

# Segmentation

1. **Definition**
2. Beyond ImageNet: Fully Convolutional Networks (FCNs)
3. Supervised Segmentation with Deep ConvNets

# Segmentation: definitions



# Def1: **Semantic Segmentation**

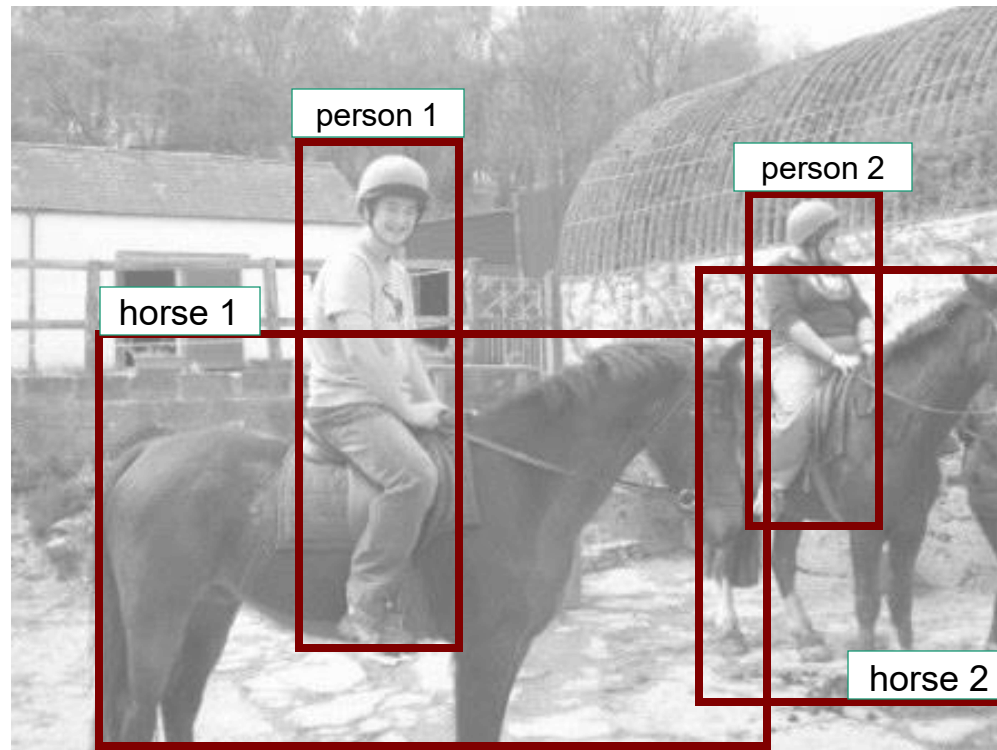
*Label each pixel with a category label*



■ horse  
■ person

# Object Detection

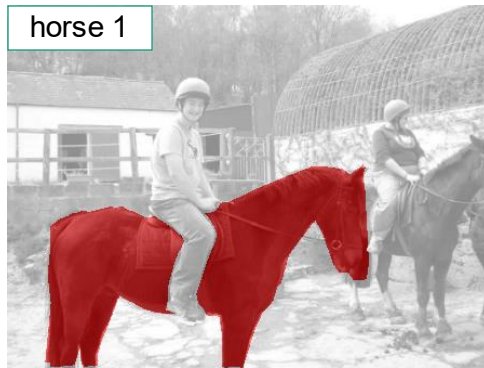
*Detect every instance of the category and localize it with a bounding box.*



## Def2: Instance segmentation

Simultaneous Detection and Segmentation

*Detect and **segment** every **instance** of the category in the image*



# Supervised Segmentation

Training data





# Supervised Segmentation

## PASCAL VOC 12

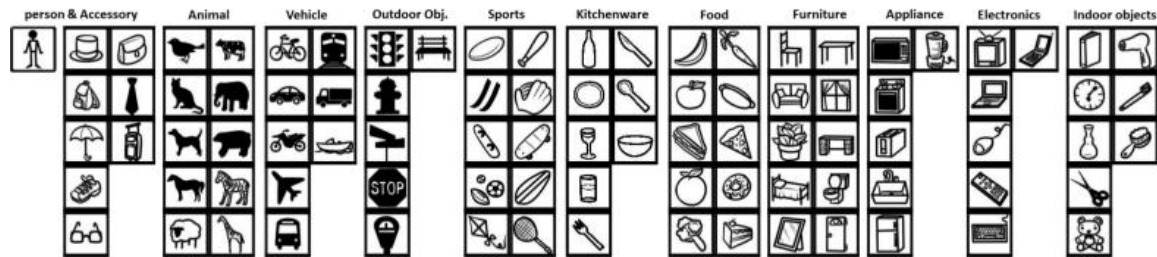
- Train 1464 images / Val 1449 images / Test 1456 images
- 21 classes: *aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, tv/monitor* + background
- Evaluation: intersection-over-union metric
- Webpage: <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/>



# Supervised Segmentation

## COCO

- Train 80k images / Val 20k images
- 91 classes, 11 super-categories:



- 3 challenges: detection, instance segmentation, captioning
- Webpage: <http://mscoco.org> [paper]



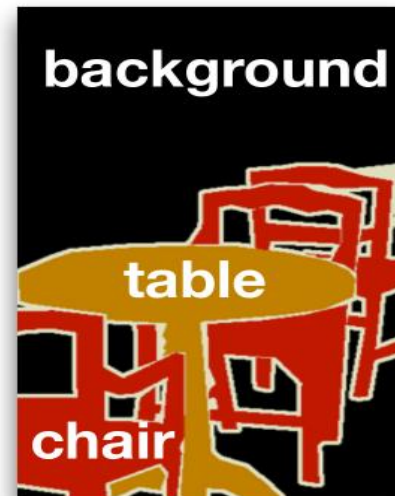


# Supervised Segmentation

## Supervised Image Segmentation Methods

### Full supervision

- Precise annotation ,
- Expensive and time consuming to obtain
  - ┆ "79s per label per image" [RBFL15] /
- Bottleneck for learning models at large scale /



# Supervised Segmentation

## Weakly Supervised Image Segmentation Methods

### Weak supervision

- Reduce supervision: class labels (or tags) / /
- Cheap to obtain
  - ┆ "1s per label per image" [RBFL15] ,
- Scalable to large number of categories ,



3 background  
7 aeroplane  
7 cat  
3 chair  
7 dog  
7 person  
7 sheep  
3 table  
7 tvmonitor

# Segmentation

1. Definition
2. **Beyond ImageNet: Fully Convolutional Networks (FCNs)**
3. Supervised Segmentation with Deep ConvNets

# From ImageNet to complex scenes

- ImageNet: huge dataset (1.2M training images) with labels ... but centered objects

ImageNet



- How to apply/ adapt/ modify learning strategies to deal with:

VOC 2012



MS COCO



# From ImageNet to complex scenes?

- Working on datasets with complex scenes (large and cluttered background), not centered objects, variable size, ...



VOC07/ 12



MIT67



15 Scene



COCO



VOC12 Action

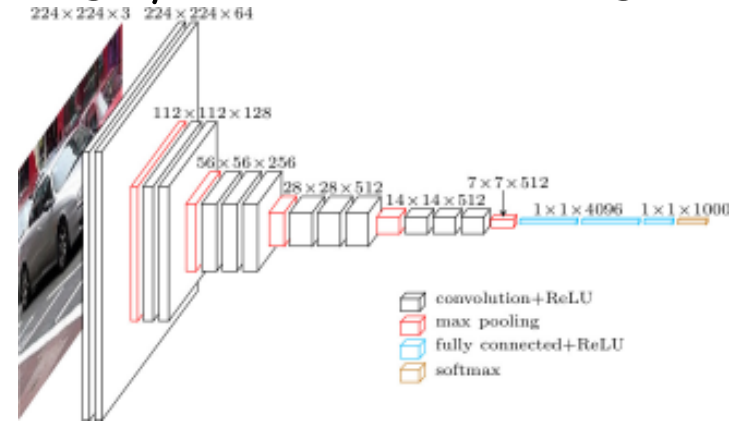
- Select relevant regions ! better prediction



- Full annotations expensive ) training with weak supervision



# How to adapt VGG16 archi. for large/complex images?

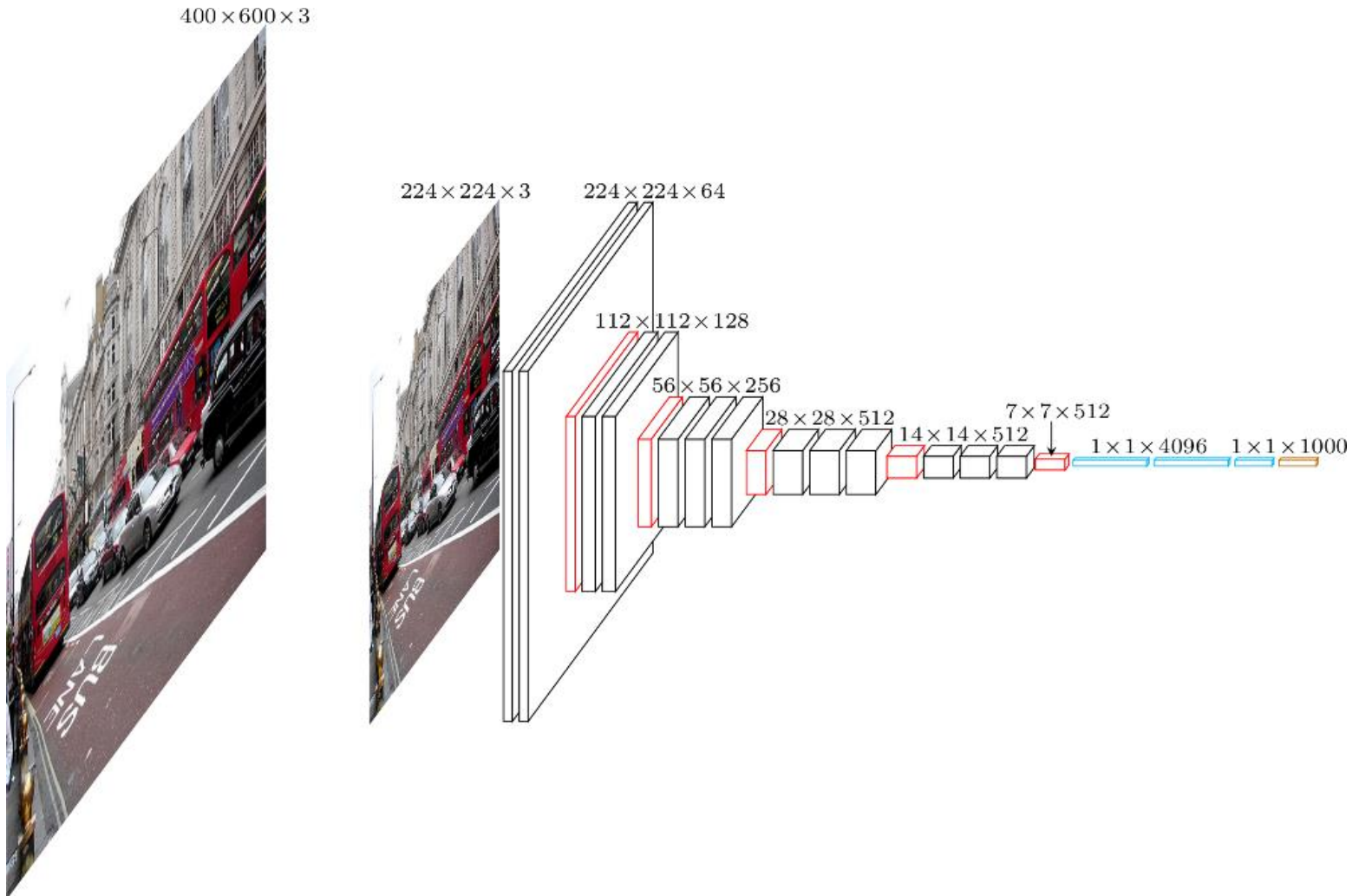


?

