

DEEP LEARNING FOR COMPUTER VISION

Summer Seminar UPC TelecomBCN, 4 - 8 July 2016



Instructors



Xavier
Giró-i-Nieto



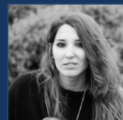
Elisa
Sayrol



Amaia
Salvador



Jordi
Torres



Eva
Mohedano



Kevin
McGuinness

Organizers



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación



Dublin City University
Oileán Chathair Bhaile Átha Cliath



Centre for Data Analytics



GPU
CENTER OF
EXCELLENCE

Co-funded by the
Erasmus+ Programme
of the European Union



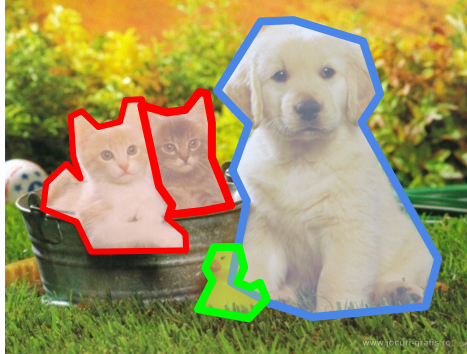
+ info: TelecomBCN.DeepLearning.Barcelona

Day 4 Lecture 2

Segmentation

Segmentation

Segmentation



Define the accurate boundaries of all objects in an image

Segmentation: Datasets



Pascal Visual Object Classes
20 Classes
~ 5.000 images



Microsoft COCO
80 Classes
~ 300.000 images

Semantic Segmentation

Label every pixel!

Don't differentiate instances (cows)

Classic computer vision problem

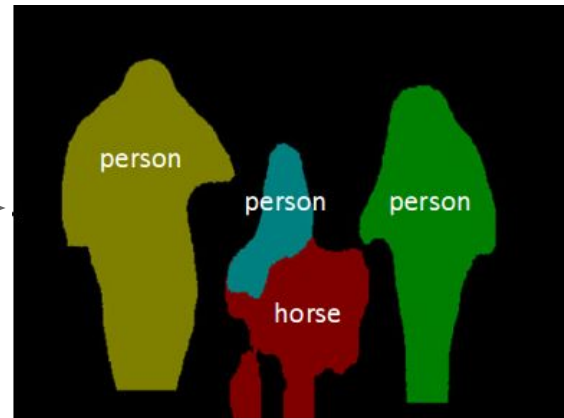


object classes	building	grass	tree	cow	sheep	sky	airplane	water	face	car
bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat

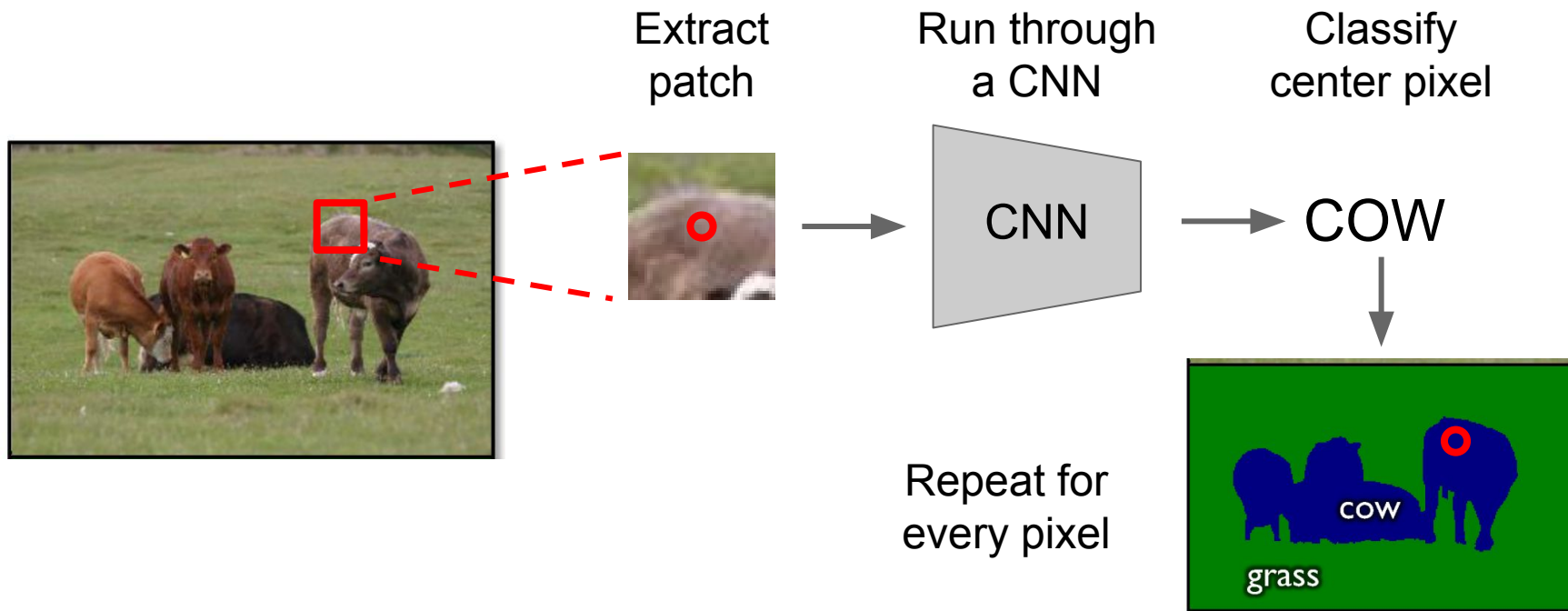
Instance Segmentation

Detect instances,
give category, label
pixels

“simultaneous
detection and
segmentation” (SDS)

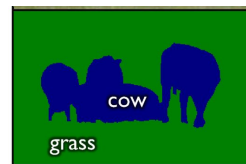
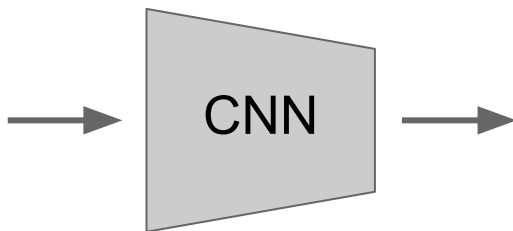


Semantic Segmentation



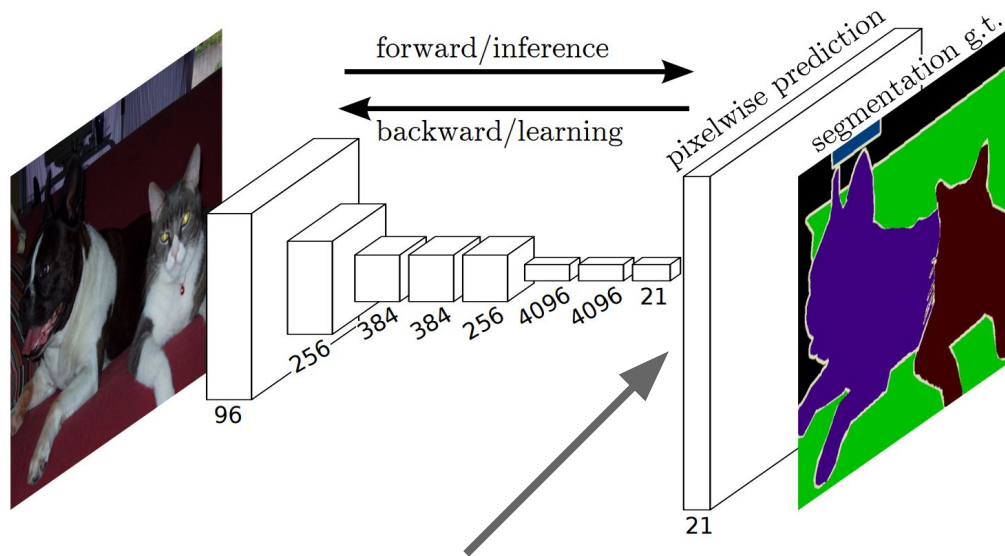
Semantic Segmentation

Run “fully convolutional” network
to get all pixels at once



Smaller output
due to pooling

Semantic Segmentation



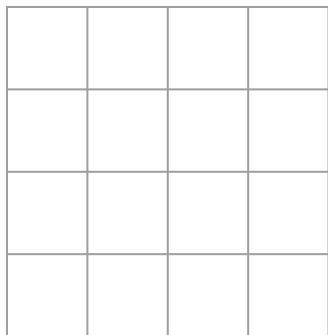
Learnable upsampling!

