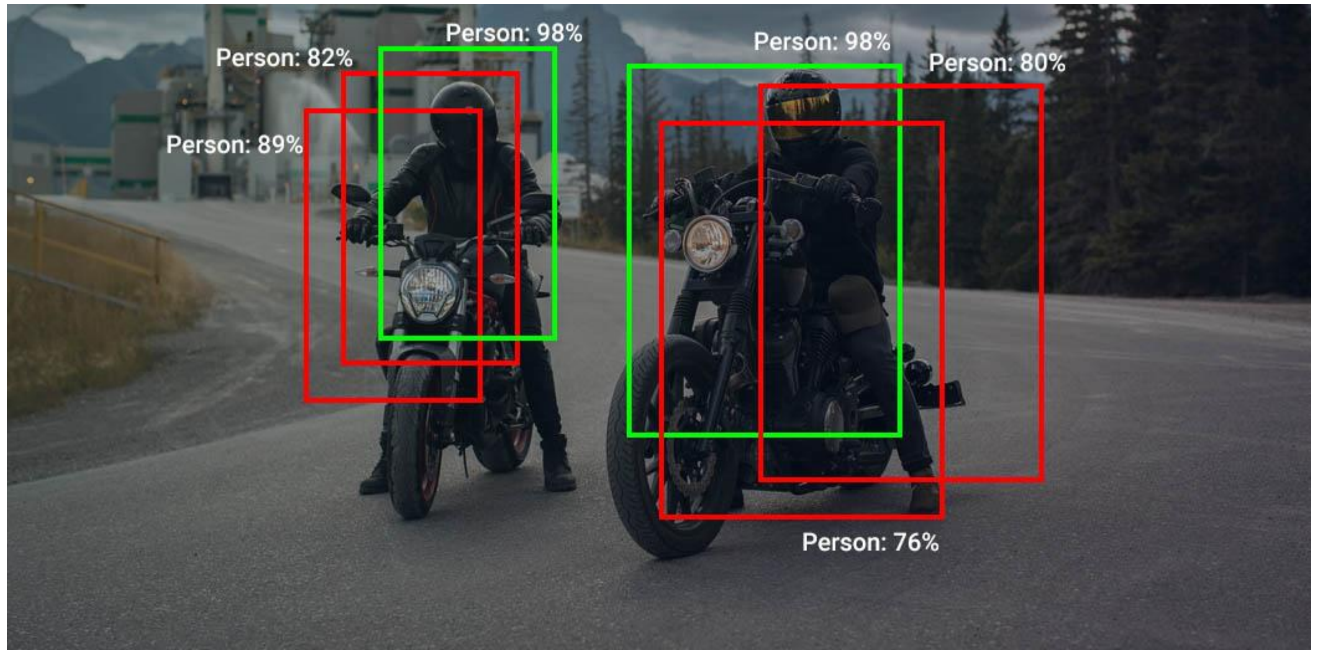
7. Non-max suppression (YOLO)

Non-max suppression helps us to remove the bounding boxes that are not similar to the ground truth/labels.

In other words, it is a way to make sure the algorithm only detects each object only once!

Here are the steps:

- Discard the bounding boxes that have p_c <= 0.6
- While there are any remaining boxes:
  - *Pick the box with the largest p_c* -> ***This will be our final prediction***
  - Discard any remaining box with IoU >= 0.5 with the box output in the previous step



In this case we showed only one class, but non-max suppression should be applied for the classes

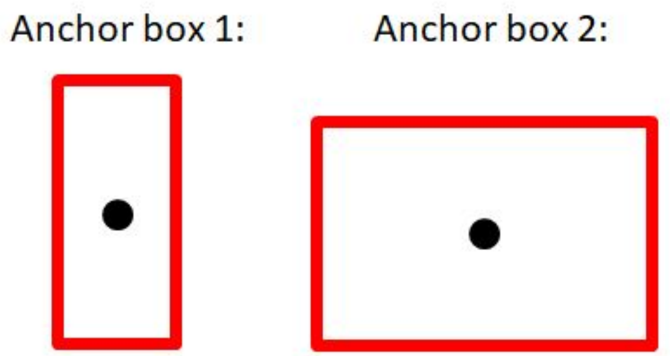
Sources:
- Coursera
- https://www.analyticsvidhya.com/blog/2020/08/selecting-the-right-bounding-box-using-non-max-suppression-with-implementation/