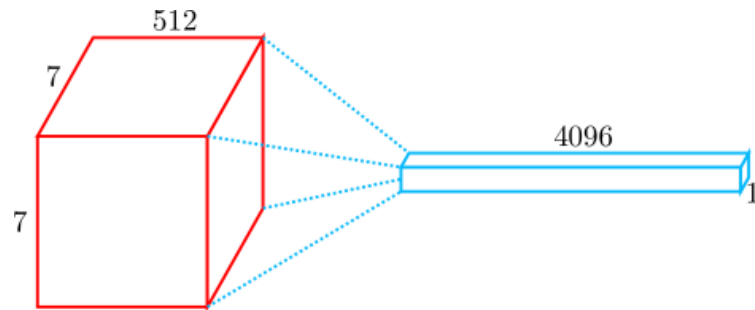
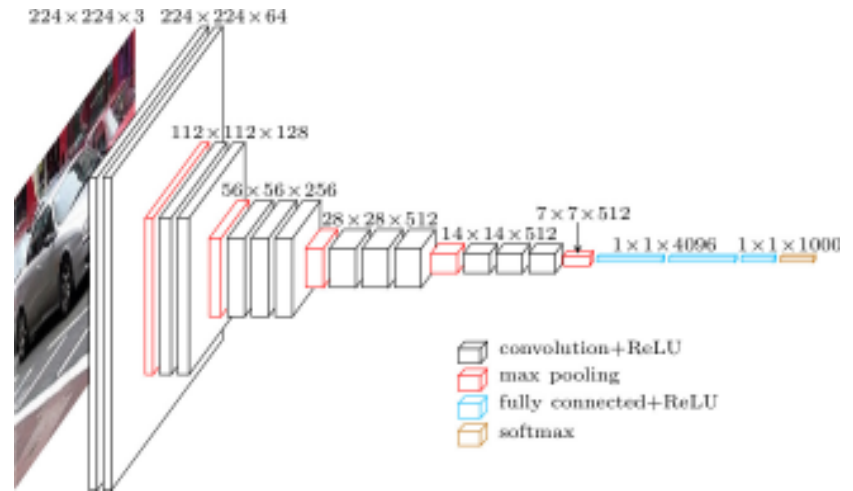


# Naïve approach: brut transfer (next Section)

- Resize the image



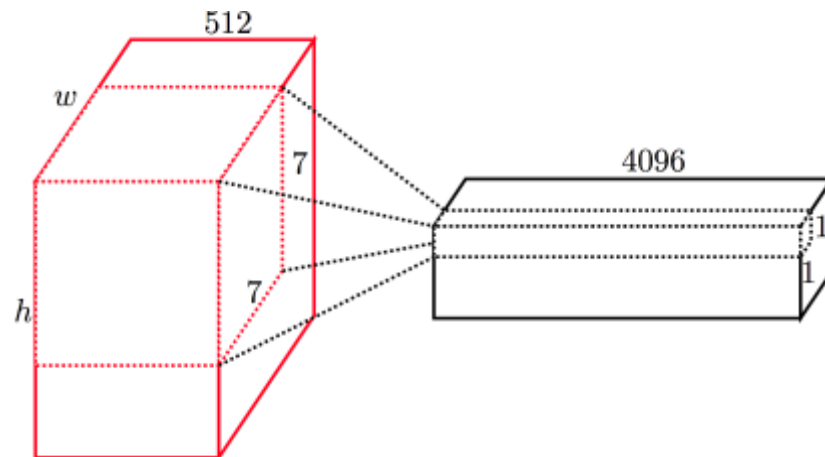
# Sliding window ) convolutional layers



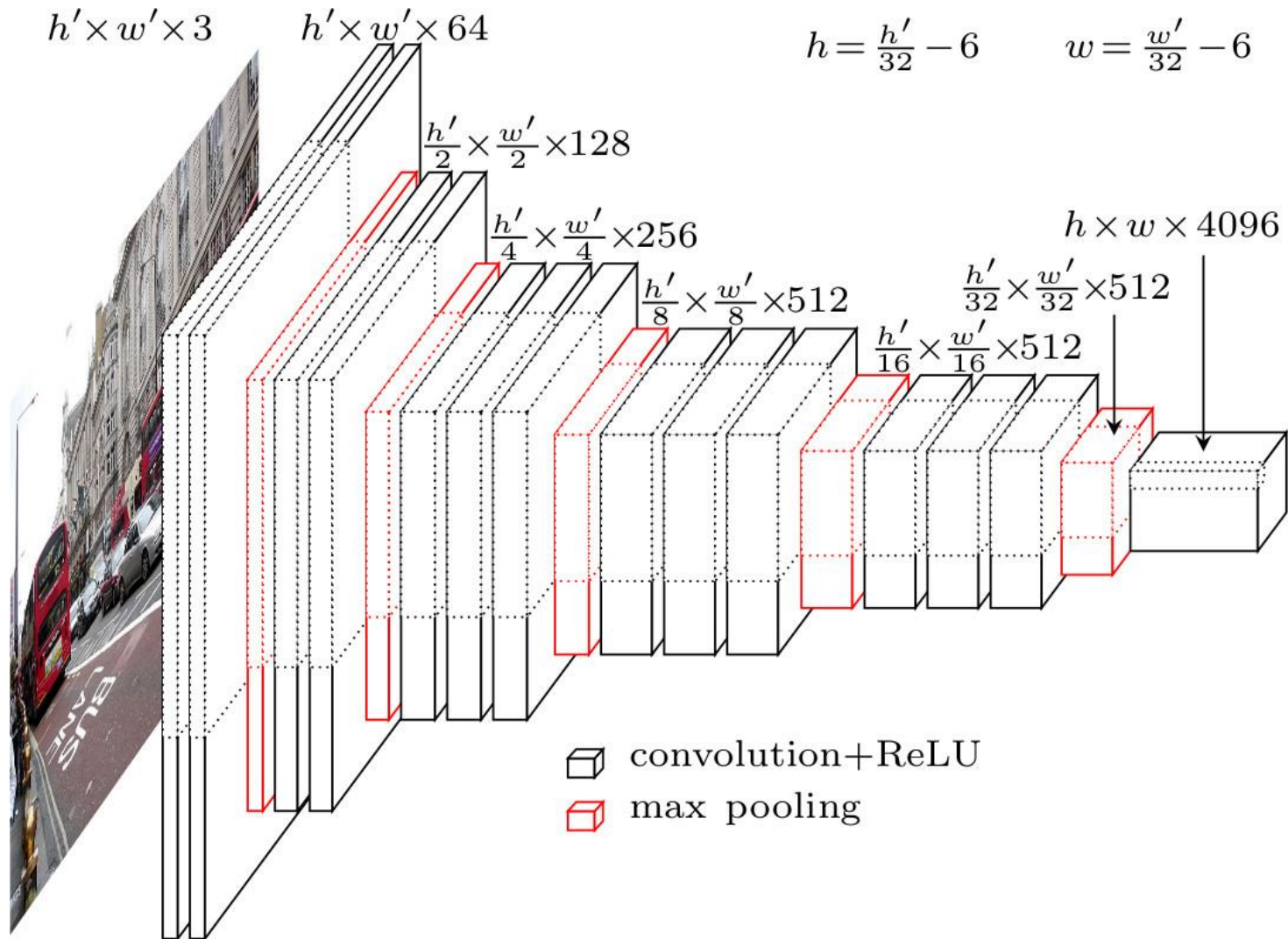
# Sliding window ) convolutional layers



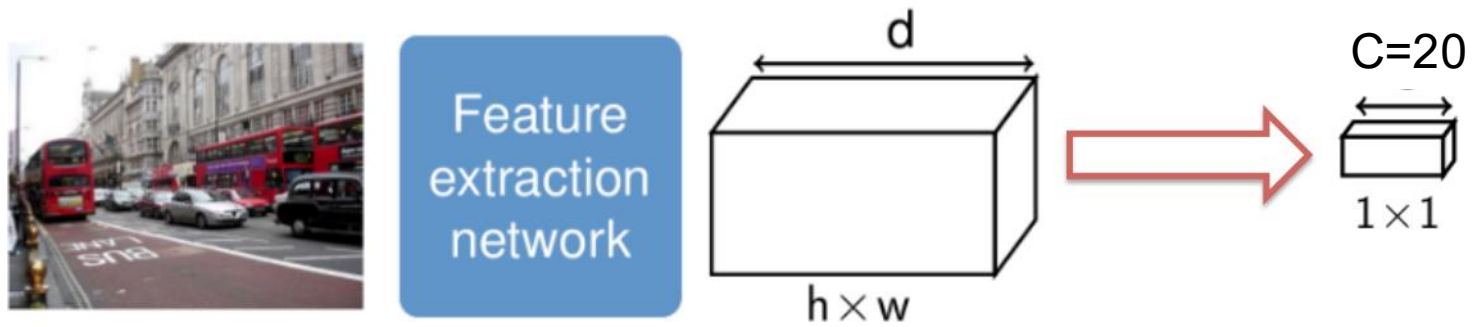
- Fully connected as convolutional layer (here 4096 conv. filters 7x7x512)



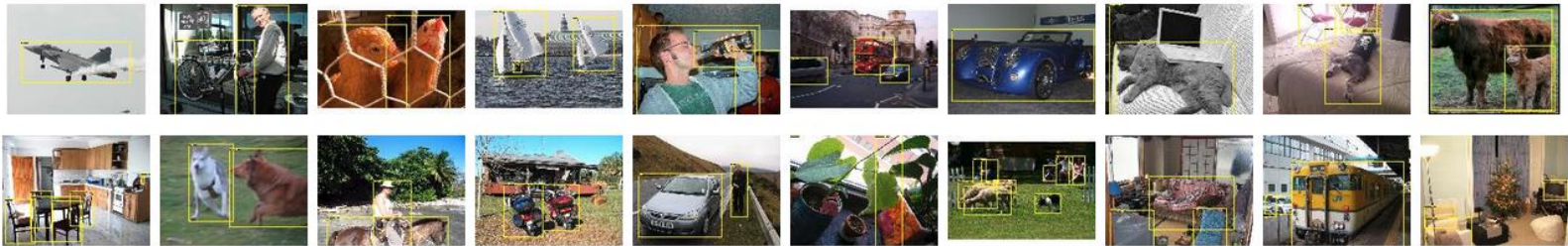
# Sliding window $\Rightarrow$ convolutional layers



