YOLO algorithm

YOLO

- There are a few different algorithms for object detection and they can be split into two groups:
- Algorithms based on classification – they work in two stages. In the first step, we're selecting from the image interesting regions. Then we're classifying those regions using convolutional neural networks.

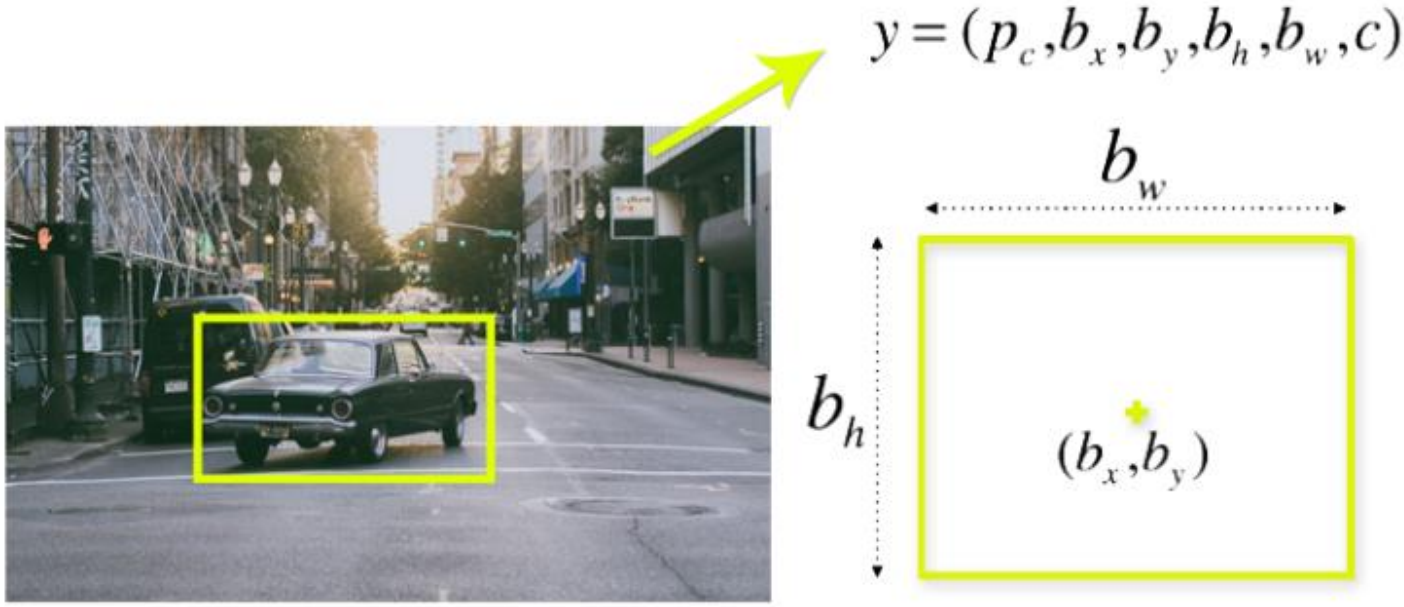
- This solution could be very slow because we have to run prediction for every selected region

.

