

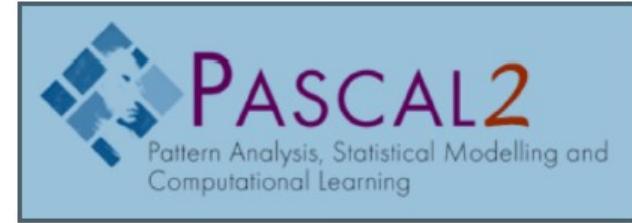
# Image Segmentation

Fast Campus  
Start Deep Learning with TensorFlow

# Semantic Segmentation



- PASCAL VOC segmentation
  - 10K images, 20 classes + bgnd
- MS COCO
  - 100K images, 80 classes + bgnd



# Fully Convolutional Networks for Semantic Segmentation

## Fully Convolutional Networks for Semantic Segmentation

Jonathan Long\*

Evan Shelhamer\*

Trevor Darrell

UC Berkeley

{jonlong, shelhamer, trevor}@cs.berkeley.edu

### Abstract

Convolutional networks are powerful visual models that yield hierarchies of features. We show that convolutional networks by themselves, trained end-to-end, pixels-to-pixels, exceed the state-of-the-art in semantic segmentation. Our key insight is to build “fully convolutional” networks that take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning. We define and detail the space of fully convolutional networks, explain their application to spatially dense prediction tasks, and draw connections to prior models. We adapt contemporary classification networks (AlexNet [22],

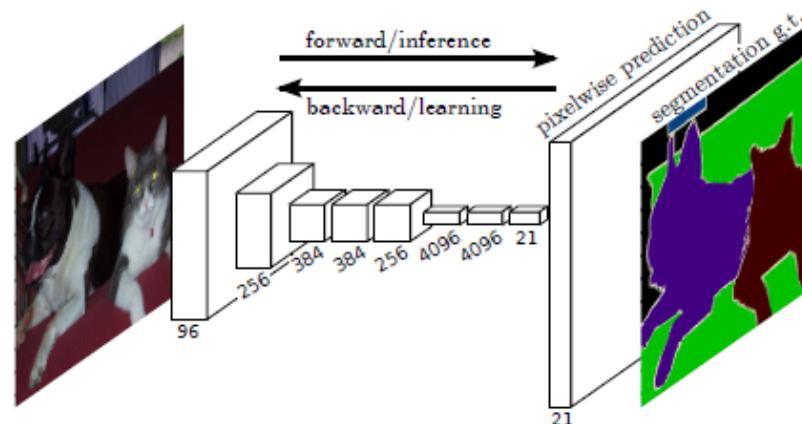
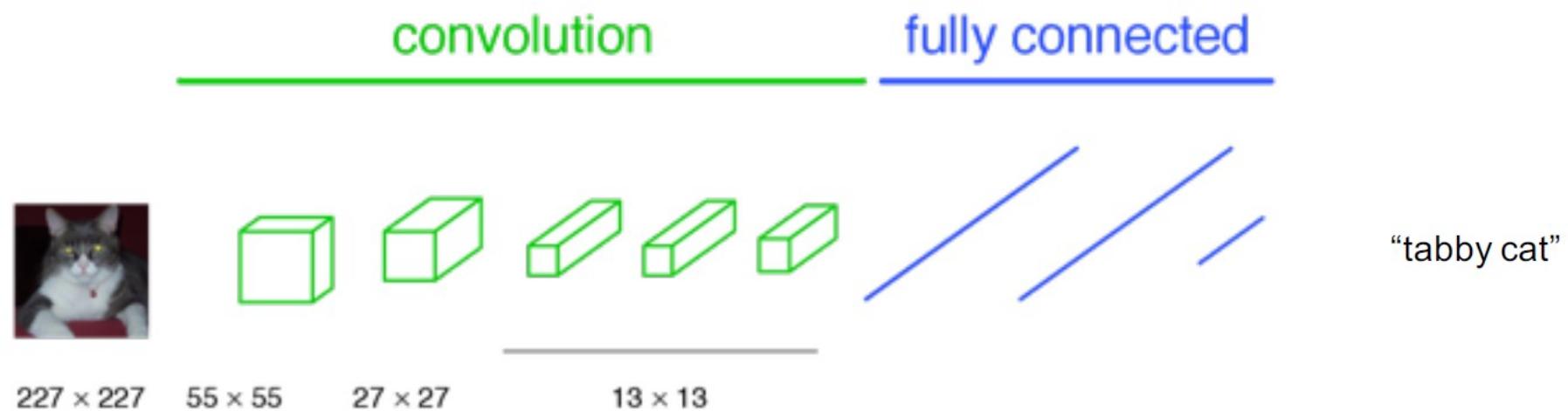


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

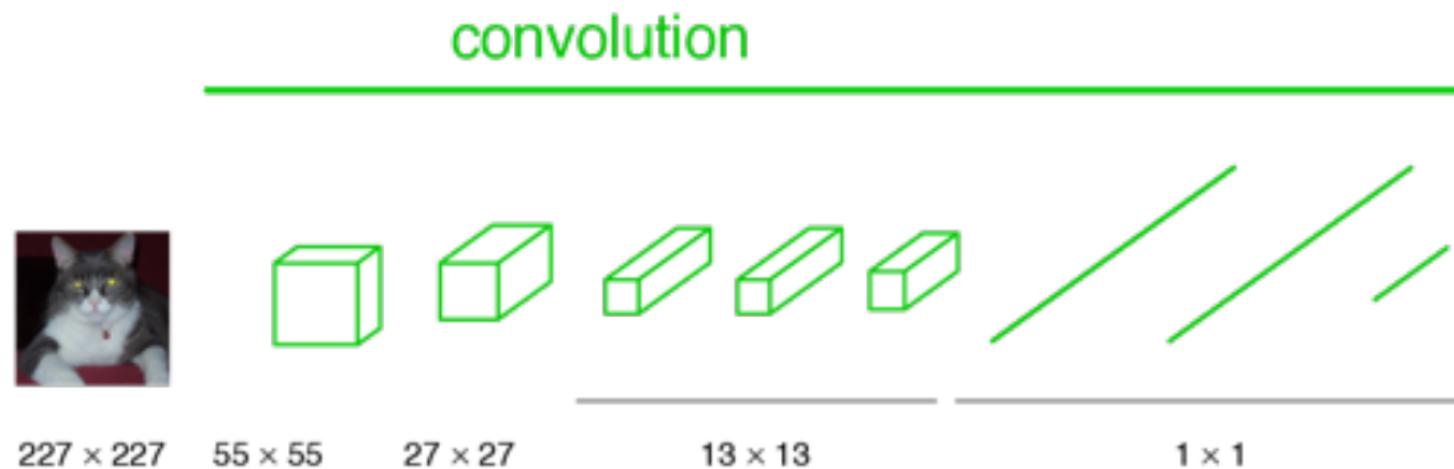
# Fully Convolutional Networks for Semantic Segmentation

- Fully convolutional network(FCN)
  - Pixel-wise prediction with end to end learning
- Fully Convolutional versions of existing networks predict dense outputs from arbitrary-sized inputs

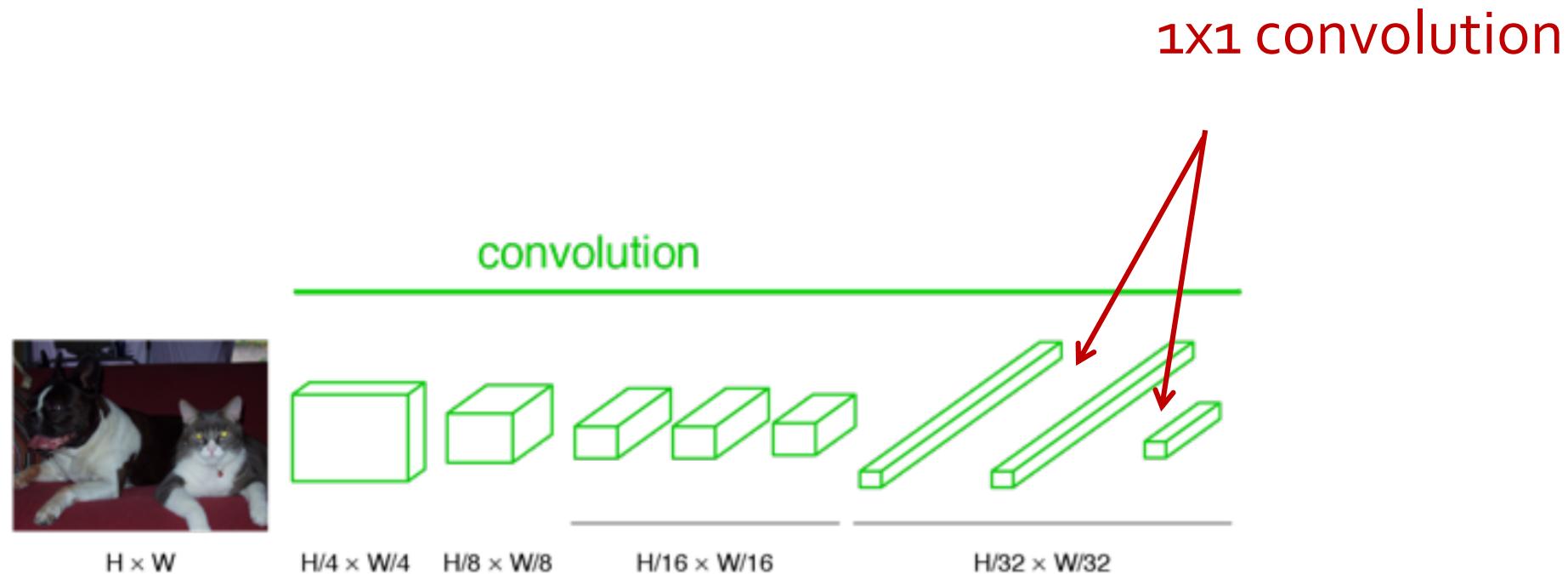
# Classification Network



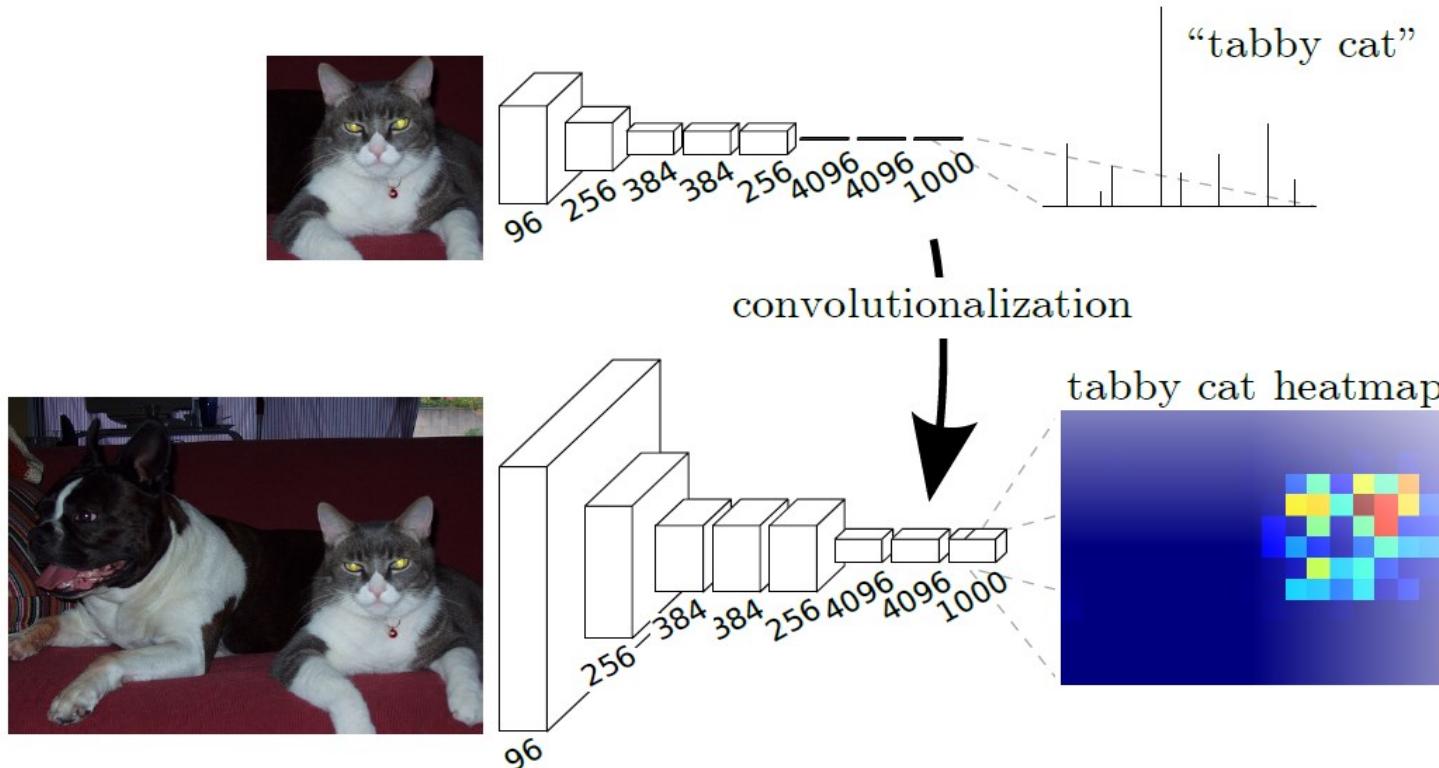
# Becoming Fully Convolutional Network



# Becoming Fully Convolutional Network

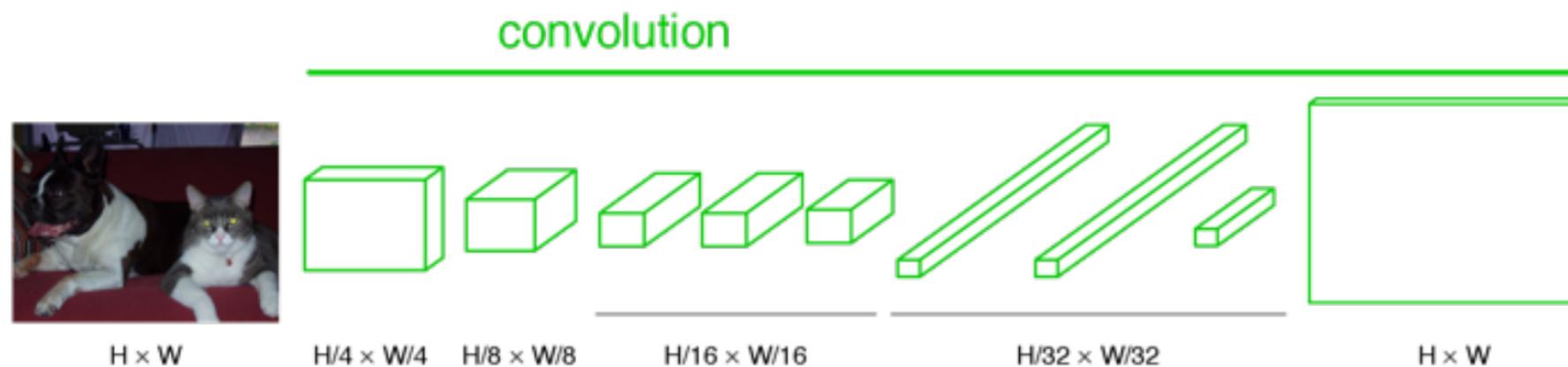


# Heatmap from FCN

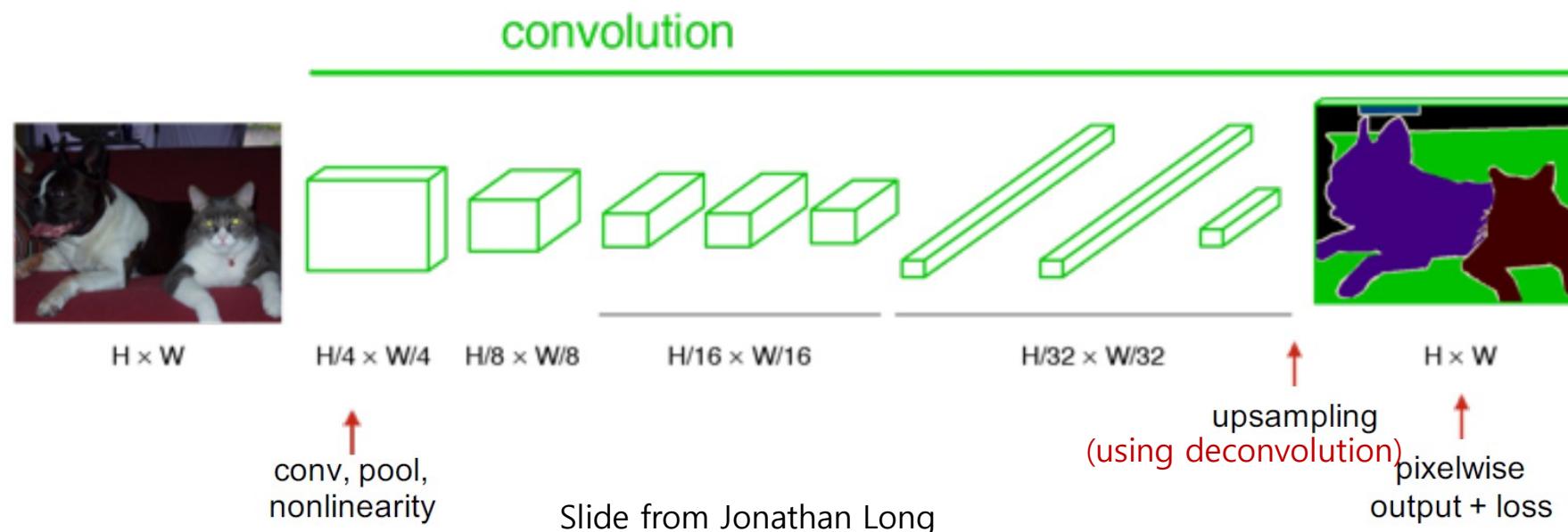


# Upsampling Output

- They use deconvolution for upsampling



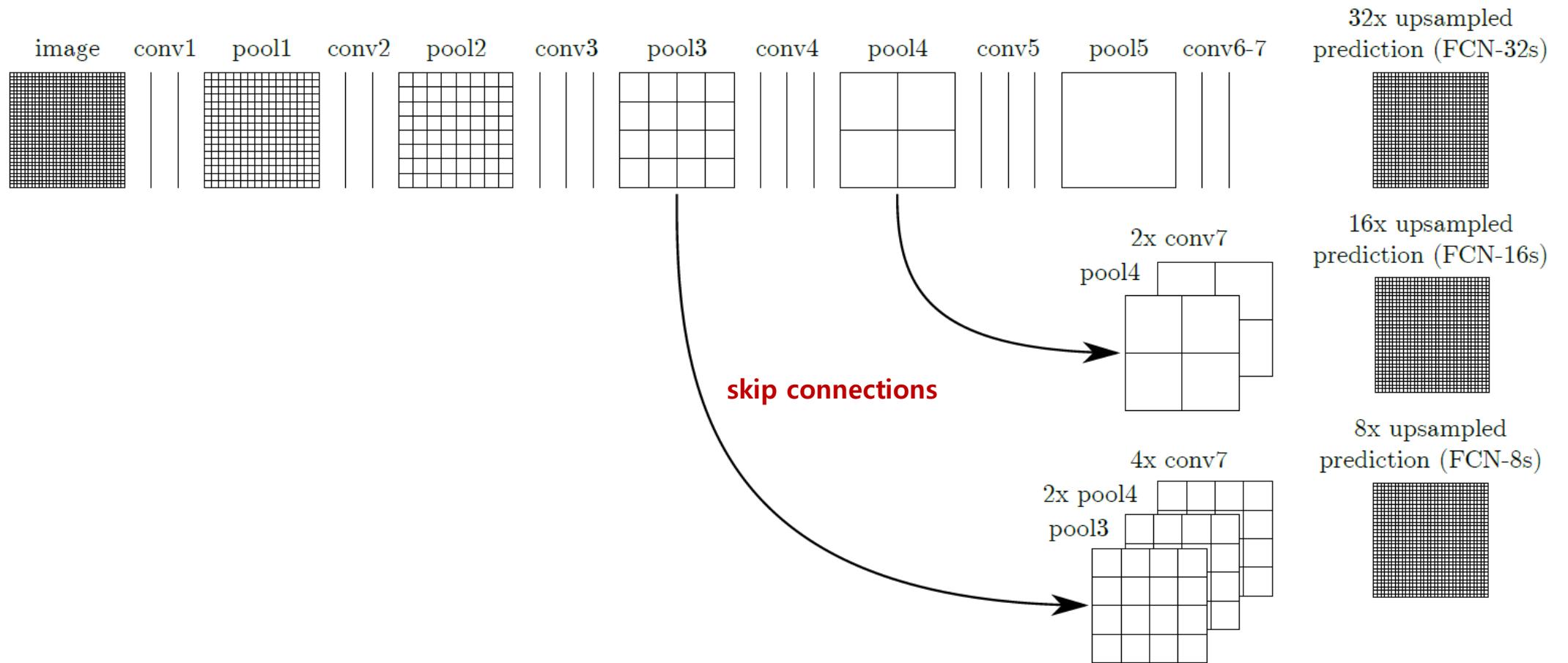
# End to End, Pixels to Pixels Network



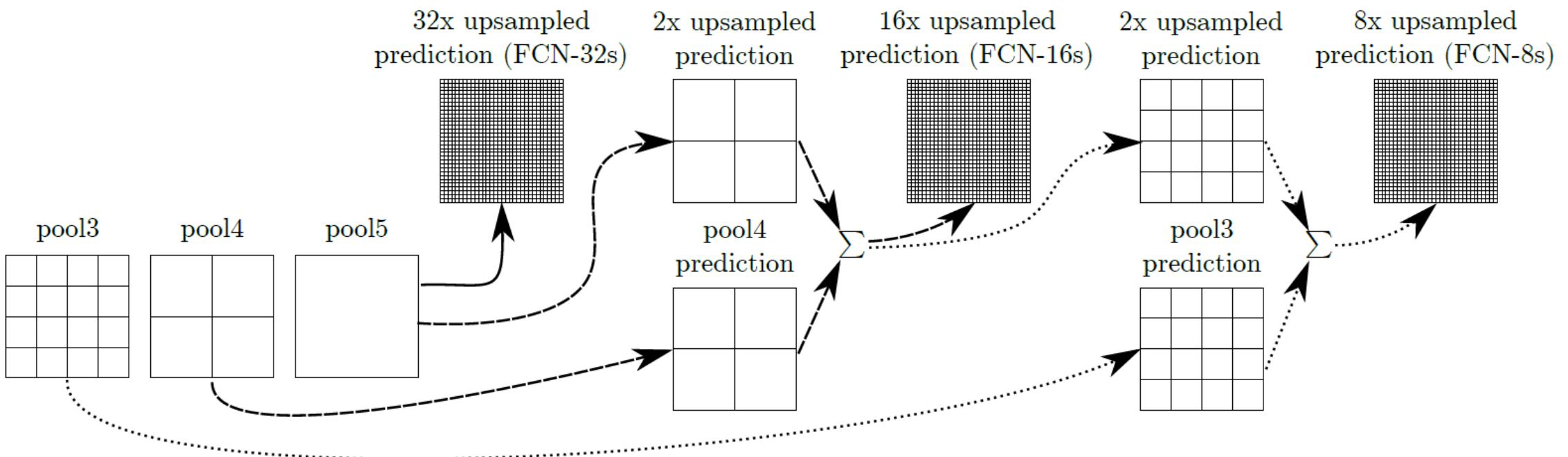
# Classifier to Dense FCN

- Convolutionalize proven classification architectures:
  - AlexNet, VGG, and GoogLeNet (reimplementation)
- Remove classification layer and convert all fully connected layers to convolutions
- Append  $1 \times 1$  convolution with channel dimensions and predict scores at each of the coarse output locations (21 categories for PASCAL)

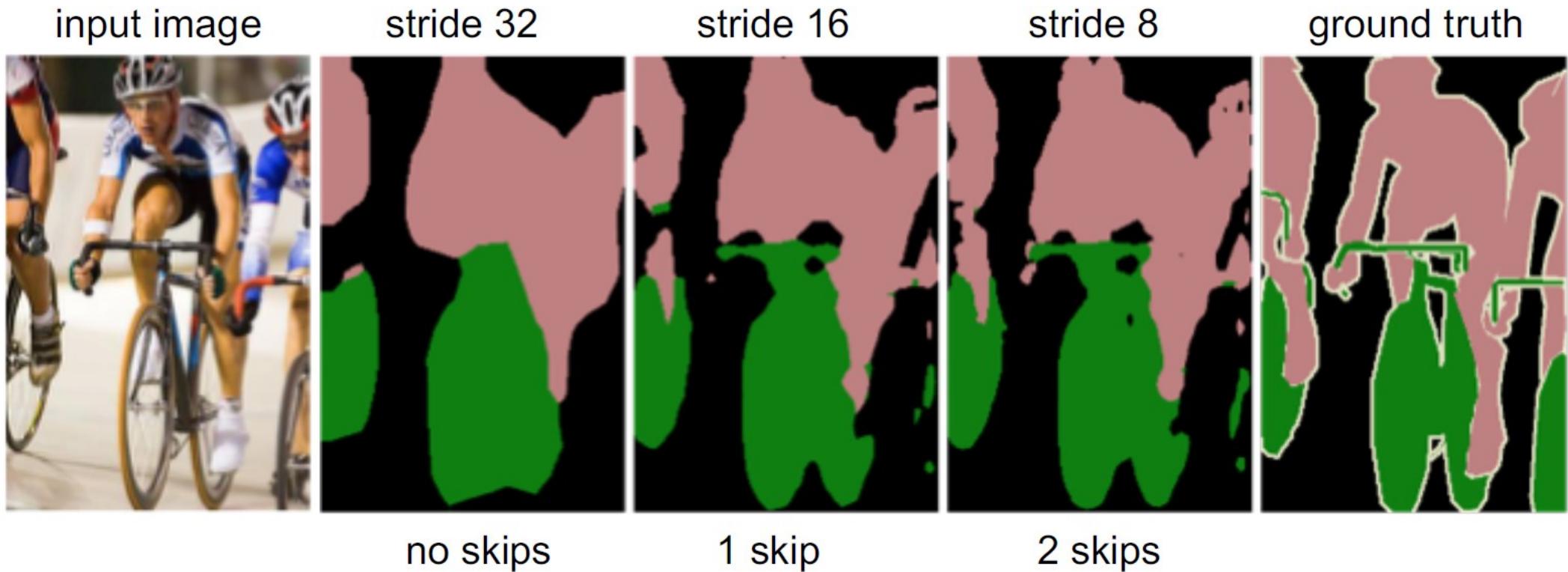
# Skip Layers



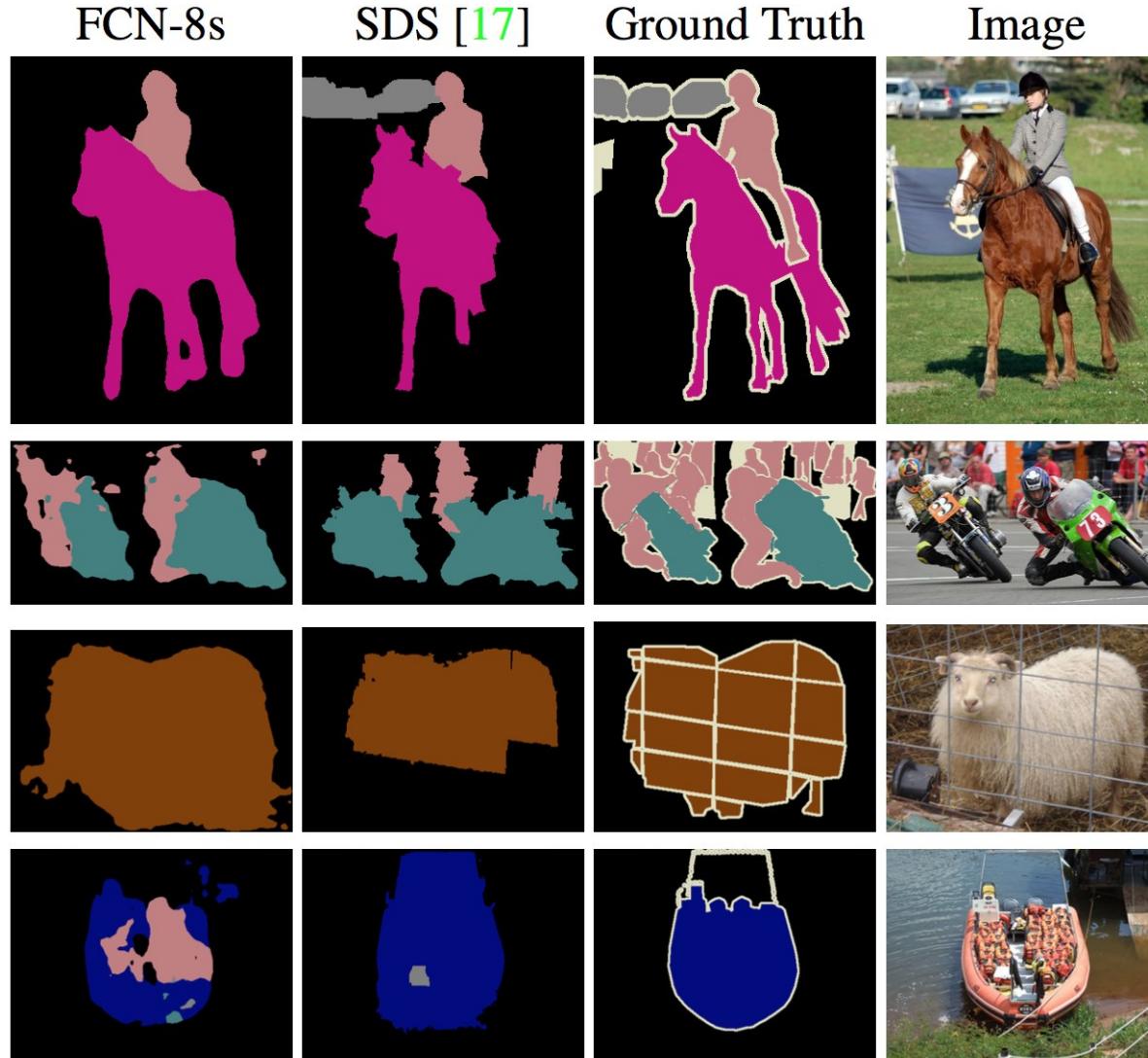
# Skip Layers



# Results



# Results



# Learning Deconvolution Network for Semantic Segmentation

## Learning Deconvolution Network for Semantic Segmentation

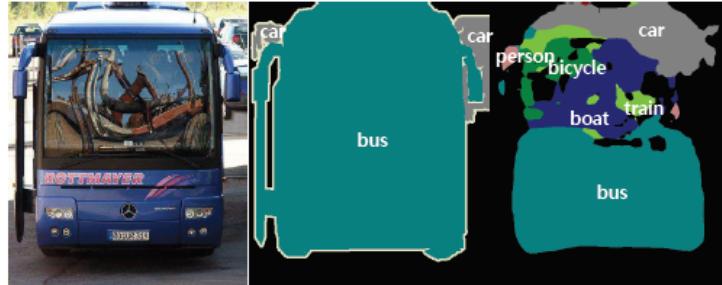
Hyeonwoo Noh

Department of Computer Science and Engineering, POSTECH, Korea

{hyeonwoonoh\_, maga33, bhhan}@postech.ac.kr

### Abstract

We propose a novel semantic segmentation algorithm by learning a deep deconvolution network. We learn the network on top of the convolutional layers adopted from VGG 16-layer net. The deconvolution network is composed of deconvolution and unpooling layers, which identify pixel-wise class labels and predict segmentation masks. We apply the trained network to each proposal in an input image, and construct the final semantic segmentation map by combining the results from all proposals in a simple manner. The proposed algorithm mitigates the limitations of the existing methods based on fully convolutional networks by integrating deep deconvolution network and proposal-wise prediction; our segmentation method typically identifies detailed structures and handles objects in multiple scales naturally. Our network demonstrates outstanding performance



