Object Detection using Convolutional Neural Networks

Masoud Pourreza

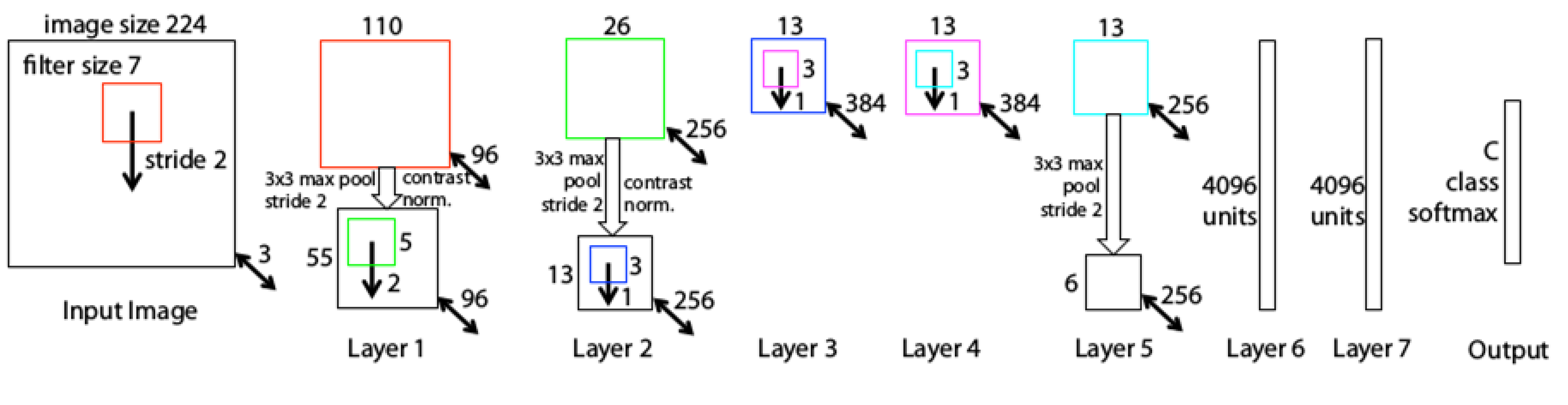
Pourreza.masoud@gmail.com

@masoudpz

HAMIM

Spatial feature

- ZF net (2013)



ZF Net Architecture

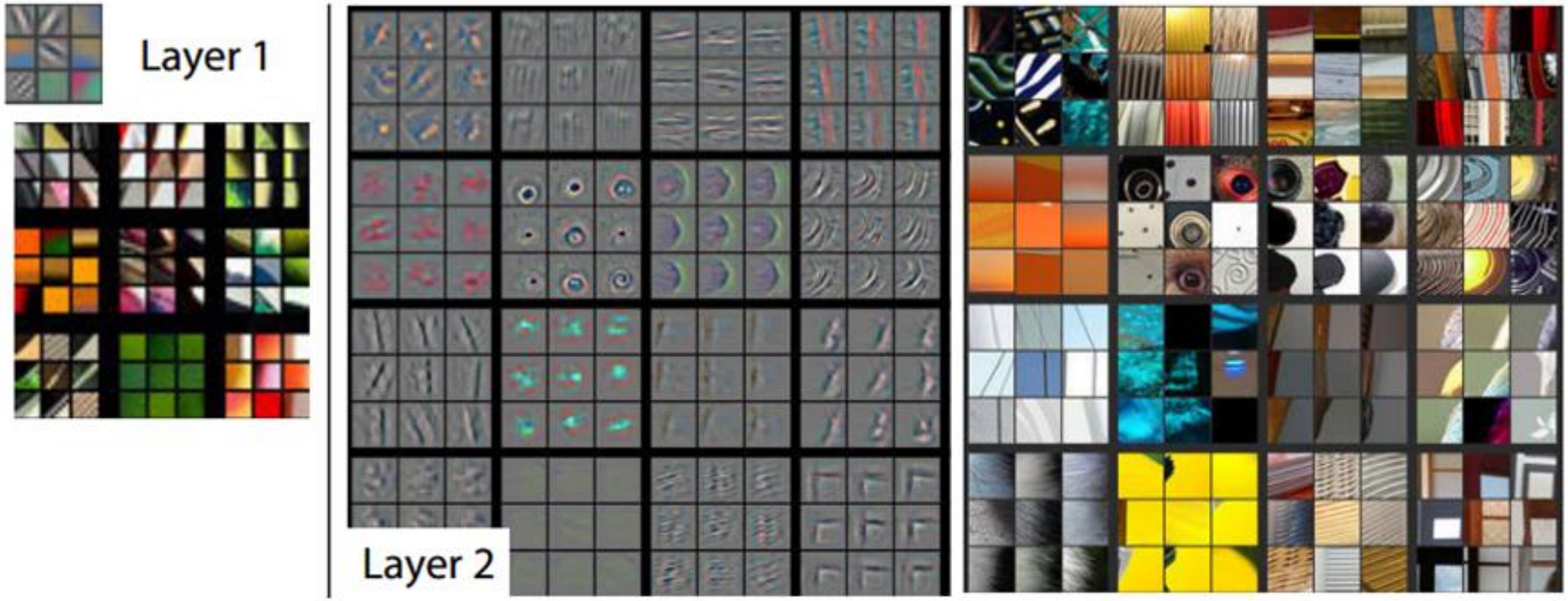
- Very similar architecture to AlexNet, except for a few minor modifications

Spatial Feature

- AlexNet trained on 15 million images, while ZF Net trained on only 1.3 million images.
- Instead of using 11x11 sized filters in the first layer (which is what AlexNet implemented), ZF Net used filters of size 7x7 and a decreased stride value. The reasoning behind this modification is that a smaller filter size in the first conv layer helps retain a lot of original pixel information in the input volume.
- A filtering of size 11x11 proved to be skipping a lot of relevant information, especially as this is the first conv layer. As the network grows, we also see a rise in the number of filters used.

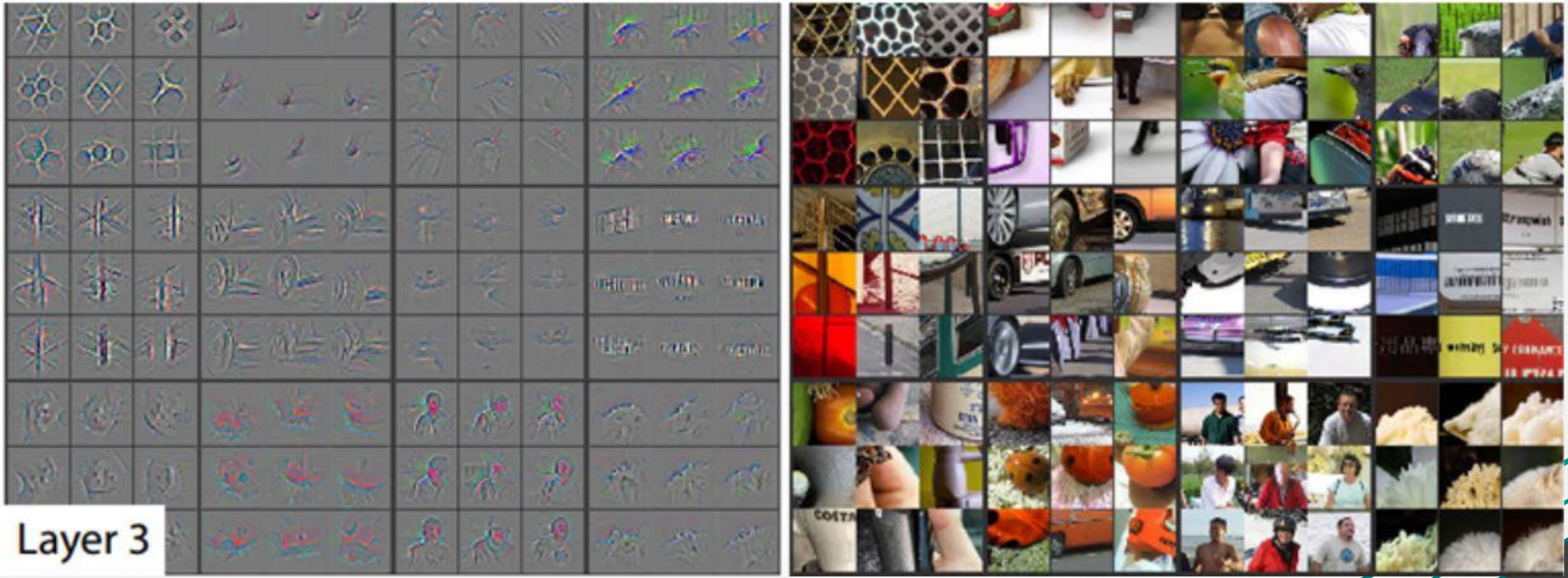
HAMIM©

Spatial features

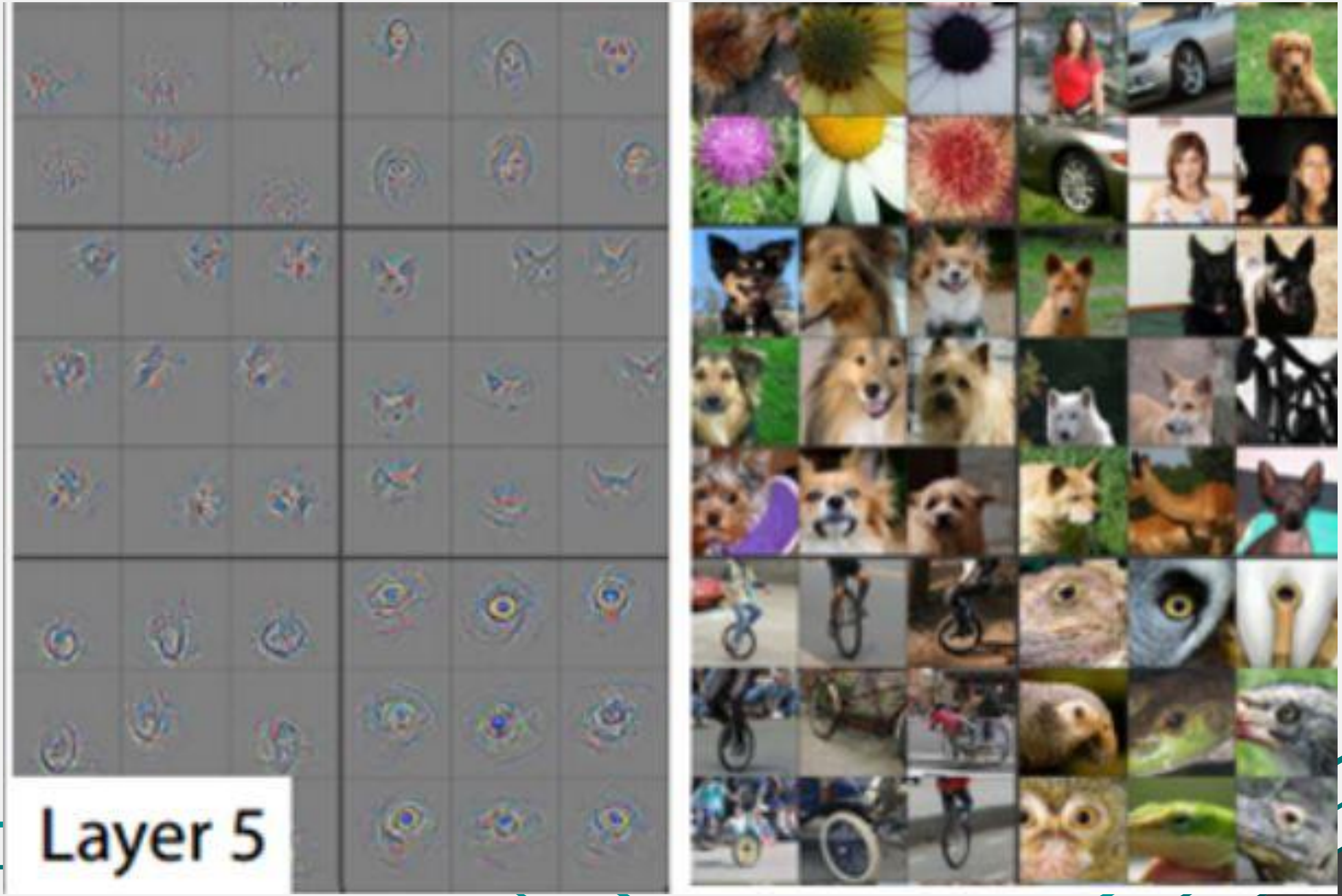


Visualizations of Layer 1 and 2. Each layer illustrates 2 pictures, one which shows the filters themselves and one that shows what part of the image are most strongly activated by the given filter. For example, in the space labled Layer 2, we have representations of the 16 different filters (on the left)