# Transfer – Pooling – Classification

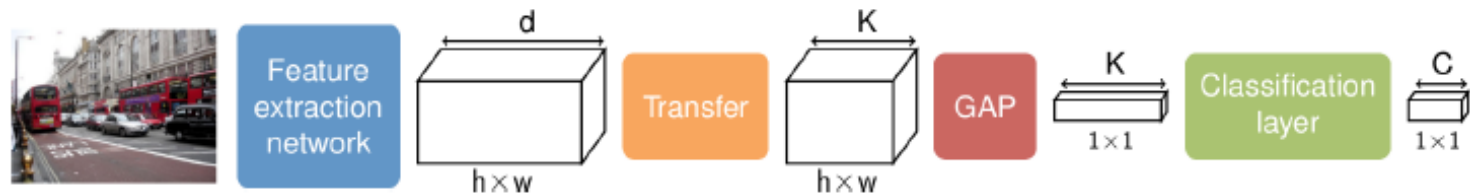


20 classes

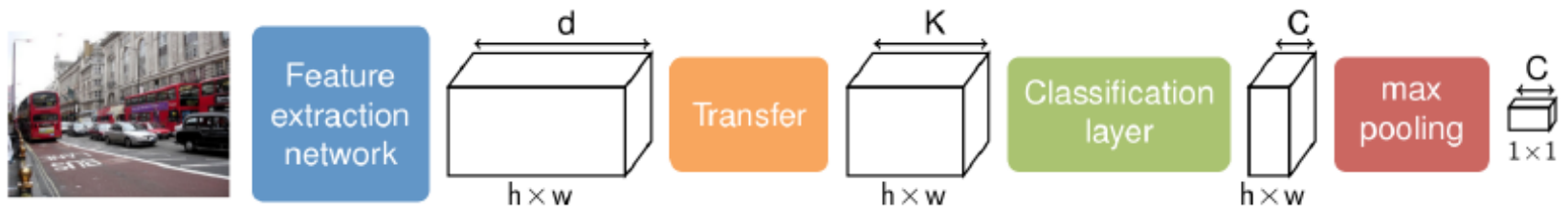


# Transfer+classif+pooling

- Transfer+pooling (GAP)+classif

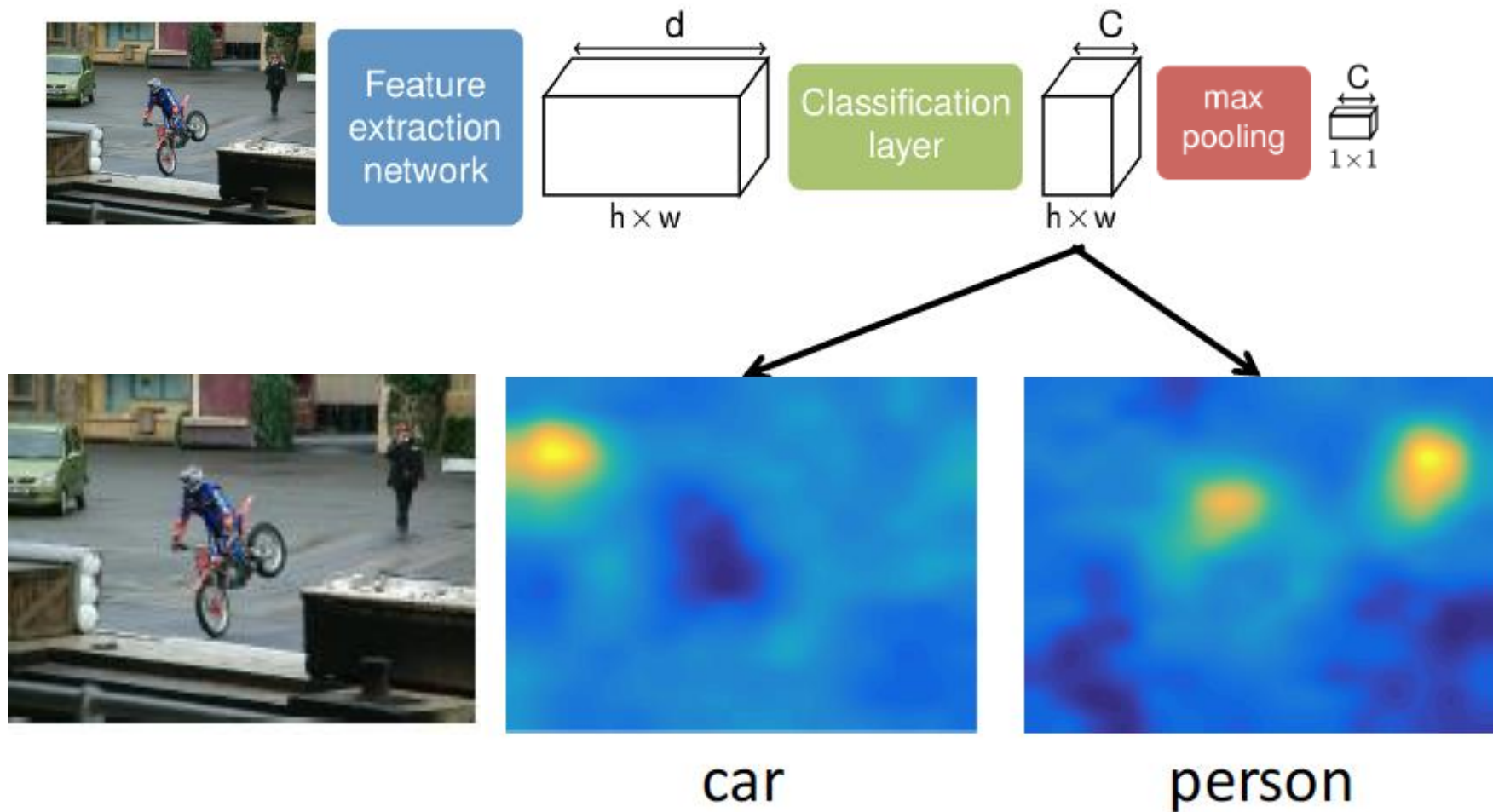


- Transfer+classif+Max pooling

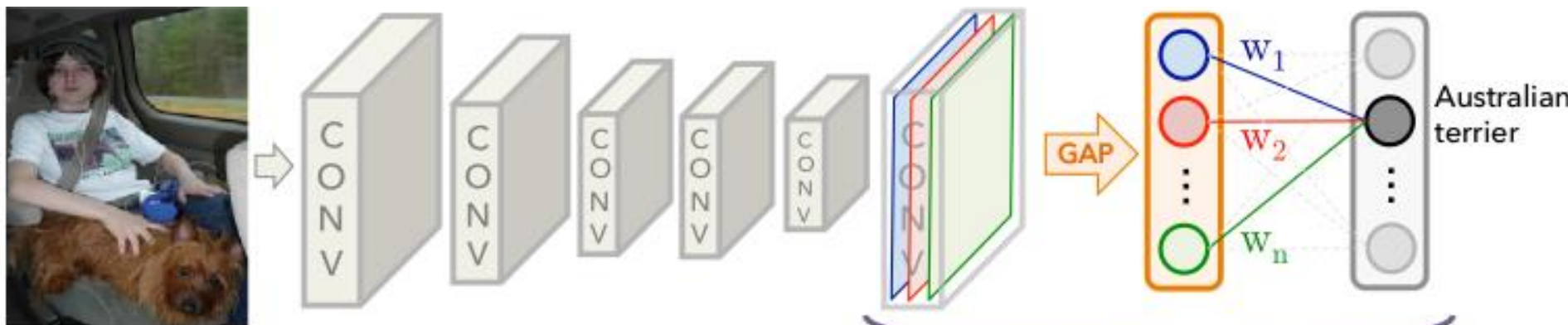




# Pixel contribution to the classification



# Pixel contribution to the classification

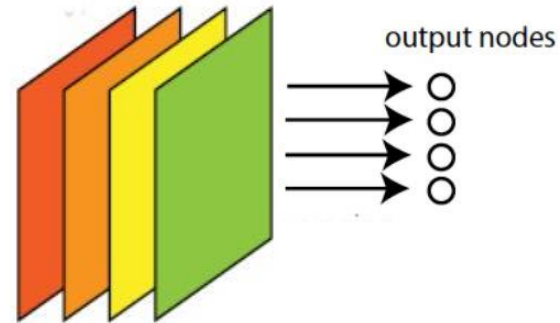


## Class Activation Mapping



# Pooling schemes

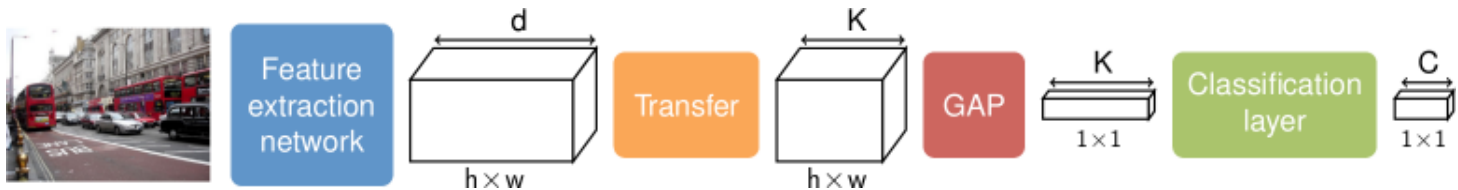
- Max [Oquab, CVPR15]