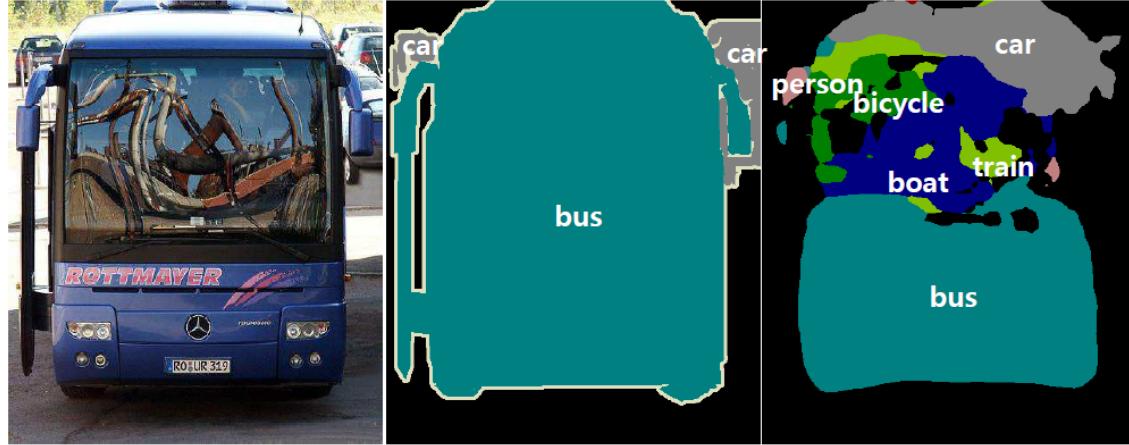
(a) Inconsistent labels due to large object size



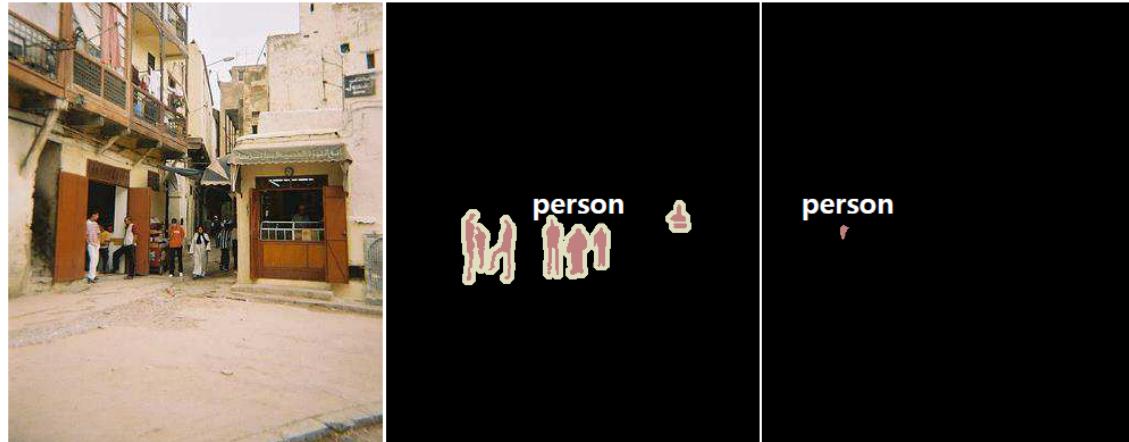
(b) Missing labels due to small object size

# FCN's Limitation

- FCN has a predefined fixed-size receptive field. Therefore, the object that is substantially larger or smaller than the receptive field may be fragmented or mislabeled
- The detailed structures of an object are often lost or smoothed because the label map is too coarse and deconvolution procedure is overly simple



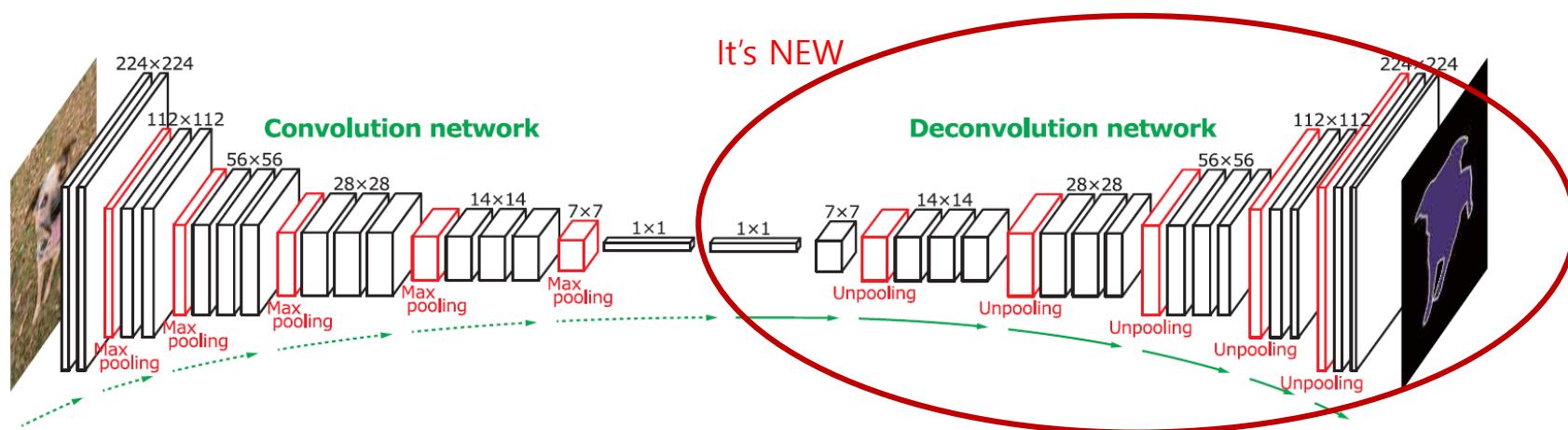
(a) Inconsistent labels due to large object size



(b) Missing labels due to small object size

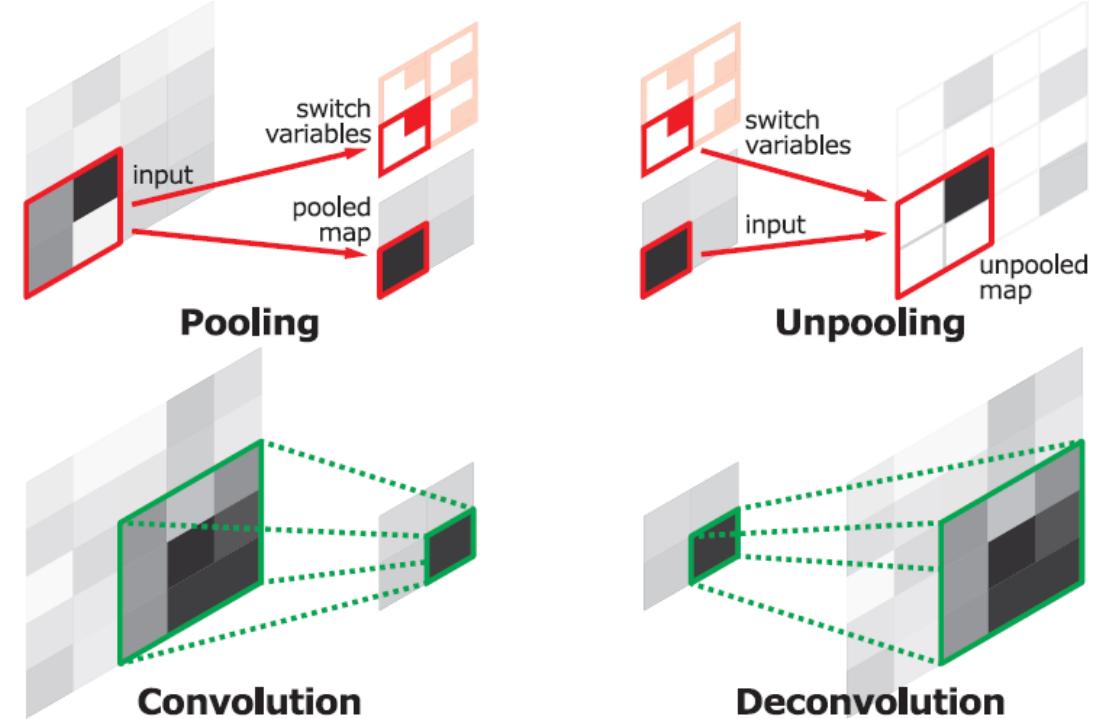
# Deconvolution Network

- To address such issues, Use “**Deconvolution**”!
- Convolution network extract features(VGG-16)
- Deconvolution network generate probability map(same size to input image)
- Probability map indicates probability of each pixel belongs to one of class



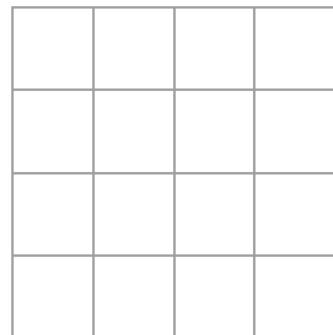
# Unpooling and Deconvolution

- Unpooling
  - Reconstruct structure of original activation map
  - Activation size is preserved but still sparse
- Deconvolution
  - Densify sparse activation map

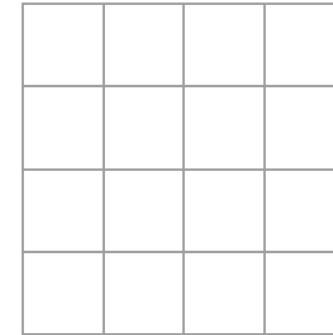


# Learnable Upsampling: “Deconvolution”

Typical  $3 \times 3$  convolution, stride 1 pad 1



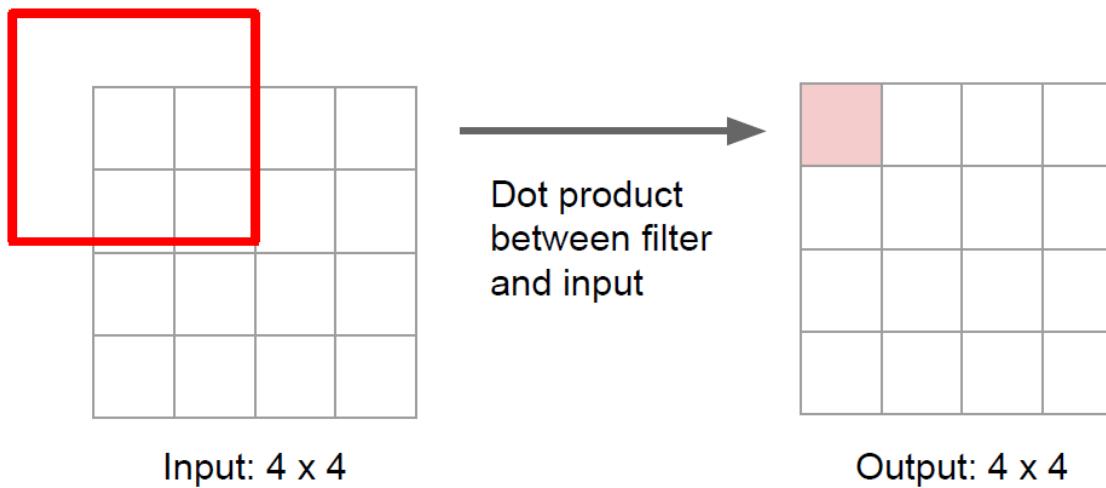
Input:  $4 \times 4$



Output:  $4 \times 4$

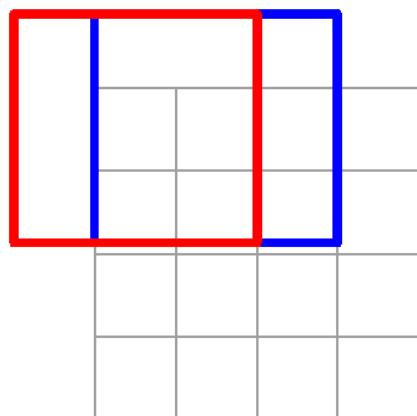
# Learnable Upsampling: “Deconvolution”

Typical  $3 \times 3$  convolution, stride 1 pad 1

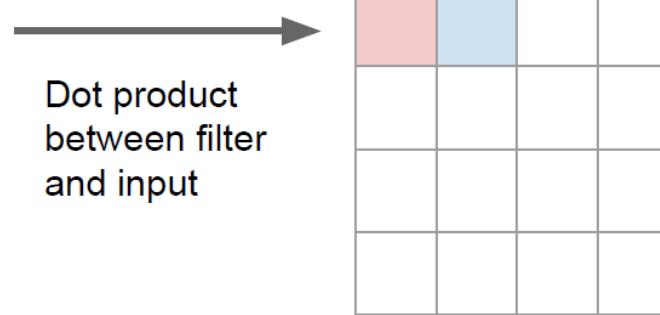


# Learnable Upsampling: “Deconvolution”

Typical  $3 \times 3$  convolution, stride 1 pad 1



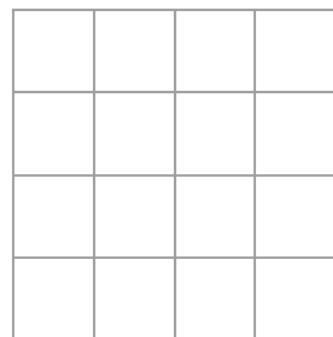
Input:  $4 \times 4$



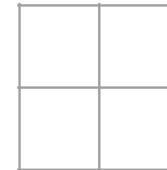
Output:  $4 \times 4$

# Learnable Upsampling: “Deconvolution”

Typical  $3 \times 3$  convolution, **stride 2** pad 1



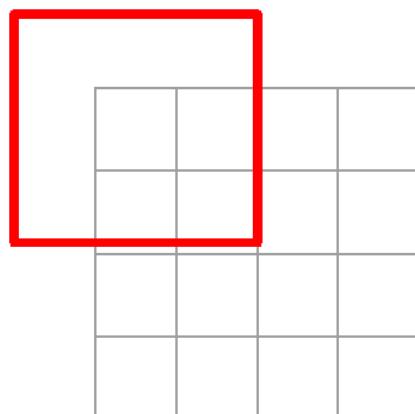
Input:  $4 \times 4$



Output:  $2 \times 2$

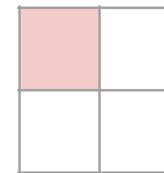
# Learnable Upsampling: “Deconvolution”

Typical  $3 \times 3$  convolution, stride 2 pad 1



Input:  $4 \times 4$

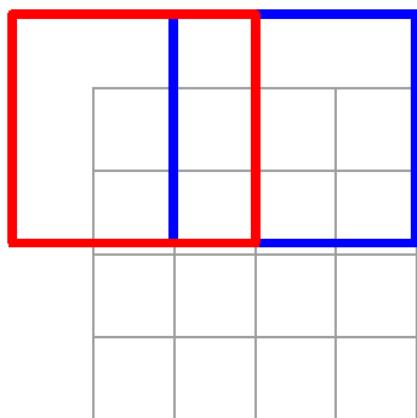
Dot product  
between filter  
and input



Output:  $2 \times 2$

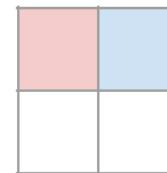
# Learnable Upsampling: “Deconvolution”

Typical  $3 \times 3$  convolution, stride 2 pad 1



Input:  $4 \times 4$

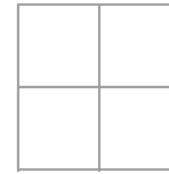
Dot product  
between filter  
and input



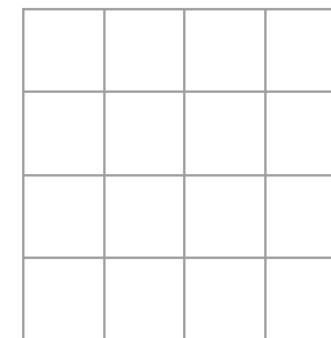
Output:  $2 \times 2$

# Learnable Upsampling: “Deconvolution”

3 x 3 “deconvolution”, stride 2 pad 1

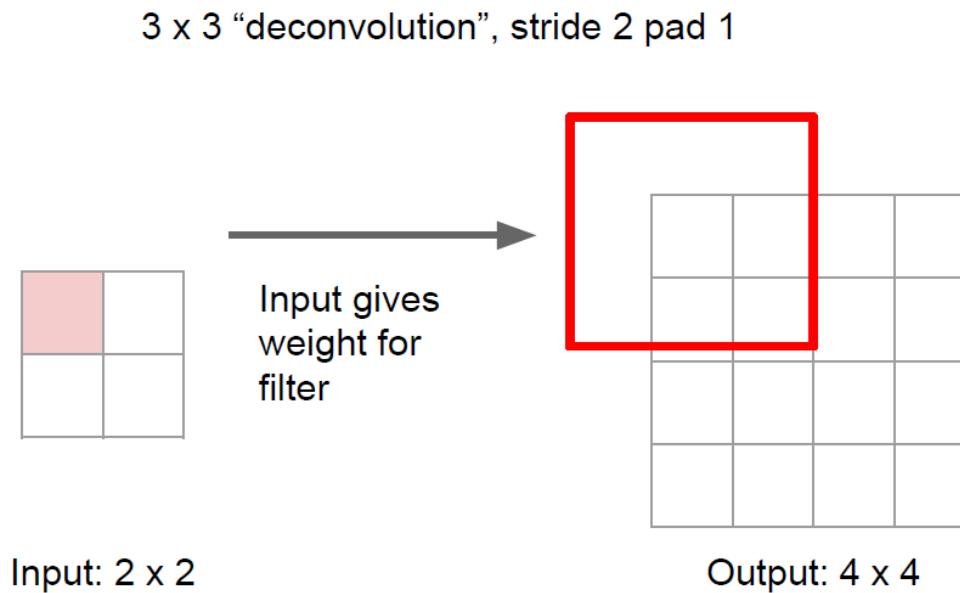


Input: 2 x 2

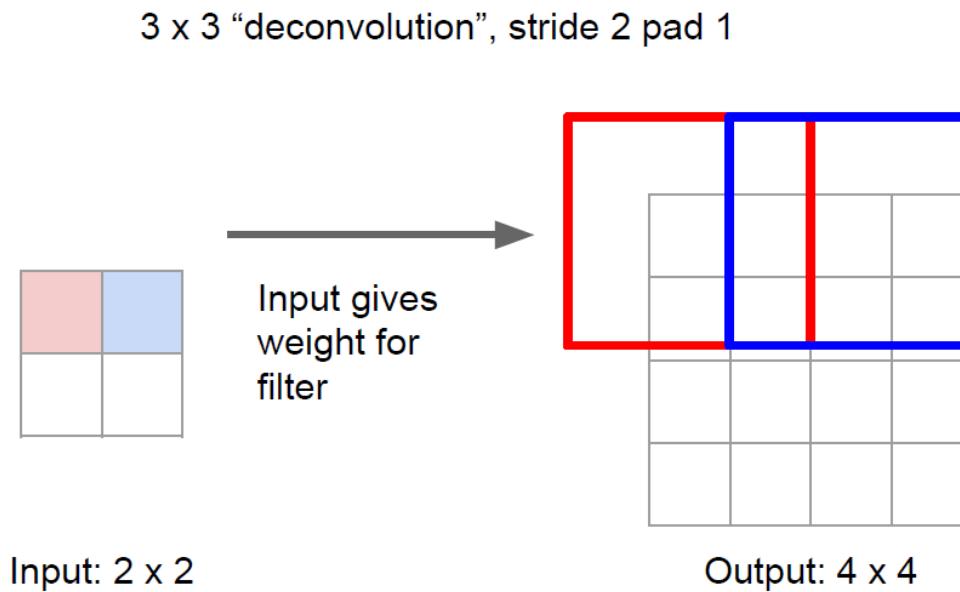


Output: 4 x 4

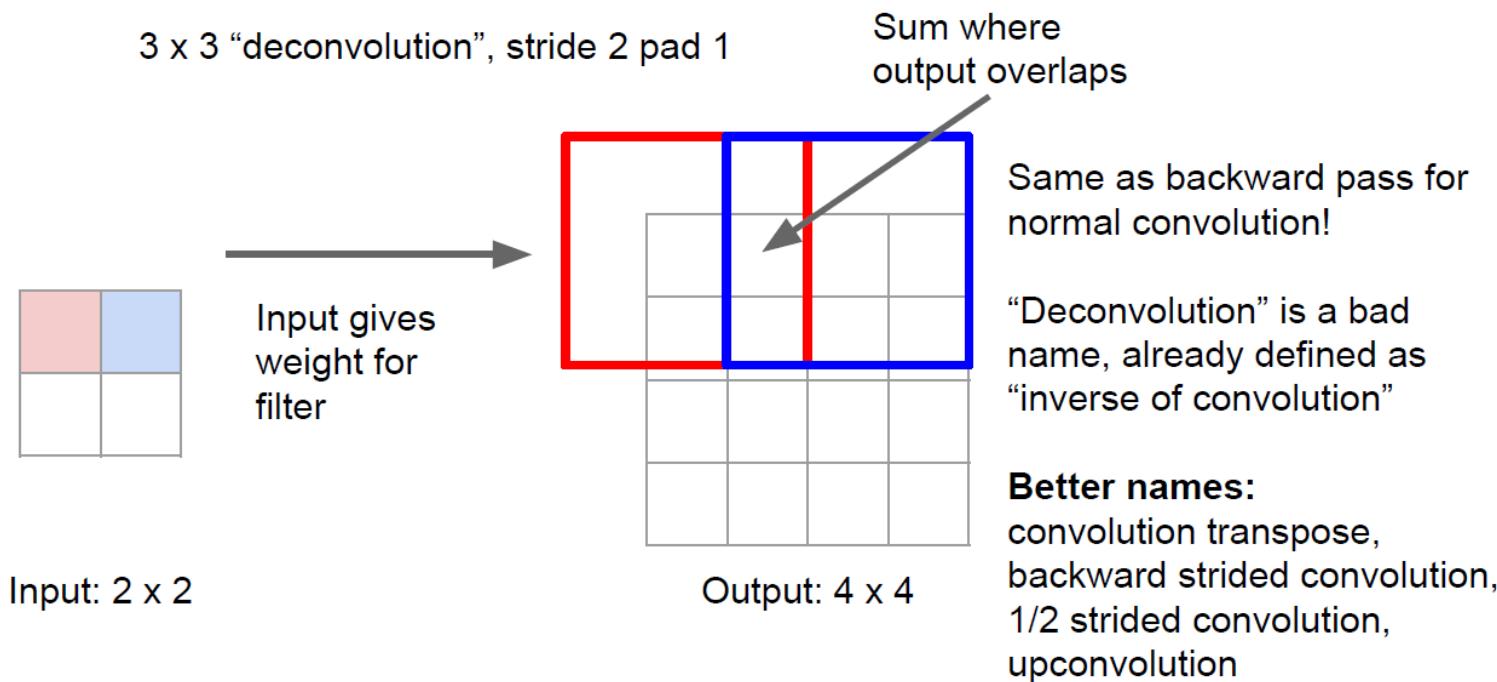
# Learnable Upsampling: “Deconvolution”



# Learnable Upsampling: “Deconvolution”



# Learnable Upsampling: “Deconvolution”



# Analysis of DeconvNet

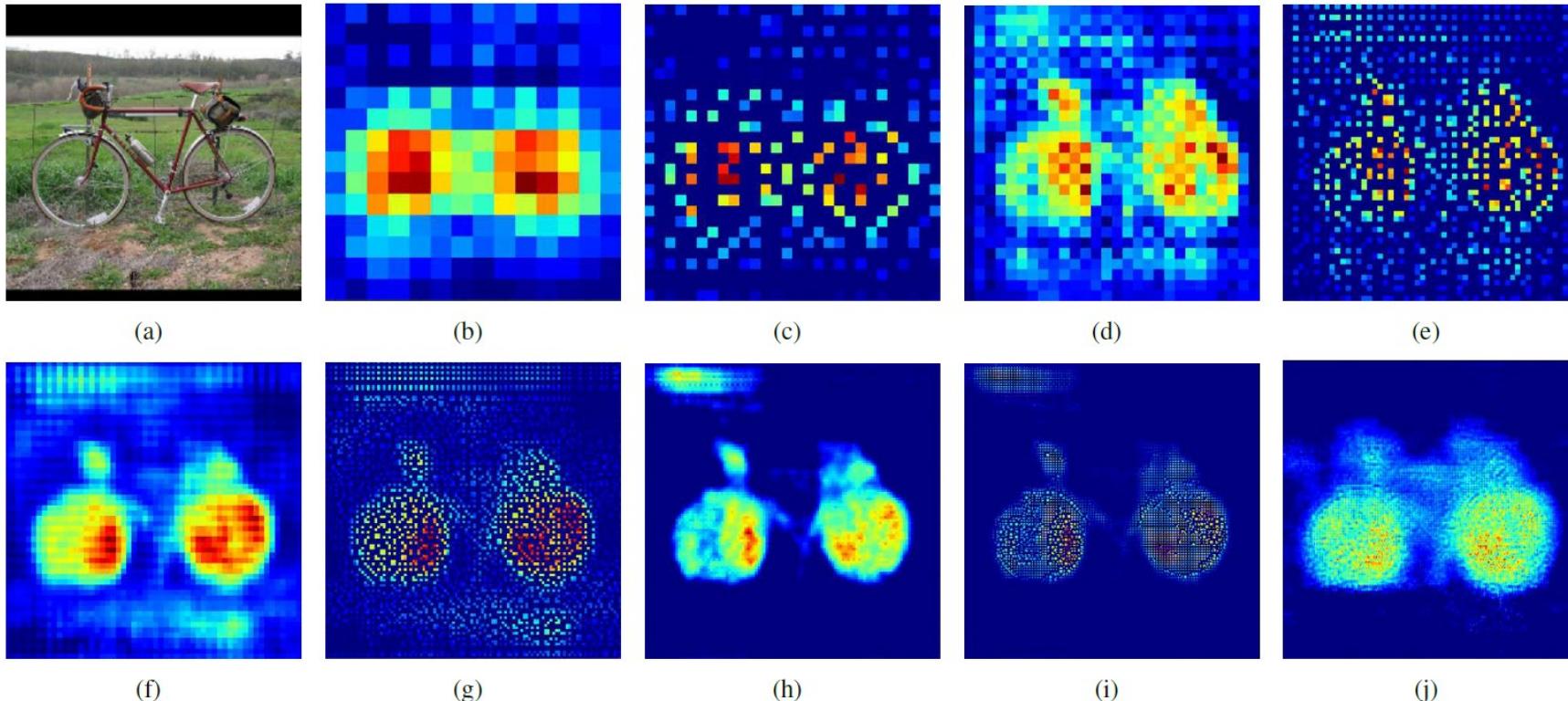


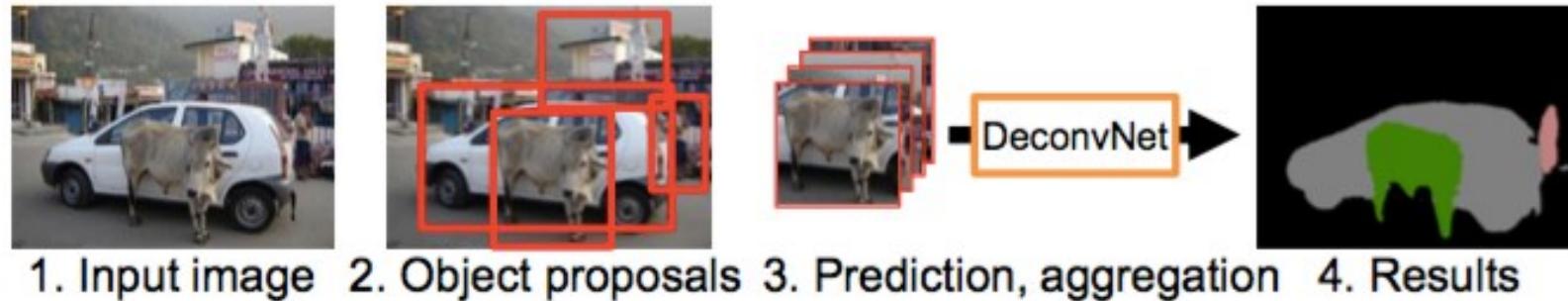
Figure 4. Visualization of activations in our deconvolution network. The activation maps from (b) to (j) correspond to the output maps from lower to higher layers in the deconvolution network. We select the most representative activation in each layer for effective visualization. The image in (a) is an input, and the rest are the outputs from (b) the last  $14 \times 14$  deconvolutional layer, (c) the  $28 \times 28$  unpooling layer, (d) the last  $28 \times 28$  deconvolutional layer, (e) the  $56 \times 56$  unpooling layer, (f) the last  $56 \times 56$  deconvolutional layer, (g) the  $112 \times 112$  unpooling layer, (h) the last  $112 \times 112$  deconvolutional layer, (i) the  $224 \times 224$  unpooling layer and (j) the last  $224 \times 224$  deconvolutional layer. The finer details of the object are revealed, as the features are forward-propagated through the layers in the deconvolution network. Note that noisy activations from background are suppressed through propagation while the activations closely related to the target classes are amplified. It shows that the learned filters in higher deconvolutional layers tend to capture class-specific shape information.

