



Computer Vision; Image Transformation; Semantic Segmentation



[YouTube Playlist](#)

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu

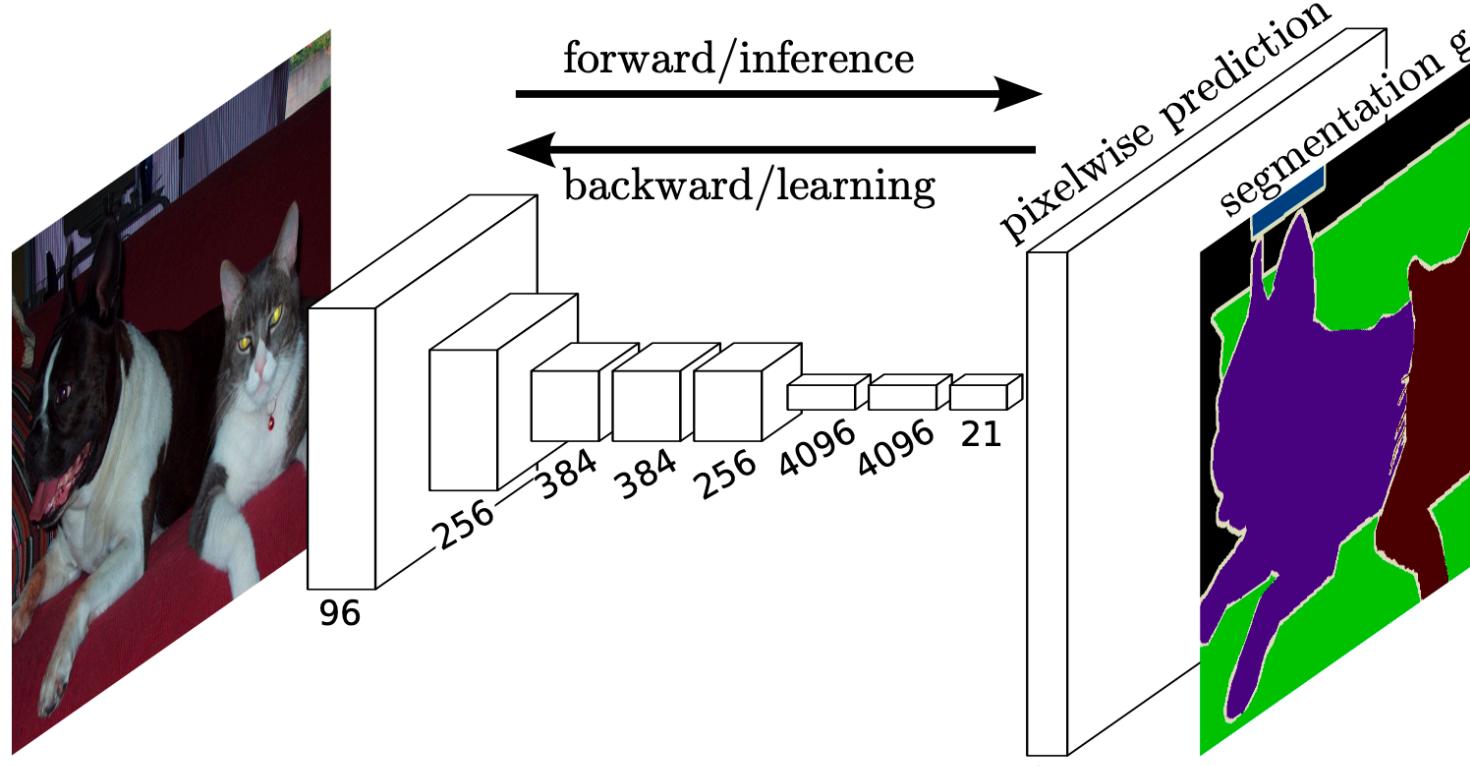


Boulder

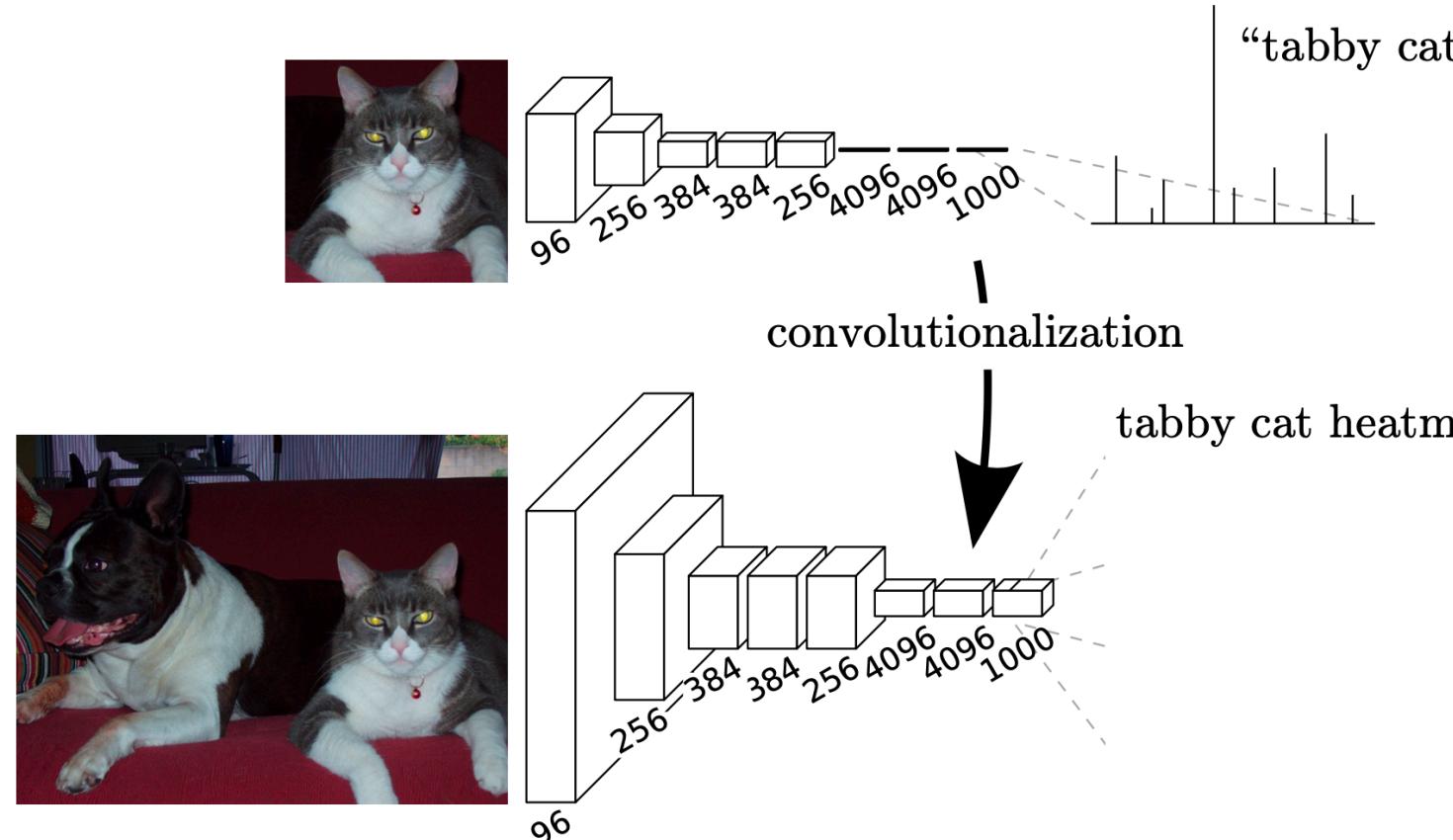


Fully Convolutional Networks for Semantic Segmentation

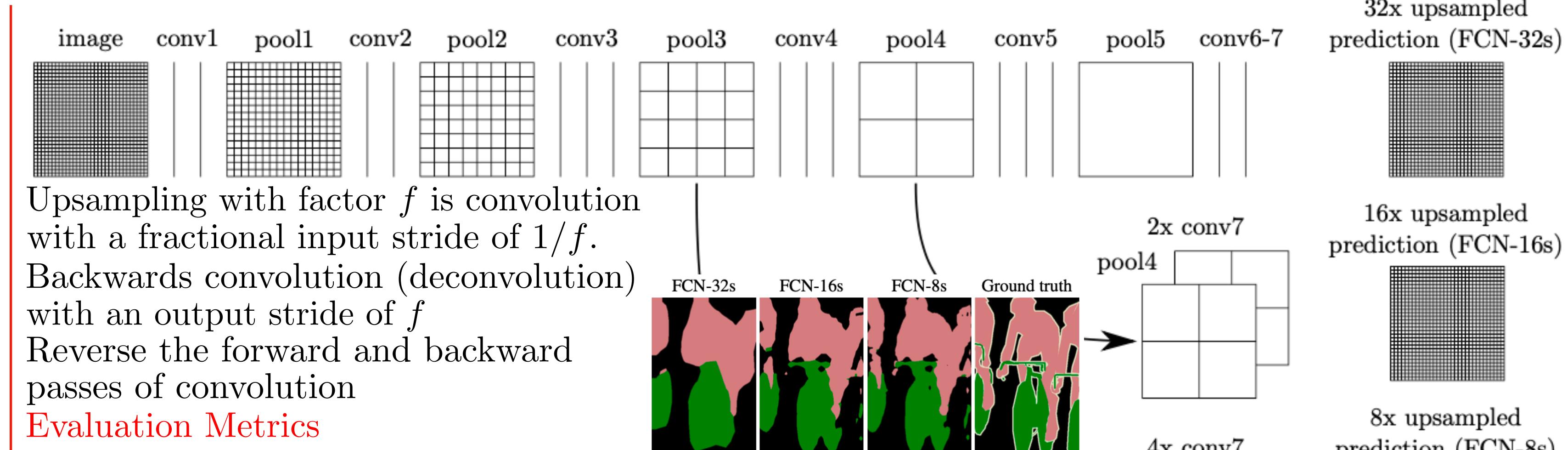
[YouTube Playlist](#)



global information resolves **what**
local information resolves **where**



The fully connected layers can also be viewed as convolutions with kernels that cover their entire input regions.



- **pixel accuracy:** $\sum_i n_{ii} / \sum_i t_i$
- **mean accuracy:** $(1/n_{cl}) \sum_i n_{ii} / t_i$
- **mean IU:** $(1/n_{cl}) \sum_i n_{ii} / (t_i + \sum_j n_{ji} - n_{ii})$
- **frequency weighted IU:**
 $(\sum_k t_k)^{-1} \sum_i t_i n_{ii} / (t_i + \sum_j n_{ji} - n_{ii})$

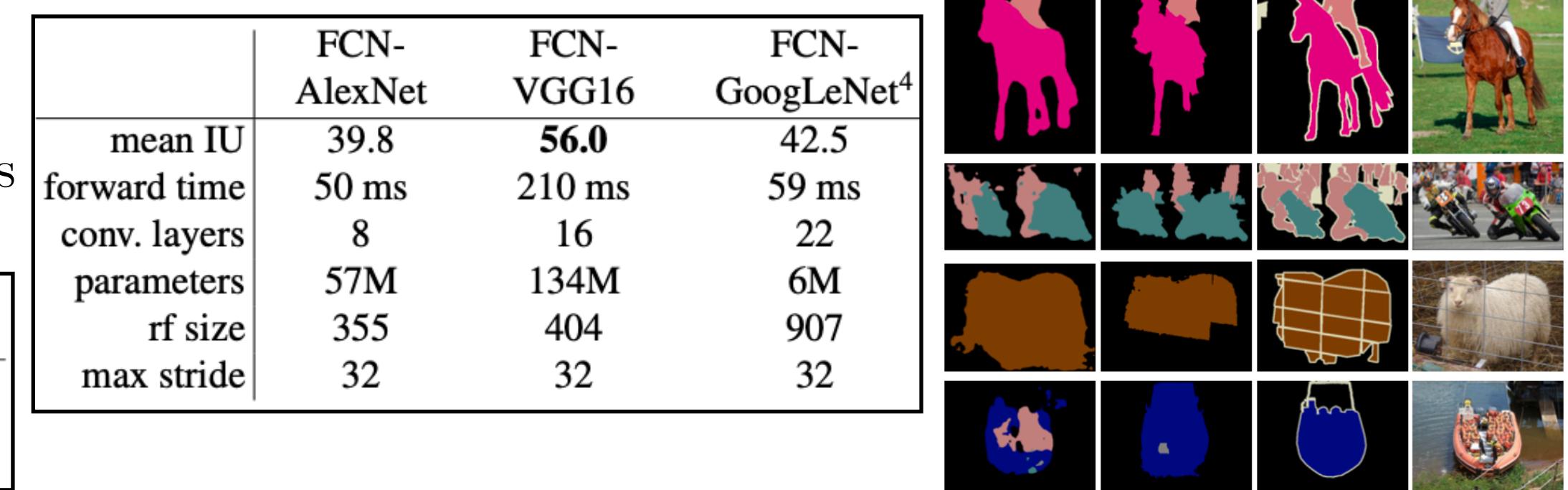
n_{ij} → number of pixels of class i predicted to belong to class j

n_{cl} → number of different classes

$t_i = \sum_j n_{ij}$ → total number of pixels of class i

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [10]	47.9	-	-
SDS [15]	52.6	51.6	~ 50 s
FCN-8s	62.7	62.2	~ 175 ms

	pixel acc. FCN-32s-fixed	mean acc. FCN-32s	mean IU FCN-32s	f.w. IU FCN-32s
FCN-32s-fixed	83.0	59.7	45.4	72.0
FCN-32s	89.1	73.3	59.4	81.4
FCN-16s	90.0	75.7	62.4	83.0
FCN-8s	90.3	75.9	62.7	83.2





Boulder

Learning Deconvolution Network for Semantic Segmentation

Pre-defined fixed-size receptive field!



(a) Inconsistent labels due to large object size



(b) Missing labels due to small object size

Instance-wise prediction!

$g_i \in \mathbb{R}^{W \times H \times C}$ → output score maps of the i -th proposal

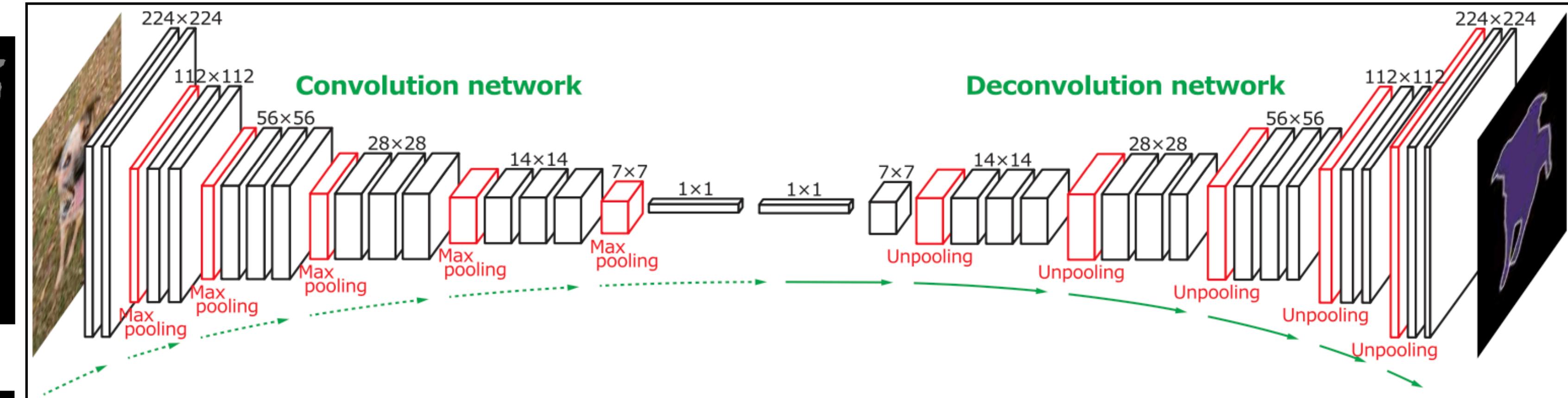
edge-box

$G_i \rightarrow$ zero padded outside g_i

$$P(x, y, c) = \max_i G_i(x, y, c)$$

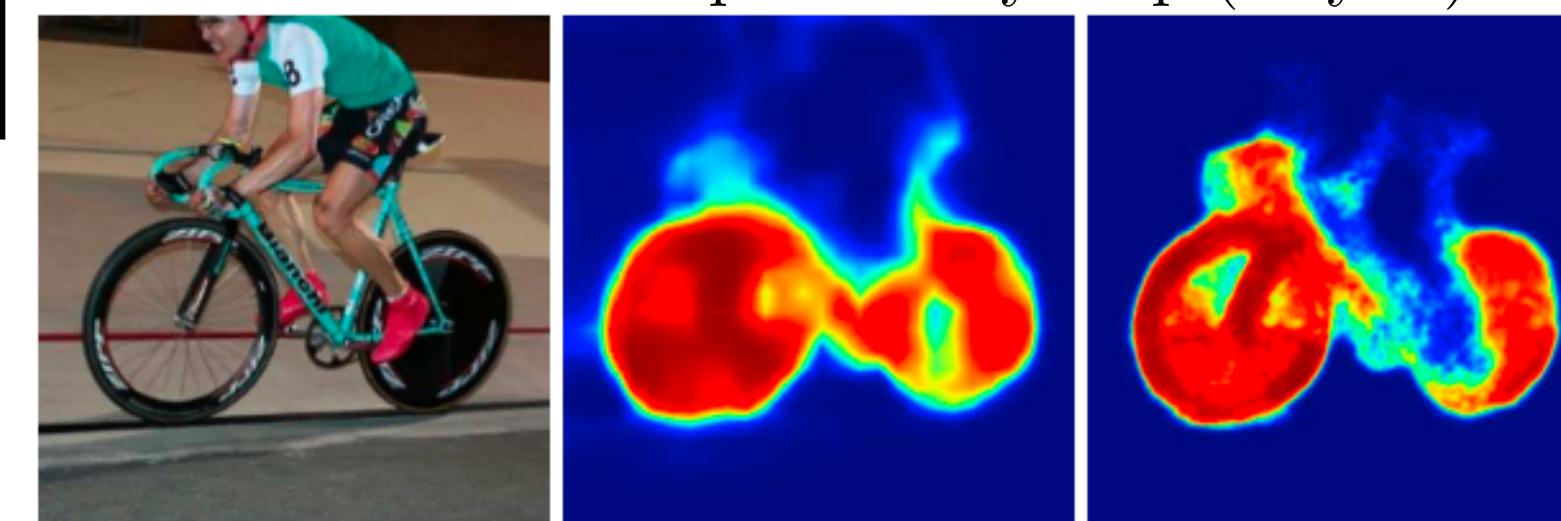
$$P(x, y, c) = \sum_i G_i(x, y, c) \quad \text{(before softmax)}$$

Noh, Hyenwoo, Seunghoon Hong, and Bohyung Han. "Learning deconvolution network for semantic segmentation." *Proceedings of the IEEE international conference on computer vision*. 2015.



