Insight
Centre for Data Analytics

DEEP LEARNING WORKSHOP

Dublin City University
27-28 April 2017

DCU

Day 2 Lecture 7

Object Segmentation

Amaia Salvador
amaia.salvador@upc.edu

PhD Candidate
Universitat Politècnica de Catalunya

UPC

Object Segmentation

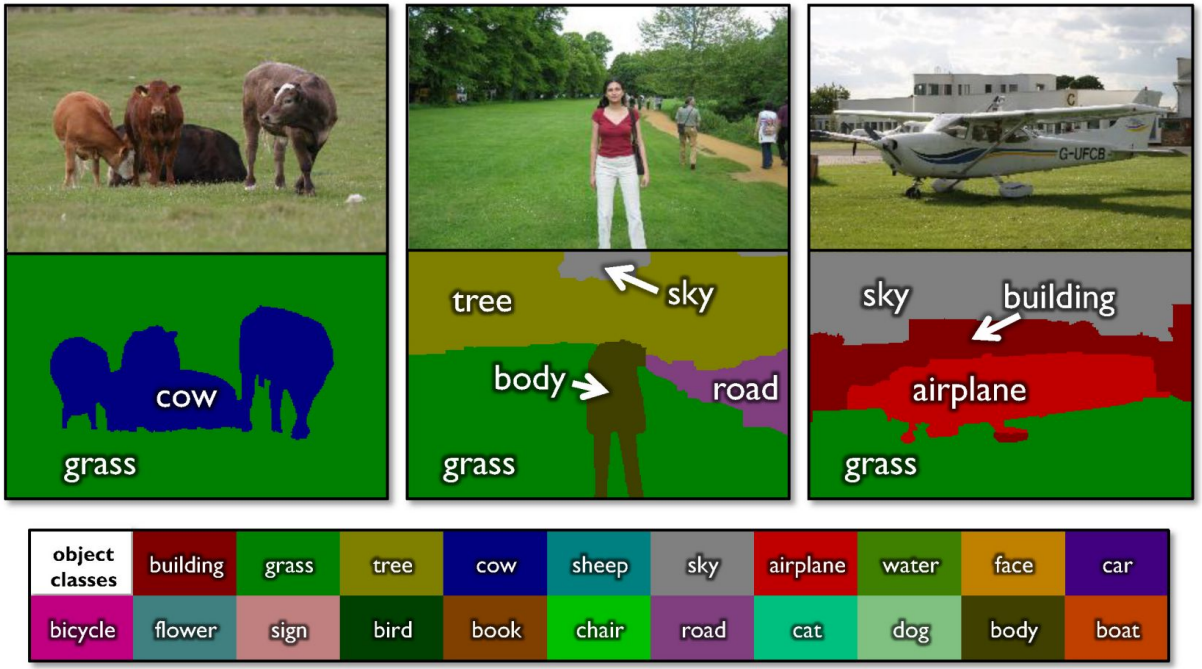
Define the accurate boundaries of all objects in an image

Semantic Segmentation

Label every pixel!

Don’t differentiate instances (cows)

Classic computer vision problem

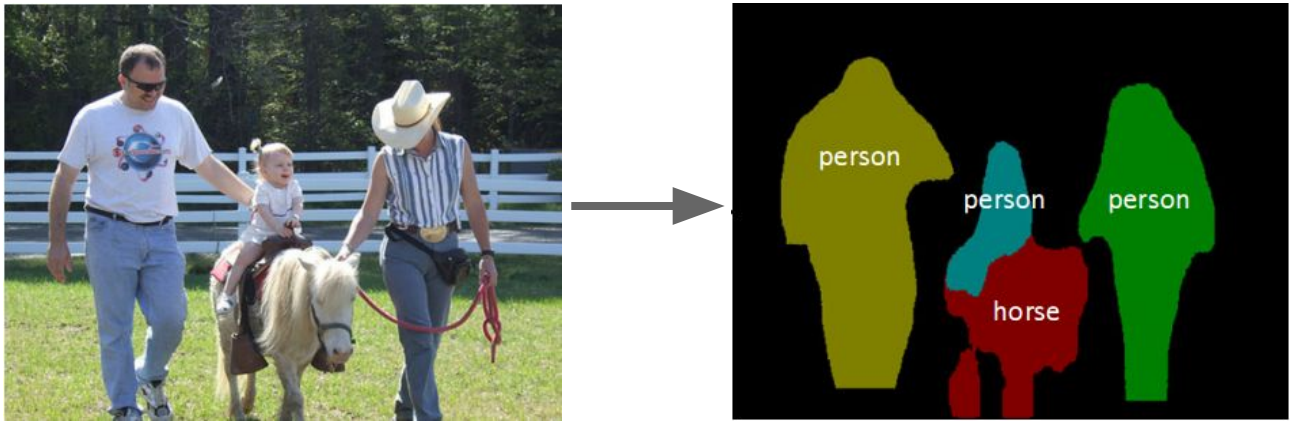


Slide Credit: CS231n

Instance Segmentation

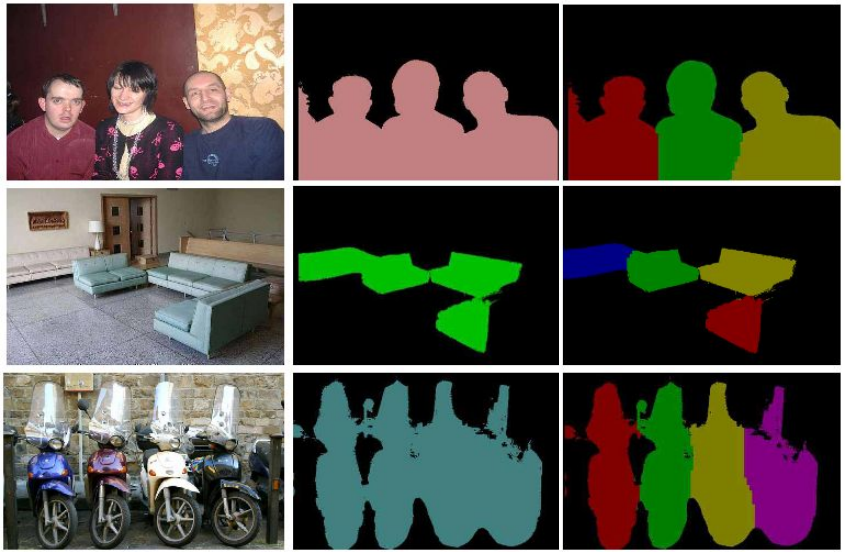
Detect instances, give category, label pixels

“simultaneous detection and segmentation” (SDS)



