Objects detection

Diane Lingrand and many contributors

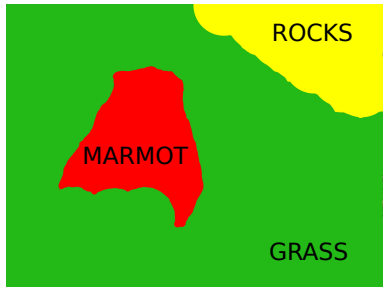
SI 4

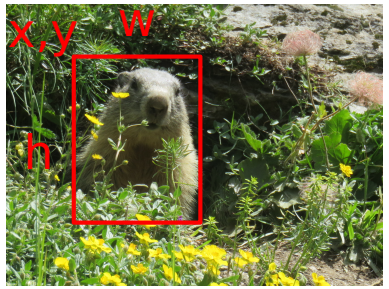
2010 - 2020

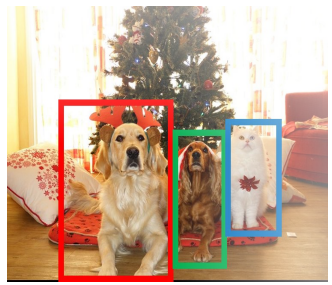
Outline

Detection

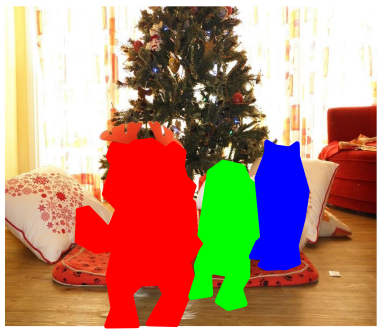
- Semantic segmentation :
  - label each pixel of the image
- Classification and Localisation : find one object in an image
  - label, position, size
- Object detection :
  - find all objects in an image
  - for each object : label, position, size
- Instance segmentation
  - segment all objects in an image :
  - label each pixel of the image. Labels for 2 different objects are different