Long et al. [Fully Convolutional Networks for Semantic Segmentation](#). CVPR 2015

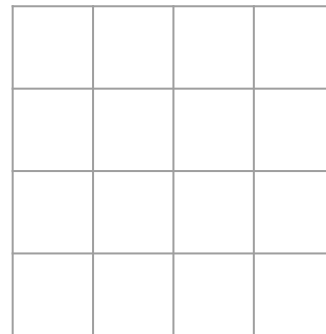
Slide Credit: [CS231n](#)

Convolutional Layer

Typical 3 x 3 convolution, stride 1 pad 1



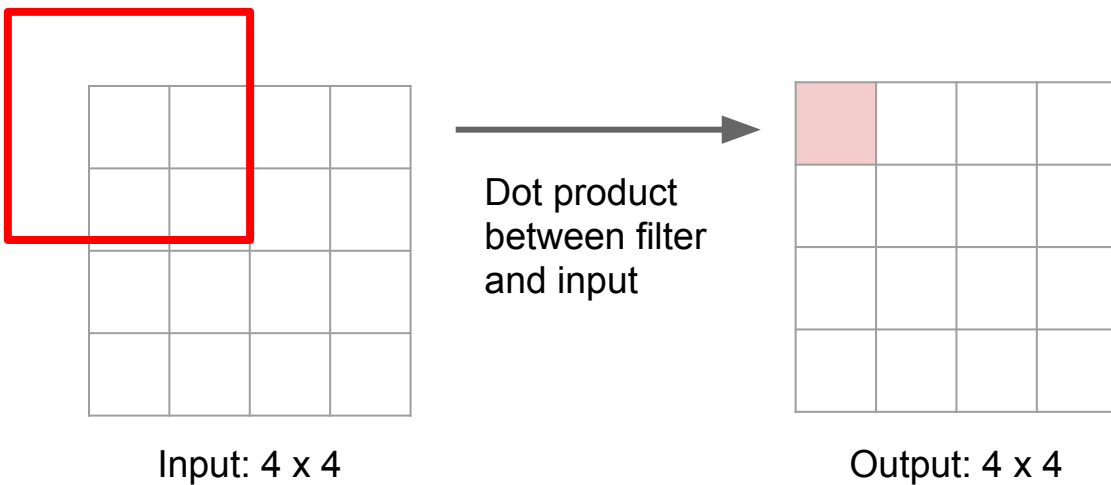
Input: 4 x 4



Output: 4 x 4

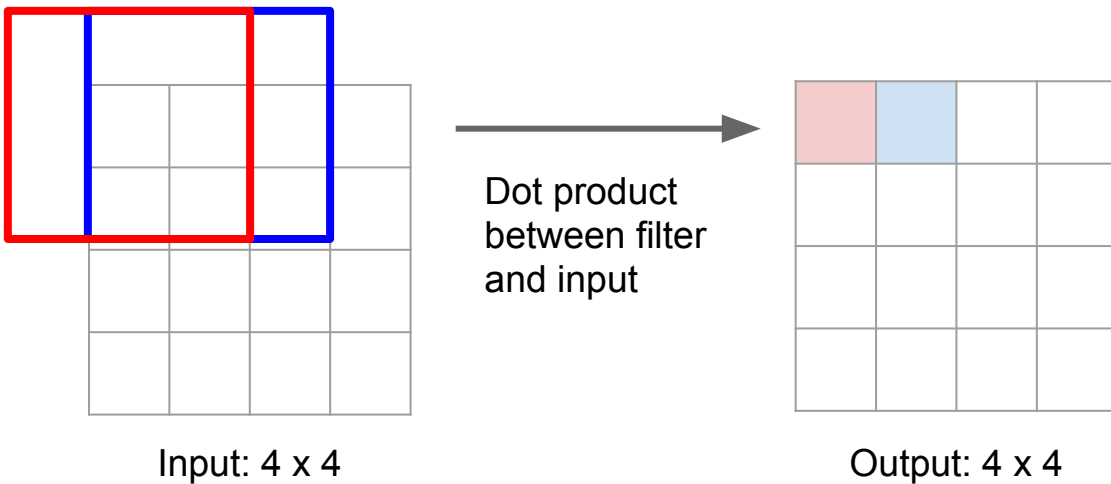
Convolutional Layer

Typical 3 x 3 convolution, stride 1 pad 1



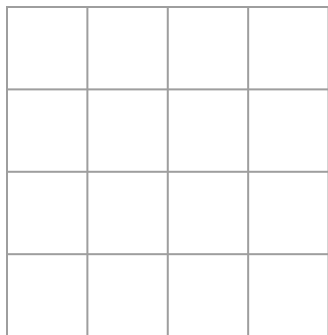
Convolutional Layer

Typical 3 x 3 convolution, stride 1 pad 1

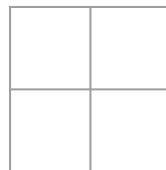


Convolutional Layer

Typical 3 x 3 convolution, **stride 2** pad 1



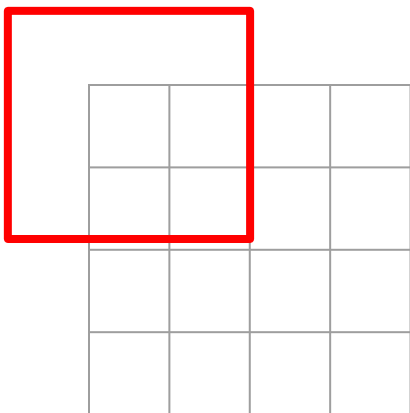
Input: 4 x 4



Output: 2 x 2

Convolutional Layer

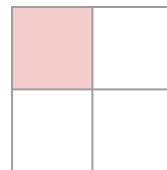
Typical 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4



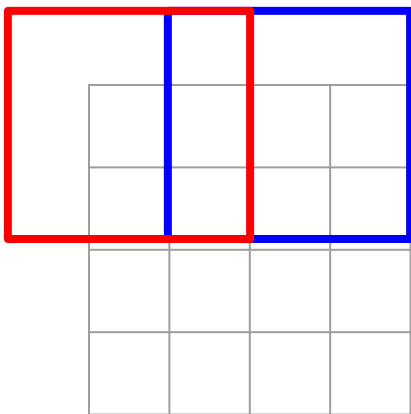
Dot product
between filter
and input



Output: 2 x 2

Convolutional Layer

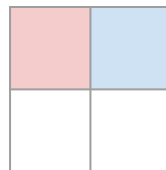
Typical 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4