$$y^c = \max_{i,j} z_{ij}^c$$

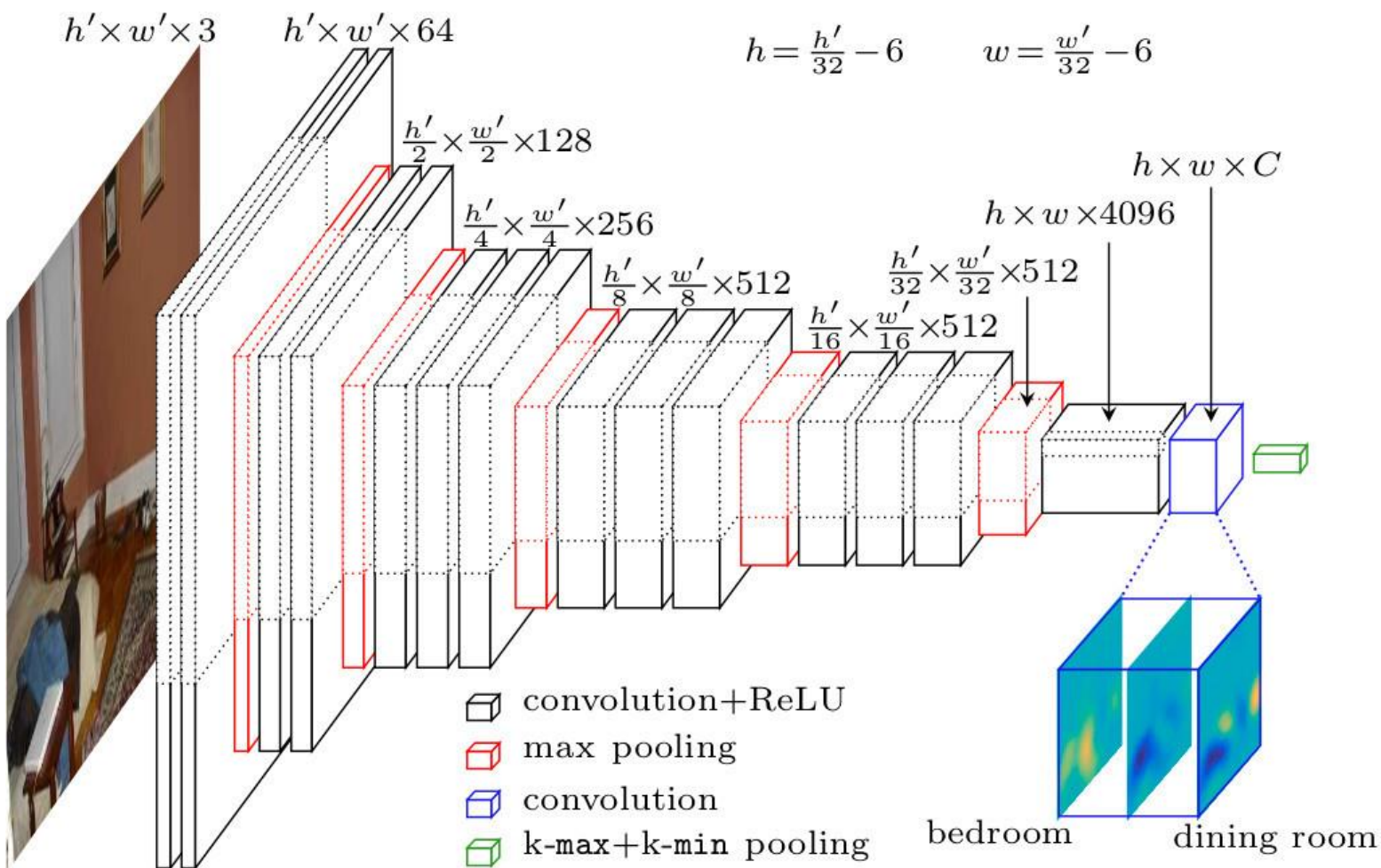
- GAP [Zhou, CVPR16]

$$y^c = \frac{1}{N} \sum_{i,j} z_{ij}^c$$



- LSE [Pinheiro, CVPR15] / SPLeap [Kulkarni, ECCV16]

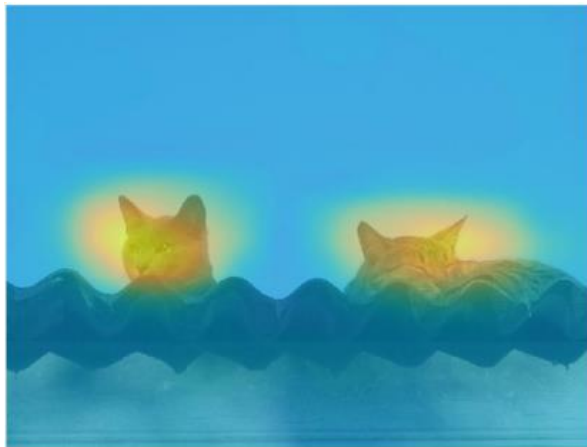
$$y^c = \frac{1}{\beta} \log \left( \frac{1}{N} \sum_{i,j} \exp(\beta \cdot z_{ij}^c) \right)$$



# Class activation maps



bus



cat



horse



aeroplane



bottle



bicycle



# Class activation maps



bicycle



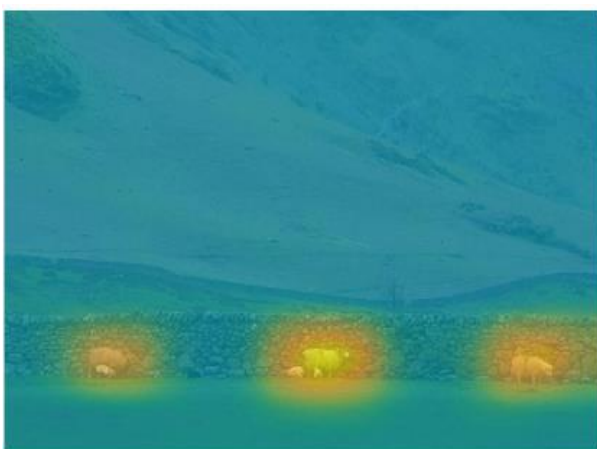
bird



motorbike



person



sheep



bird



# Class activation maps



cow



motorbike



horse



person

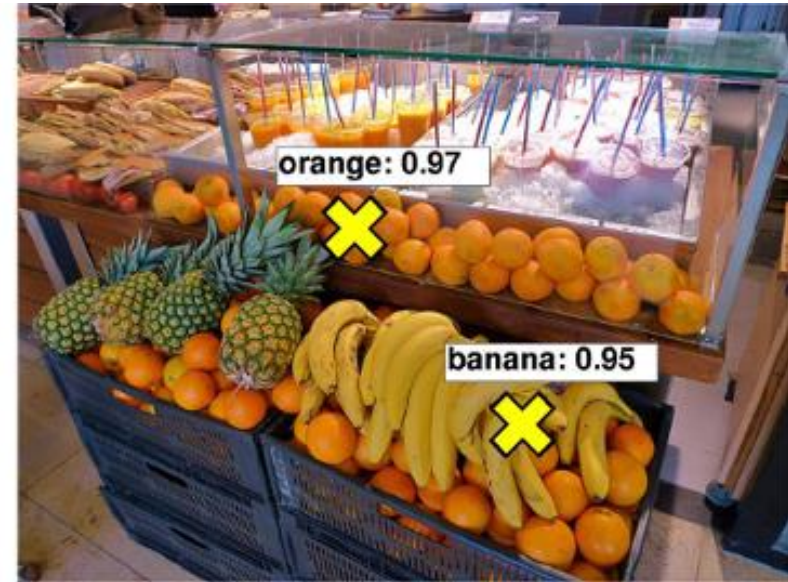
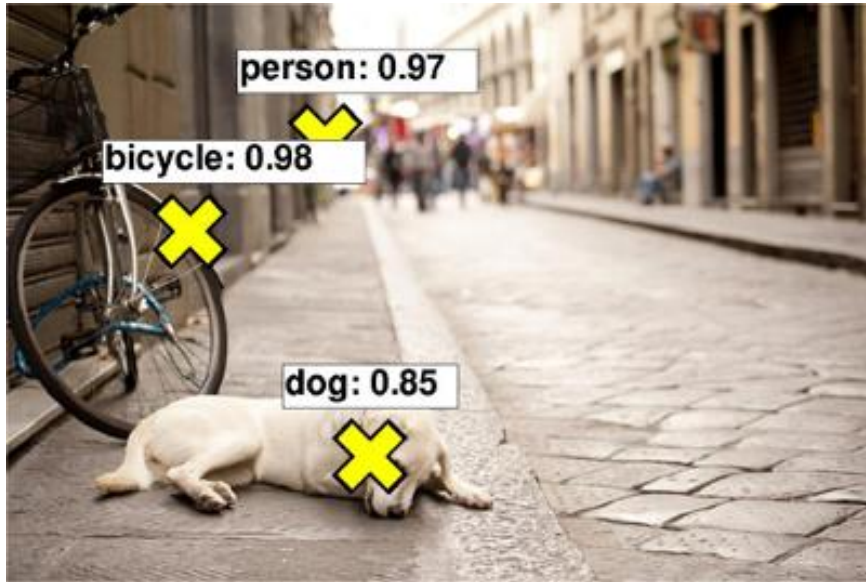


car



person

# Visual recognition task: localization



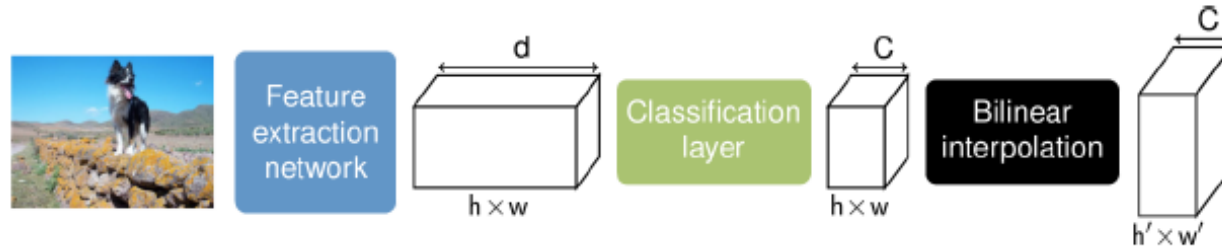
Method	VOC 2012	MS COCO
Deep MIL	74.5	41.2
ProNet	77.7	46.4
WSLocalization	79.7	49.2

# Segmentation

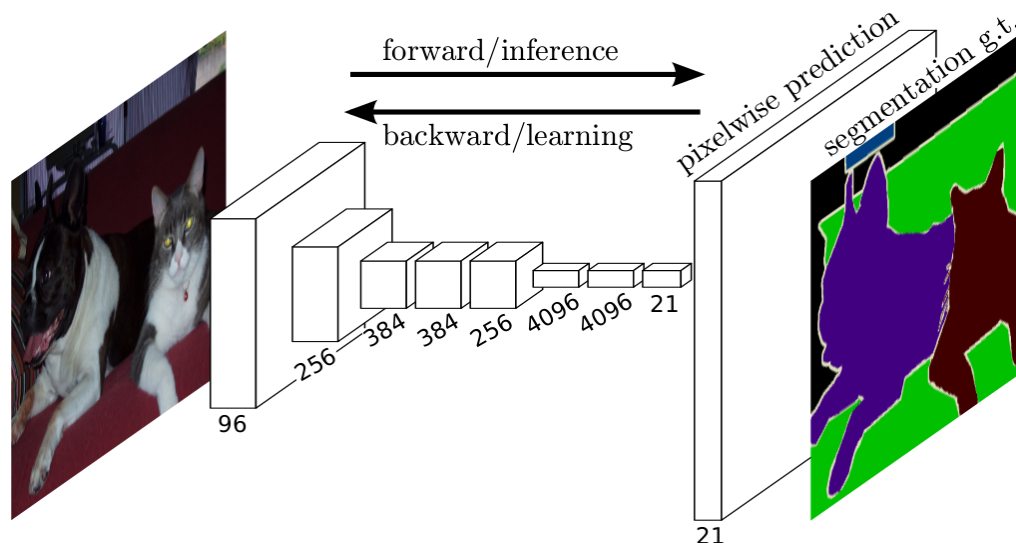
1. Definition
2. Beyond ImageNet: Fully Convolutional Networks (FCNs)
3. **Supervised Segmentation with Deep ConvNets**

