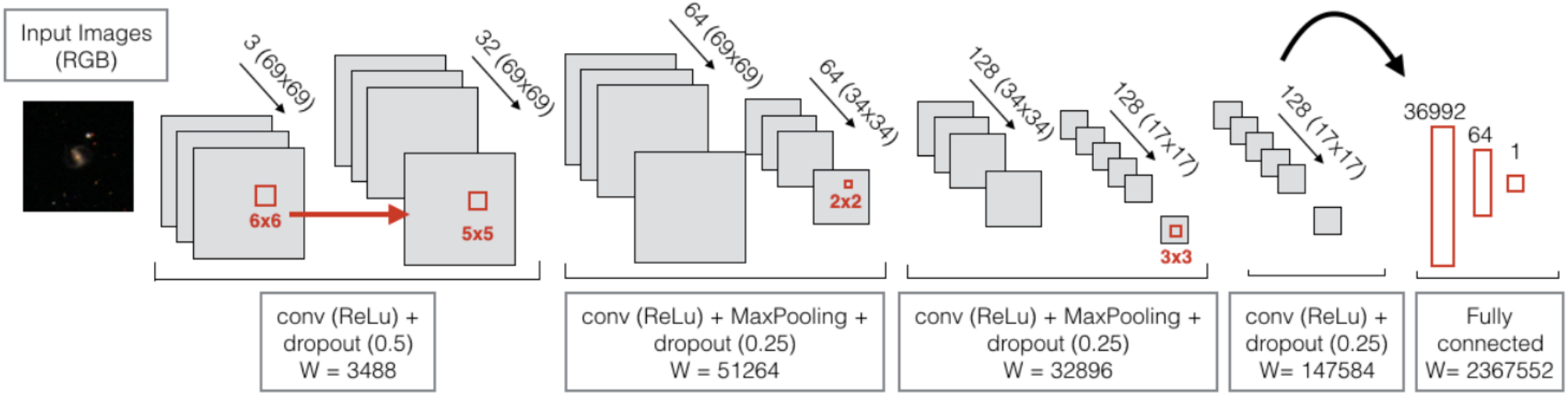
BEYOND CLASSIFICATION: IMAGE2IMAGE NETWORKS

UP TO NOW CNNs MAP IMAGES (SIGNALS) INTO FLOATS



Dominguez-Sanchez+18

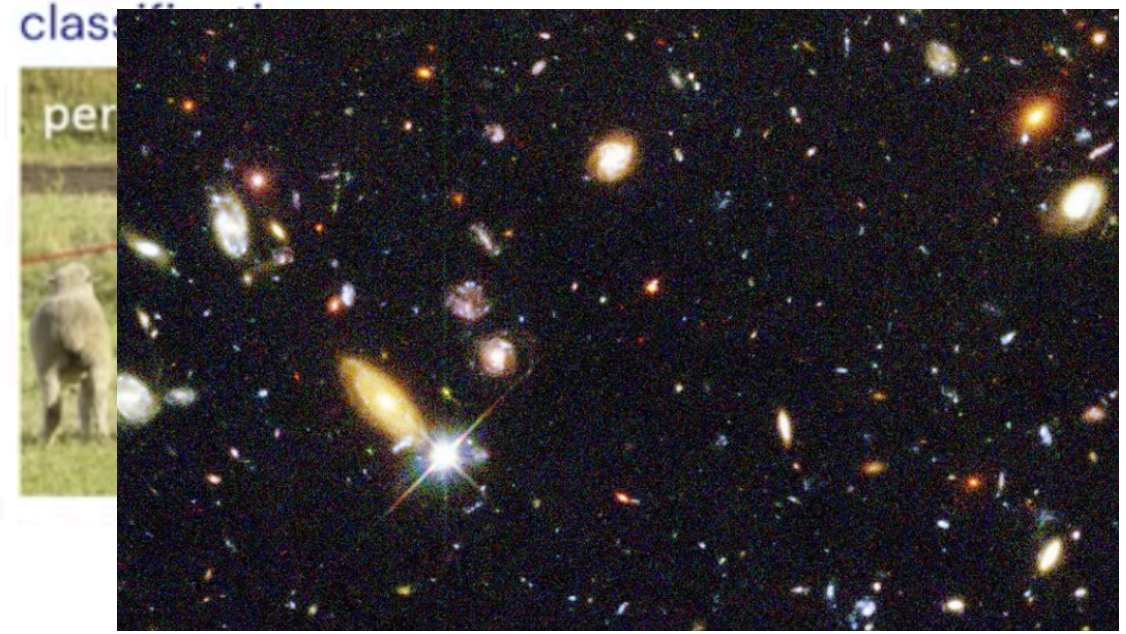
Classification has its limits ….

HOW DO I CLASSIFY THIS IMAGE?

Classification has its limits ….

HOW DO I CLASSIFY THIS IMAGE?

Classification has its limits ….

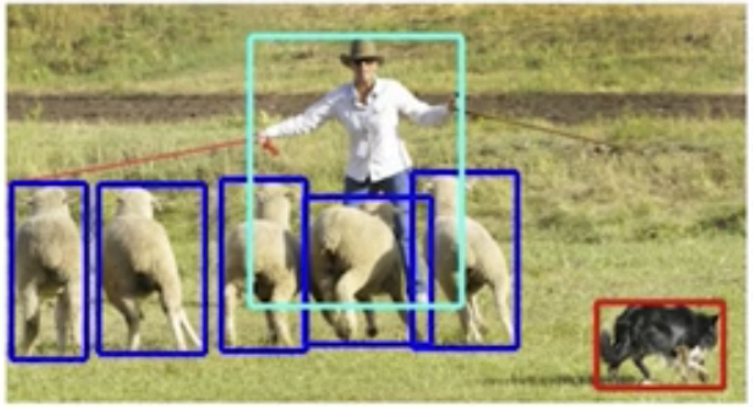


HOW DO I CLASSIFY THIS IMAGE?