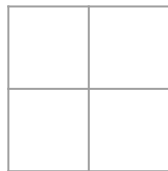
Dot product
between filter
and input



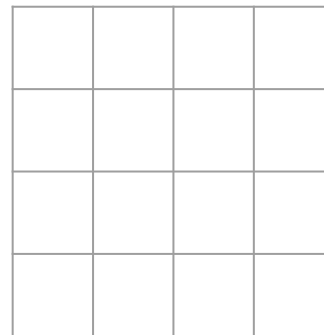
Output: 2 x 2

Deconvolutional Layer

3 x 3 “deconvolution”, stride 2 pad 1



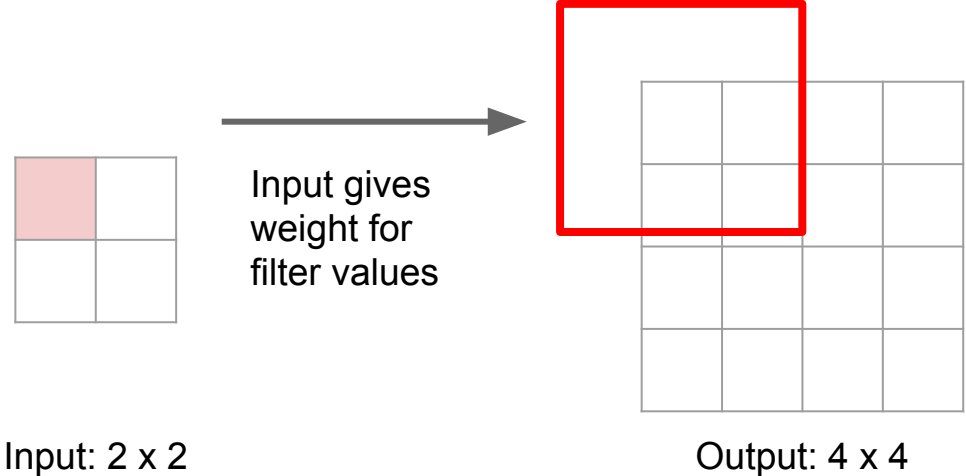
Input: 2 x 2



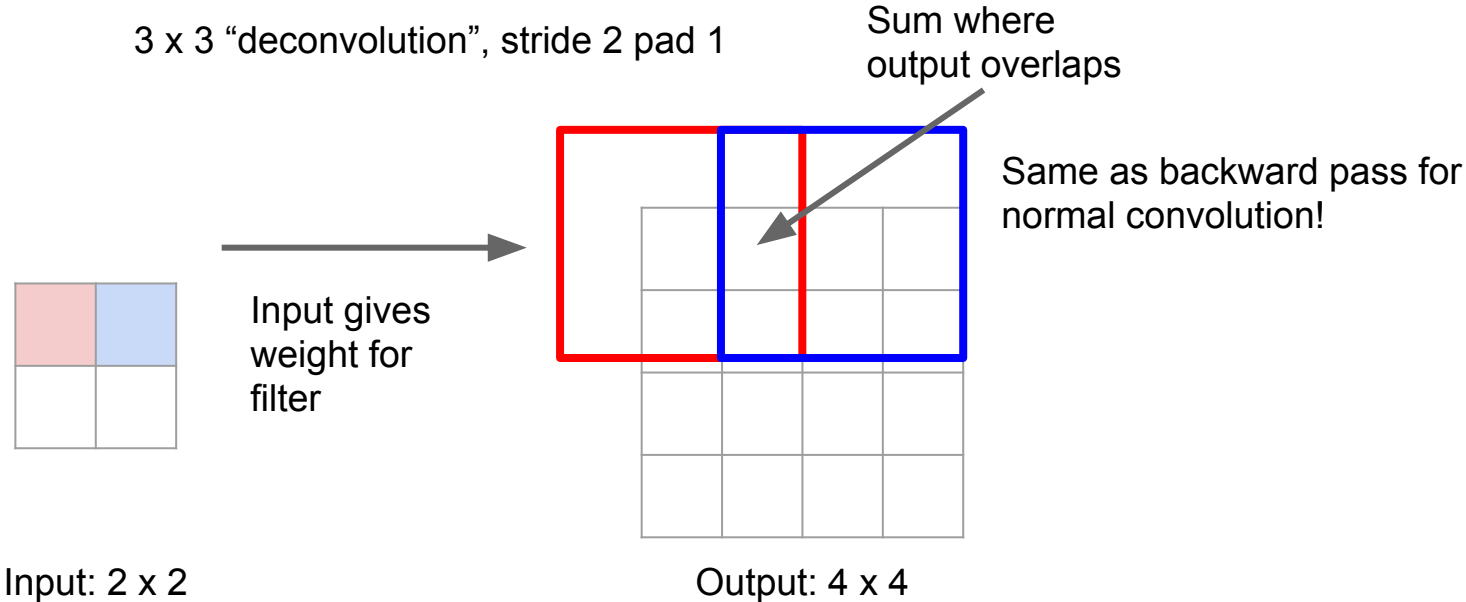
Output: 4 x 4

Deconvolutional Layer

3 x 3 “deconvolution”, stride 2 pad 1



Deconvolutional Layer



Deconvolutional Layer