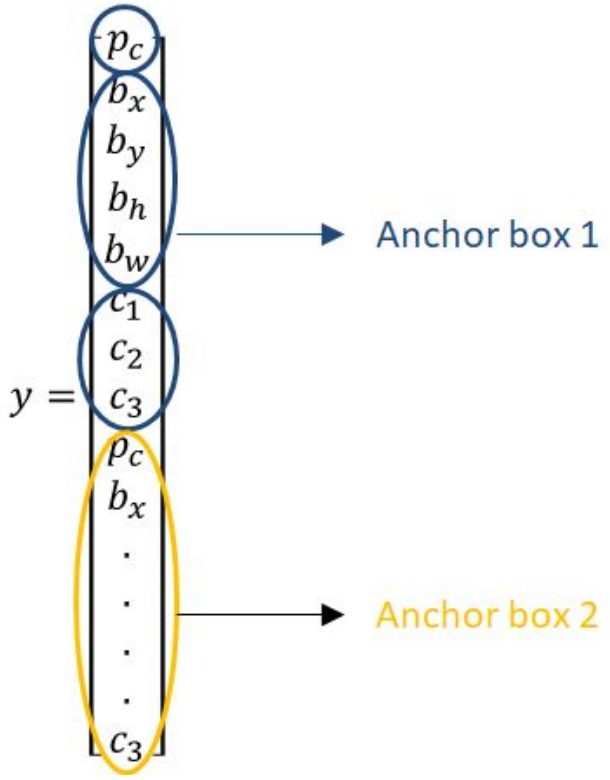
8. Anchor boxes (YOLO)

Anchor boxes allows each grid cell to detect more than one object.

Labels for each grid cell will contain information regarding all the number of bounding boxes we have.