Slide Credit: CS231n

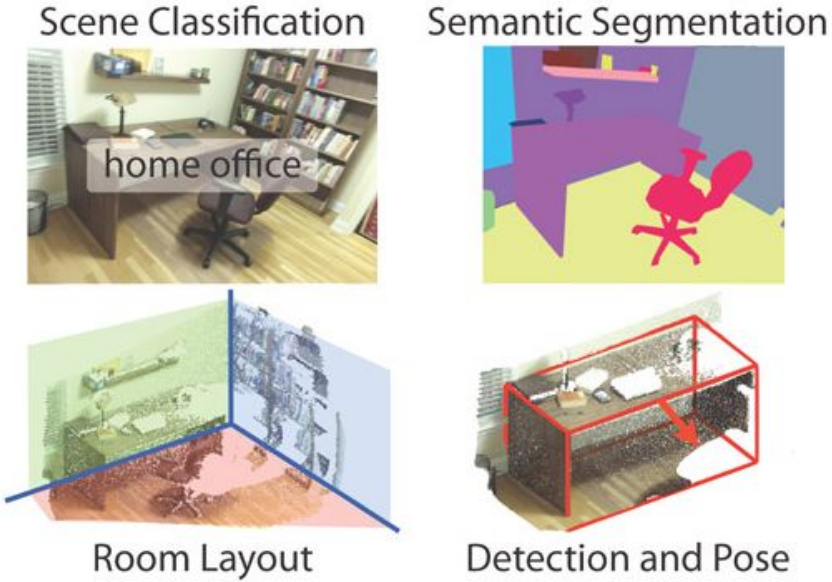
Object Segmentation: Datasets



[Pascal Visual Object Classes]
20 Classes
~ 5.000 images

[Pascal Context]
540 Classes
~ 10.000 images

Object Segmentation: Datasets

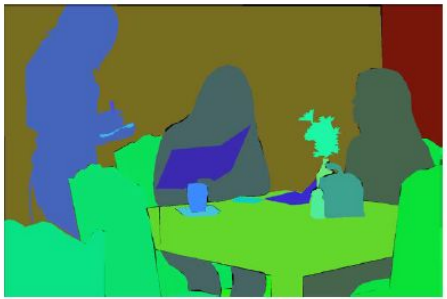


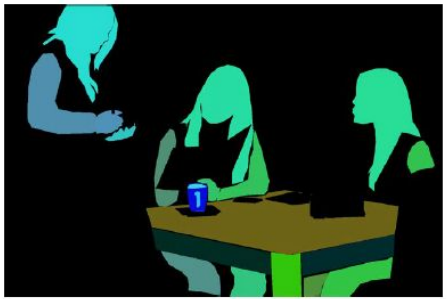
SUN RGB-D
19 Classes
~ 10.000 images

Microsoft COCO
80 Classes
~ 300.000 images

Object Segmentation: Datasets

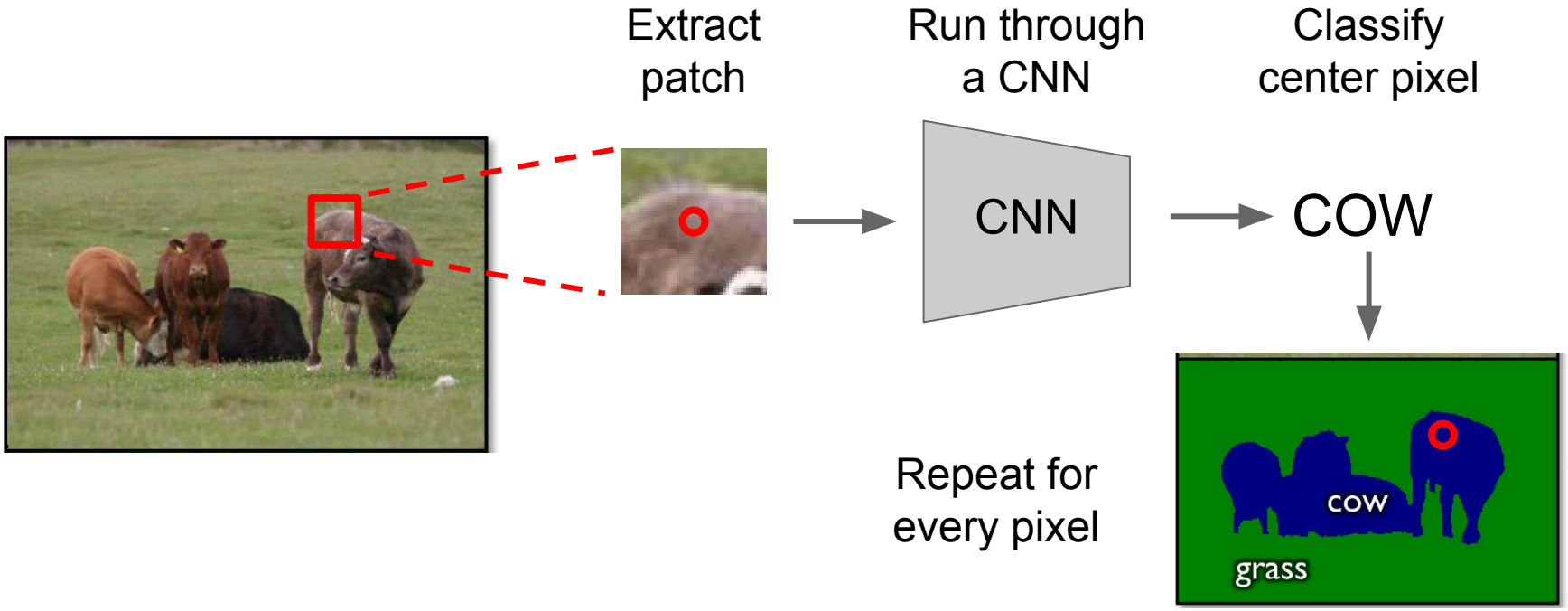
CityScapes
30 Classes
~ 25.000 images





ADE20K
>150 Classes
~ 22.000 images

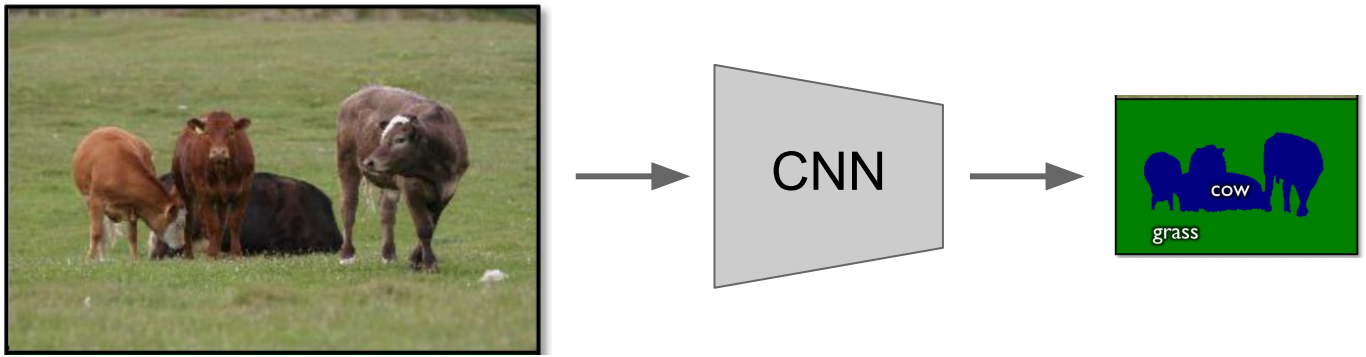
Semantic Segmentation



Slide Credit: CS231n

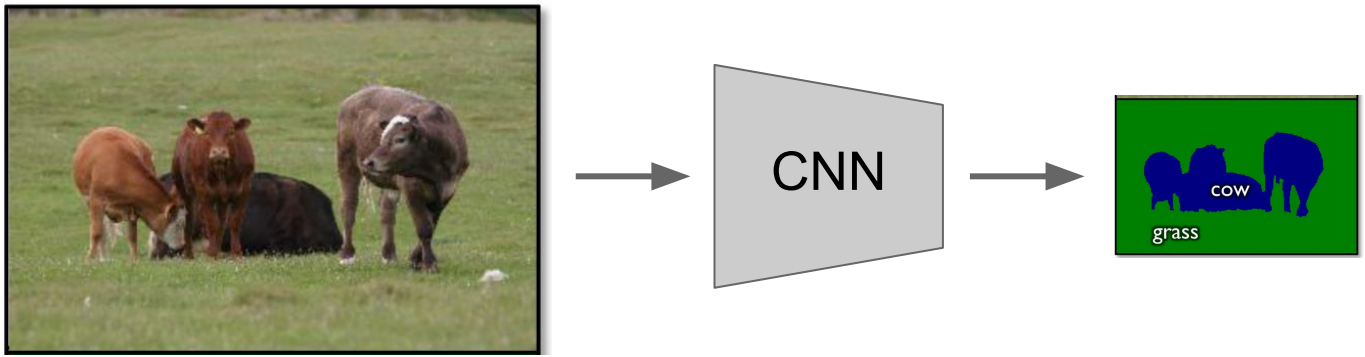
Semantic Segmentation

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



Slide Credit: CS231n

Semantic Segmentation

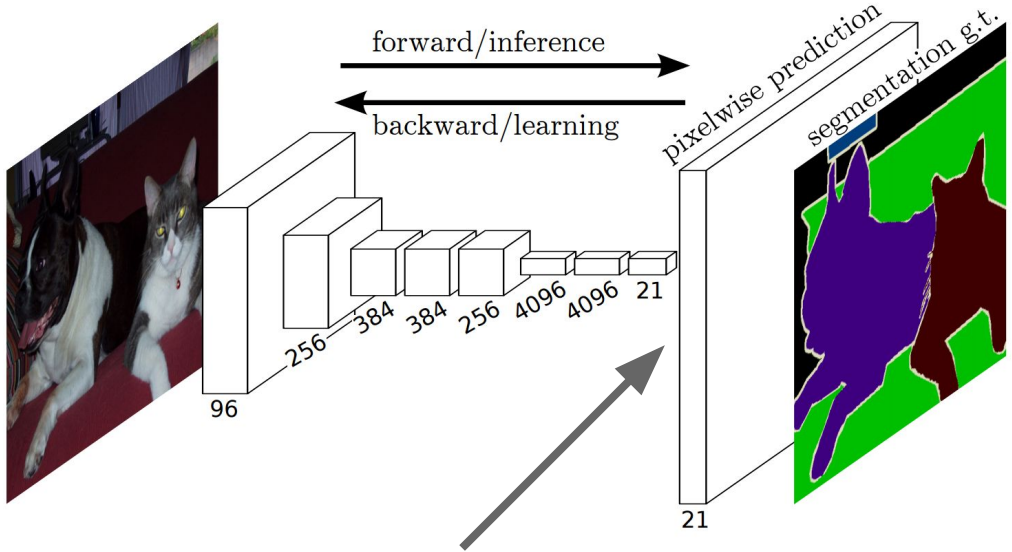


Problem 1:

Smaller output due to pooling

Slide Credit: CS231n

Learnable upsampling

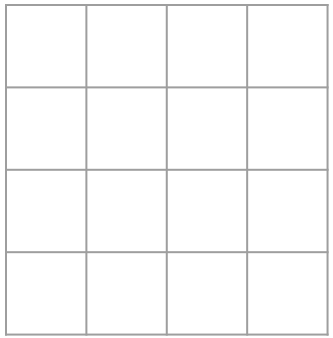


Long et al. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015

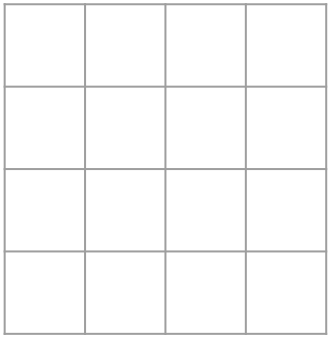
Slide Credit: CS231n

Reminder: Convolutional Layer

Typical 3 x 3 convolution, stride 1 pad 1



Input: 4 x 4

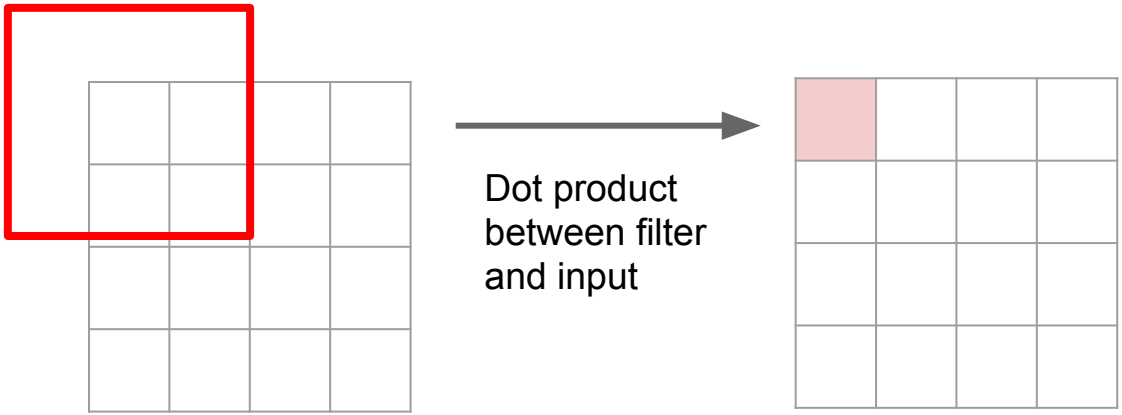


Output: 4 x 4

Slide Credit: CS231n

Reminder: Convolutional Layer

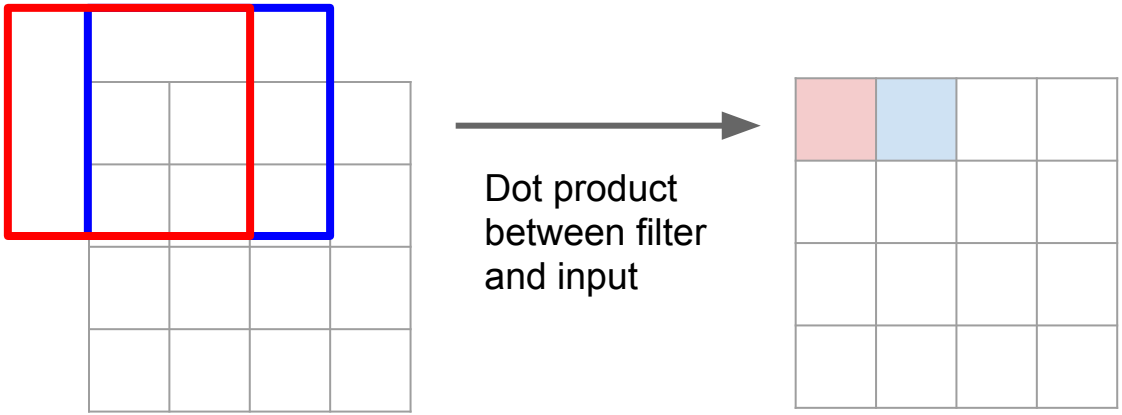
Typical 3 x 3 convolution, stride 1 pad 1