Going beyond classification: increasing complexity

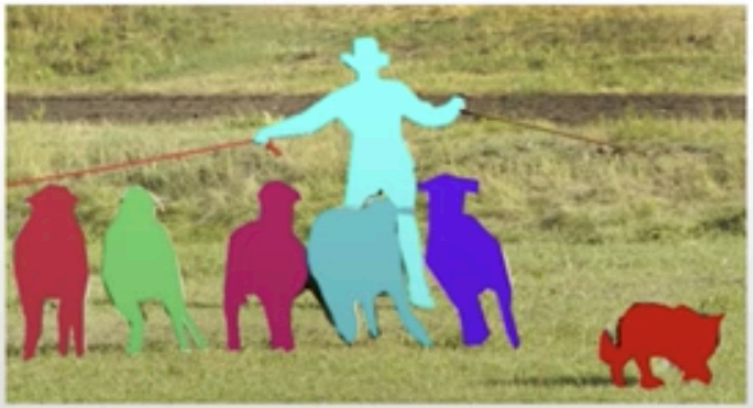
classification

object detection



semantic segmentation

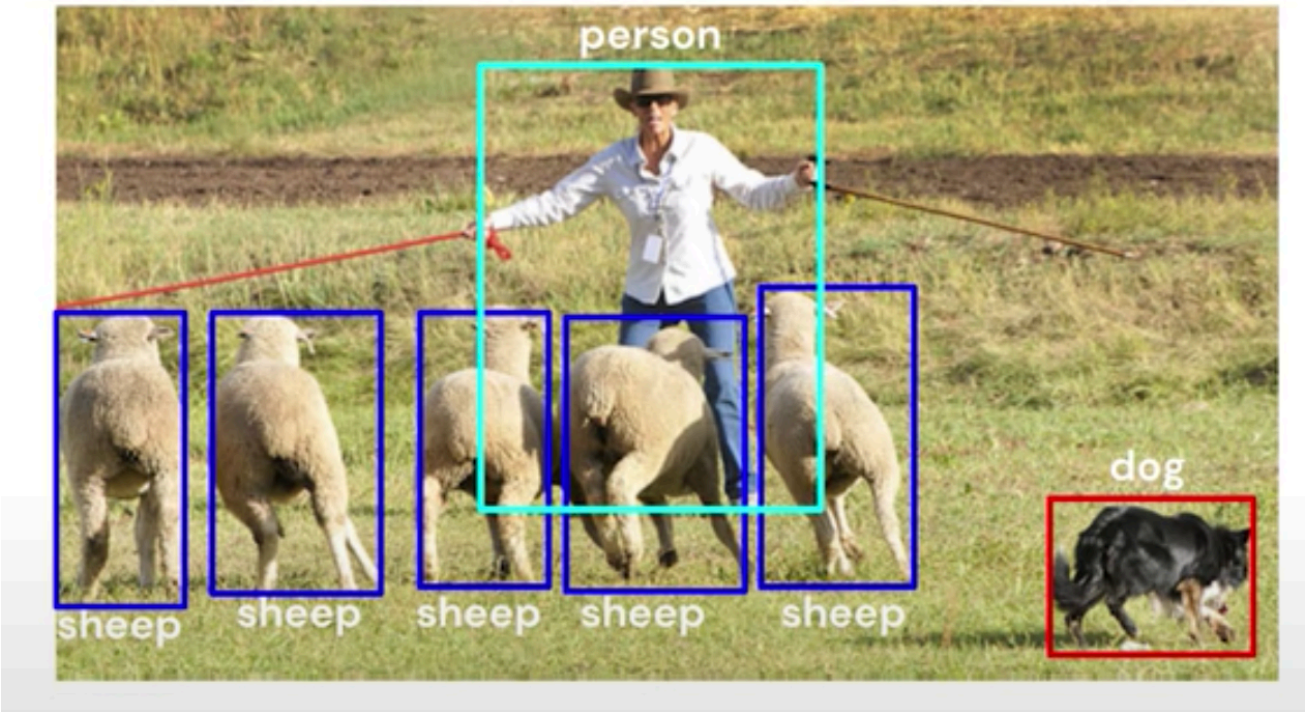
instance segmentation



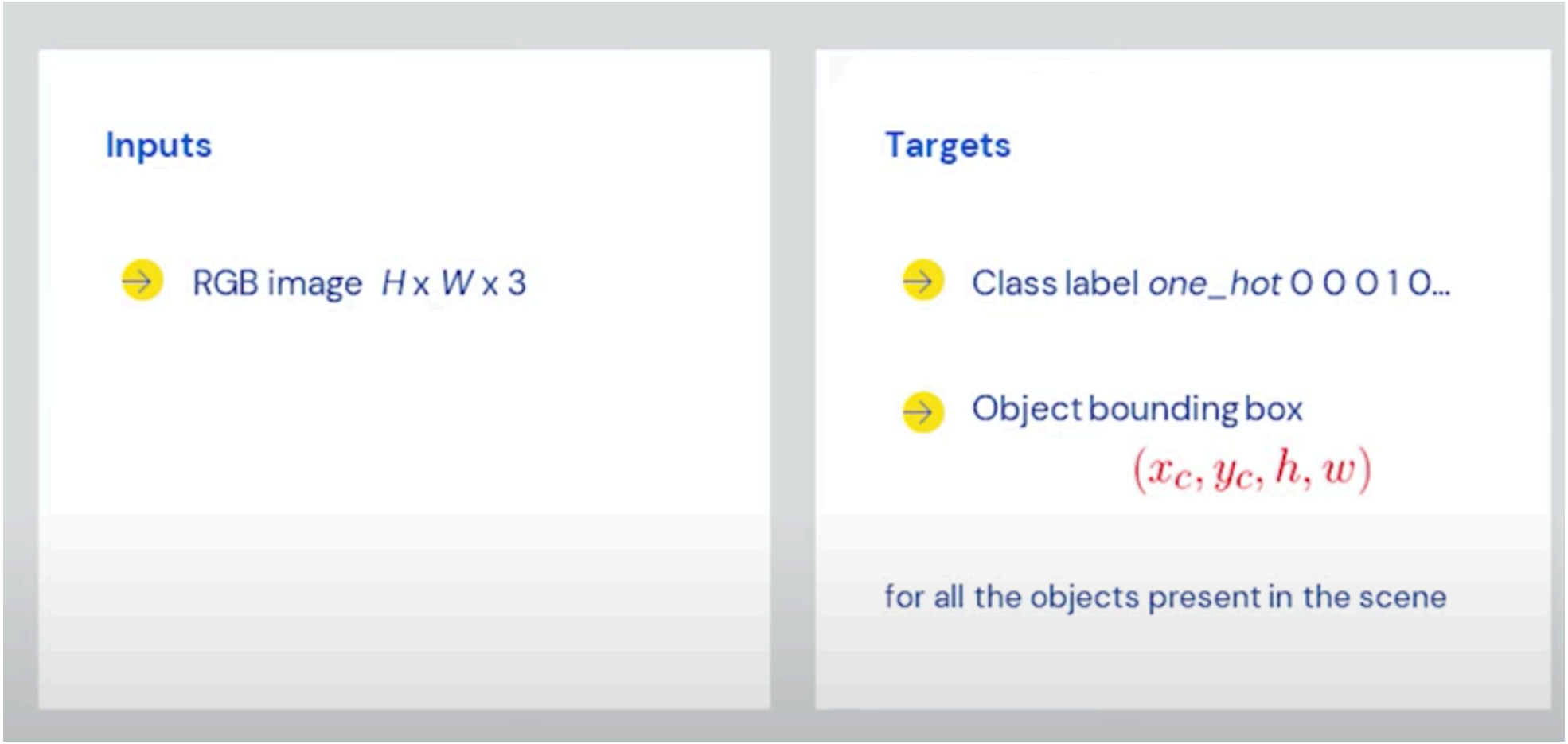
@deepmind (V. Patraucean)

Object detection

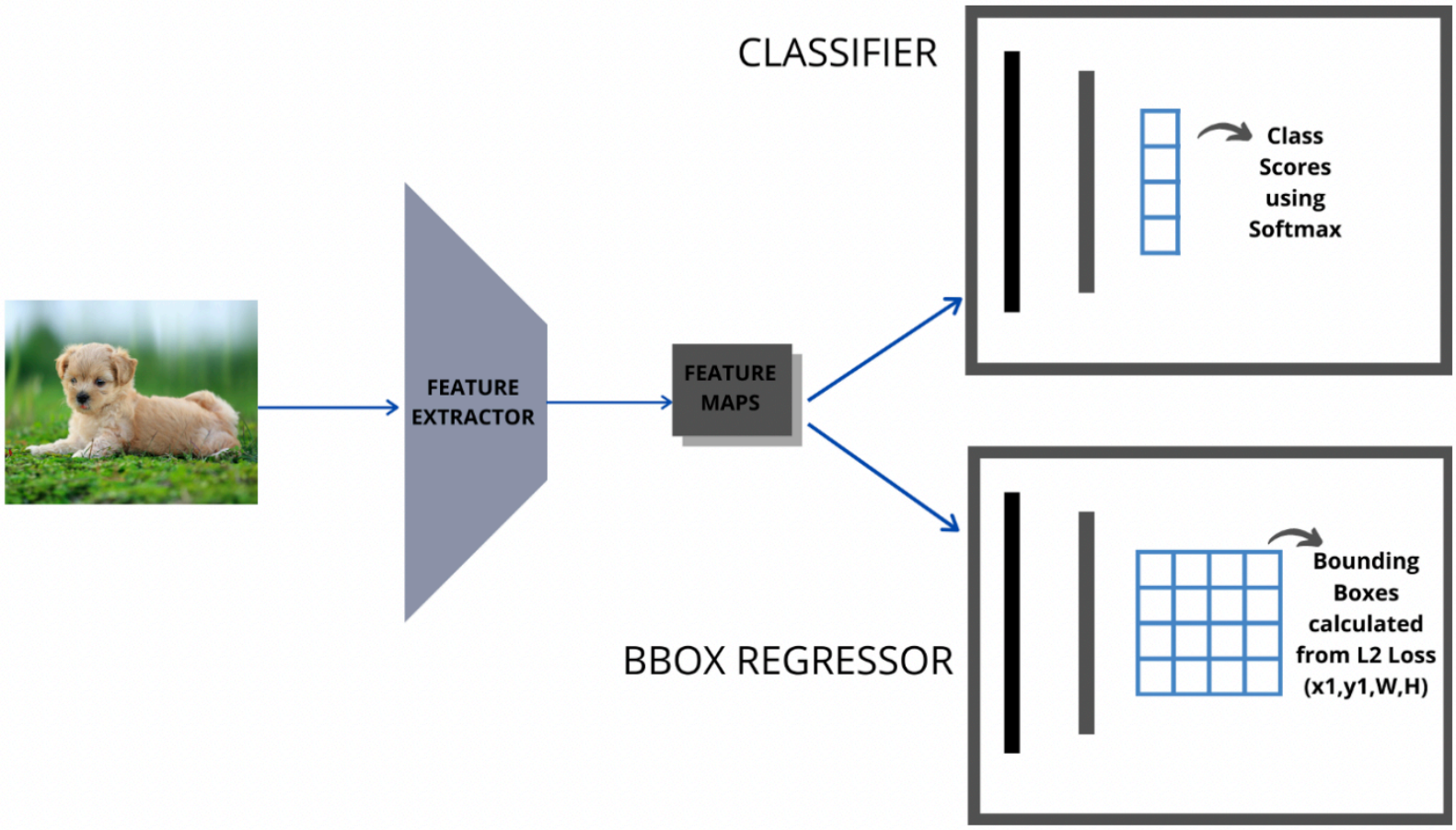
First task is to find a bounding box for every object. How we do that?