¹It is more proper to say “convolutional transpose operation” rather than “deconvolutional” operation. Hence, we will be using the term “convolutional transpose” from now.

Im et al. [Generating images with recurrent adversarial networks](#). arXiv 2016

A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions)

Radford et al. [Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks](#). ICLR 2016

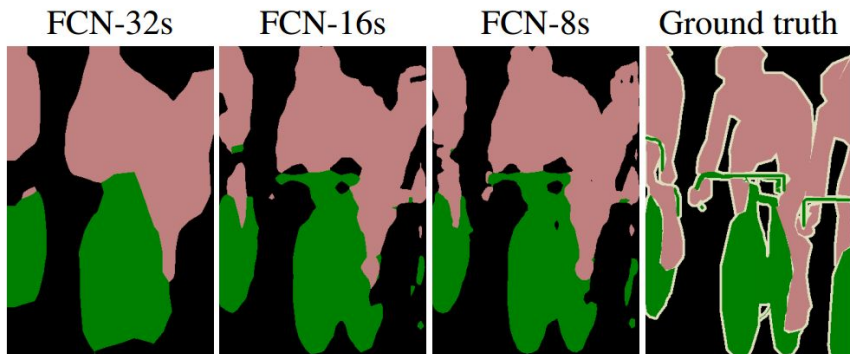
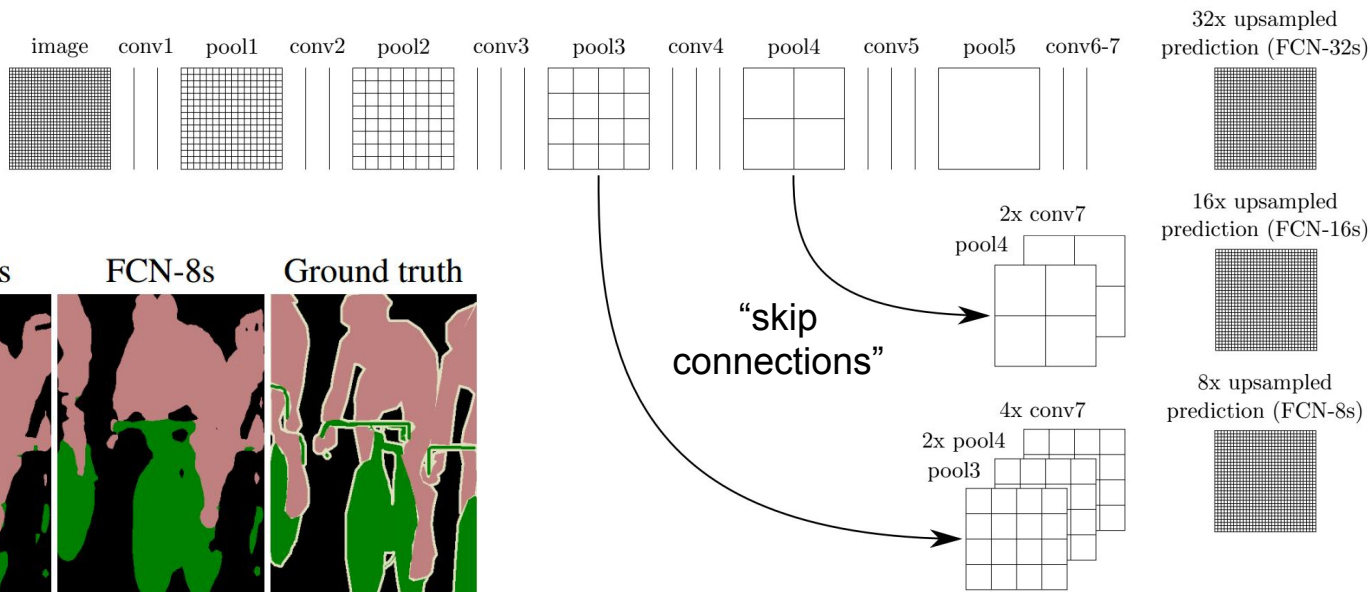
“Deconvolution” is a bad name, already defined as “inverse of convolution”