# More details of DeconvNet

- Instance-wise segmentation
- Use batch normalization
- Two-stage training
- Ensemble with FCN
  - FCN, DeconvNet are complementary relationship
- Use CRF

# Instance-wise Segmentation

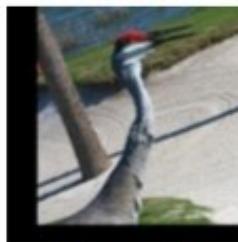
- Input proposal instances in network (not entire image)
- Get proposal instance using Edgebox algorithm
- Identify more details of object with multi scale
- Reduce search space, so can reduce memory at train



Slide from NamHyuk Ahn

# Two-stage Training

- DeconvNet has lots of parameters, but don't have many segmentation data(10K in PASCAL VOC)
  - Use two-stage training to address this issue
  - First stage: Input center-cropped images
  - Second stage: Input proposal sub-images
- So network generalize better



1<sup>st</sup> stage



2<sup>nd</sup> stage

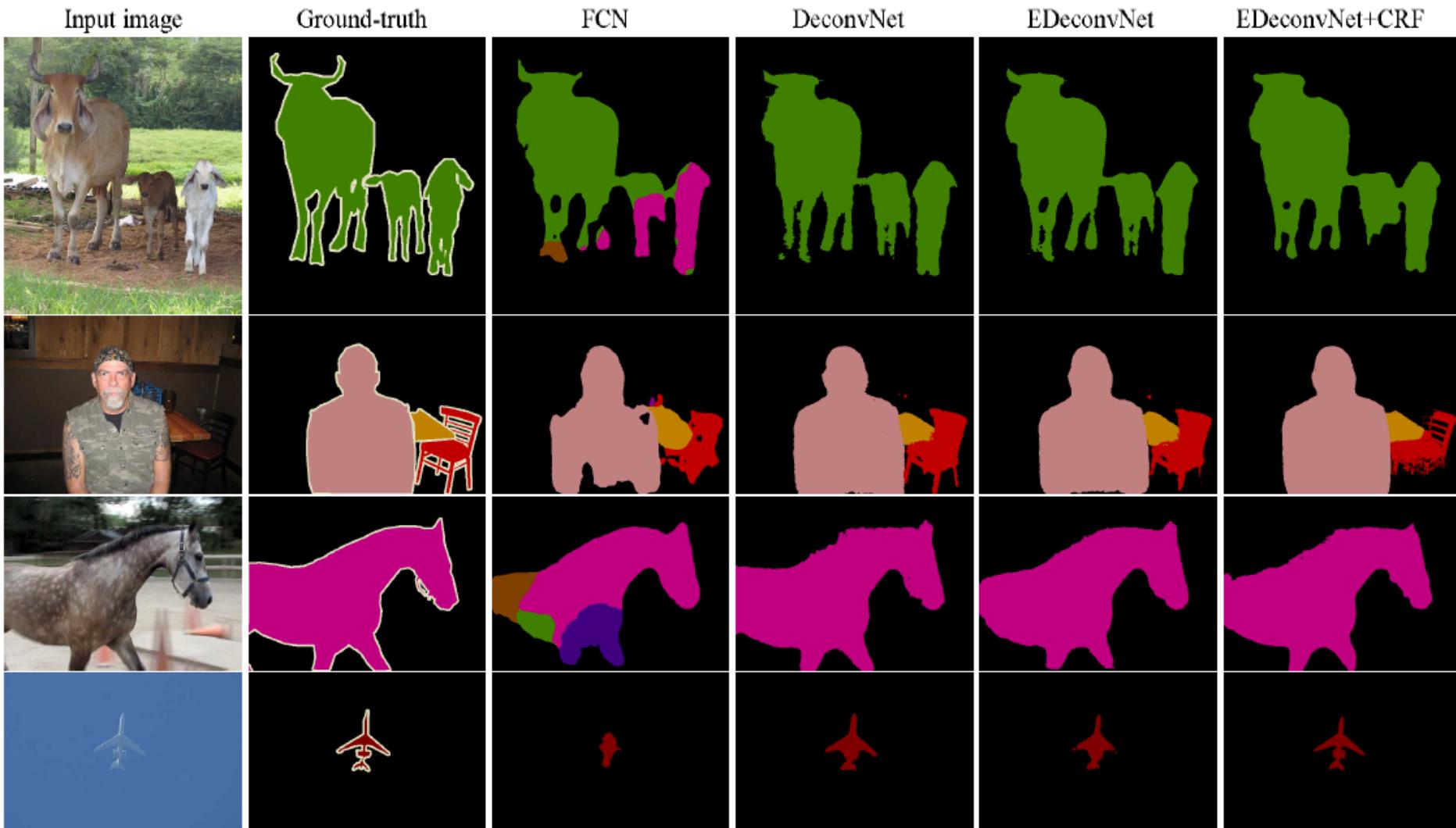
Slide from NamHyuk Ahn

# Results

Table 1. Evaluation results on PASCAL VOC 2012 test set. (Asterisk (\*) denotes the algorithms that also use Microsoft COCO for training.)

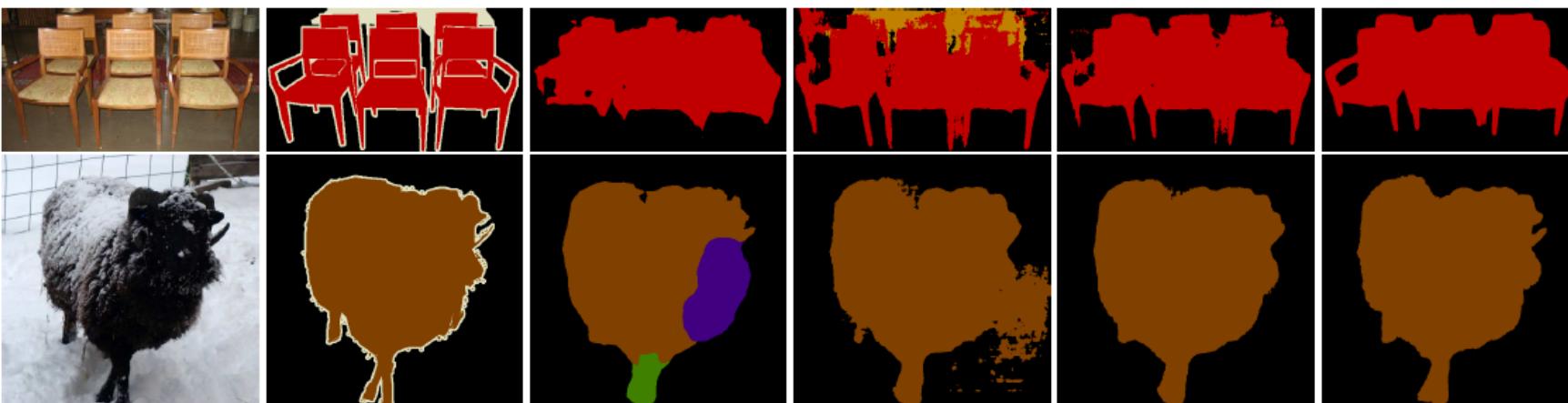
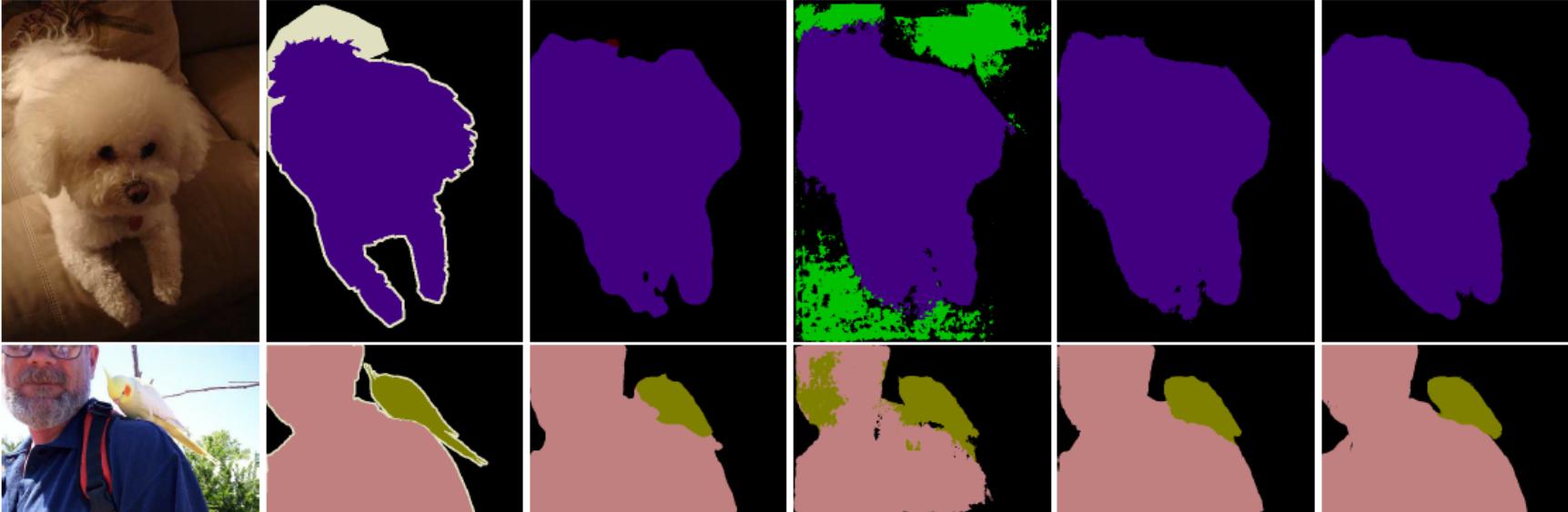
Method	bkg	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mean
Hypercolumn [11]	88.9	68.4	27.2	68.2	47.6	61.7	76.9	72.1	71.1	24.3	59.3	44.8	62.7	59.4	73.5	70.6	52.0	63.0	38.1	60.0	54.1	59.2
MSRA-CFM [3]	87.7	75.7	26.7	69.5	48.8	65.6	81.0	69.2	73.3	30.0	68.7	51.5	69.1	68.1	71.7	67.5	50.4	66.5	44.4	58.9	53.5	61.8
FCN8s [19]	91.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
TTI-Zoomout-16 [20]	89.8	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3	64.4
DeepLab-CRF [1]	93.1	84.4	<b>54.5</b>	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	<b>59.7</b>	82.2	50.4	73.1	63.7	71.6
DeconvNet	92.7	85.9	42.6	78.9	62.5	66.6	87.4	77.8	79.5	26.3	73.4	60.2	70.8	76.5	79.6	77.7	58.2	77.4	52.9	75.2	59.8	69.6
DeconvNet+CRF	92.9	87.8	41.9	80.6	63.9	67.3	<b>88.1</b>	78.4	81.3	25.9	73.7	61.2	72.0	77.0	79.9	78.7	59.5	78.3	<b>55.0</b>	75.2	61.5	70.5
EDeconvNet	92.9	88.4	39.7	79.0	63.0	67.7	87.1	<b>81.5</b>	84.4	27.8	76.1	61.2	78.0	79.3	83.1	79.3	58.0	82.5	52.3	80.1	64.0	71.7
EDeconvNet+CRF	93.1	<b>89.9</b>	39.3	79.7	63.9	<b>68.2</b>	87.4	81.2	<b>86.1</b>	28.5	<b>77.0</b>	62.0	79.0	<b>80.3</b>	<b>83.6</b>	80.2	58.8	<b>83.4</b>	54.3	<b>80.7</b>	65.0	<b>72.5</b>
* WSSL [21]	93.2	85.3	36.2	<b>84.8</b>	61.2	67.5	84.7	81.4	81.0	<b>30.8</b>	73.8	53.8	77.5	76.5	82.3	<b>81.6</b>	56.3	78.9	52.3	76.6	63.3	70.4
* BoxSup [2]	<b>93.6</b>	86.4	35.5	79.7	<b>65.2</b>	65.2	84.3	78.5	83.7	30.5	76.2	<b>62.6</b>	<b>79.3</b>	76.1	82.1	81.3	57.0	78.2	<b>55.0</b>	72.5	<b>68.1</b>	71.0

# Results



(a) Examples that our method produces better results than FCN [19].

# Results



# Multi-Scale Context-Aggregation by Dilated Convolutions

## MULTI-SCALE CONTEXT AGGREGATION BY DILATED CONVOLUTIONS

**Fisher Yu**

Princeton University

**Vladlen Koltun**

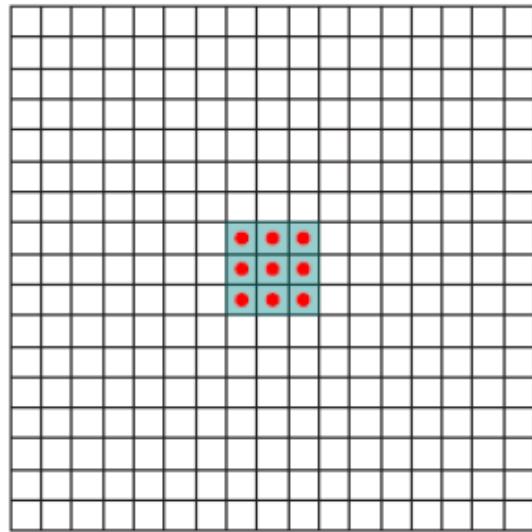
Intel Labs

[cs.CV] 30 Apr 2016

### ABSTRACT

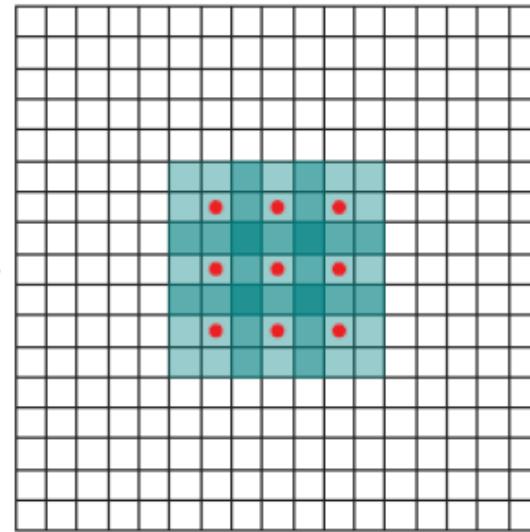
State-of-the-art models for semantic segmentation are based on adaptations of convolutional networks that had originally been designed for image classification. However, dense prediction problems such as semantic segmentation are structurally different from image classification. In this work, we develop a new convolutional network module that is specifically designed for dense prediction. The presented module uses dilated convolutions to systematically aggregate multi-scale contextual information without losing resolution. The architecture is based on the fact that dilated convolutions support exponential expansion of the receptive field without loss of resolution or coverage. We show that the presented context module increases the accuracy of state-of-the-art semantic segmentation systems. In addition, we examine the adaptation of image classification networks to dense prediction and show that simplifying the adapted network can increase accuracy.

# Dilated Convolution



3x3 Conv r=1

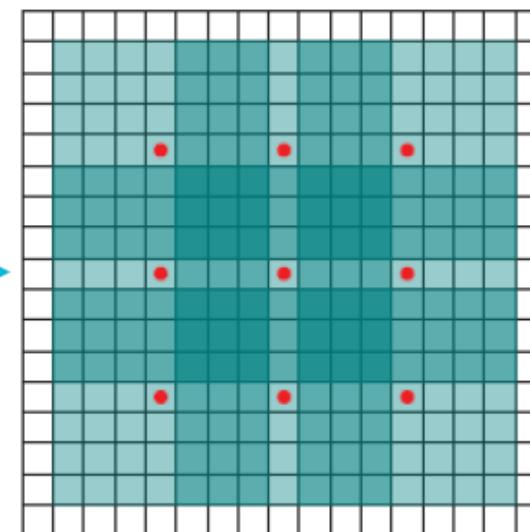
3x3 Range



3x3 Conv r=1

3x3 Conv r=2

7x7 Range



3x3 Conv r=1

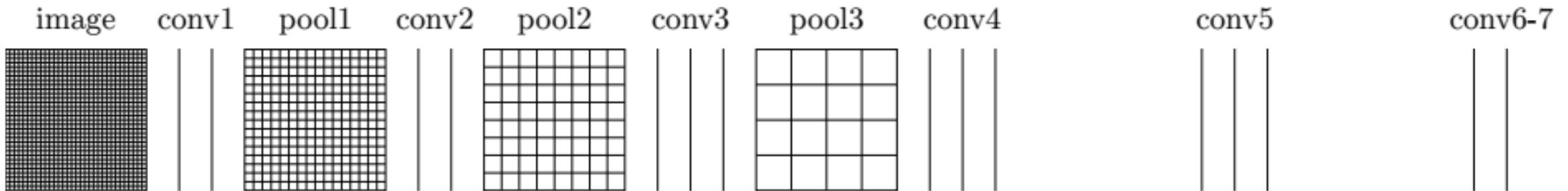
3x3 Conv r=2

3x3 Conv r=4

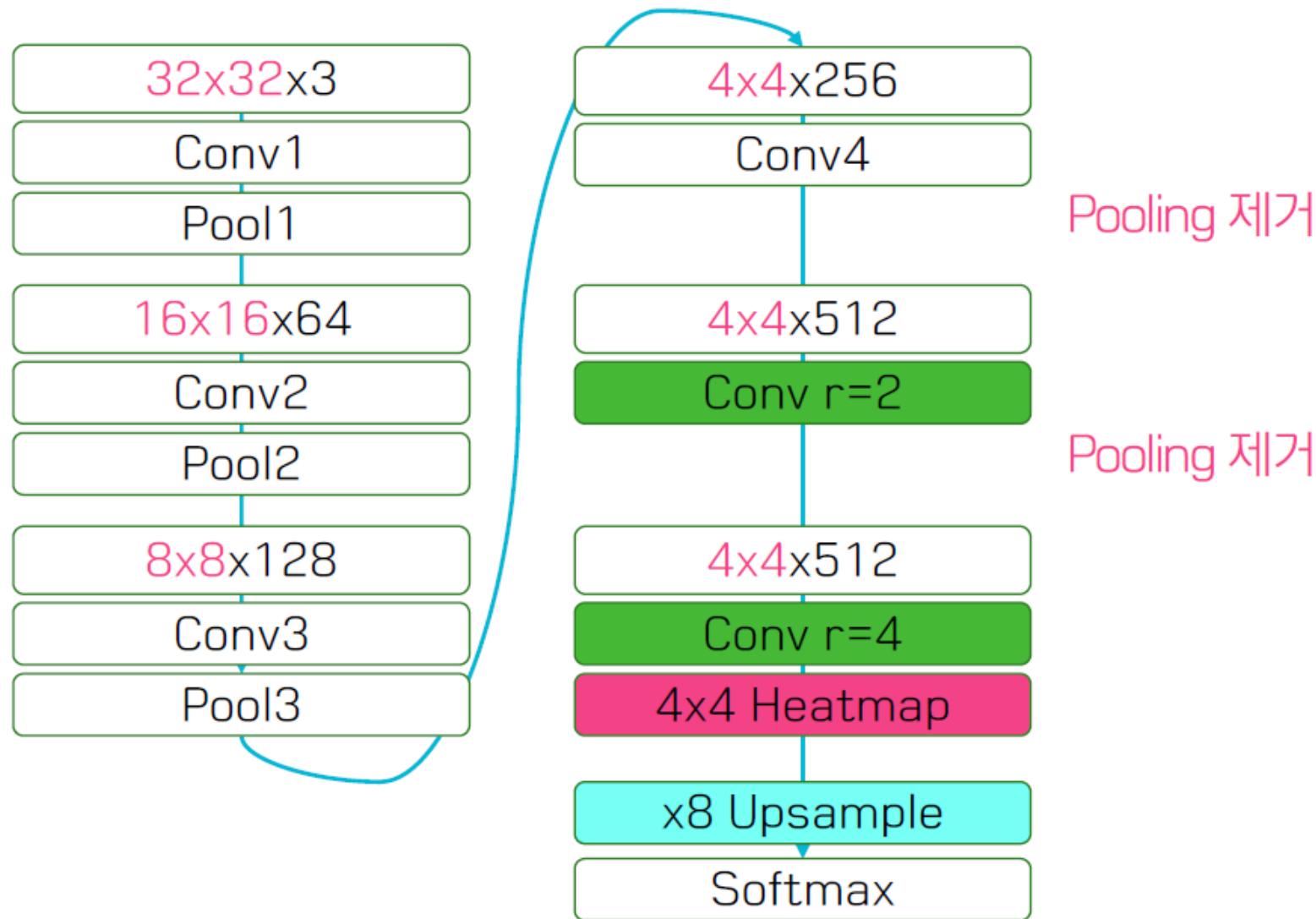
15x15 Range

# Front-End Module

- FCN에서 pool4, pool5 제거
- conv5 → 2-dilated convolution
- conv6 → 4-dilated convolution



# Front-End Module

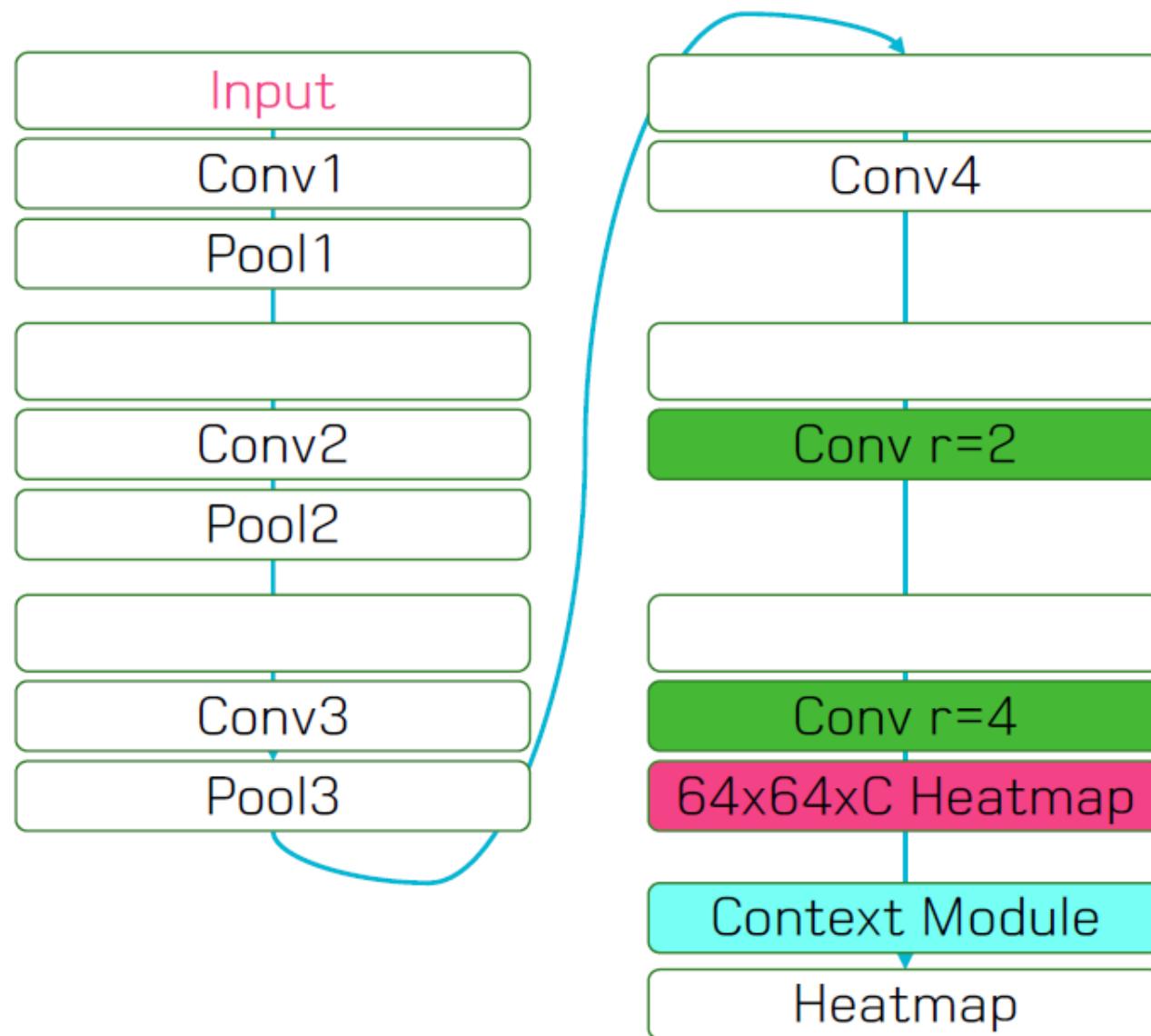


# Context Network

Layer	1	2	3	4	5	6	7	8
Convolution	$3 \times 3$	$3 \times 3$	$3 \times 3$	$3 \times 3$	$3 \times 3$	$3 \times 3$	$3 \times 3$	$1 \times 1$
Dilation	1	1	2	4	8	16	1	1
Truncation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Receptive field	$3 \times 3$	$5 \times 5$	$9 \times 9$	$17 \times 17$	$33 \times 33$	$65 \times 65$	$67 \times 67$	$67 \times 67$
Output channels								
Basic	$C$	$C$	$C$	$C$	$C$	$C$	$C$	$C$
Large	$2C$	$2C$	$4C$	$8C$	$16C$	$32C$	$32C$	$C$

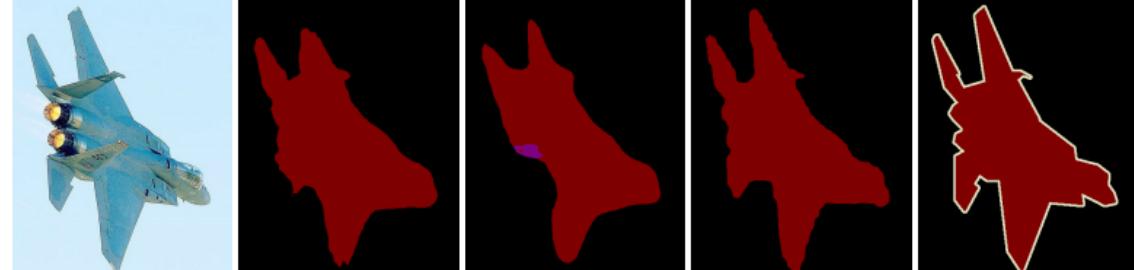
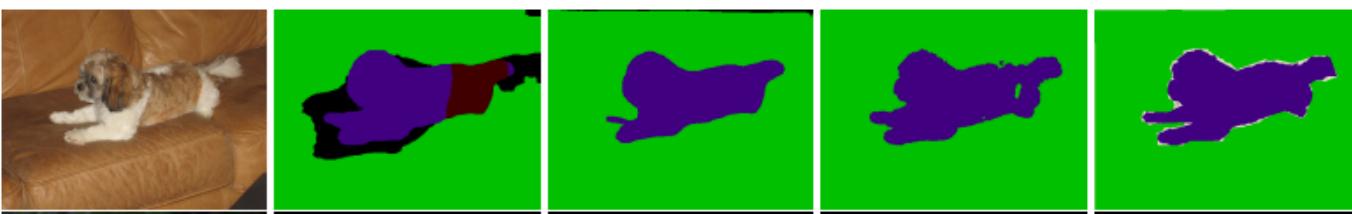
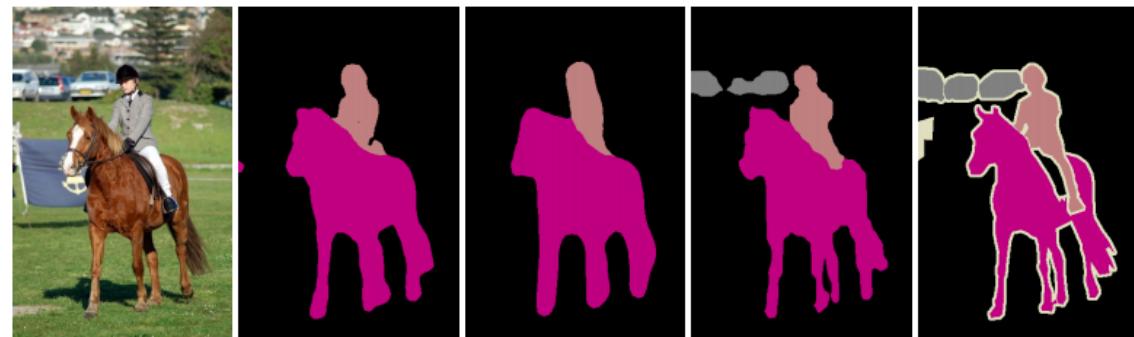
Table 1: Context network architecture. The network processes  $C$  feature maps by aggregating contextual information at progressively increasing scales without losing resolution.