@deepmind (V. Patraucean)



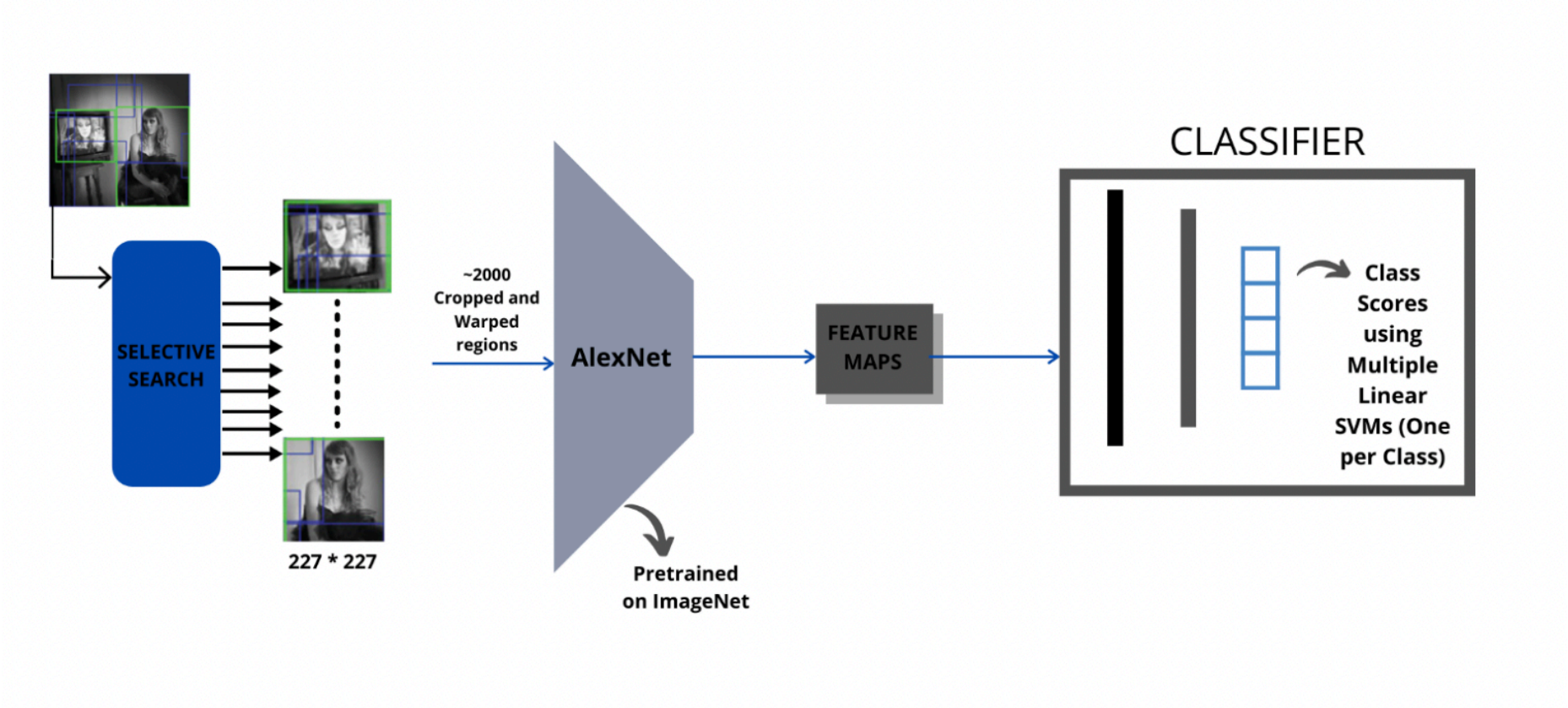
@deepmind (V. Patraucean)


CLASSIFIER
Class Scores using Softmax
FEATURE EXTRACTOR
FEATURE MAPS
BBOX REGRESSOR
Bounding Boxes calculated from L2 Loss (x1,y1,W,H)

The first ideas were based on first using existing methods such as Hierarchical Clustering

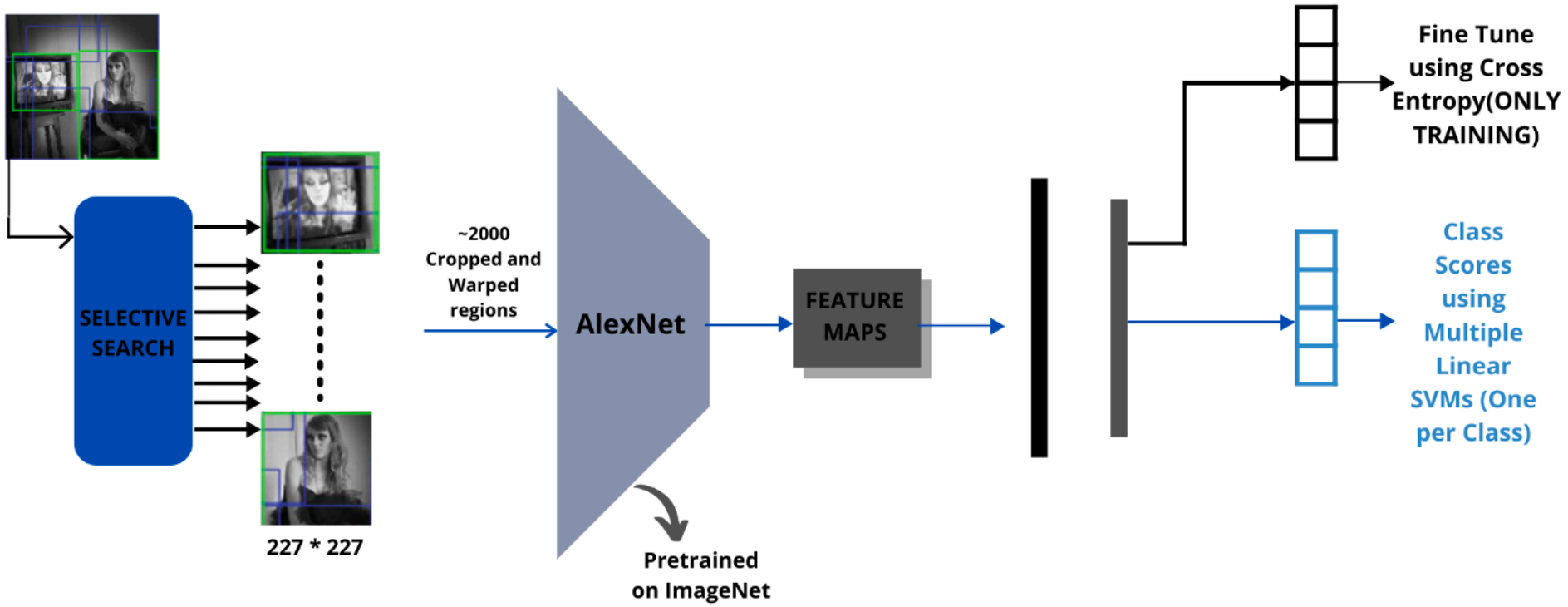
Grouping of pixels based on texture, color, composition …