# Supervised Segmentation with Deep ConvNets

## F-CN Fully Convolutional Network



- Fully-convolutional network: classify each “pixel”
- Upsampling output
- Network architecture: AlexNet, VGG16, GoogleNet
- Loss: soft-max per pixel

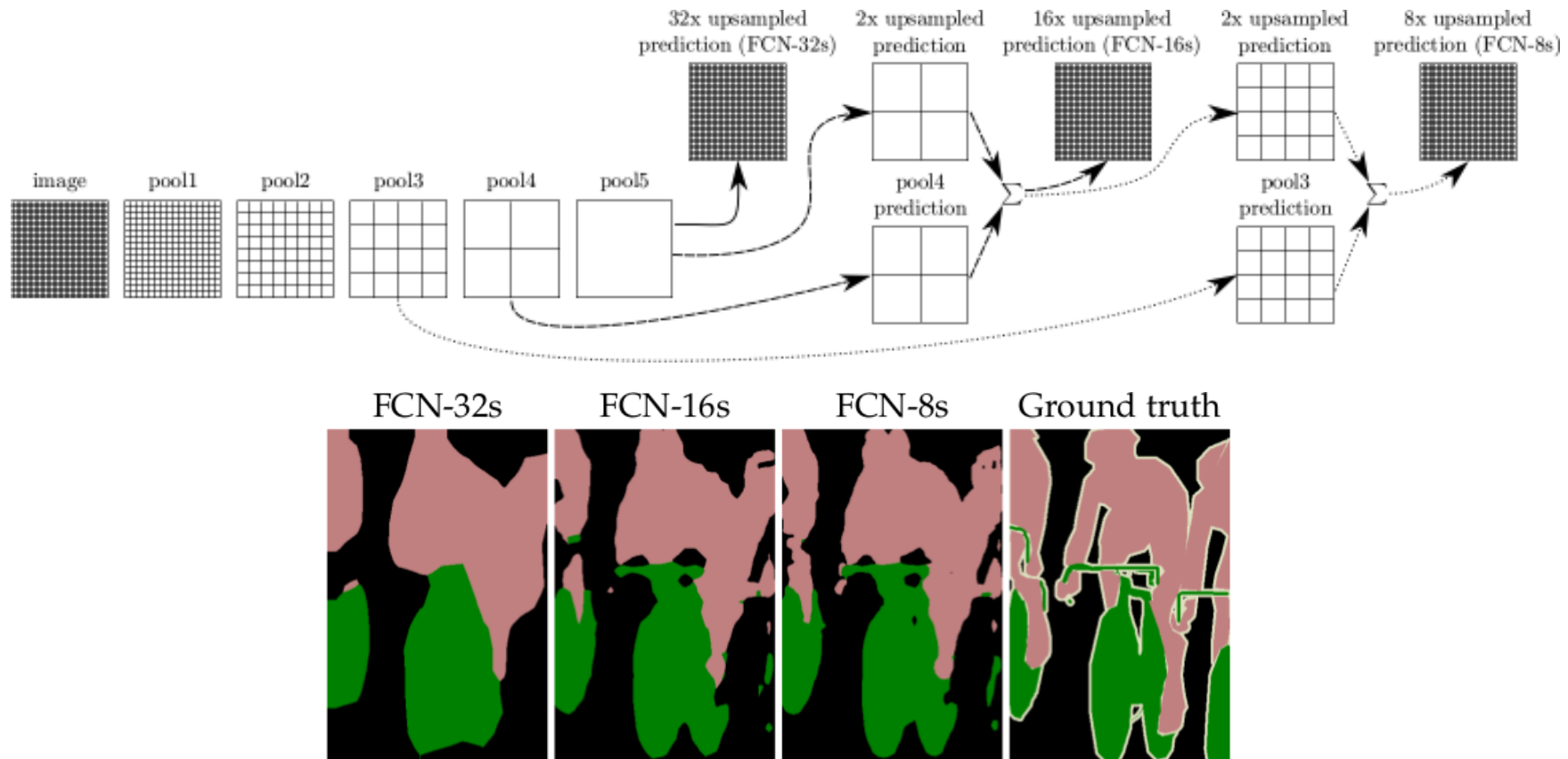


# Supervised Segmentation with Deep ConvNets

## F-CN Fully Convolutional Network

Solution of the FCN approach

- Problem: max pooling and striding reduces spatial resolution
- Dense prediction: combines feature hierarchies
- Initialized with the parameters of coarse net
- Fine-tuning all layers by backpropagation





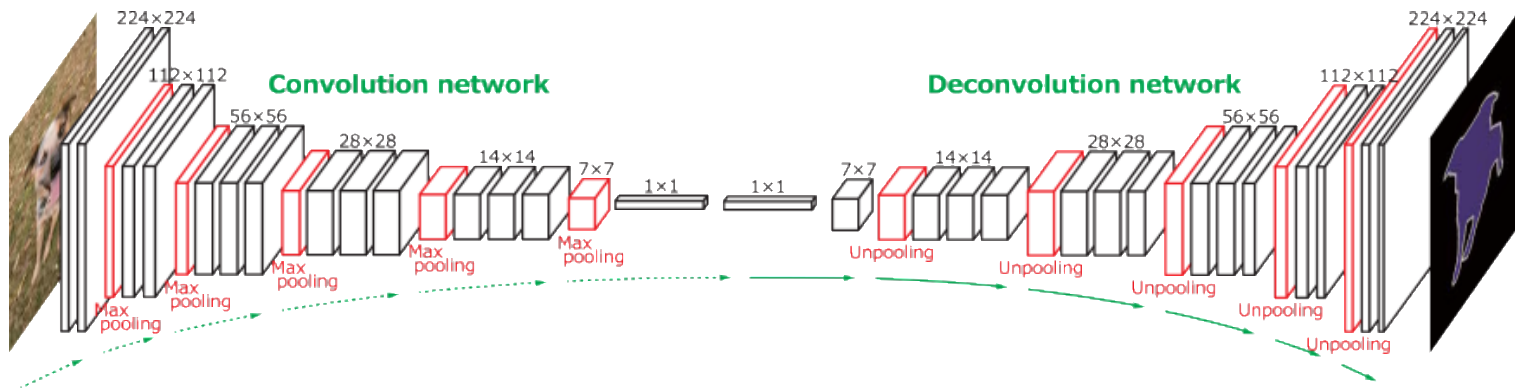
# Supervised Segmentation with Deep ConvNets

1. F-CN Fully Convolutional Network
- 2. Deconvolution Networks**
3. A-trous solution: DeepLab approach

# Supervised Segmentation with Deep ConvNets

## Deconvolution Network

- Learn a multi-layer deconvolution network
- Network is composed of two parts:
  1. Convolution: feature extractor
  2. Deconvolution: shape generator that produces object segmentation from the feature extracted
- Deconvolution net is a mirrored version of the convolution net

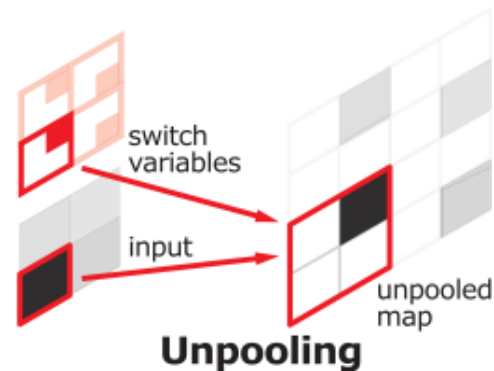
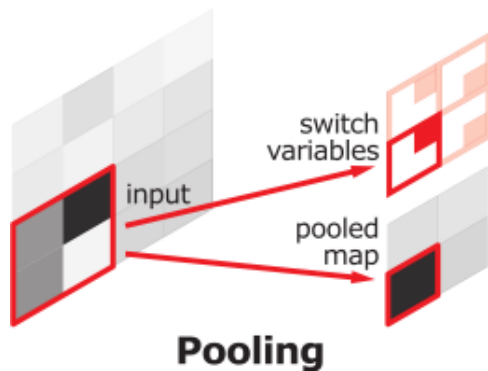


# Supervised Segmentation with Deep ConvNets

## Deconvolution Network

### Unpooling

- Perform the reverse operation of pooling
- Reconstruct the original size of activations
- Useful to reconstruct the structure of input object
- Output: sparse activation map

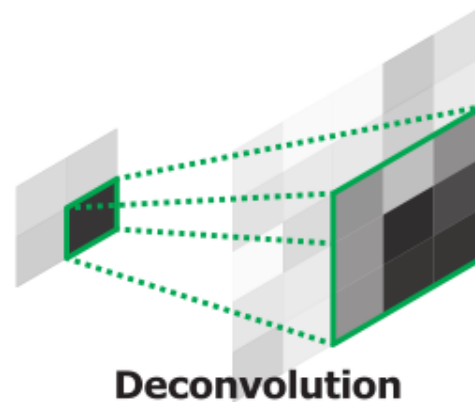
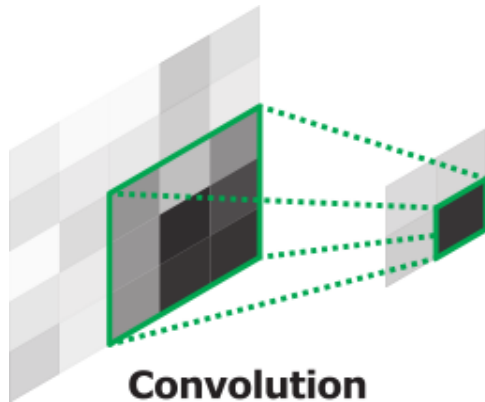


