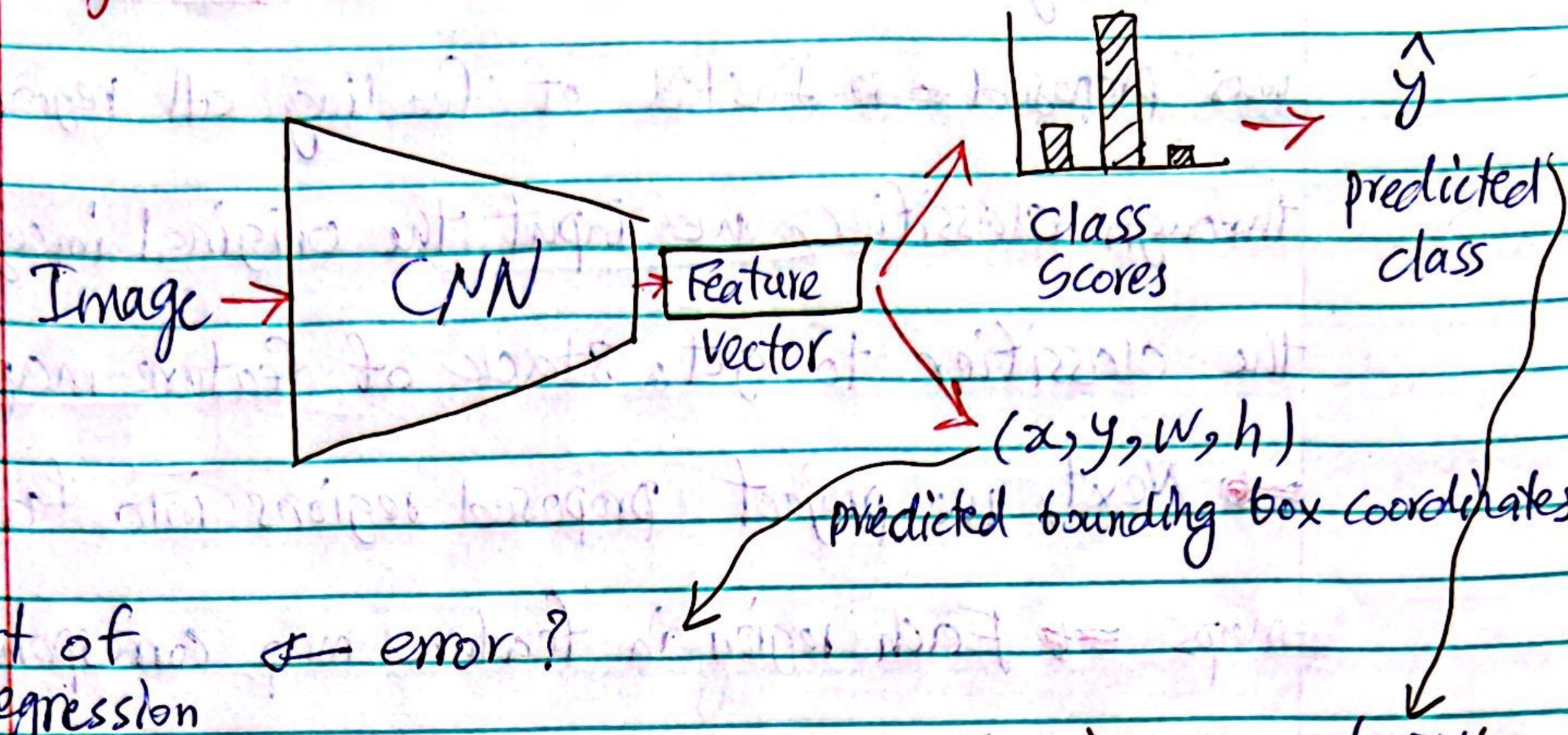
"Object Detection"



sort of regression ↳ L1, MSE, Smooth L1 loss	error?	Cross-entropy error

(3) The method presented above works for one object in the image → What if we have multiple of objects?!

→ We need region proposals! → We can randomly generate several croped parts of the image with different bounding box sizes → feed them to CNN → classify them as background or some specific objects! → Too SLOW! due to huge amount of croped images → This is RCNN which R stands for "regional" → We have several random regions → We make them standard size → feed them to classifier and so on!

① To remedy the slowness of R-CNN, Fast RCNN was proposed → Instead of feeding all regions through classifier, we input the original image into the classifier to get a stack of feature maps.