In this case we're showing only two anchor boxes, but there can be 5, 6 or 10

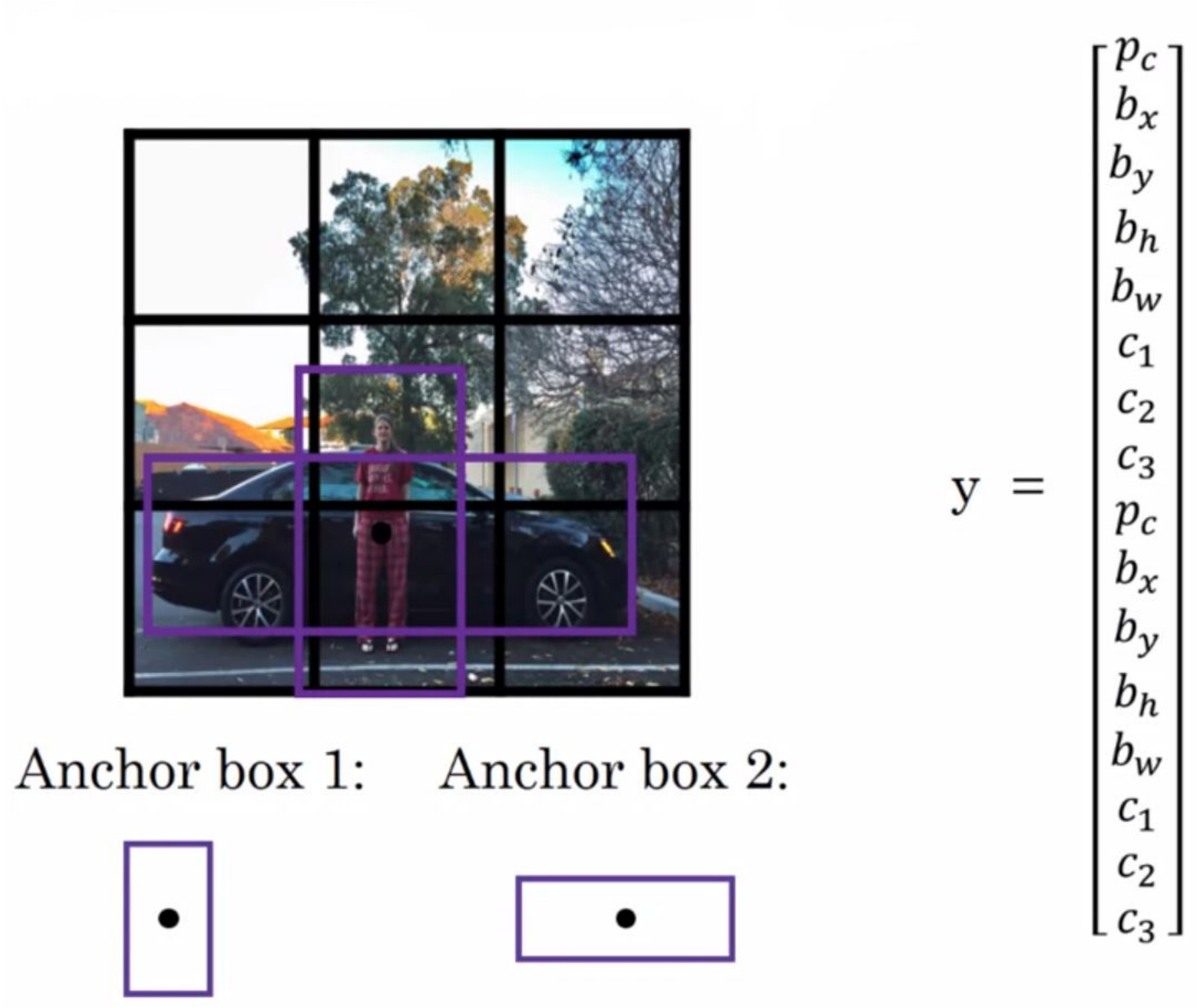




Sources:
- Coursera
- http://datahacker.rs/deep-learning-anchor-boxes/

8. Anchor boxes (YOLO)

Another example of anchor boxes and the labels size



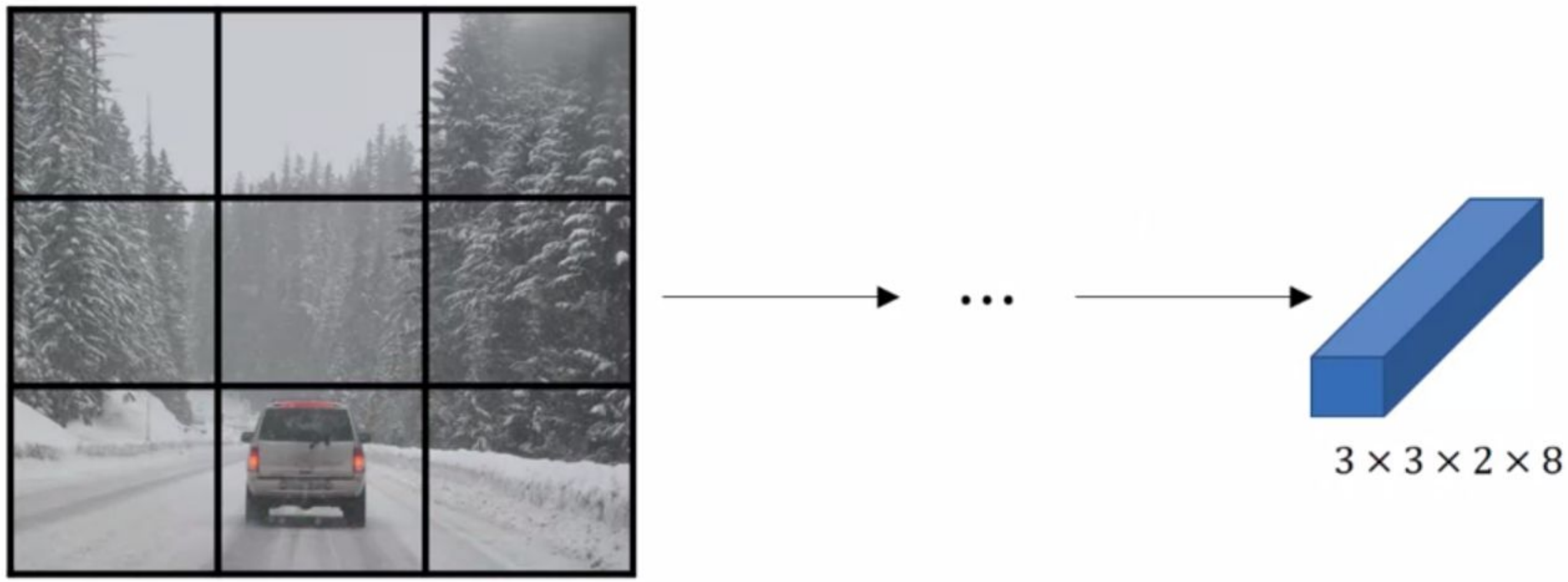
When there is a grid cell that only contains one anchor box, $p_c = 0$ for the other anchor boxes