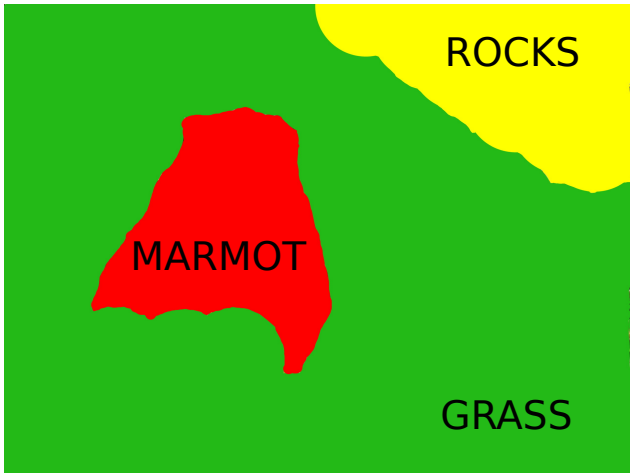
DOG DOG CAT



DOG DOG CAT

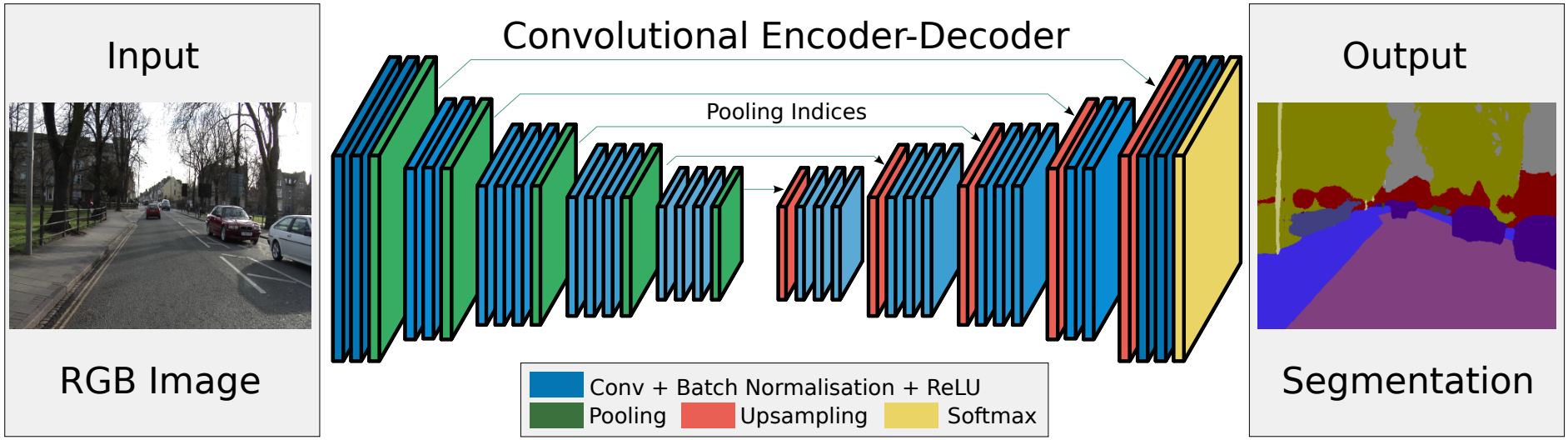
Outline

Semantic segmentation

From  to 

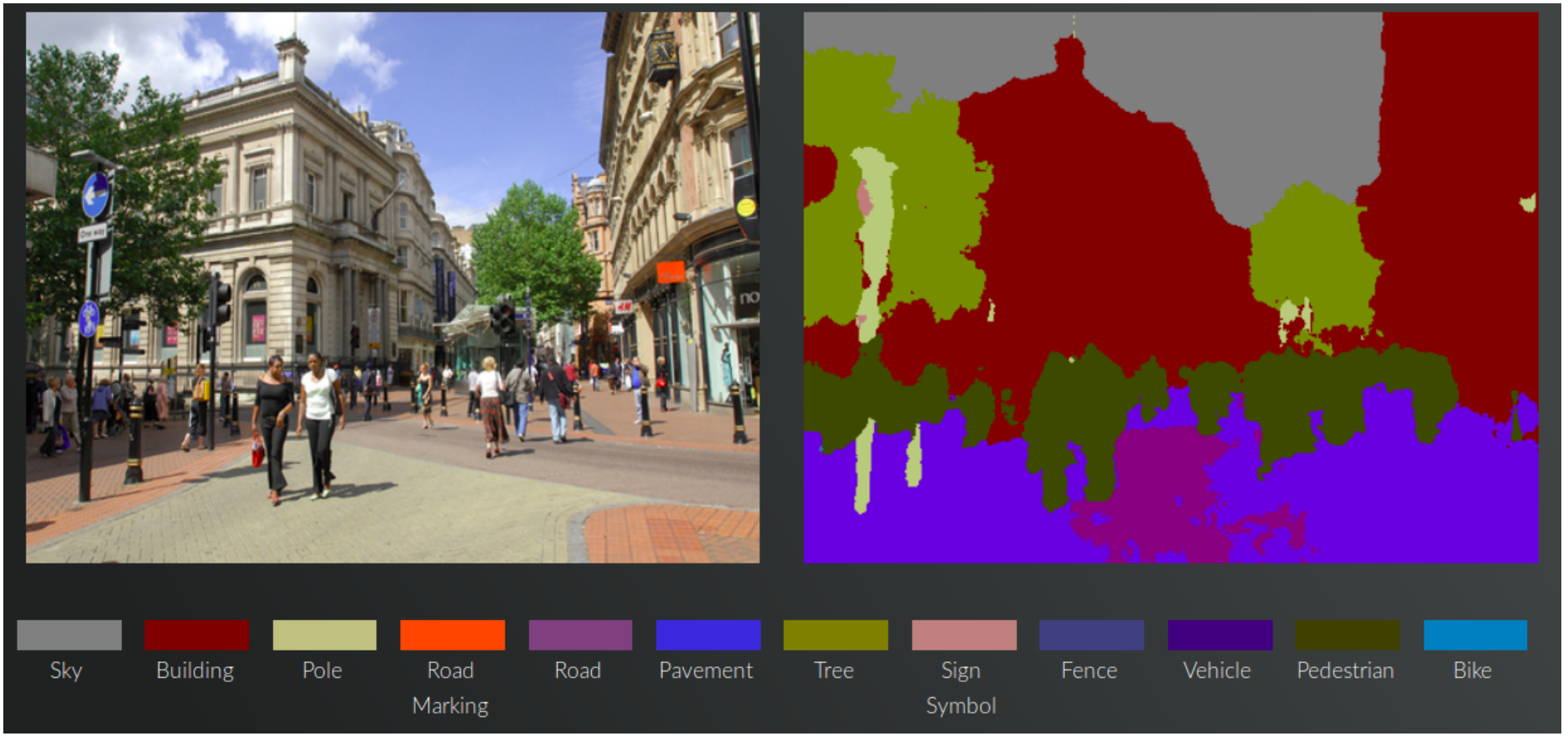
Convolution Autoencoder

- simply replace dense layers by convolutional layers
  - in keras, replace `Dense` by `Conv2D`
- application to segmentation : SegNet, 2015