The detailed structures of an object are often lost or smoothed because the label map, input to the deconvolutional layer, is too coarse and deconvolution procedure is overly simple.

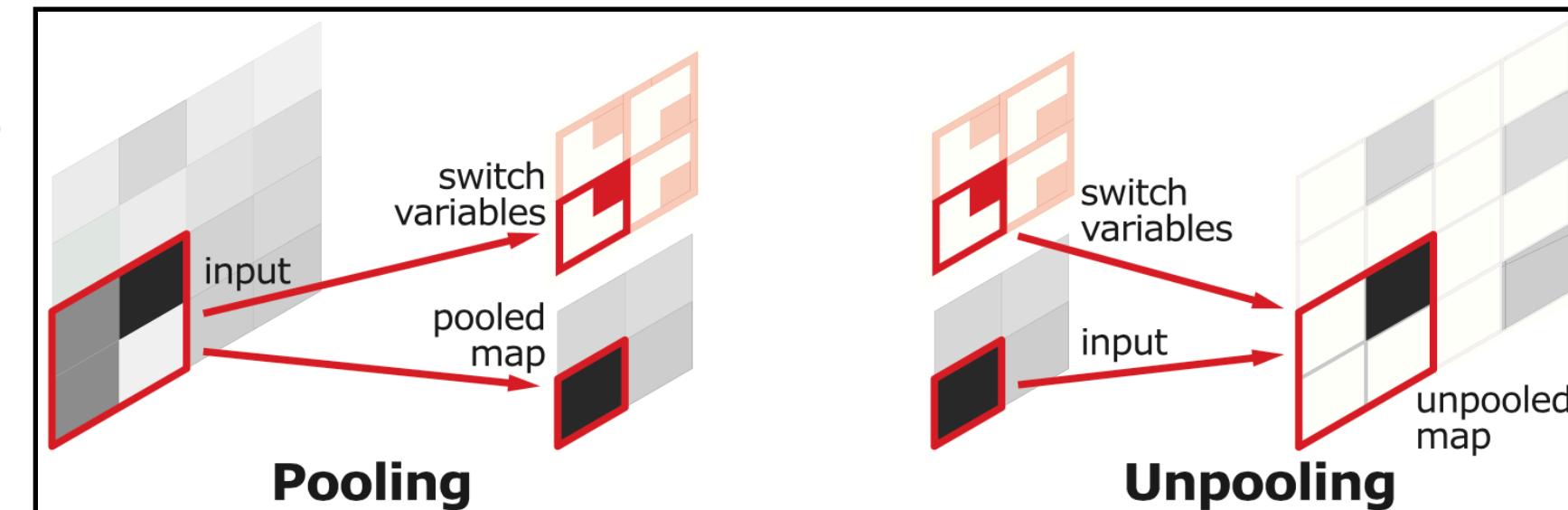
Class conditional probability map (bicycle)



(a) Input image

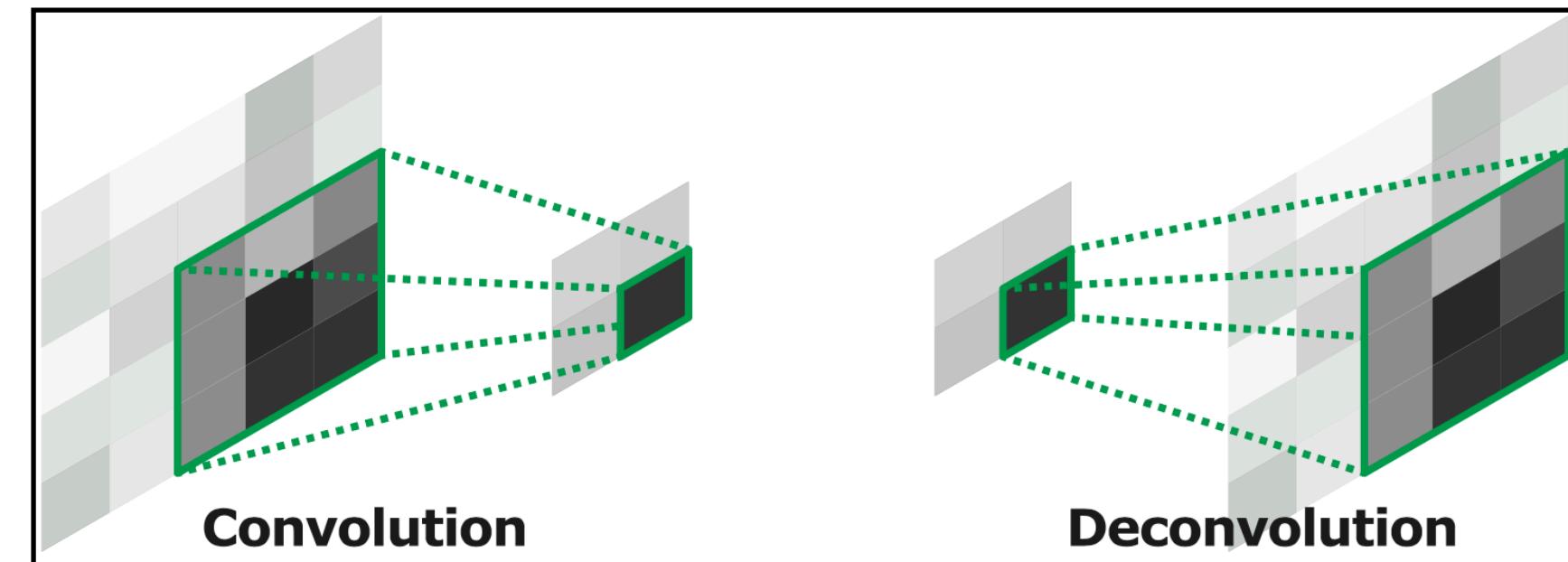
(b) FCN-8s

(c) Ours



Pooling

Unpooling



Convolution

Deconvolution

Batch Normalization

Two-stage Training:

- 1) ground-truth bounding boxes and 2) object proposals (≥ 0.5 in IoU)

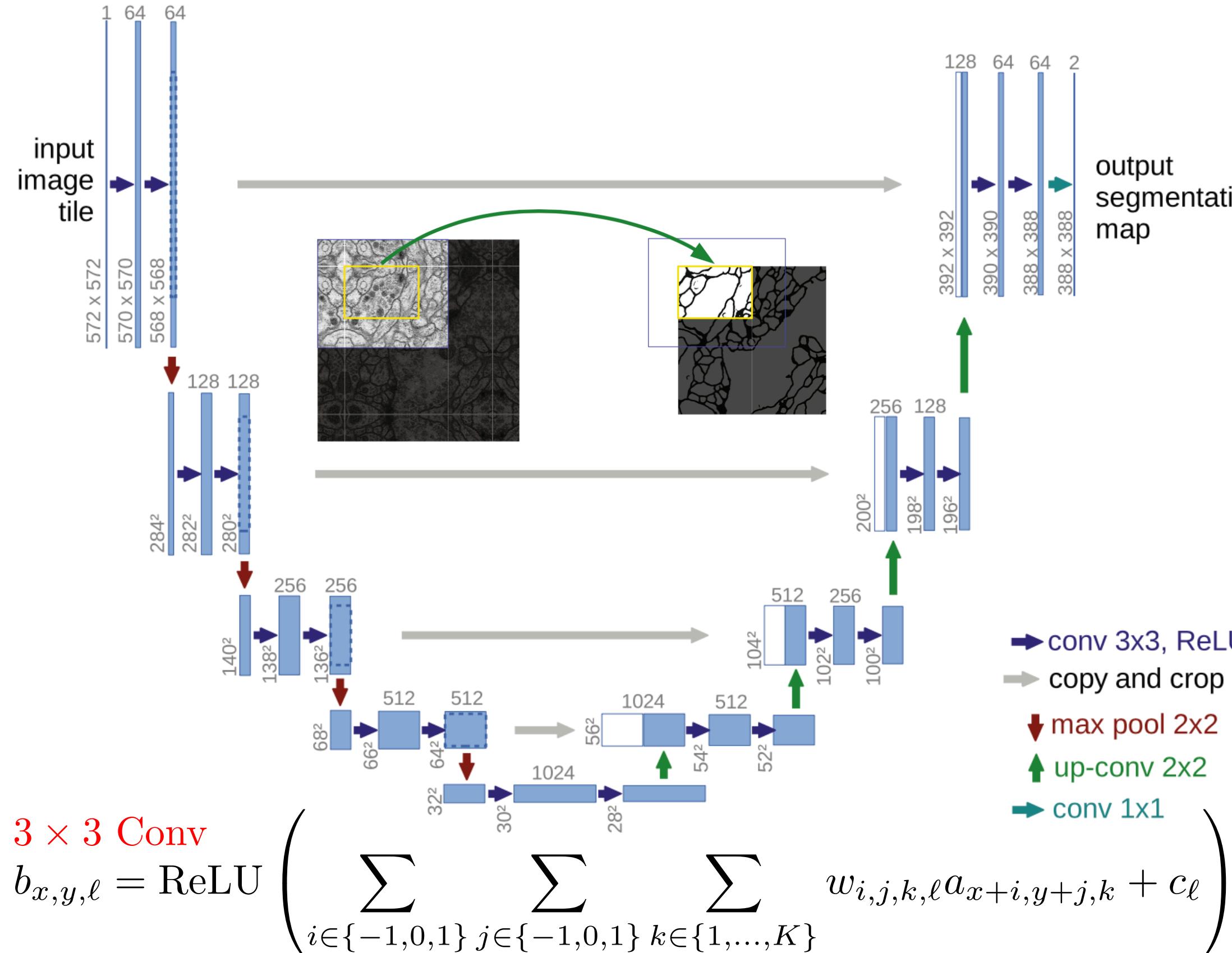


Boulder

U-Net: Convolutional Networks for Biomedical Image Segmentation



[YouTube Video](#)



3 × 3 Conv

$$b_{x,y,\ell} = \text{ReLU} \left(\sum_{i \in \{-1,0,1\}} \sum_{j \in \{-1,0,1\}} \sum_{k \in \{1, \dots, K\}} w_{i,j,k,\ell} a_{x+i,y+j,k} + c_\ell \right)$$

2 × 2 maxpooling

$$b_{x,y,k} = \max_{i,j \in \{0,1\}} a_{2x+i,2y+j,k} \rightarrow \text{stride} = 2$$

2 × 2 up-conv

$$b_{2x+i,2y+j,k} = \text{ReLU} \left(\sum_{k \in \{1, \dots, K\}} w_{i,j,k,\ell} a_{x,y,k} + c_\ell \right) \text{ for } i, j \in \{0, 1\}$$

$$\mathcal{L} = - \sum_{(x,y) \in \Omega} w(x,y) \log p_{\ell(x,y)}(x,y)$$

$$\ell : \Omega \rightarrow \{1, \dots, K\}$$

$$(x,y) \mapsto \ell(x,y)$$

↙ true label of each pixel

$$p_k(x,y) = \exp(a_k(x,y)) / \left(\sum_{k'=1}^K \exp(a_{k'}(x,y)) \right)$$

$$w(x,y) = w_c(x,y) + w_0 \exp\left(-\frac{(d_1(x,y) + d_2(x,y))^2}{2\sigma^2}\right)$$

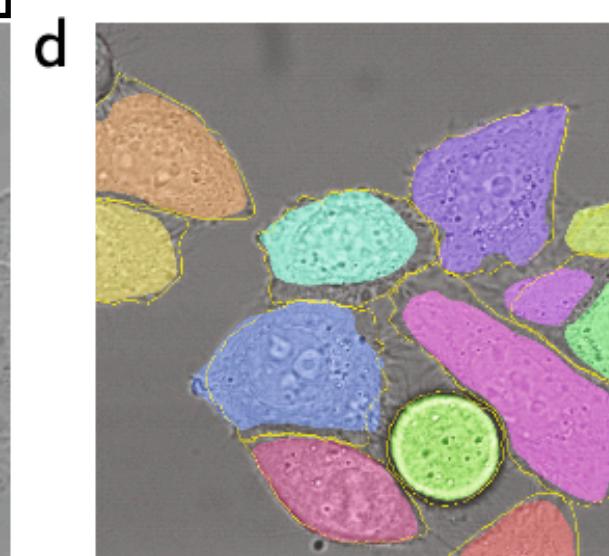
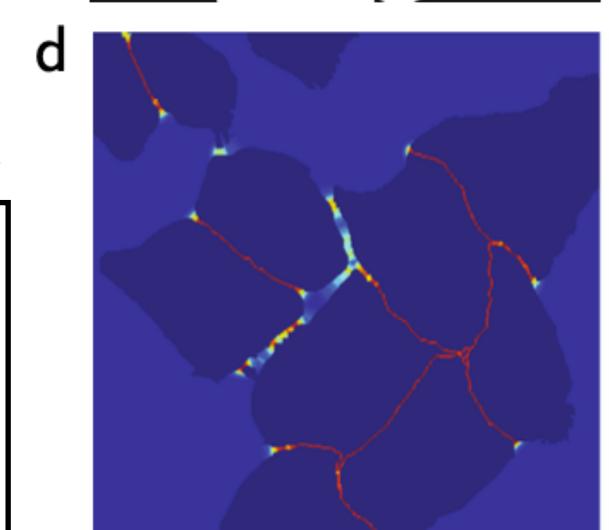
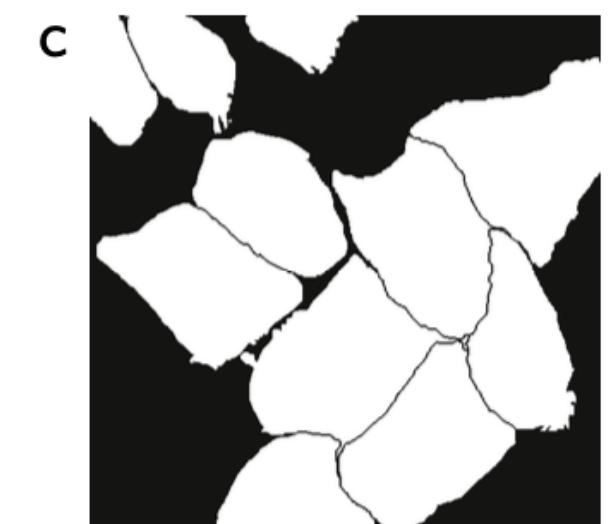
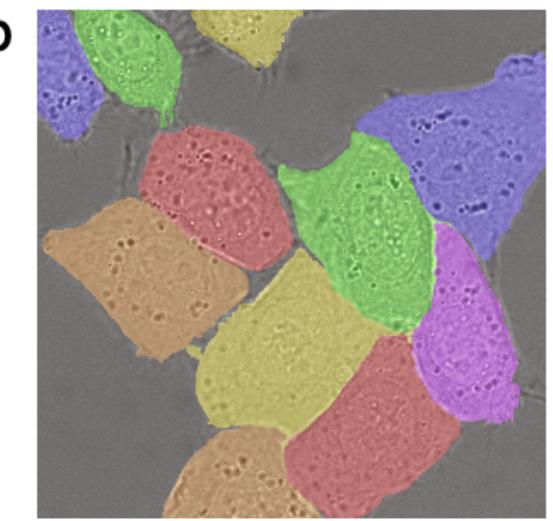
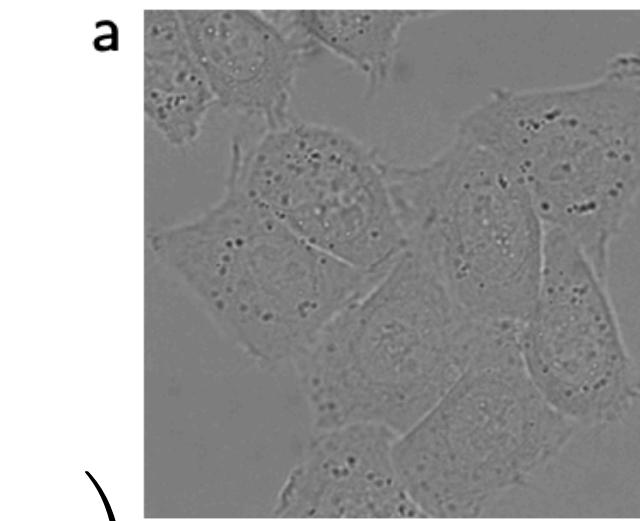
$w_c(x,y) \rightarrow$ weight map to balance class frequencies