# Front-End Module + Context Module



Size에 맞게 Input Padding  
C = 21 (Class 개수)

# Results



(a) Image

(b) FCN-8s

(c) DeepLab

(d) Our front end

(e) Ground truth

(a) Image

(b) Front end

(c) + Context

(d) + CRF-RNN

(e) Ground truth

# DeepLab(v2)

## DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

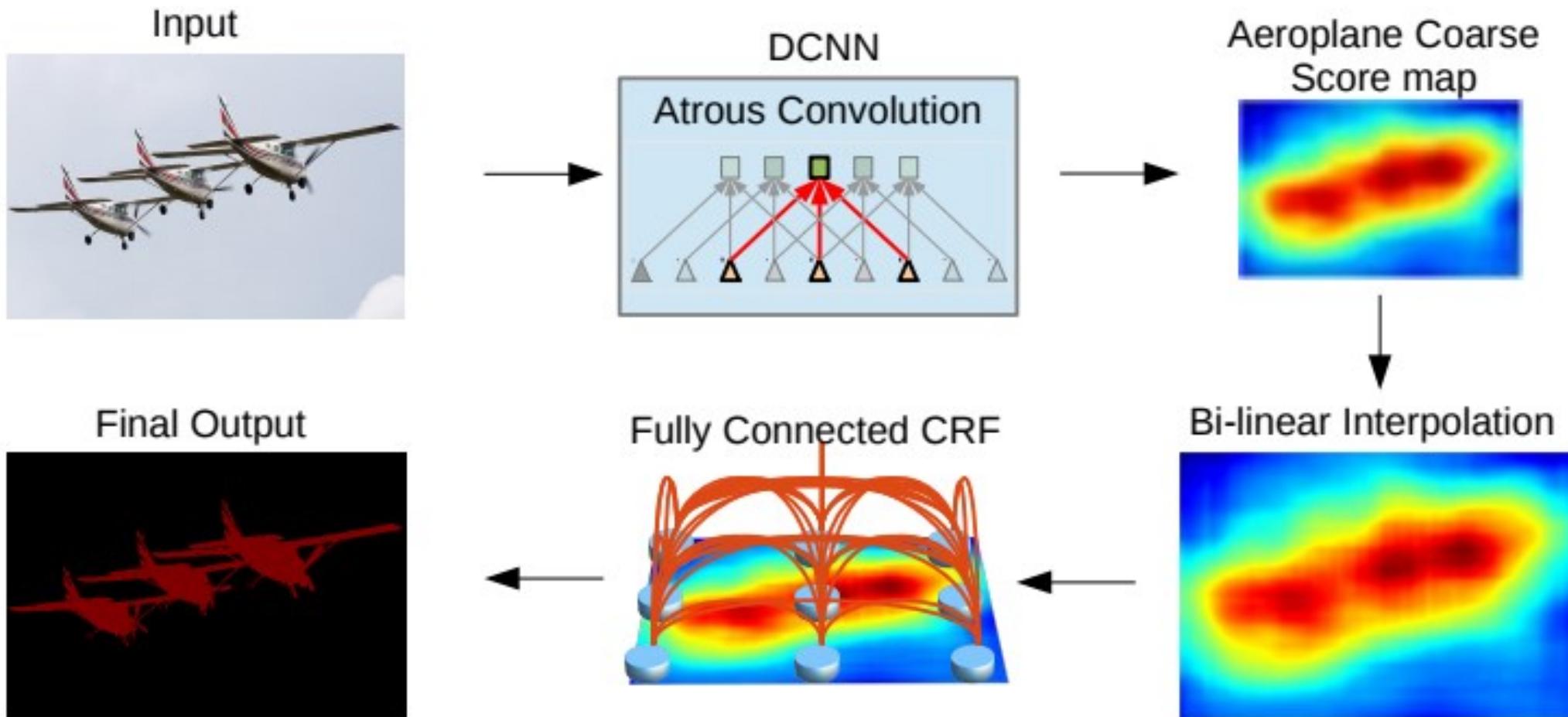
Liang-Chieh Chen, George Papandreou, *Senior Member, IEEE*, Iasonas Kokkinos, *Member, IEEE*, Kevin Murphy, and Alan L. Yuille, *Fellow, IEEE*

**Abstract**—In this work we address the task of semantic image segmentation with Deep Learning and make three main contributions that are experimentally shown to have substantial practical merit. *First*, we highlight convolution with upsampled filters, or ‘atrous convolution’, as a powerful tool in dense prediction tasks. Atrous convolution allows us to explicitly control the resolution at which feature responses are computed within Deep Convolutional Neural Networks. It also allows us to effectively enlarge the field of view of filters to incorporate larger context without increasing the number of parameters or the amount of computation. *Second*, we propose atrous spatial pyramid pooling (ASPP) to robustly segment objects at multiple scales. ASPP probes an incoming convolutional feature layer with filters at multiple sampling rates and effective fields-of-views, thus capturing objects as well as image context at multiple scales. *Third*, we improve the localization of object boundaries by combining methods from DCNNs and probabilistic graphical models. The commonly deployed combination of max-pooling and downsampling in DCNNs achieves invariance but has a toll on localization accuracy. We overcome this by combining the responses at the final DCNN layer with a fully connected Conditional Random Field (CRF), which is shown both qualitatively and quantitatively to improve localization performance. Our proposed “DeepLab” system sets the new state-of-art at the PASCAL VOC-2012 semantic image segmentation task, reaching 79.7% mIOU in the test set, and advances the results on three other datasets: PASCAL-Context, PASCAL-Person-Part, and Cityscapes. All of our code is made publicly available online.

**Index Terms**—Convolutional Neural Networks, Semantic Segmentation, Atrous Convolution, Conditional Random Fields.

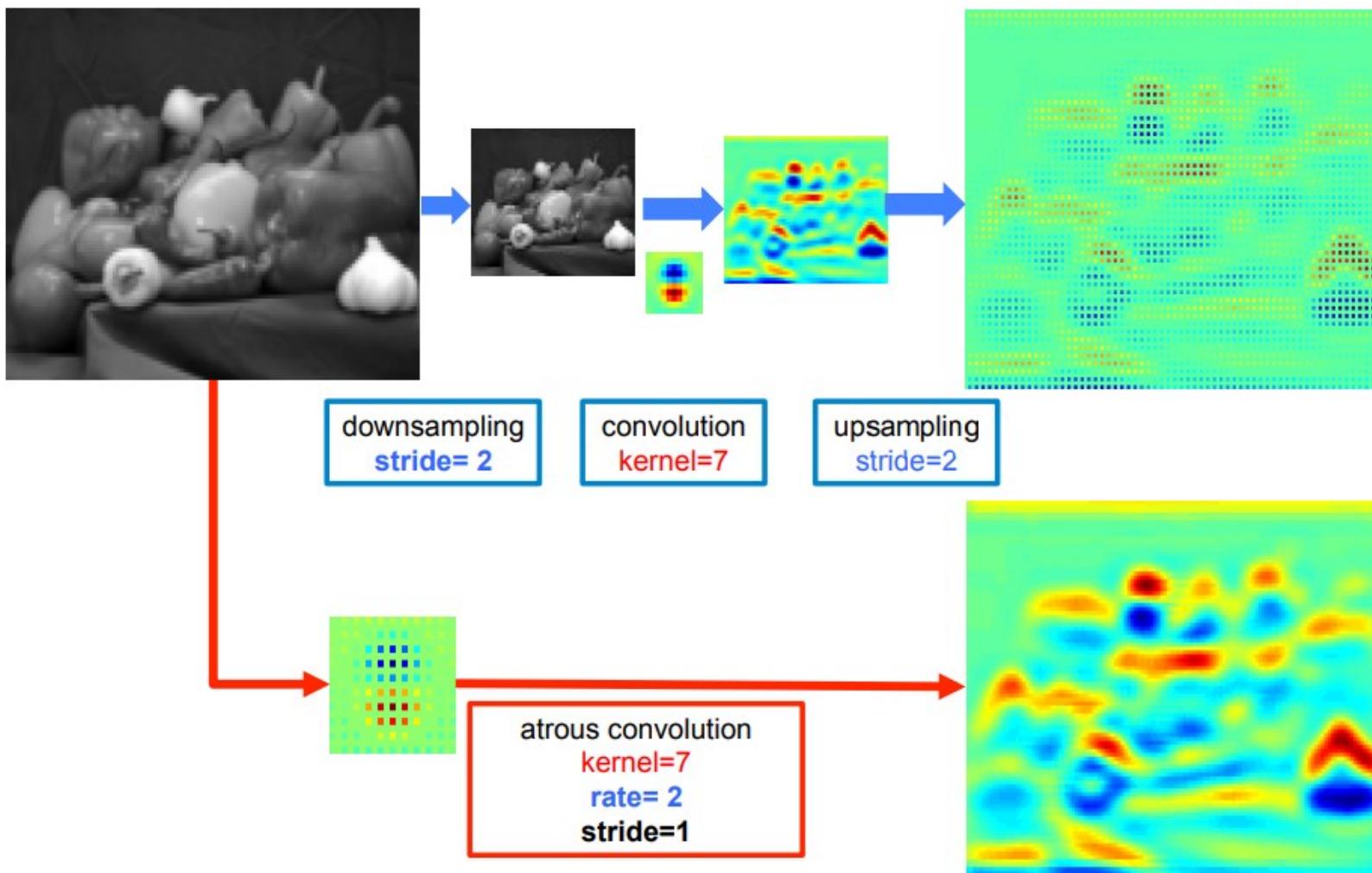
# Model Illustrartion

- Atrous Convolution + ASPP + Fully Connected CRF

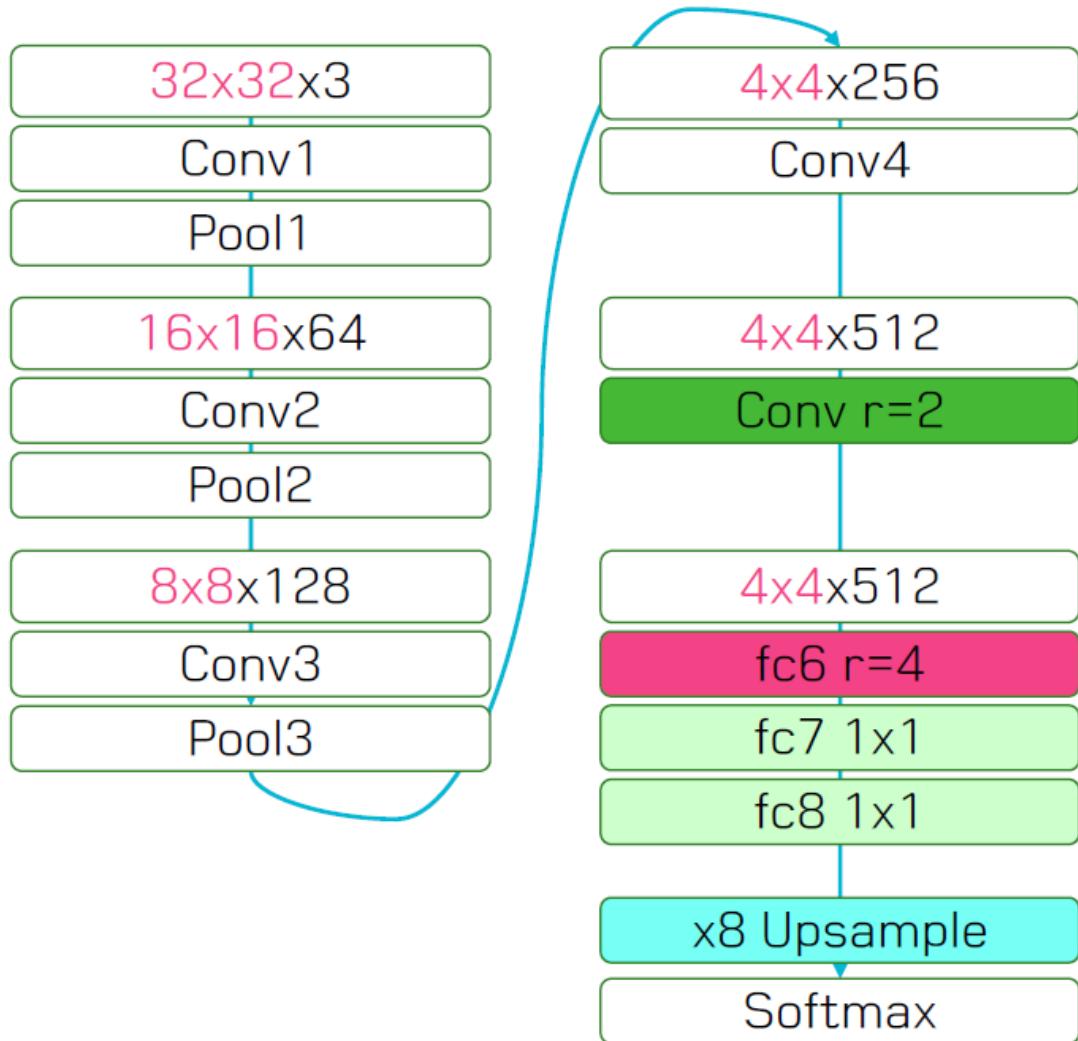


# Atrous Convolution = Dilated Convolution

- Dense Feature Extraction



# Atrous Convolution (DeepLab v1 : VGG16, DeepLab v2 : ResNet)



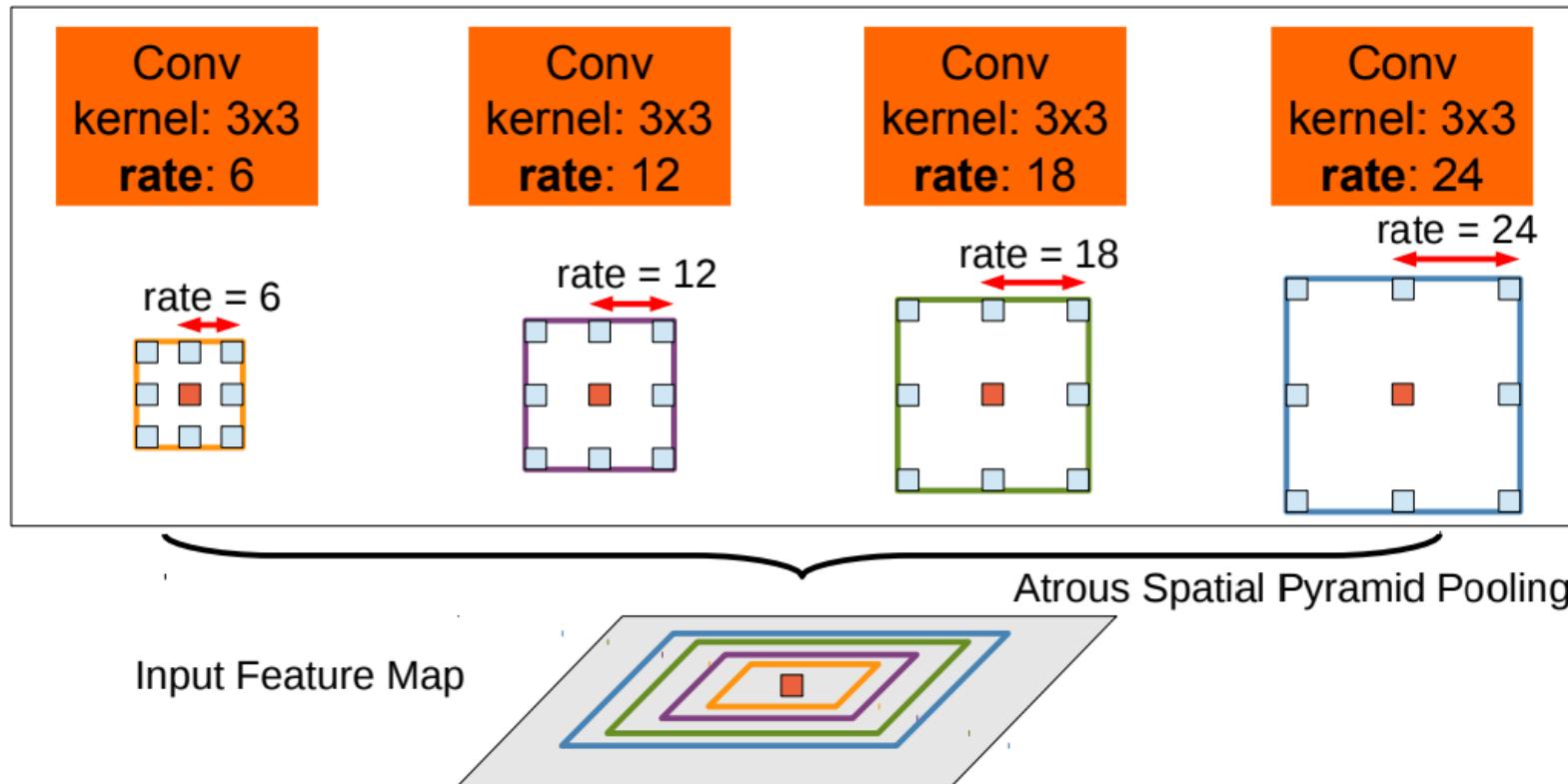
DilatedConv와 비슷 (Front-End)  
Fc6 Layer의 rate가 다름

Kernel	Rate	FOV	Params	Speed	bef/aft CRF
$7 \times 7$	4	224	134.3M	1.44	64.38 / 67.64
$4 \times 4$	4	128	65.1M	2.90	59.80 / 63.74
$4 \times 4$	8	224	65.1M	2.90	63.41 / 67.14
$3 \times 3$	12	224	20.5M	4.84	62.25 / 67.64

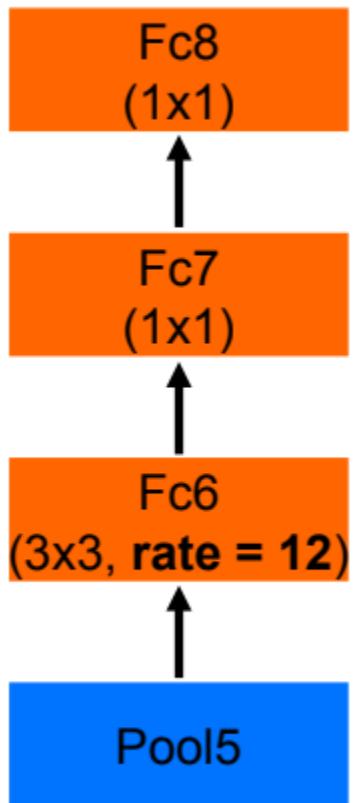
TABLE 1: Effect of Field-Of-View by adjusting the kernel size and atrous sampling rate  $r$  at 'fc6' layer. We show number of model parameters, training speed (img/sec), and *val* set mean IOU before and after CRF. DeepLab-LargeFOV (kernel size  $3 \times 3$ ,  $r = 12$ ) strikes the best balance.

# ASPP : Atrous Spatial Pyramid Pooling

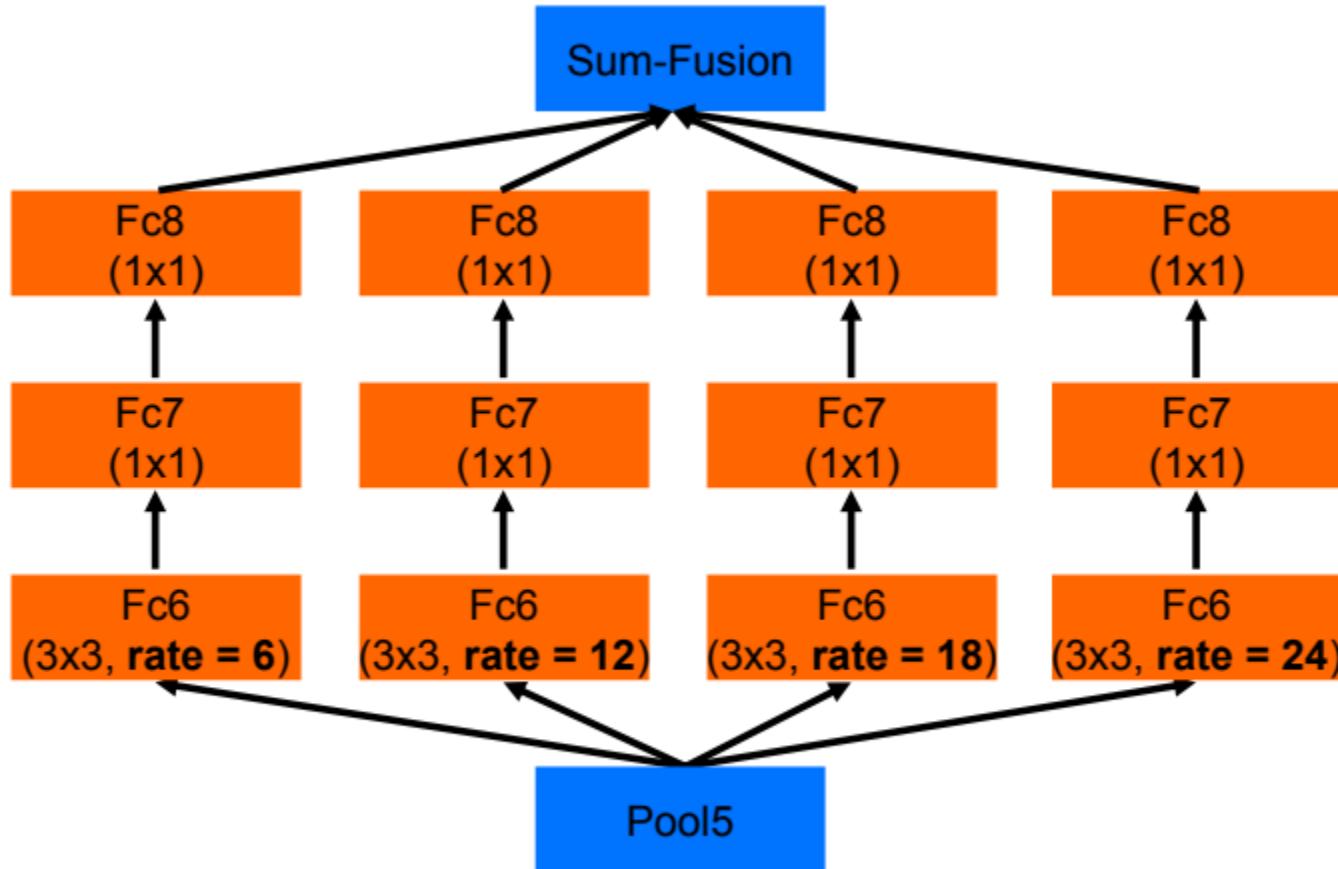
- Dilated Conv is not enough for Multi-scale view
- To classify the center pixel (orange), ASPP exploits multi-scale features by employing multiple parallel filters with different rates.



# ASPP



(a) DeepLab-LargeFOV



(b) DeepLab-ASPP

# Fully Connected Conditional Random Field

Maximize Posterior

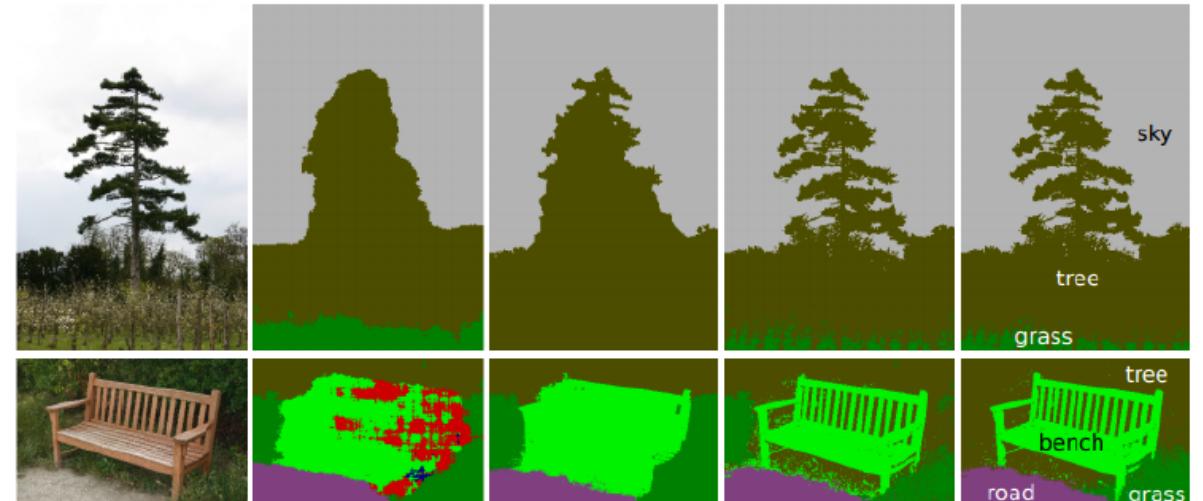
$$P(X|I) = \frac{1}{Z(I)} \exp(-\sum \phi_c(X_c|I))$$

Image      Label      Normalization

Minimize Energy

$$E(X|I) = \sum \phi_c(X_c|I)$$
$$E(X) = \sum \psi_c(X_c)$$
$$E(x) = \sum_i \psi_i(x_i) + \sum_{i,j} \psi_{i,j}(x_i, x_j)$$

Unary      Fully Connected



# Fully Connected Conditional Random Field

$$E(x) = \sum_i \psi_i(x_i) + \sum_{i,j} \psi_{i,j}(x_i, x_j)$$

Unary Term (from Classifier)  $\psi_i(x_i) = -\log P(x_i)$

Pairwise Term

$$\psi_{i,j}(x_i, x_j) = \boxed{\mu(x_i, x_j)} [w_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2}\right) - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}] + w_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\right)]$$



$$\mu(x_i, x_j) = \begin{cases} 1 & x_i \neq x_j \\ 0 & \text{Otherwise} \end{cases}$$

$p_i, p_j$ : Pixel 위치

$I_i, I_j$ : Pixel RGB값

$w_1, w_2$ : Kernel Weights

$\sigma_\alpha, \sigma_\beta, \sigma_\gamma$ : Hyper-parameter

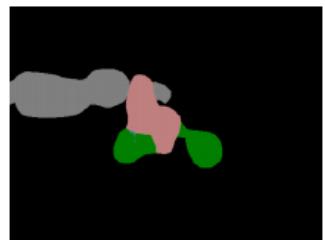
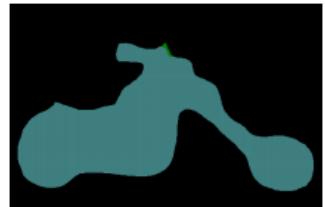
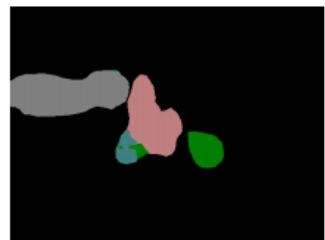
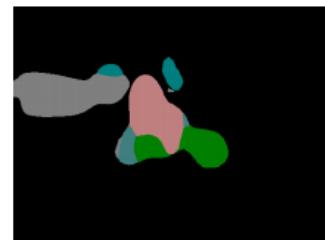
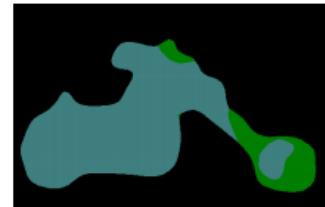
의미?

Pixel이 서로 비슷한데  
(위치적으로, RGB상으로)

Label이 서로 다르면

Energy 증가하여 Penalty

# Results

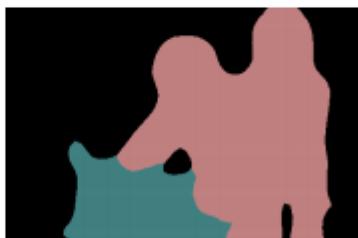
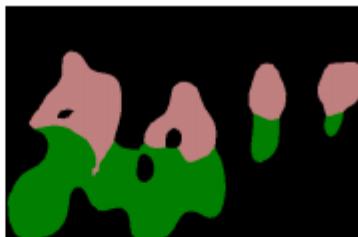


(a) Image

(b) LargeFOV

(c) ASPP-S

(d) ASPP-L



Image

VGG-16 Bef.

VGG-16 Aft.

ResNet Bef.

ResNet Aft.