from Badrinarayanan etal, arXiv :1511.00561v3

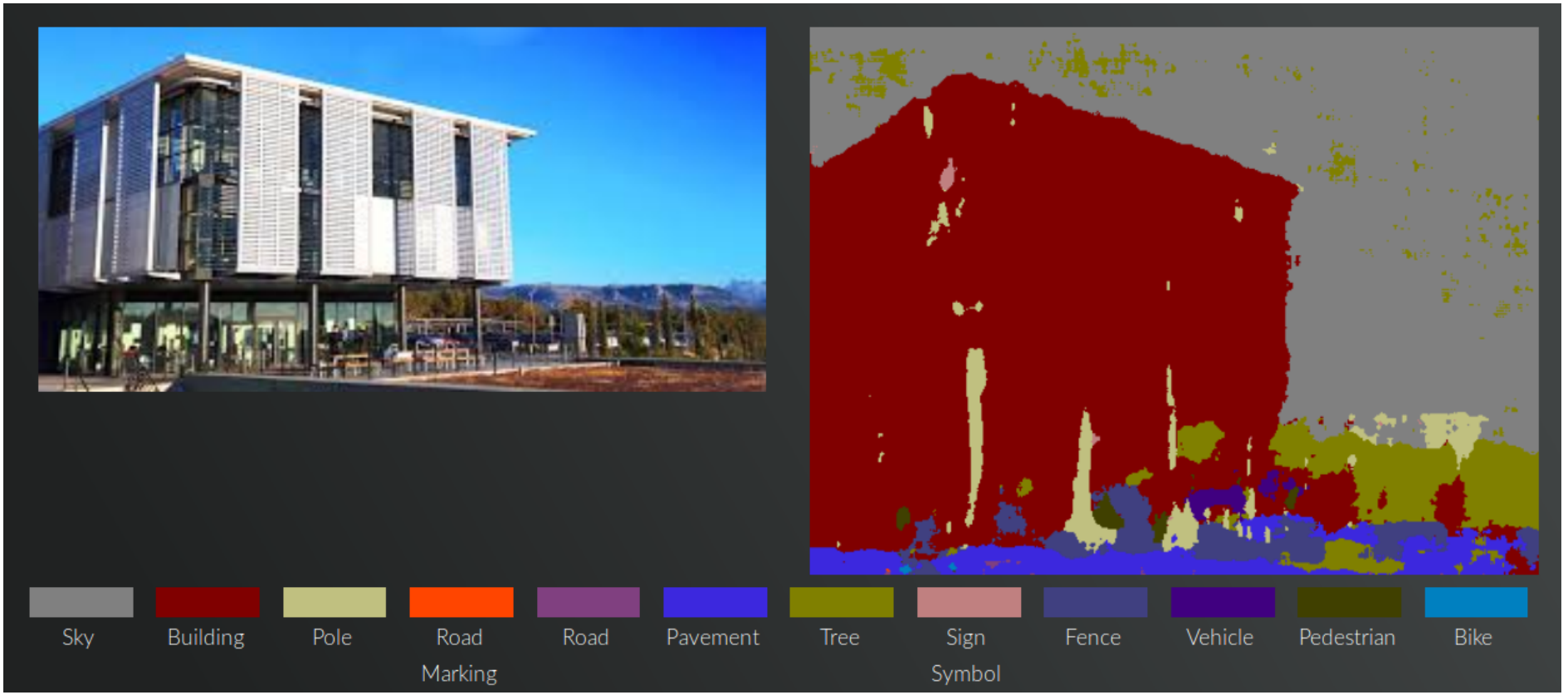
SegNet example I



also on youtube : `https://www.youtube.com/watch?v=CxanE_W46ts`

SegNet example II

try it yourself :
`http://mi.eng.cam.ac.uk/projects/segnet/demo.php`

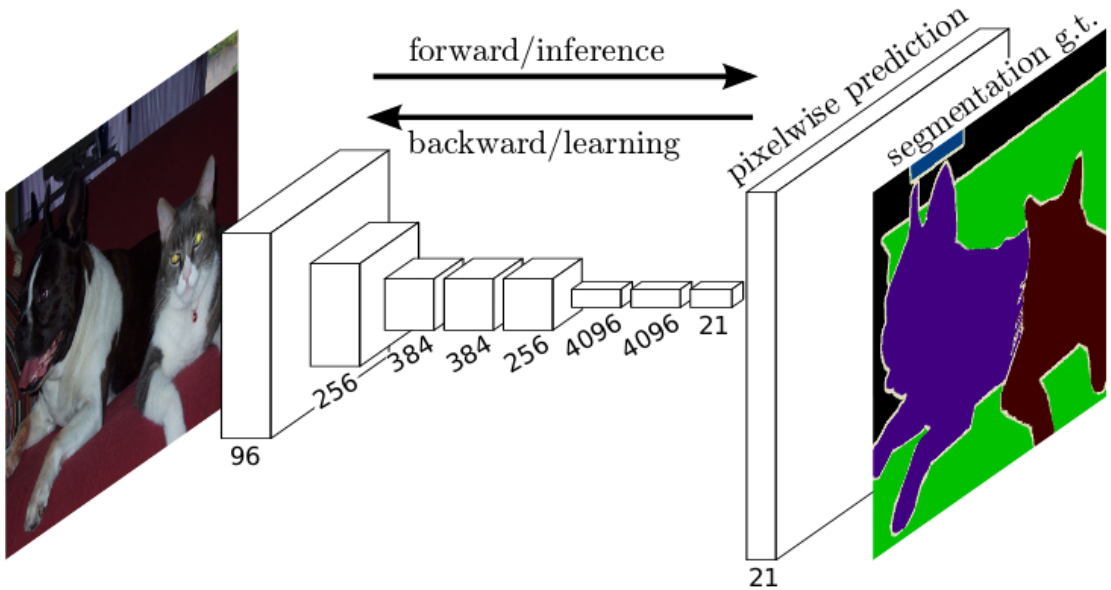


Fully Convolutional Networks (FCN), 2015

Fully Convolutional Networks for Semantic Segmentation

Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley

https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf



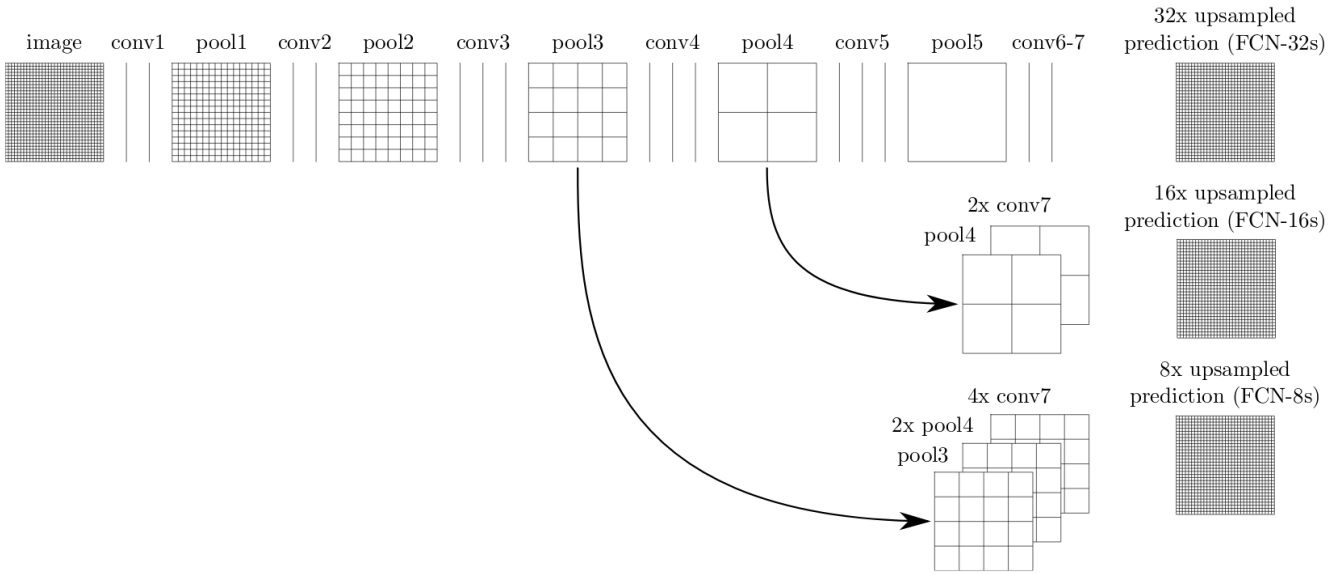


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Pooling and prediction layers are shown as grids that reveal relative spatial coarseness, while intermediate layers are shown as vertical lines. First row (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Second row (FCN-16s): Combining predictions from both the final layer and the `pool4` layer, at stride 16, lets our net predict finer details, while retaining high-level semantic

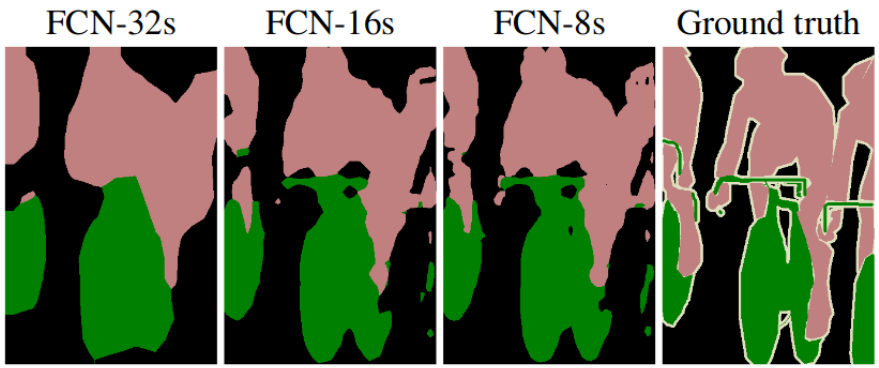


Figure 4. Refining fully convolutional nets by fusing information from layers with different strides improves segmentation detail. The first three images show the output from our 32, 16, and 8 pixel stride nets (see Figure 3).