$w_0 = 10$ & $\sigma \approx 5$ pixels

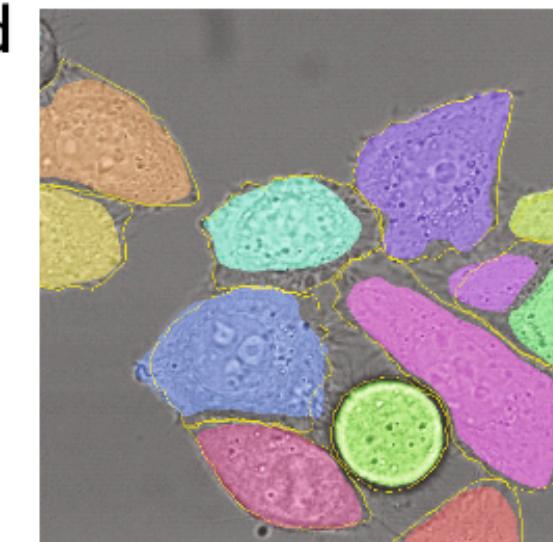
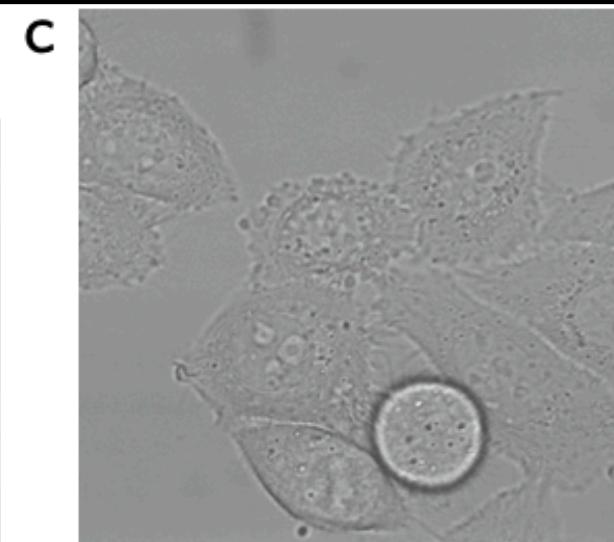
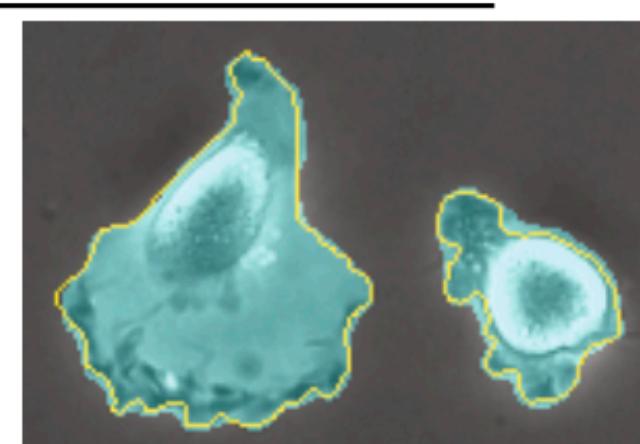
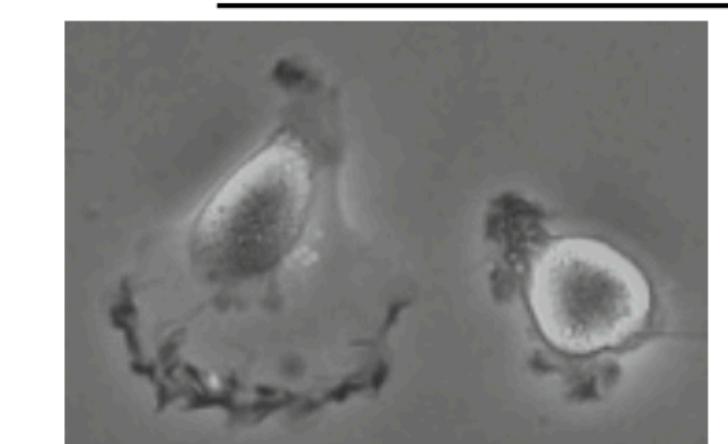
$d_1(x,y) \rightarrow$ distance to the border of the nearest cell

$d_2(x,y) \rightarrow$ distance to the border of the second nearest cell

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)	0.9203	0.7756



Essential data augmentation: shift and rotation invariances as well as robustness to deformations and gray value variations





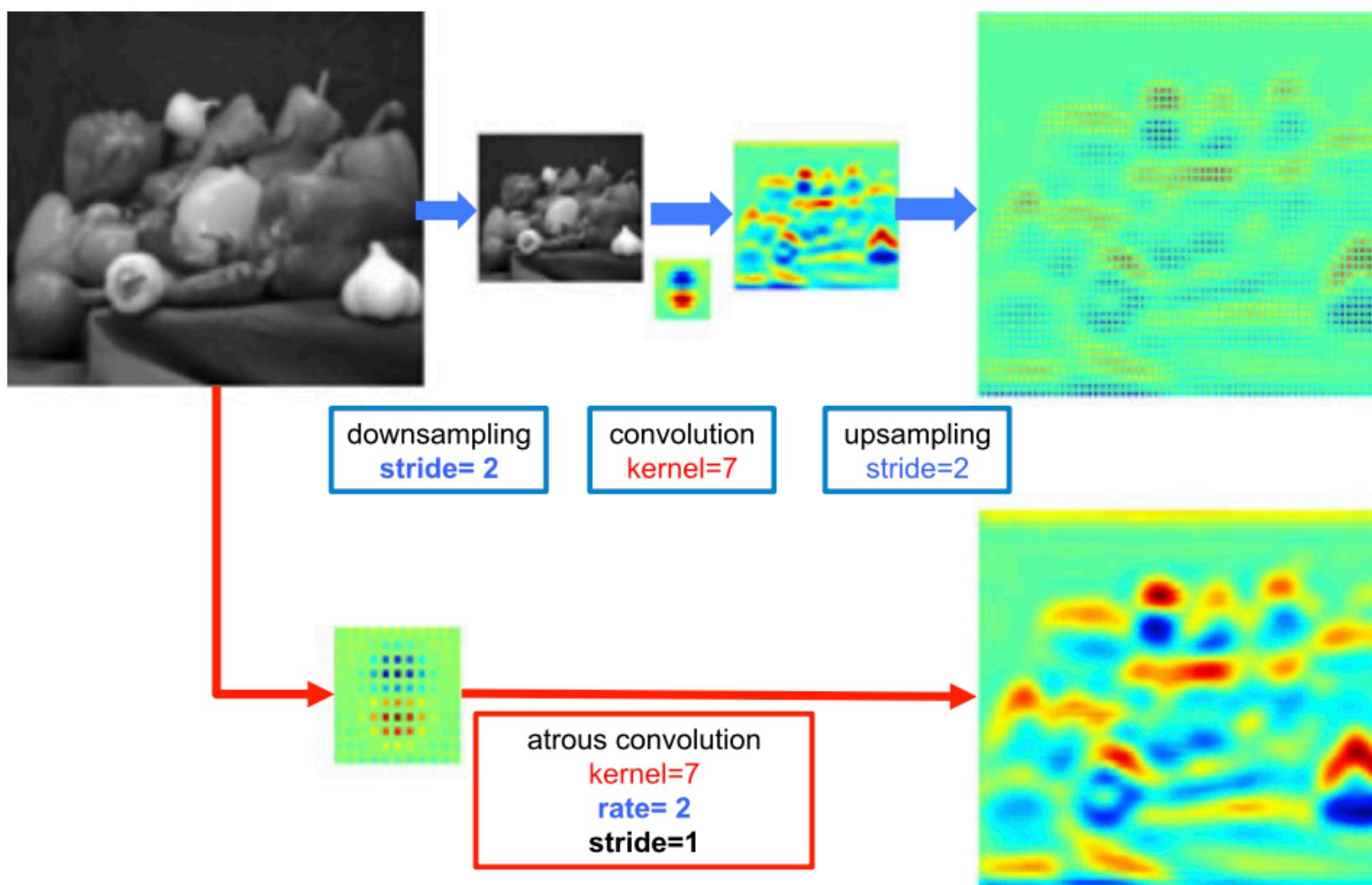
DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs


[YouTube Playlist](#)

Three challenges in the application of DCNNs to semantic image segmentation: (1) reduced feature resolution, (2) existence of objects at multiple scales, and (3) reduced localization accuracy due to DCNN invariance.

Atrous Convolution

Reduce the degree of signal downampling due to max-pooling and striding (from 32x down to 8x).

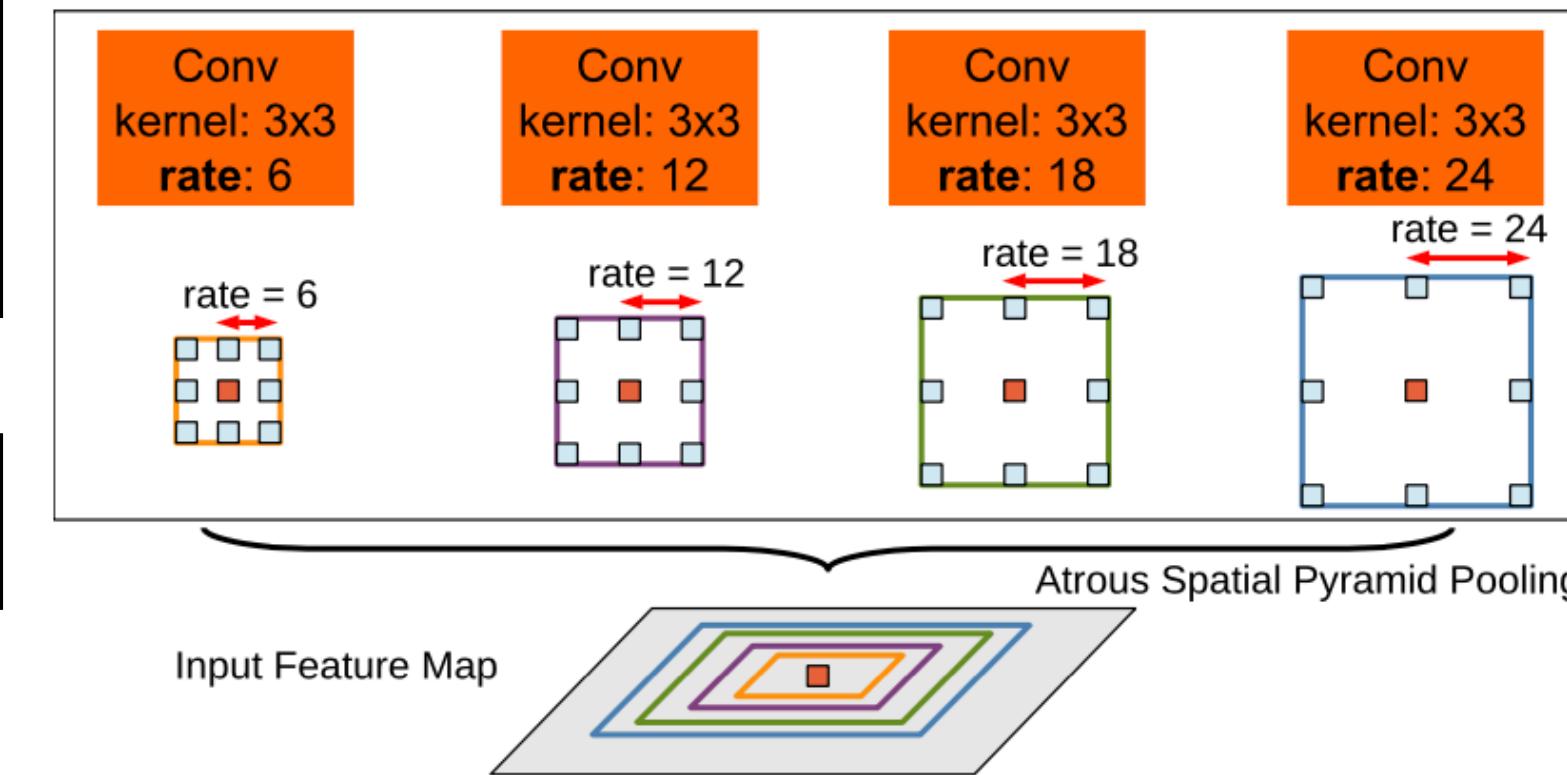


$$y[i] = \sum_{k=1}^K x[i + r \cdot k] w[k]$$

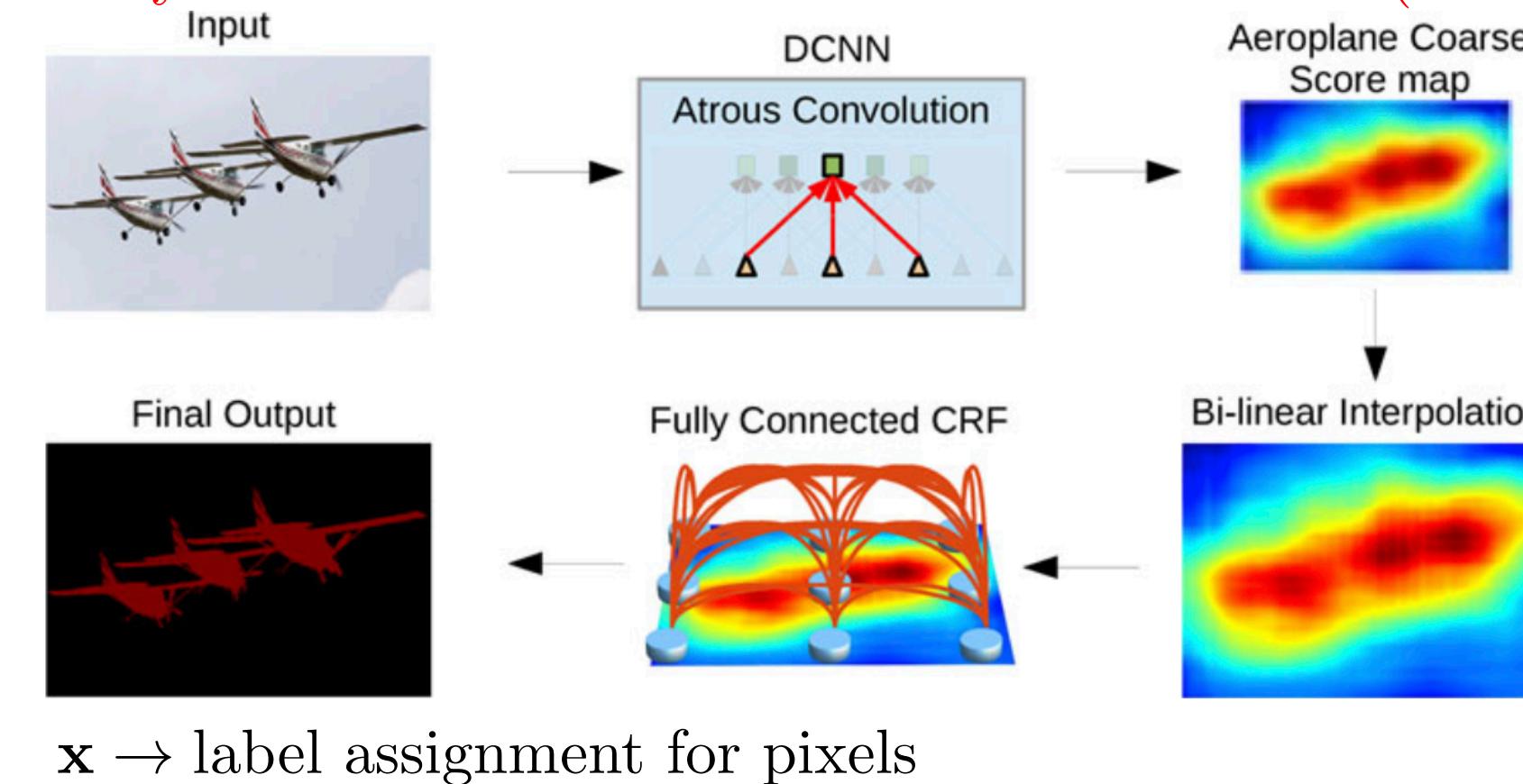
rate parameter

Same number of parameters and amount of computation

Atrous Spatial Pyramid Pooling (ASPP)



Fully-Connected Conditional Random Fields (CRF)

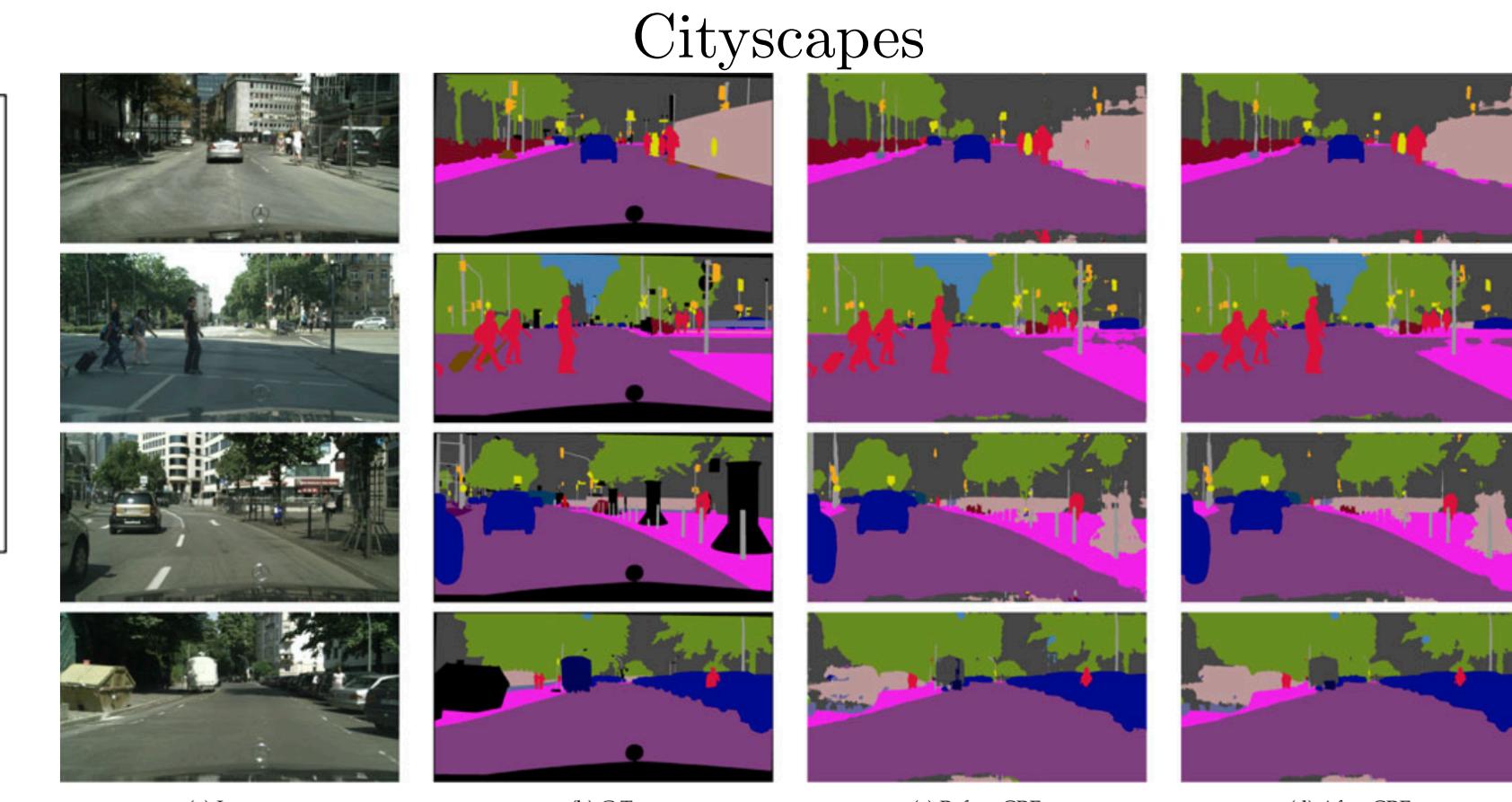


$x \rightarrow$ label assignment for pixels

$$E(x) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j) \quad \theta_i(x_i) = -\log P(x_i)$$

energy function

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



Method	before CRF			after CRF		
	PASCAL VOC	VGG-16	ResNet-101	PASCAL VOC	LargeFOV	mIOU
LargeFOV	65.76	69.84	68.72			
ASPP-S	66.98	69.73	71.27			
ASPP-L	68.96	71.57	73.28			
MSC						
COCO						
Aug						
LargeFOV						
ASPP						
CRF						