Slide Credit: CS231n

Reminder: Convolutional Layer

Typical 3 x 3 convolution, stride 1 pad 1



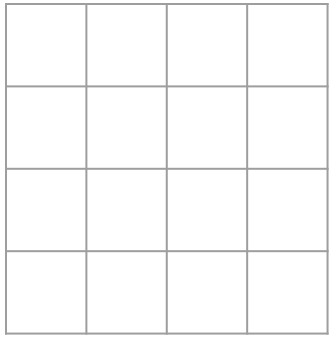
Input: 4 x 4

Output: 4 x 4

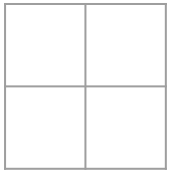
Slide Credit: CS231n

Reminder: Convolutional Layer

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



Input: 4 x 4

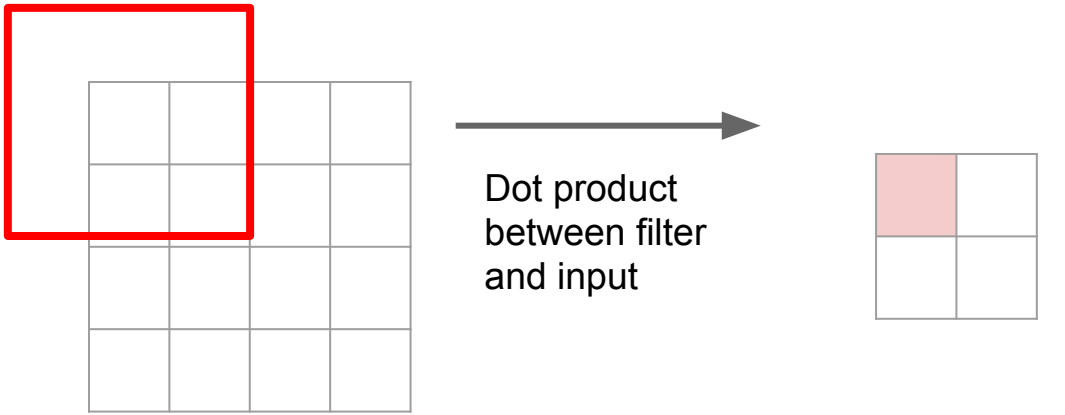


Output: 2 x 2

Slide Credit: CS231n

Reminder: Convolutional Layer

Typical 3 x 3 convolution, stride 2 pad 1



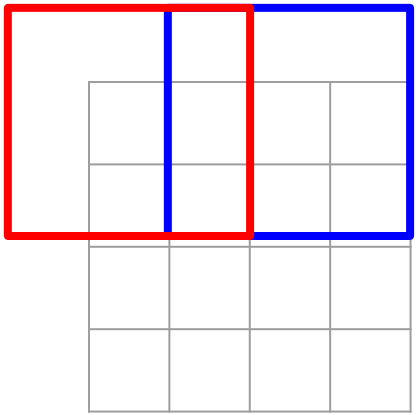
Input: 4 x 4

Output: 2 x 2

Slide Credit: CS231n

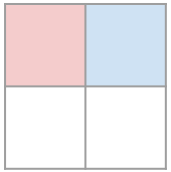
Reminder: Convolutional Layer

Typical 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4

Dot product between filter and input

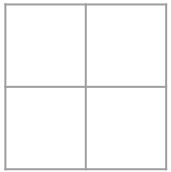


Output: 2 x 2

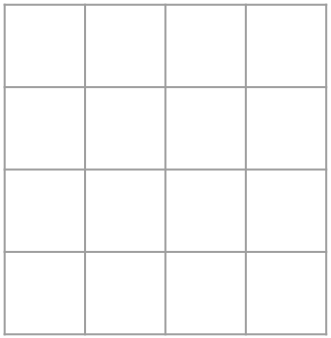
Slide Credit: CS231n

Learnable Upsample: Deconvolutional Layer

3 x 3 “deconvolution”, stride 2 pad 1



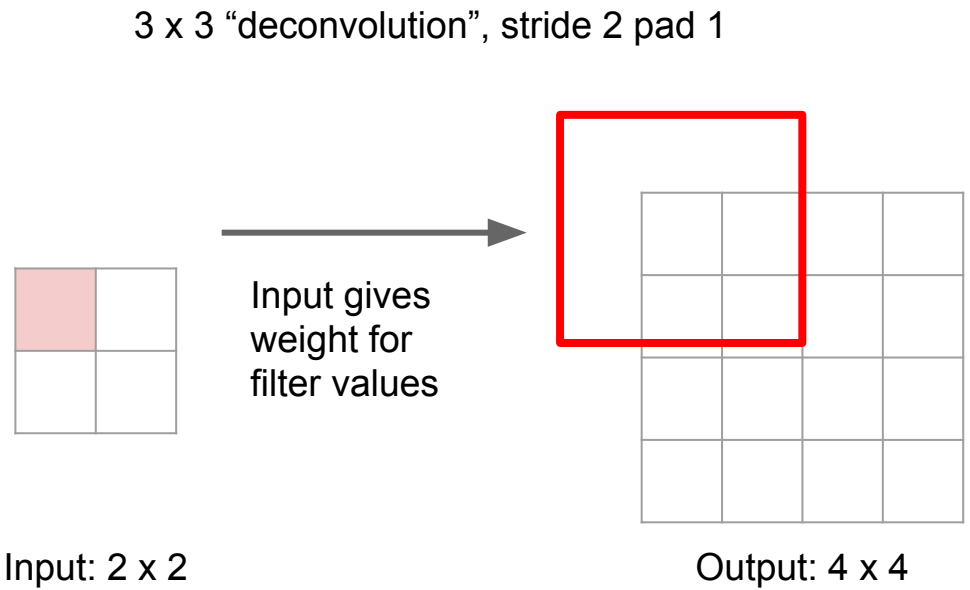
Input: 2 x 2



Output: 4 x 4

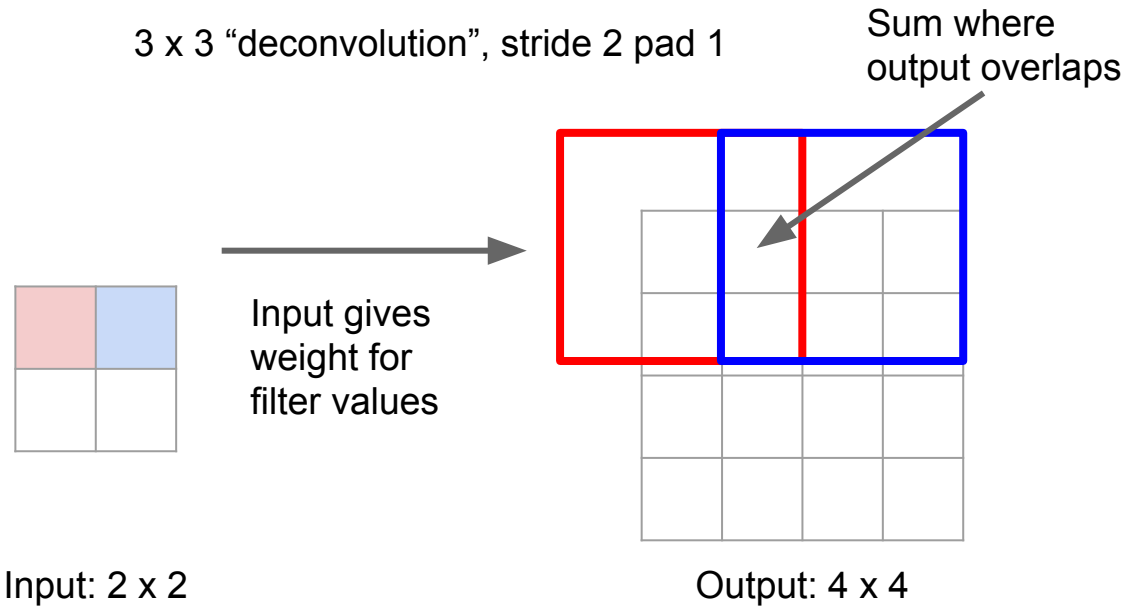
Slide Credit: CS231n

Learnable Upsample: Deconvolutional Layer



Slide Credit: CS231n

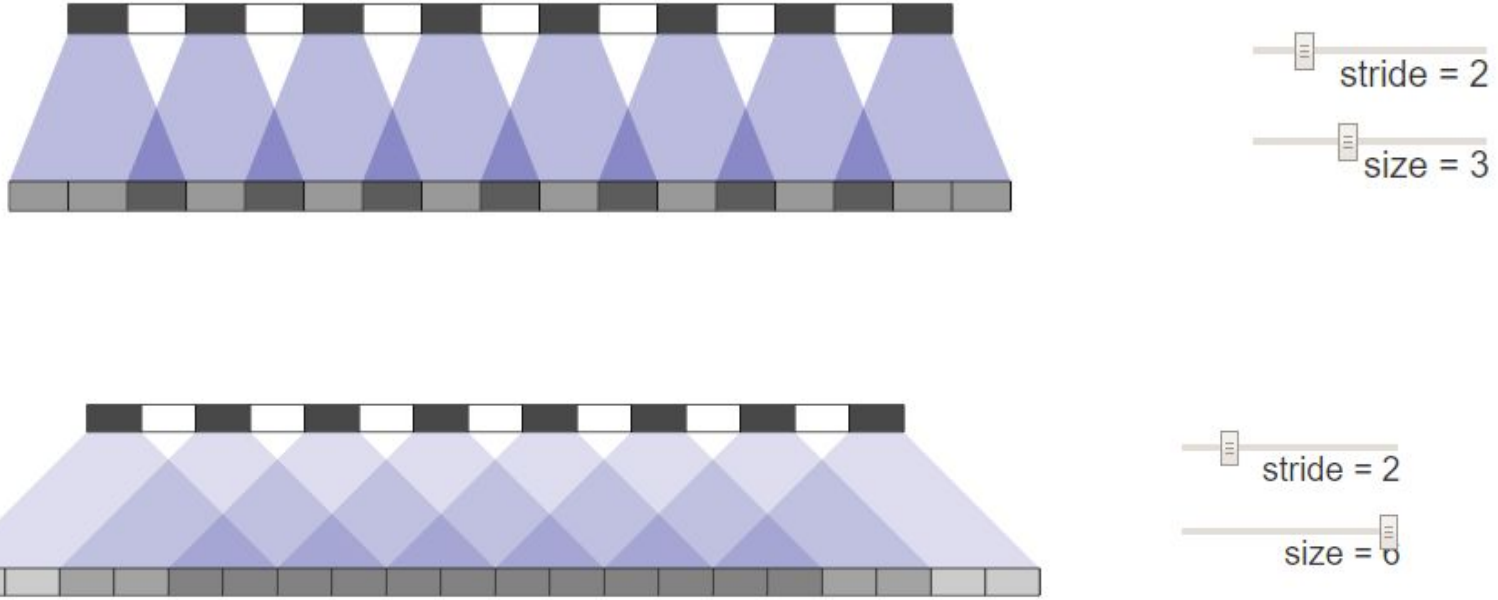
Learnable Upsample: Deconvolutional Layer