Better names:

convolution transpose,
backward strided convolution,
1/2 strided convolution, upconvolution

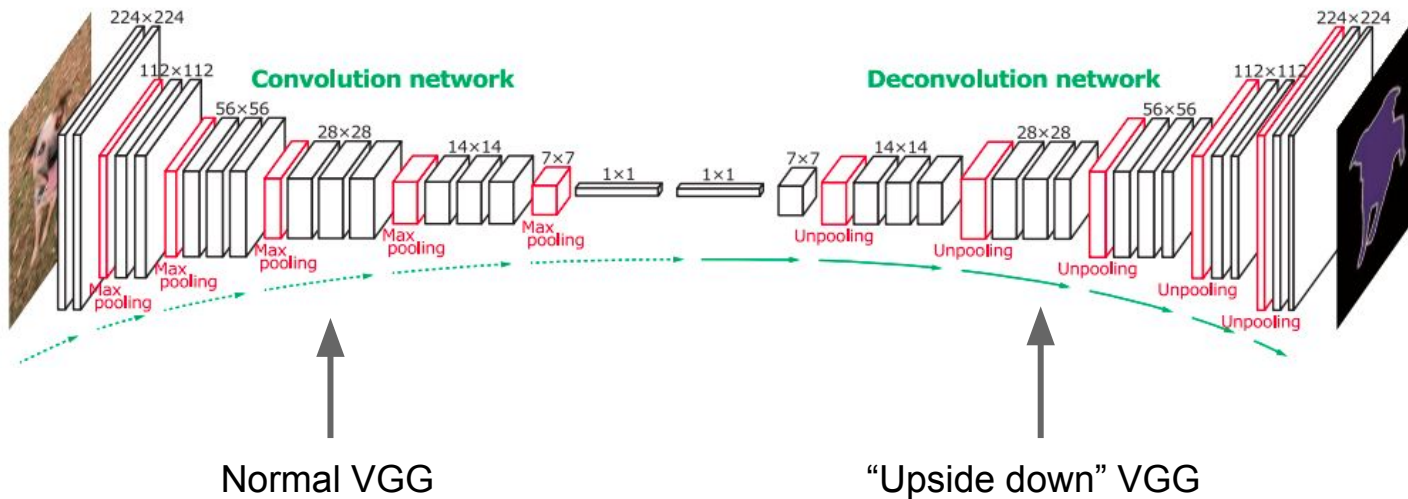
Slide Credit: [CS231n](#)