Spatial features

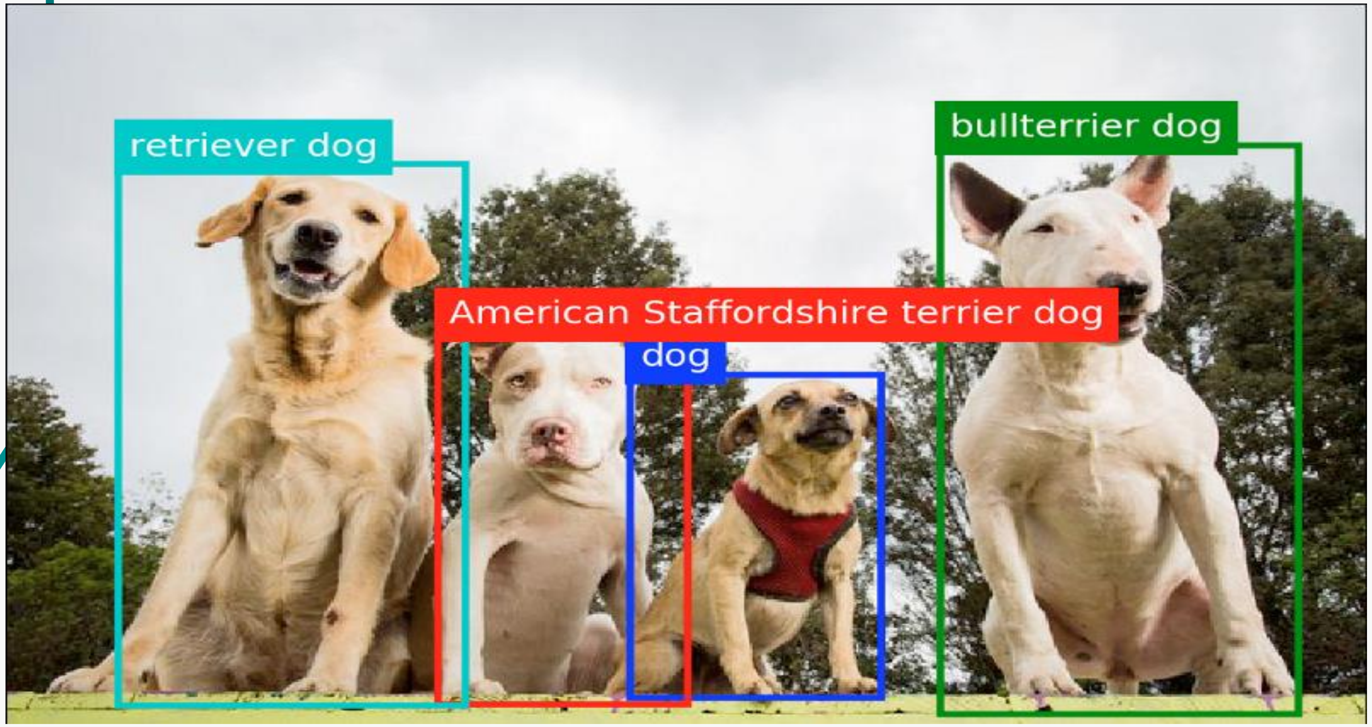


Spatial features



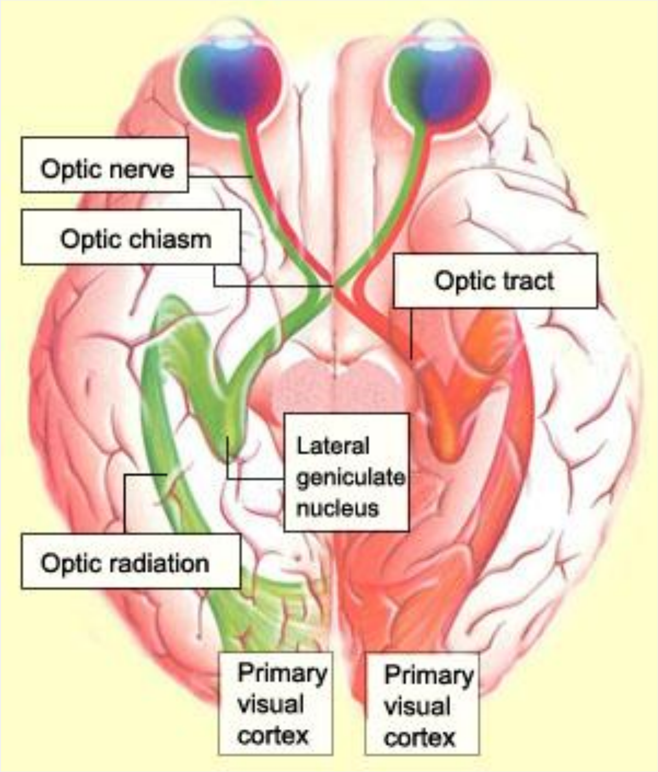
Object Detection Problem

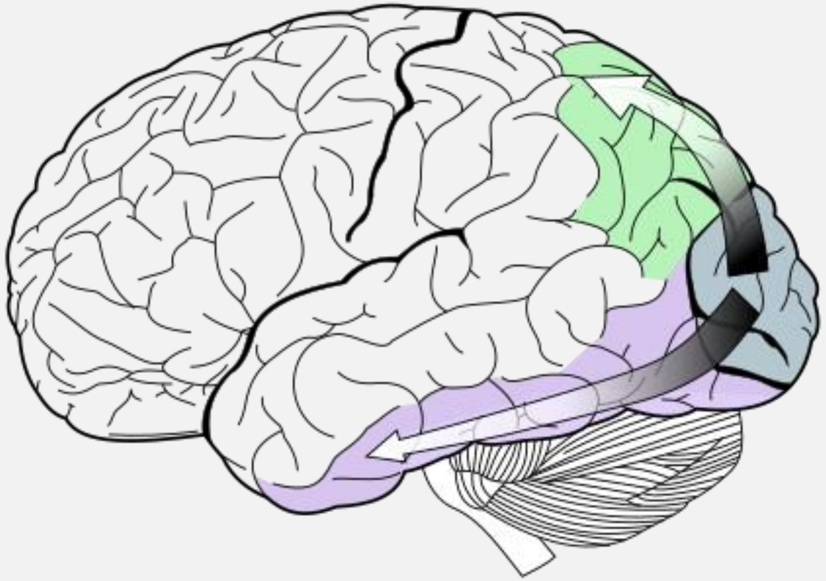
- Detection vs Recognition



HAMIM©

The dorsal and ventral stream in brain

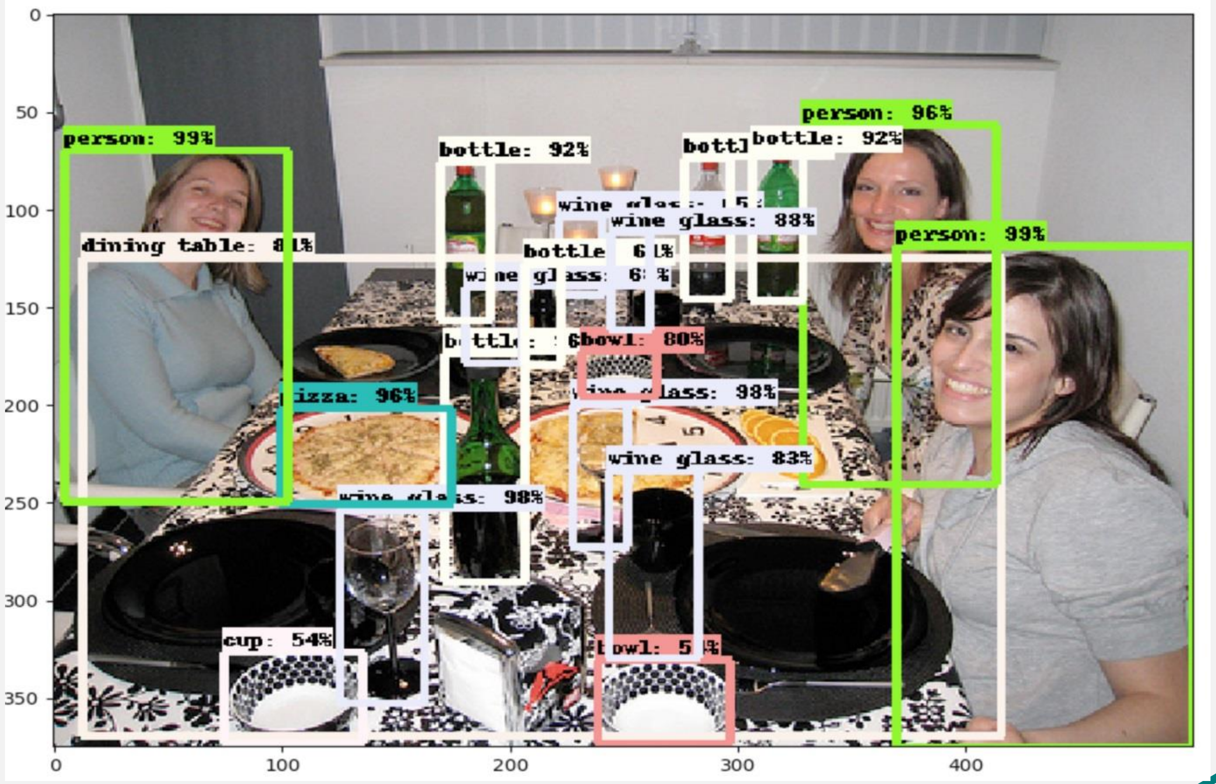




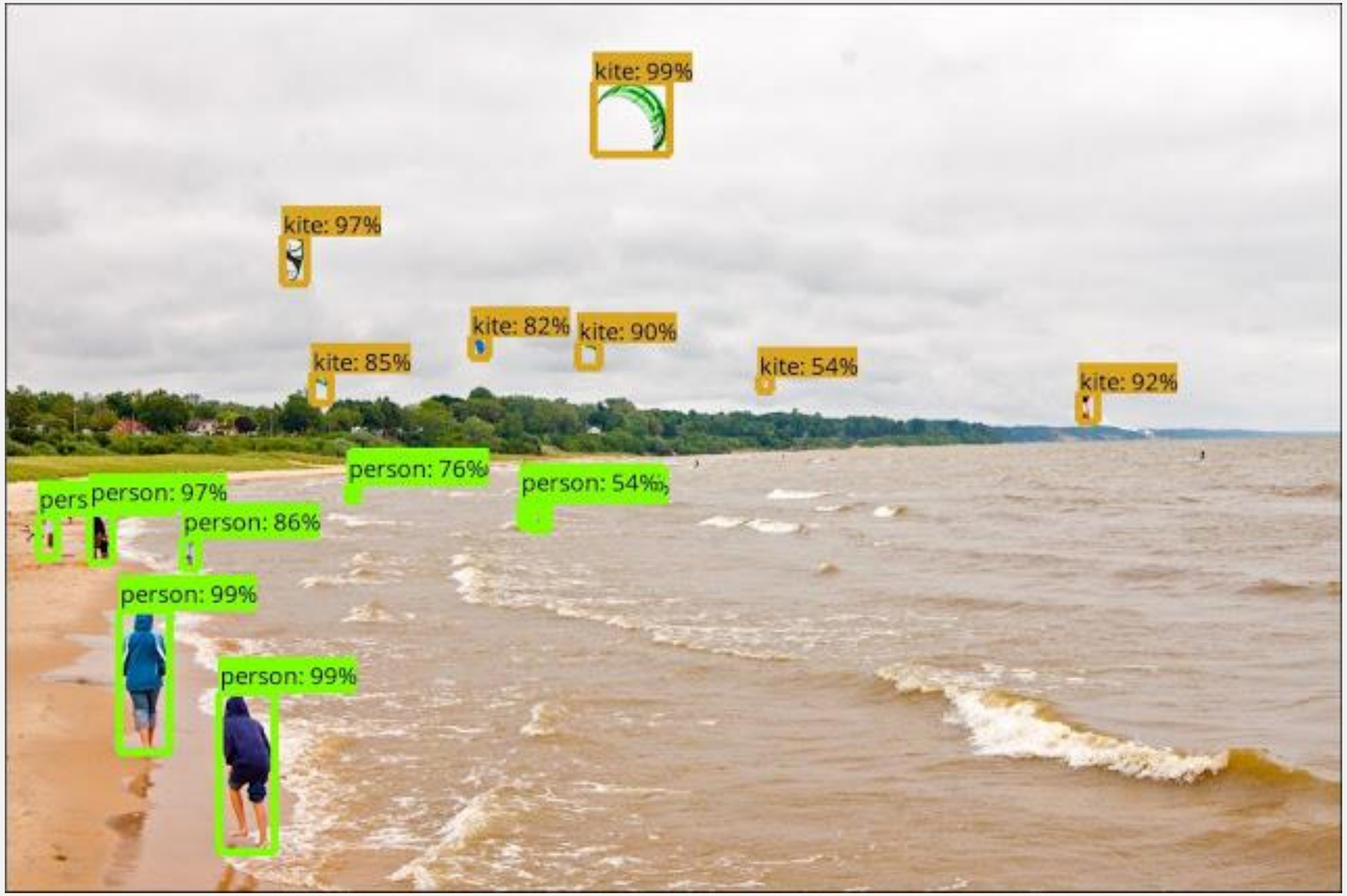
HAMIM

Object Detection Problem

- Different sizes
- Variable number of objects
- Different color and texture



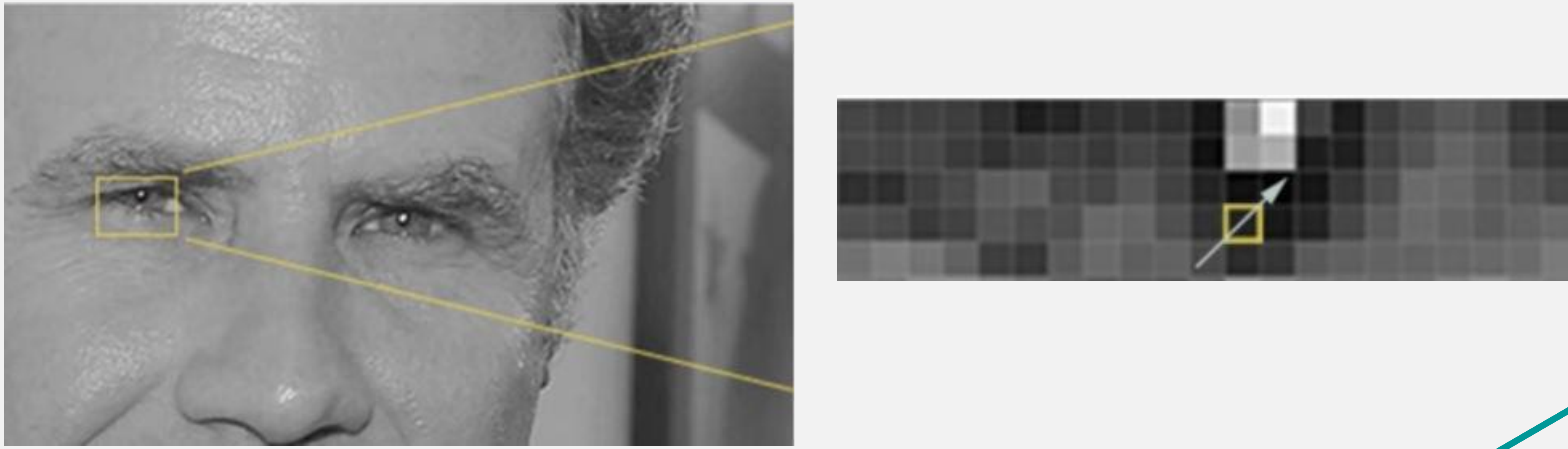
Object Detection Problem