R-CNN

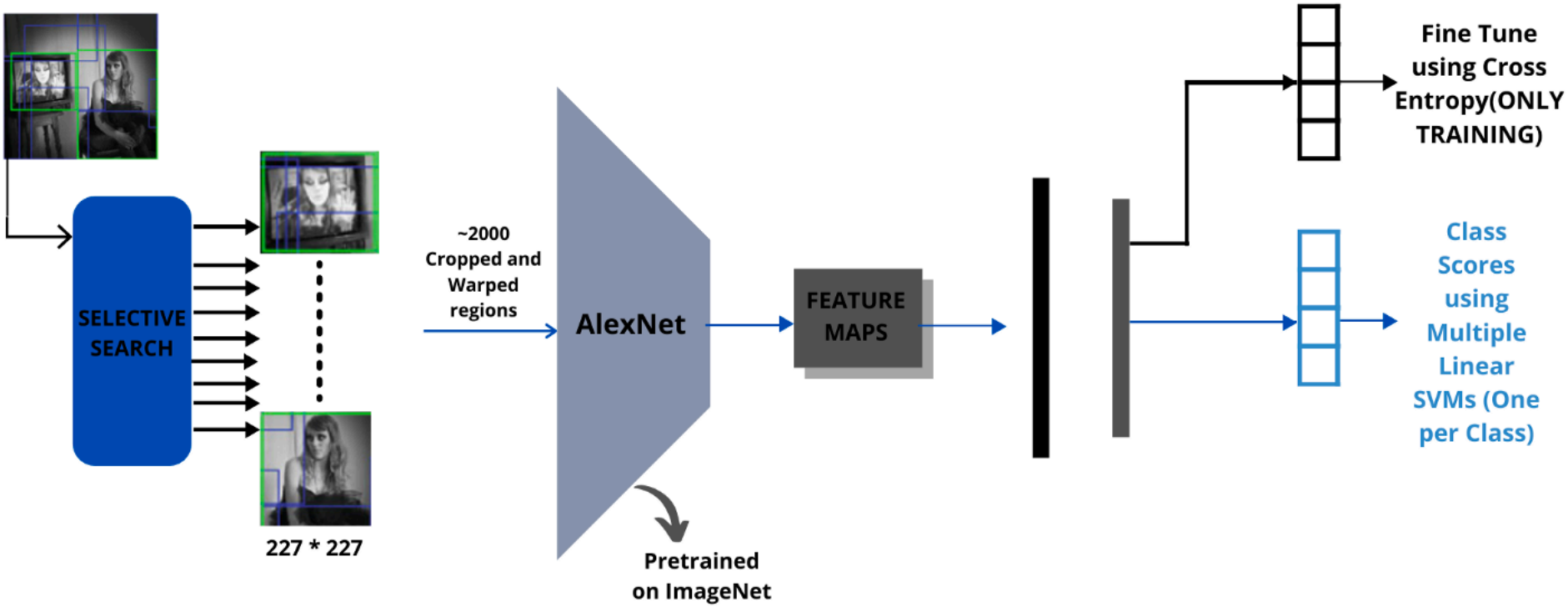


The hierarchical clustering is then combined with a CNN for classification of each region proposal.

The accuracy of this approach is not very high, requires changing the image aspect ratios

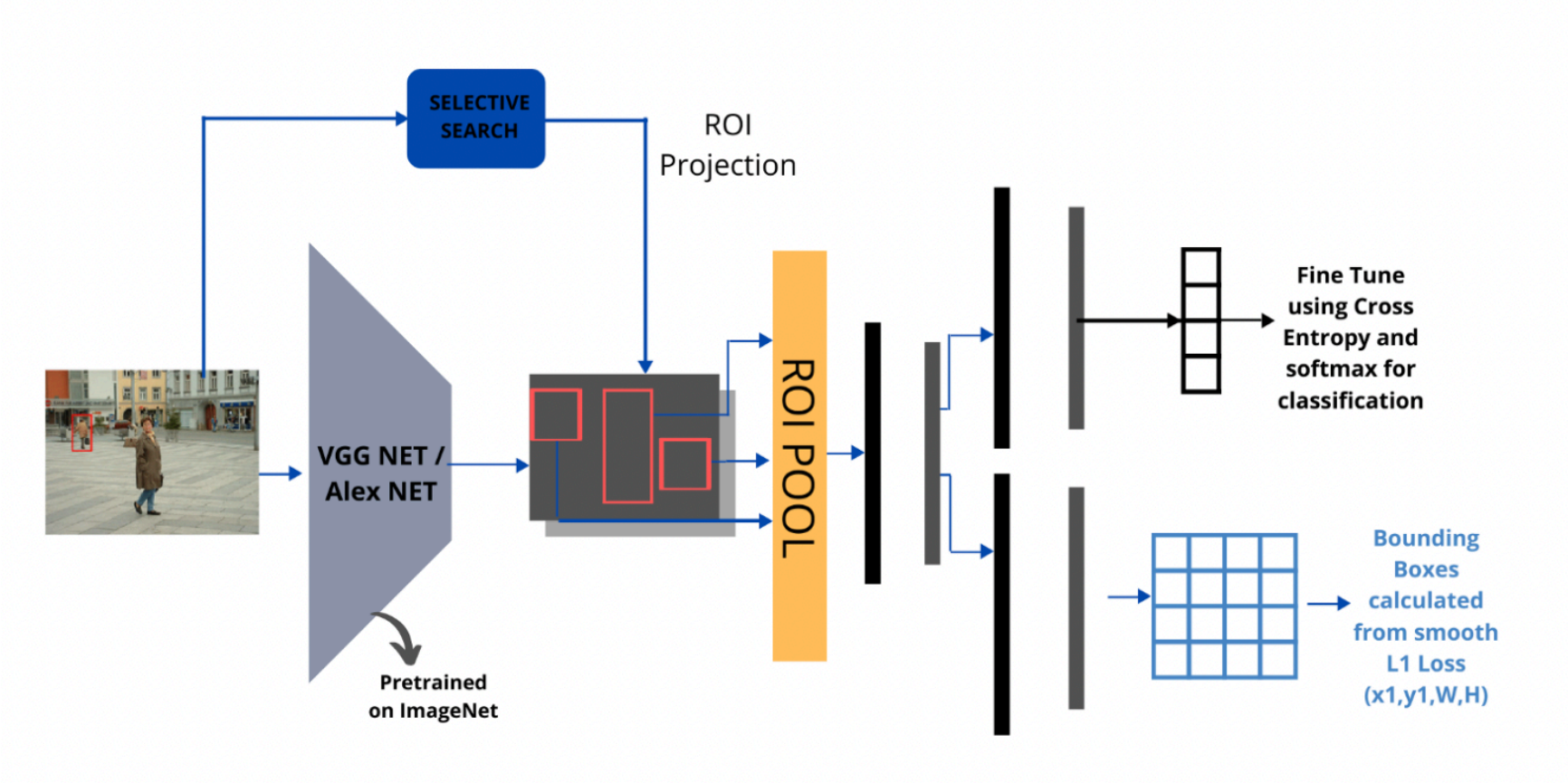


A solution was to add a fine tuning of the weights using an additional cross entropy loss