$P(x_i) \rightarrow$ label assignment prob. at pixel i (DCNN)

$p_i \rightarrow$ pixel position

$I_i \rightarrow$ RGB color

$\mu(x_i, x_j) = 1$ iff $x_i \neq x_j$



Boulder



[YouTube Playlist](#)

Multi-scale Context Aggregation by Dilated Convolutions

$F : \mathbb{Z}^2 \rightarrow \mathbb{R}$
discrete function

$$\Omega_r = [-r, r]^2 \cap \mathbb{Z}^2$$

$k : \Omega_r \rightarrow \mathbb{R}$
discrete filter of size $(2r + 1)^2$

$$(F * k)(p) = \sum_{t \in \Omega_r} F(p - t)k(t)$$

discrete convolution operator

$$(F *_{\ell} k)(p) = \sum_{t \in \Omega_r} F(p - \ell t)k(t)$$

ℓ -dilated convolution

algorithme à trous
(an algorithm for
wavelet decomposition)
uses dilated convolutions

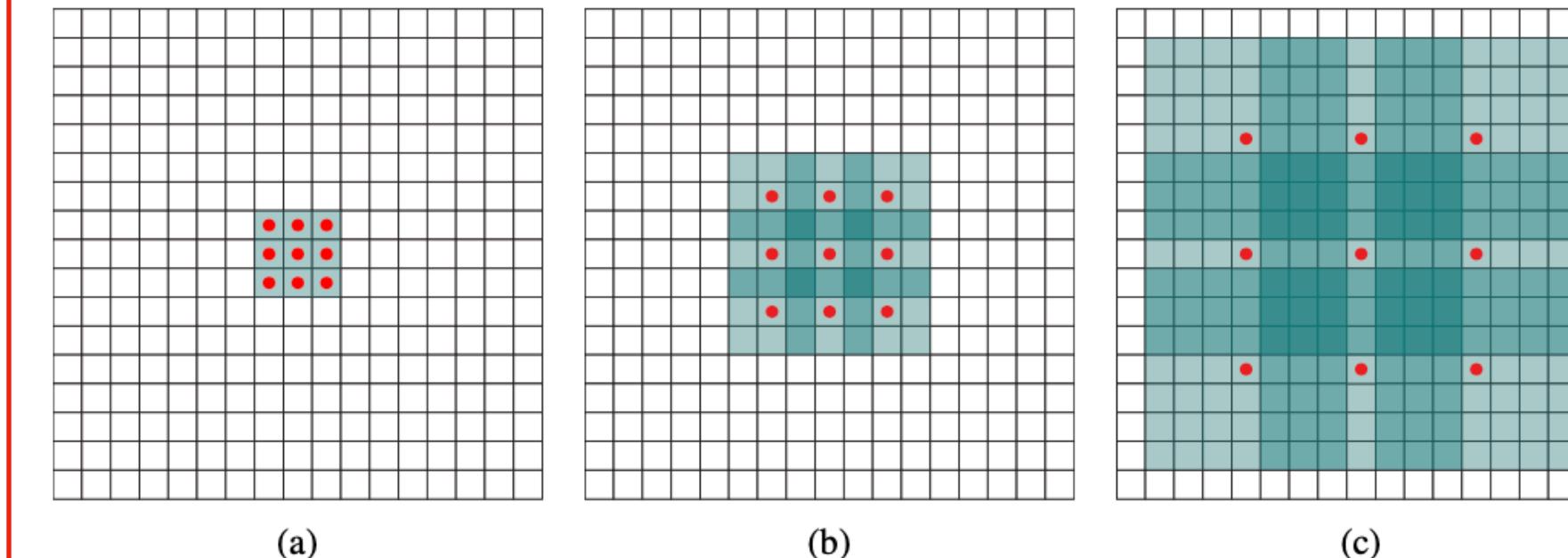
Dilated convolutions support
exponentially expanding
receptive fields without losing
resolution or coverage.

$$F_{i+1} = F_i *_{2^i} k_i \quad \text{for } i = 0, 1, \dots, n-2.$$

3×3 filters

Receptive field of an element p
in F_{i+1} is the set of elements in
 F_0 that modify the value of $F_{i+1}(p)$

Size of the receptive field of each element in F_{i+1} is $(2^{i+2} - 1) \times (2^{i+2} - 1)$.



Context network architecture

Layer	1	2	3	4	5	6	7	8
Convolution	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1
Dilation	1	1	2	4	8	16	1	1
ReLU	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Receptive field	3×3	5×5	9×9	17×17	33×33	65×65	67×67	67×67

Output channels								
Basic	C	C	C	C	C	C	C	C
Large	$2C$	$2C$	$4C$	$8C$	$16C$	$32C$	$32C$	C

$$k^b(t, a) = 1_{[t=0]} 1_{[a=b]} \rightarrow \text{identity initialization}$$

$a \rightarrow$ index of the input feature map

$b \rightarrow$ index of the output feature map

$$k^b(t, a) = \begin{cases} \frac{C}{c_{i+1}} & t = 0 \text{ and } \left\lfloor \frac{aC}{c_i} \right\rfloor = \left\lfloor \frac{bC}{c_{i+1}} \right\rfloor \rightarrow \text{identity initialization (Large)} \\ \varepsilon & \text{otherwise } \varepsilon \sim \mathcal{N}(0, \sigma^2) \text{ and } \sigma \ll C/c_{i+1} \end{cases}$$

c_i and $c_{i+1} \rightarrow$ number of feature maps in two consecutive layers

$C \rightarrow$ divides both c_i & c_{i+1}

Front End VGG-16

input: padded images (reflection padding)

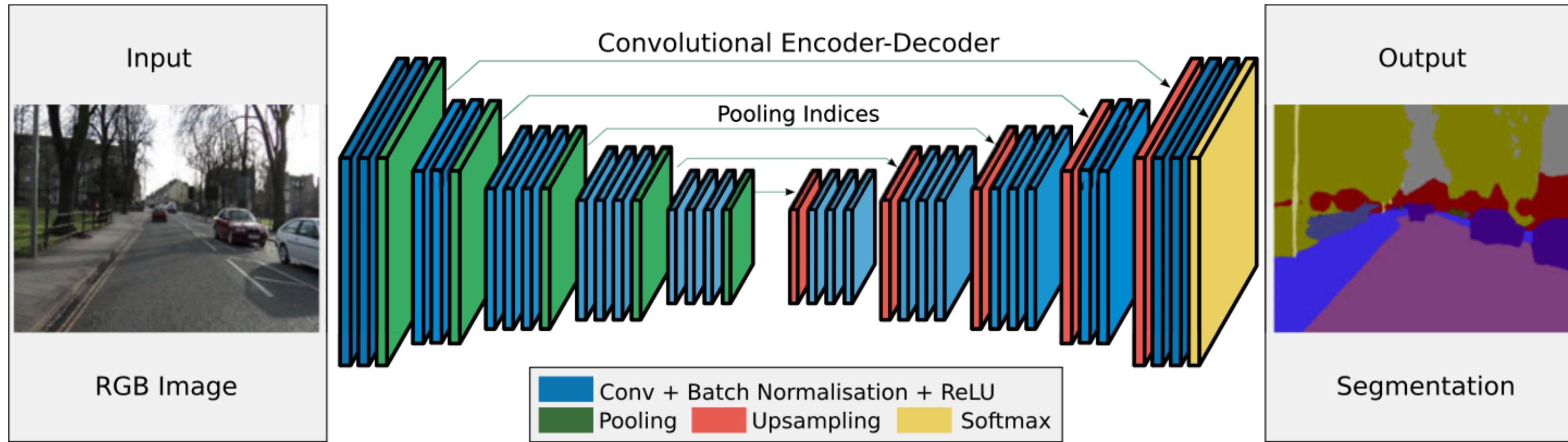
output: $64 \times 64 \times 21$ feature maps $C = 21$

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
FCN-8s	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
DeepLab	72	31	71.2	53.7	60.5	77	71.9	73.1	25.2	62.6	49.1	68.7	63.3	73.9	73.6	50.8	72.3	42.1	67.9	52.6	62.1
DeepLab-Msc	74.9	34.1	72.6	52.9	61.0	77.9	73.0	73.7	26.4	62.2	49.3	68.4	64.1	74.0	75.0	51.7	72.7	42.5	67.2	55.7	62.9
Our front end	82.2	37.4	72.7	57.1	62.7	82.8	77.8	78.9	28	70	51.6	73.1	72.8	81.5	79.1	56.6	77.1	49.9	75.3	60.9	67.6

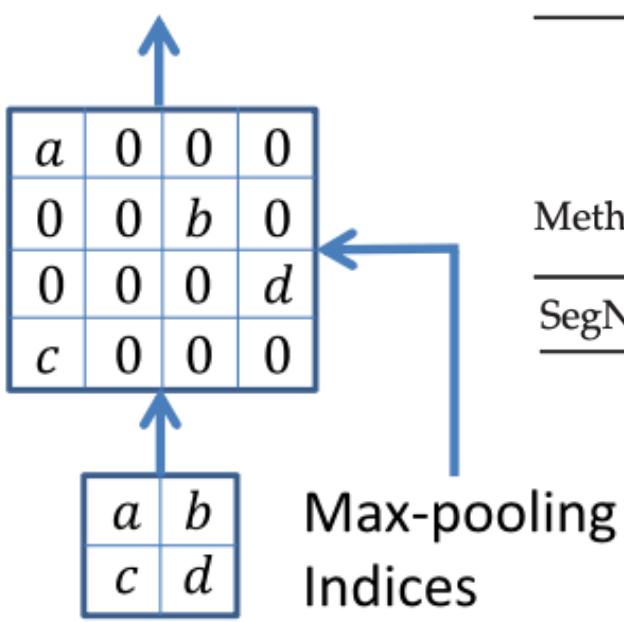


	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
Front end	86.3	38.2	76.8	66.8	63.2	87.3	78.7	82	33.7	76.7	53.5	73.7	76	76.6	83	51.9	77.8	44	79.9	66.3	69.8
Front + Basic	86.4	37.6	78.5	66.3	64.1	89.9	79.9	84.9	36.1	79.4	55.8	77.6	81.6	79	83.1	51.2	81.3	43.7	82.3	65.7	71.3
Front + Large	87.3	39.2	80.3	65.6	66.4	90.2	82.6	85.8	34.8	81.9	51.7	79	84.1	80.9	83.2	51.2	83.2	44.7	83.4	65.6	72.1
Front end + CRF	89.2	38.8	80	69.8	63.2	88.8	80	85.2	33.8	80.6	55.5	77.1	80.8	77.3	84.3	53.1	80.4	45	80.7	67.9	71.6
Front + Basic + CRF	89.1	38.7	81.4	67.4	65	91	81	86.7	37.5	81	57	79.6	83.6	79.9	84.6	52.7	83.3	44.3	82.6	67.2	72.7
Front + Large + CRF	89.6	39.9	82.7	66.7	67.5	91.1	83.3	87.4	36	83.3	52.5	80.7	85.7	81.8	84.4	52.6	84.4	45.3	83.7	66.7	73.3
Front end + RNN	88.8	38.1	80.8	69.1	65.6	89.9	79.6	85.7	36.3	83.6	57.3	77.9	83.2	77	84.6	54.7	82.1	46.9	80.9	66.7	72.5
Front + Basic + RNN	89	38.4	82.3	67.9	65.2	91.5	80.4	87.2	38.4	82.1	57.7	79.9	85	79.6	84.5	53.5	84	45	82.8	66.2	73.1
Front + Large + RNN	89.3	39.2	83.6	67.2	69	92.1	83.1	88	38.4	84.8	55.3	81.2	86.7	81.3	84.3	53.6	84.4	45.8	83.8	67	73.9

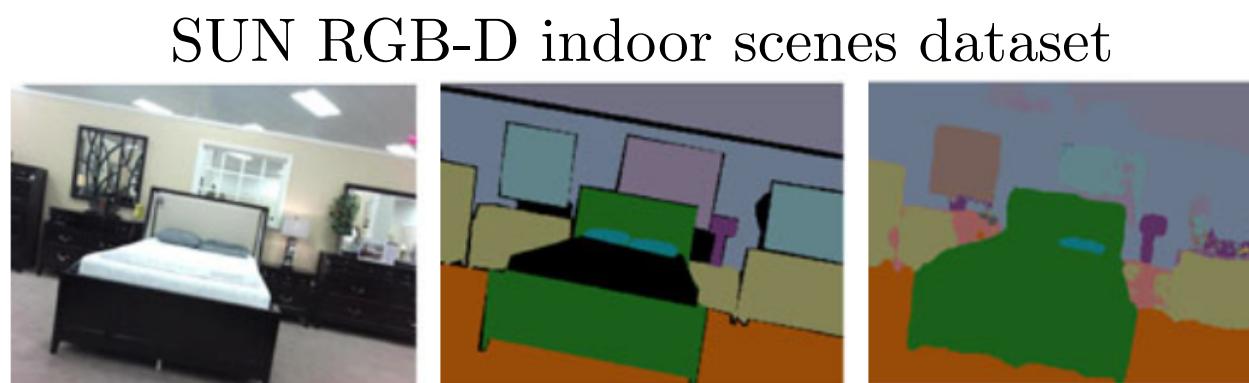
SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation


[YouTube Video](#)


Convolution with trainable decoder filters

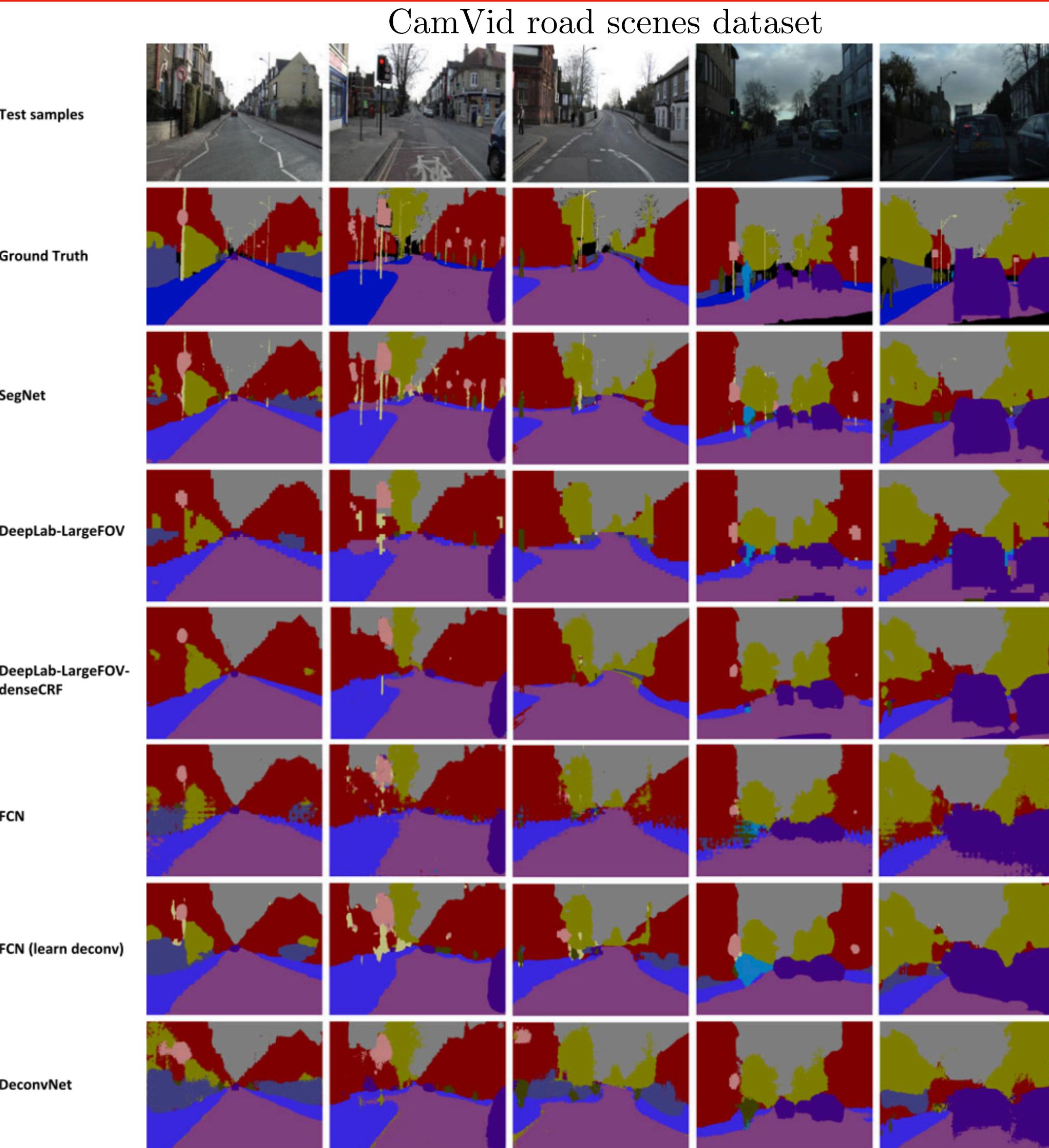


Method	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column-Pole	Side-walk	Bicyclist	mIoU
SegNet (3.5K dataset training - 140K)	89.6	83.4	96.1	87.7	52.7	96.4	62.2	53.45	32.1	93.3	36.5	90.40