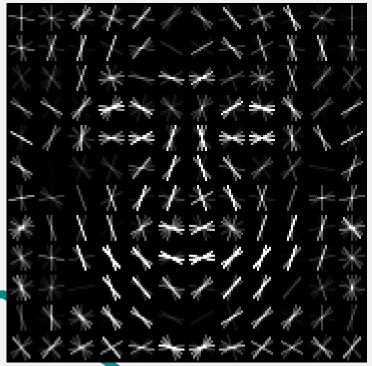


HAMIM

Traditional Approaches

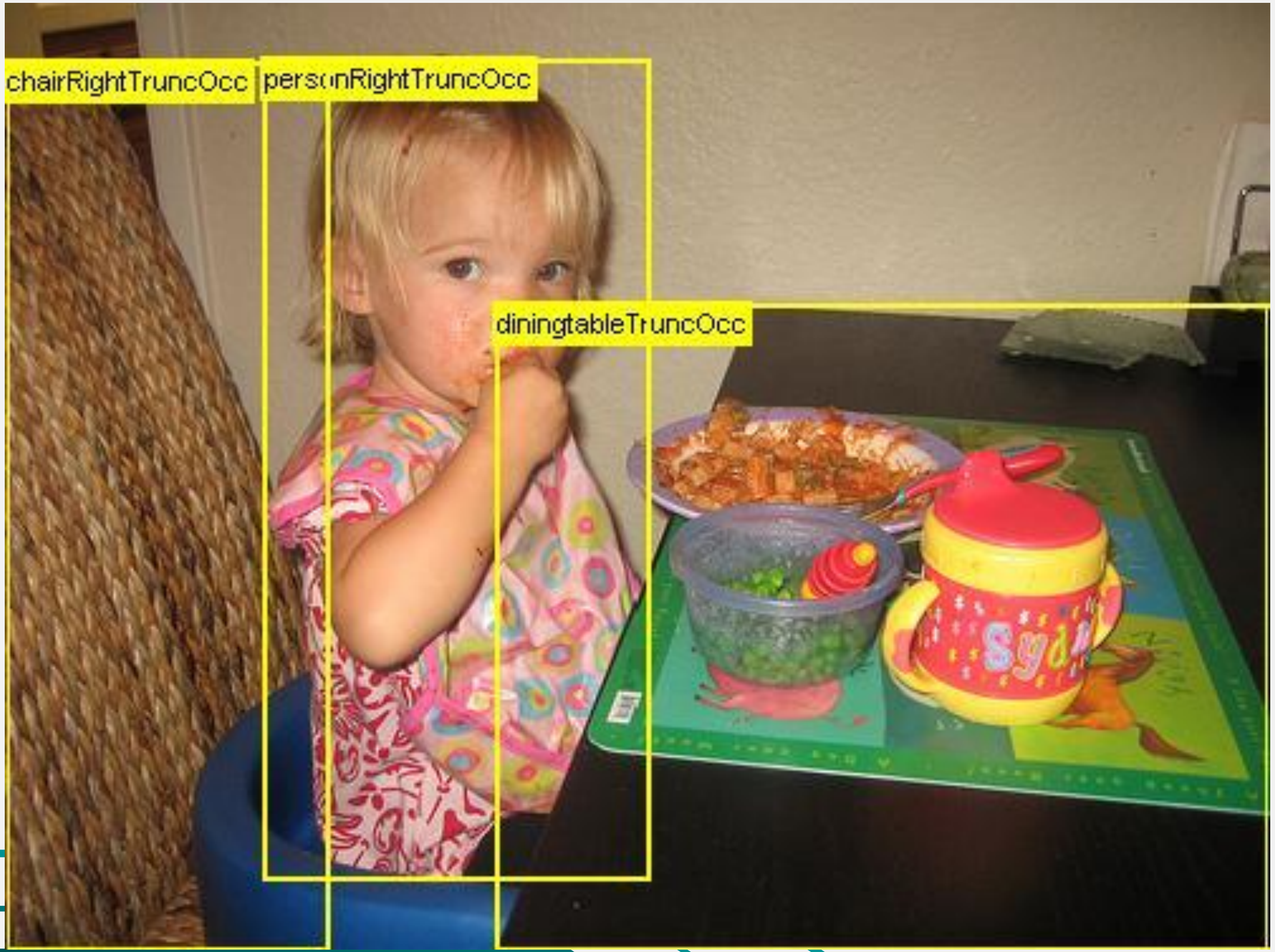
- HOG
- HAAR
- SIFT
- SURF
- …

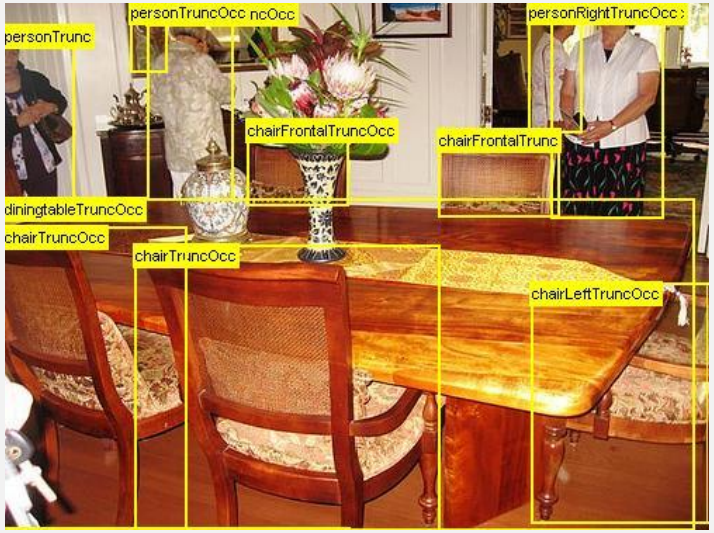


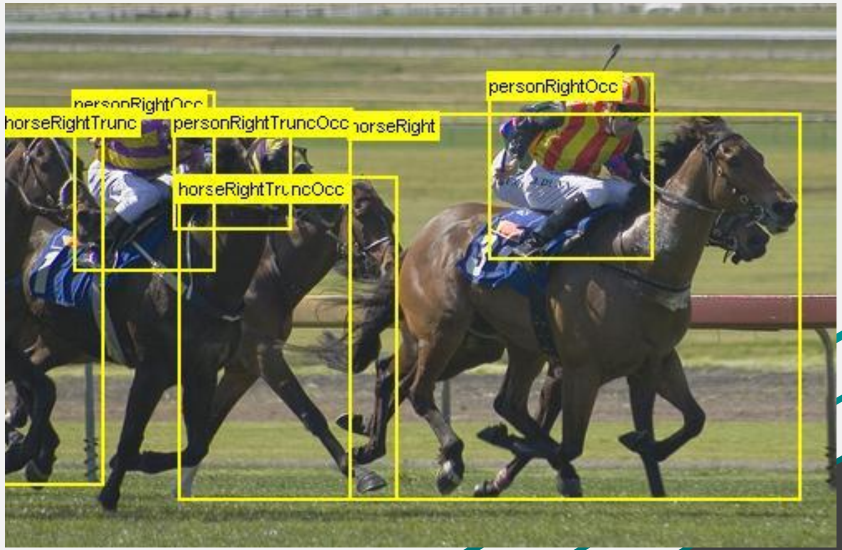


Dataset Samples

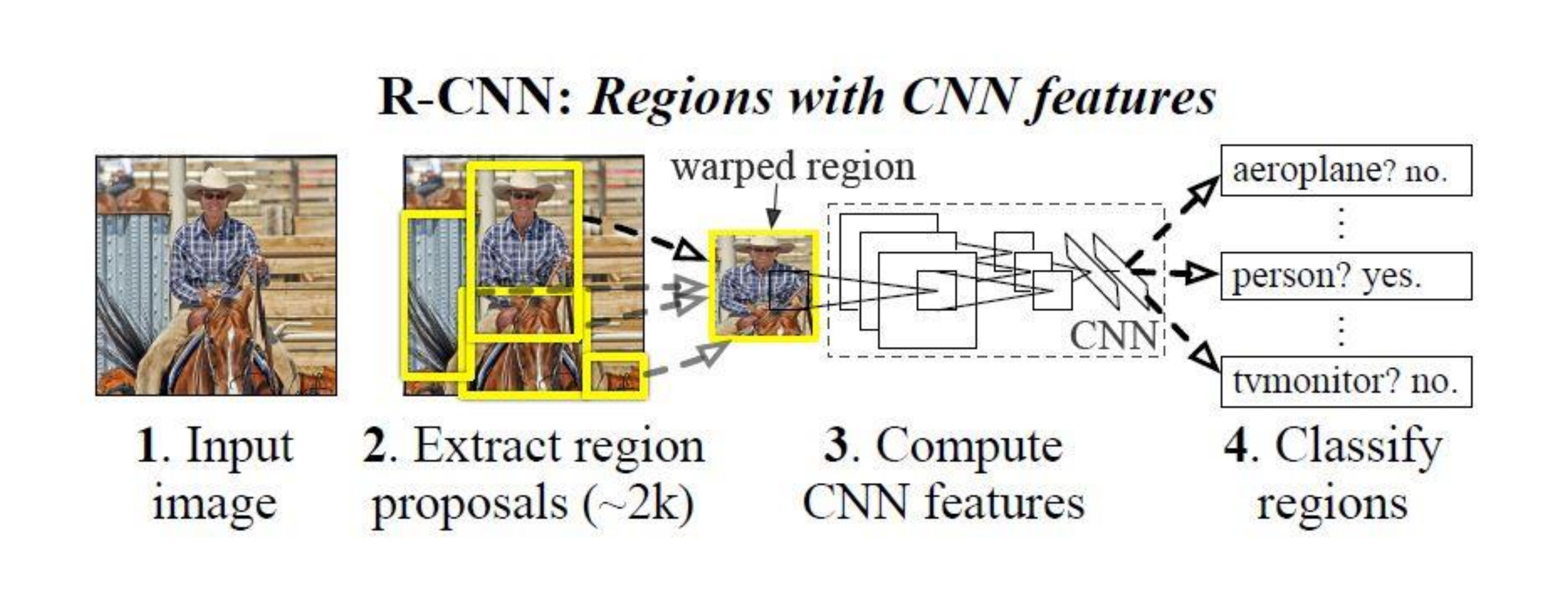
- Pascal voc 2012



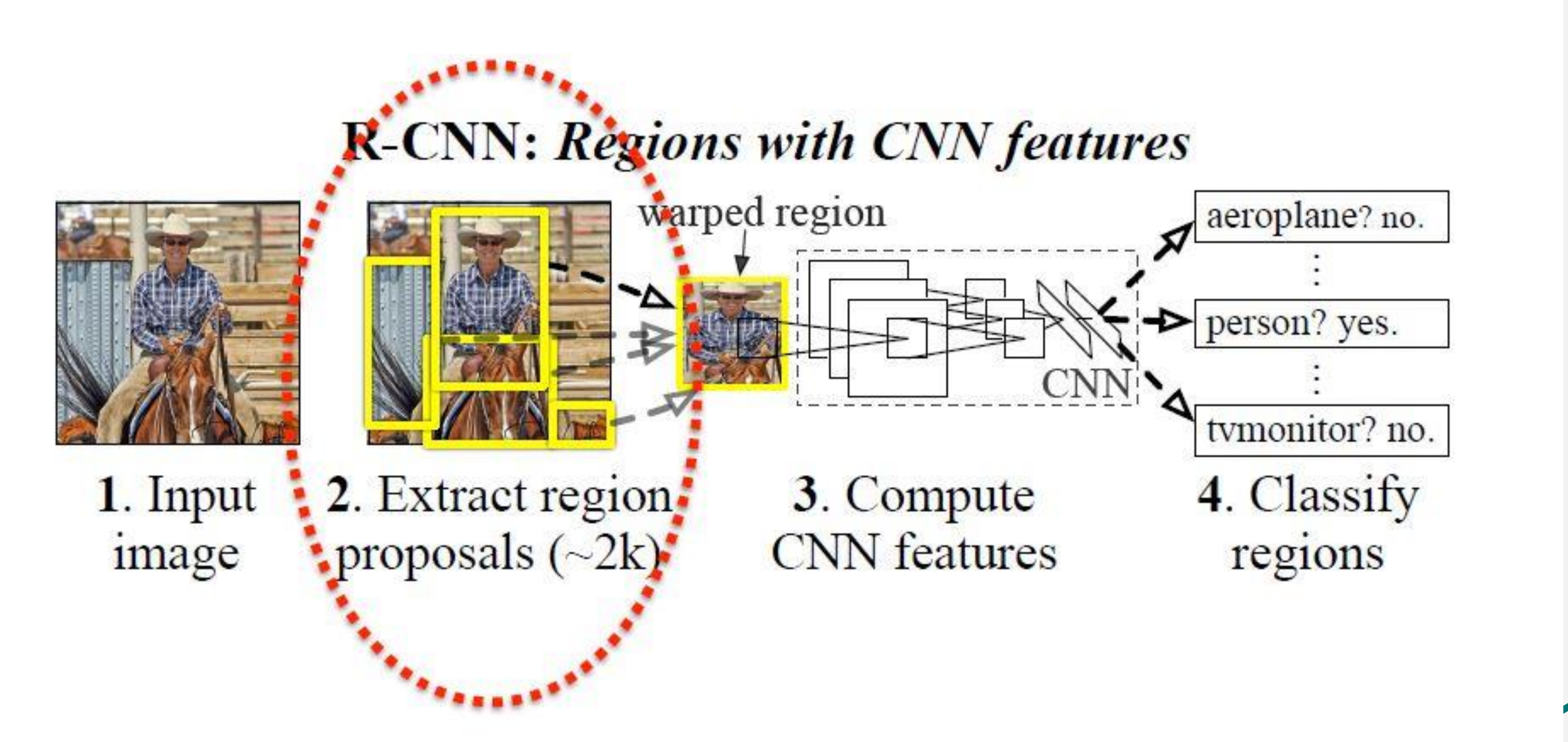




Region-based Convolutional Neural Network

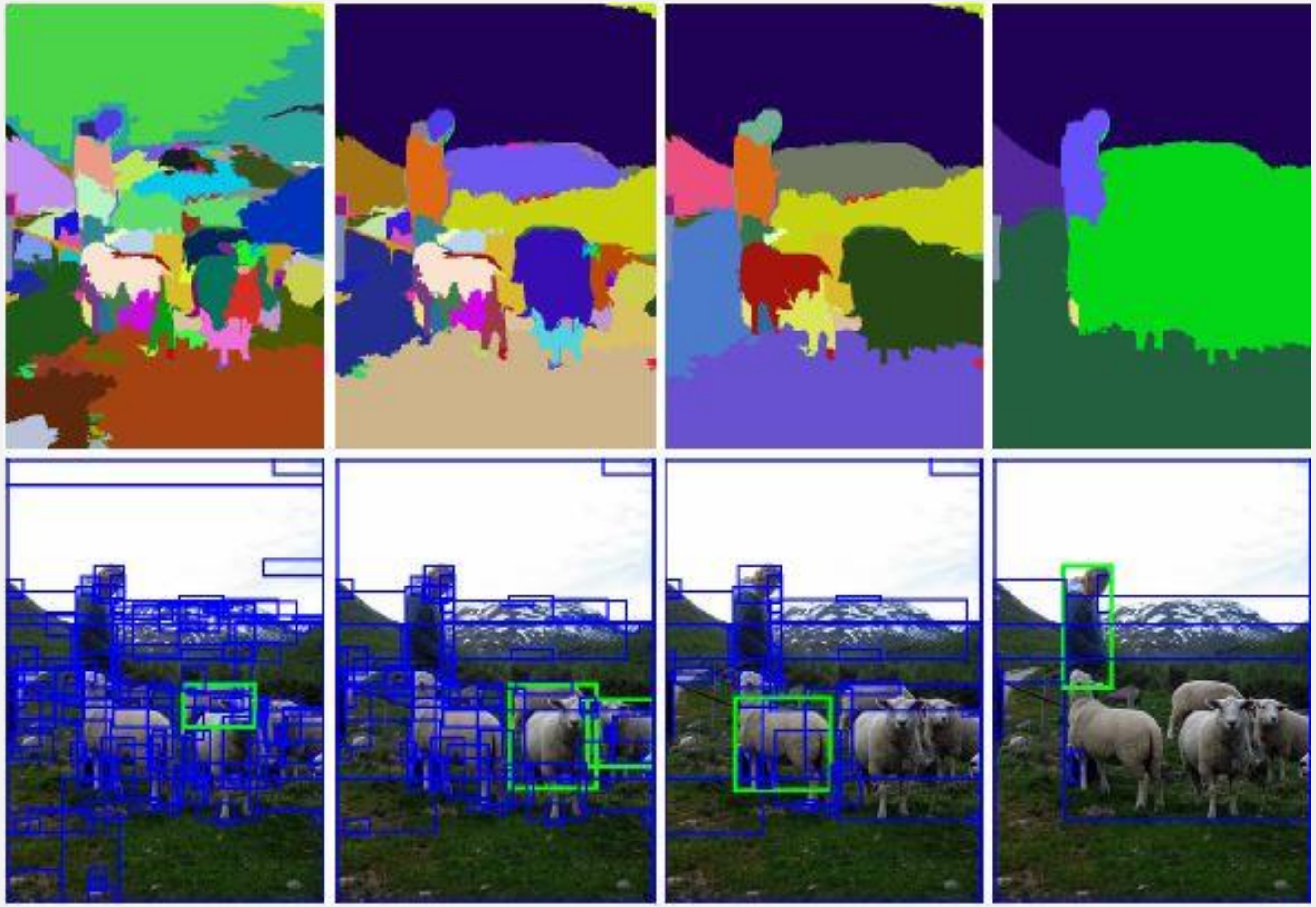


Region proposal extraction

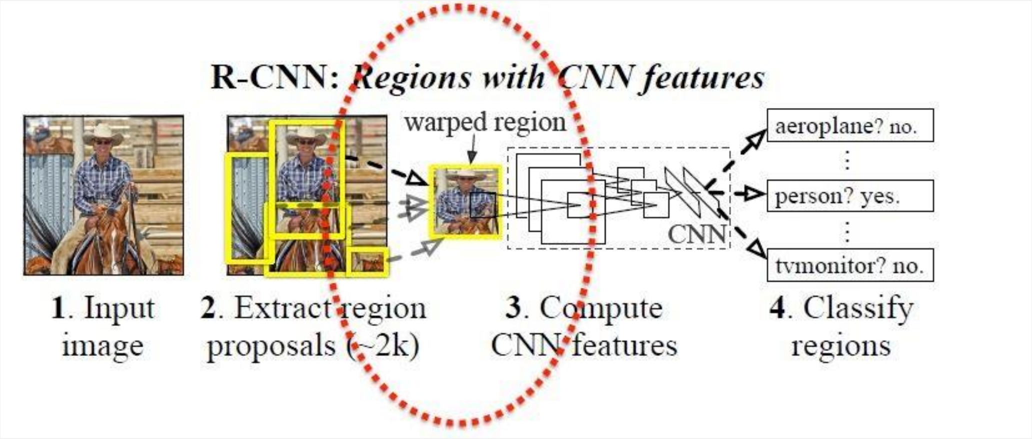


HAMIM

Selective Search Method