# U-Net: Convolutional Networks for Biomedical Image Segmentation

## U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer, and Thomas Brox

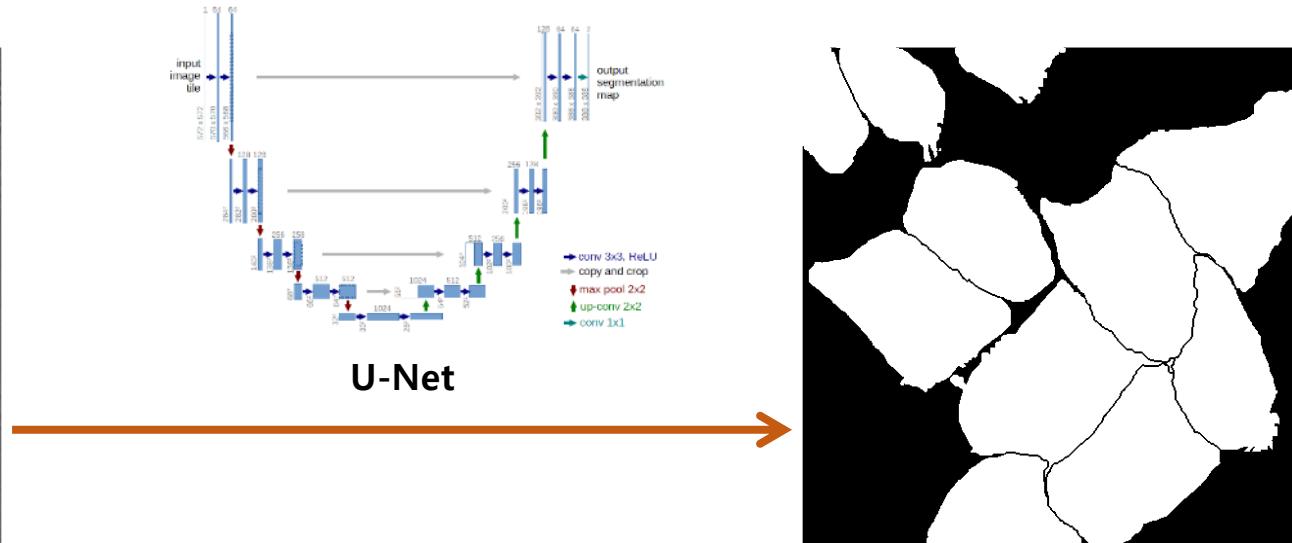
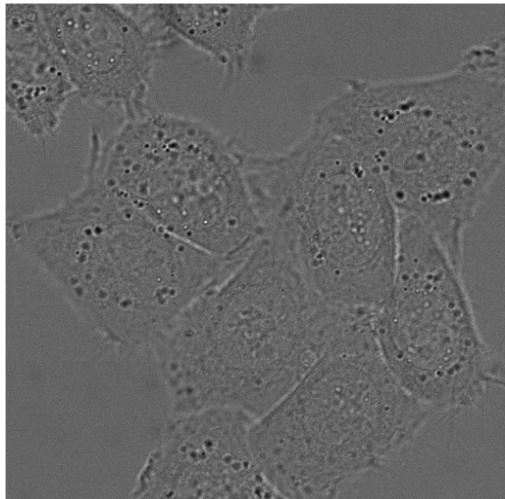
Computer Science Department and BIOSS Centre for Biological Signalling Studies,  
University of Freiburg, Germany

[ronneber@informatik.uni-freiburg.de](mailto:ronneber@informatik.uni-freiburg.de),

WWW home page: <http://lmb.informatik.uni-freiburg.de/>

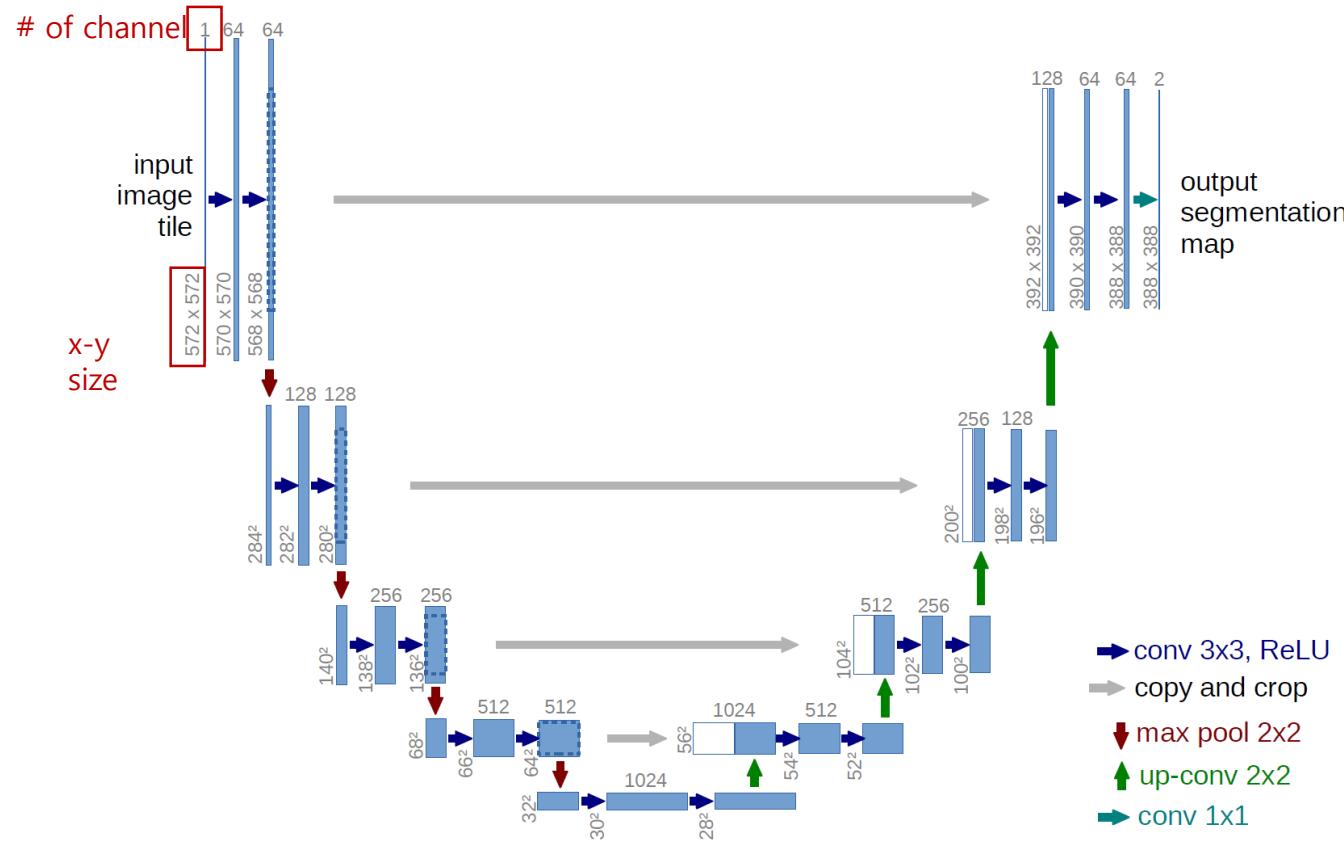
**Abstract.** There is large consent that successful training of deep networks requires many thousand annotated training samples. In this paper, we present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. We show that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding-window

# Biomedical Image Segmentation with U-net

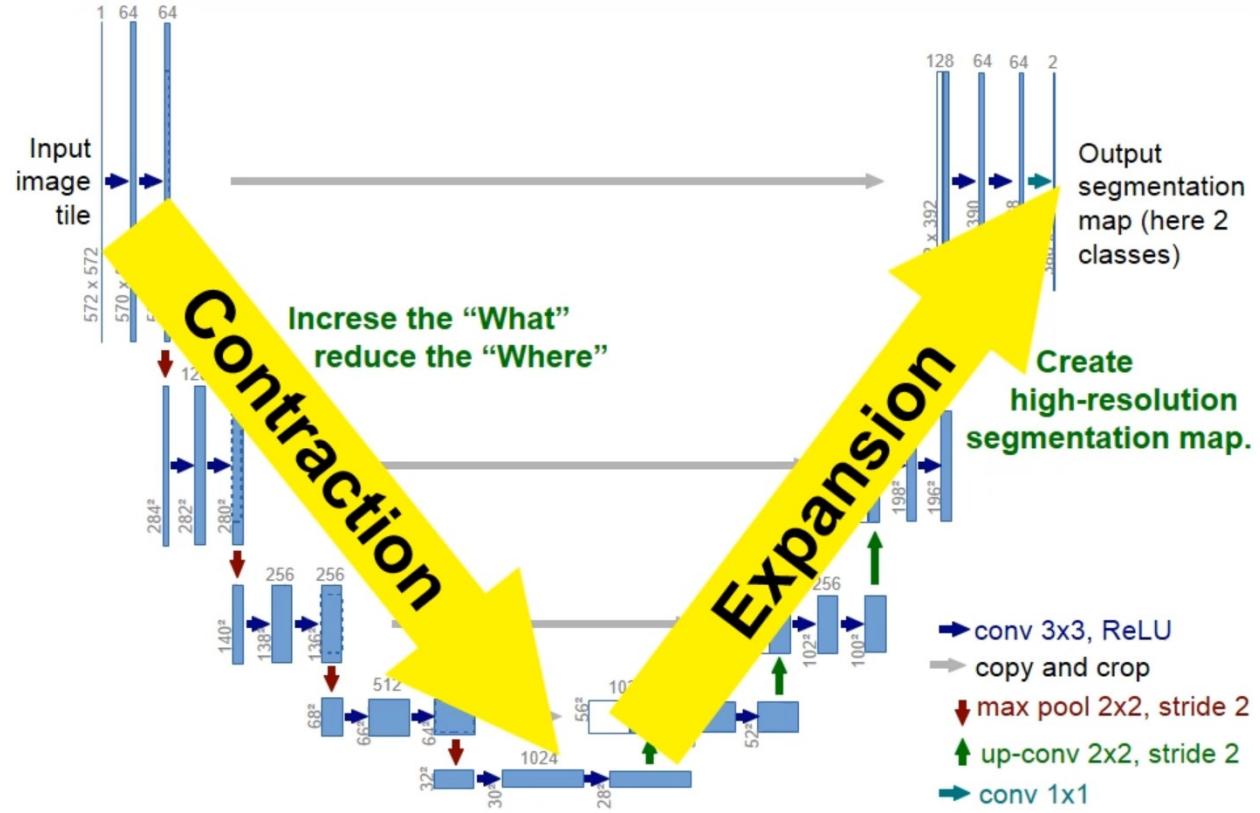


- U-Net learns segmentation in an end-to-end setting
- Very few annotated images (approx. 30 per application)

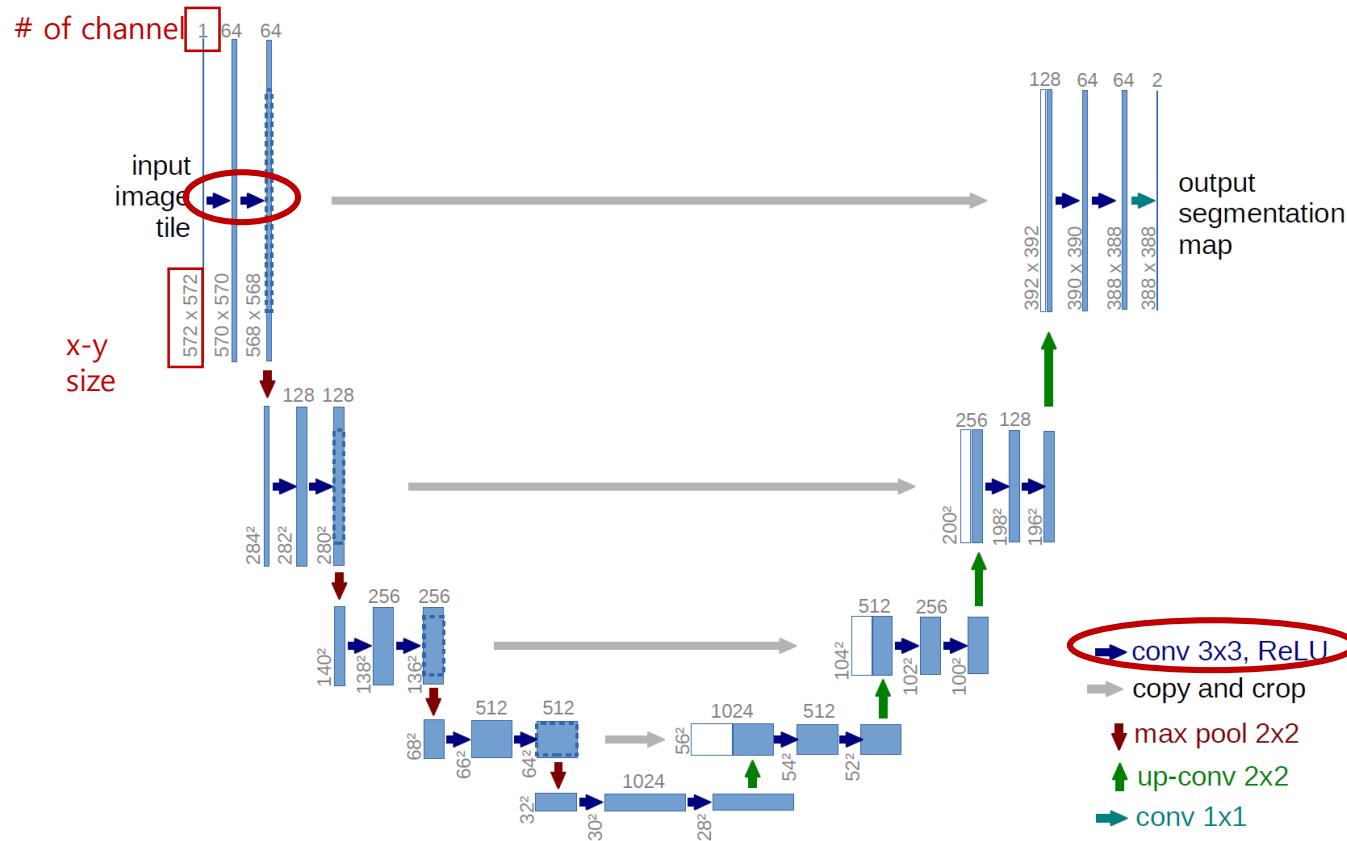
# U-Net Architecture



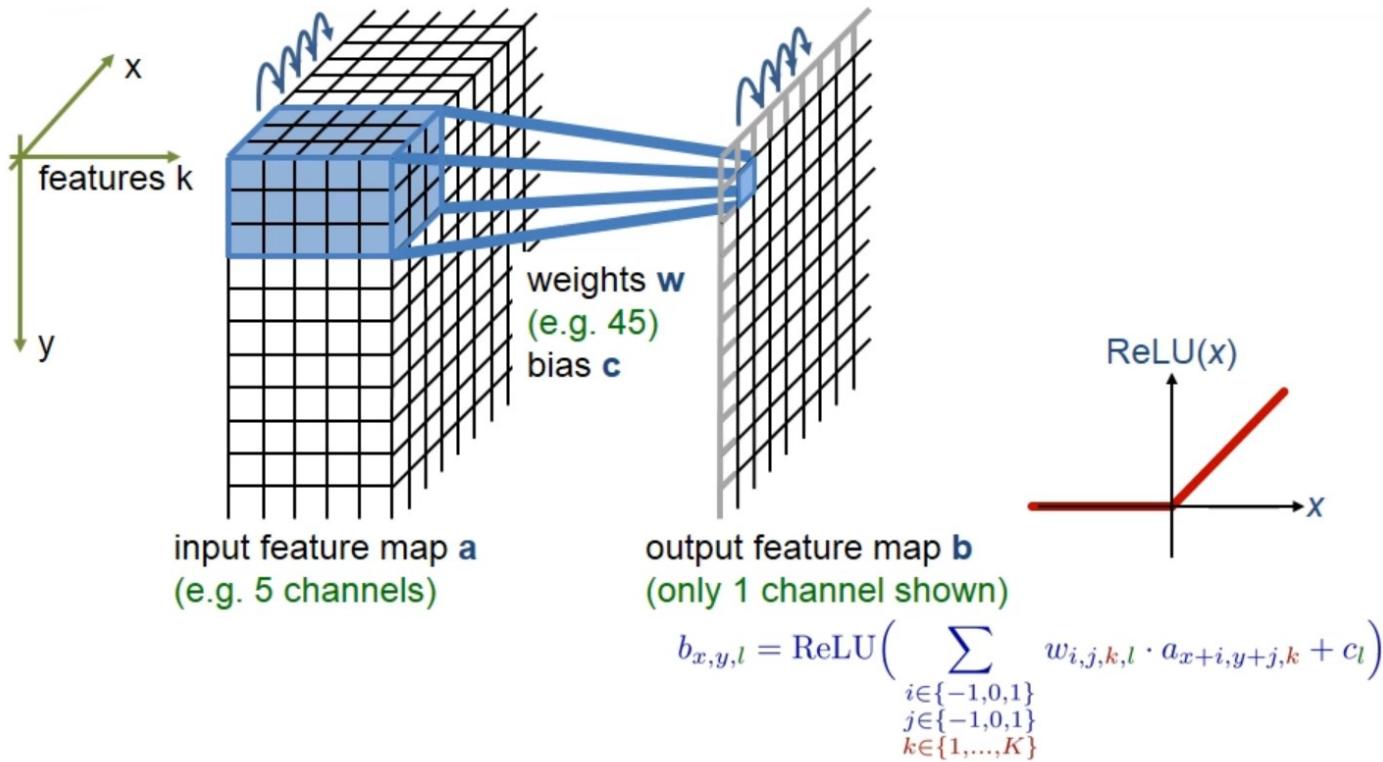
# U-Net Architecture



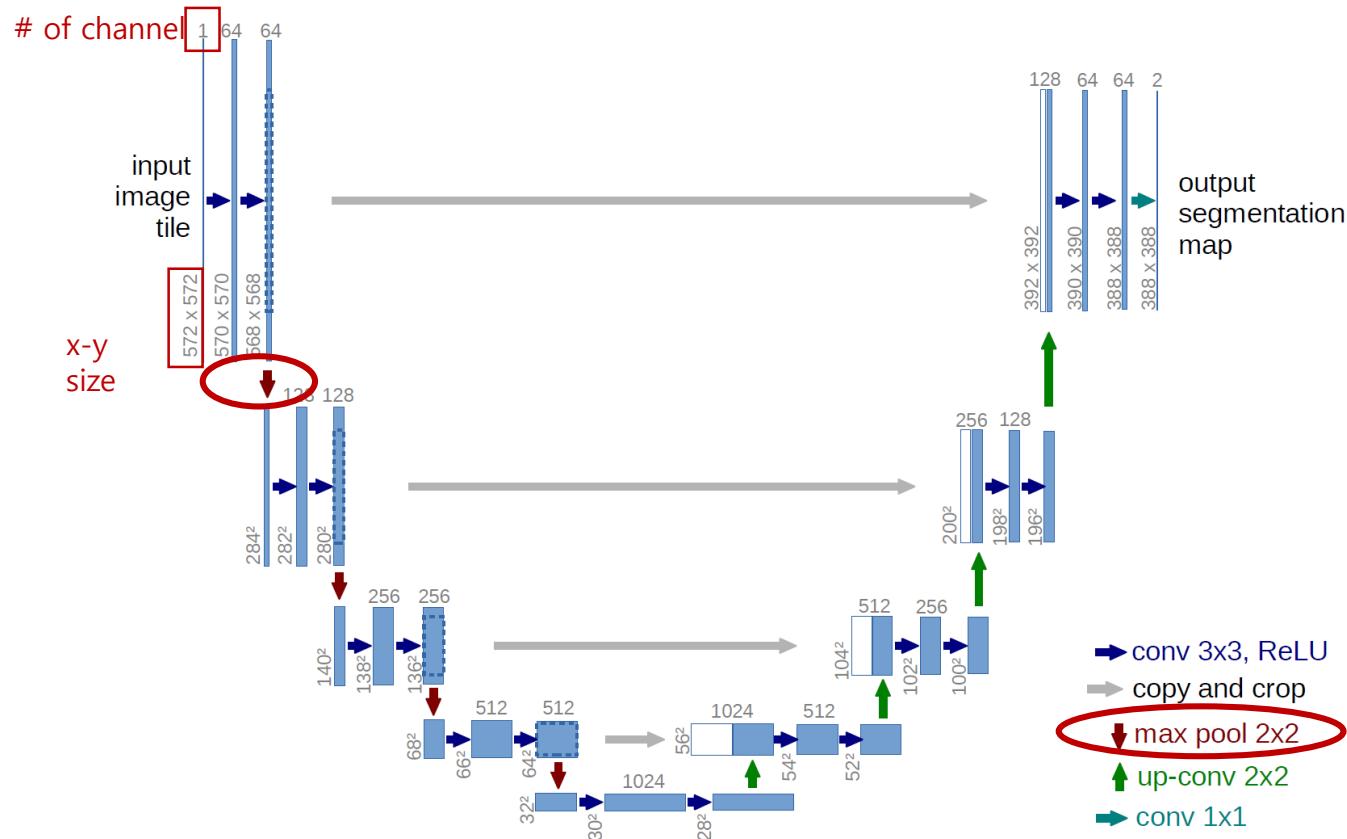
# U-Net Architecture



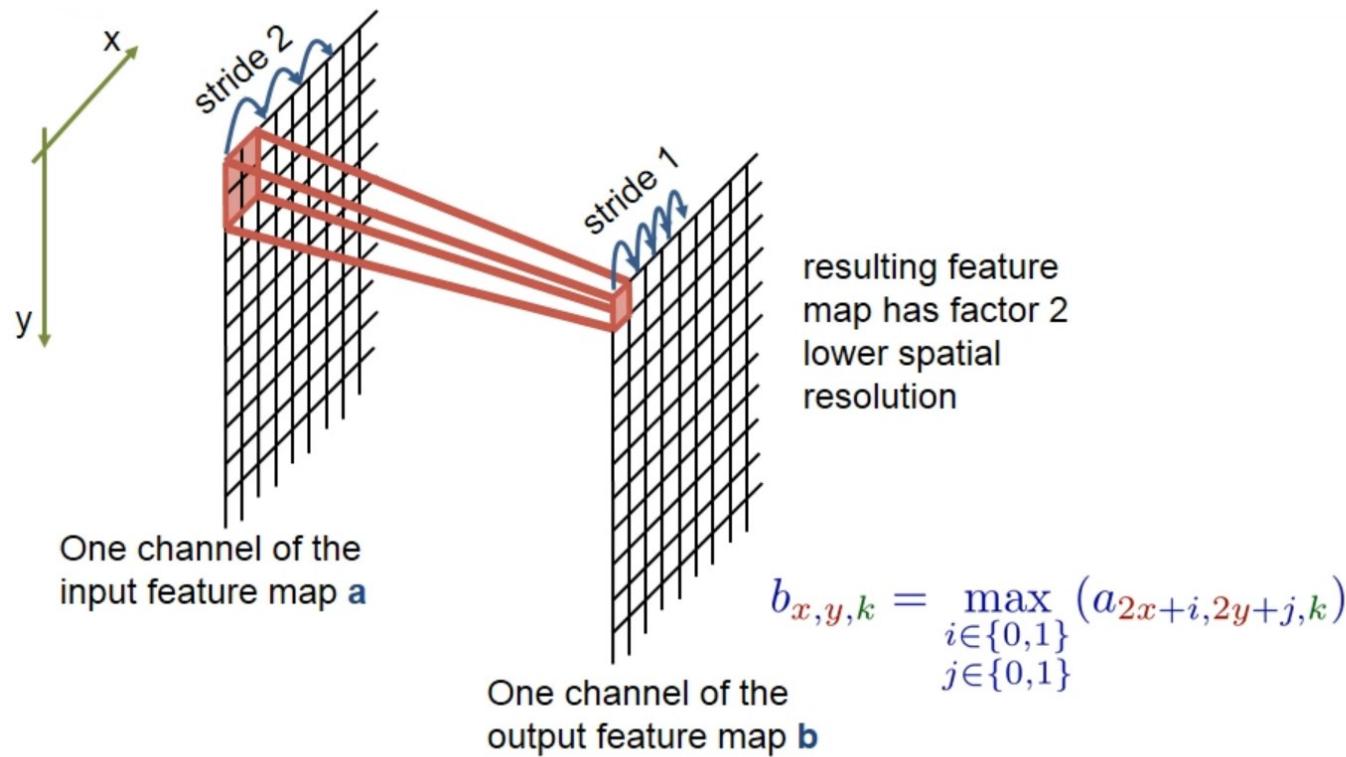
# 3x3 Convolution + ReLU



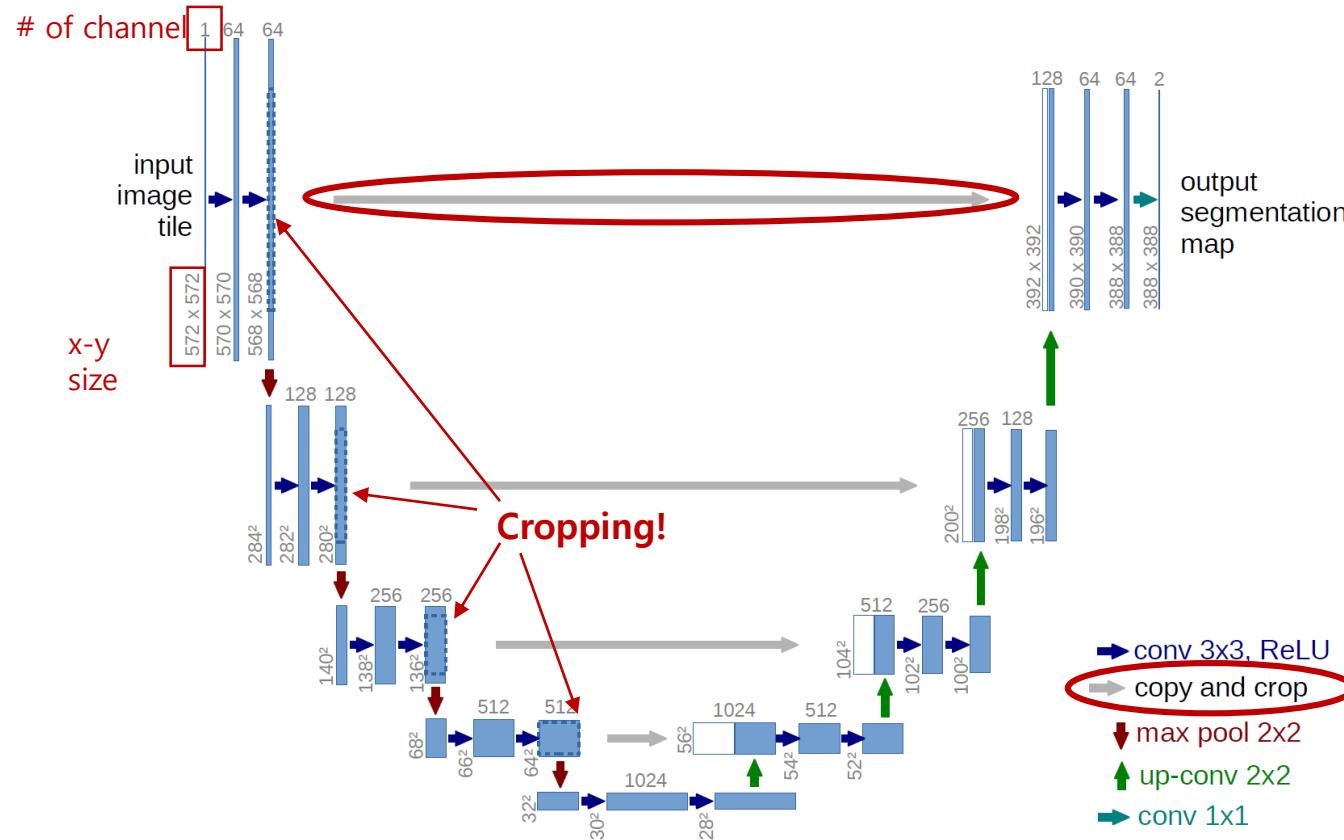
# U-Net Architecture



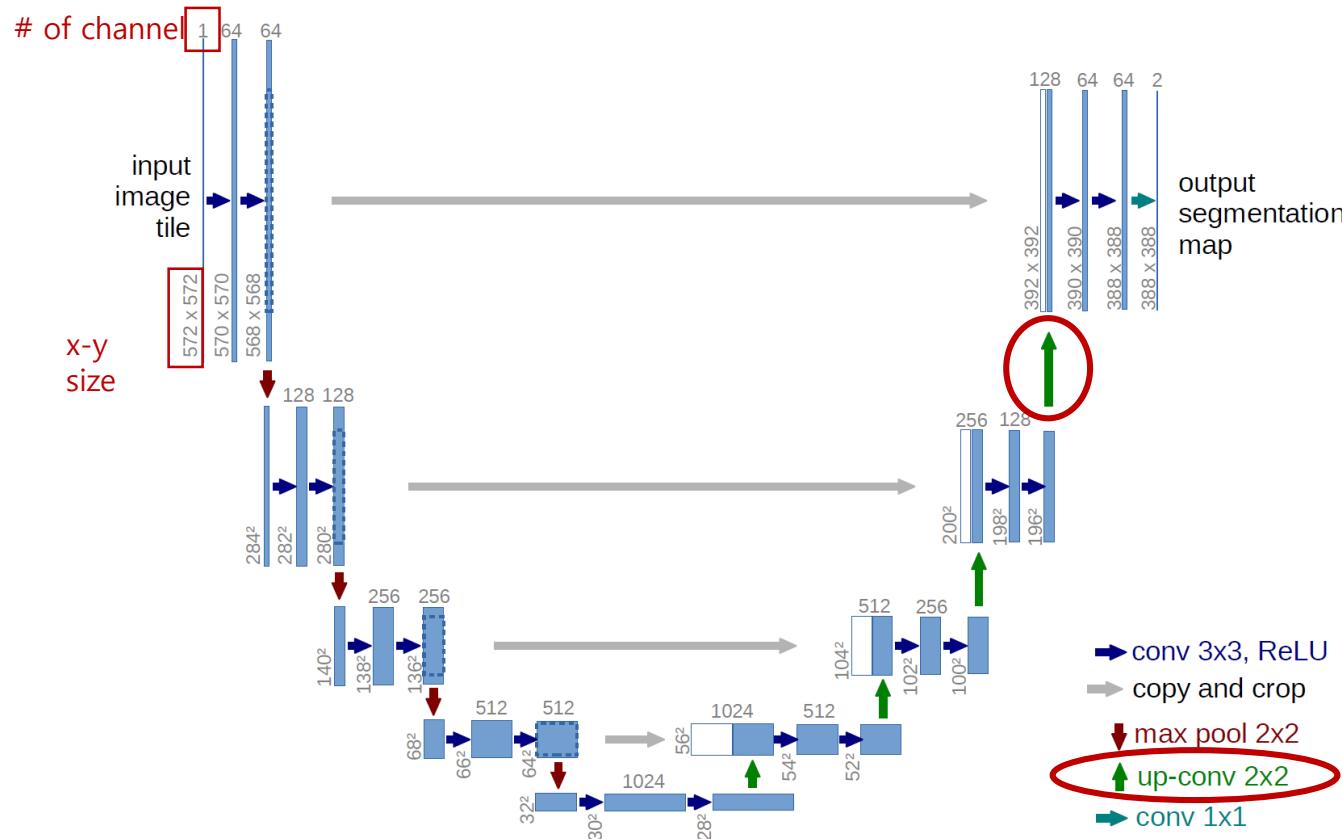
# 2x2 Max-pooling



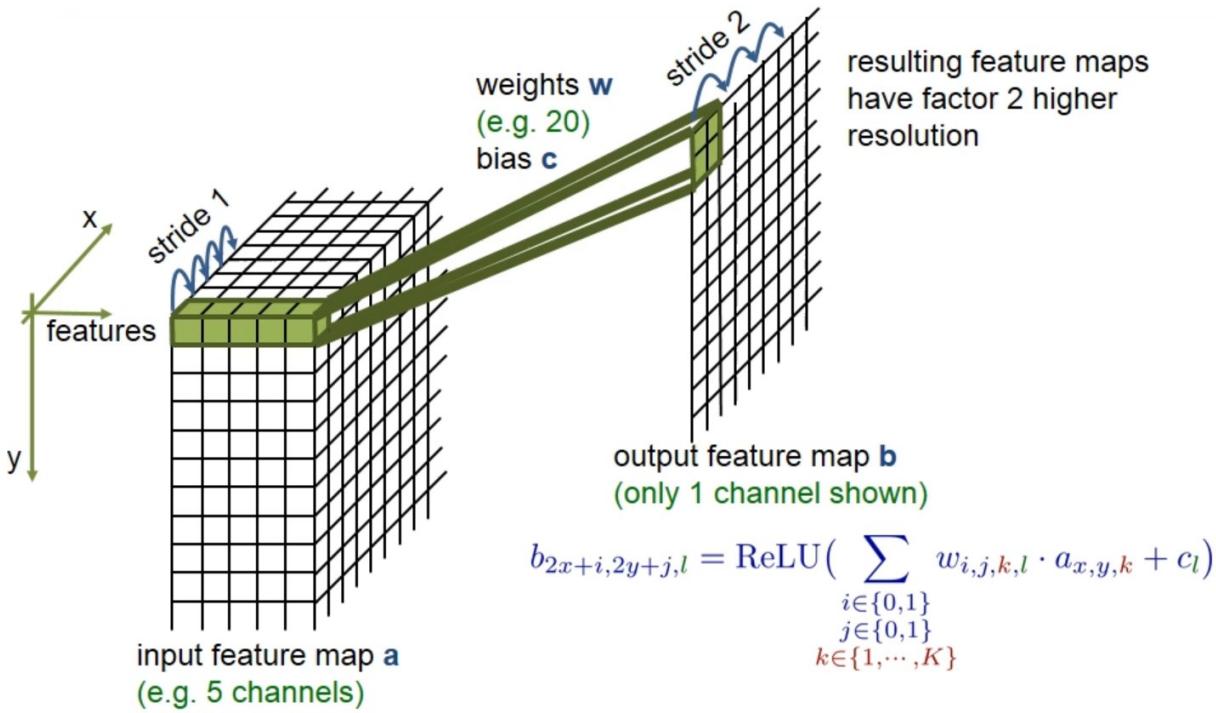
# U-Net Architecture



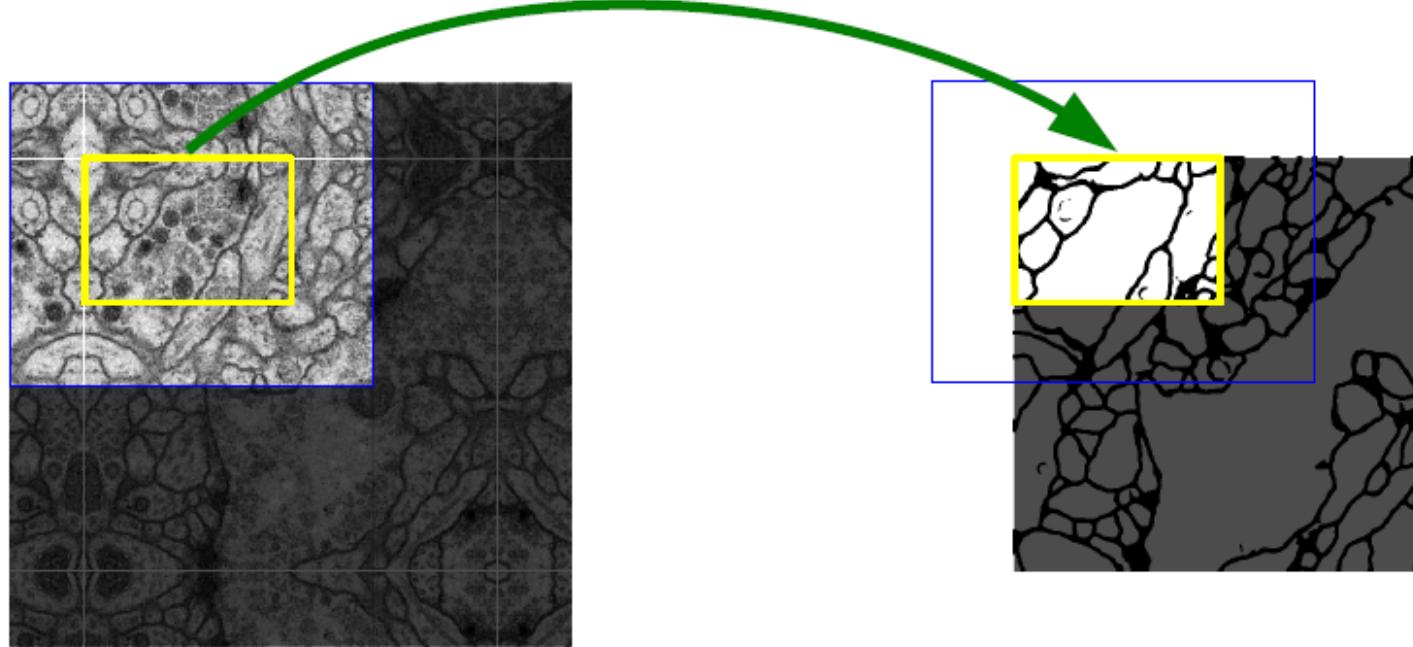
# U-Net Architecture



# 2x2 Up-convolution



# Overlap-tile strategy for arbitrary large images



- Segmentation of the yellow area uses input data of the blue area
- Raw data extrapolation by mirroring