→ Next, we project proposed regions into feature maps → Each region in feature map corresponds to larger region in the original image → we can pass each small region in feature map into FC layer and evade time-consuming stage of image processing in CNN classifier.

after passing them from Region of Interest (ROI) pooling to make sizes uniform.

→ Fast R-CNN is faster than R-CNN, but not fast (10 times!) enough.

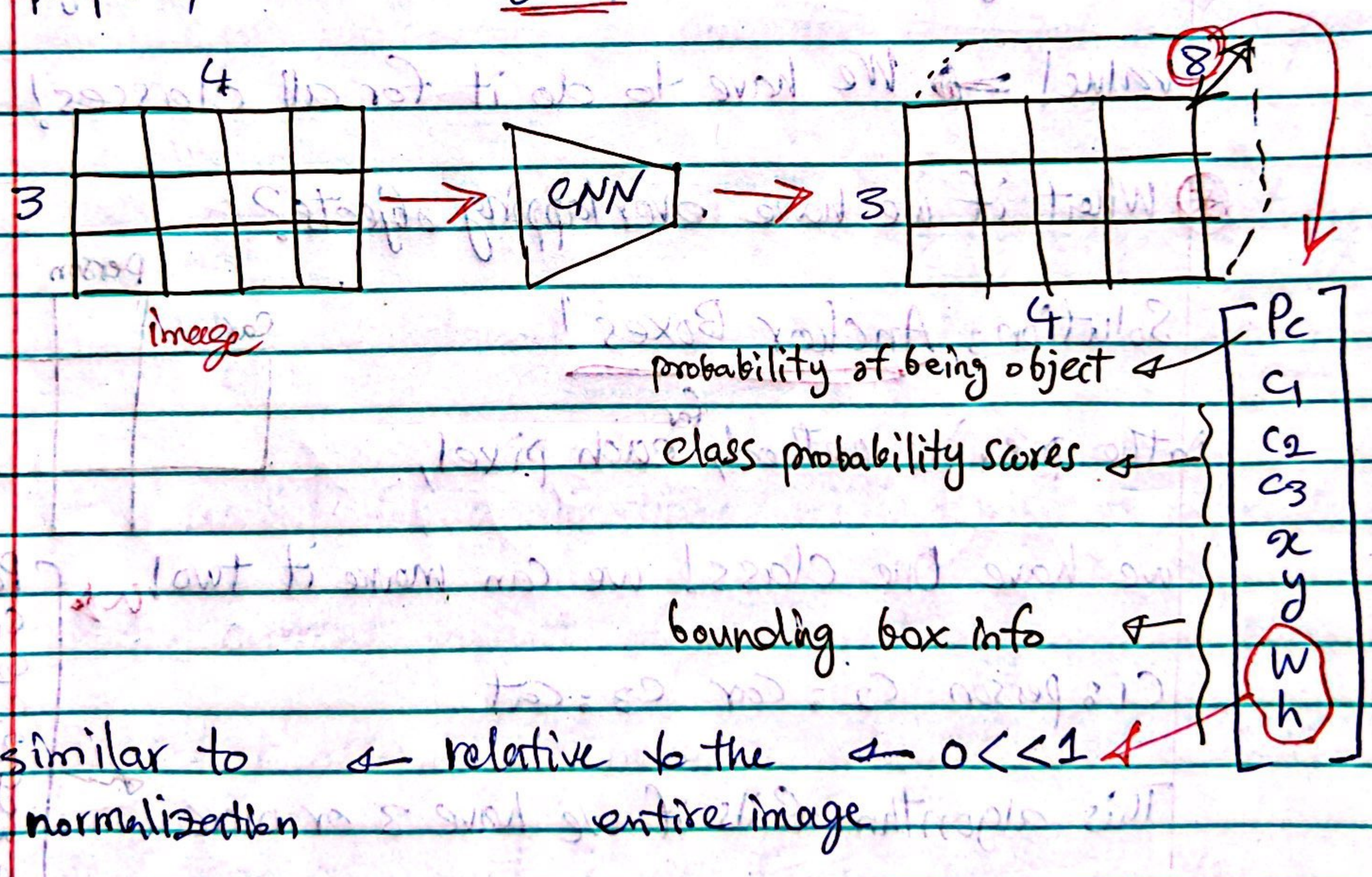
② Faster R-CNN, inputs the image into a network upto some layers (similar to Fast RCNN) and then feeds these processed feature into the Region Proposal Network → If some regions in the feature map is rich in detected edges or other features, it will

be proposed as RoI → Then for each region, the network performs a quick binary identification: object or non-object → Then discards non-Object regions → the Rest is similar to R-CNN.