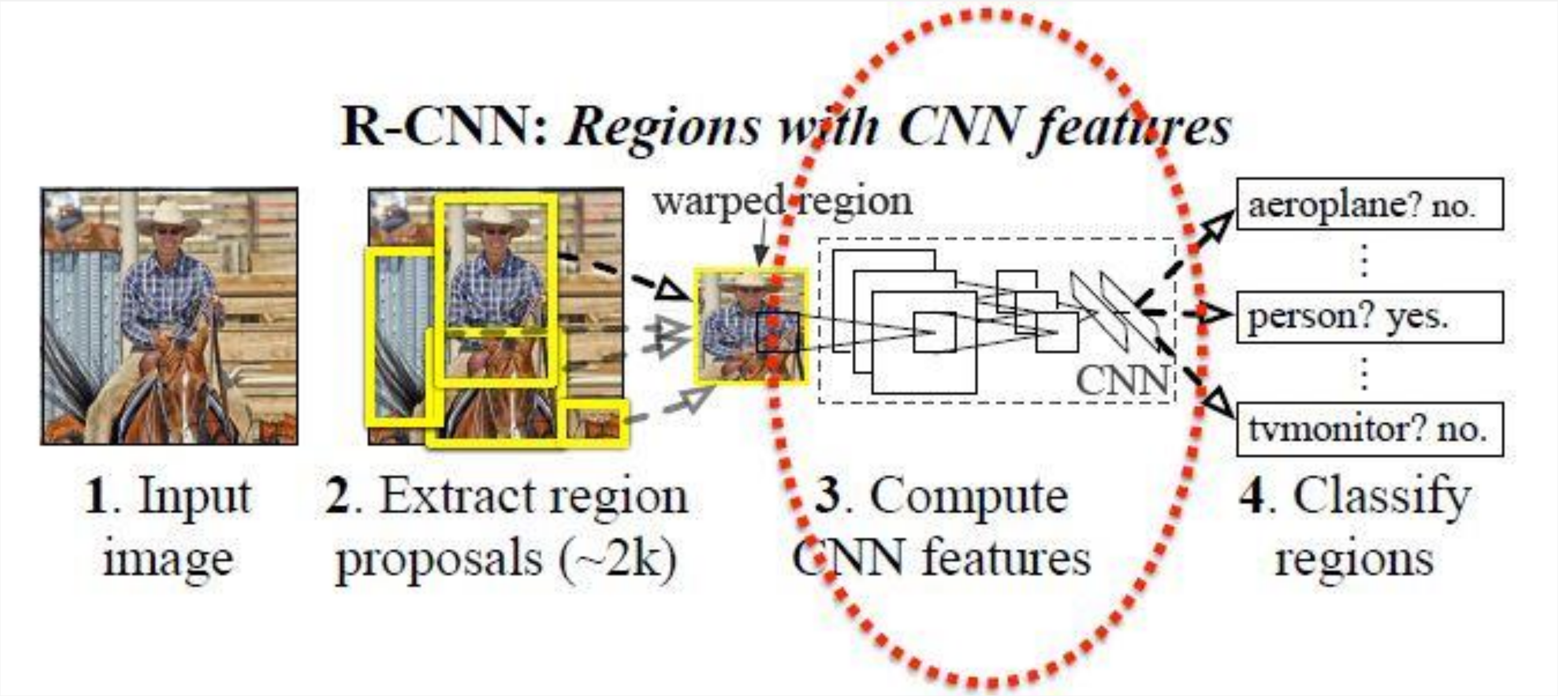


Region Warping

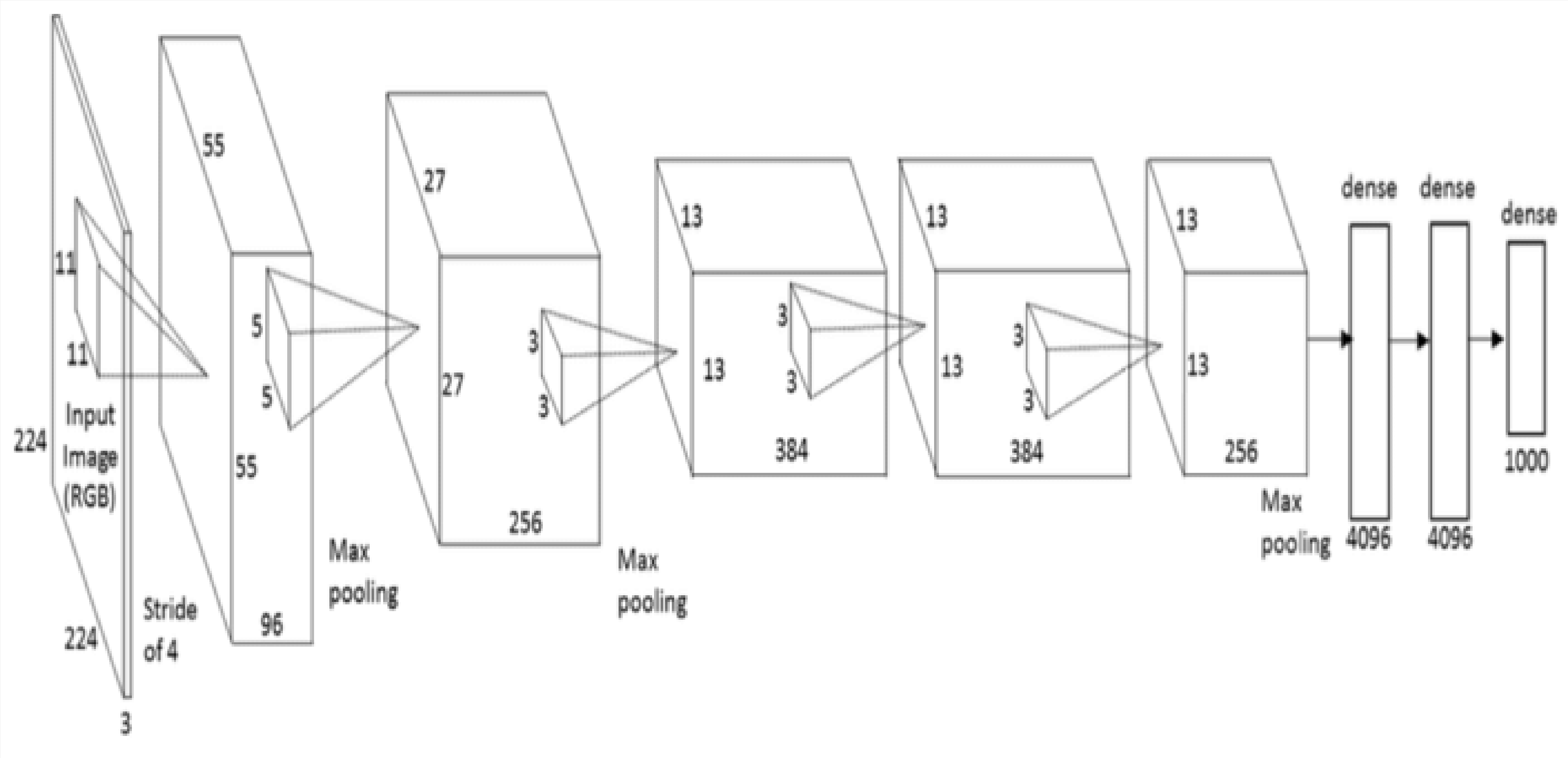


Warp to
224*224 Patch
=
4096 feature
vector

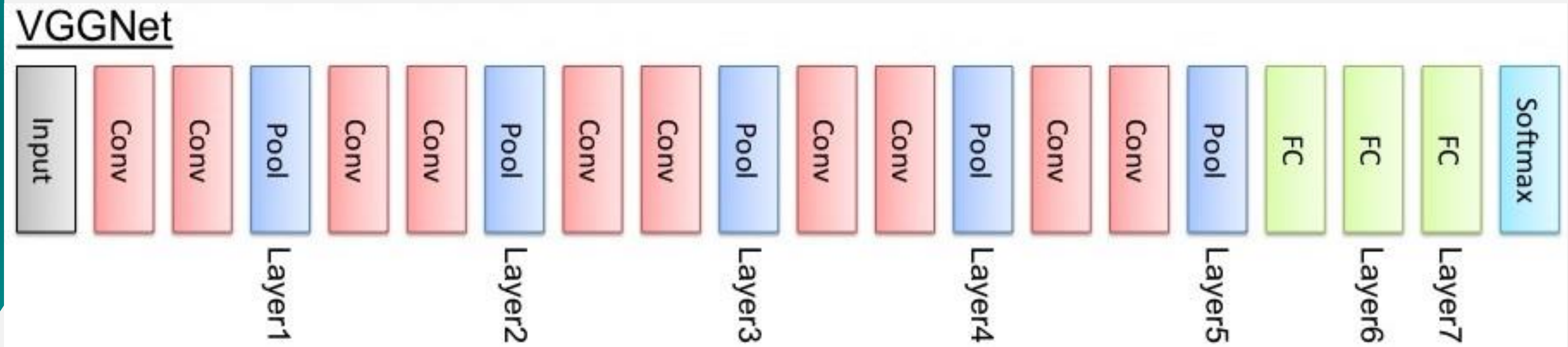
Feature Extraction using CNNs



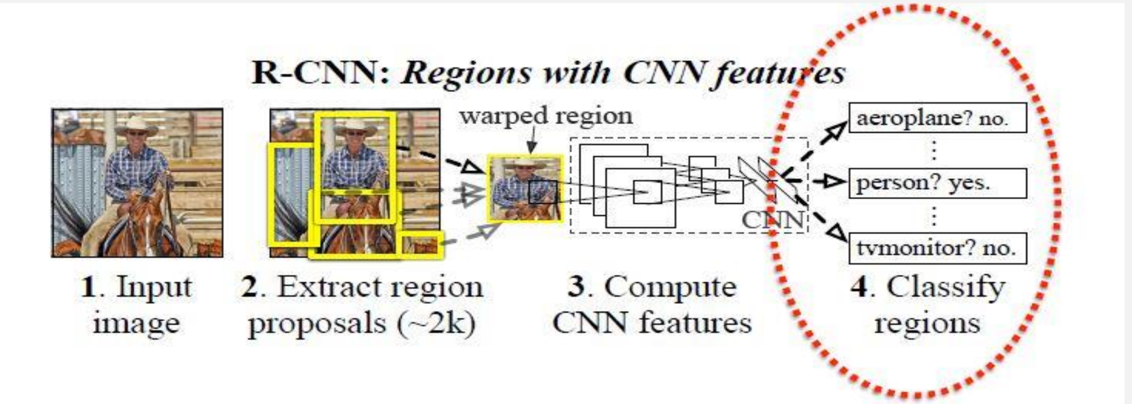
Feature Extractor : AlexNet

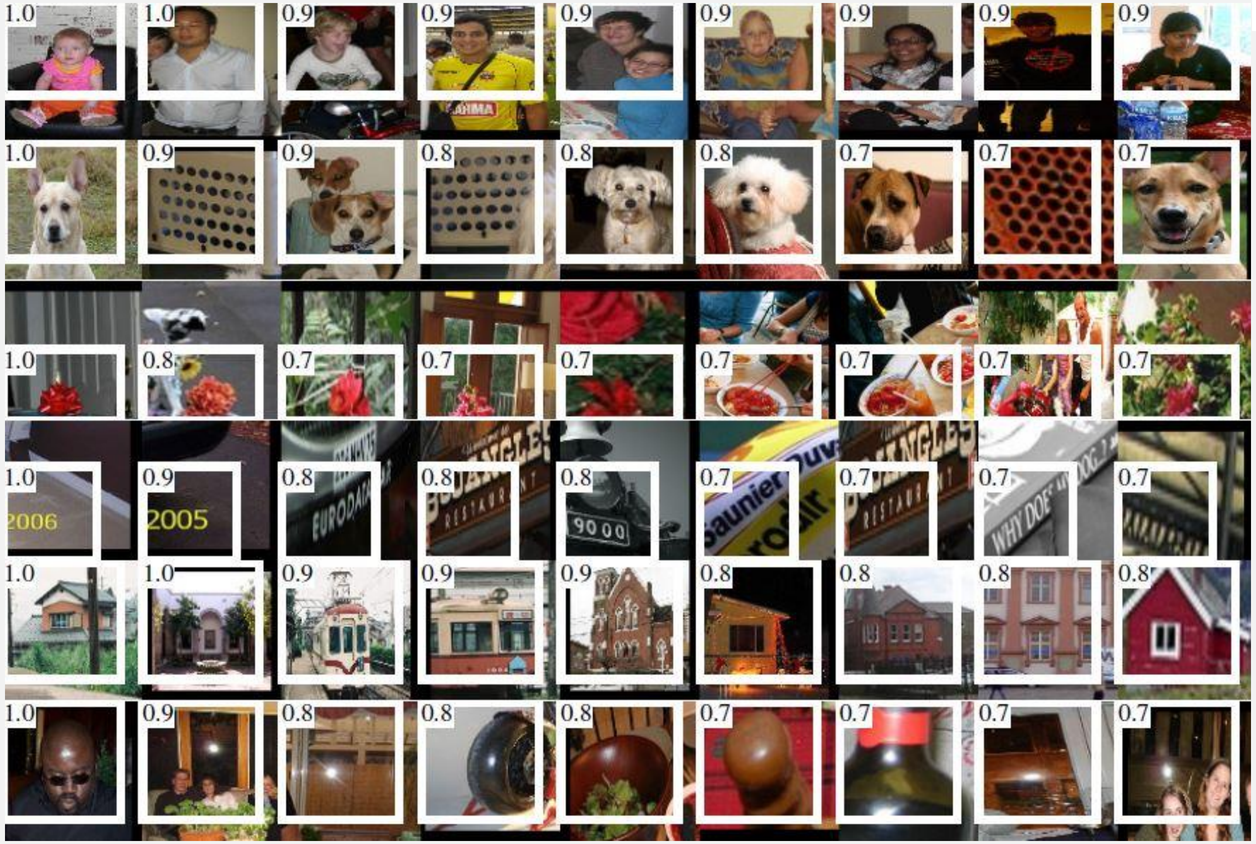


Feature Extractor : VGG Net



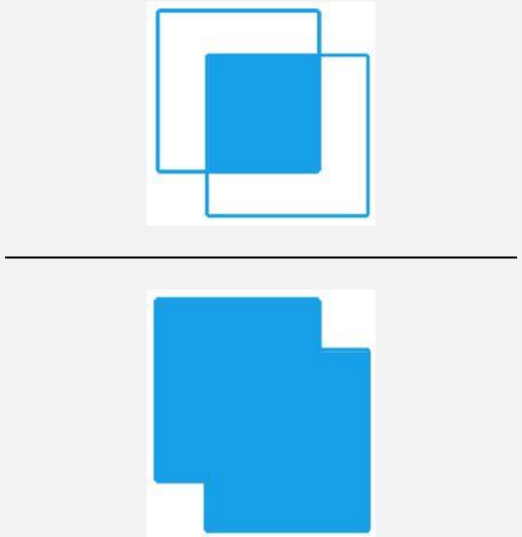
Classification Using SVM

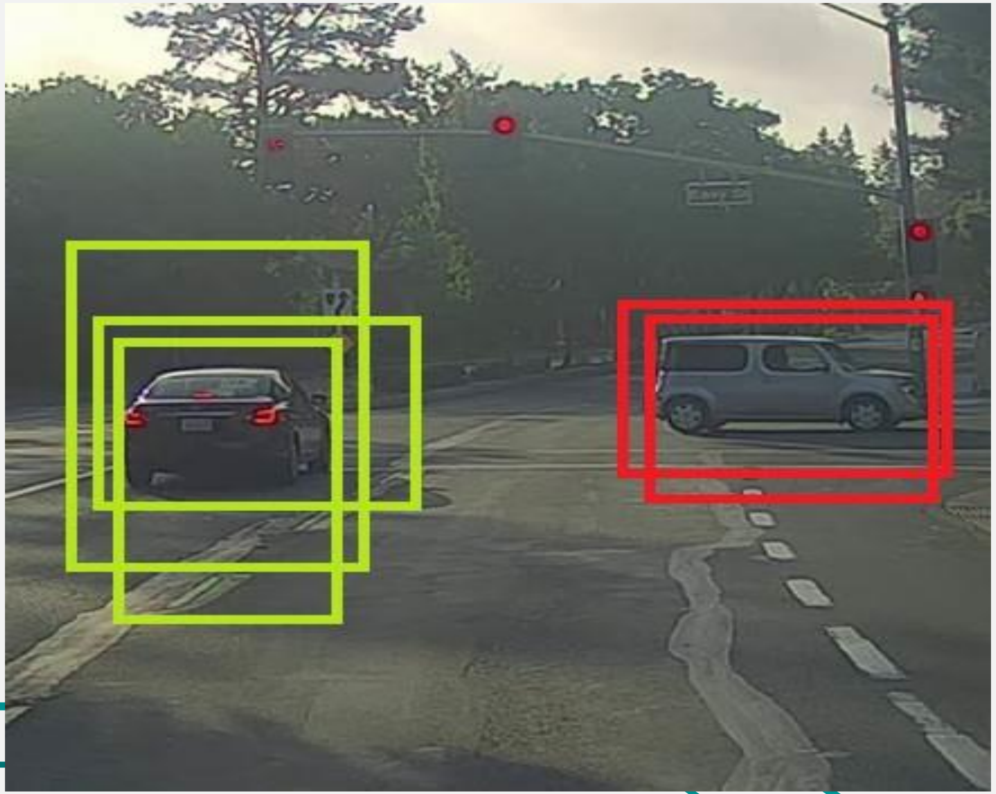


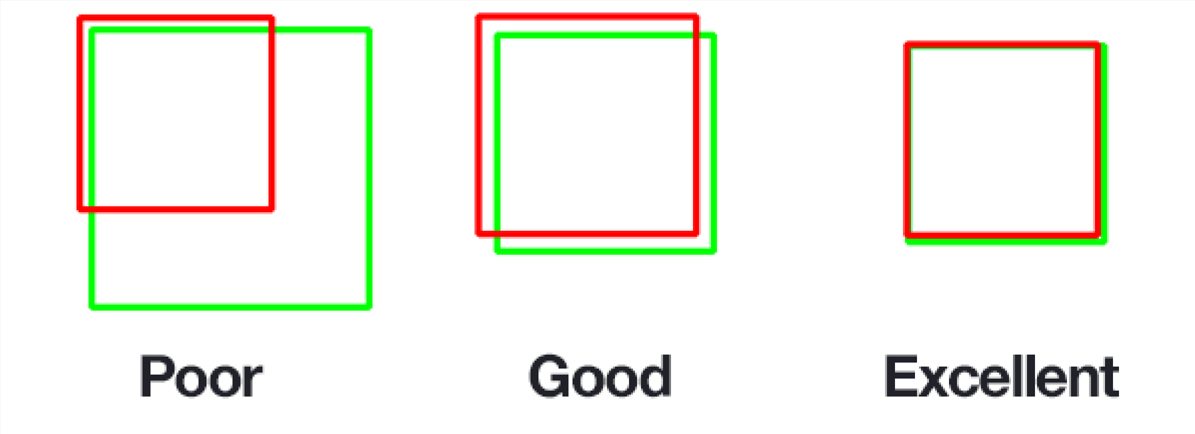


Intersection Over Union (IoU)