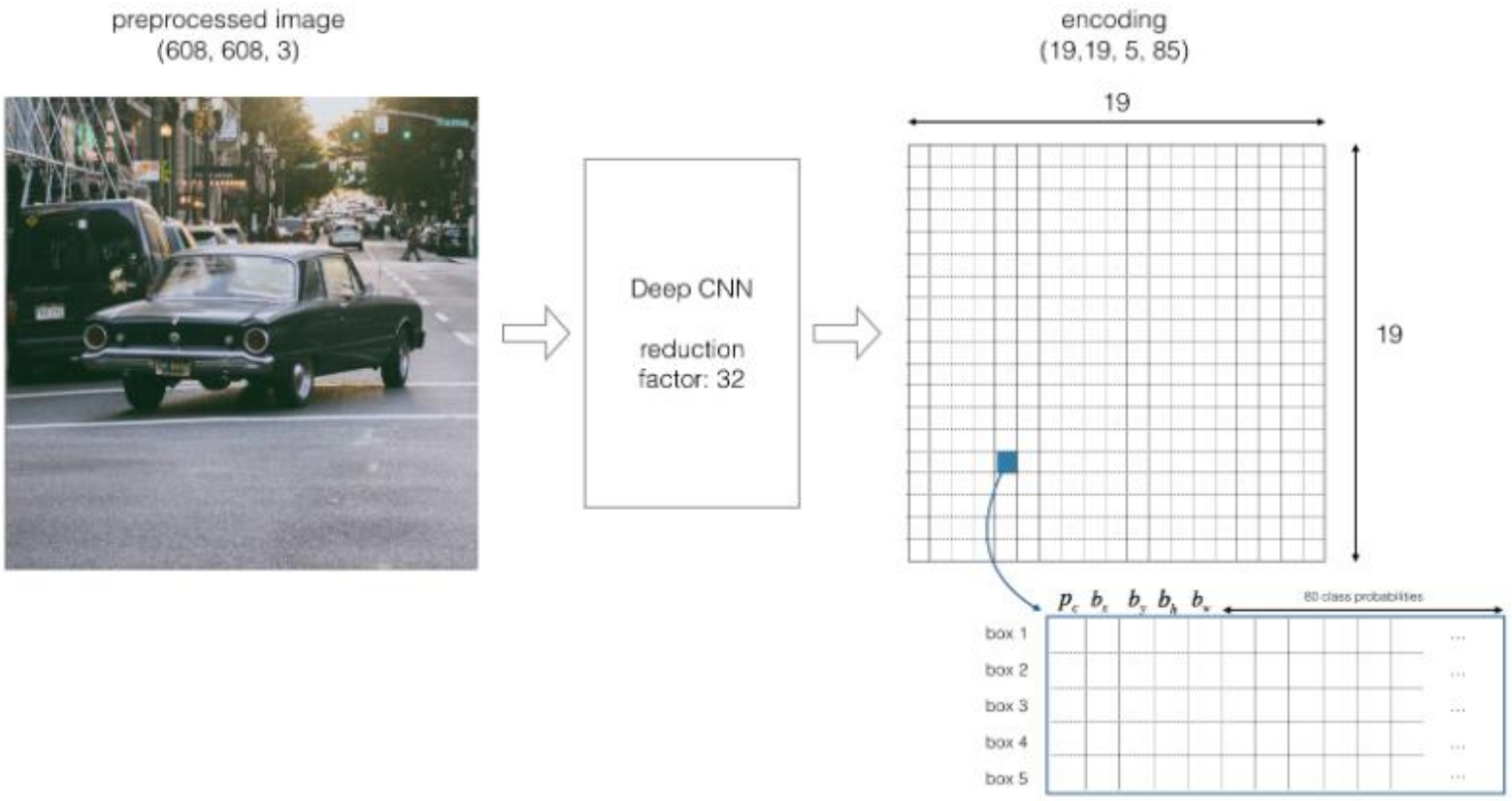
- Algorithms based on regression – instead of selecting interesting parts of an image, we're predicting classes and bounding boxes for the whole image in one run of the algorithm. Most known example of this type of algorithms is YOLO (You only look once) commonly used for real-time object detection.

- Before we go into YOLOs details we have to know what we are going to predict. Our task is to predict a class of an object and the bounding box specifying object location. Each bounding box can be described using four descriptors:
- center of a bounding box ($\mathbf{b}_x$ $\mathbf{b}_y$)
- width ($\mathbf{b}_w$)
- height ($\mathbf{b}_h$)
- value **c** is corresponding to a class of an object (f.e. car, traffic lights,…).

YOLO Algorithm



- YOLO algorithm we're not searching for interested regions on our image that could contain some object. Instead of that we are splitting our image into cells, typically its 19×19 grid. Each cell will be responsible for predicting 5 bounding boxes (in case there's more than one object in this cell). This will give us 1805 bounding boxes for an image and that's a really big number!



Non-max suppression

- Removing boxes with low object probability and bounding boxes with the highest shared area in the process called **non-max suppression**.

- End