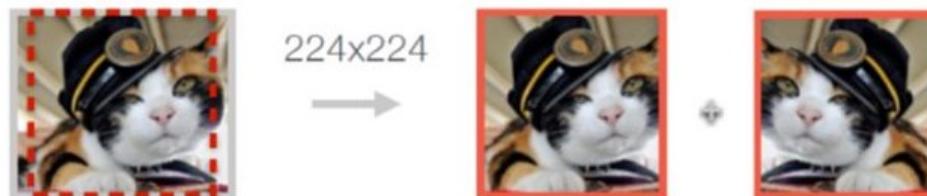
# Data Augmentation

- Data 수가 적을 때 일반적으로 사용
  - Random scale, random crop, changing aspect ratio, flip, rotation, adding noise or distortion 등이 많이 사용됨

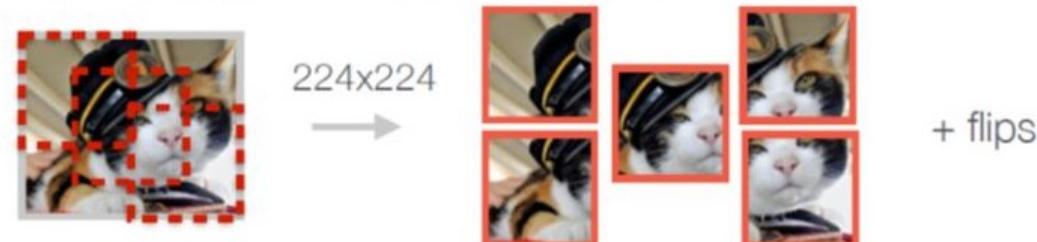
a. No augmentation (= 1 image)



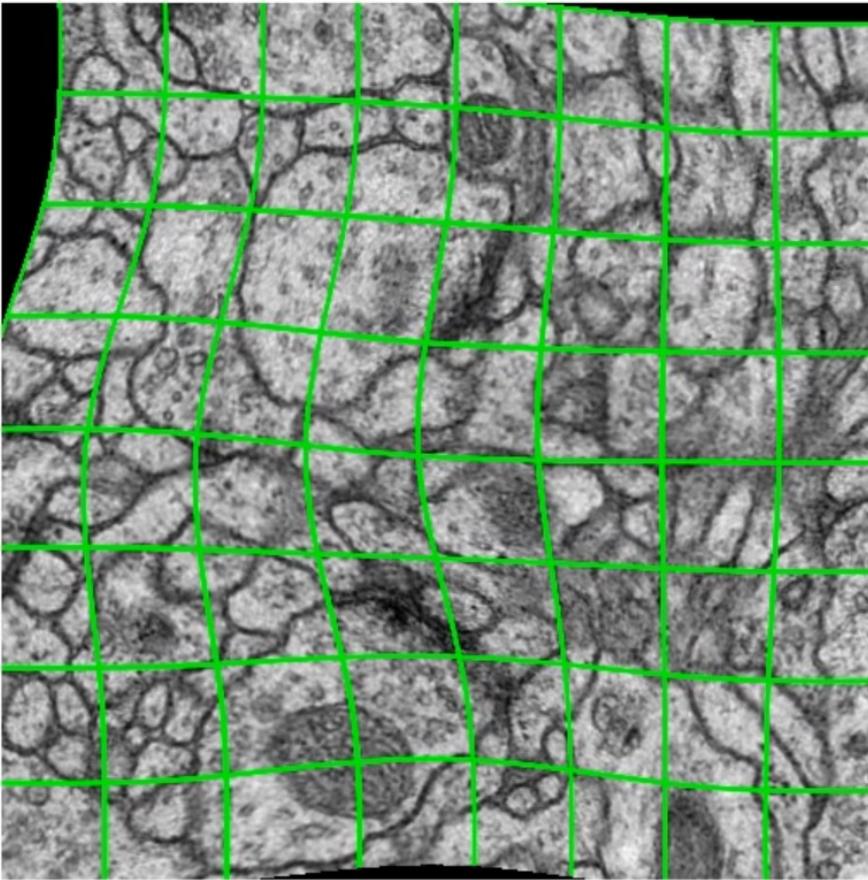
b. Flip augmentation (= 2 images)



c. Crop+Flip augmentation (= 10 images)

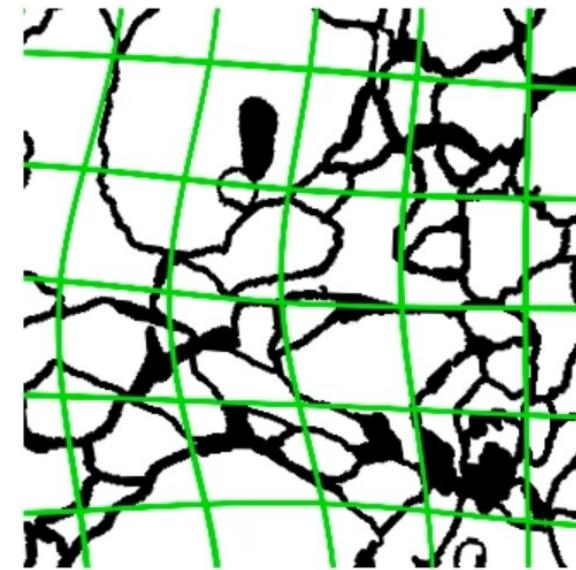


# Augment Training Data using Deformations



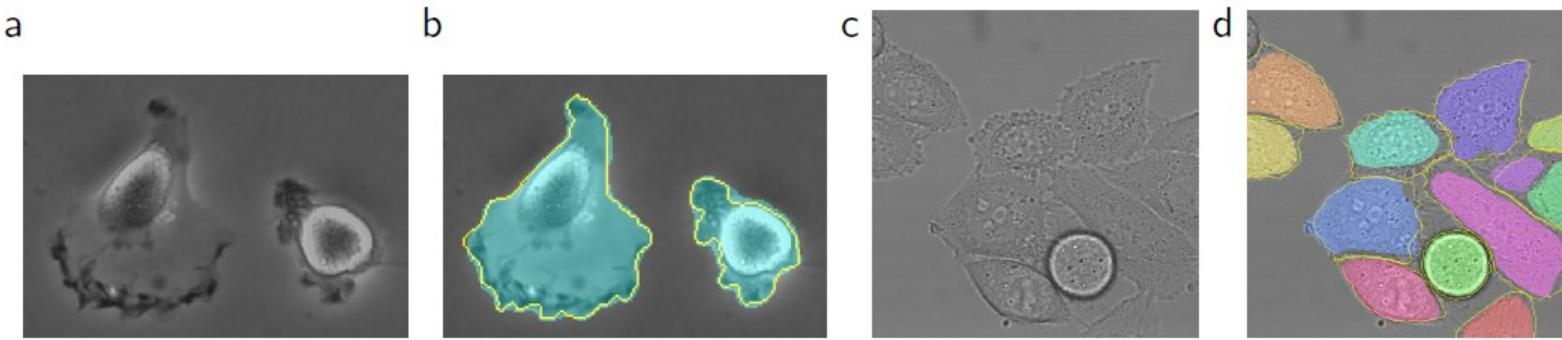
resulting deformed image

(for visualization: no rotation, no shift, no extrapolation)



correspondingly deformed  
manual labels

# ISBI 2015 Result



Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	<b>0.9203</b>	<b>0.7756</b>

# FusionNet: A deep fully residual convolutional neural network for image segmentation in connectomics

## FusionNet: A deep fully residual convolutional neural network for image segmentation in connectomics

Tran Minh Quan<sup>1</sup>, David G. C. Hildebrand<sup>2</sup> and Won-Ki Jeong<sup>1\*</sup>

<sup>1</sup>Ulsan National Institute of Science and Technology (UNIST),

<sup>2</sup>Harvard University

quantm@unist.ac.kr, david@hildebrand.name, wkjeong@unist.ac.kr

### Abstract

*Electron microscopic connectomics is an ambitious research direction with the goal of studying comprehensive brain connectivity maps by using high-throughput, nanoscale microscopy. One of the main challenges in connectomics research is developing scalable image analysis algorithms that require minimal user intervention. Recently, deep learning has drawn much attention in computer vision because of its exceptional performance in image classification tasks. For this reason, its application to connectomic analyses holds great promise, as well. In this paper, we introduce a novel deep neural network architecture, FusionNet, for the automatic segmentation of*

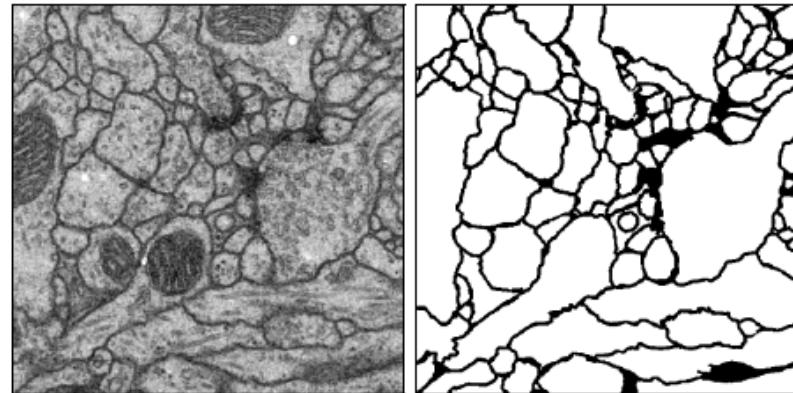
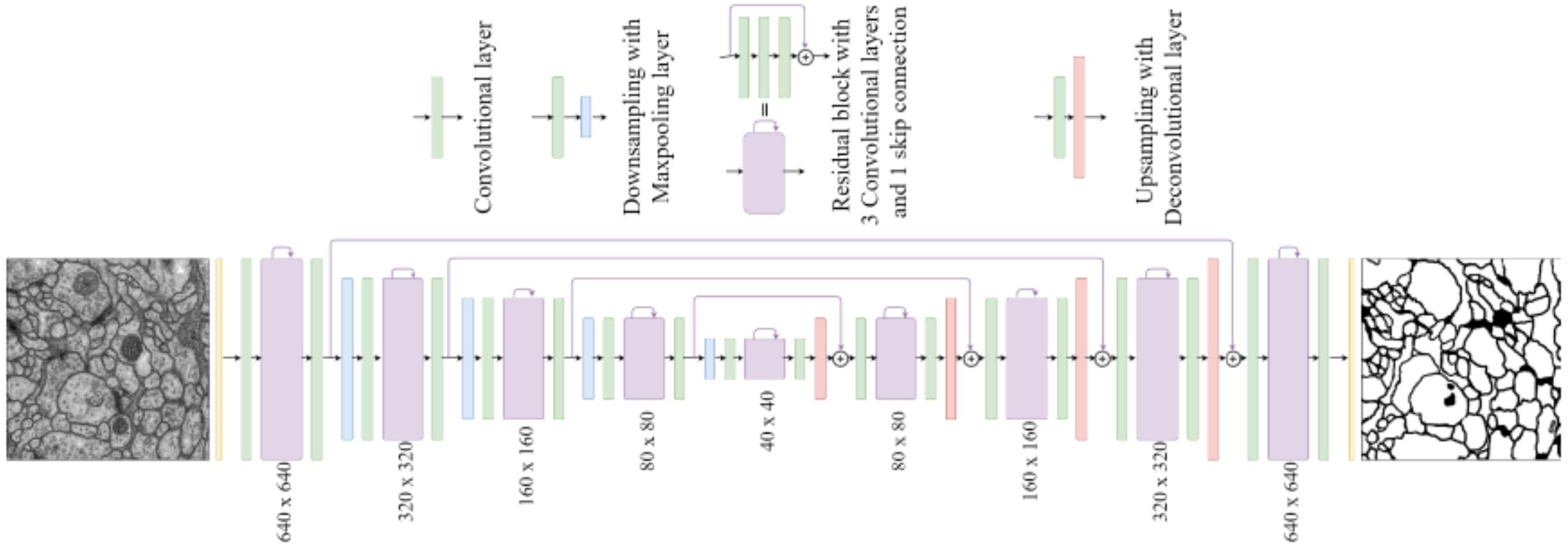


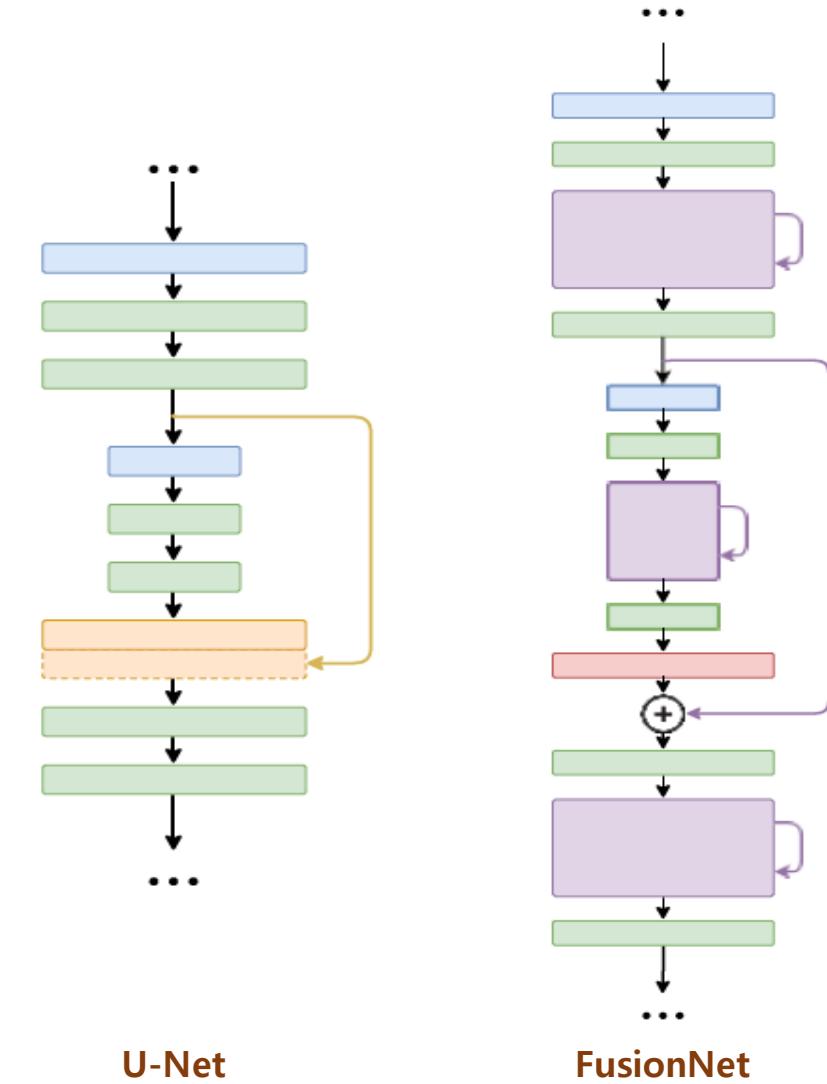
Figure 1. An example of EM image (left) and its cell membrane segmentation result (right).

# Architecture



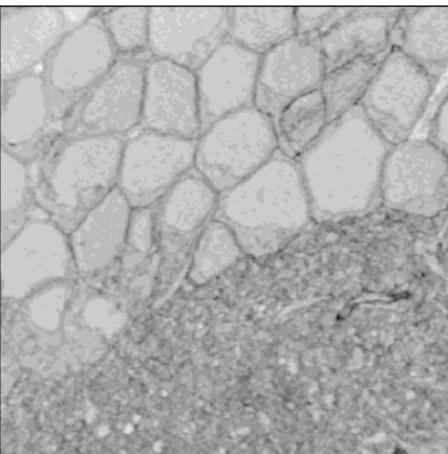
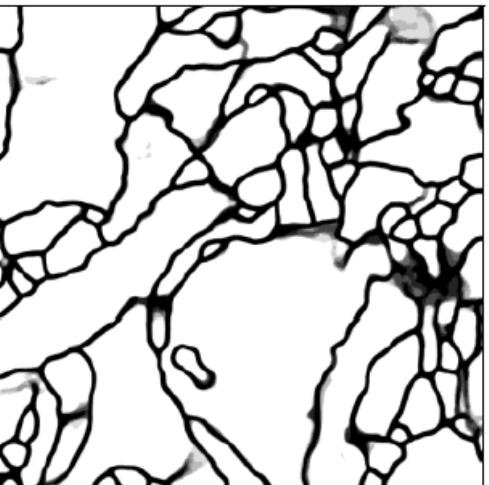
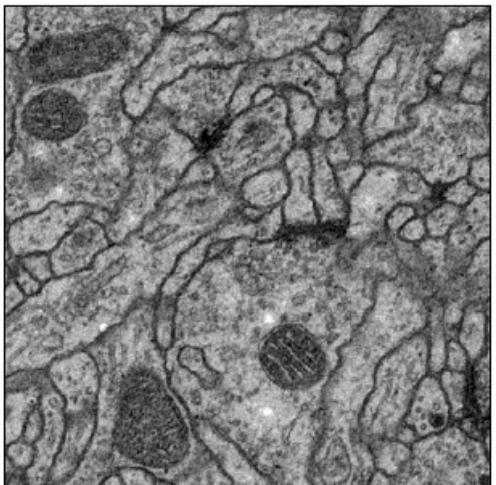
# Difference between U-Net and FusionNet

- FusionNet is much deeper compared to U-Net
- Fully residual network due to summation-based skip connections

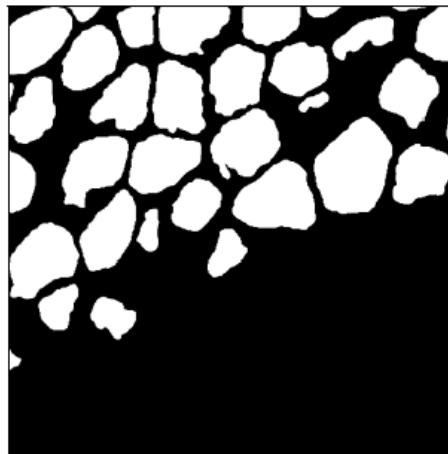


# Result

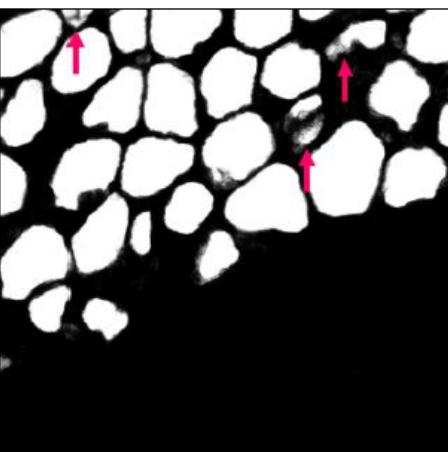
Methods	$V_{rand}$	$V_{info}$
Our approach (Nov 2016)	<b>0.978042575</b>	<b>0.989945379</b>
IAL IC [22]	0.977345721	0.989240736
Masters [34]	0.977141154	0.987534429
CUMedVision [8]	0.976824580	0.988645822
LSTM [30]	0.975366444	0.987425430
Our approach (May 2016)	0.972797555	0.987597732
U-net [25]	0.972760748	0.986616590



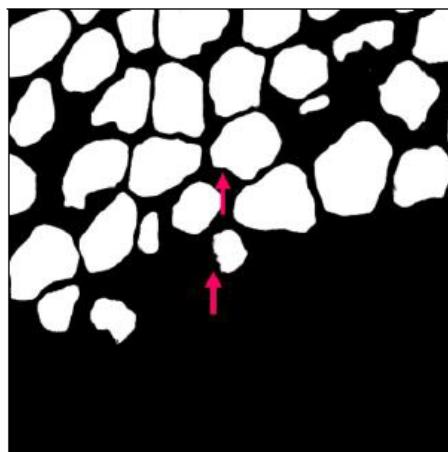
(a)



(b)



(c)



(d)

Figure 7. Top row: Raw image (a) of Slice 195/512 on the test set and its manual segmentation (b); Bottom row: U-net [25] result (c) and our result (d). Note that our result shows less false-positive and more accurate segmentation (see red arrows).

# Pyramid Scene Parsing Network(PSPNet)

## Pyramid Scene Parsing Network

Hengshuang Zhao<sup>1</sup> Jianping Shi<sup>2</sup> Xiaojuan Qi<sup>1</sup> Xiaogang Wang<sup>1</sup> Jiaya Jia<sup>1</sup>

<sup>1</sup>The Chinese University of Hong Kong <sup>2</sup>SenseTime Group Limited

{hszhao, xjqi, leojia}@cse.cuhk.edu.hk, xgwang@ee.cuhk.edu.hk, shijianping@sensetime.com

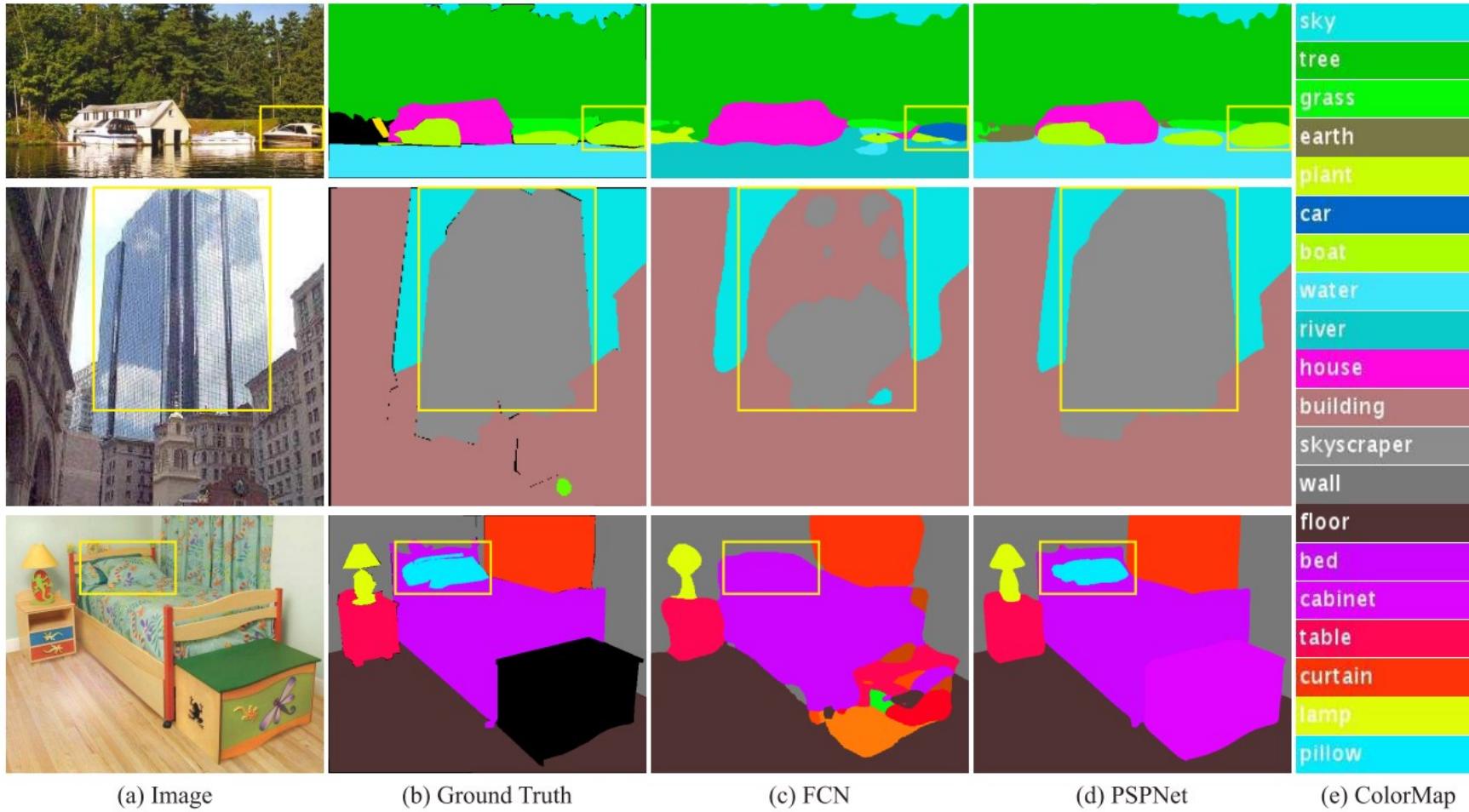
### Abstract

*Scene parsing is challenging for unrestricted open vocabulary and diverse scenes. In this paper, we exploit the capability of global context information by different-region-based context aggregation through our pyramid pooling module together with the proposed pyramid scene parsing network (PSPNet). Our global prior representation is effective to produce good quality results on the scene parsing task, while PSPNet provides a superior framework for pixel-level prediction. The proposed approach achieves state-of-the-art performance on various datasets. It came first in ImageNet scene parsing challenge 2016, PASCAL VOC 2012 benchmark and Cityscapes benchmark. A single PSPNet yields the new record of mIoU accuracy 85.4% on PASCAL VOC 2012, and 88.2% on Cityscapes.*



# Pyramid Scene Parsing Network(PSPNet)

- Deep Network with a suitable global-scene-level prior can much improve the performance of scene parsing