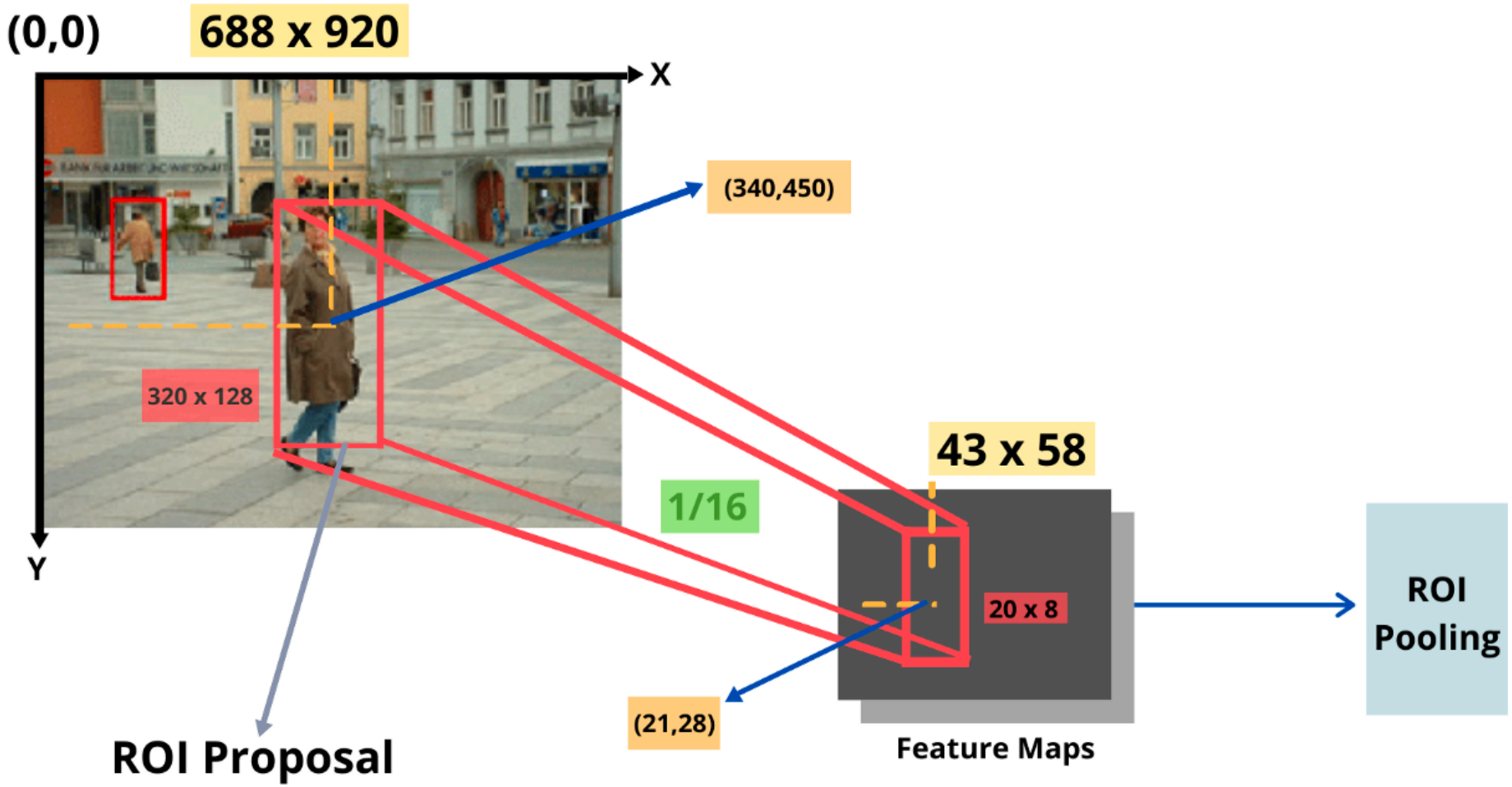


However, this requires a forward pass for every region proposal, which can be high

Fast R CNN

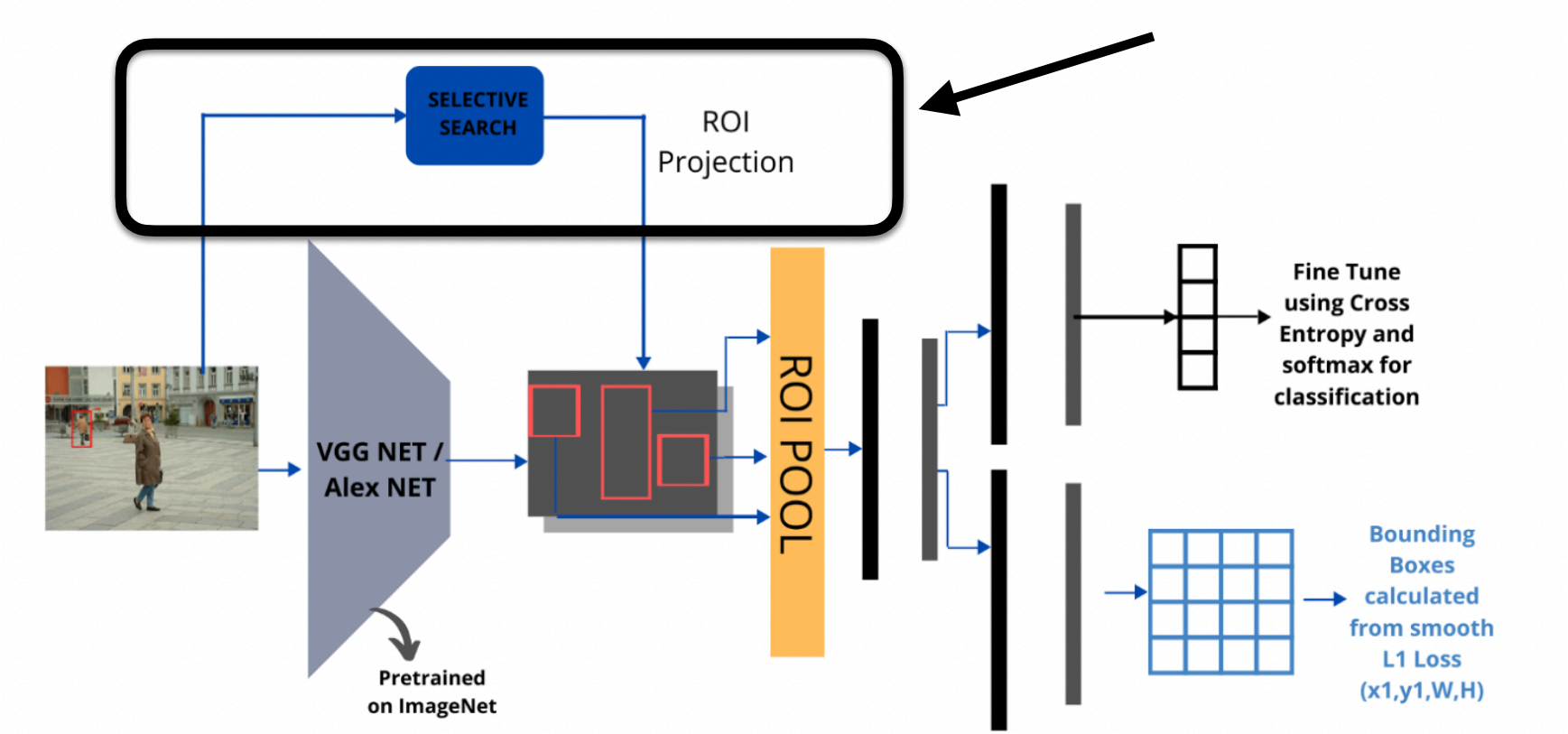


Fast R CNN uses only pass of the entire image by the CNN.


(0,0)
688 x 920
X
Y
(340,450)
320 x 128
1/16
43 x 58
20 x 8
(21,28)
Feature Maps
ROI Pooling
ROI Proposal

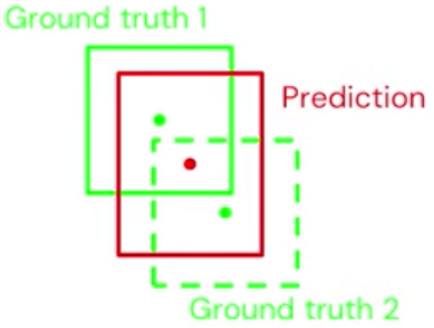
Fast R CNN

Why not
transforming this into a CNN?