A Comparison of Computational Time and Hardware Resources Required for Various Deep Architectures

Network	Forward pass(ms)	Backward pass(ms)	GPU training memory (MB)	GPU inference memory (MB)	Model size (MB)
SegNet	422.50	488.71	6803	1,052	117
DeepLab-LargeFOV [3]	110.06	160.73	5618	1,993	83
FCN (learnt deconv) [2]	317.09	484.11	9735	1,806	539
DeconvNet [4]	474.65	602.15	9731	1,872	877





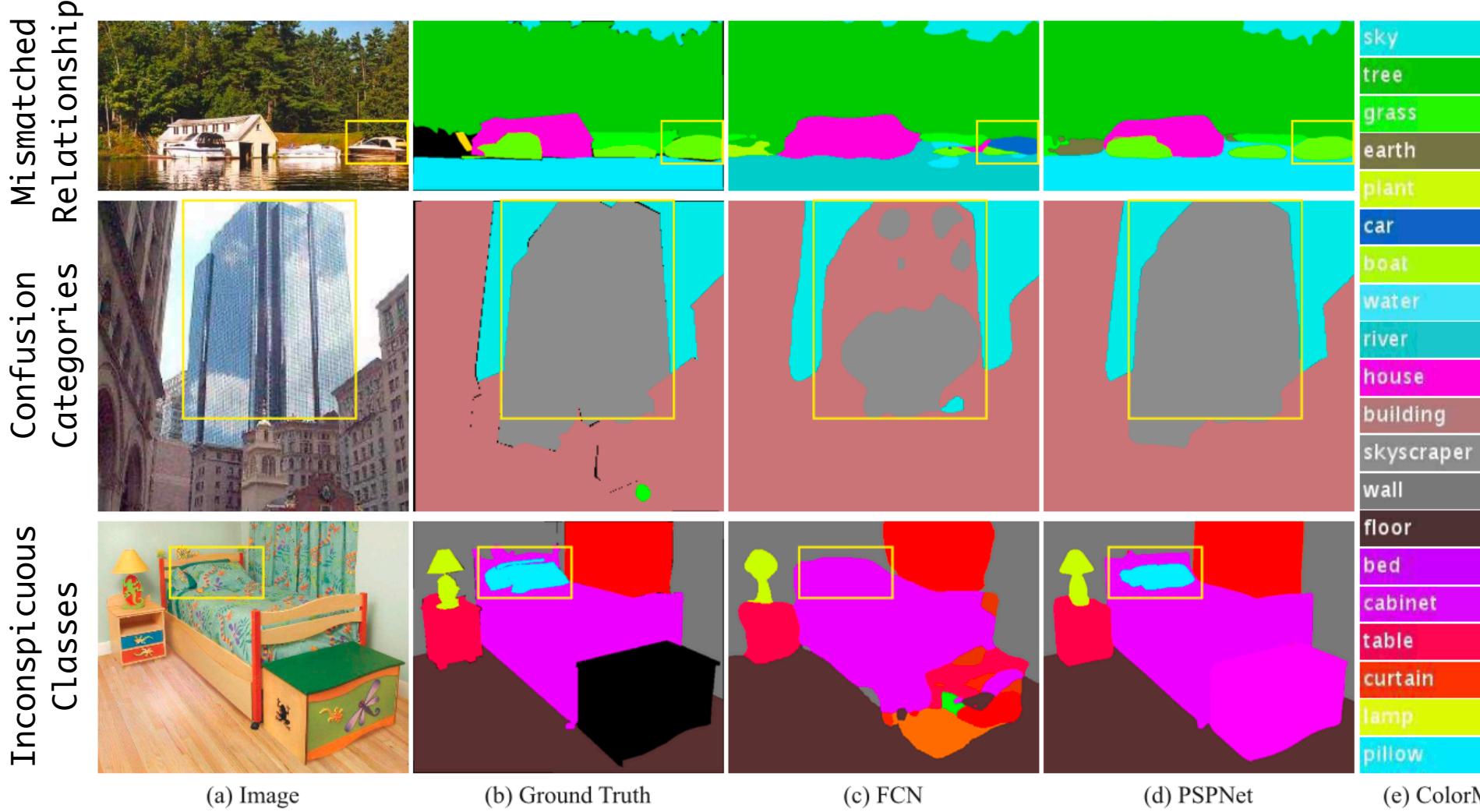
Boulder

Pyramid Scene Parsing Network



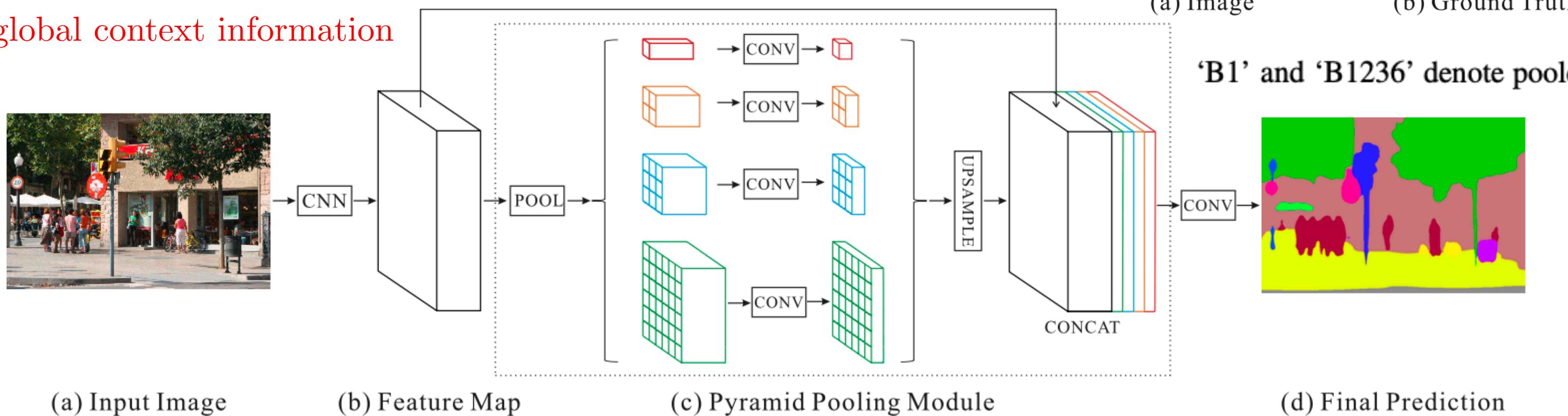
[YouTube Video](#)

Scene parsing issues on ADE20K dataset



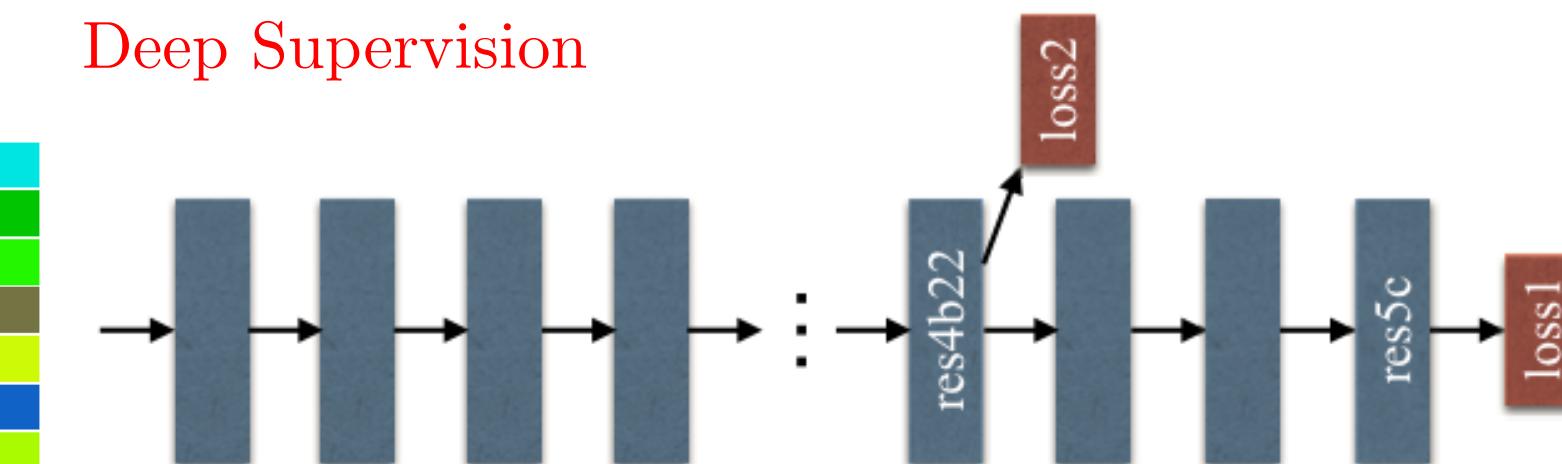
ADE20K dataset contains 150 stuff/object category labels (e.g., wall, sky, and tree) and 1,038 image-level scene descriptors (e.g., airport terminal, bedroom, and street).

global context information

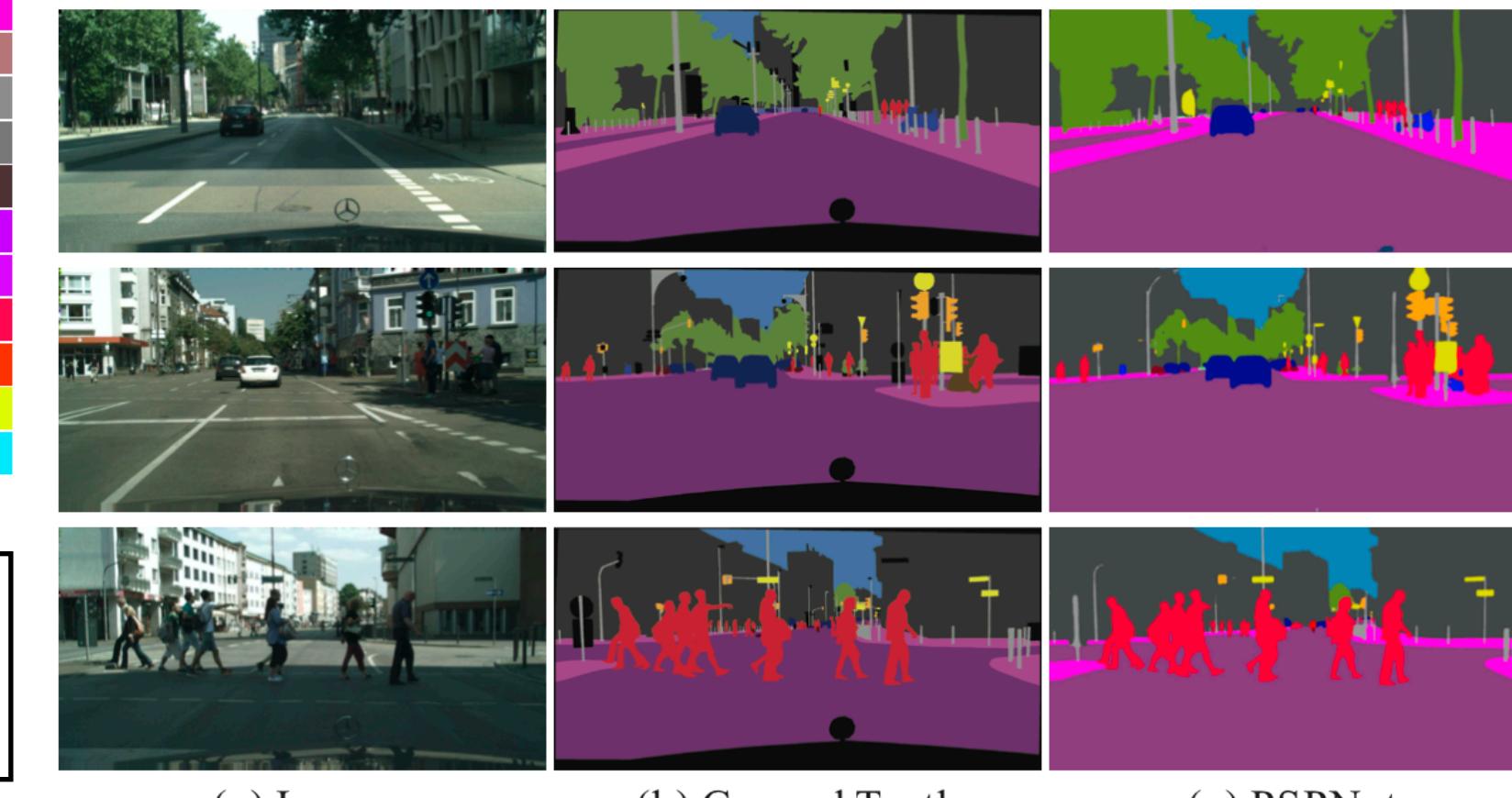


Zhao, Hengshuang, et al. "Pyramid scene parsing network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Deep Supervision

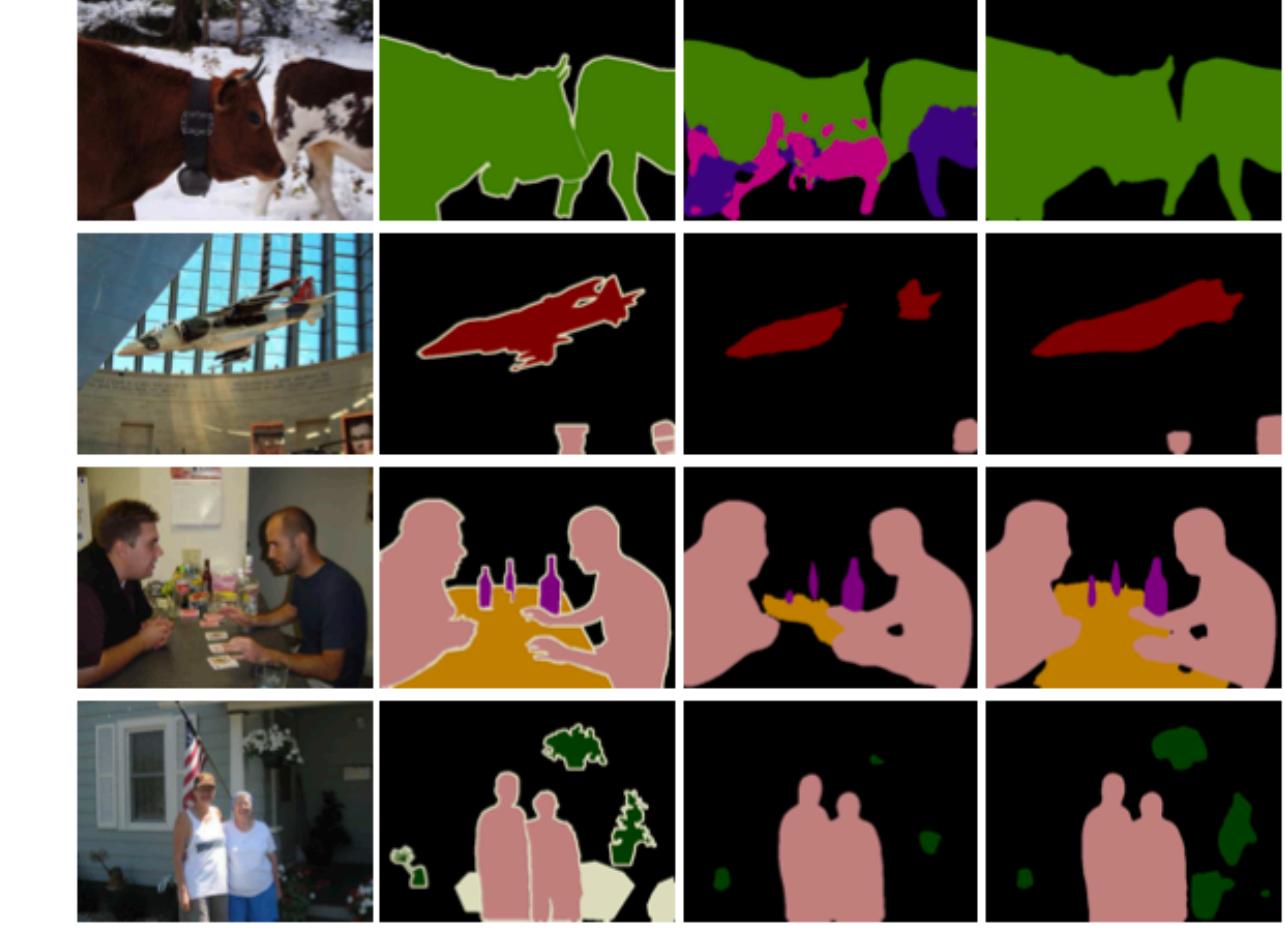


Cityscapes dataset



Loss Weight α	Mean IoU(%)	Pixel Acc.(%)
ResNet50 (without AL)	35.82	77.07
ResNet50 (with $\alpha = 0.3$)	37.01	77.87
ResNet50 (with $\alpha = 0.4$)	37.23	78.01
ResNet50 (with $\alpha = 0.6$)	37.09	77.84
ResNet50 (with $\alpha = 0.9$)	36.99	77.87