# PSPNet Architecture

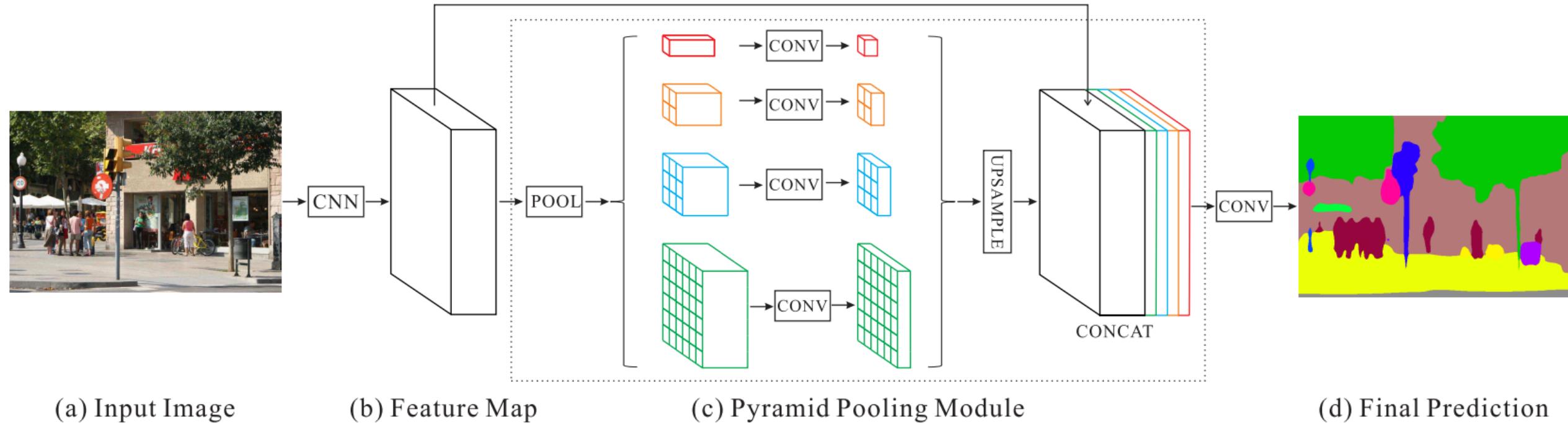
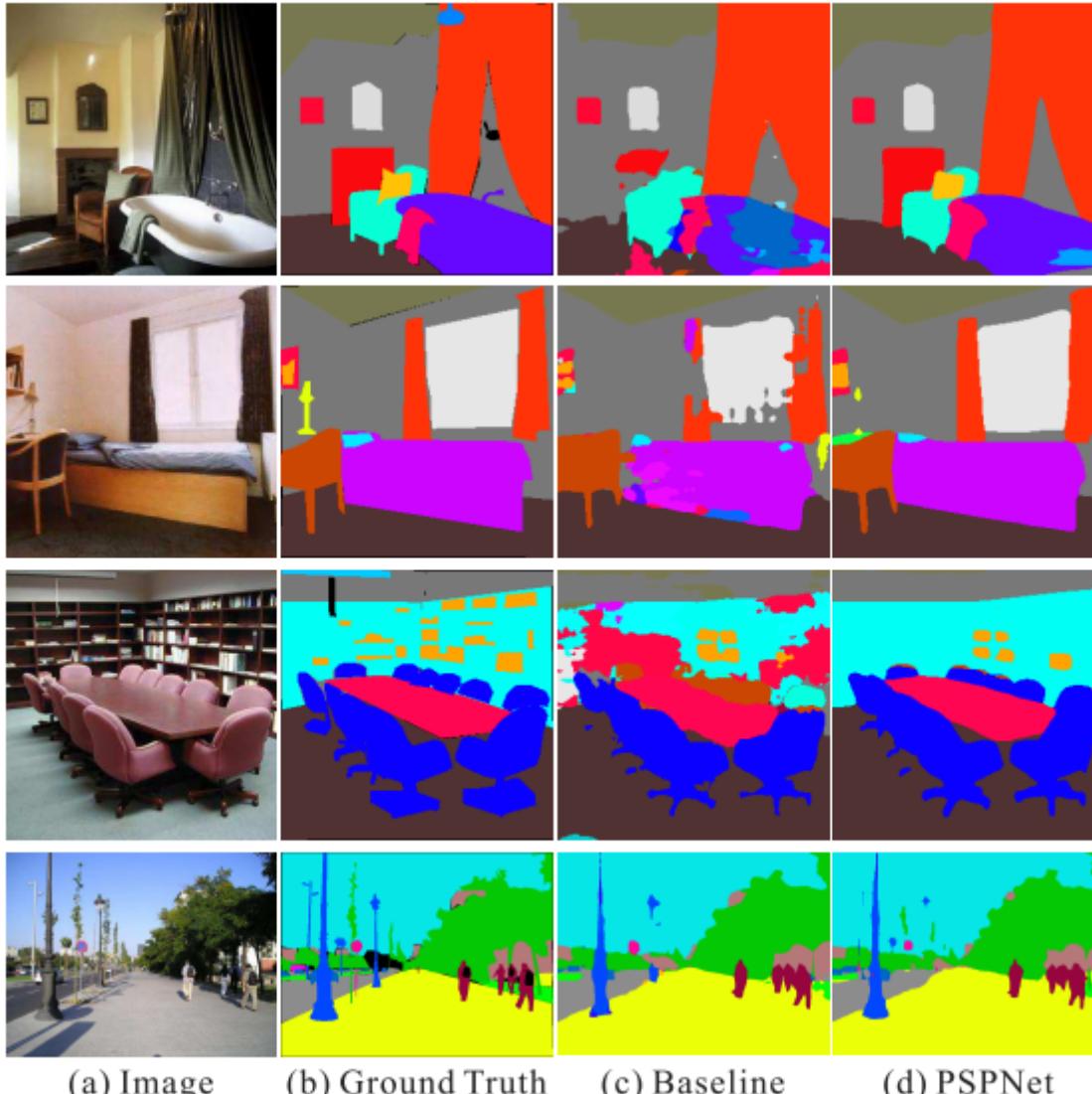
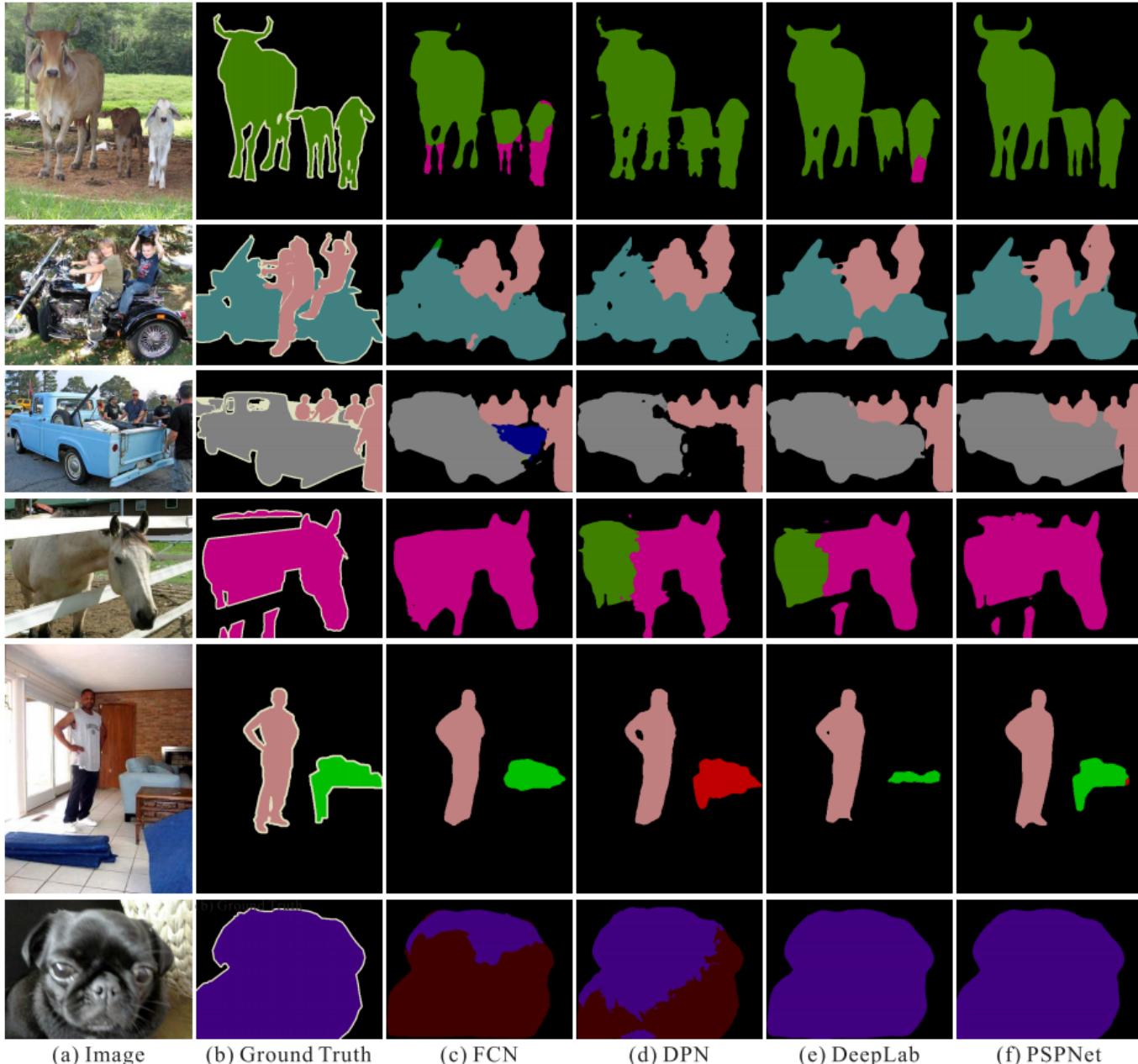
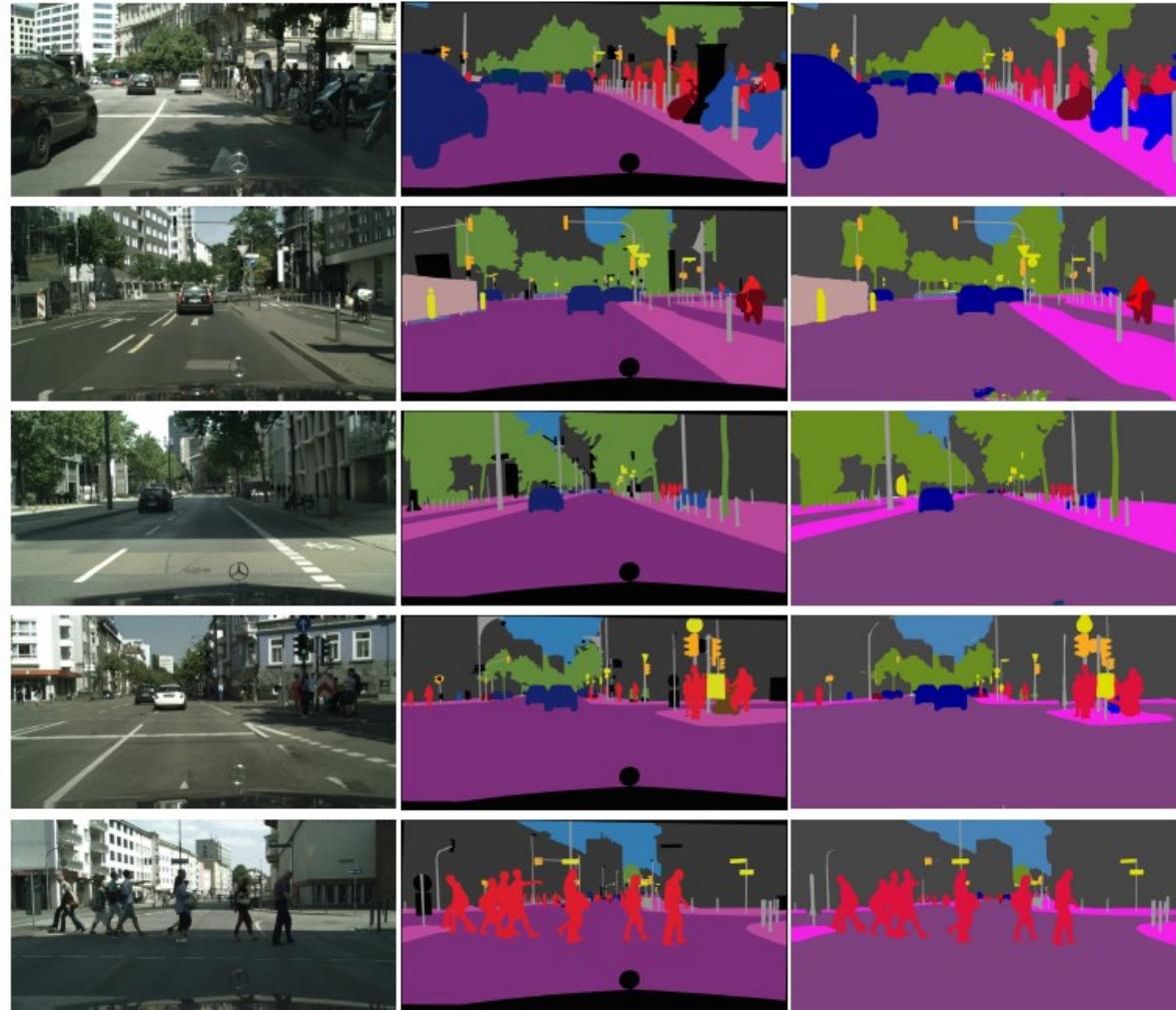


Figure 3. Overview of our proposed PSPNet. Given an input image (a), we first use CNN to get the feature map of the last convolutional layer (b), then a pyramid parsing module is applied to harvest different sub-region representations, followed by upsampling and concatenation layers to form the final 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).

# Results



# Results



# Mask R-CNN

## Mask R-CNN

Kaiming He

Georgia Gkioxari

Piotr Dollár

Ross Girshick

Facebook AI Research (FAIR)

### Abstract

We present a conceptually simple, flexible, and general framework for object instance segmentation. Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework. We show top results in all three tracks of the COCO suite of challenges, including instance segmentation, bounding

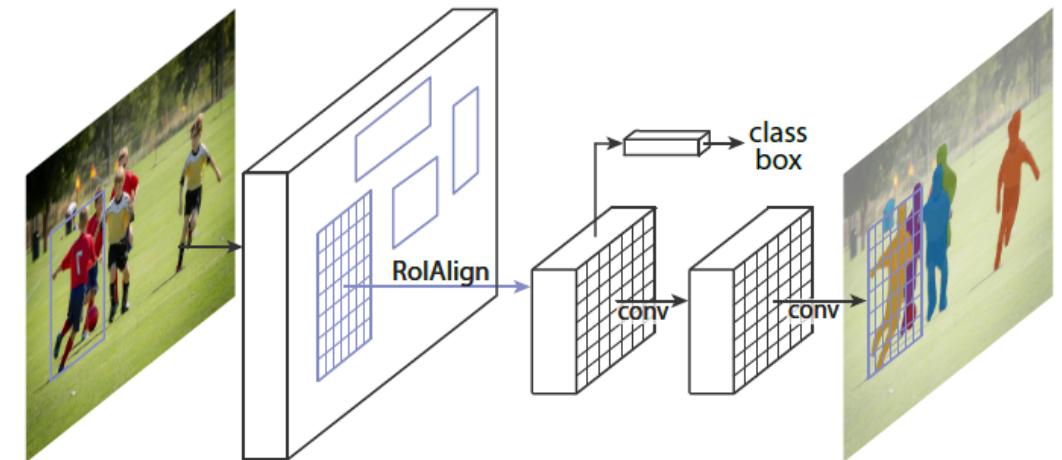


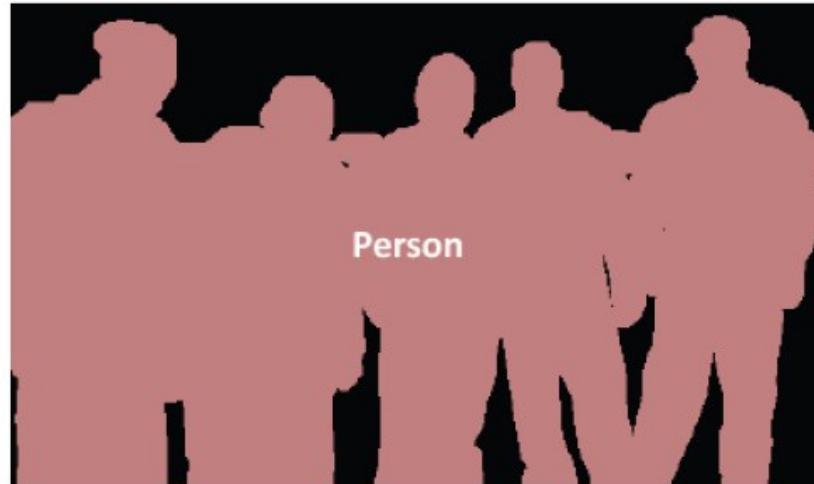
Figure 1. The **Mask R-CNN** framework for instance segmentation.

segmentation, where the goal is to classify each pixel into a fixed set of categories without differentiating object instances.<sup>1</sup> Given this, one might expect a complex method

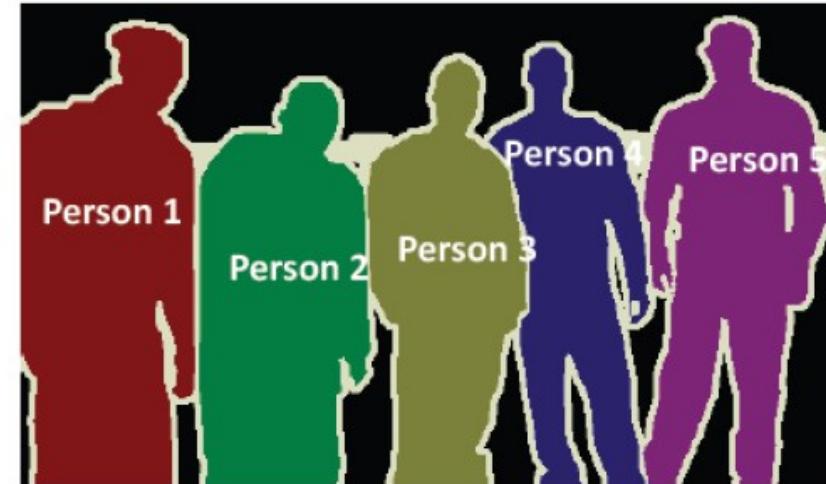
# Instance Segmentation



Object Detection



Semantic Segmentation



Instance Segmentation



# Instance Segmentation

# entries on COCO  
leaderboard

31



Object Det.

Instance Seg.

# entries on Cityscapes  
leaderboard

58



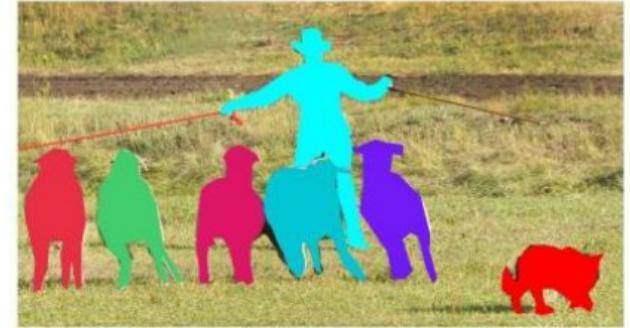
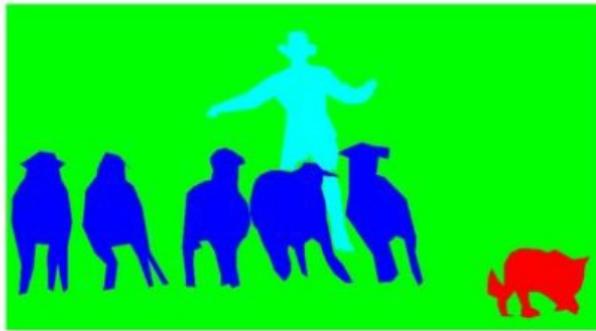
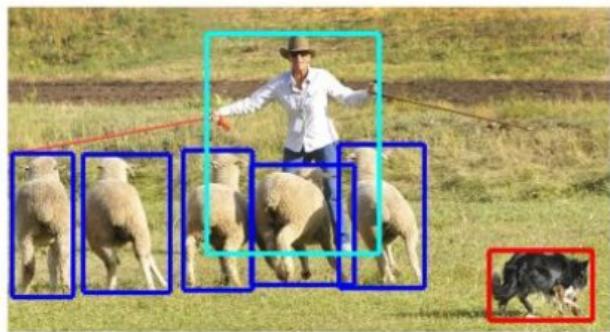
11



Semantic Seg.

Instance Seg.

# Instance Segmentation



BBox  
Classification

+

Segmentation  
Classification

=

Segmentation  
in BBox  
Classification

Can Separate  
Cannot Segment

Cannot Separate  
Can Segment

Faster R-CNN

+

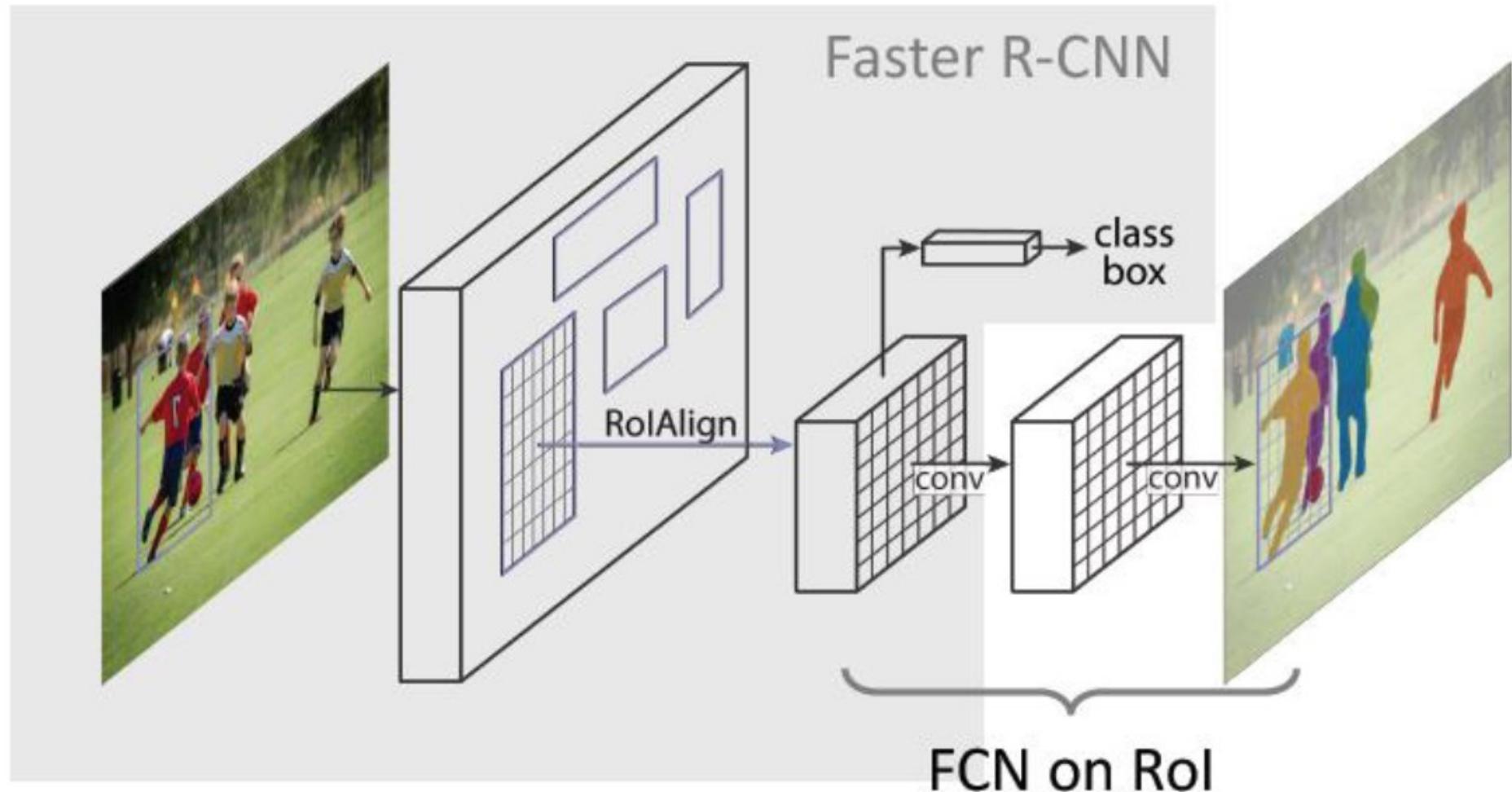
FCN

=

FCN  
on BBOX !

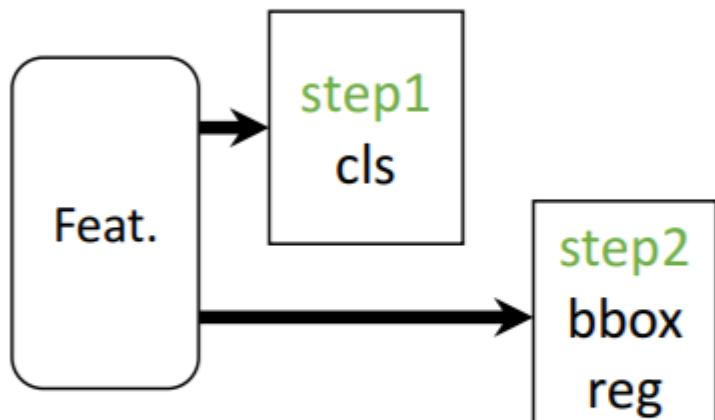
# Mask R-CNN

- Mask R-CNN = **Faster R-CNN** with **FCN** on Rols

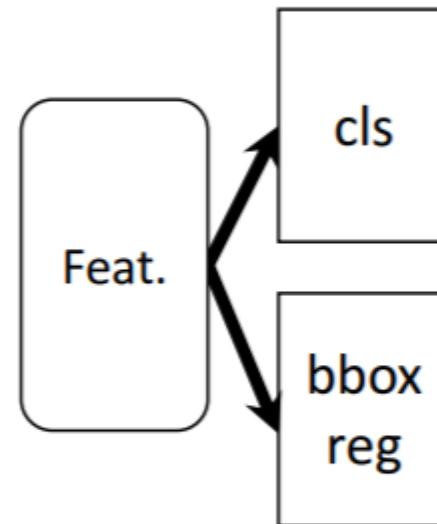


# Parallel Heads

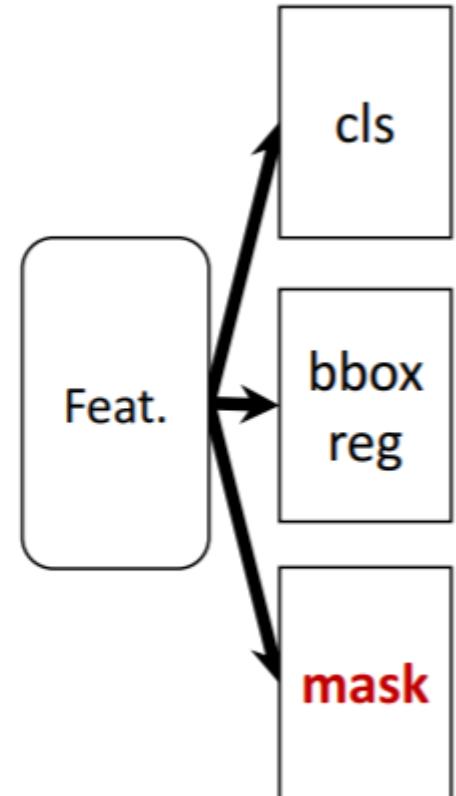
- Easy, fast to implement and train



(slow) R-CNN

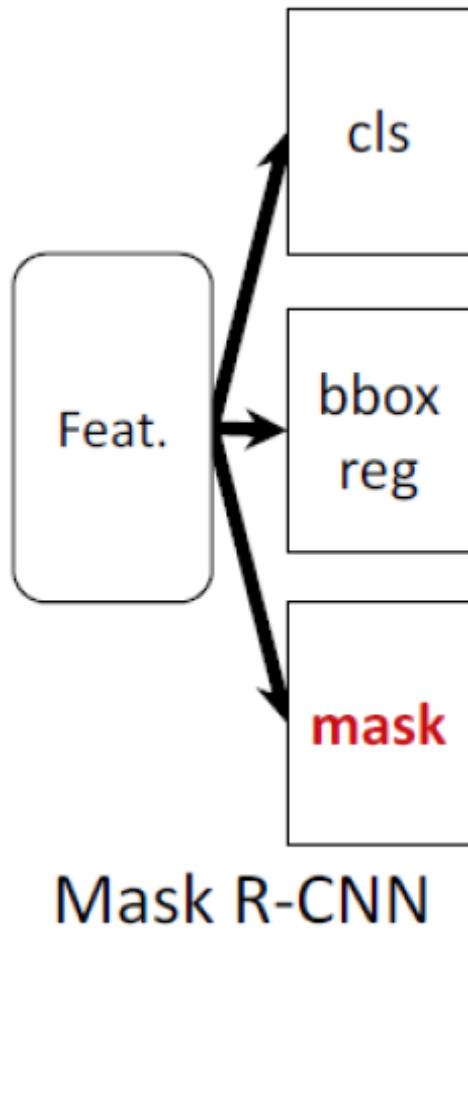


Fast/er R-CNN



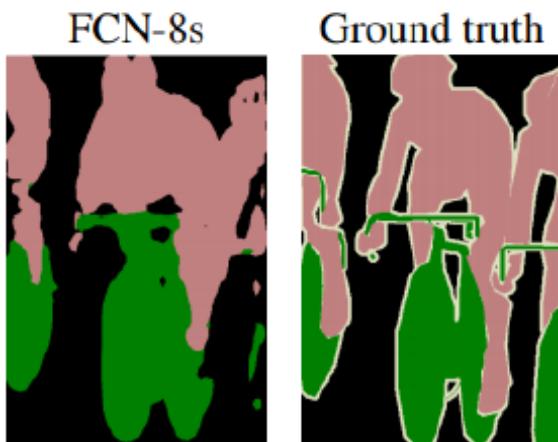
Mask R-CNN

# Parallel Heads



## Faster R-CNN

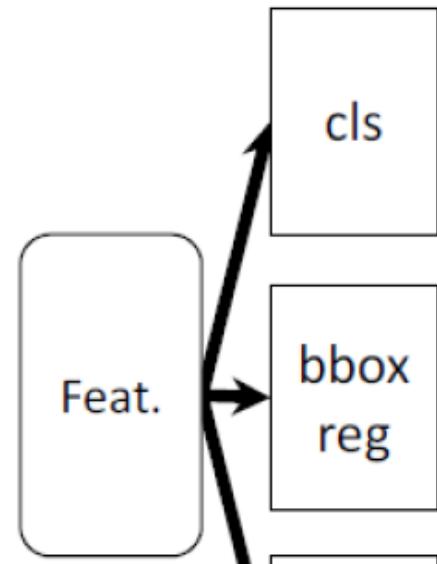
- Classification
- Instance Level RoI



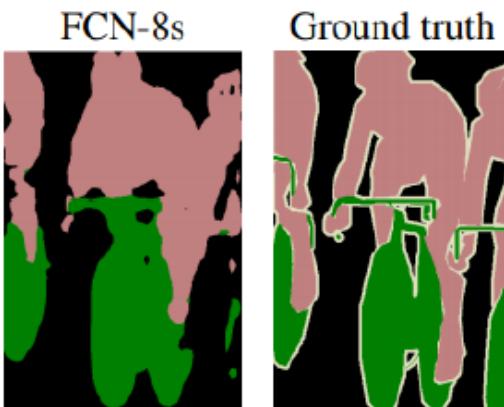
## FCN

- Pixel-level Classification
- Per Pixel Softmax (Multinomial)
- Multi Instance

# Parallel Heads



Mask R-CNN



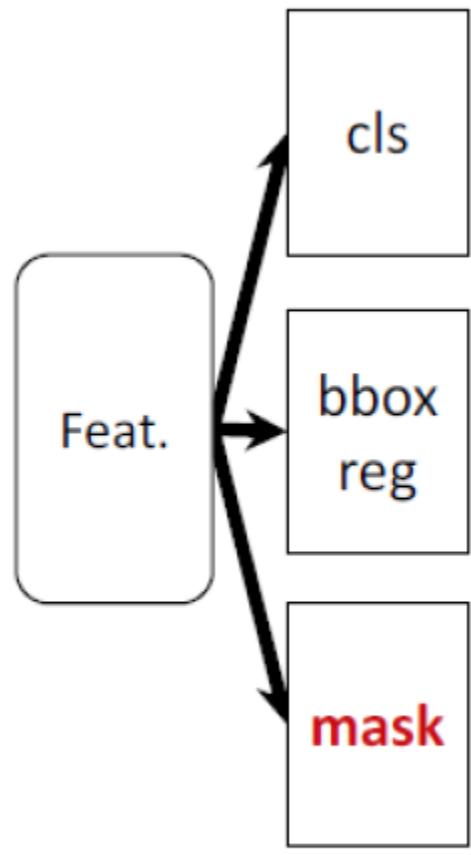
## Faster R-CNN

- Classification
- Instance Level ROI

FCN

- ~~Pixel level Classification~~
- Per Pixel ~~Softmax~~ → Sigmoid (Binary)
- ~~Multi Instance~~