WHAT WOULD BE THE LOSS FUNCTION OF SUCH A PROBLEM?

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - p_i)^2$$

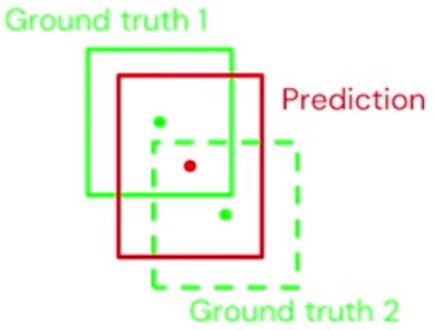


IT IS A REGRESSION WITH A SIMPLE QUADRATIC LOSS. WE TRY TO FIND THE BEST COORDINATES OF THE BOUNDING BOX.

@deepmind (V. Patraucean)

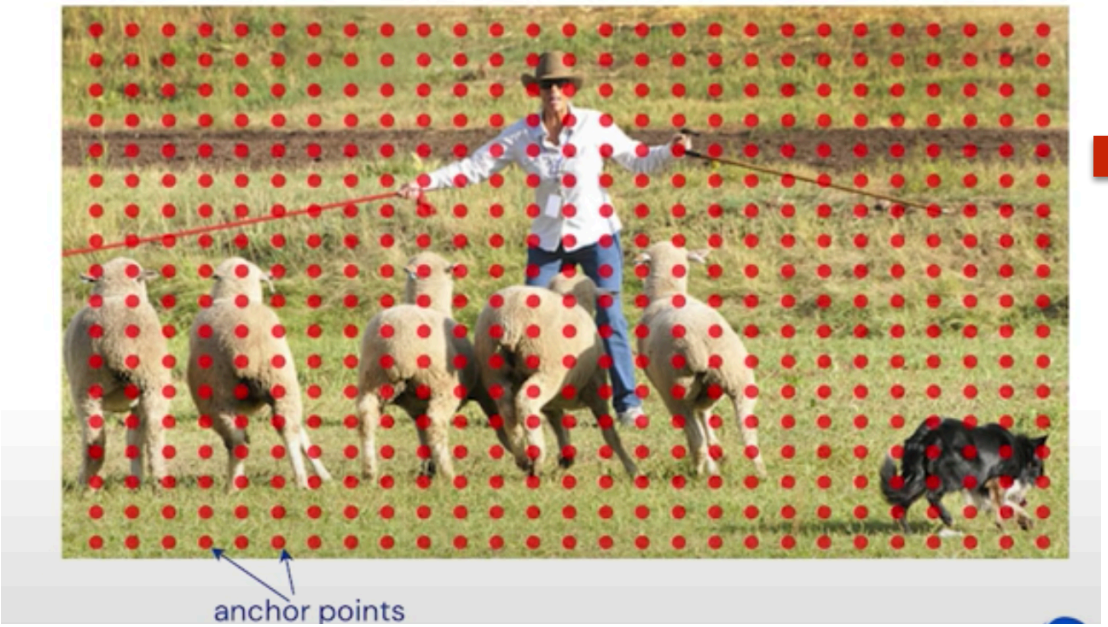
WHAT WOULD BE THE LOSS FUNCTION OF SUCH A PROBLEM?

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - p_i)^2$$



IT IS A REGRESSION WITH A SIMPLE QUADRATIC LOSS. WE TRY TO FIND THE BEST COORDINATES OF THE BOUNDING BOX.

IT BECOMES MESSY QUITE RAPIDLY…



Discretize Bounding Box Space

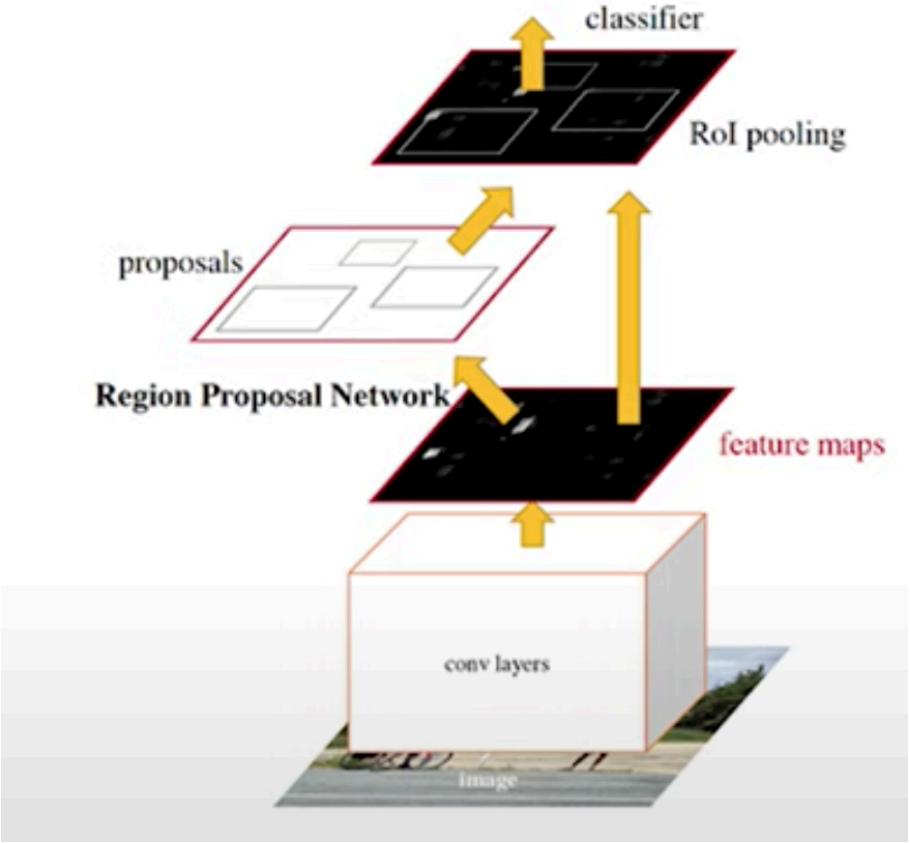
Choose n candidates per position

Predict objectness score (classification)

Sort and keep top K

@deepmind (V. Patraucean)

FASTER R-CNN



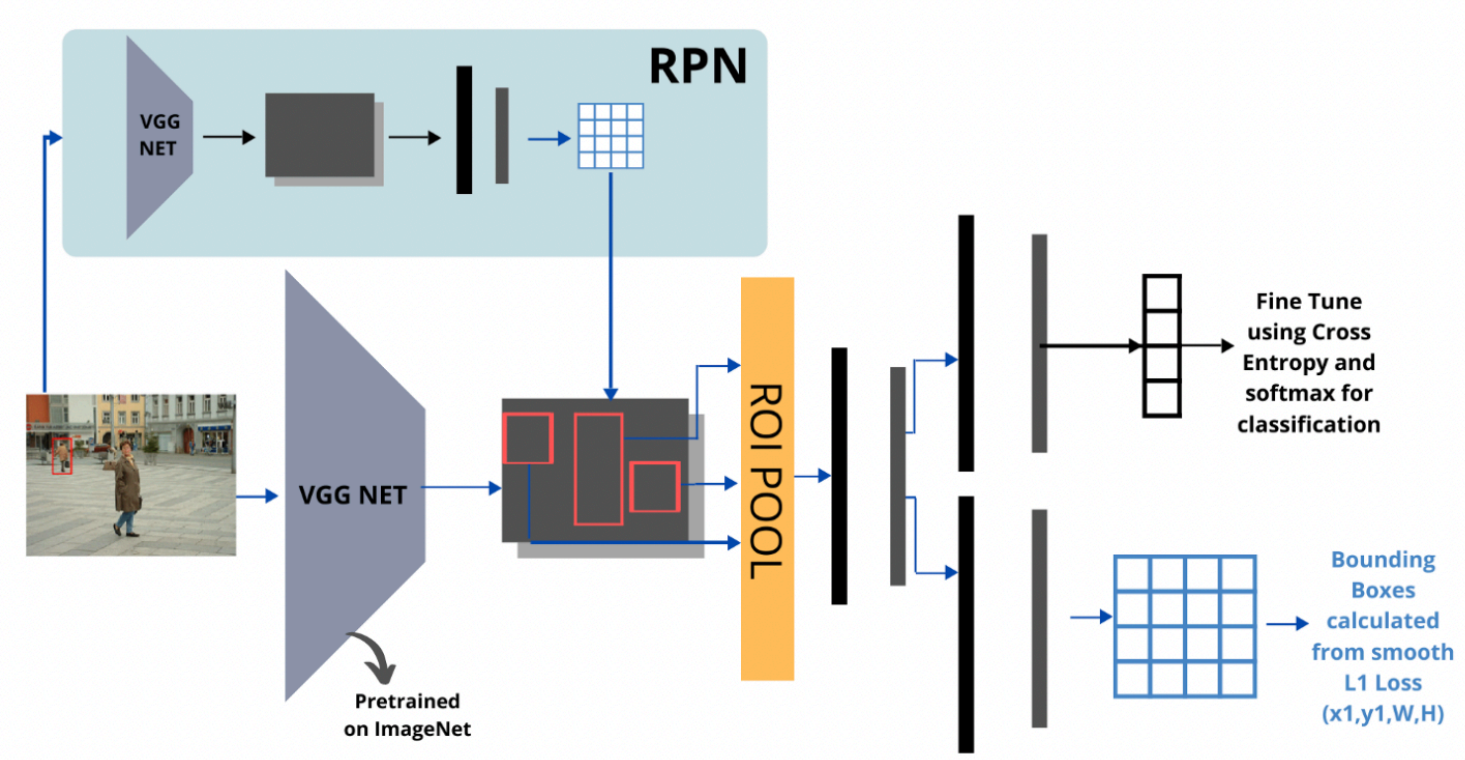
We divide the task in 2 steps:

1. Identify bounding box candidates (CLASSIFICATION)

2. Classify and Refine (REGRESSION)

Ren+16

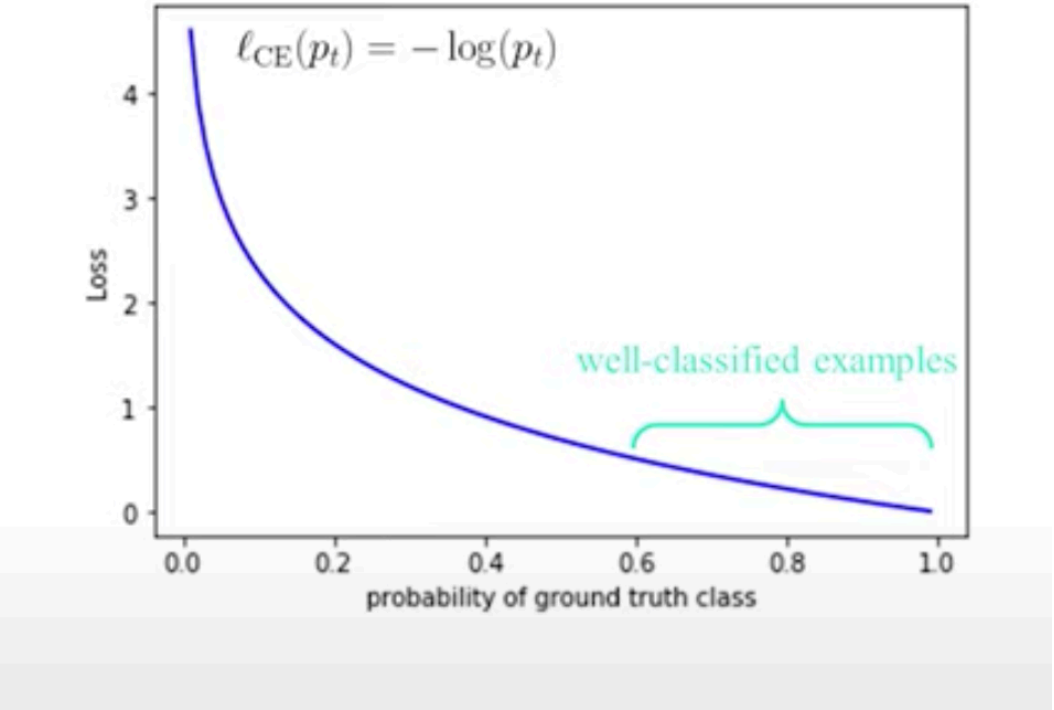
FASTER R-CNN


RPN
VGG NET
VGG NET
Pretrained on ImageNet
ROI POOL
Fine Tune using Cross Entropy and softmax for classification
Bounding Boxes calculated from smooth L1 Loss (x1,y1,W,H)

WHY NOT DOING IT ONE STAGE?

MOST OF THE CANDIDATES
ARE BACKGROUND,
EASY TO IDENTIFY

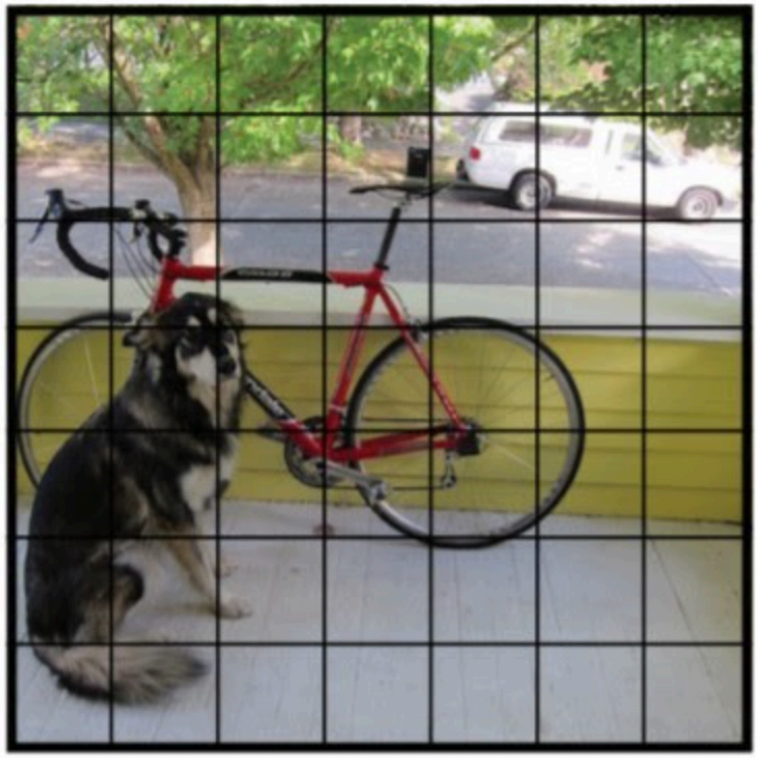
THE LOSS OF THE MANY
EASY EXAMPLES
DOMINATES OVER THE
RARE USEFUL ONES



@deepmind (V. Patraucean)