(?) Is it possible to perform object detection without region proposals?

Yes! E.g., YOLO and SSD.

In YOLO, instead of using a sliding window for region proposal, we use a grid!



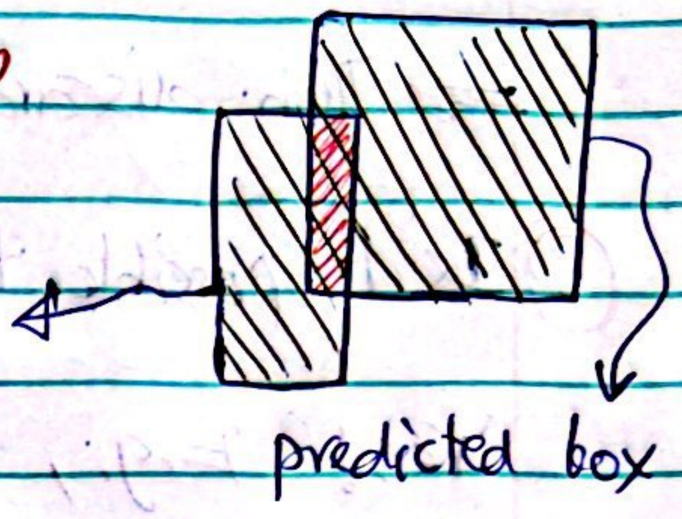
(*) How to resolve too many boxes? → non-maximal suppression

≡ finding bounding boxes that best matches an object in an image! → How we can compare two bounding boxes

in terms of being good bounding box for an object!

→ Answer: Intersecton of Union (IoU)

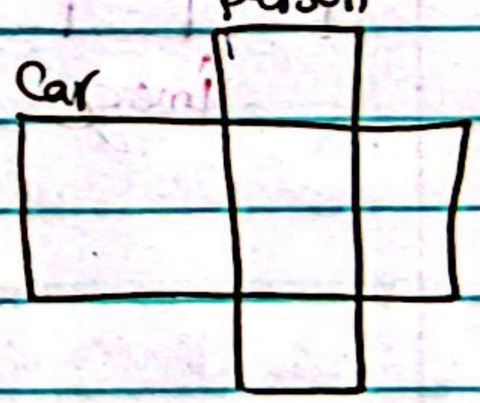
$$IoU = \frac{\text{Area of Interseetion}}{\text{Area of Union}}$$



→ Non-Maximal Suppression: removes overlapping bounding boxes and keep the box with the highest p_c value! → We have to do it for all classes!

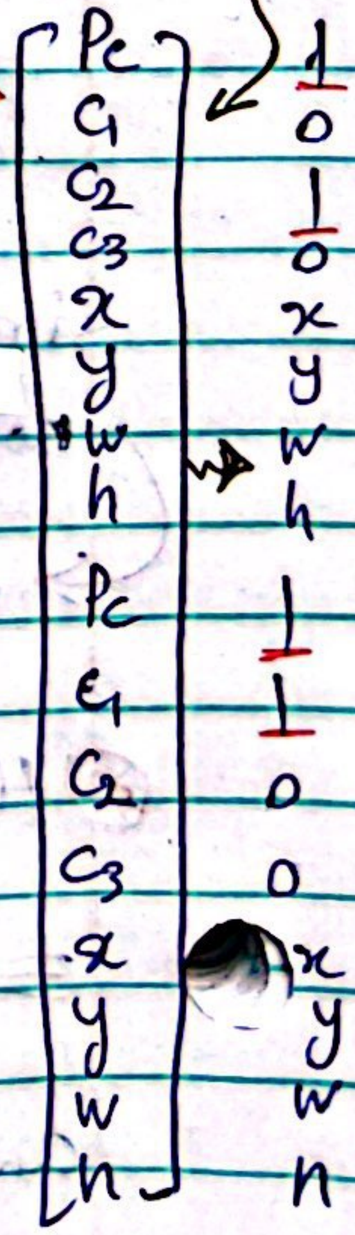
⊛ What if we have overlapping objects?

Solution: Anchor Boxes!



In the previous method, for each pixel, we have one class! we can make it two!

c_1: person c_2: car c_3: cat



This algorithm fails if we have 3 or more overlapping objects. → but it is rare!