# Parallel Heads



DB  
BBox + Class + Mask

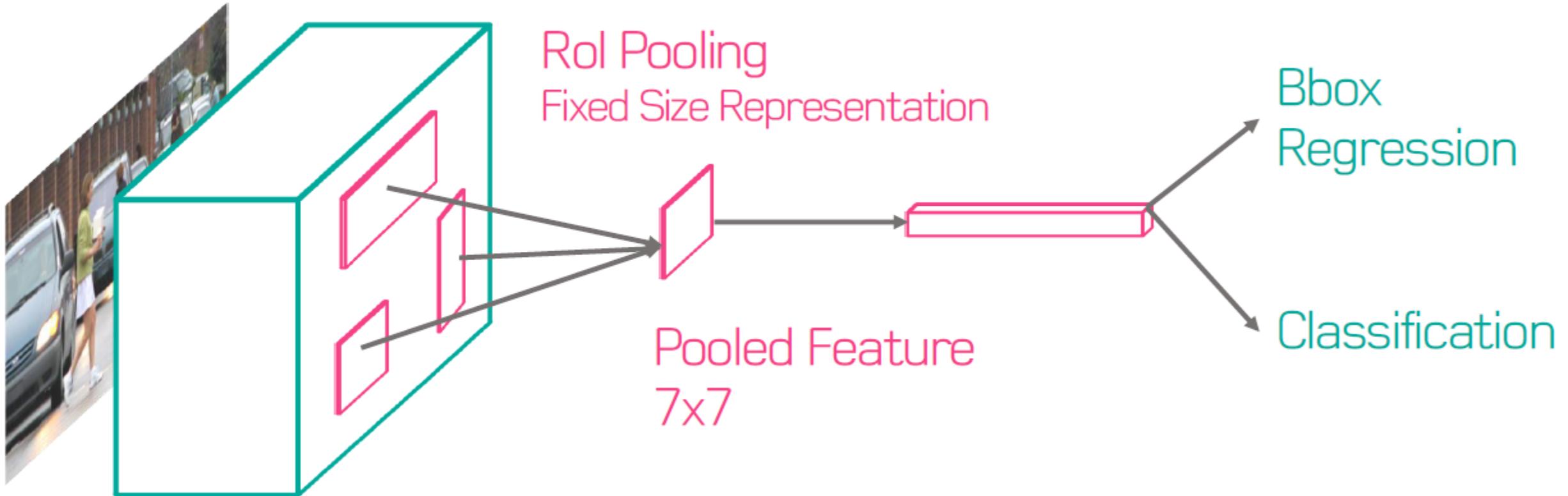
$$L = L_{cls} + L_{box} + L_{mask}$$

$L_{cls}$ : Softmax Cross Entropy

$L_{box}$ : Regression

$L_{mask}$ : Binary Cross Entropy

# Recap : Faster R-CNN

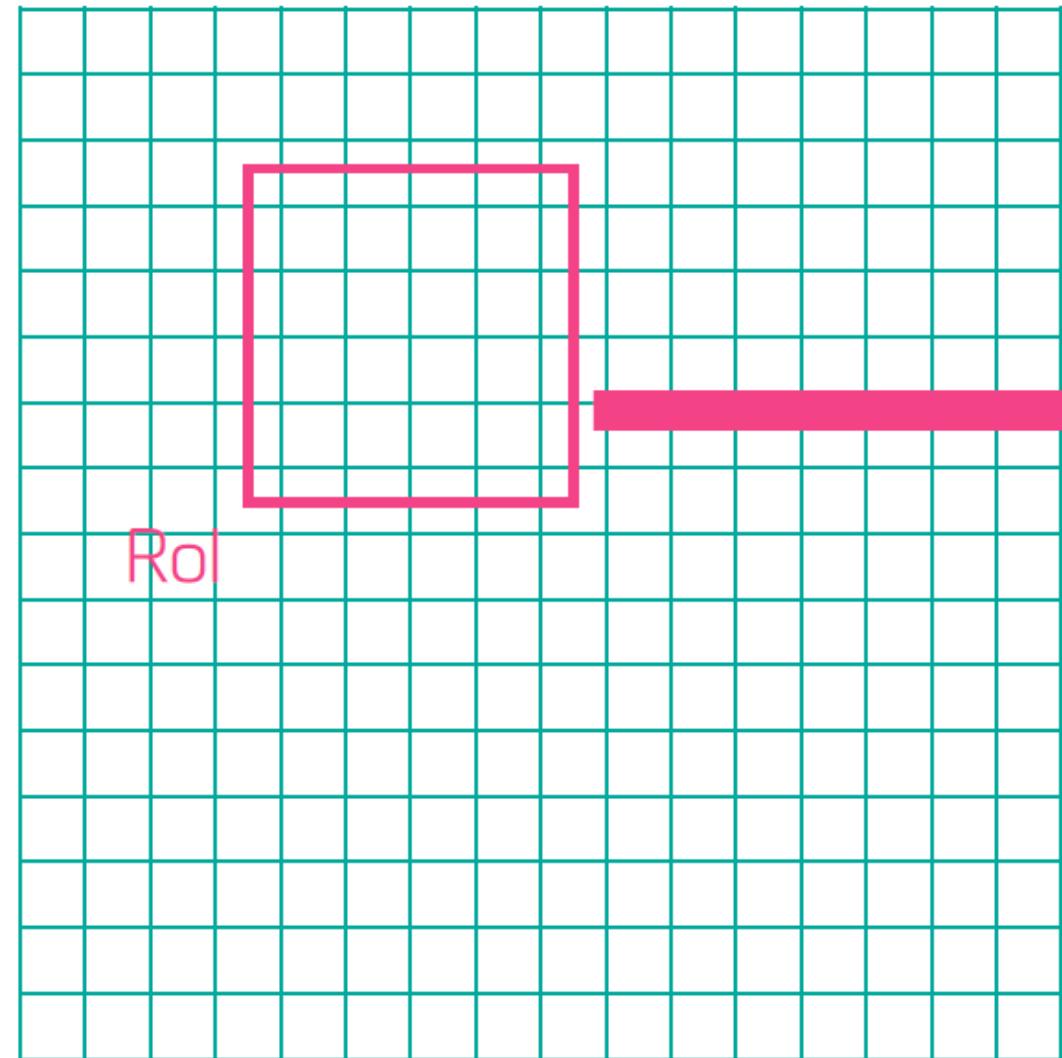


# Problems of Faster R-CNN

- ROI Projection and Pooling has some quantization operations
- These quantizations introduce misalignments between the ROI and the extracted features
- While this may not impact classification, it can make a negative effect on predicting bbox

# Roi Pooling

Feature Map

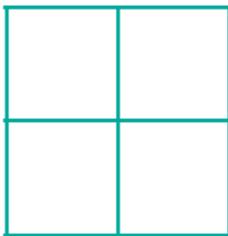
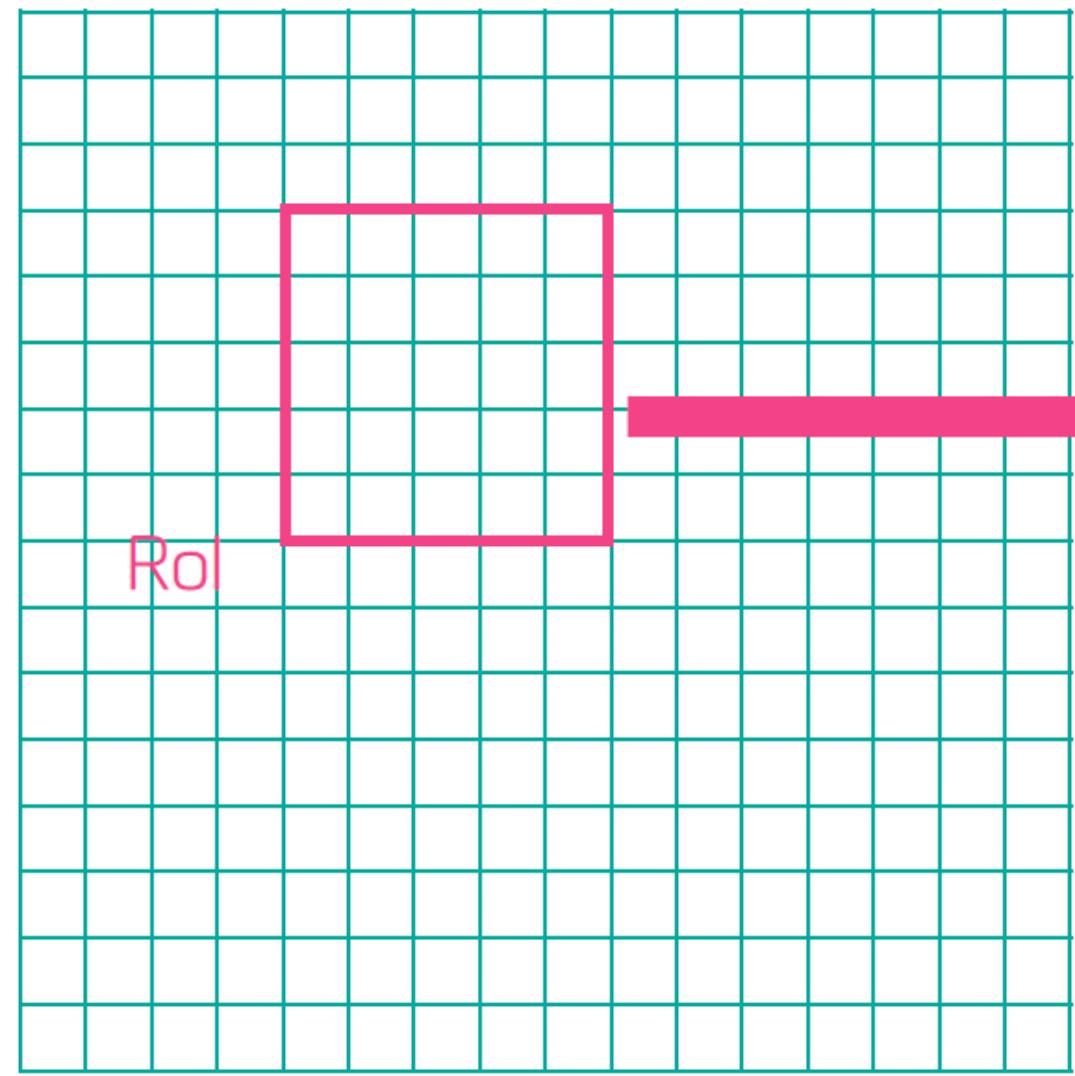


Note:

Region Proposal Network Roi Prediction  
= Floating Point Representation

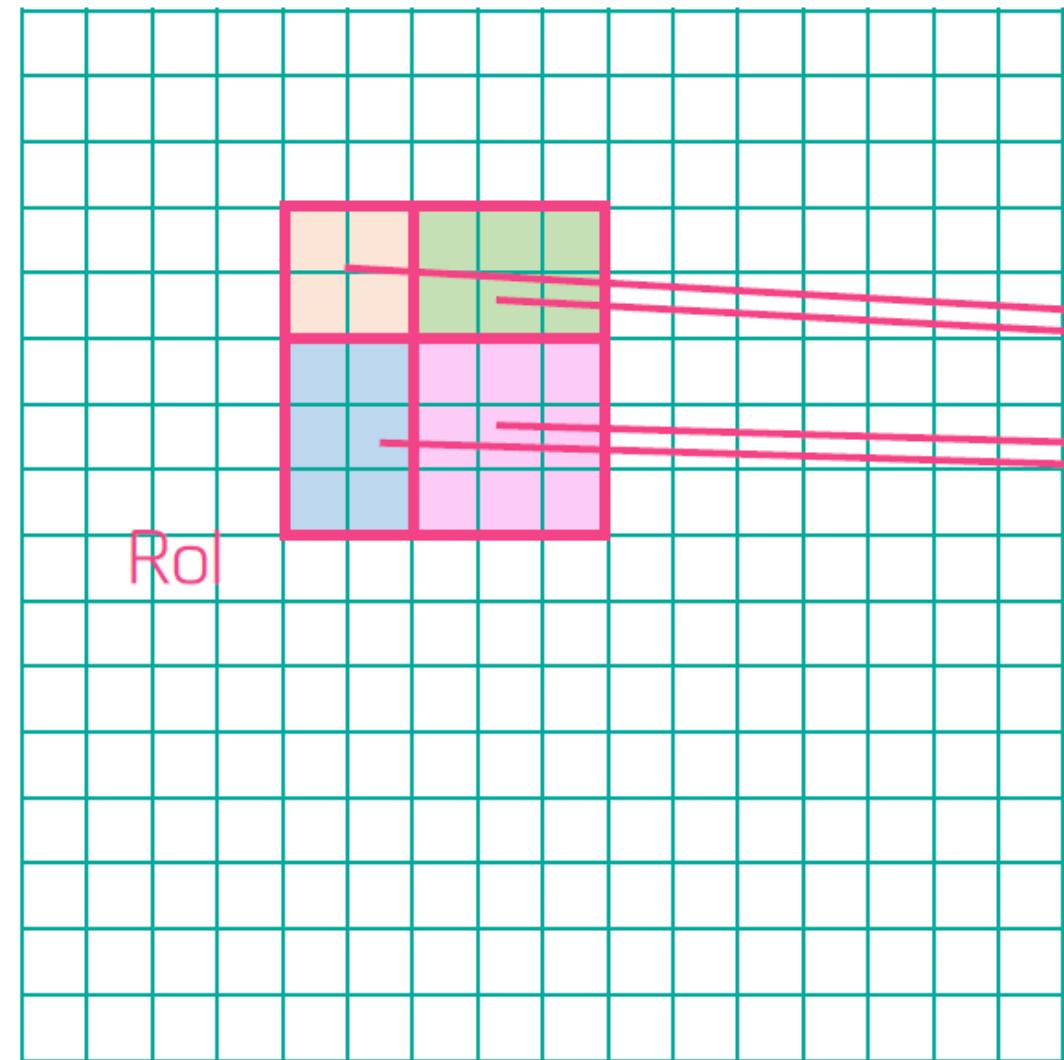
# Roi Pooling

Feature Map

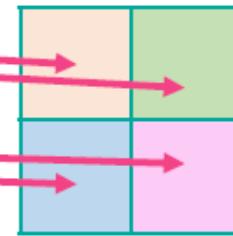


# Roi Pooling

Feature Map

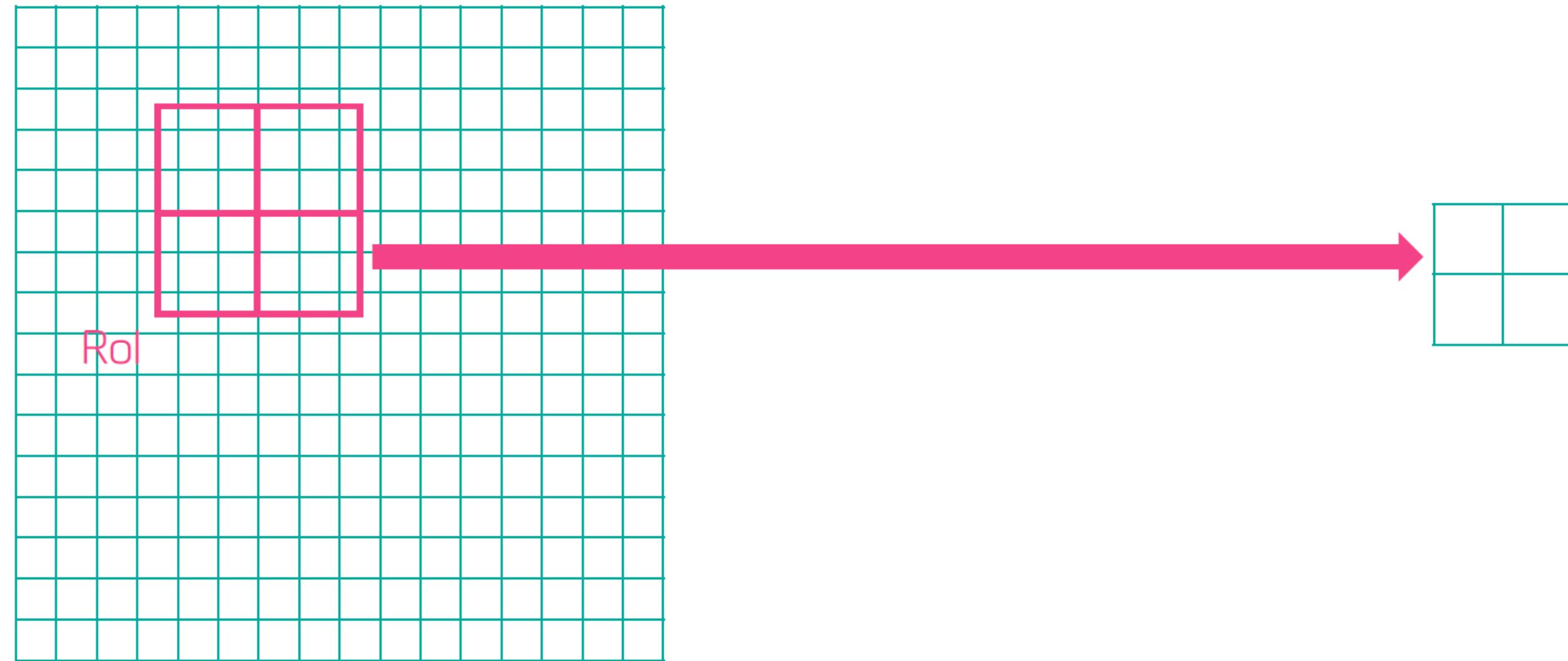


Max Pooling



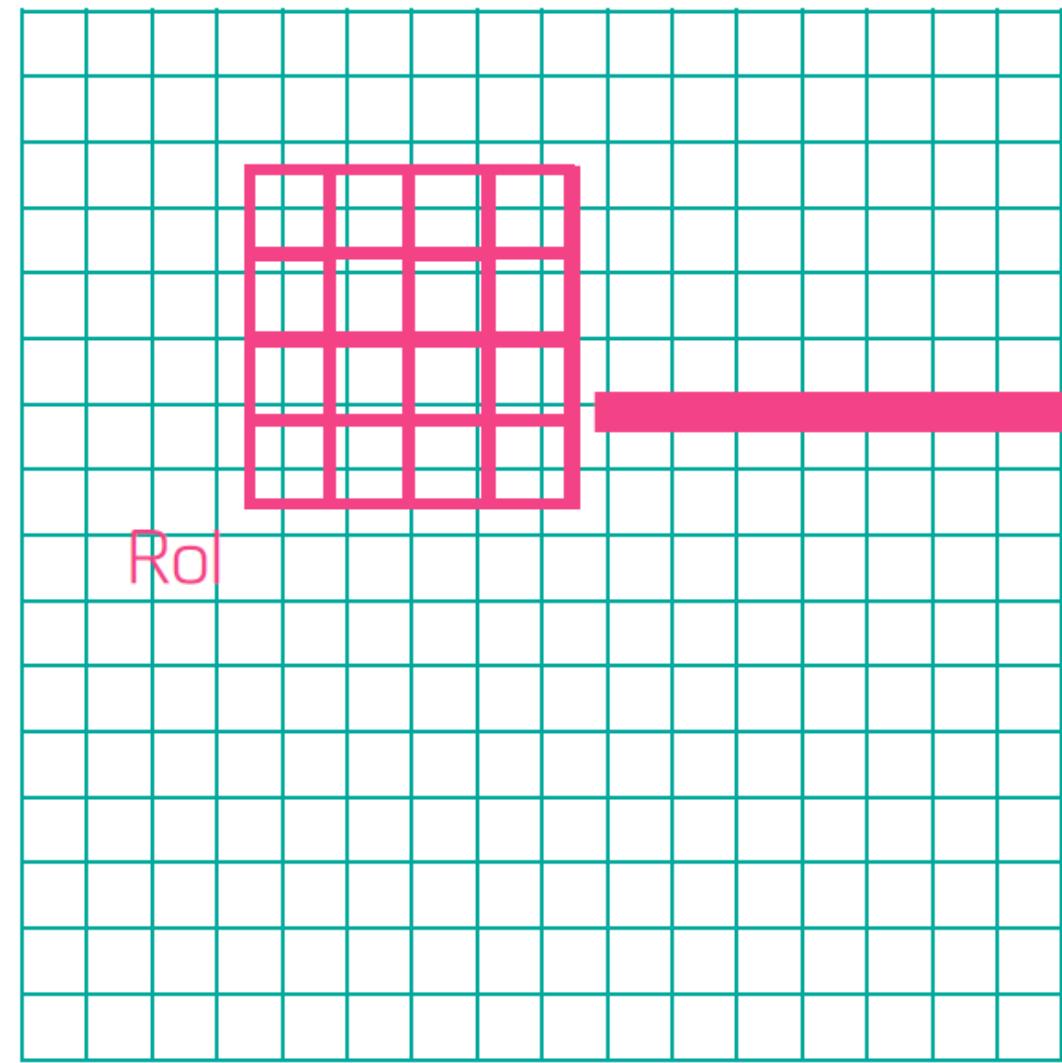
# RoI Align

Feature Map



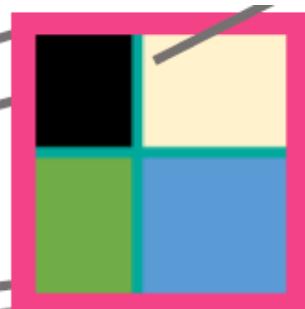
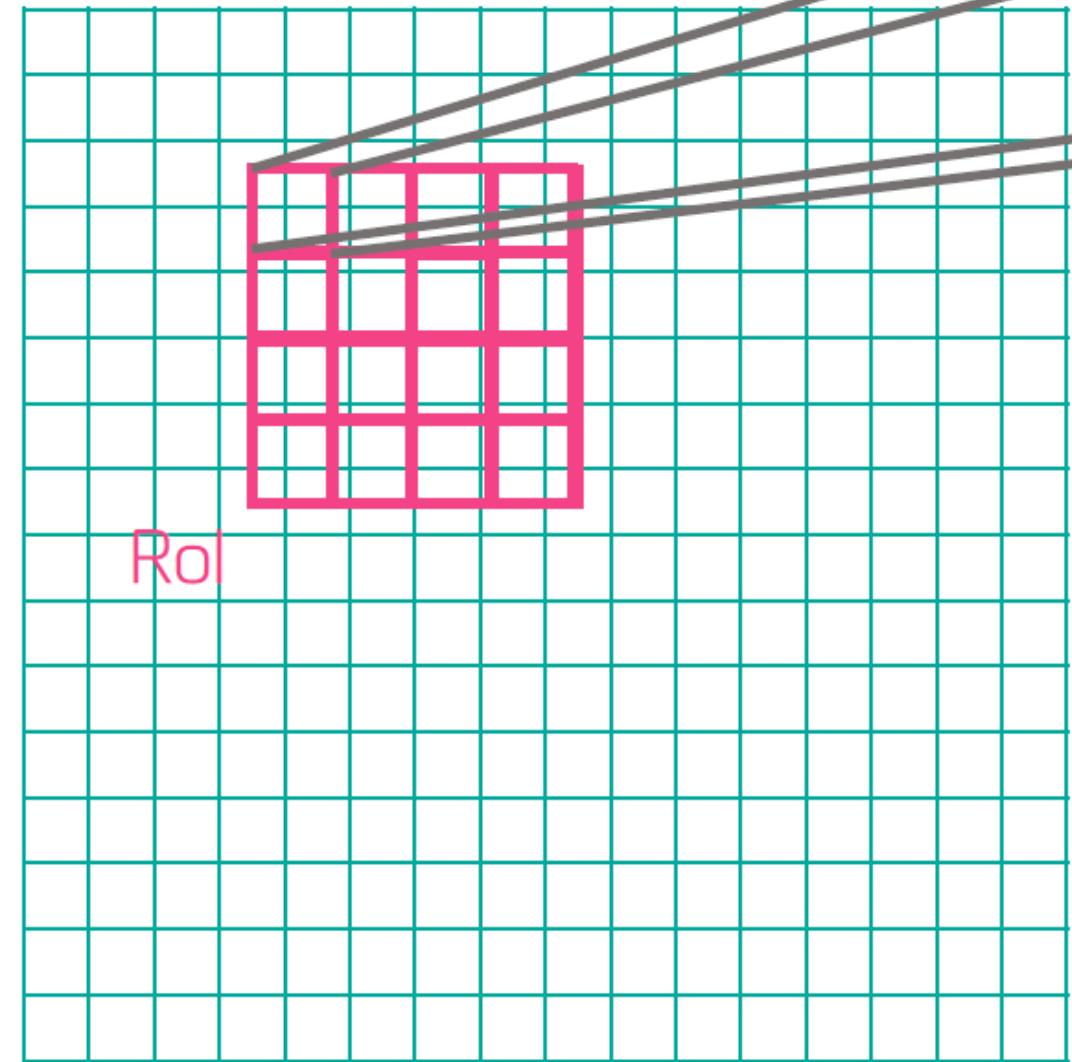
# RoI Align

Feature Map



2x2 Subcells for Precision

# RoI Align



= 0.15



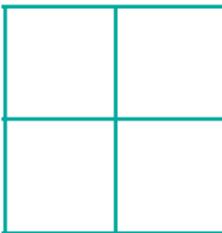
+ 0.25



+ 0.25

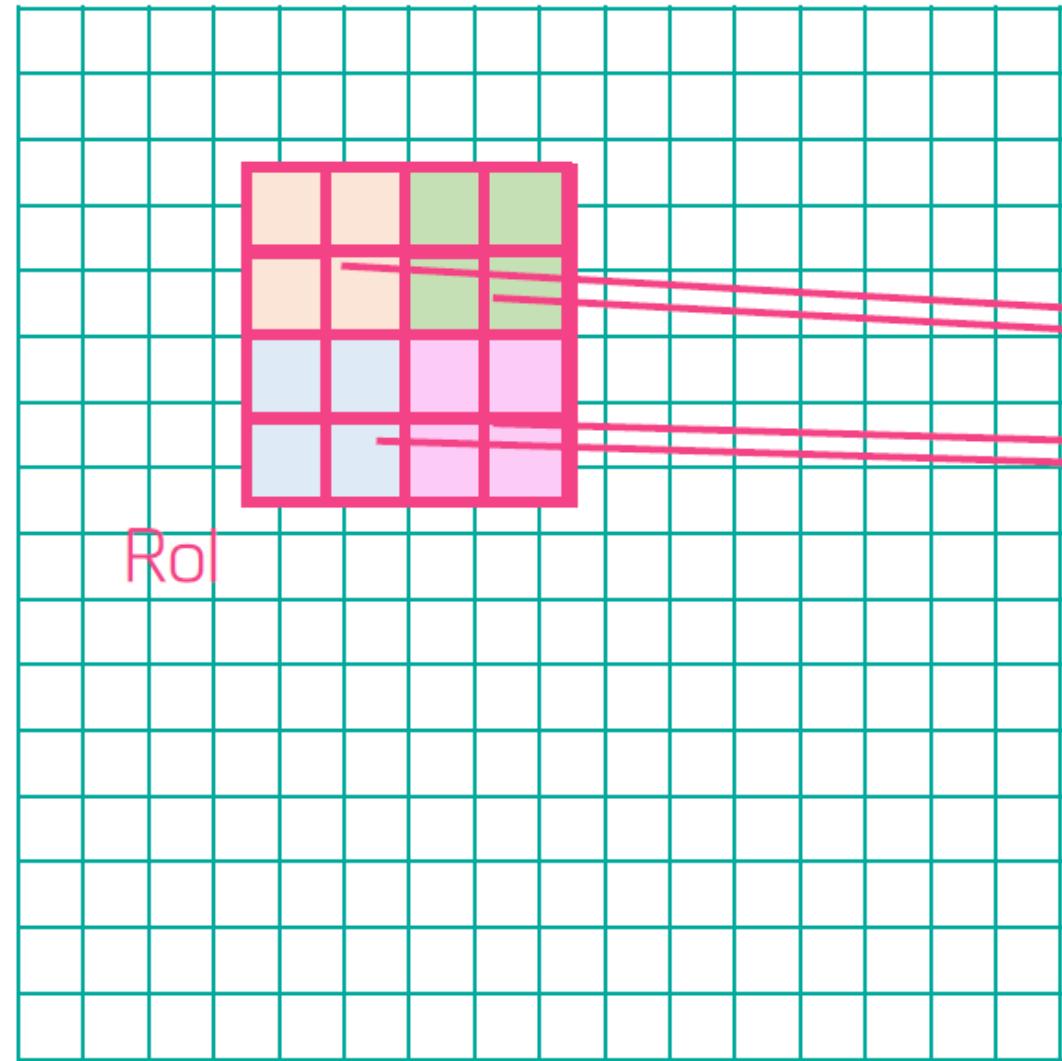


+ 0.35

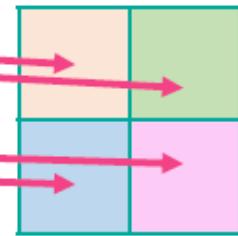


# RoI Align

Feature Map



2x2 Subcell Max Pooling



# Result