Sources:
- Coursera

9. YOLO Algorithm

Let's put all the components together to form the YOLO algorithm.

- For training, we first need to create the labels "y" for all our grid cells and anchor boxes



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

One vector like this for each grid cell.

This is when we have four objects and 2 anchor boxes.

Sources:
- Coursera

9. YOLO Algorithm

Let's put all the components together to form the YOLO algorithm.

- Then you run non-max suppression:

1. For each grid cell, get 2 predicted bounding boxes (in the example in which we have two bounding boxes of course)

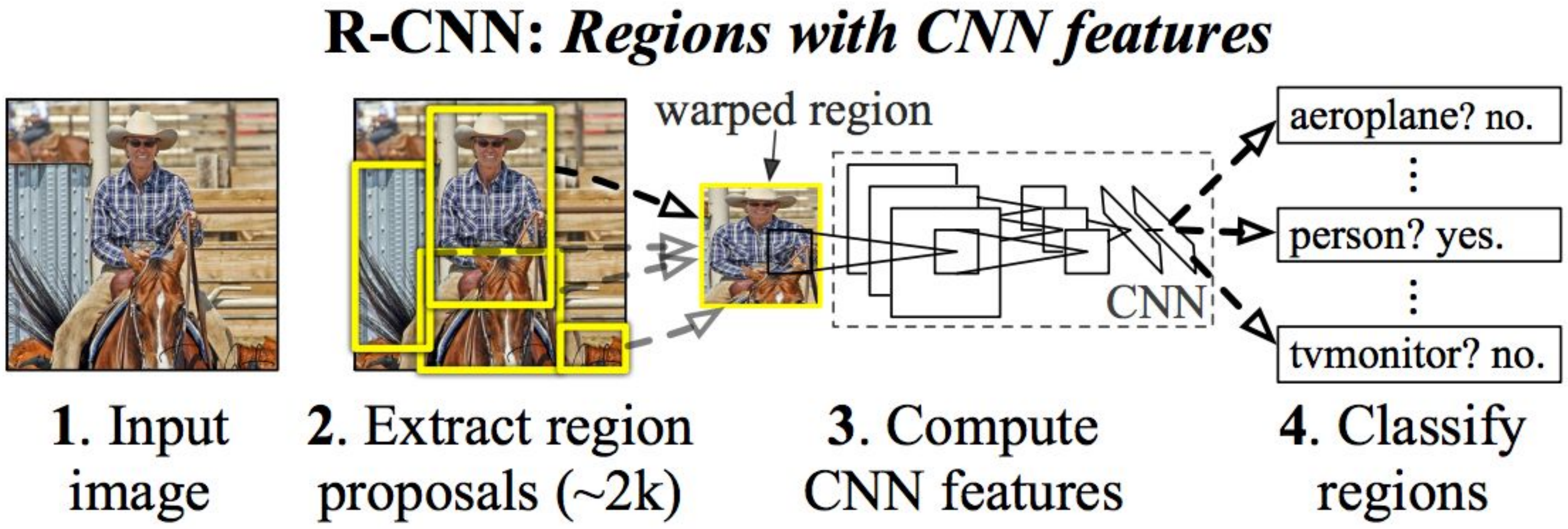
2. Get rid of low probability prediction

3. For each class use non-max suppression to generate final predictions

Sources:
- Coursera

Good to know (R-CNN, Fast R-CNN, Faster R-CNN)

There is another approach to detect objects that is called region proposals or R-CNN.



We first use a region proposal network and then run the convolutional sliding window on the regions!

Because it has two steps, it is a bit slower than YOLO

Sources:
- Coursera
- https://towardsdatascience.com/step-by-step-r-cnn-implementation-from-scratch-in-python-e97101ccde55