# Supervised Segmentation with Deep ConvNets

## Deconvolution Network

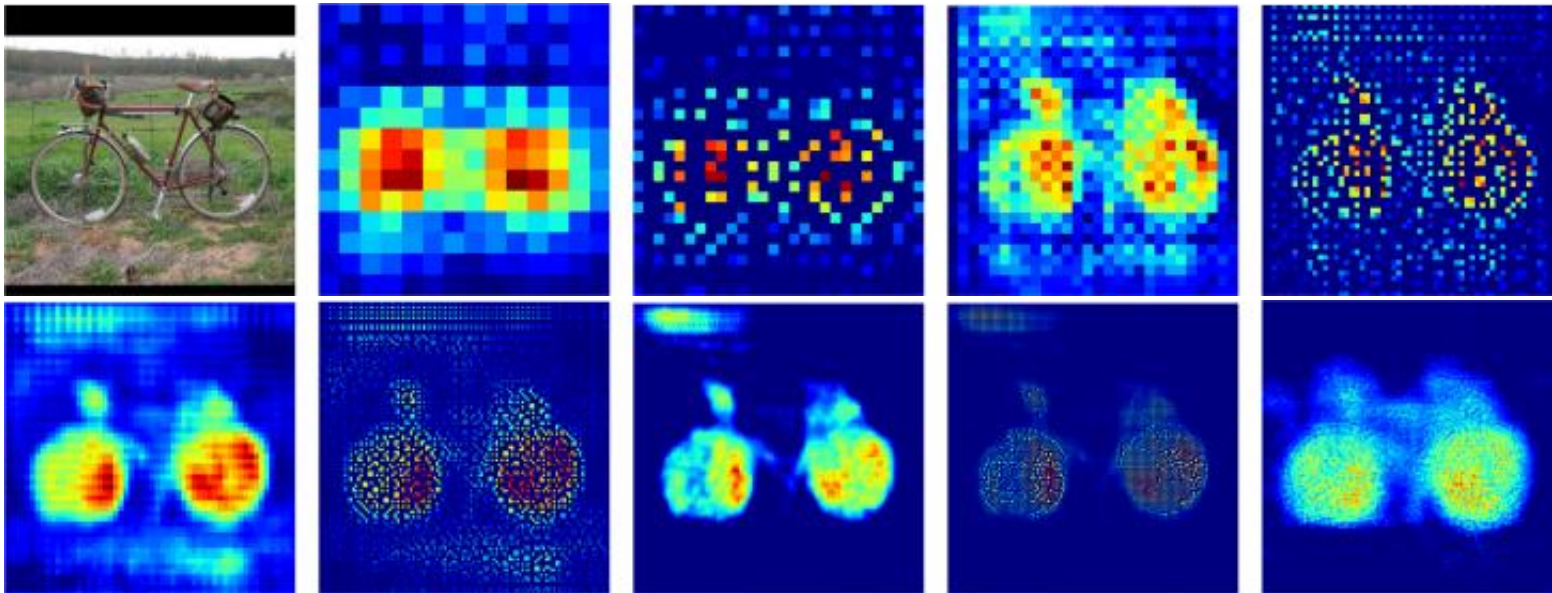
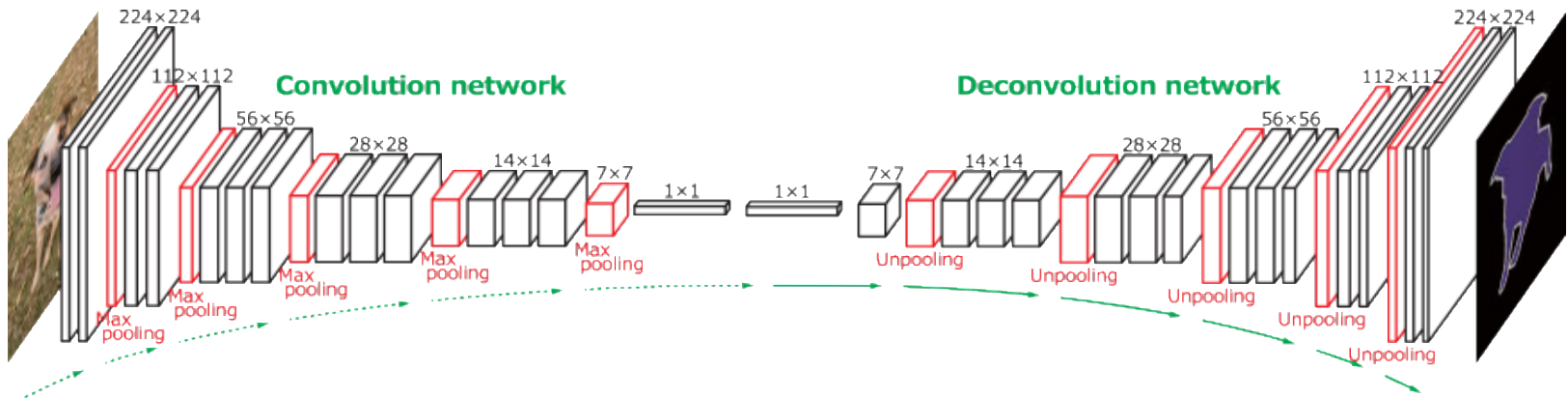
### Deconvolution

- Connect single input activation to a multiple activations
- Learned filters correspond to bases to reconstruct shape of an input object
- Output: enlarged and dense activation map



# Supervised Segmentation with Deep ConvNets

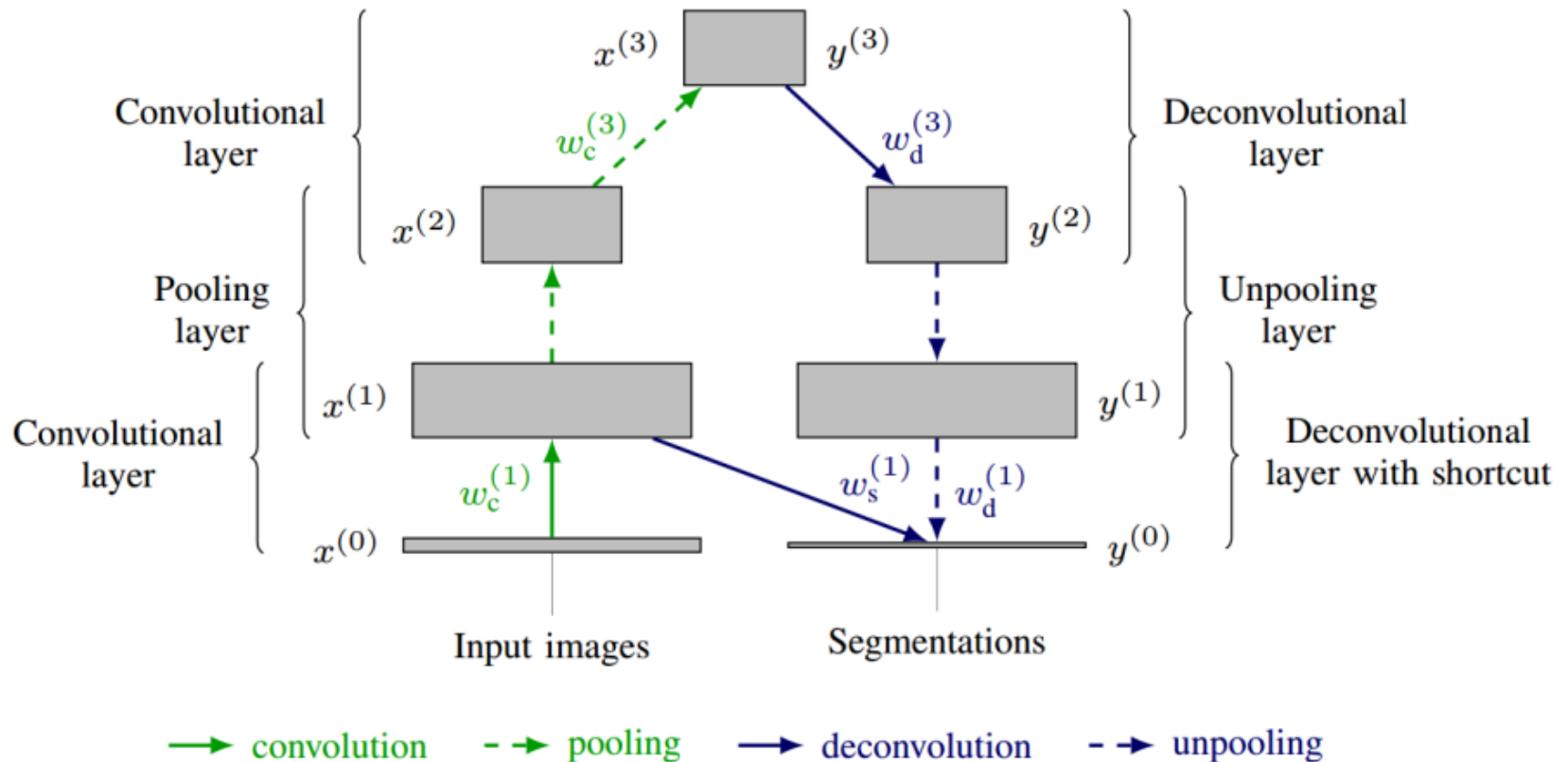
Deconvolution Net: hourglass structure + unpooling switch variables





# Supervised Segmentation with Deep ConvNets

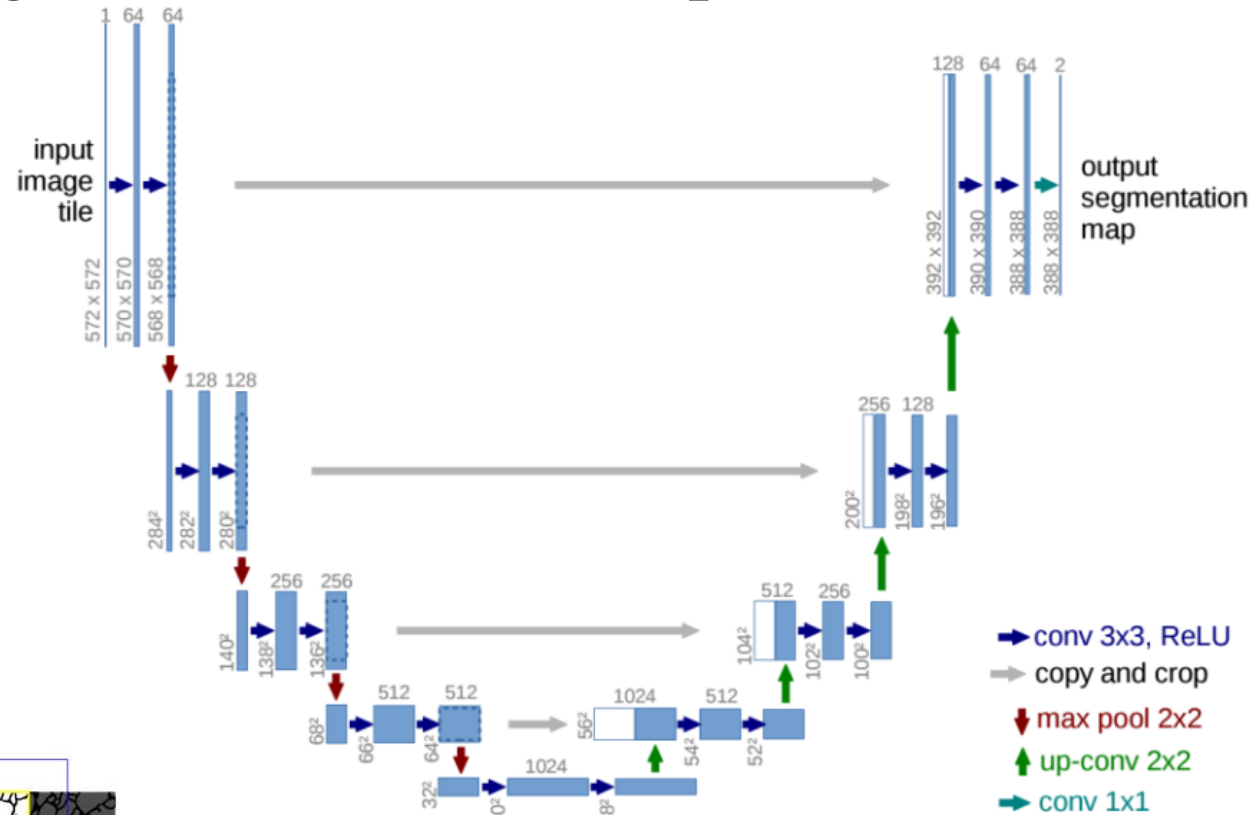
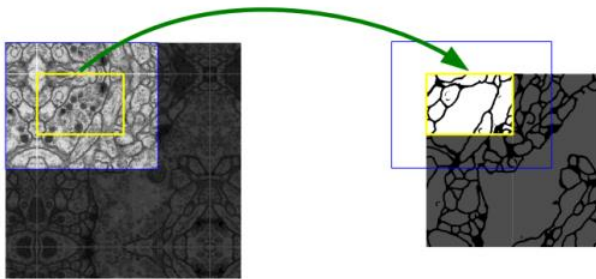
Deconvolution Net + **shortcut connection**