Slide Credit: CS231n

Learnable Upsample: Deconvolutional Layer

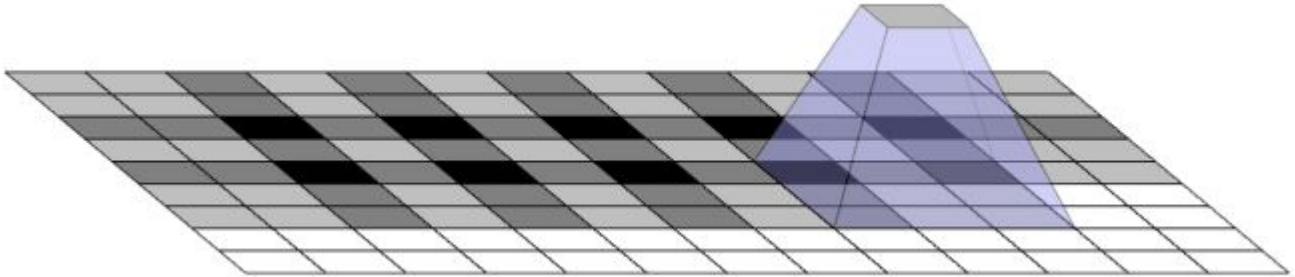
Warning: Checkerboard effect when kernel size is not divisible by the stride



Source: distill.pub

Learnable Upsample: Deconvolutional Layer

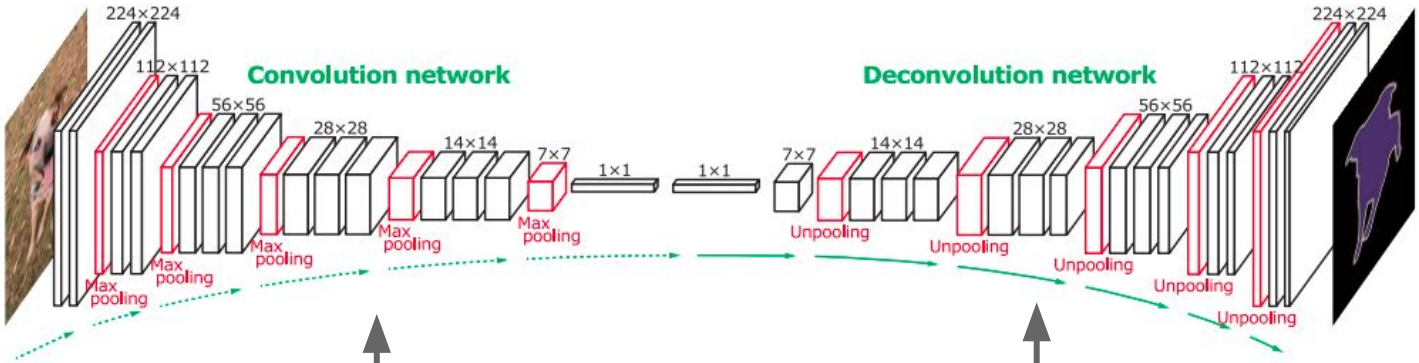
Warning: Checkerboard effect when kernel size is not divisible by the stride



stride = 2, kernel_size = 3

Source: distill.pub

Semantic Segmentation



“Regular” VGG

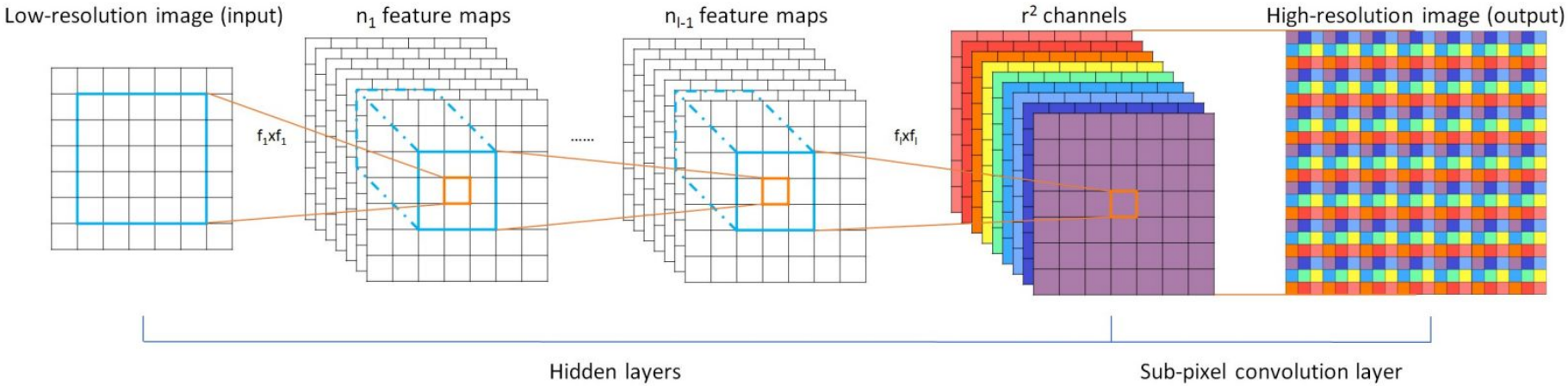
“Upside down” VGG

Noh et al. Learning Deconvolution Network for Semantic Segmentation. ICCV 2015

Slide Credit: CS231n

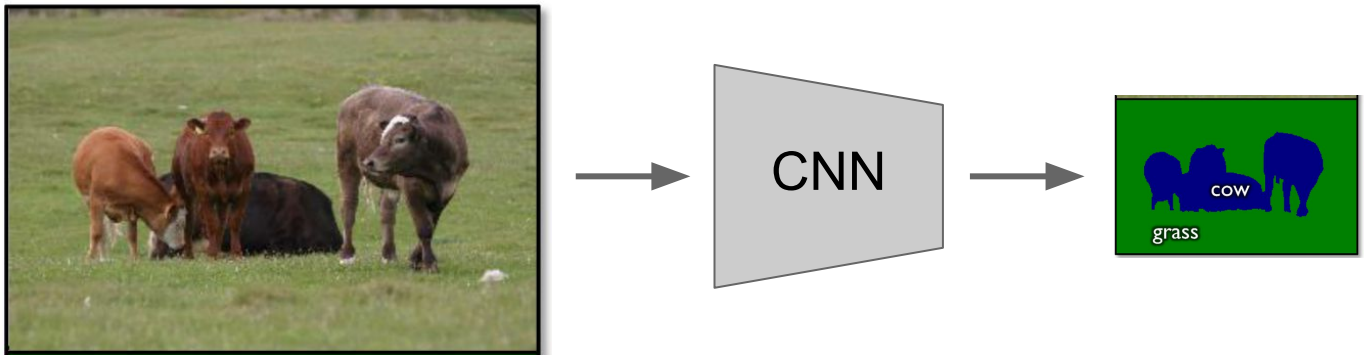
Better Upsampling: Subpixel

Re-arange features in previous convolutional layer to form a higher resolution output



Shi et al.[Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network](#).CVPR 2016

Semantic Segmentation



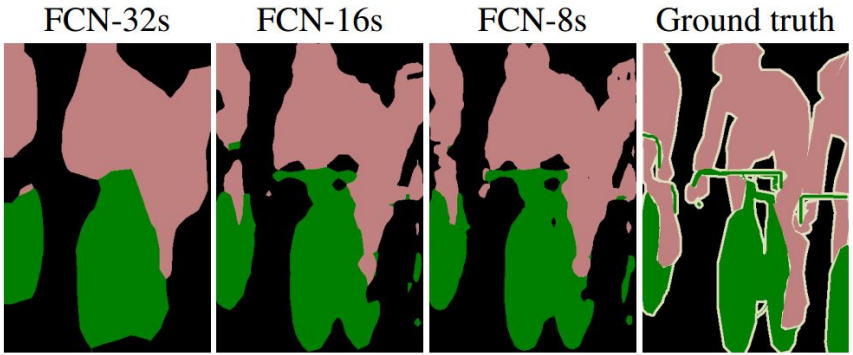
Problem 2:

Blobby-like segmentations

High-level features (e.g. conv5 layer) from a pretrained classification network are the input for the segmentation branch