Skip Connections



Skip connections = Better results

Semantic Segmentation



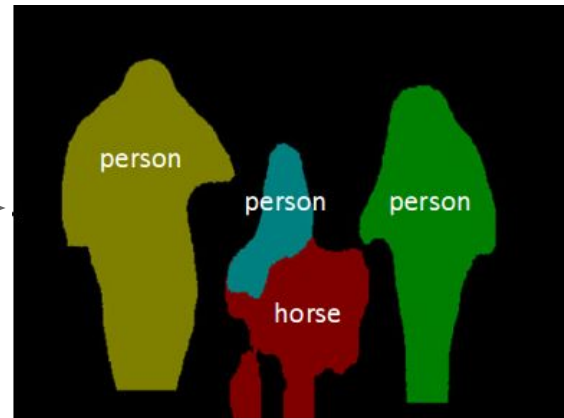
Noh et al. [Learning Deconvolution Network for Semantic Segmentation](#). ICCV 2015

Slide Credit: [CS231n](#)

Instance Segmentation

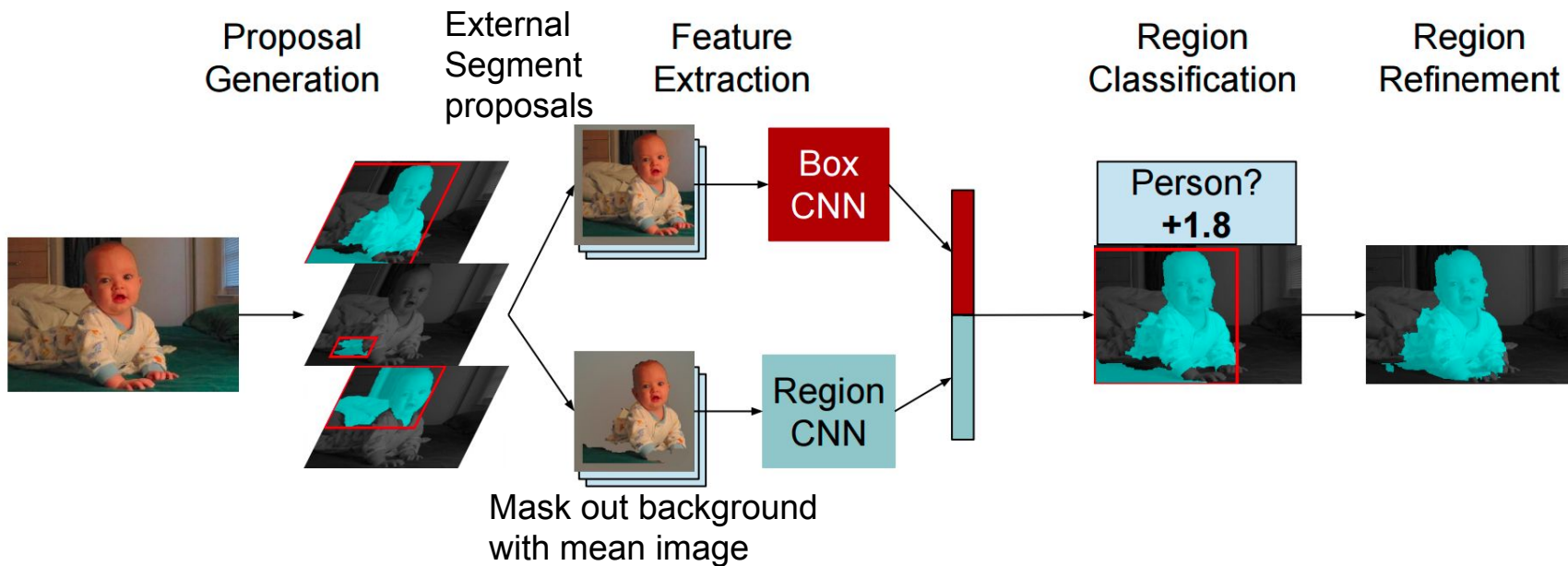
Detect instances,
give category, label
pixels

“simultaneous
detection and
segmentation” (SDS)



Instance Segmentation

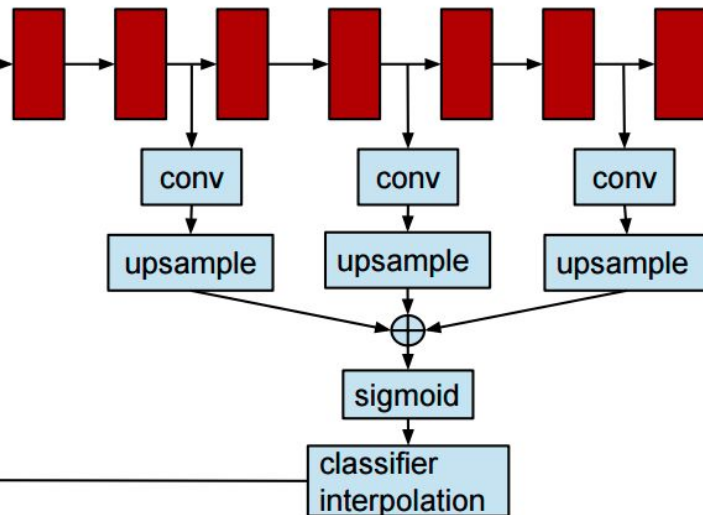
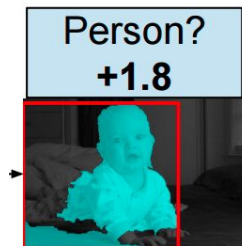
Similar to R-CNN, but with segments