PSPNet Pyramid Scene Parsing Network (2016-17)

Pyramid Scene Parsing Network

Hengshuang Zhao[1] Jianping Shi[2] Xiaojuan Qi[1] Xiaogang Wang[1] Jiaya Jia[1]
[1]The Chinese University of Hong Kong [2]SenseTime Group Limited

https://arxiv.org/pdf/1612.01105.pdf

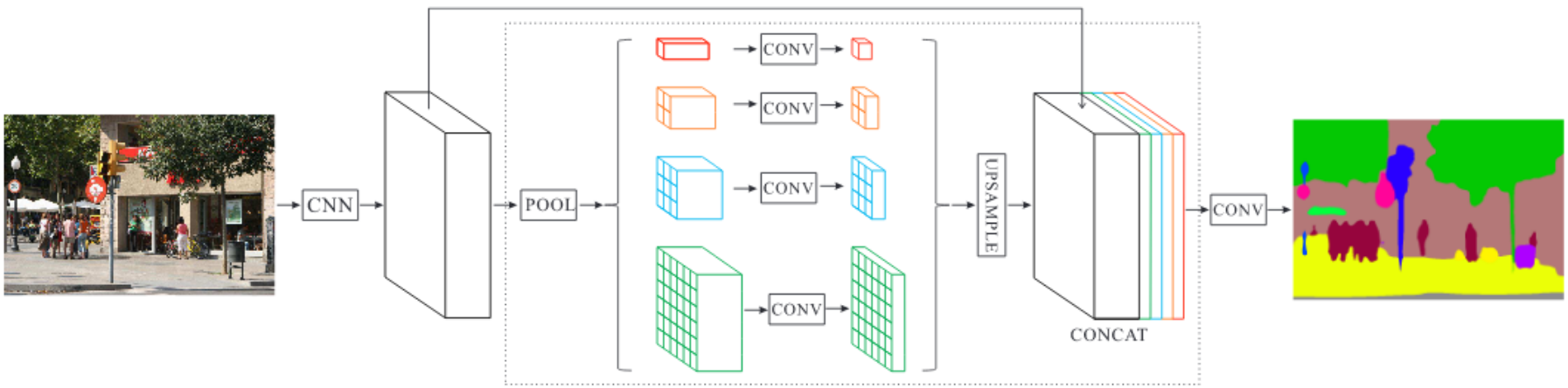
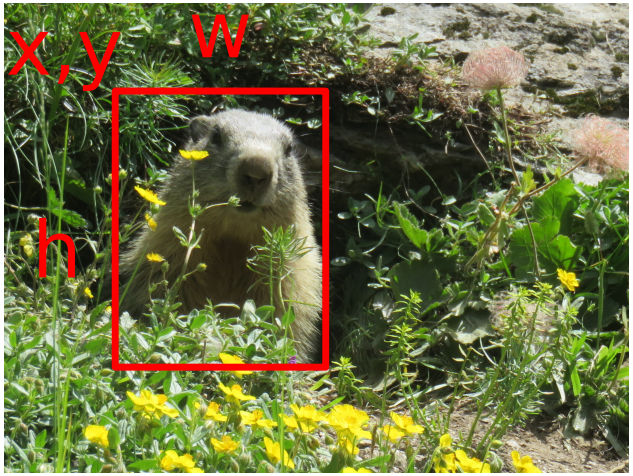
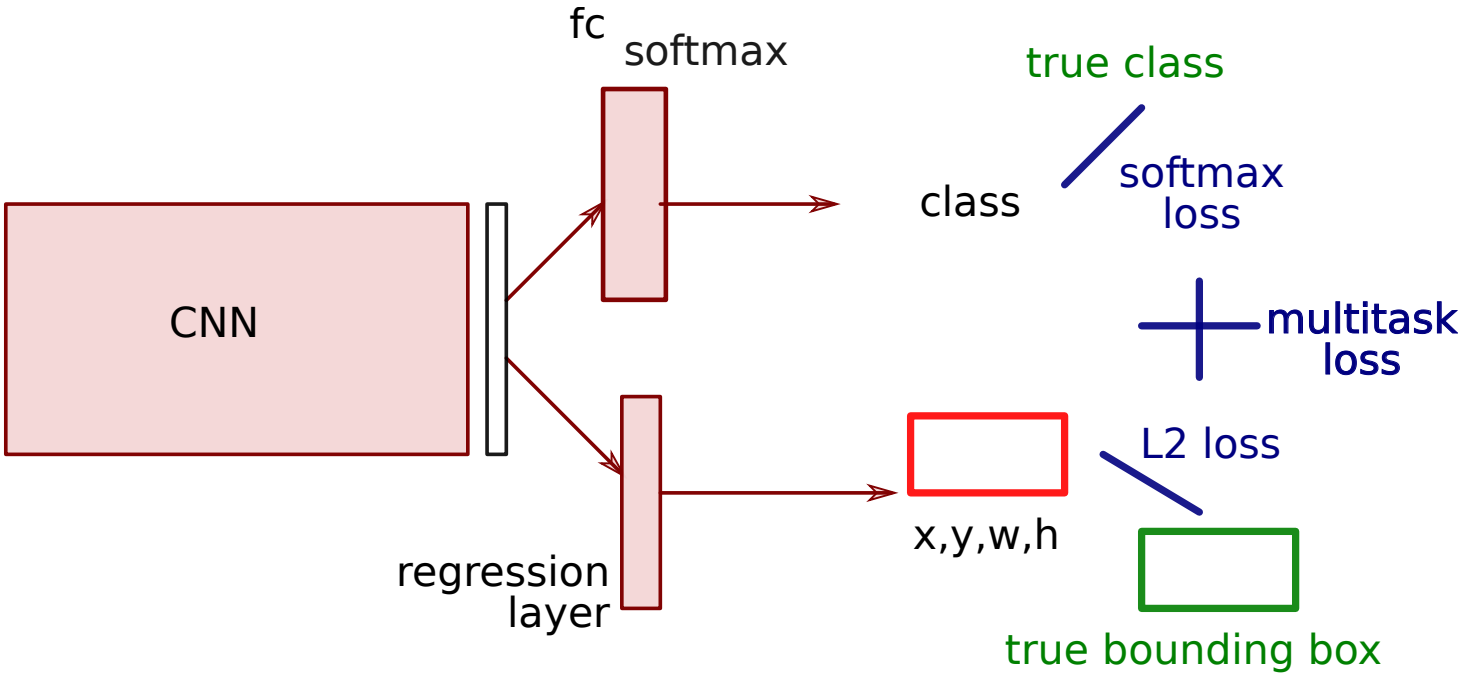


Figure 3. Overview of our proposed PSPNet. Given an input image (a), we first use CNN to get the feature map of the last convolutional layer (b), then a pyramid parsing module is applied to harvest different sub-region representations, followed by upsampling and concatenation layers to form the final feature representation, which carries both local and global context information in (c). Finally, the representation is fed into a convolution layer to get the final per-pixel prediction (d).