Skip Connections

Recovering low level features from early layers



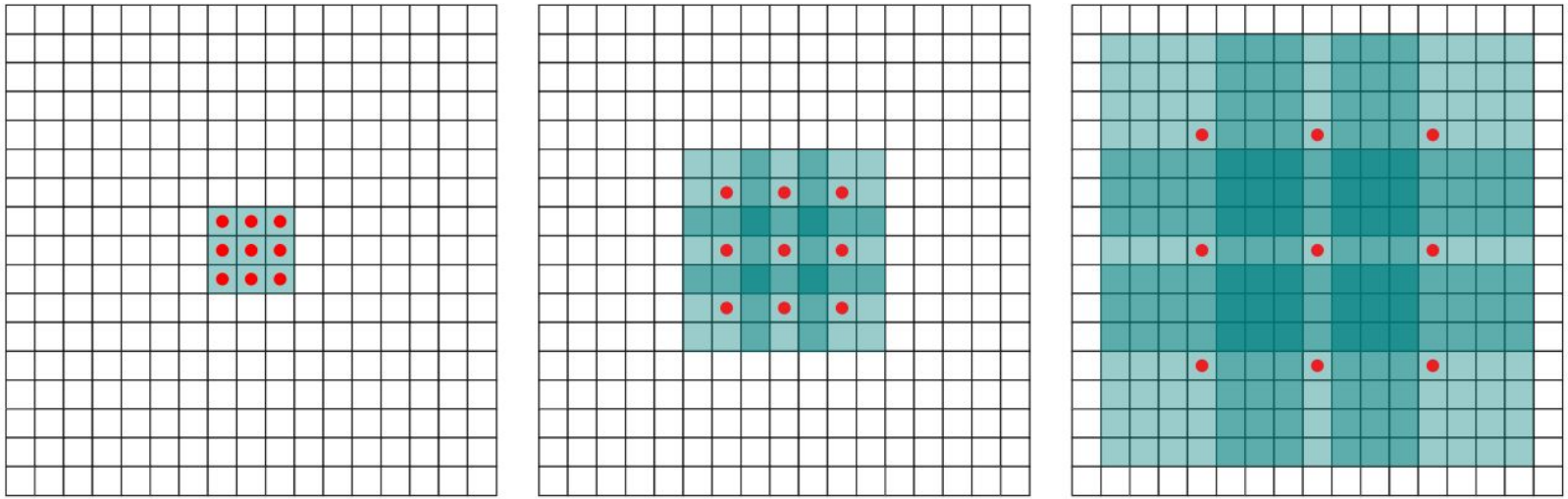
Skip connections = Better results

Long et al. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015

Slide Credit: CS231n

Dilated Convolutions

Structural change in convolutional layers for dense prediction problems (e.g. image segmentation)



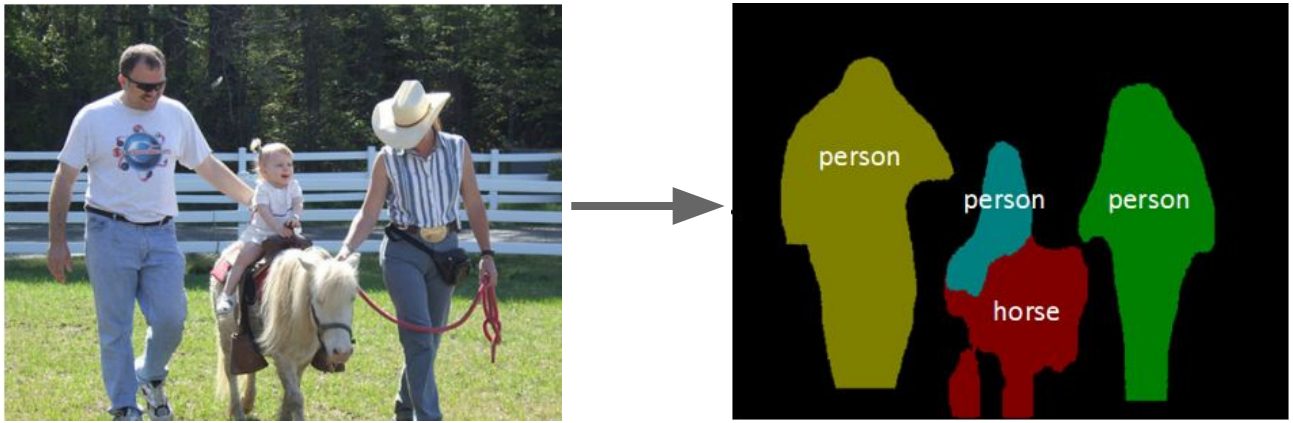
- The receptive field grows exponentially as you add more layers → more context information in deeper layers wrt regular convolutions
- Number of parameters increases linearly as you add more layers

Yu & Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. ICLR 2016

Instance Segmentation

Detect instances, give category, label pixels

“simultaneous detection and segmentation” (SDS)



Slide Credit: CS231n

Instance Segmentation

More challenging than Semantic Segmentation

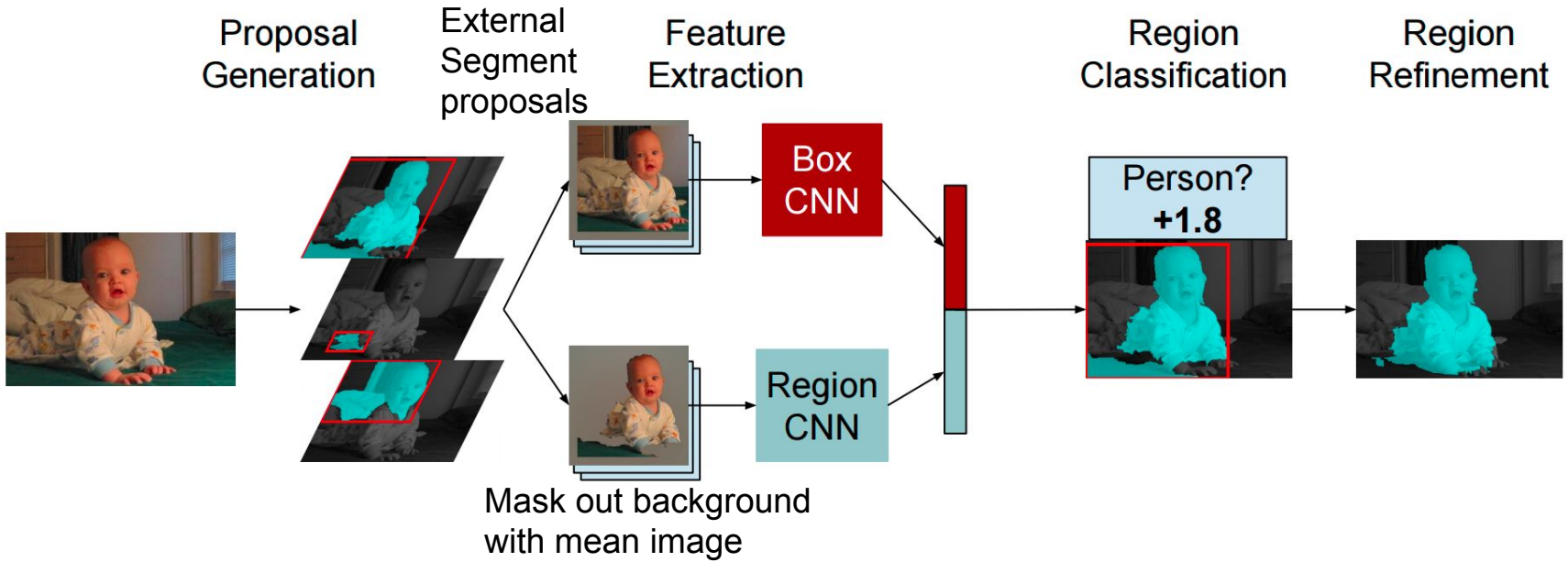
- Number of objects is variable
- No unique match between predicted and ground truth objects (cannot use instance IDs)

Several attack lines:

- Proposal-based methods
- Recurrent Neural Networks

Proposal-based

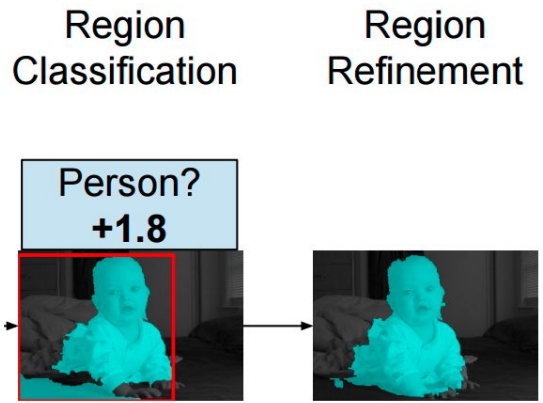
Similar to R-CNN, but with segment proposals