PASCAL VOC 2012 data



Method	Mean IoU(%)	Pixel Acc.(%)
ResNet50-Baseline	37.23	78.01
ResNet50+B1+MAX	39.94	79.46
ResNet50+B1+AVE	40.07	79.52
ResNet50+B1236+MAX	40.18	79.45
ResNet50+B1236+AVE	41.07	79.97
ResNet50+B1236+MAX+DR	40.87	79.61
ResNet50+B1236+AVE+DR	41.68	80.04

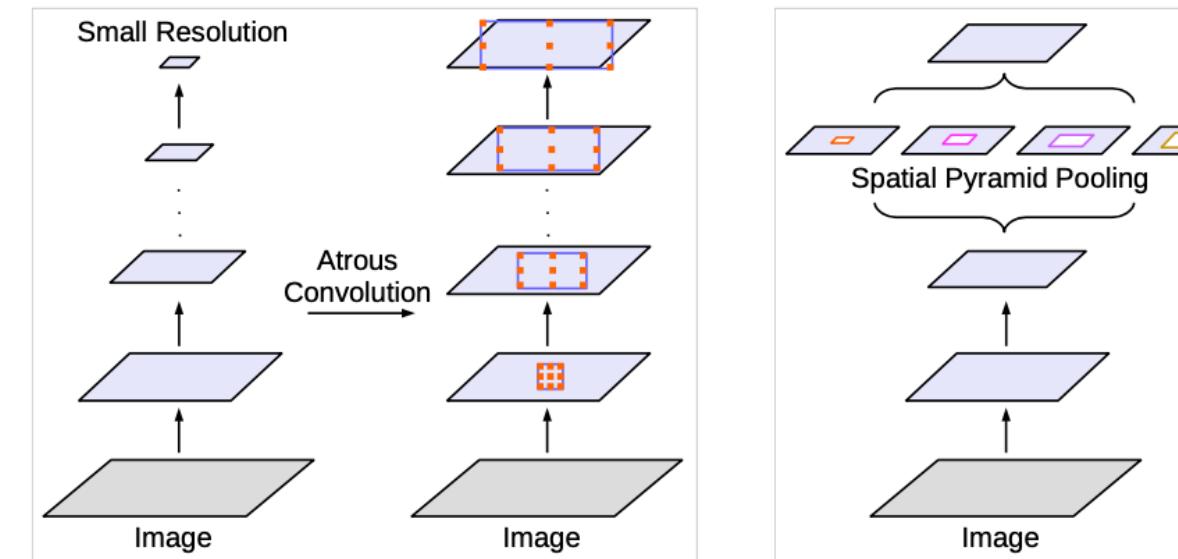
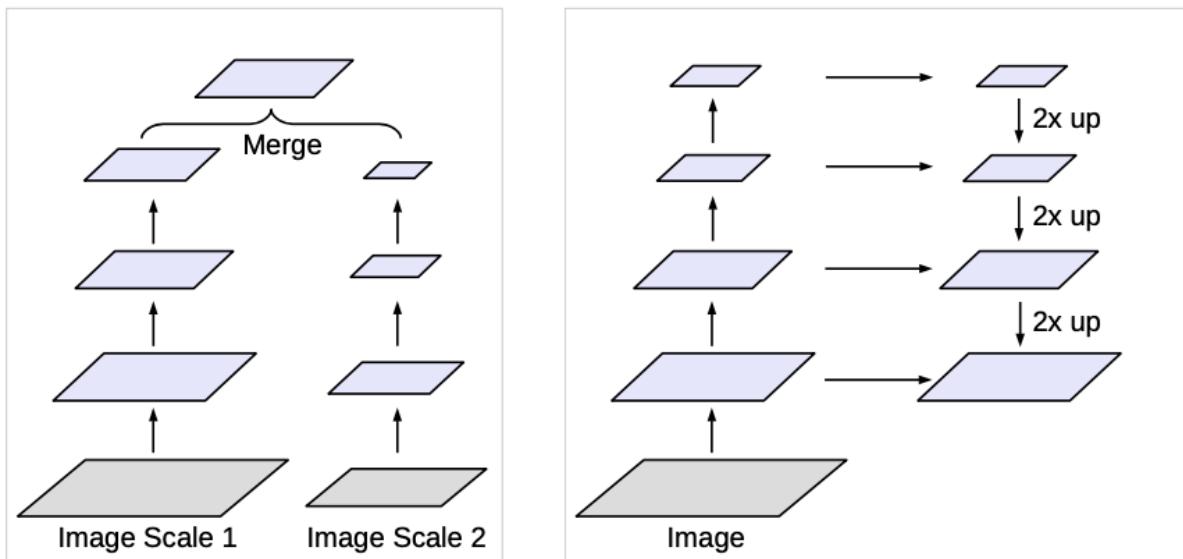


Boulder

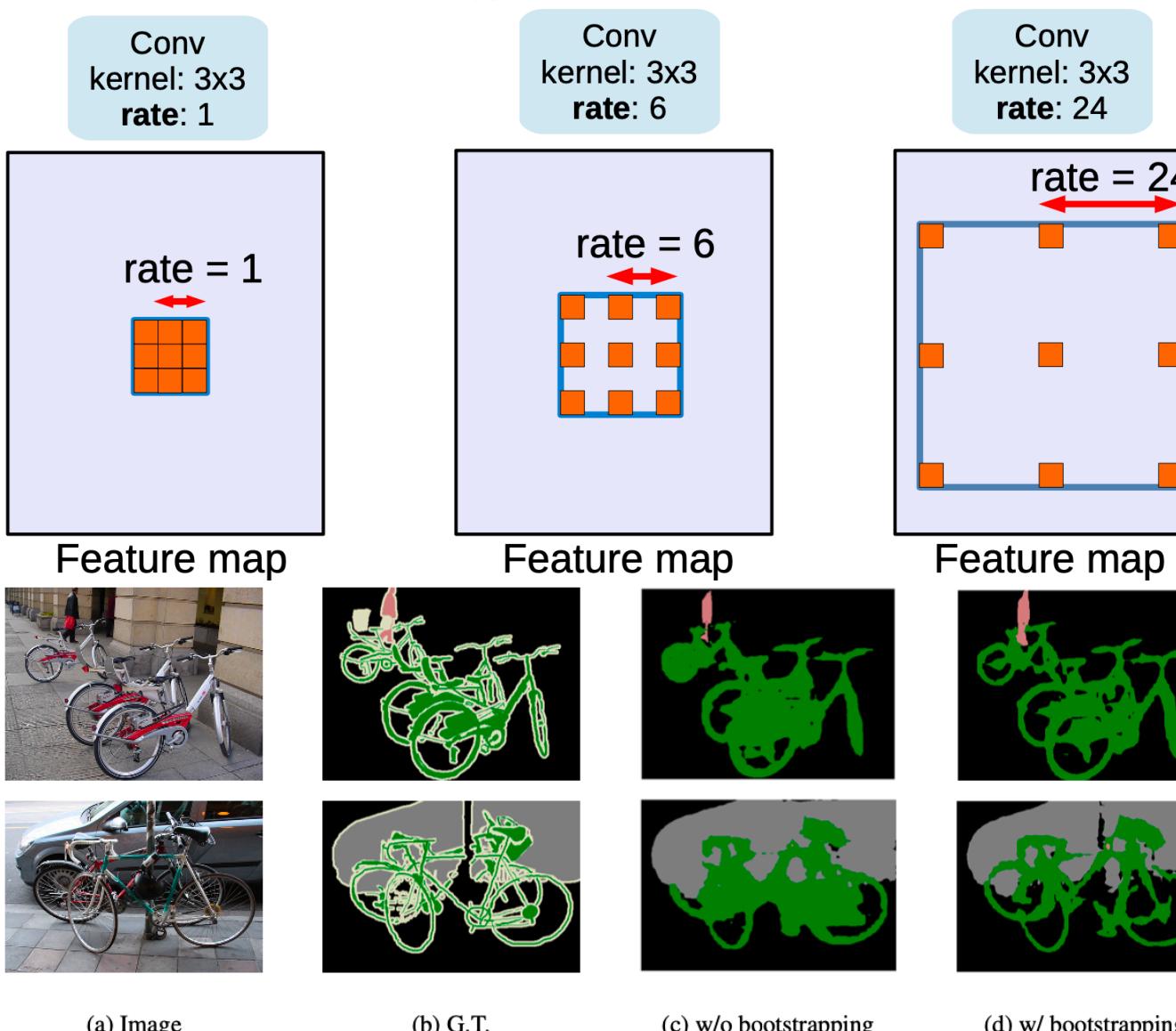
Rethinking Atrous Convolution for Semantic Image Segmentation

Two challenges in applying Deep Convolutional Neural Networks (DCNNs):

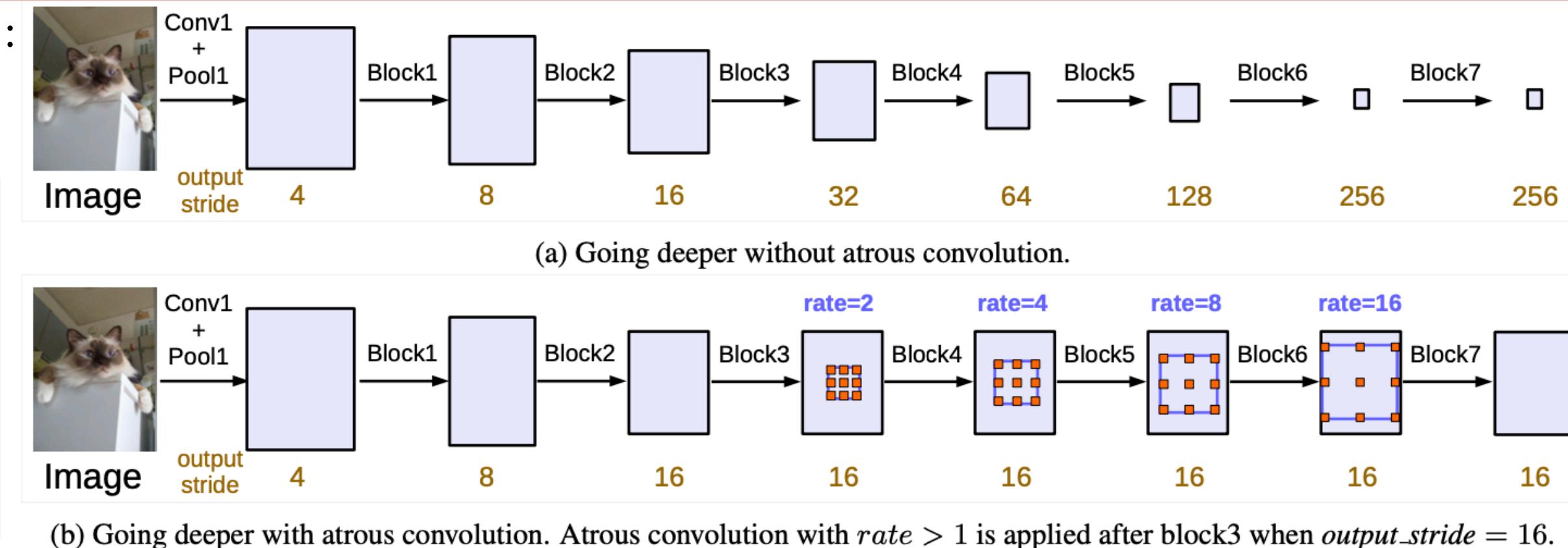
- reduced feature resolution
- objects at multiple scales



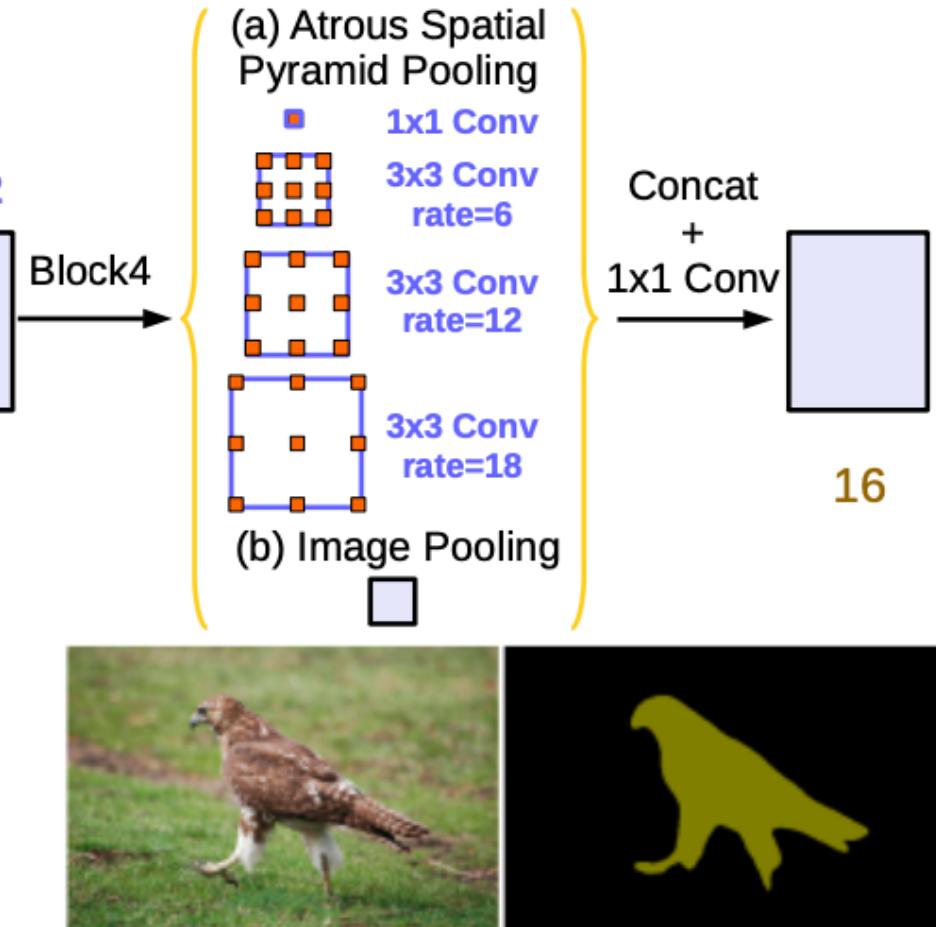
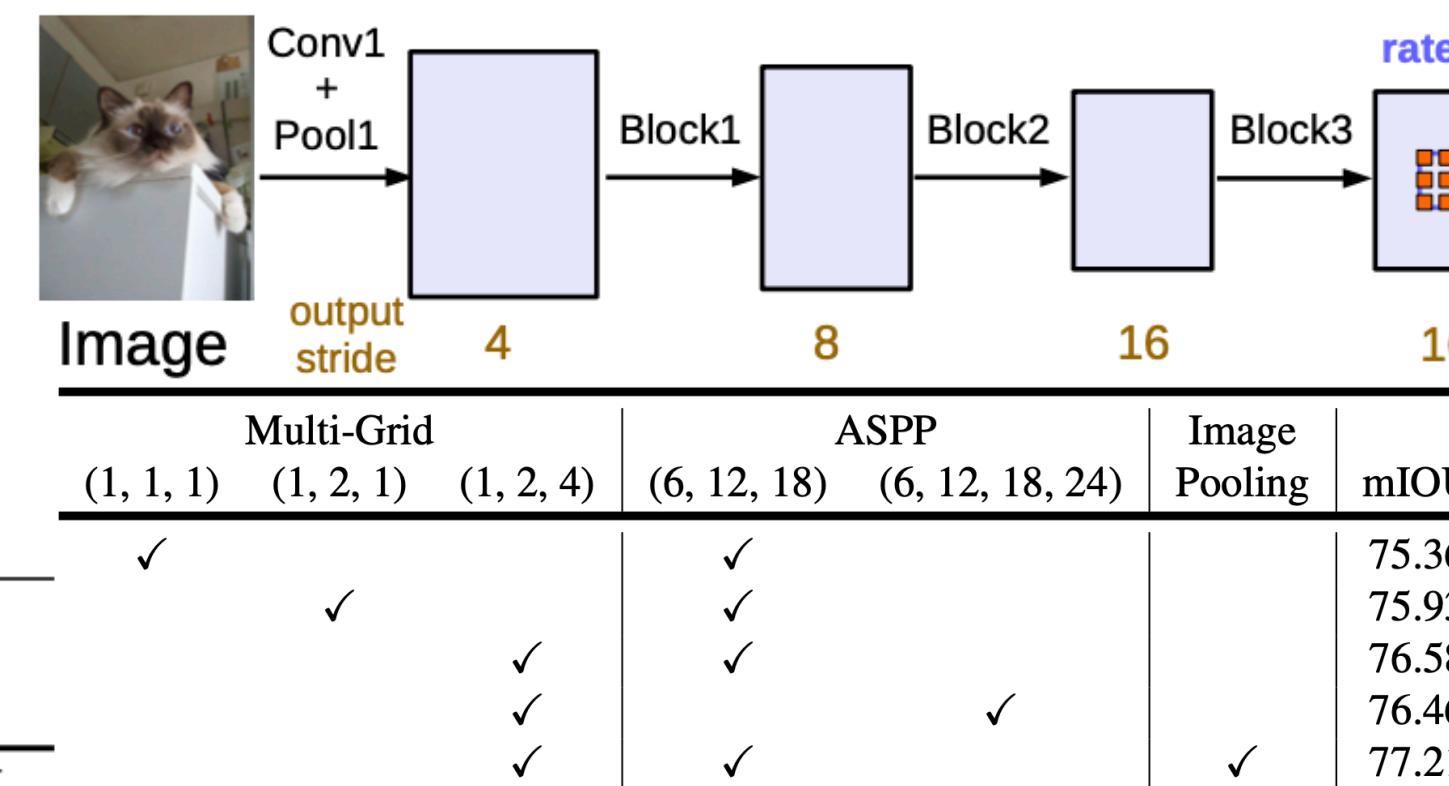
$$y[\mathbf{i}] = \sum_{\mathbf{k}} \mathbf{x}[\mathbf{i} + r \cdot \mathbf{k}] \mathbf{w}[\mathbf{k}]$$