- $IOU=\frac{\text{Area of Overlap}}{\text{Area of Union}}$

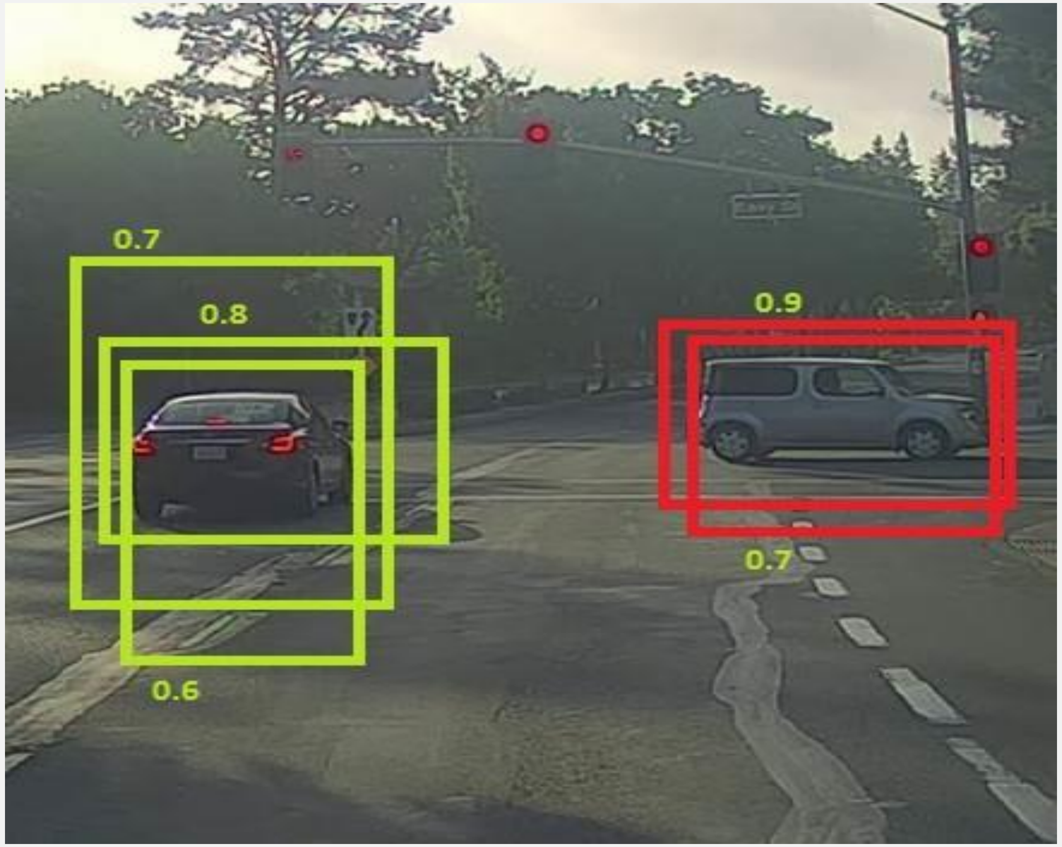


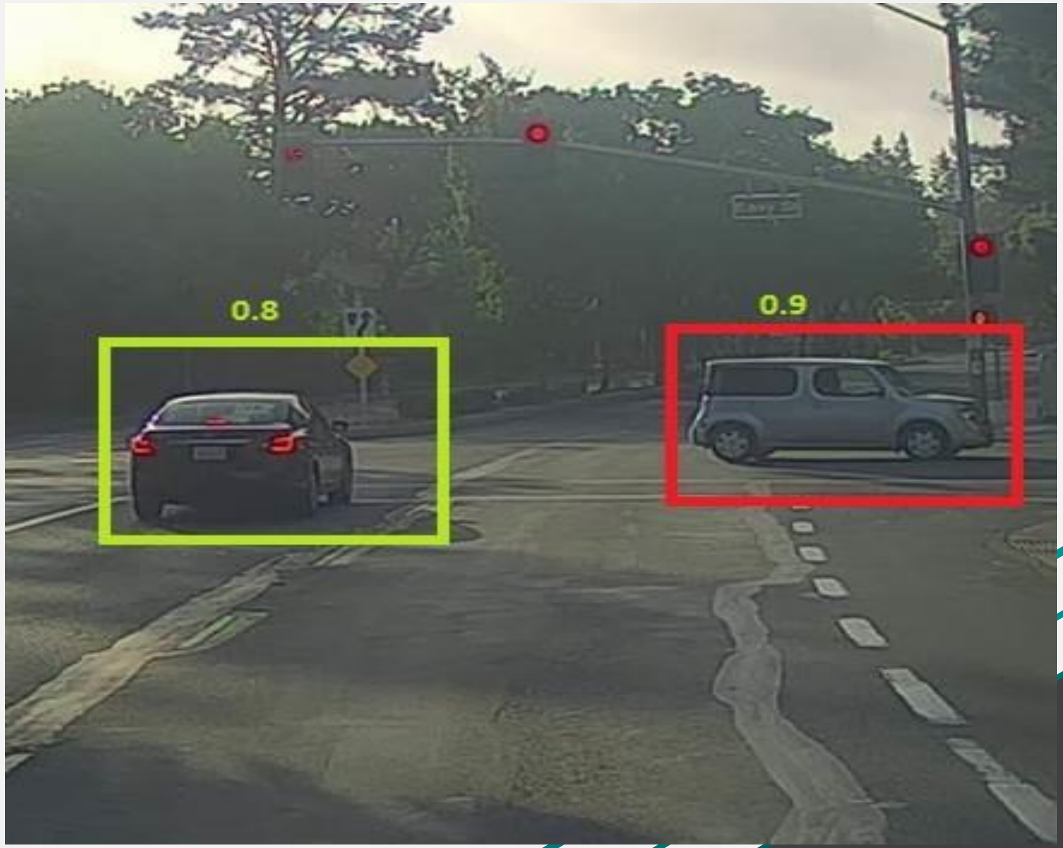




Non-max Suppression

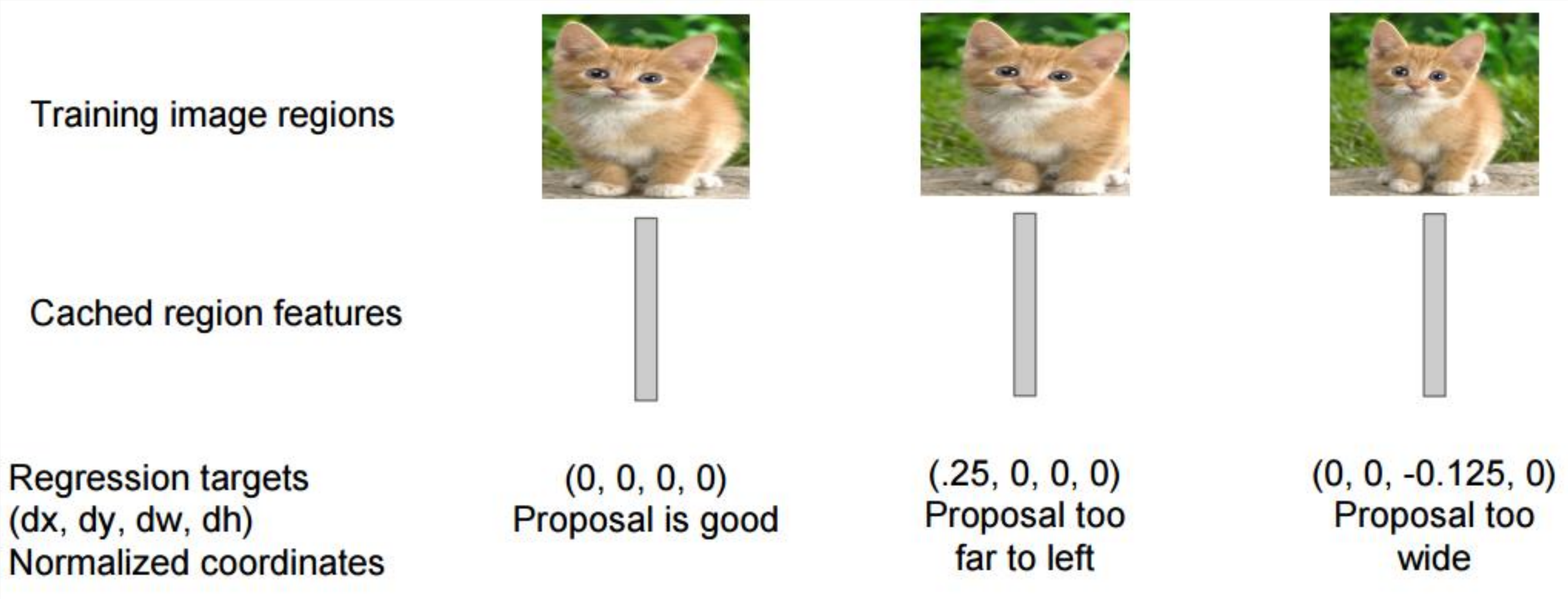
- Rejects a region if it has IOU overlap with a higher scoring selected region