# Supervised Segmentation with Deep ConvNets

U-Net:  
Hourglass/U shape Net  
+ **shortcut connection**  
by feature copies

Very popular in medical  
Works well with low  
training data



**Fig. 1.** U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

**Fig. 2.** Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

# Supervised Segmentation with Deep ConvNets

1. F-CN Fully Convolutional Network
2. Deconvolution Networks
3. **A-trous solution: DeepLab approach**

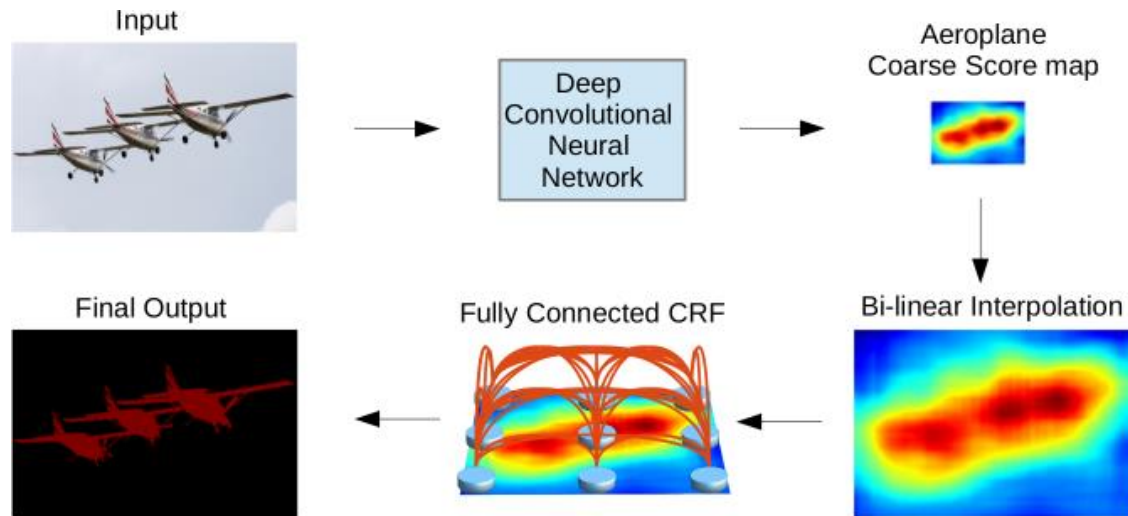
# Supervised Segmentation with Deep ConvNets

## DeepLab (v123) approach for supervised segmentation

Problem of the spatial resolution reduction

Solution of the  
DeepLab  
approach

1. Learn CNN for dense prediction tasks (Atrous)
2. Improve the localization of object boundaries with fully-connected CRF [?] (FC-CRF)



# Supervised Segmentation with Deep ConvNets

## DeepLab approach: **Atrous filtering** algo

- Remove the down-sampling from the last pooling layers.
- Up-sample the original filter by a factor of the **strides**:

Atrous convolution for 1-D signal:

$$y[i] = \sum_{k=1}^K x[i + r \cdot k] w[k]$$


$x[i]$  1-D input signal

$w[k]$  filter of length  $K$

$r$  *rate* parameter corresponds to the stride with which we sample the input signal.

$y[i]$  output of atrous convolution.

Introduce zeros between filter values

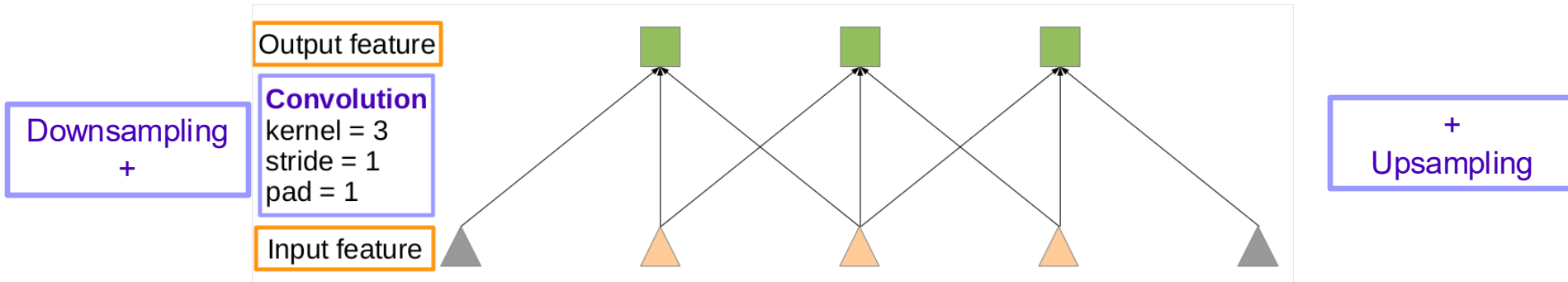


- Note: standard convolution is a special case for *rate*  $r=1$ .

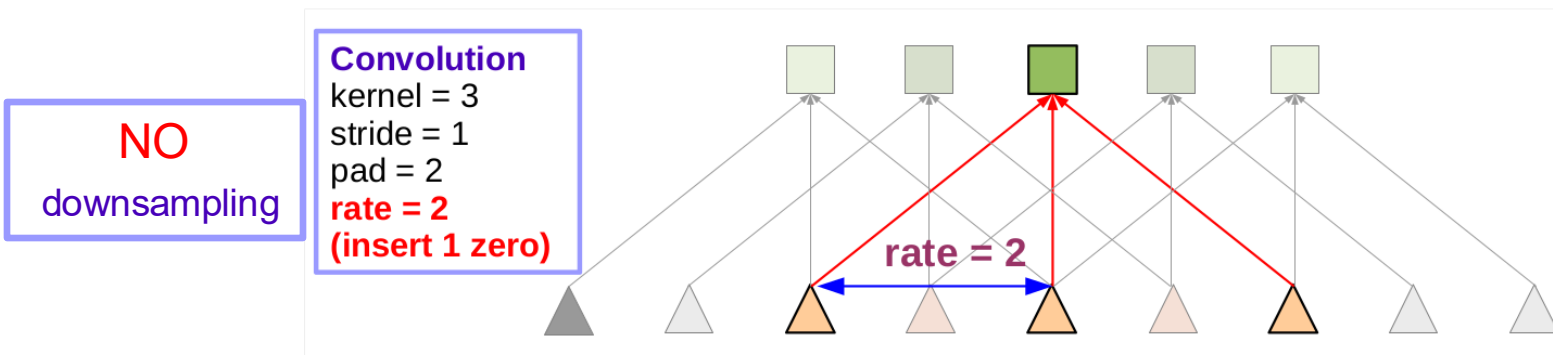


# Supervised Segmentation with Deep ConvNets

## Classical filtering/pooling/downsampling



## DeepLab approach: Atrous filtering algo



# Supervised Segmentation with Deep ConvNets

DeepLab approach: **Atrous filtering** algo

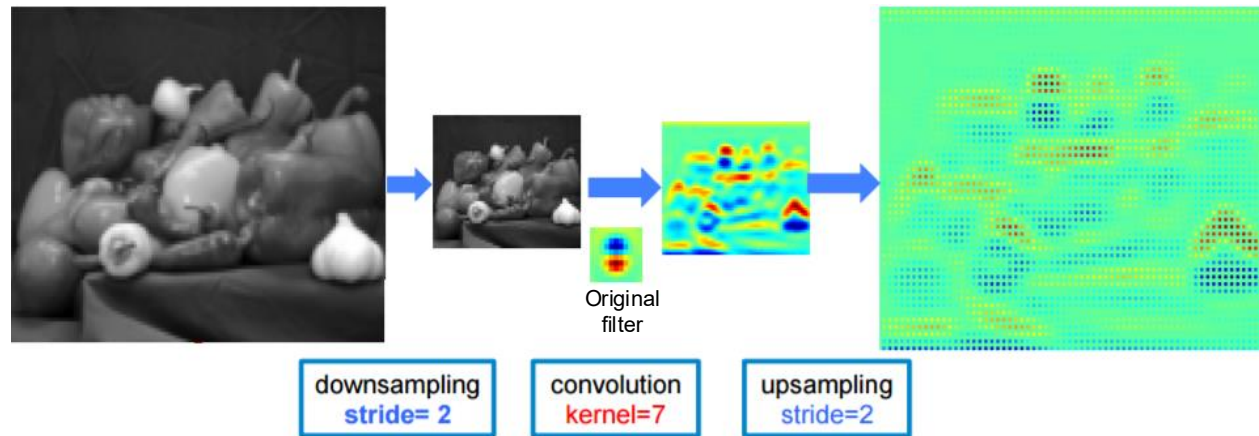
## Filters field-of-view

- **Small** field-of-view → accurate **localization**
- **Large** field-of-view → **context** assimilation
- ‘Holes’: Introduce zeros between filter values.
- **Effective filter size increases** (enlarge the **field-of-view** of filter):  $k \times k$  filter to  $k_e = k + (k - 1)(r - 1)$
- However, we take into account **only** the **non-zero** filter values:
  - ✓ Number of filter parameters is the same.
  - ✓ Number of operations per position is the same.