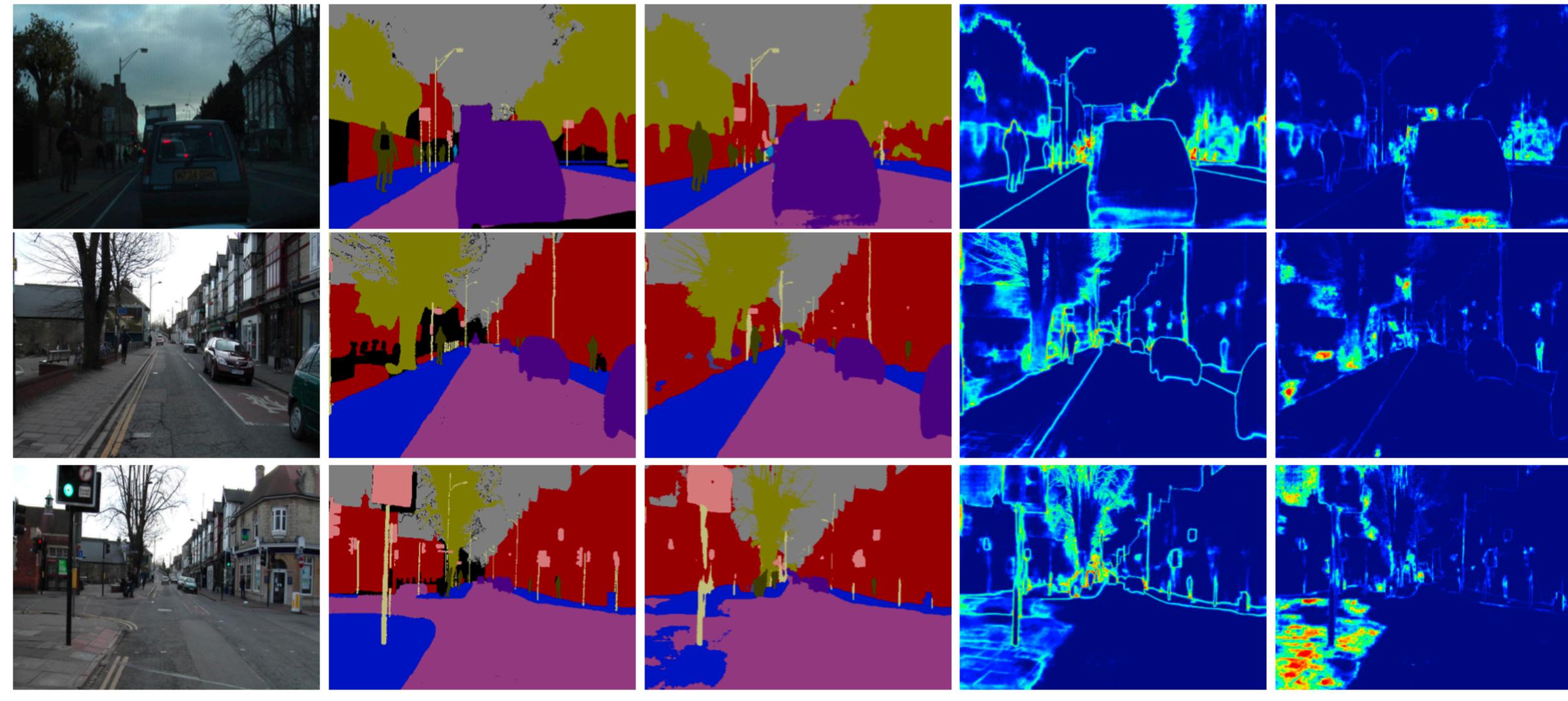
Bootstrapping on hard images improves segmentation



Method	mIOU	output_stride	8	16	32	64	128	256	
mIOU	75.18	73.88	70.06	59.99	42.34	20.29			
Adelaide_VeryDeep_FCN_VOC [85]	79.1								
LRR_4x_ResNet-CRF [25]	79.3								
DeepLabv2-CRF [11]	79.7								
CentraleSupelec Deep G-CRF [8]	80.2								
HikSeg_COCO [80]	81.4								
SegModel [75]	81.8								
Deep Layer Cascade (LC) [52]	82.7								
TuSimple [84]	83.1								
Large_Kernel_Matters [68]	83.6								
Multipath-RefineNet [54]	84.2								
ResNet-38_MS_COCO [86]	84.9								
PSPNet [95]	85.4								
IDW-CNN [83]	86.3								
CASIA-IVA-SDN [23]	86.6								
DIS [61]	86.8								
DeepLabv3	85.7	Multi-Grid	(1, 1, 1)	(1, 2, 1)	(1, 2, 4)	ASPP	(6, 12, 18)	Image Pooling	mIOU
DeepLabv3-JFT	86.9	✓	✓	✓	✓	✓	✓	✓	77.21



What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?



(a) Input Image (b) Ground Truth (c) Semantic Segmentation (d) Aleatoric Uncertainty (e) Epistemic Uncertainty

Aleatoric Uncertainty: noise inherent in the observations

- homoscedastic: constant for different inputs
- heteroscedastic: depends on the inputs to the model

Epistemic (Model) Uncertainty: can be explained away given enough data

Epistemic Uncertainty in Bayesian Deep Learning

$\mathbf{W} \sim \mathcal{N}(0, I)$ → prior distribution over the weights of neural network

$\mathbf{f}^{\mathbf{W}}(\mathbf{x})$ → random output of a Bayesian Neural Network

$p(\mathbf{y}|\mathbf{f}^{\mathbf{W}}(\mathbf{x}))$ → model likelihood

$\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}, \mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$ → dataset

$p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$ → posterior over the weights (Bayesian inference)

$p(\mathbf{W}|\mathbf{X}, \mathbf{Y}) = p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{W})/p(\mathbf{Y}|\mathbf{X})$

$p(\mathbf{Y}|\mathbf{X}) \rightarrow$ marginal probability (cannot be evaluated analytically)

$q_{\theta}^{*}(\mathbf{W}) \rightarrow$ a simple distribution approximating the posterior

$$\mathcal{L}(\theta, p) = -\frac{1}{N} \sum_{i=1}^N \log p(\mathbf{y}_i | \mathbf{f}^{\widehat{\mathbf{W}}_i}(\mathbf{x}_i)) + \frac{1-p}{2N} \|\theta\|^2$$

$\widehat{\mathbf{W}}_i \sim q_{\theta}^{*}(\mathbf{W}) \rightarrow$ dropout distribution

$p \rightarrow$ dropout probability

$\theta \rightarrow$ parameters of the simple distribution (weight matrices)

Heteroscedastic Aleatoric Uncertainty (Regression)

$$-\log p(\mathbf{y}_i | \mathbf{f}^{\widehat{\mathbf{W}}_i}(\mathbf{x}_i)) \propto \frac{1}{2\widehat{\sigma}_i^2} \|\mathbf{y}_i - \widehat{\mathbf{y}}_i\|^2 + \frac{1}{2} \log \widehat{\sigma}_i^2$$

$$[\widehat{\mathbf{y}}_i, \widehat{\sigma}_i^2] = \mathbf{f}^{\widehat{\mathbf{W}}_i}(\mathbf{x}_i) \quad \underbrace{\hspace{1cm}}_{\text{learned loss attenuation}}$$

$$\frac{1}{T} \sum_{t=1}^T \widehat{\mathbf{y}}_t^2 - \left(\underbrace{\frac{1}{T} \sum_{t=1}^T \widehat{\mathbf{y}}_t}_{\text{predictive mean}} \right)^2 + \frac{1}{T} \sum_{t=1}^T \widehat{\sigma}_t^2 \rightarrow \text{predictive variance}$$

CamVid dataset for road scene segmentation

DenseNet (Our Implementation)	67.1
+ Aleatoric Uncertainty	67.4
+ Epistemic Uncertainty	67.2
+ Aleatoric & Epistemic	67.5

IoU

$$\widehat{\mathbf{y}}_t, \widehat{\sigma}_t^2 = \mathbf{f}^{\widehat{\mathbf{W}}_t}(\mathbf{x})$$

Heteroscedastic Aleatoric Uncertainty (Classification)

$$p(\mathbf{y}_i | \mathbf{f}^{\widehat{\mathbf{W}}_i}(\mathbf{x}_i)) = \mathbf{y}_i^T \text{softmax}(\widehat{\mathbf{y}}_i + \widehat{\sigma}_i \epsilon_i), \epsilon_i \sim \mathcal{N}(0, I)$$

$$\mathbf{p} = \frac{1}{T} \sum_{t=1}^T \text{softmax}(\widehat{\mathbf{y}}_t + \widehat{\sigma}_t \epsilon_t), \epsilon_t \sim \mathcal{N}(0, I)$$

$$H(\mathbf{p}) = - \sum_{c=1}^C p_c \log p_c \rightarrow \text{uncertainty of probability vector } \mathbf{p}$$

Each datapoint and each pixel will have its own prediction and uncertainty!

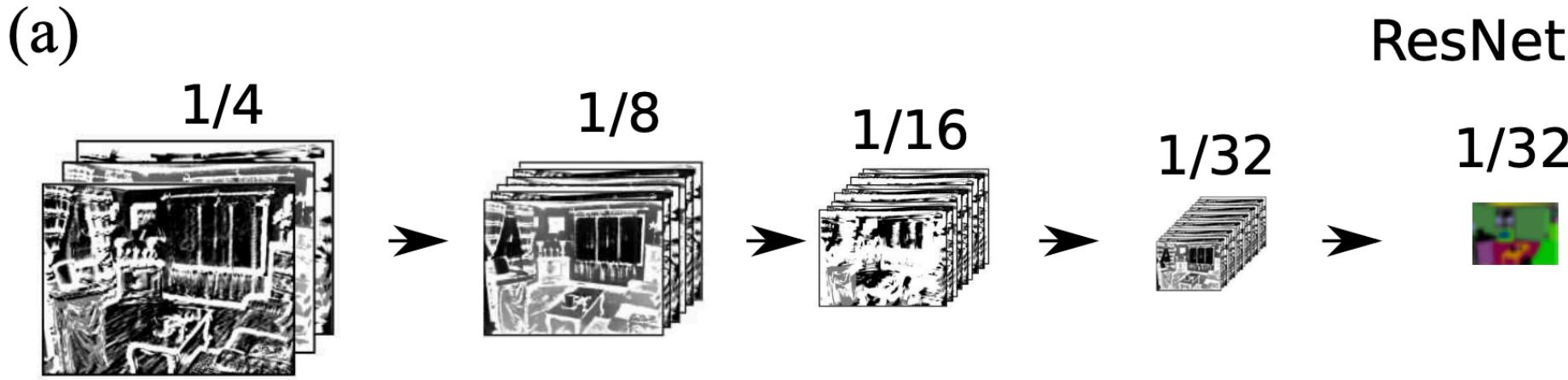


Boulder

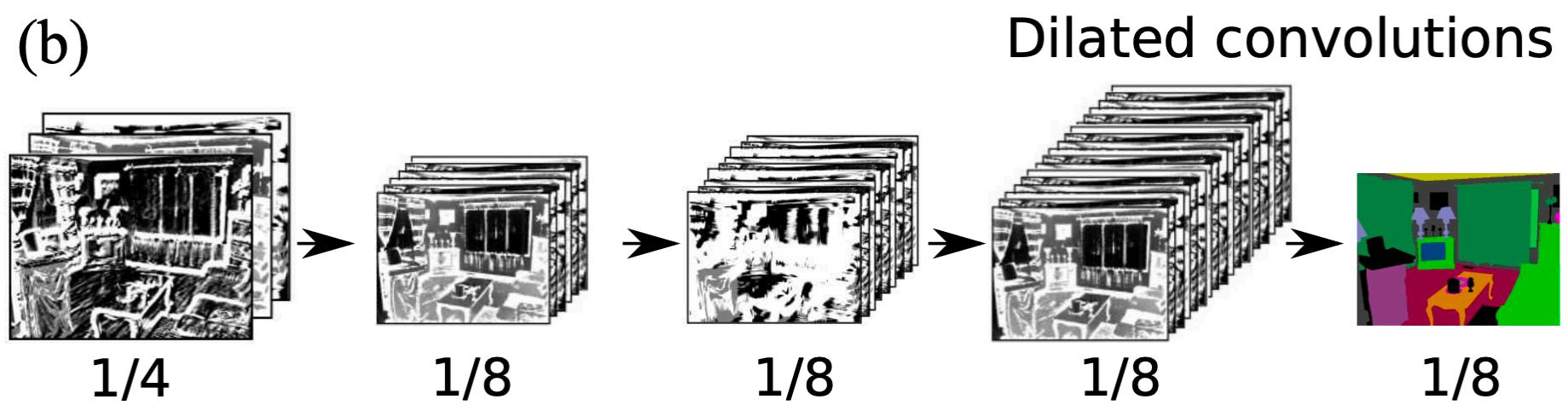
RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation



object parsing (left) and semantic segmentation (right)

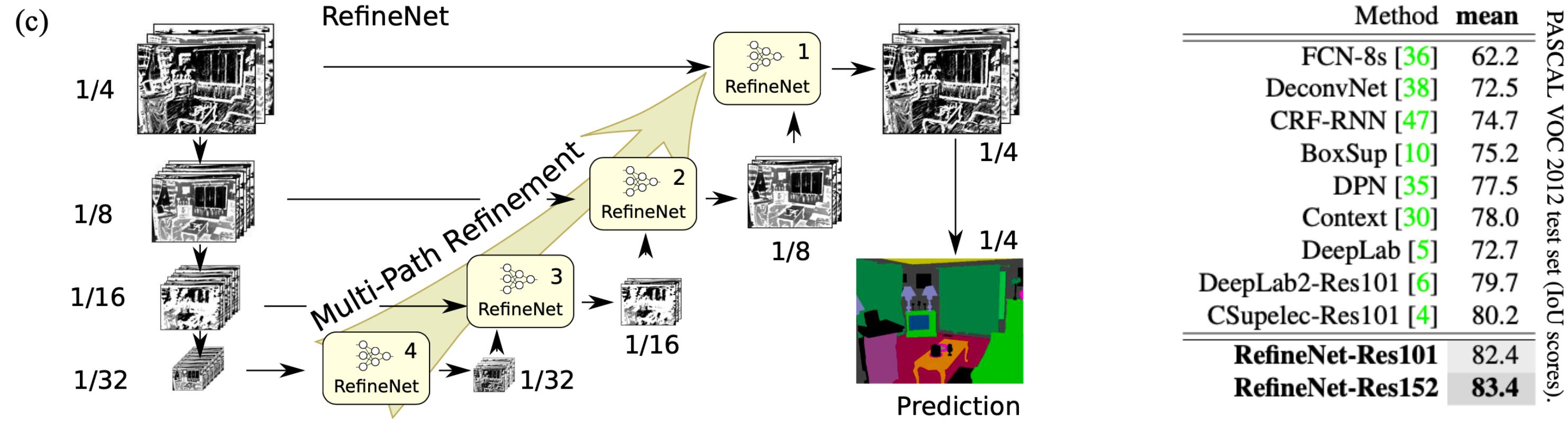


suffers from downscaling of the feature maps



computationally expensive to train and quickly reaches memory limits

Effectively combine high-level semantics and low-level features to produce high-resolution segmentation maps.