YOLO: You Only Look Once

Transform everything in a big regression



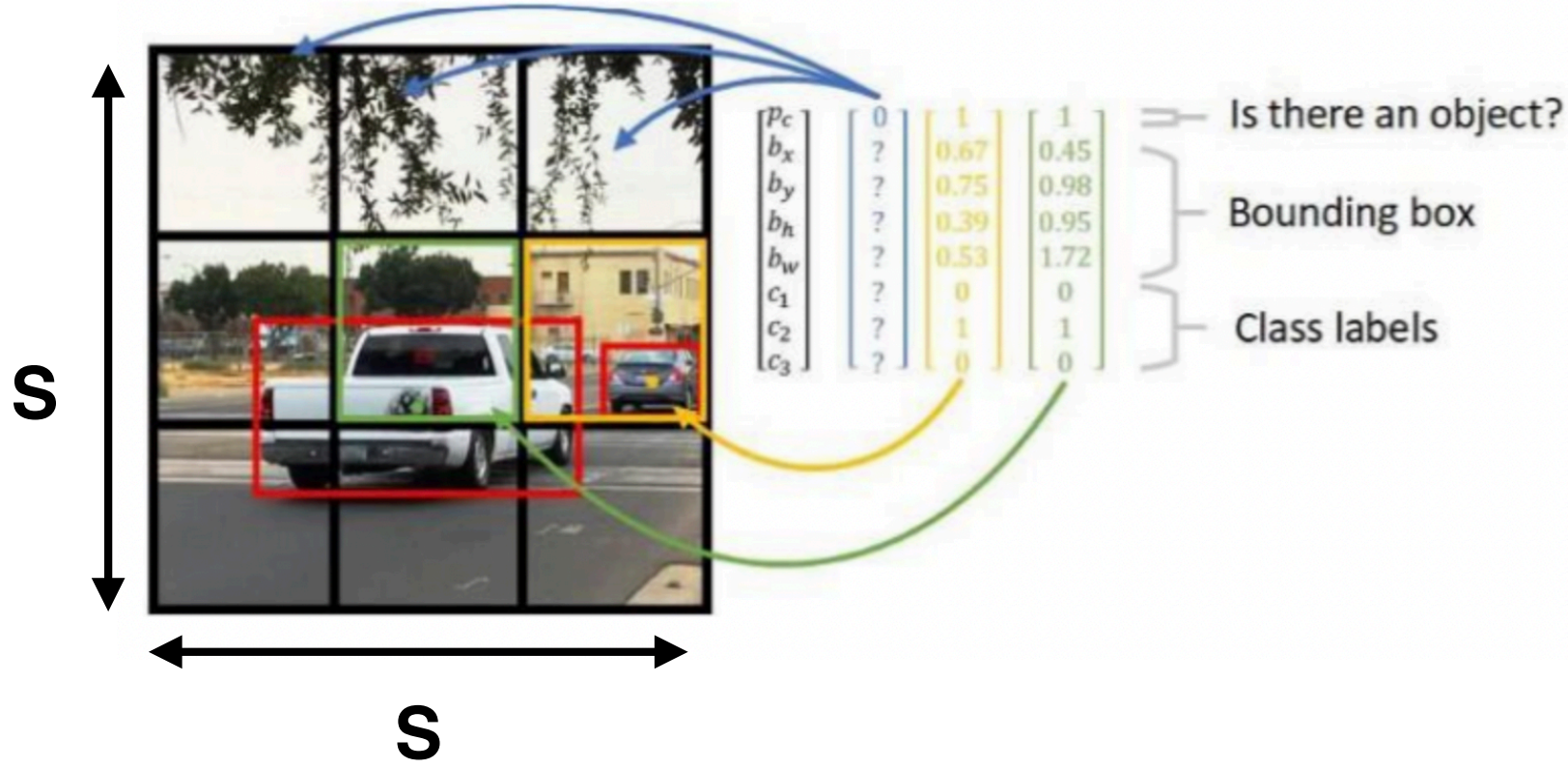
S × S grid on input

1. Divide the image in cells

2. Each cell is responsible for detecting B bounding boxes as well as a confidence class for each box

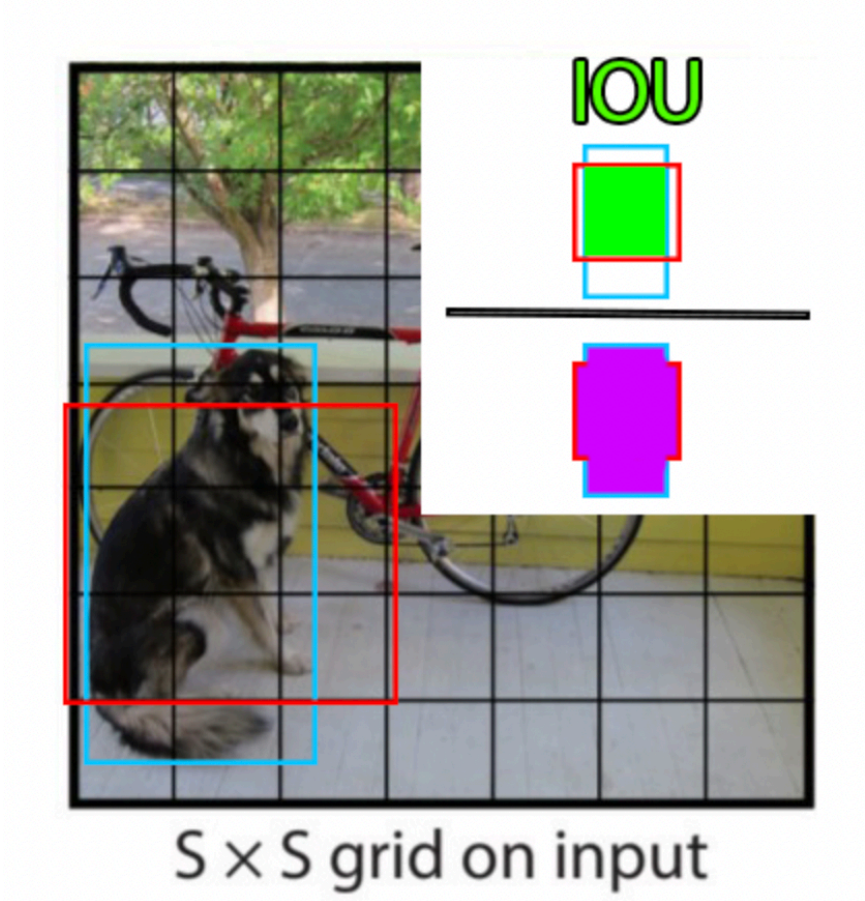
Each bounding box is parametrised with 5 numbers:
- coordinates: x,y
- heights width: w, h
- confidence: c

Then we attach to each bounding box, a one-hot encoder vector contains the class labels

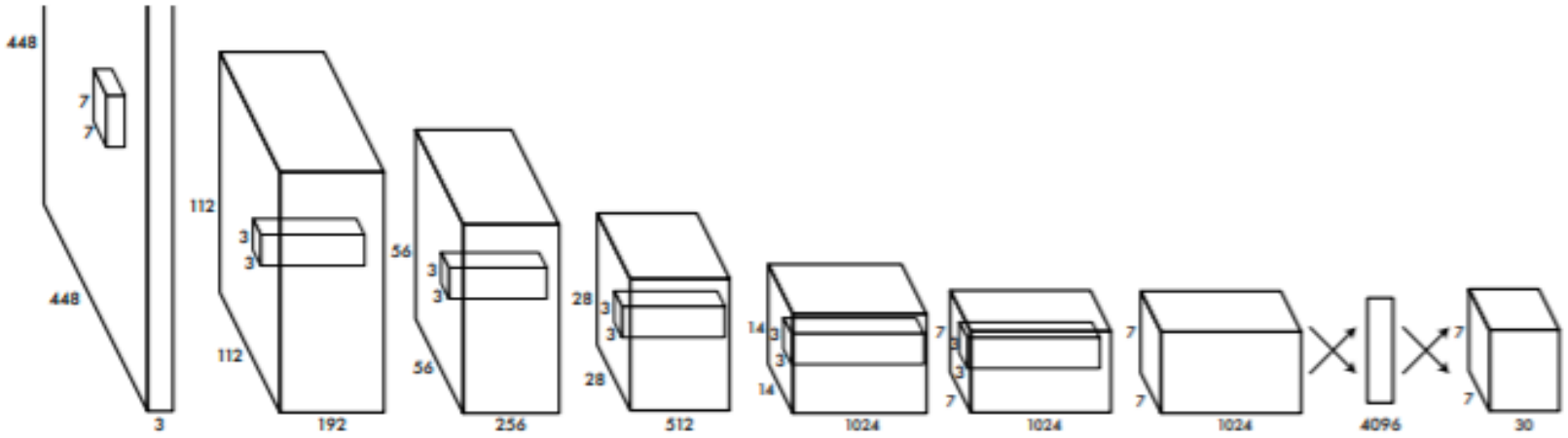


Measuring the “confidence” of a box

WE TYPICALLY USE THE INTERSECTION OVER UNION LOSS



YOLO REGRESSION NETWORK



$$S \times S \times (C + B * 5)$$

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2$$

loss related to the predicted bounding box position (x,y)

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2$$

loss related to the predicted bounding box height and width

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

loss related to the correctness of the bounding box