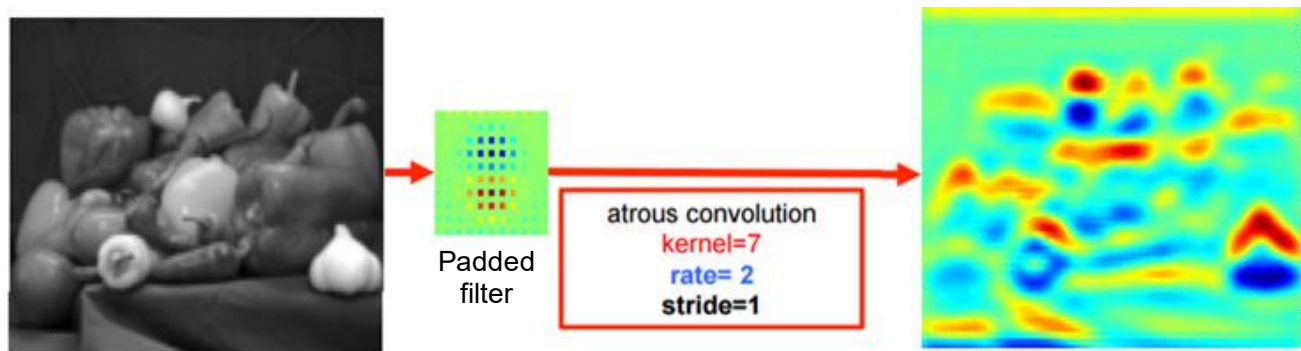
# Supervised Segmentation with Deep ConvNets

DeepLab approach: **Atrous filtering** algo

Standard  
convolution



Atrous  
convolution



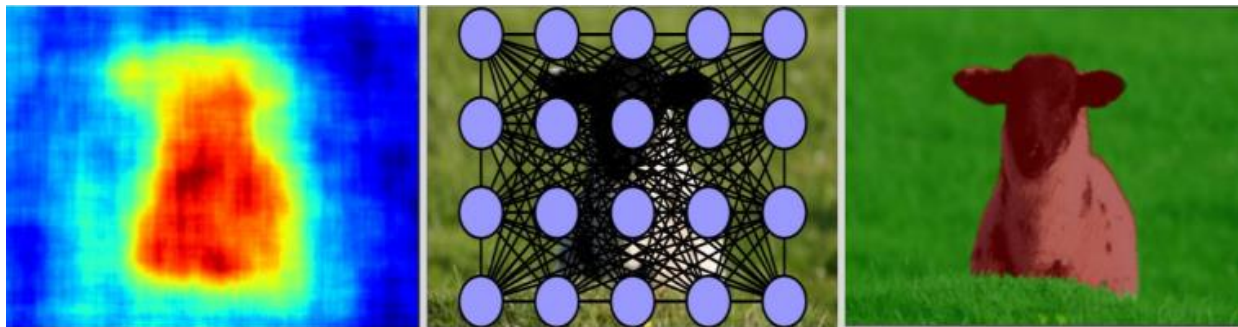
# Supervised Segmentation with Deep ConvNets

## DeepLab: Fully-Connected CRF

- Problem: poor object delineation (spatial and appearance consistency neglected)
- Solution: fully-connected CRF accounts for contextual information in the image

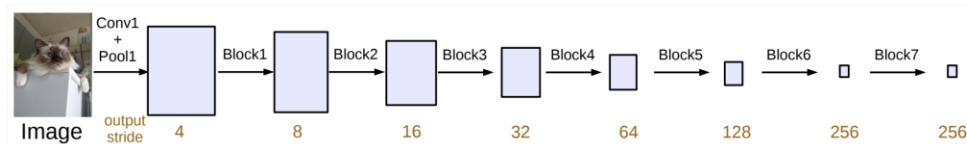
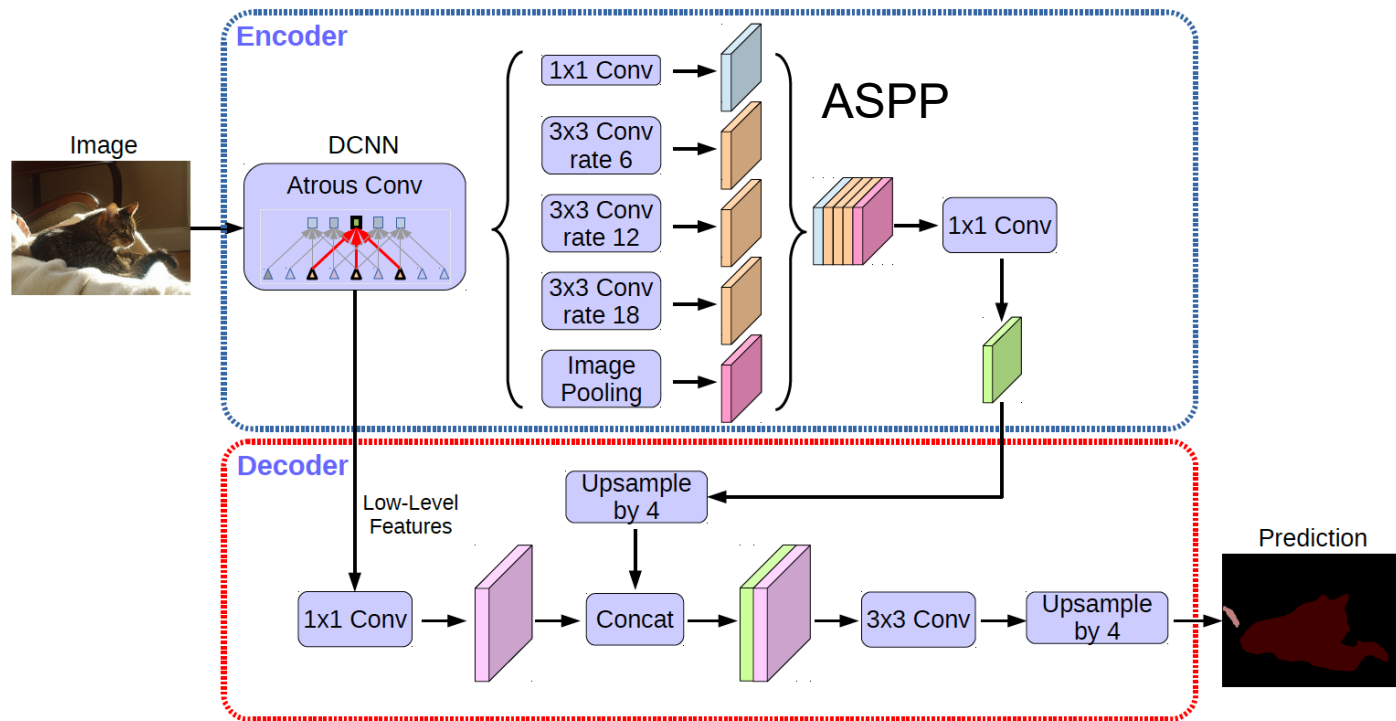
$$E(y) = \sum_i \psi_i(y_i) + \sum_{i,j} \psi_{ij}(y_i, y_j)$$

- | Unary term: output of FCN (upscaled)
- | Pairwise term: penalizes similar pixels having different labels

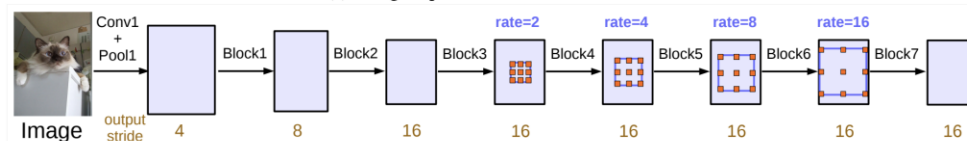


# Supervised Segmentation with Deep ConvNets

- DeepLab V3+ [ECCV 2018]



(a) Going deeper without atrous convolution.



(b) Going deeper with atrous convolution. Atrous convolution with  $rate > 1$  is applied after block3 when  $output\_stride = 16$ .

Figure 3. Cascaded modules without and with atrous convolution.

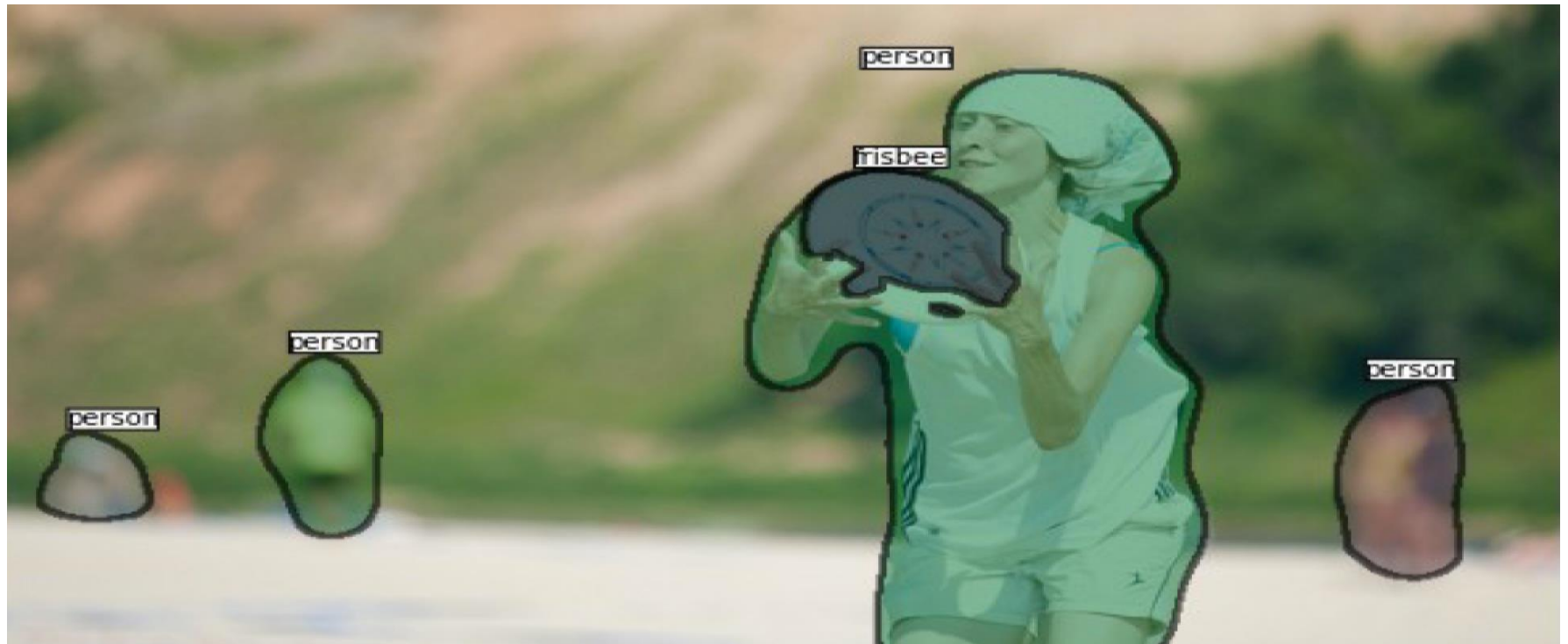


# Segmentation Results



@Y. LeCun

# Segmentation Results



@Y. LeCun