Instance Segmentation

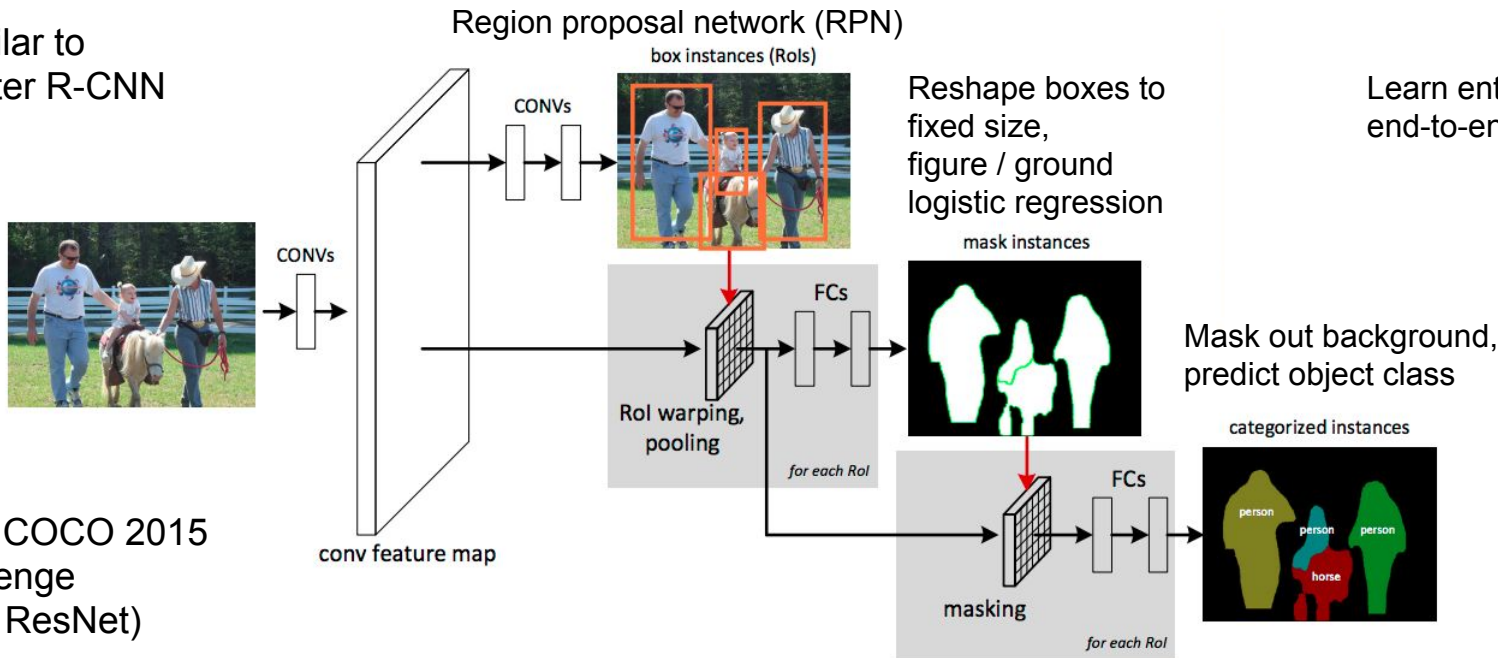
Region
Classification

Region
Refinement



Instance Segmentation

Similar to
Faster R-CNN



Learn entire model
end-to-end!

Won COCO 2015
challenge
(with ResNet)

Dai et al. [Instance-aware Semantic Segmentation via Multi-task Network Cascades](#). arXiv 2015

Slide Credit: [CS231n](#)

Resources

- CS231n Lecture @ Stanford [[slides](#)][[video](#)]
- Code for Semantic Segmentation
 - [FCN](#) (Caffe)
- Code for Instance Segmentation
 - [SDS](#) (Caffe)
 - [SDS using Hypercolumns & sharing conv computations](#) (Caffe)
 - [Instance-aware Semantic Segmentation via Multi-task Network Cascades](#) (Caffe)