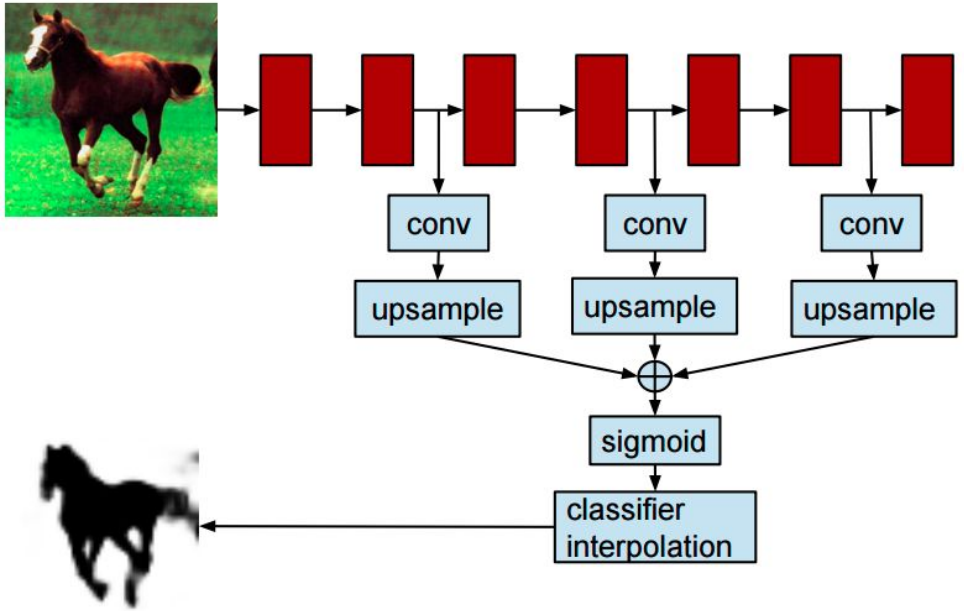


Hariharan et al. Simultaneous Detection and Segmentation. ECCV 2014

Slide Credit: CS231n

Proposal-based



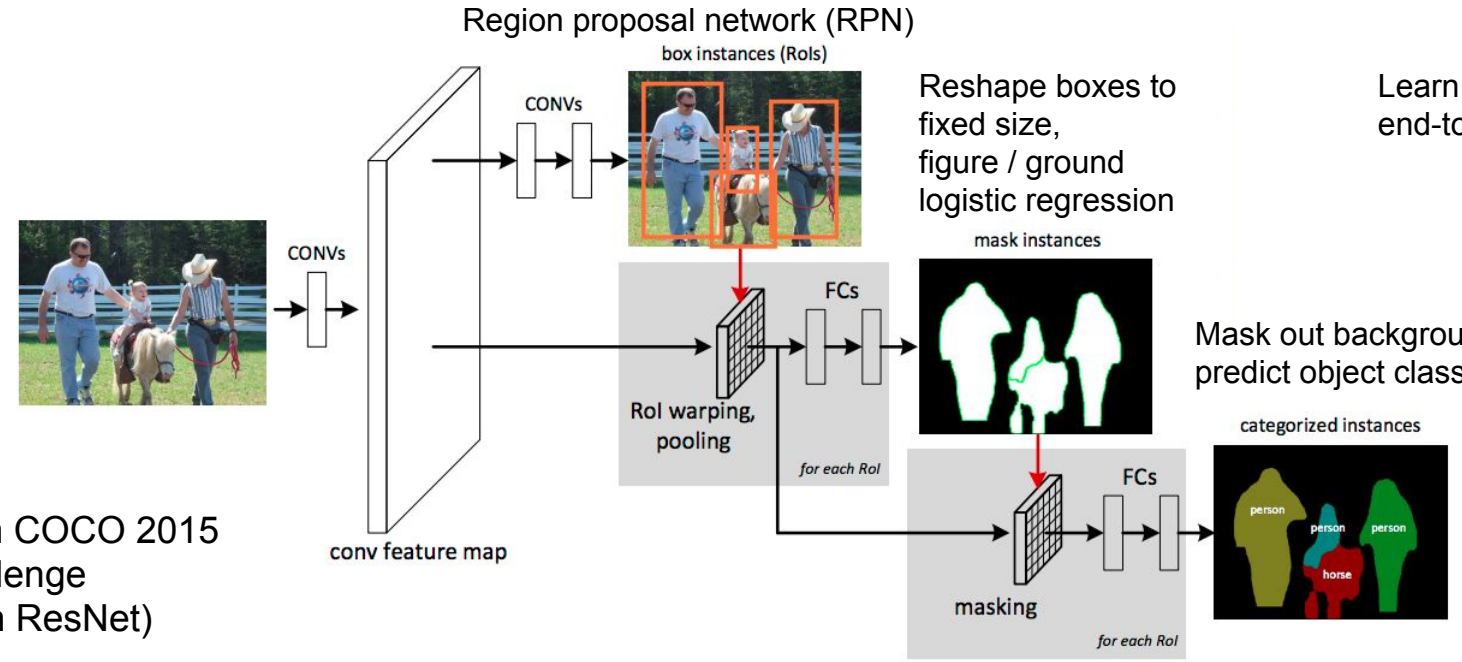


Hariharan et al. Hypercolumns for Object Segmentation and Fine-grained Localization. CVPR 2015

Slide Credit: CS231n

Proposal-based Instance Segmentation: MNC

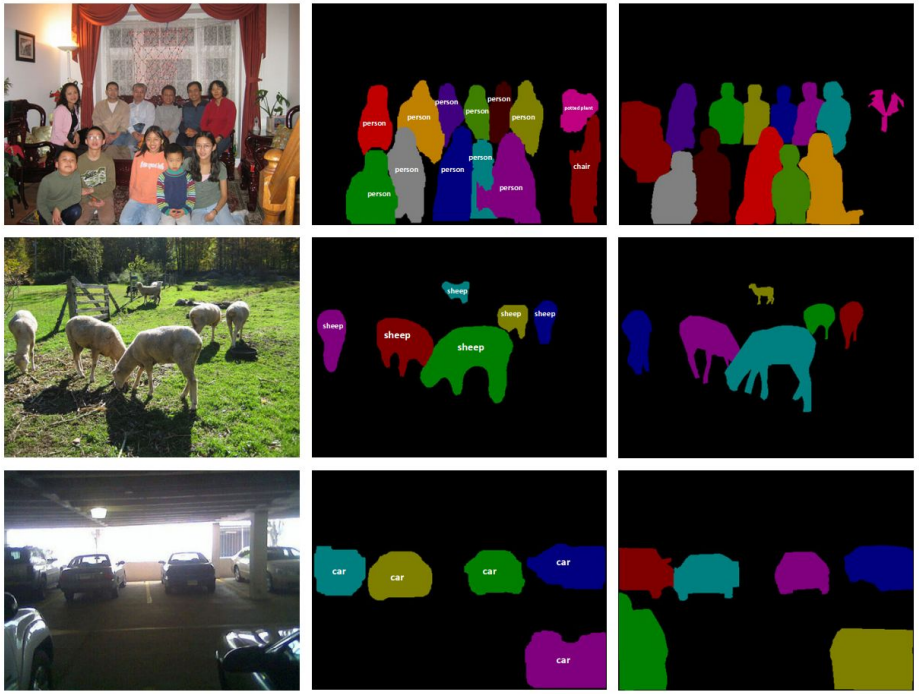
Faster R-CNN for Pixel Level Segmentation in a multi-stage cascade strategy



Won COCO 2015 challenge (with ResNet)

Dai et al. Instance-aware Semantic Segmentation via Multi-task Network Cascades. CVPR 2016

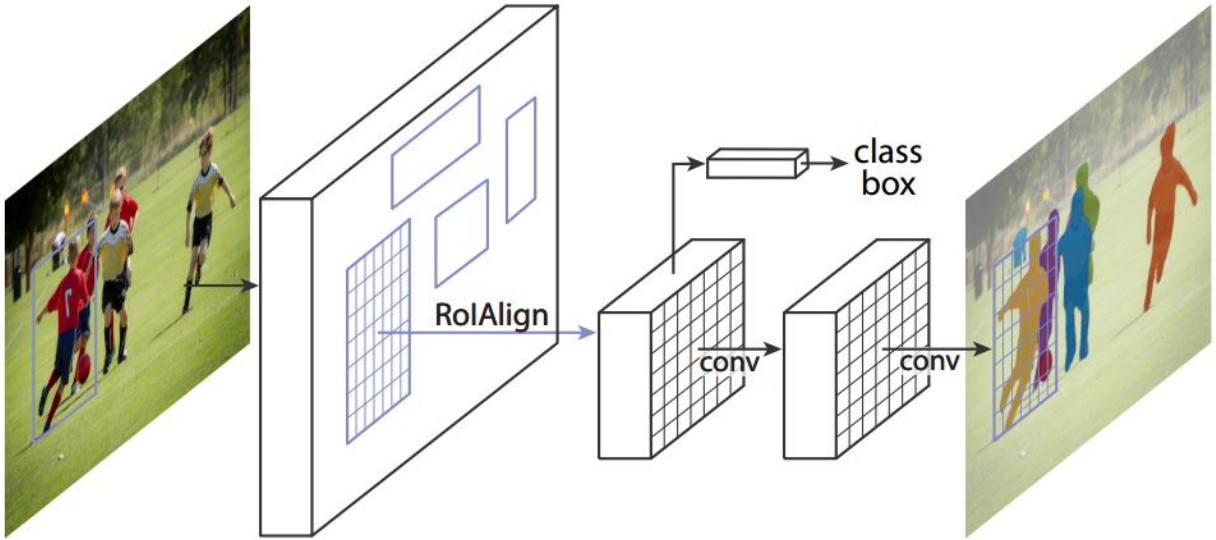
Proposal-based Instance Segmentation: MNC



Dai et al. Instance-aware Semantic Segmentation via Multi-task Network Cascades. CVPR 2016

Proposal-based Instance Segmentation: Mask R-CNN

Faster R-CNN for Pixel Level Segmentation as a **parallel prediction of masks and class labels**

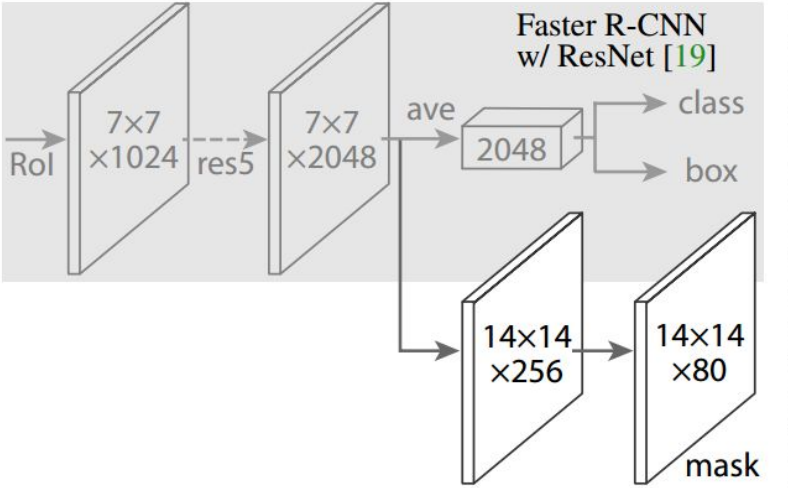


He et al. Mask R-CNN. arXiv Mar 2017

Mask R-CNN

- Classification & box detection losses are identical to those in Faster R-CNN
- Addition of a new loss term for mask prediction:

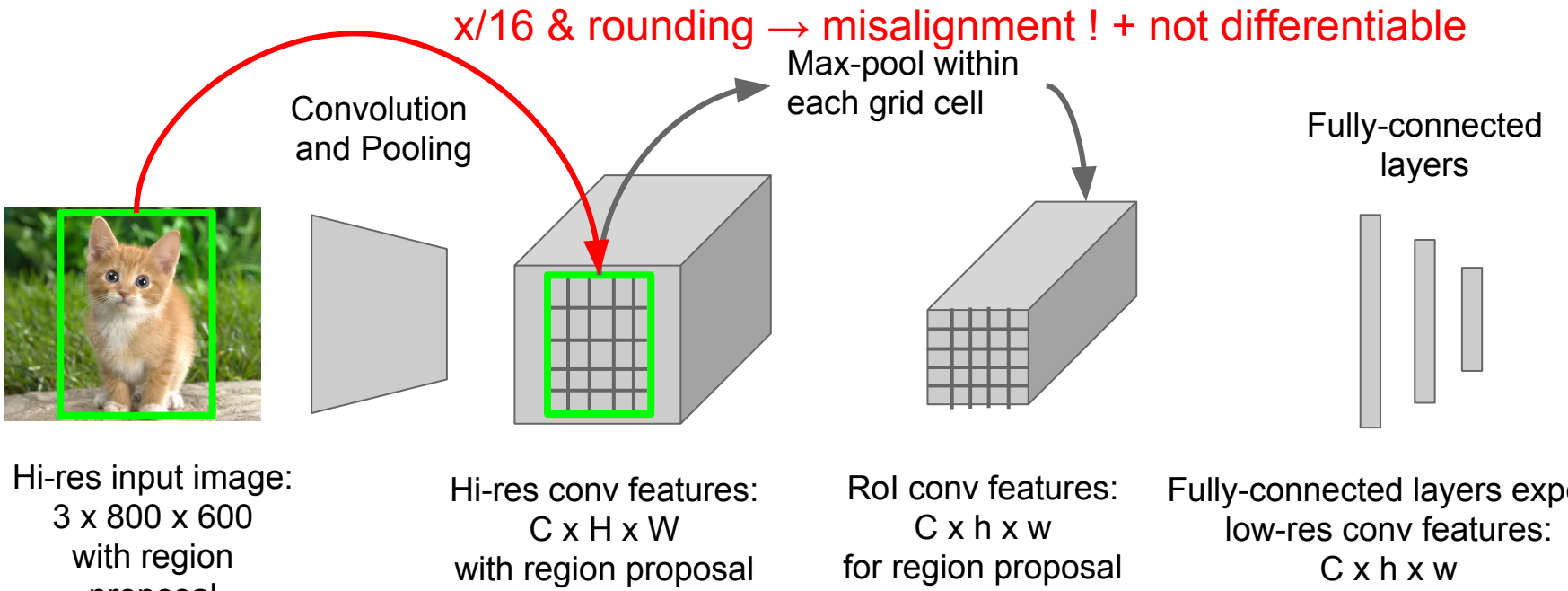
The network outputs a *K x m x m* volume for mask prediction, where *K* is the number of categories and *m* is the size of the mask (square)