Outline

Classification and Localisation





Outline

Object detection

- Focus on some classes and add the *background* class
- You don't know how many objects you expect to find
- First idea : sliding window
  - huge number of locations and scales
- Region proposals
  - using basic image processing (edges,...)
  - select a "small" numbers of windows (around 2000)
  - most windows are noisy windows
  - Then apply classification on each window

Slow R-CNN

Regions of Interest (RoI) from a proposal method (~2k)

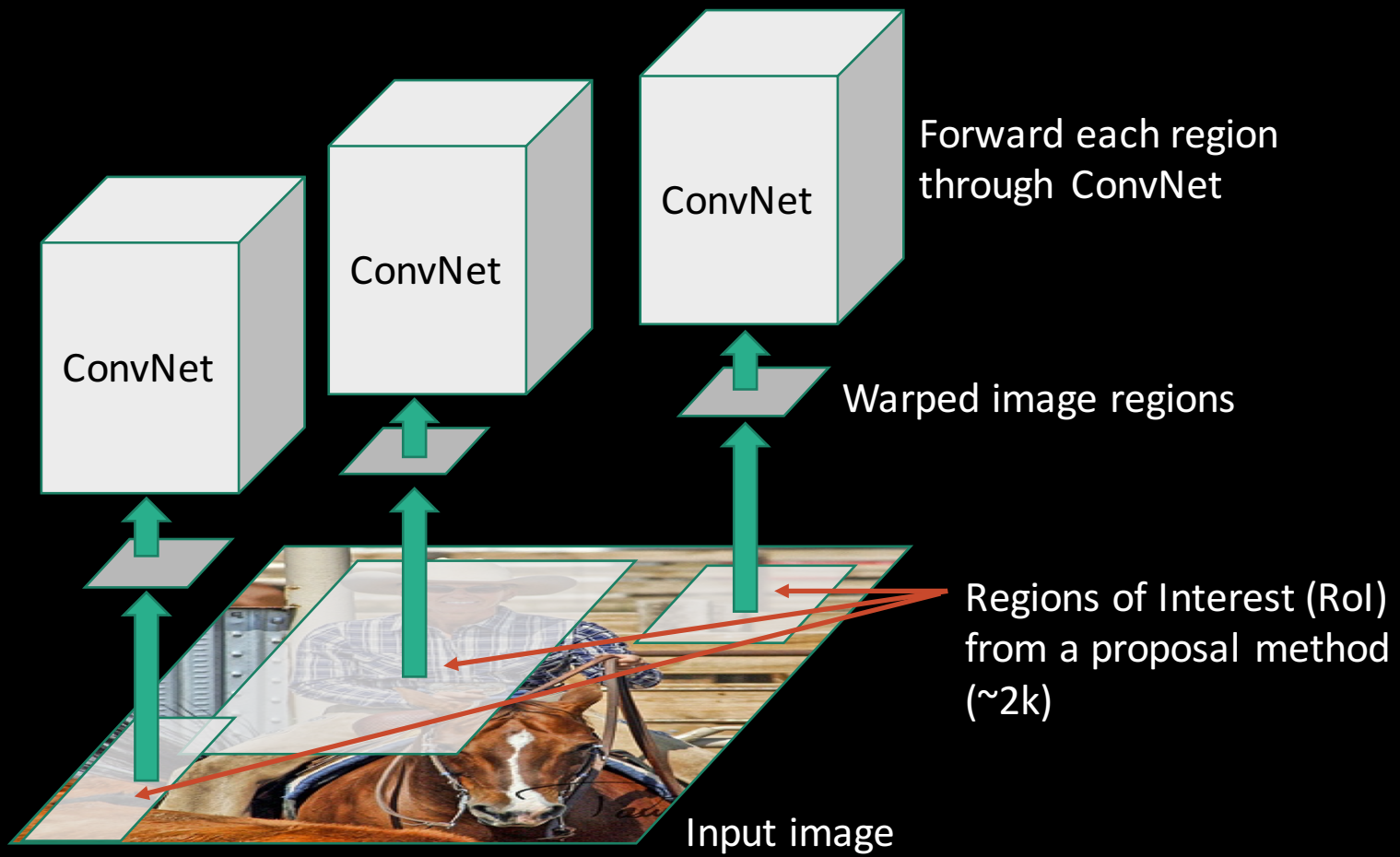
Input image

Girshick et al. CVPR14.

Slow R-CNN

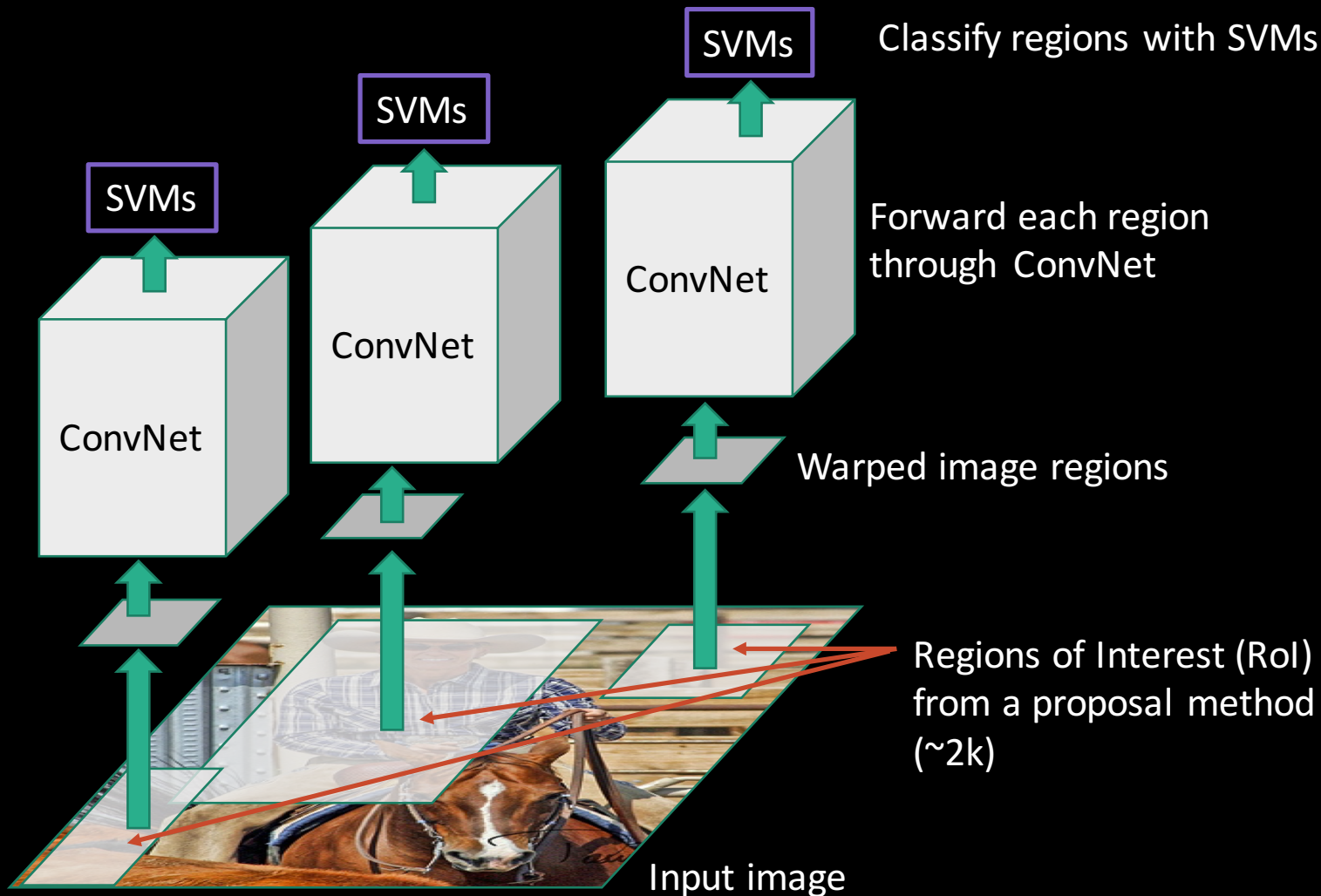
Girshick et al. CVPR14.