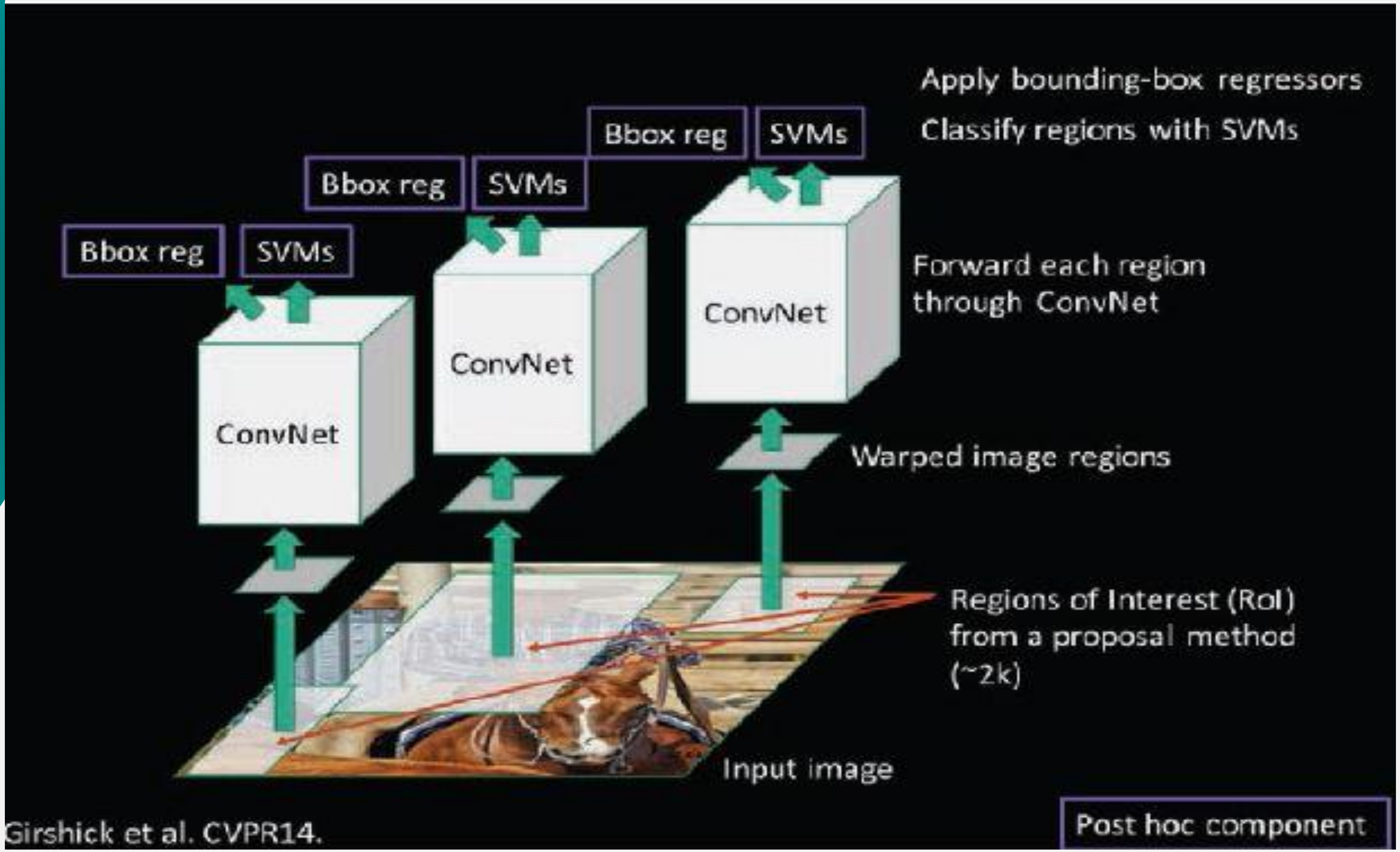


HAMIM©

Localize Object using Regression



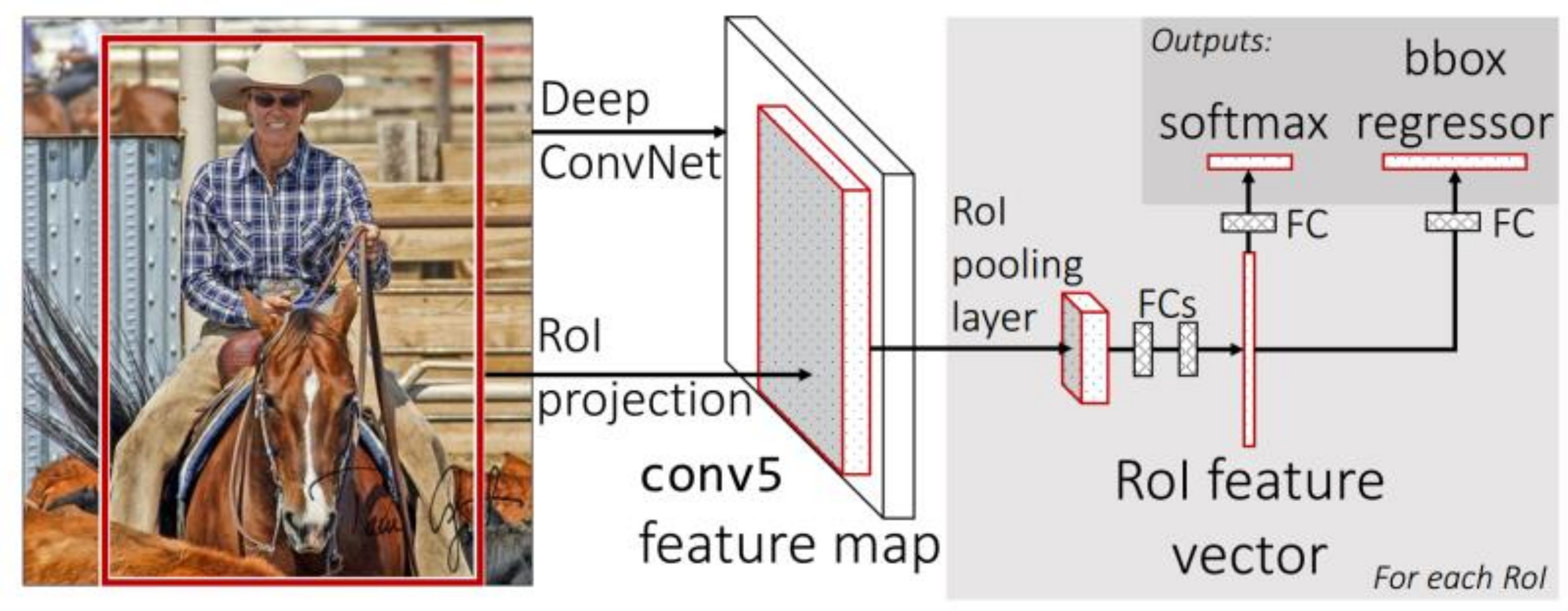
R-CNN



Limitation:
3 stage : CNN, SVM, Regression

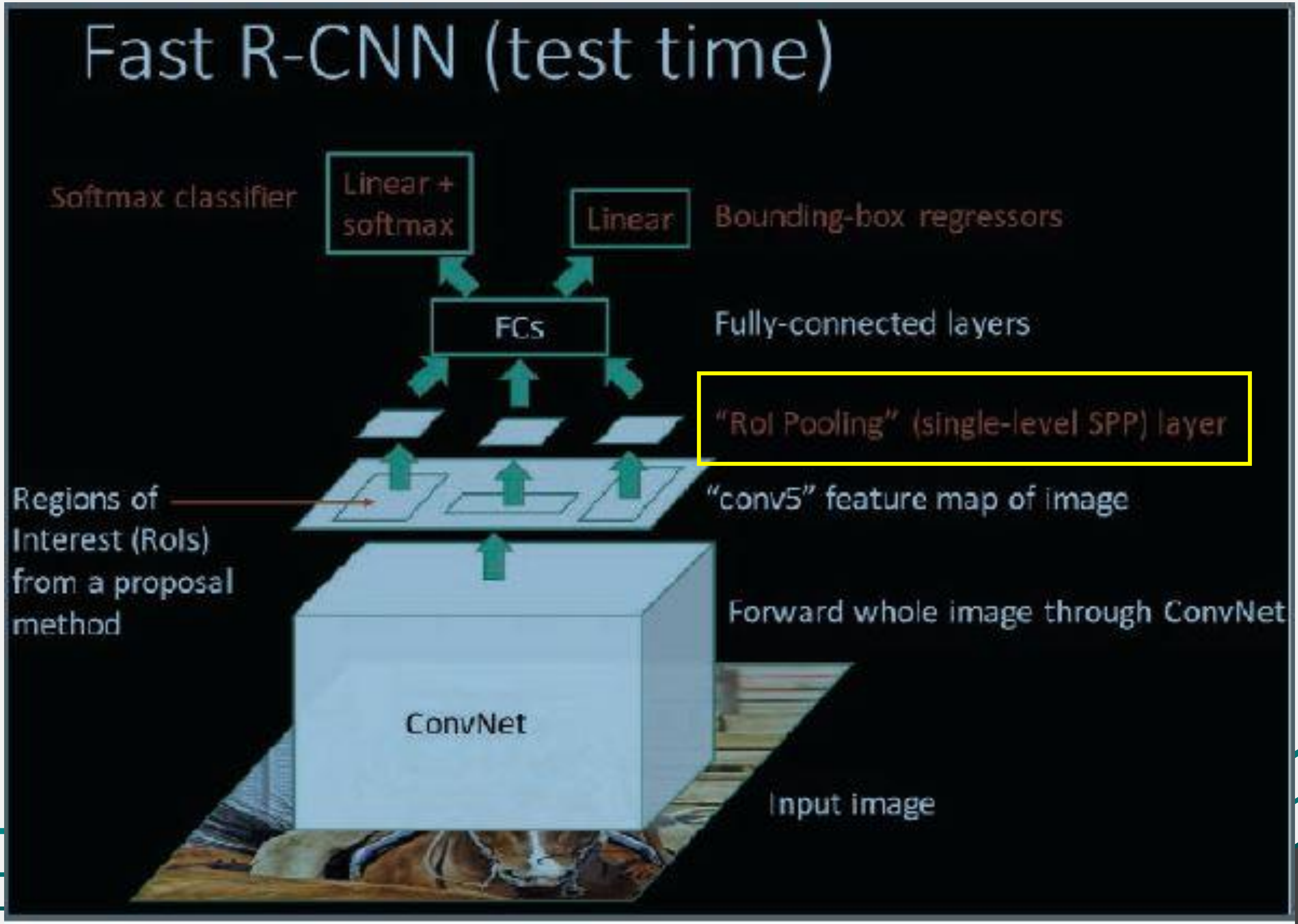
HAMIM

Fast R-CNN



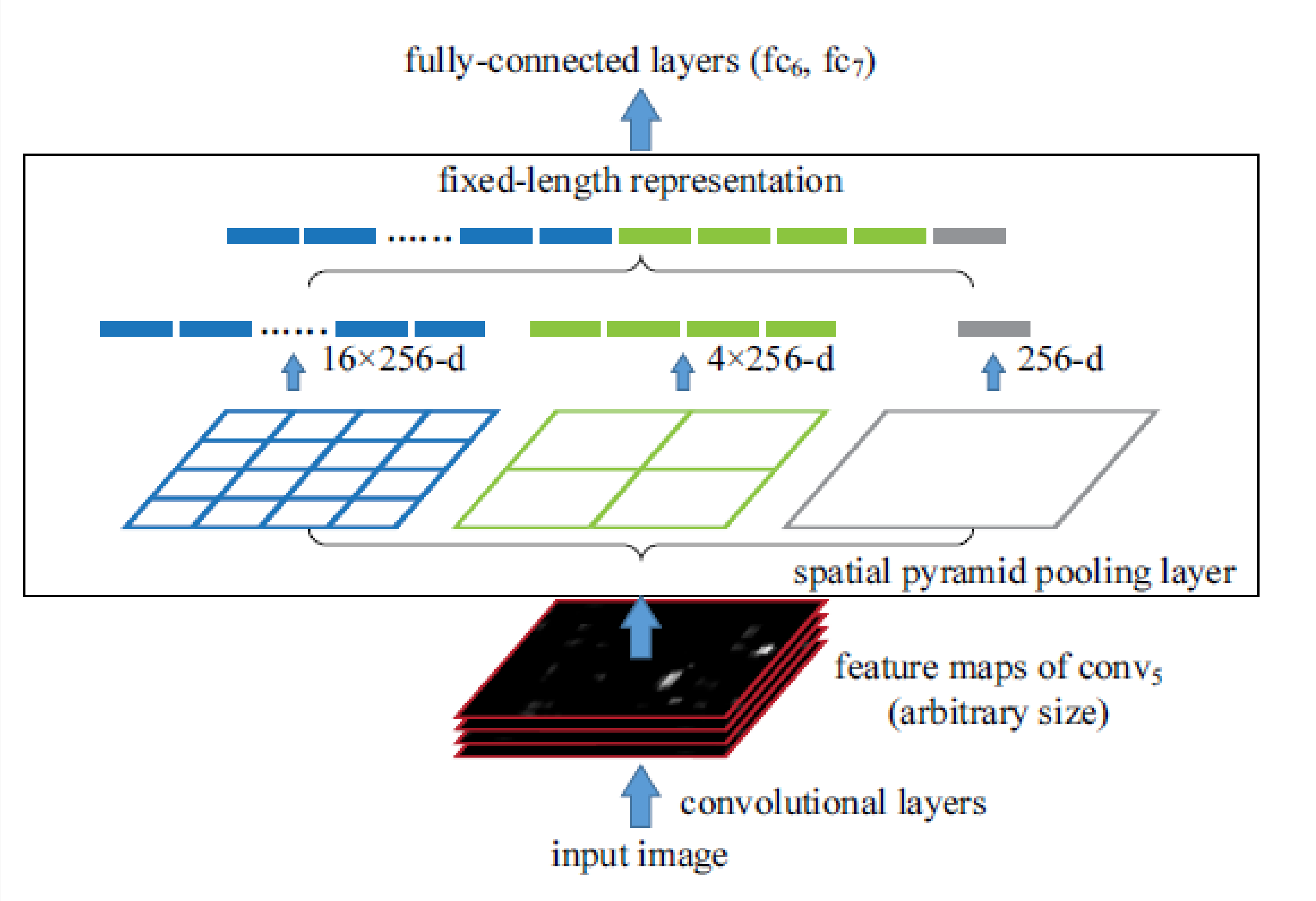
HAMIM

Fast R-CNN



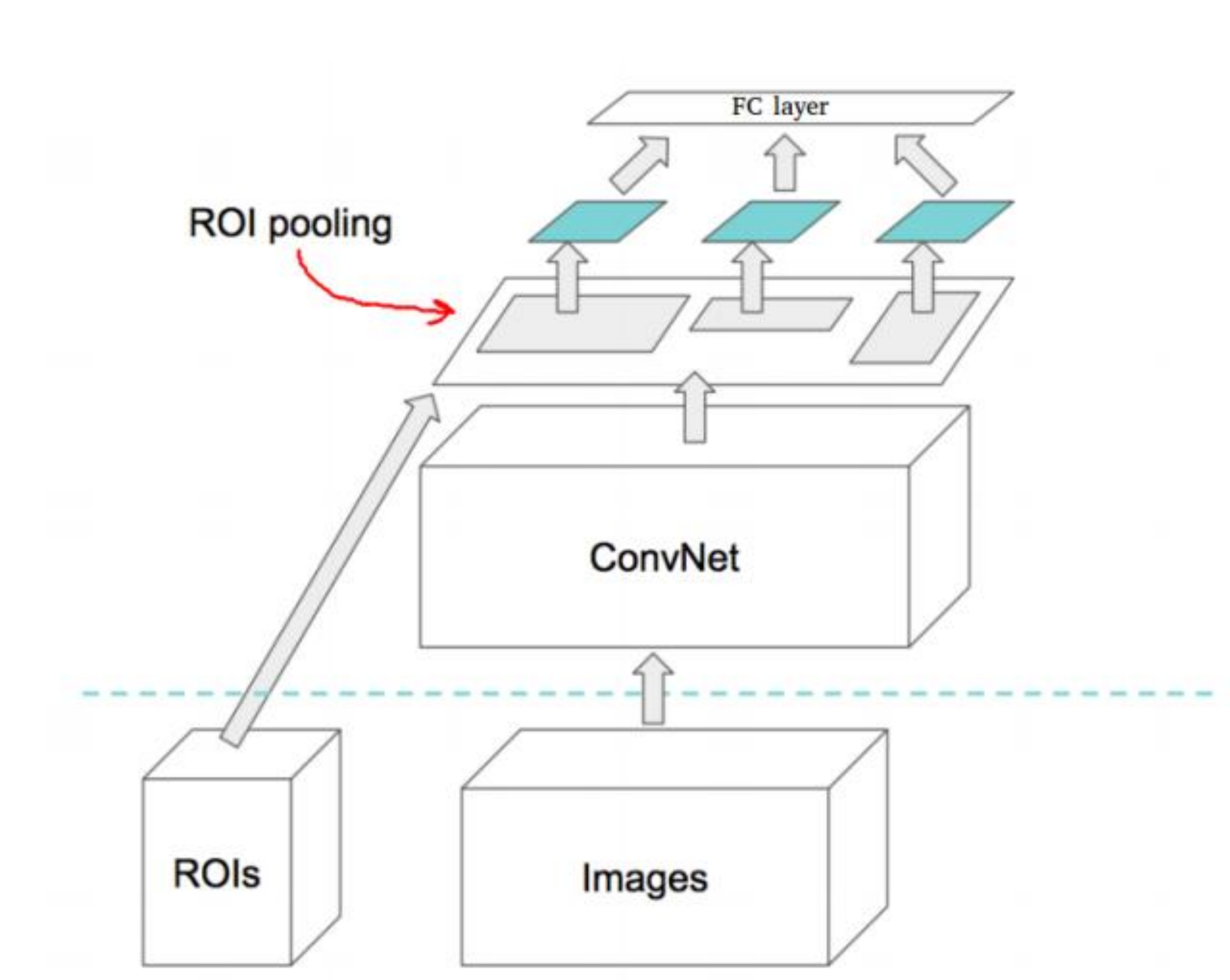
Fast R-CNN (test time)
Softmax classifier
Linear + softmax
Linear
Bounding-box regressors
FCs
Fully-connected layers
"RoI Pooling" (single-level SPP) layer
Regions of Interest (RoIs) from a proposal method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image

ROI Pooling- Spatial Pyramid Polling



HAMIM

Fast R-CNN



HAMIM

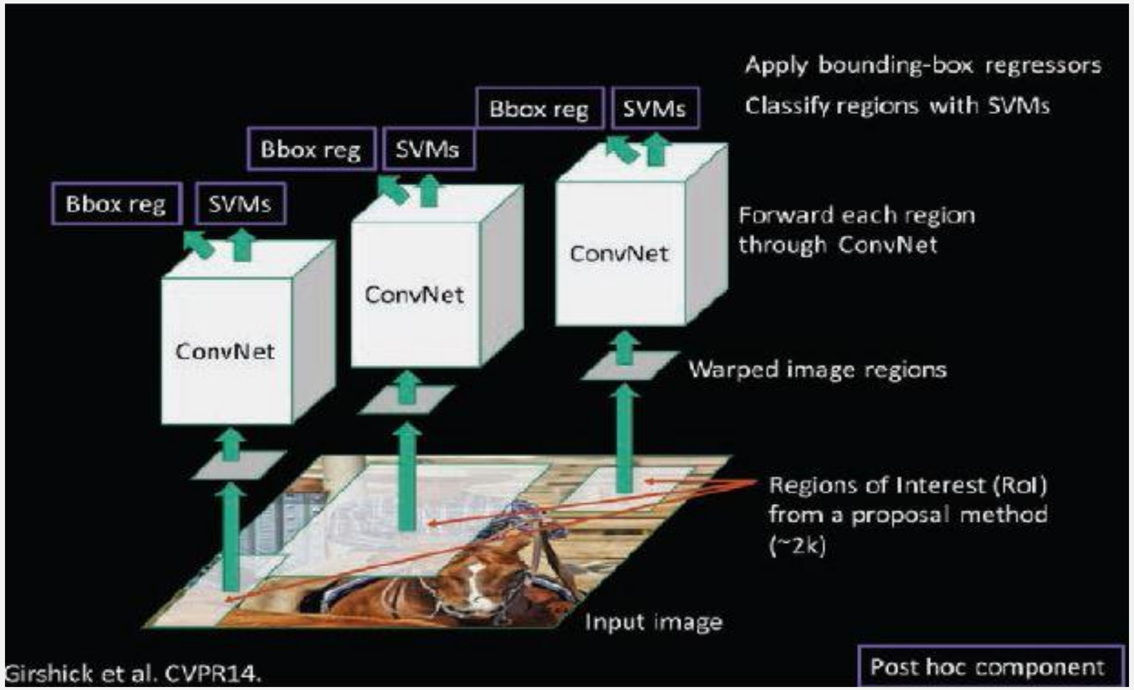
Fast R-CNN and R-CNN

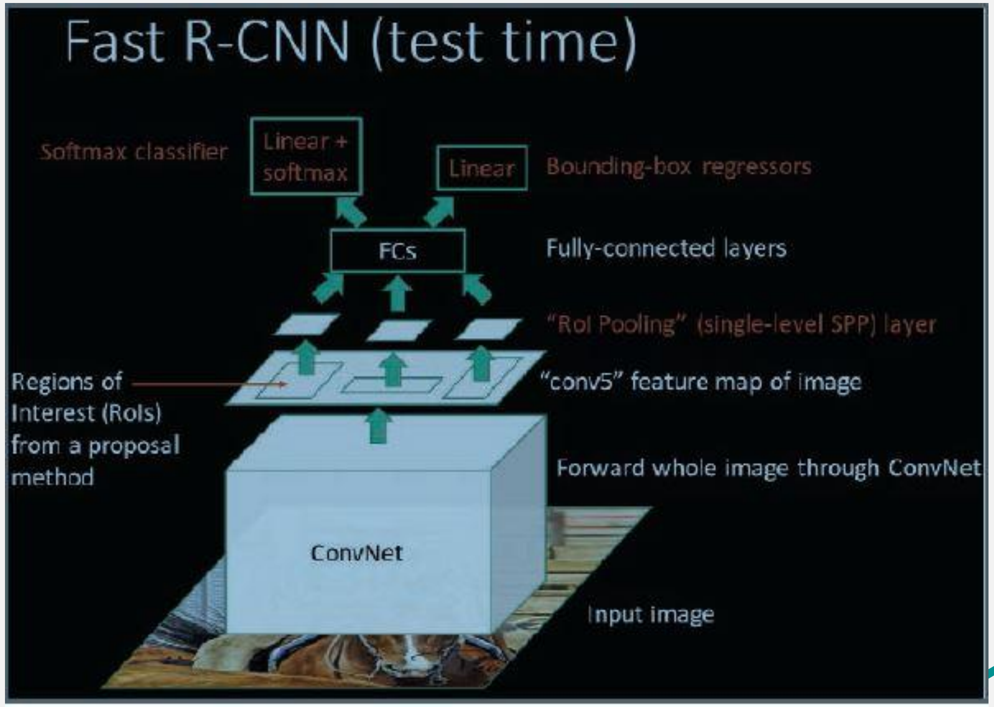
	R-CNN	Fast R-CNN
Test time per image	47 seconds	**0.32 seconds**
(Speedup)	1x	**146x**
Test time per image with Selective Search	50 seconds	**2 seconds**
(Speedup)	1x	**25x**