He et al. Mask R-CNN. arXiv Mar 2017

Mask R-CNN: RoI Align

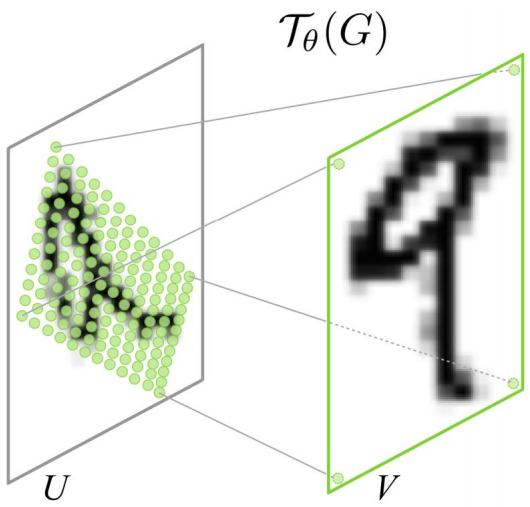
Reminder: RoI Pool from Fast R-CNN



He et al. Mask R-CNN. arXiv Mar 2017

Mask R-CNN: RoI Align

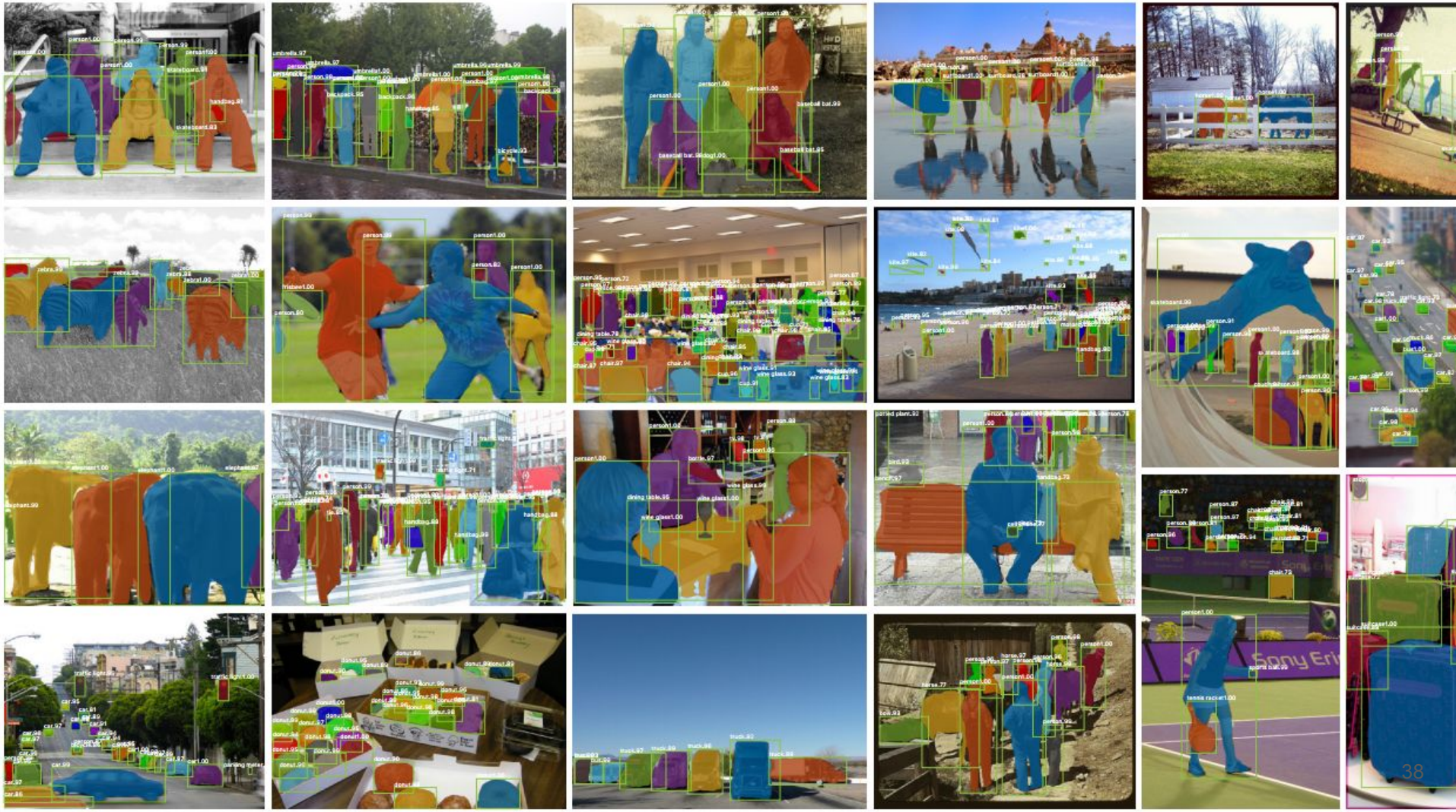
Use bilinear interpolation instead of cropping + maxpool



Mapping given by box coordinates
(θ_{12} and θ_{21} = 0 translation + scale)

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

Jaderberg et al. Spatial Transformer Networks. NIPS 2015



Mask R-CNN

Instance Segmentation

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

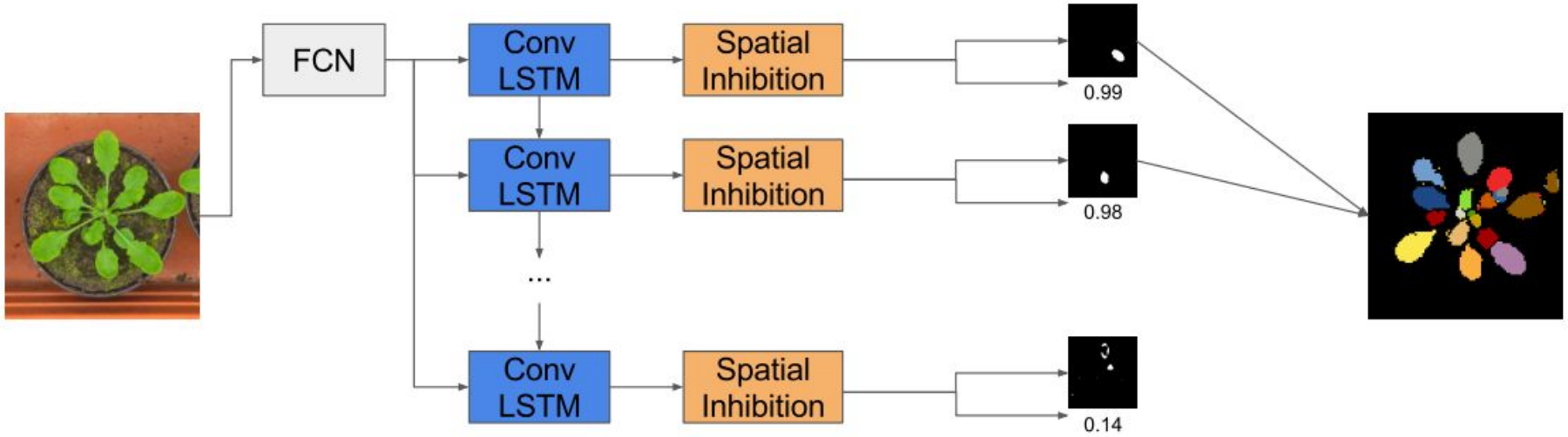
Object Detection

	backbone	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_S	AP^{bb}_M	AP^{bb}_L
Faster R-CNN+++ [19]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [27]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [21]	Inception-ResNet-v2 [37]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [36]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
Faster R-CNN, RoIAlign	ResNet-101-FPN	37.3	59.6	40.3	19.8	40.2	48.8
Mask R-CNN	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Mask R-CNN	ResNeXt-101-FPN	**39.8**	**62.3**	**43.4**	**22.1**	**43.2**	51.2

He et al. Mask R-CNN. arXiv Mar 2017

Recurrent Instance Segmentation

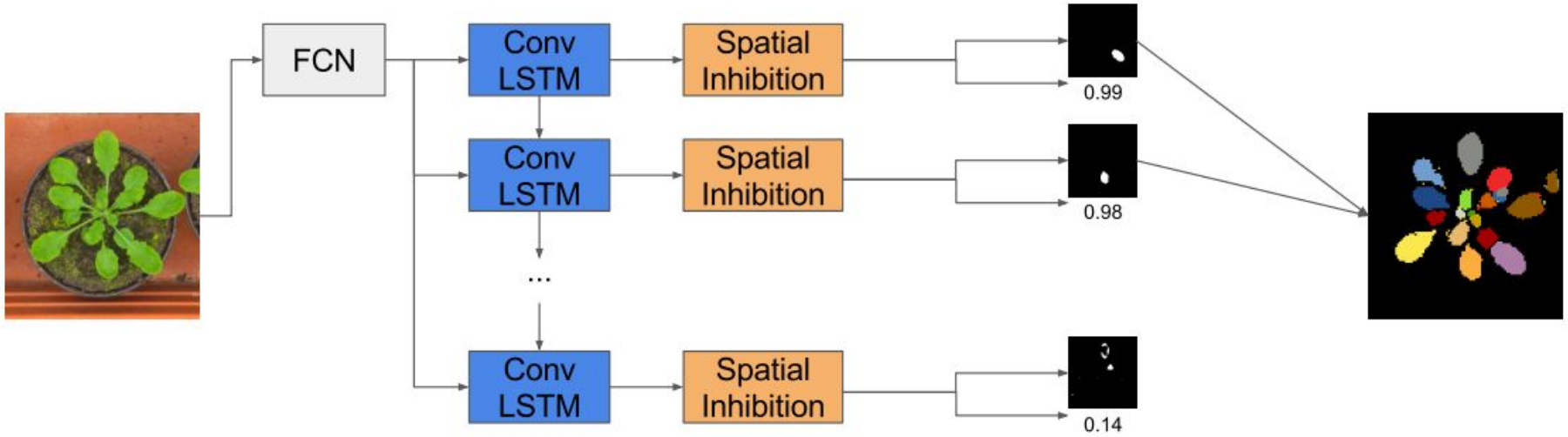
Sequential mask generation



Romera-Paredes & H.S. Torr. Recurrent Instance Segmentation ECCV 2016

Recurrent Instance Segmentation

Mapping between ground truth and predicted masks ?