Slow R-CNN



Girshick et al. CVPR14.

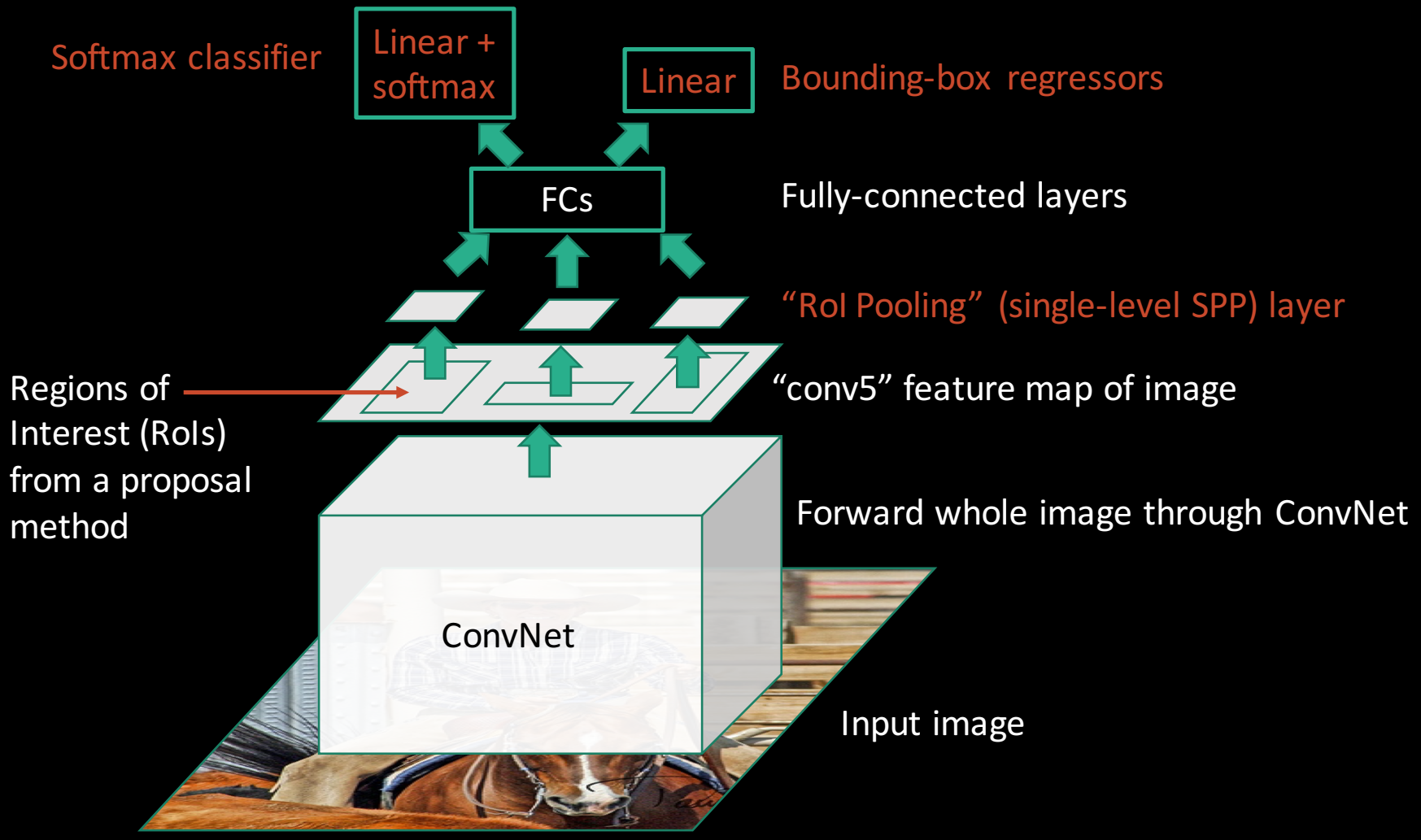
Slow R-CNN



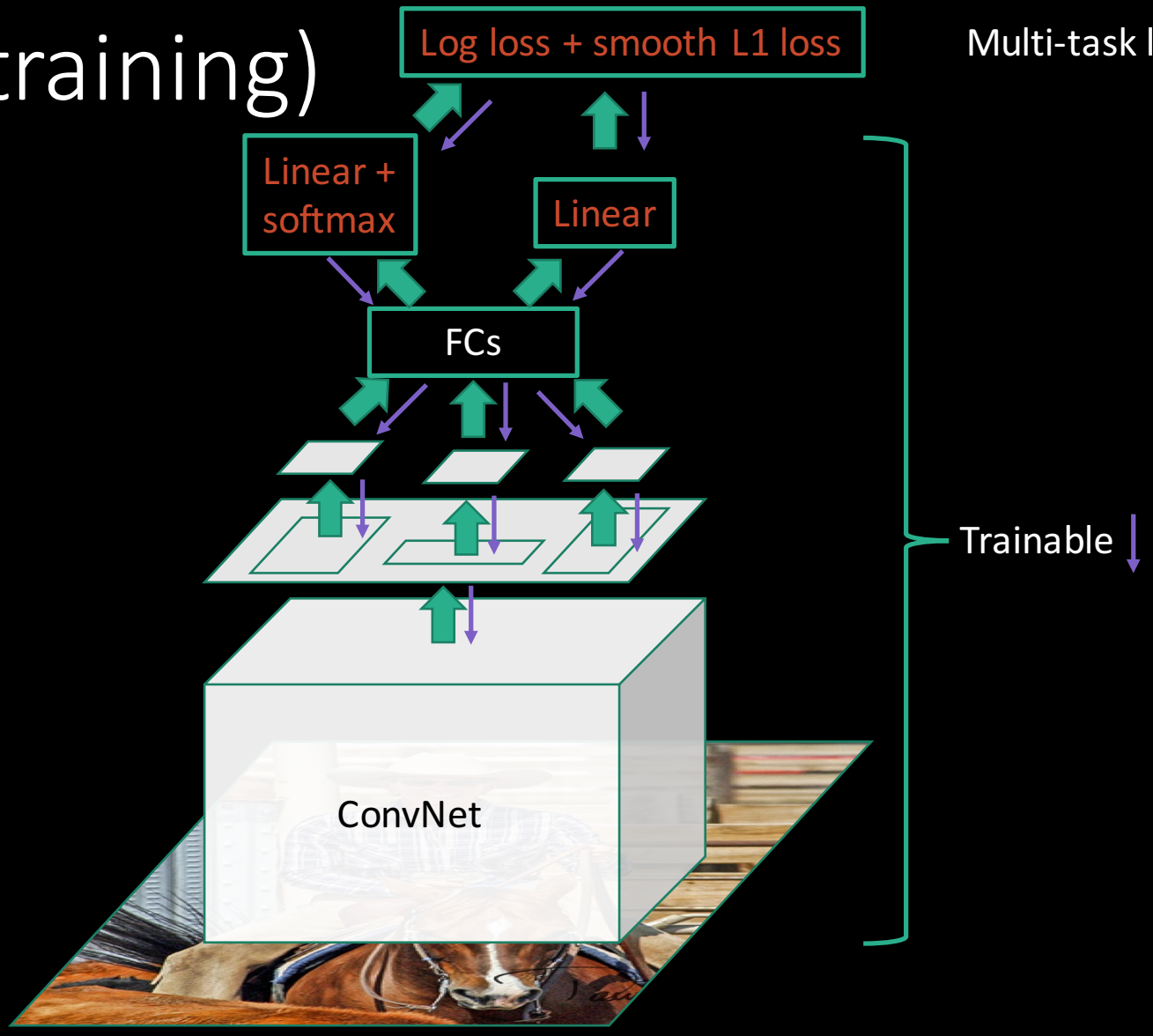
Girshick et al. CVPR14.

Post hoc component