bottleneck

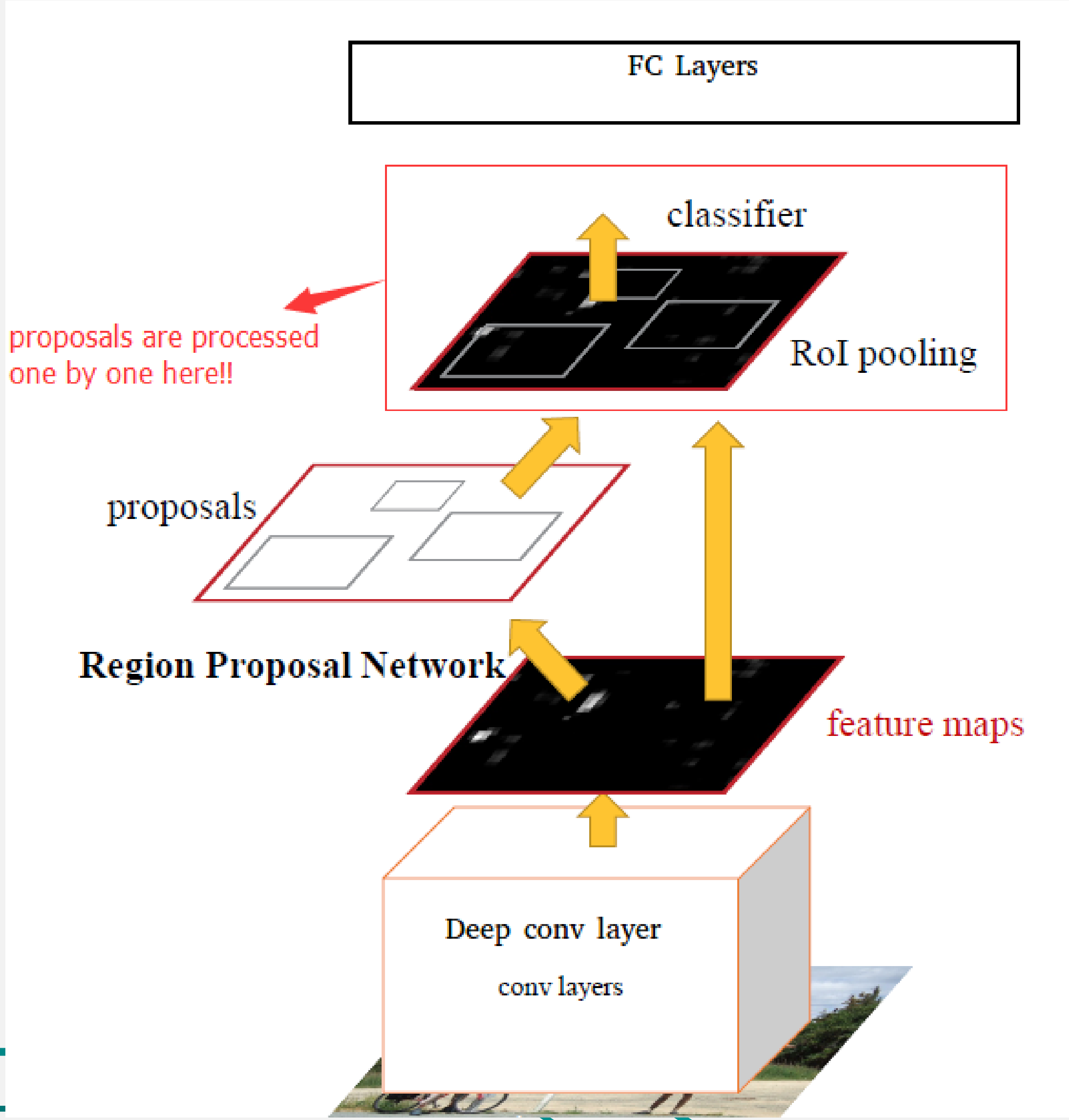
HAMIM©

R-CNN and Fast R-CNN



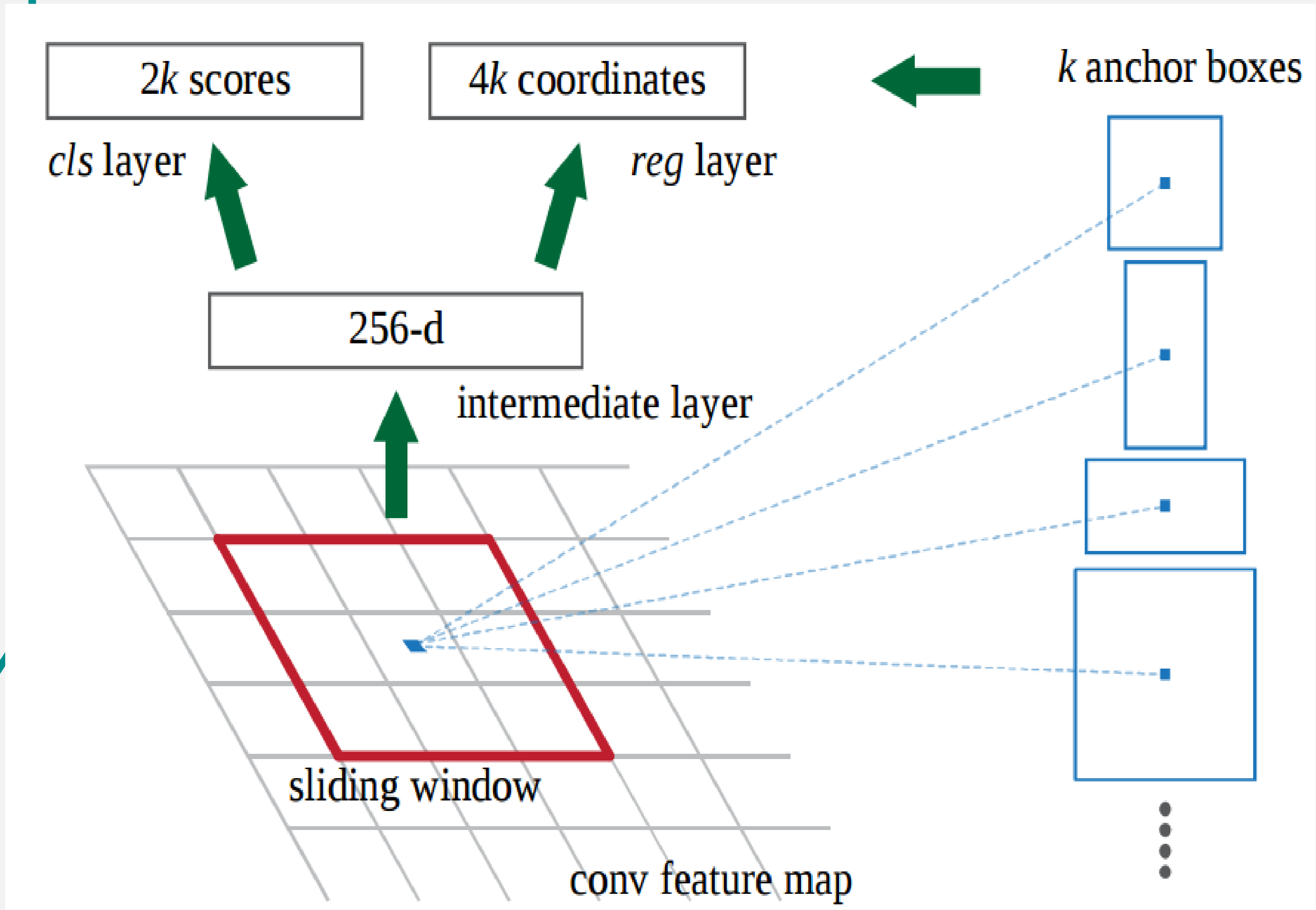


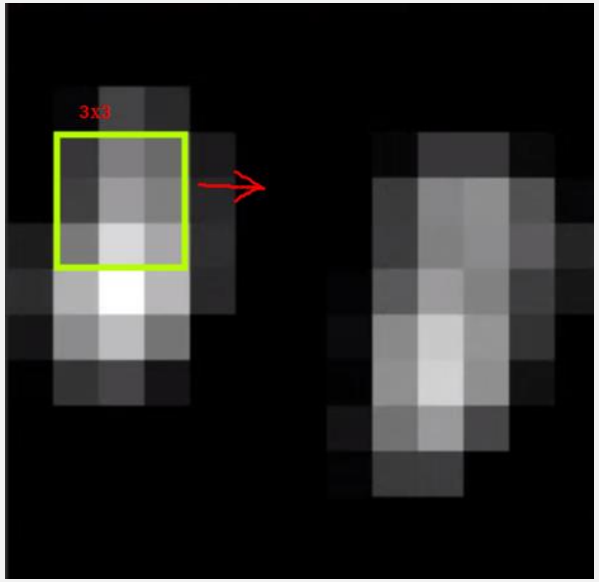
Faster R-CNN



The main idea is use the last (or deep) conv layers to infer region proposals.

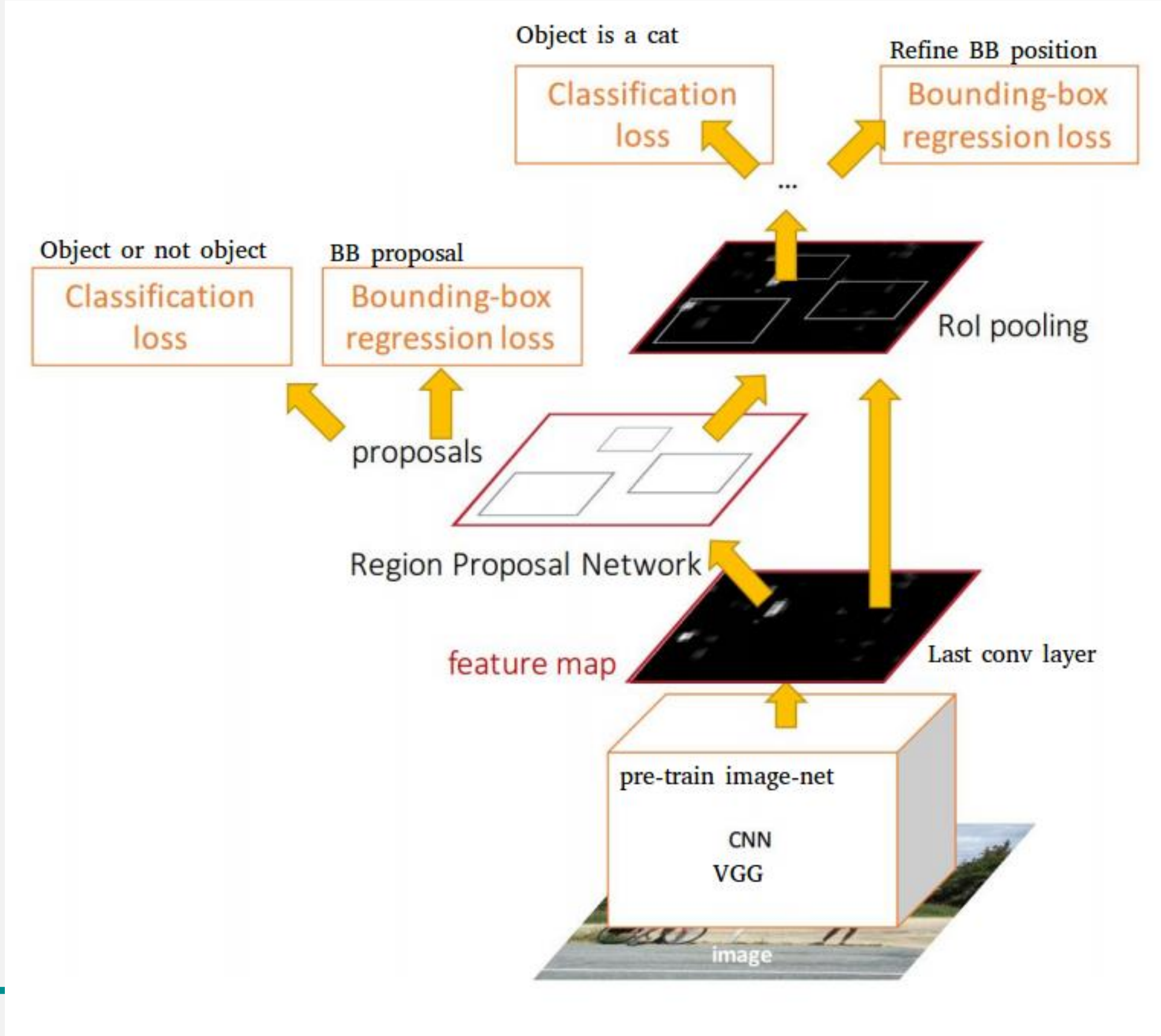
RPN network structure





HAMIM

Faster R-CNN Train



Faster R-CNN=RPN + Fast R-CNN

Faster RCNN Results

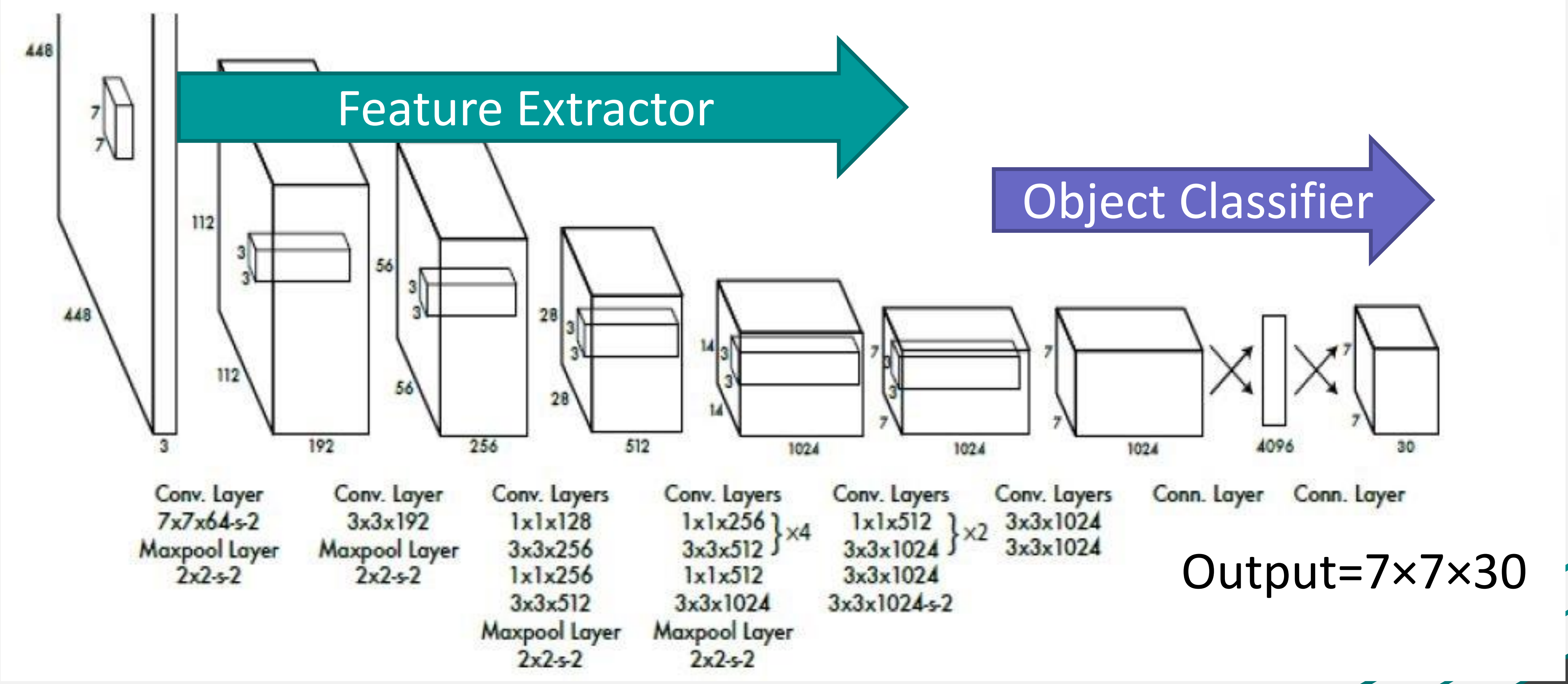
	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

$$\text{mAP}=\frac{1}{|classes|}\sum_{c\in classes}\frac{\#TP(c)}{\#TP(c)+\#FP(c)}$$