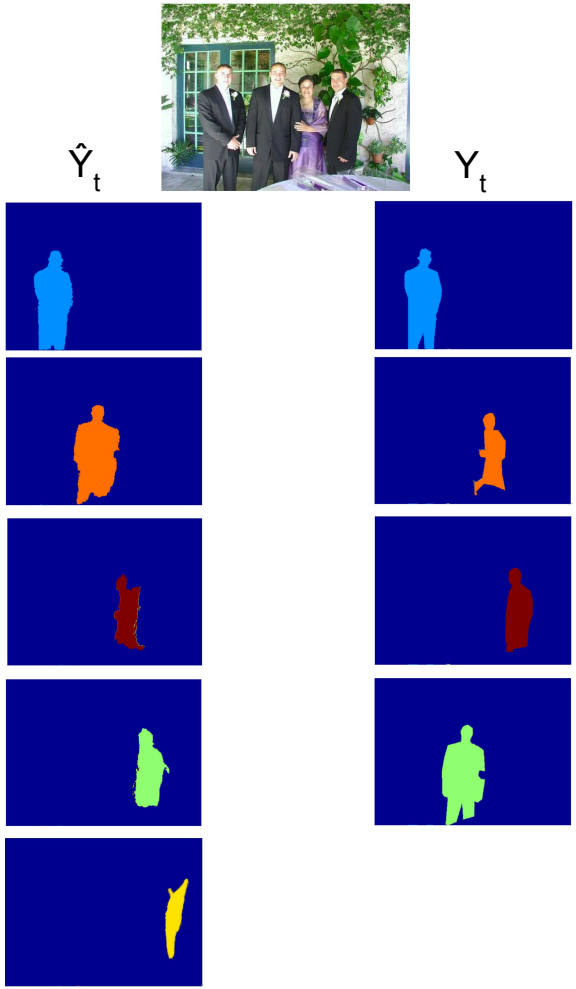
Romera-Paredes & H.S. Torr. Recurrent Instance Segmentation ECCV 2016

Recurrent Instance Segmentation:

Coverage Loss

1-Compute the IoU for all pairs of Predicted/GT masks

$$f_{IoU}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\langle \hat{\mathbf{y}}, \mathbf{y} \rangle}{\|\hat{\mathbf{y}}\|_1 + \|\mathbf{y}\|_1 - \langle \hat{\mathbf{y}}, \mathbf{y} \rangle}$$



Romera-Paredes & H.S. Torr. Recurrent Instance Segmentation ECCV 2016

Slide Credit: M. Baradad, ReadCV@UPC

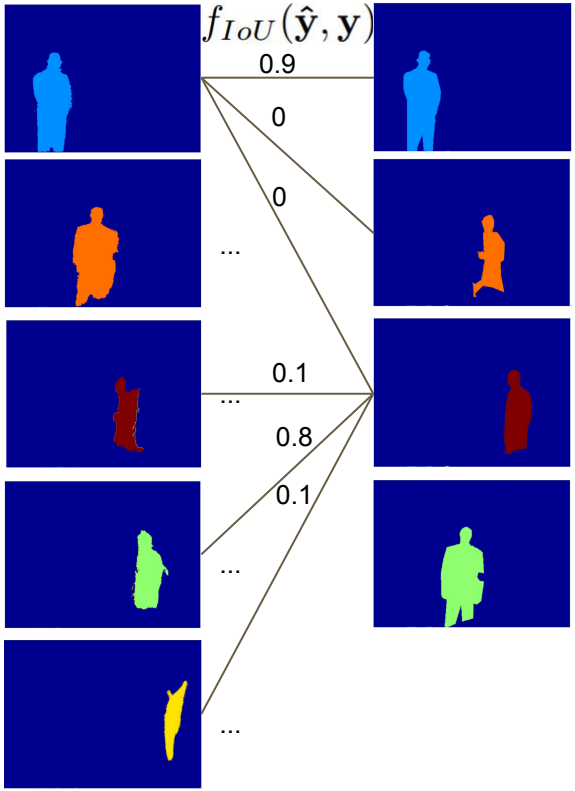
Recurrent Instance Segmentation:

Coverage Loss

1-Compute the IoU for all pairs of Predicted/GT masks

$$f_{IoU}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\langle \hat{\mathbf{y}}, \mathbf{y} \rangle}{\|\hat{\mathbf{y}}\|_1 + \|\mathbf{y}\|_1 - \langle \hat{\mathbf{y}}, \mathbf{y} \rangle}$$

$$\begin{pmatrix} 0.9 & 0 & 0 & 0.08 \\ 0.12 & 0.14 & 0 & 0.87 \\ 0 & 0.68 & 0.1 & .0.7 \\ 0 & 0.1 & 0.8 & 0 \\ 0 & 0 & 0.1 & 0 \end{pmatrix}$$



Romera-Paredes & H.S. Torr. Recurrent Instance Segmentation ECCV 2016

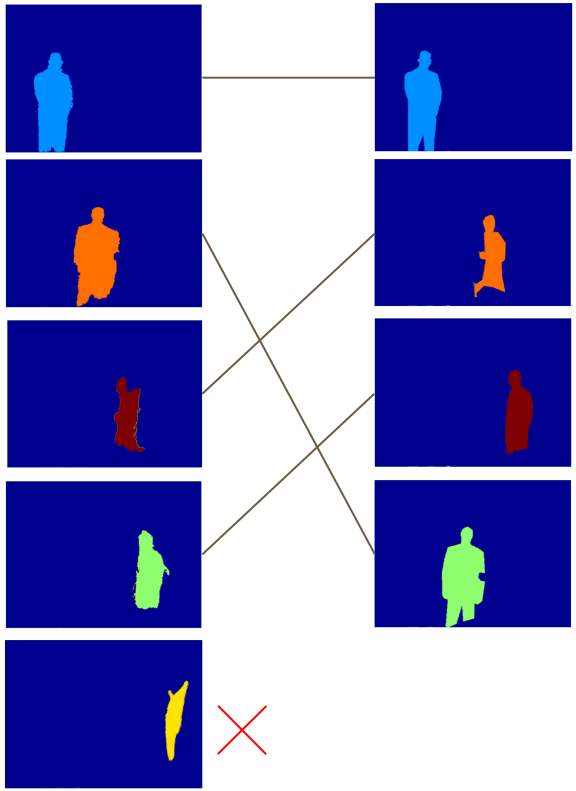
Slide Credit: M. Baradad, ReadCV@UPC

Recurrent Instance Segmentation:

Coverage Loss

2-Find best matching:

$$\begin{pmatrix} \underline{0.9} & 0 & 0 & 0.08 \\ 0.12 & 0.14 & 0 & \underline{0.87} \\ 0 & \underline{0.68} & 0.1 & .0.7 \\ 0 & 0.1 & \underline{0.8} & 0 \\ 0 & 0 & 0.1 & 0 \end{pmatrix}$$



Loss: Sum of the Intersections over the union for the best matching (*-1)

Romera-Paredes & H.S. Torr. Recurrent Instance Segmentation ECCV 2016

Slide Credit: M. Baradad, ReadCV@UPC

Recurrent Instance Segmentation:

Coverage Loss

3-Also take into account the scores

$$\lambda \sum_{t=1}^{\hat{n}} f_{BCE}\left([t \leq n], s_t\right)$$

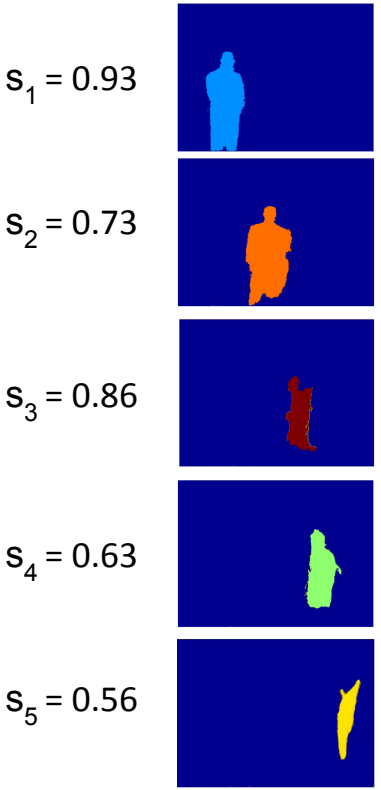
Where:

$f_{BCE}(a, b)$ is the binary cross entropy:

$f_{BCE}(a, b) = -\left(a\log(b) + (1-a)\log(1-b)\right)$

$[\cdot]$ is the Iverson bracket which:

Is 1 if the condition is true and 0 else



Romera-Paredes & H.S. Torr. Recurrent Instance Segmentation ECCV 2016

Slide Credit: M. Baradad, ReadCV@UPC

Recurrent Instance Segmentation:

Coverage Loss

4-Add everything together

$$\ell(\hat{\mathbf{Y}}, \mathbf{s}, \mathbf{Y}) = \min_{\delta \in \mathcal{S}} - \sum_{\hat{t}=1}^{\hat{n}} \sum_{t=1}^{n} f_{IoU}\left(\hat{\mathbf{Y}}_{\hat{\mathbf{t}}}, \mathbf{Y}_{\mathbf{t}}\right) \delta_{\hat{t},t} + \lambda \sum_{t=1}^{\hat{n}} f_{BCE}\left([t \leq n], s_t\right)$$

Romera-Paredes & H.S. Torr. Recurrent Instance Segmentation ECCV 2016

Slide Credit: M. Baradad, ReadCV@UPC

Recurrent Instance Segmentation

Romera-Paredes & H.S. Torr. Recurrent Instance Segmentation ECCV 2016

Summary

Segmentation Datasets

Semantic Segmentation Methods

- Deconvolution
- Dilated Convolution
- Skip Connections

Instance Segmentation Methods

- Proposal-Based
- Recurrent

Questions ?