Fast R-CNN (test time)



Fast R-CNN (training)



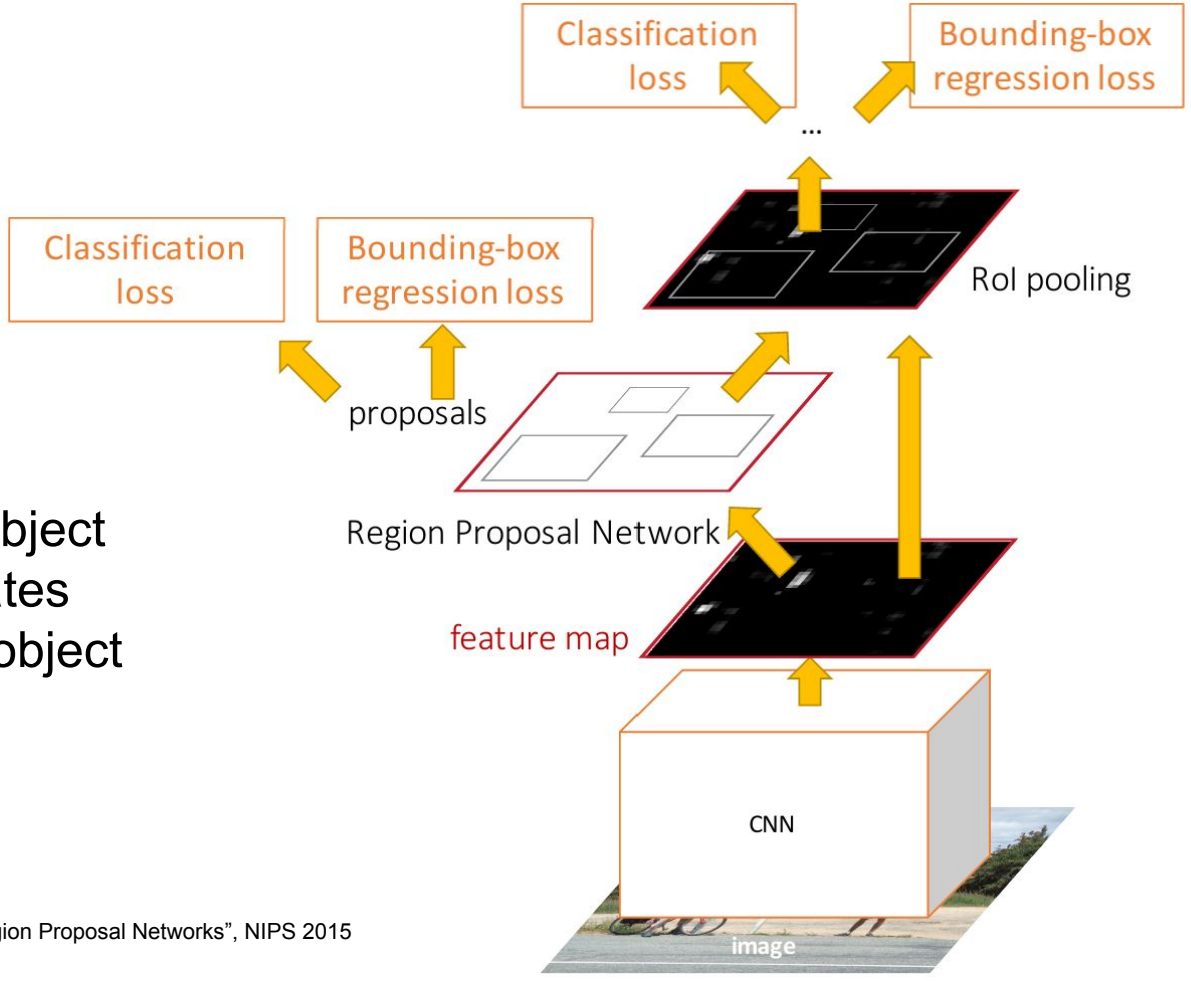
Faster R-CNN:

Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features

Jointly train with 4 losses:

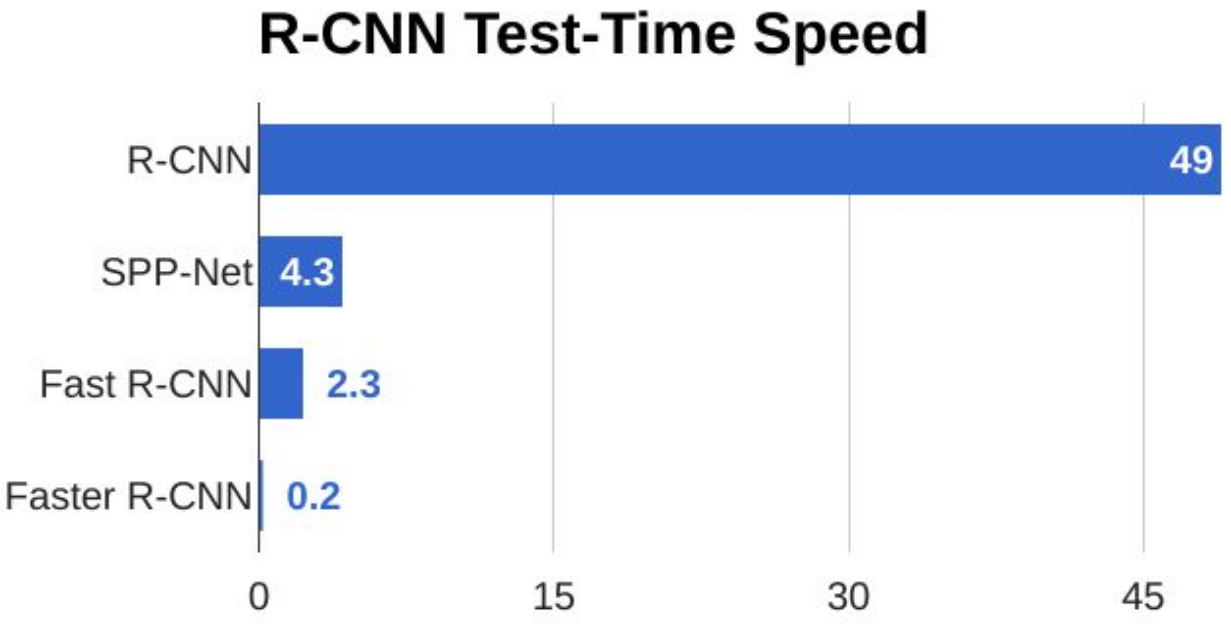
1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates



Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Faster R-CNN:

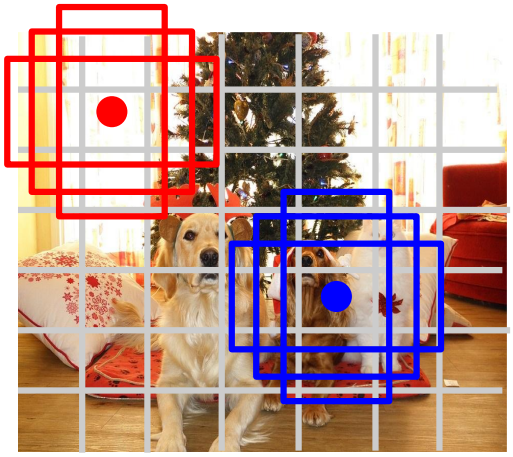
Make CNN do proposals!



Detection without Proposals: YOLO / SSD

Go from input image to tensor of scores with one big convolutional network!

Input image
3 x H x W



Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

Within each grid cell:

- Regress from each of the B base boxes to a final box with 5 numbers: (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

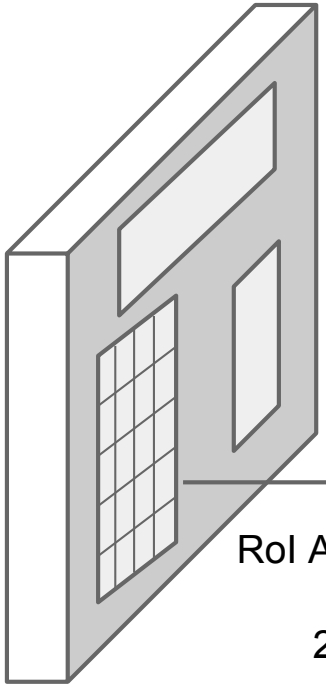
Output:
7 x 7 x (5 * B + C)

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016

Outline

Mask R-CNN

CNN



RoI Align

256 x 14 x 14

Conv

256 x 14 x 14

Conv

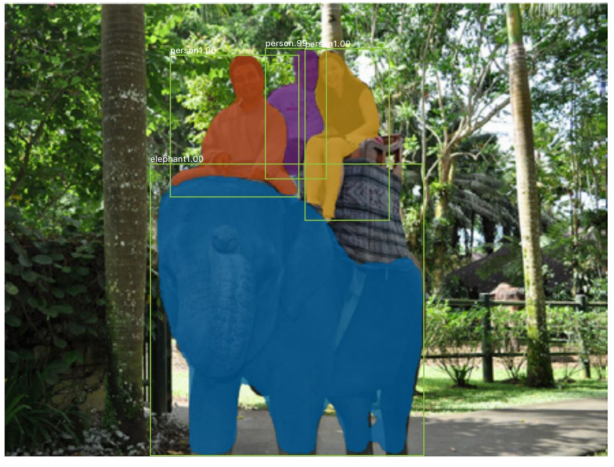
Classification Scores: C
Box coordinates (per class): 4 * C

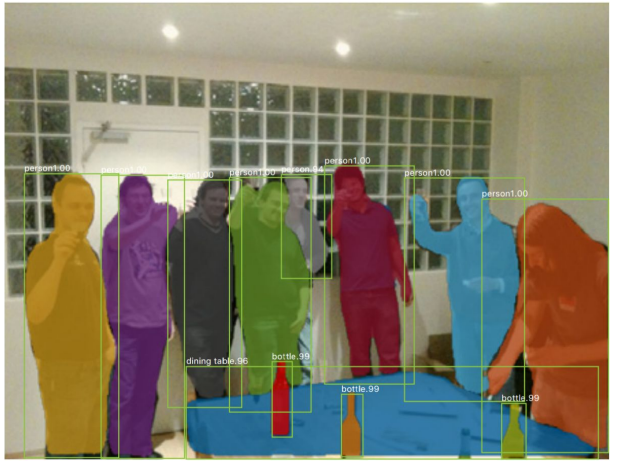
Predict a mask for each of C classes

C x 14 x 14

He et al, “Mask R-CNN”, arXiv 2017

Mask R-CNN: Very Good Results!





He et al, “Mask R-CNN”, arXiv 2017