HAMIM©

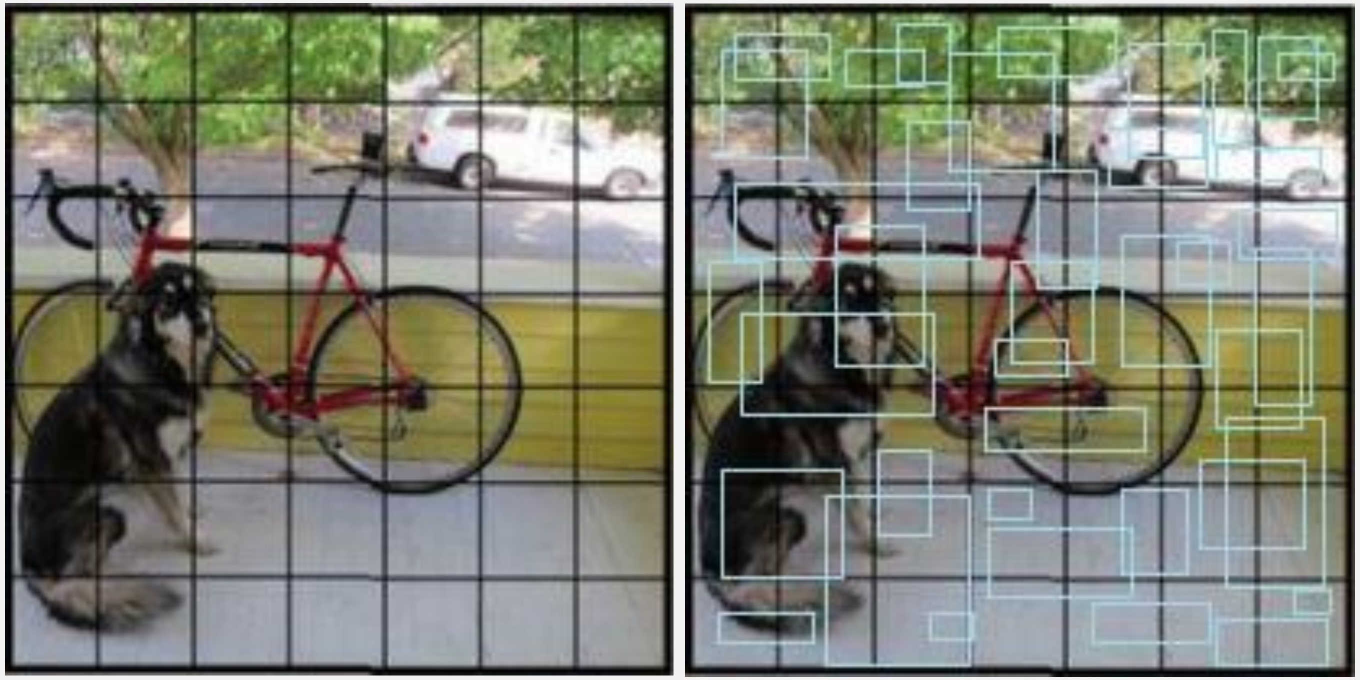
Yolo Demo

YOLO : You Only Look Once



Train on voc dataset : 20 different classes

YOLO- Bounding Box Concept



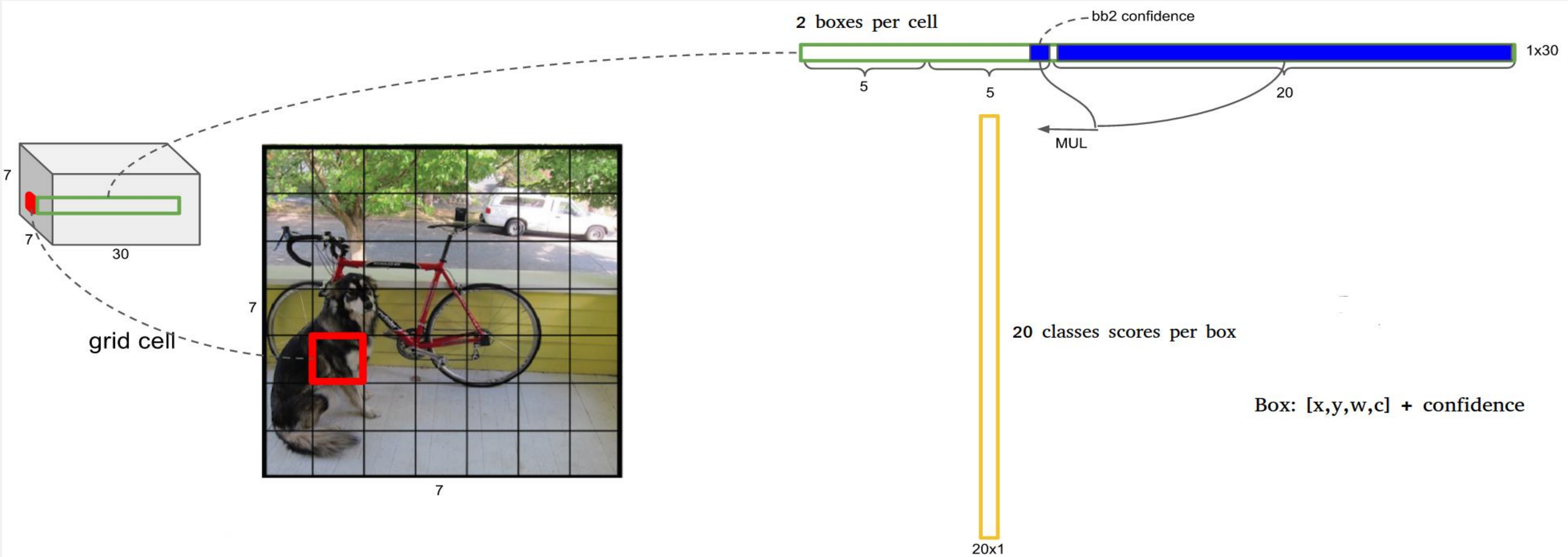
Confidence score: reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts.(for each bounding box)

2 Box definitions: (consisting of: x, y, width ,height , "is object" confidence)
20 class probabilities (only considered if the "is object" confidence is high)

$$[x_1\ y_1\ w_1\ h_1\ \text{is\_object}_1\ x_2\ y_2\ w_2\ h_2\ \text{is\_object}_2\ C_1\ C_2\ C_3\ \ldots\ C_{20}\ ]$$

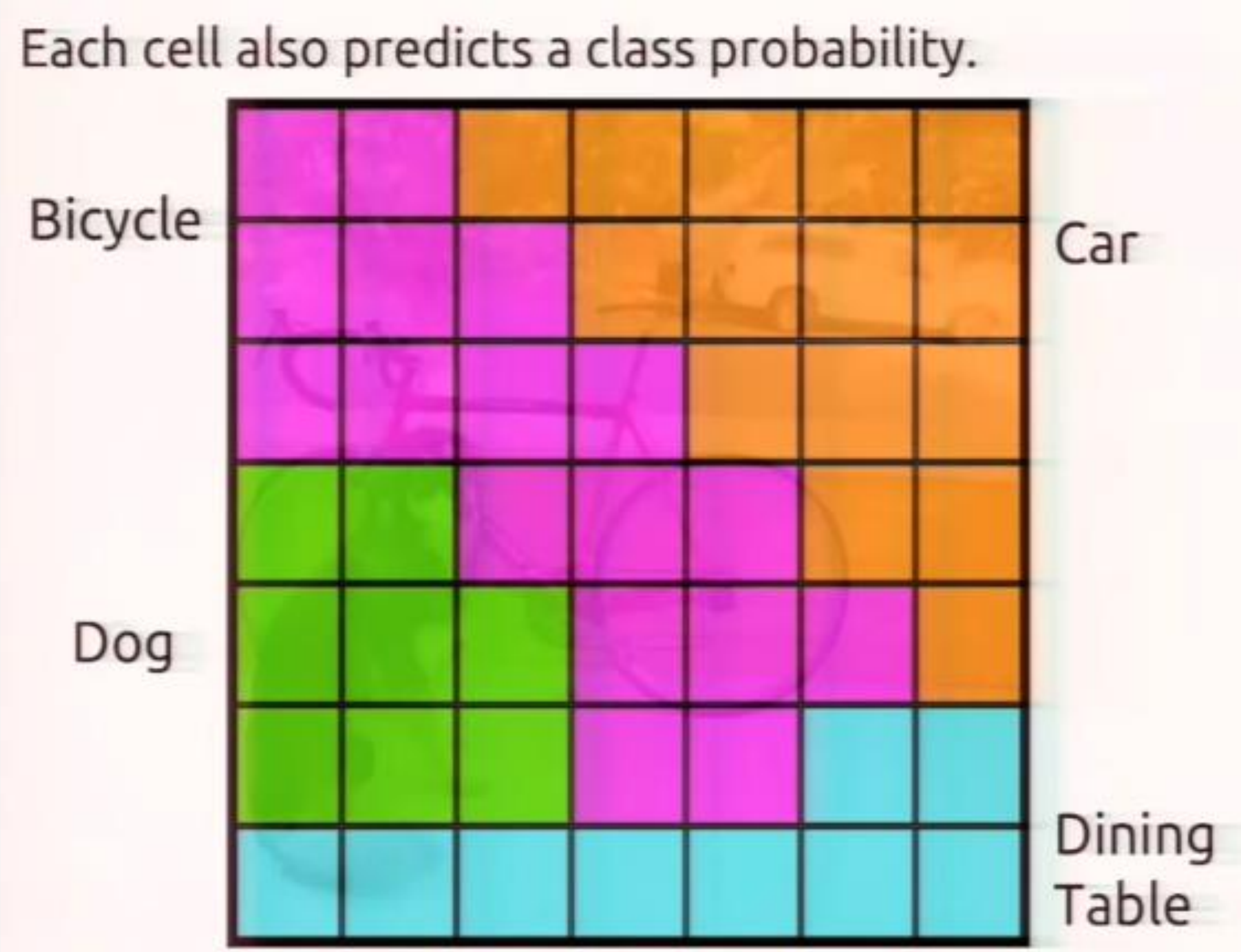
HAMIM

What this 7x7 tensor represents?

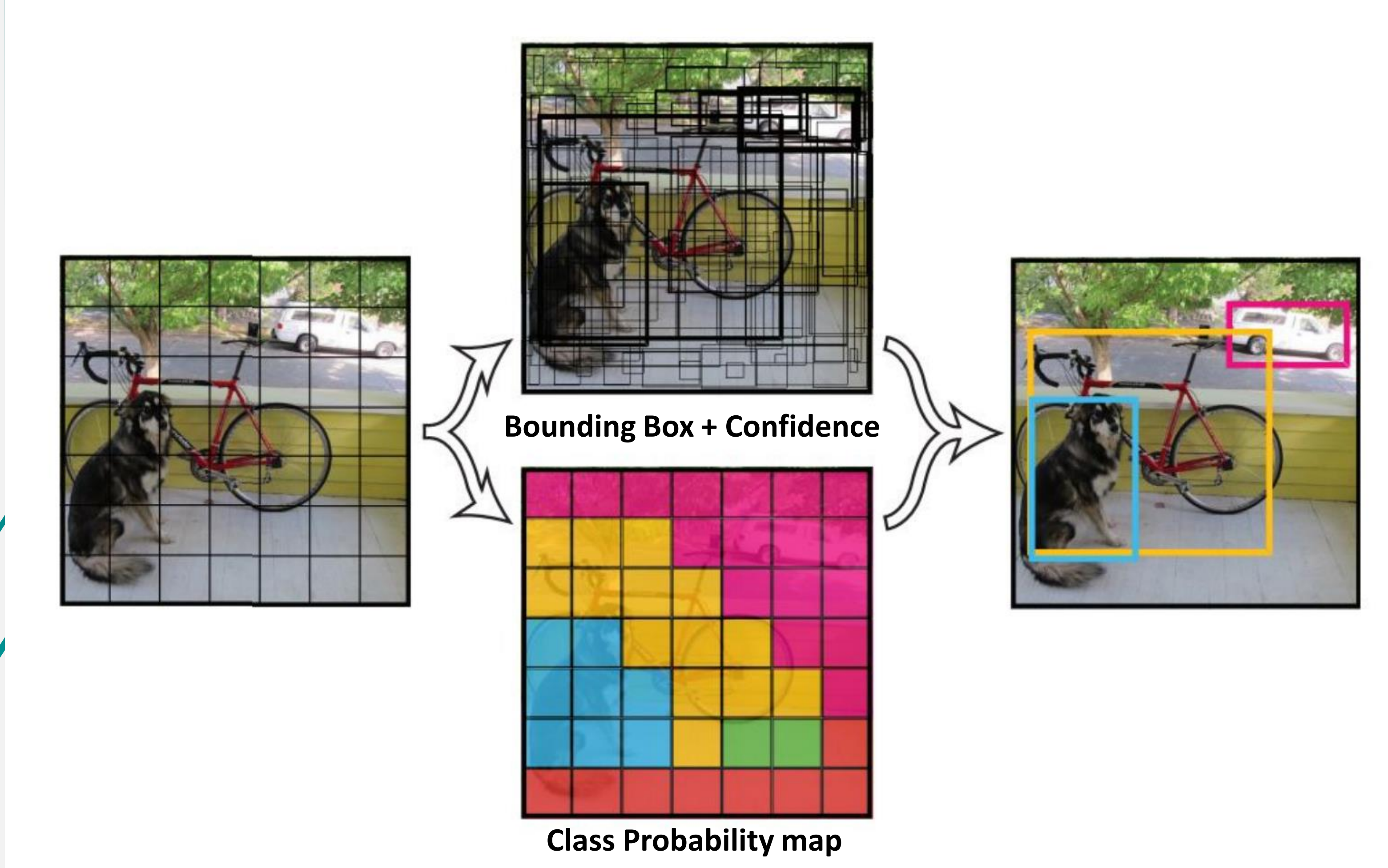


HAMIM

Class probability



Yolo Details



Confidence score

$$\Pr(\text{Object}) \times \text{IOU}_{pred}^{gt}$$

Conditional class probabilities

$$\Pr(\text{Class}_i \mid \text{Object})$$

HAMIM

Training YoLo

- Look which cell is near the center of the bounding box of the Ground truth. (Matching phase)

- Check from a particular cell which of it's bounding boxes overlaps more with the ground truth (IoU), then decrease the confidence of the bounding box that overlap less. (Each bounding box has it's on confidence)

- Decrease the confidence of all bounding boxes from each cell that has no object. Also don't adjust the box coordinates or class probabilities from those cells.

- Decrease the bounding boxes confidence of the cells that don't contain any object.

Yolo-Test Time

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}^{\text{truth}}_{\text{pred}} = \Pr(\text{Class}_i) * \text{IOU}^{\text{truth}}_{\text{pred}}$$

At test time we multiply the conditional class probabilities and the individual box confidence predictions.

Yolo Loss function

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

(X,Y) coordinate Loss

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

(W,H) Loss

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

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

Object/no object Loss

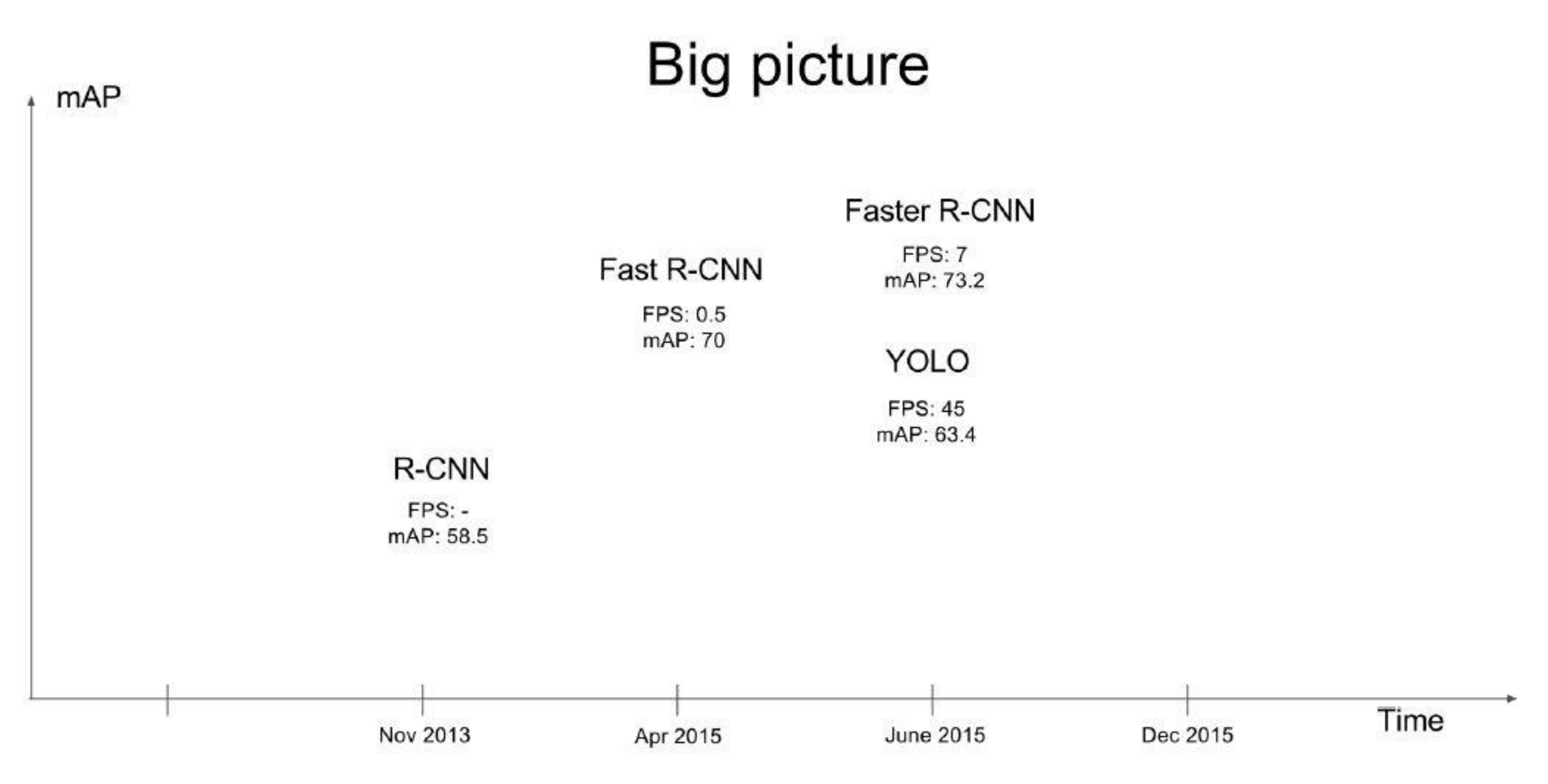
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Classification Loss

S: Grid size (7)

B: Number of bounding boxes

Comparison to other detection system



Limitations of YoLo

- Group of small objects
- Unusual aspect ratio