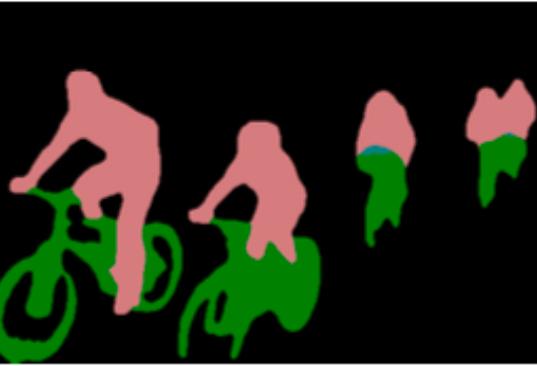
Method	mean
FCN-8s [36]	62.2
DeconvNet [38]	72.5
CRF-RNN [47]	74.7
BoxSup [10]	75.2
DPN [35]	77.5
Context [30]	78.0
DeepLab [5]	72.7
DeepLab2-Res101 [6]	79.7
CSupelec-Res101 [4]	80.2
RefineNet-Res101	82.4
RefineNet-Res152	83.4



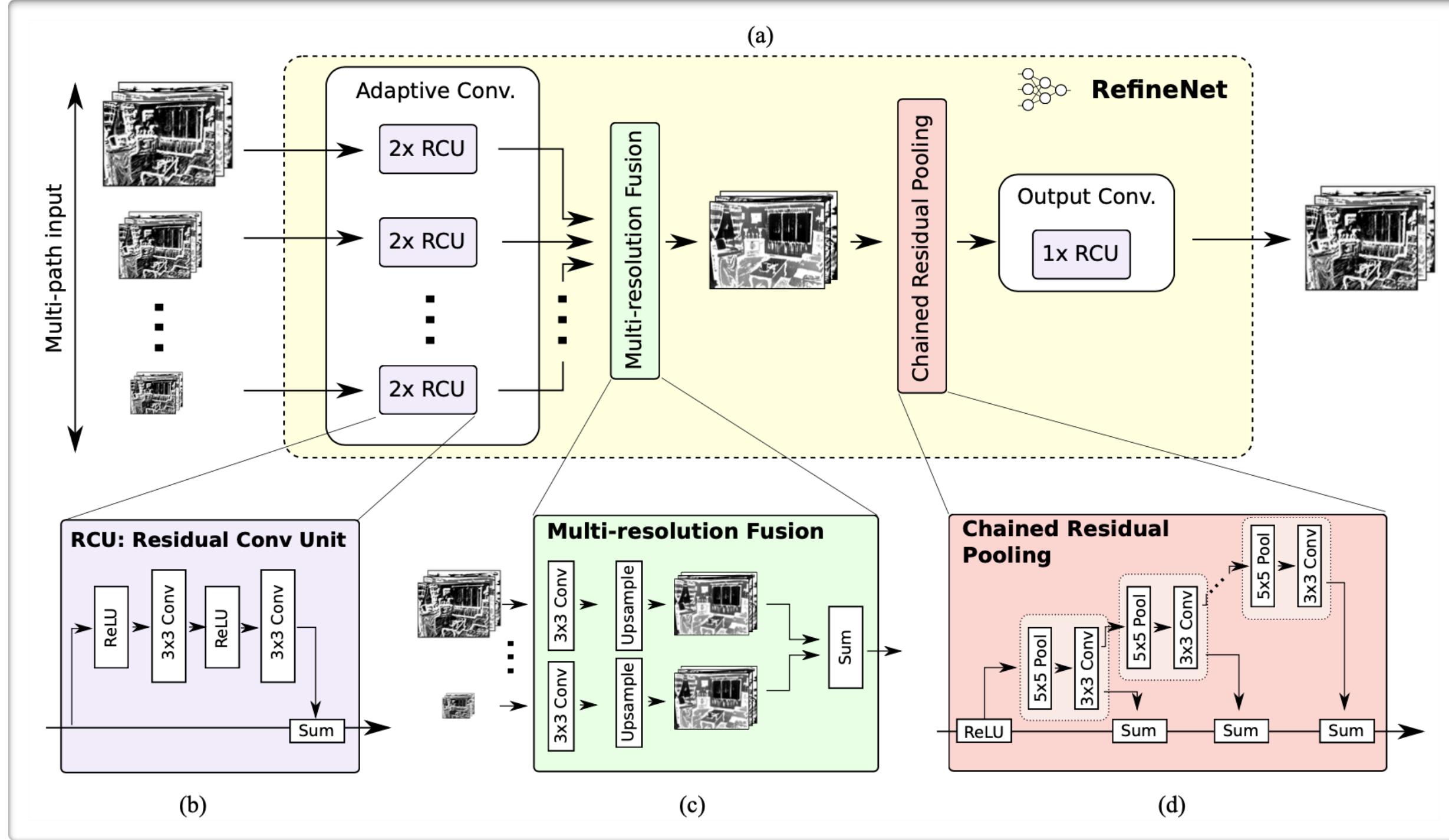
(a) Test Image



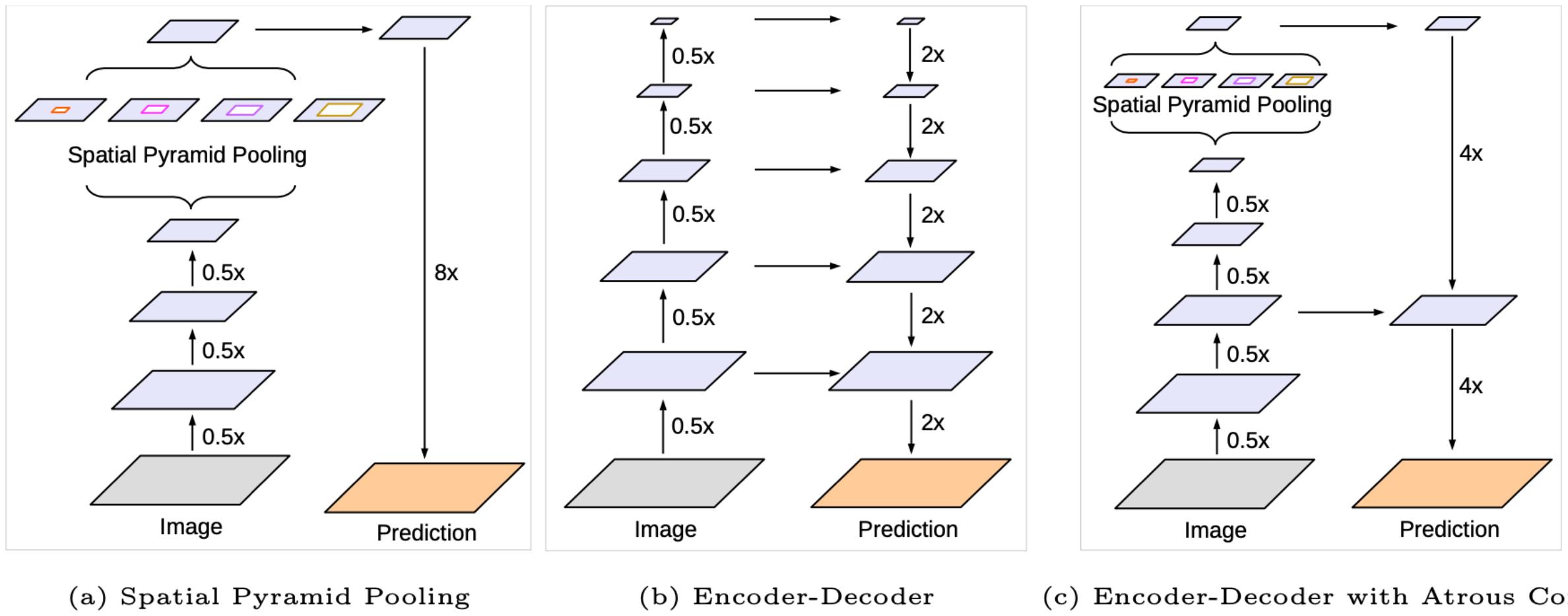
(b) Ground Truth



(c) Prediction



Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation

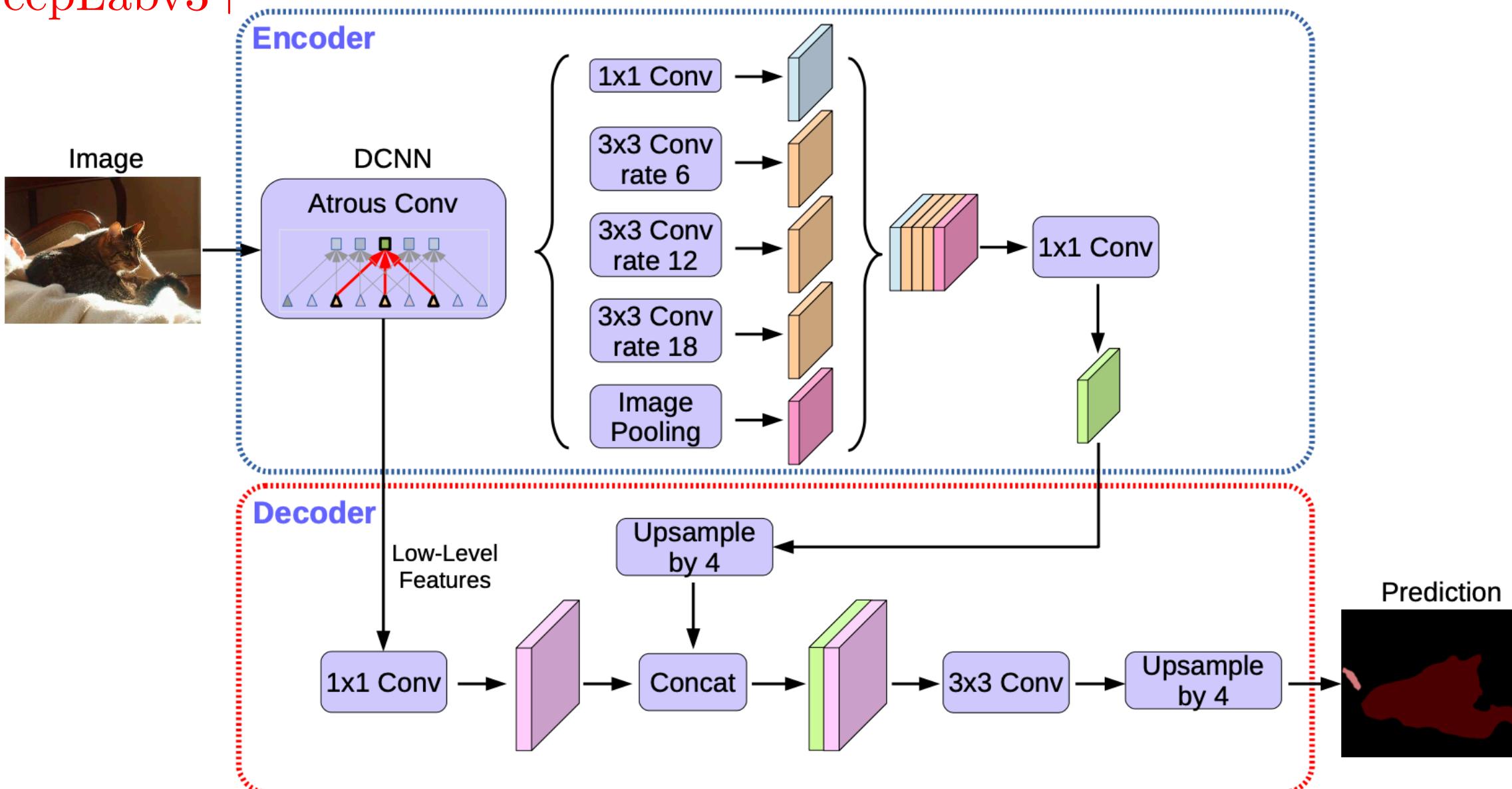

[YouTube Video](#)


(a) Spatial Pyramid Pooling

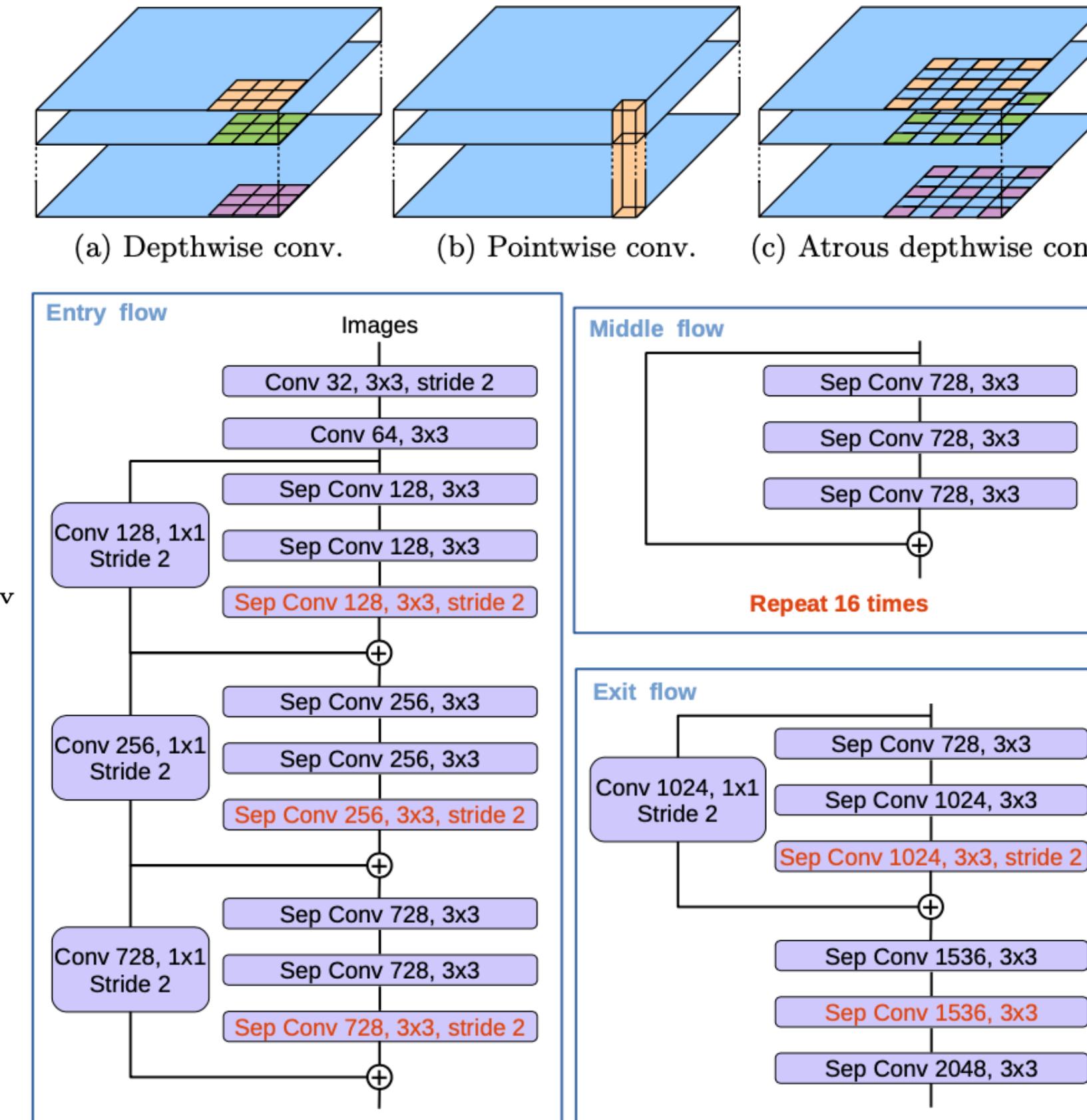
(b) Encoder-Decoder

(c) Encoder-Decoder with Atrous Conv

DeepLabv3+



Chen, Liang-Chieh, et al. "Encoder-decoder with atrous separable convolution for semantic image segmentation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

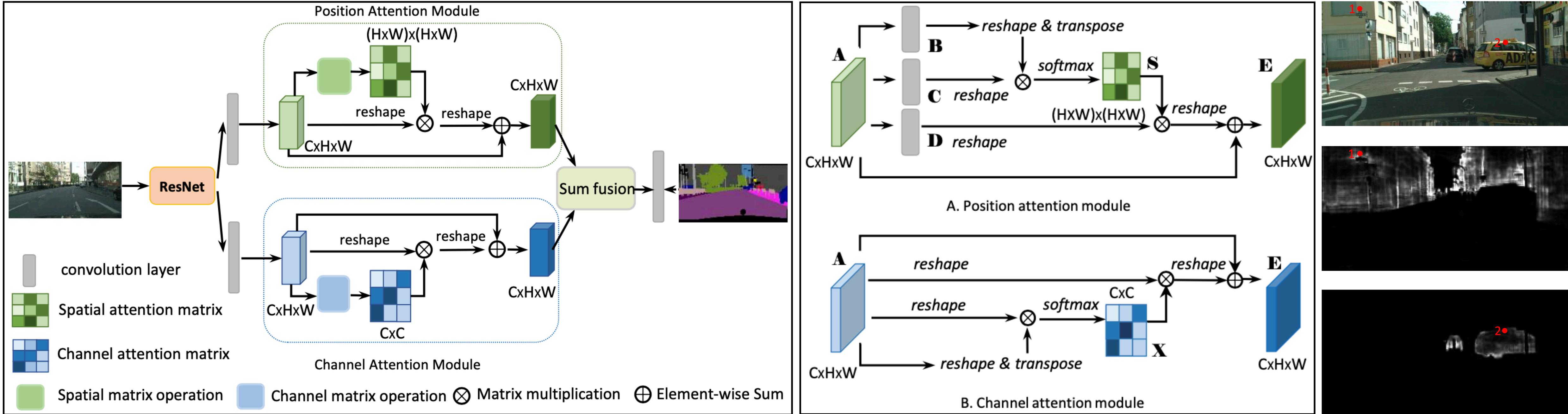


PASCAL VOC 2012 test set	
Method	mIOU
Deep Layer Cascade (LC) [82]	82.7
TuSimple [77]	83.1
Large_Kernel_Matters [60]	83.6
Multipath-RefineNet [58]	84.2
ResNet-38_MS_COCO [83]	84.9
PSPNet [24]	85.4
IDW-CNN [84]	86.3
CASIA_IVA_SDN [63]	86.6
DIS [85]	86.8
DeepLabv3 [23]	85.7
DeepLabv3-JFT [23]	86.9
DeepLabv3+ (Xception)	87.8
DeepLabv3+ (Xception-JFT)	89.0



Boulder

Dual Attention Network for Scene Segmentation



Dual Attention Network (DANet)
– Cityscapes – PASCAL VOC2012
– PASCAL Context – COCO Stuff
Stuff: sky, road, grass, etc.

Objects: person, car, bicycle, etc.

Position Attention Module

$A \in \mathbb{R}^{C \times H \times W}$ → local feature

$B, C \in \mathbb{R}^{C \times H \times W}$ → after conv on A

$B, C \in \mathbb{R}^{C \times N}$ → reshape ($N = HW$)

$S \in \mathbb{R}^{N \times N}$ → spatial attention map

$$S = \text{softmax}(B^T C)$$

$$s_{ji} = \frac{\exp(B_i \cdot C_j)}{\sum_{i=1}^N \exp(B_i \cdot C_j)}$$

$$\sum_{i=1}^N s_{ji} = 1$$

$$D \in \mathbb{R}^{C \times H \times W} \rightarrow \text{after conv on } A$$

$$D \in \mathbb{R}^{C \times N} \rightarrow \text{reshape}$$

$$DS^T \in \mathbb{R}^{C \times H \times W} \rightarrow \text{after reshape}$$

$$E \in \mathbb{R}^{C \times H \times W}$$

$$E_j = \alpha \sum_{i=1}^N s_{ji} D_i + A_j$$

Channel Attention Module

$A \in \mathbb{R}^{C \times H \times W} \rightarrow \text{local feature}$

$A \in \mathbb{R}^{C \times N} \rightarrow \text{reshape}$

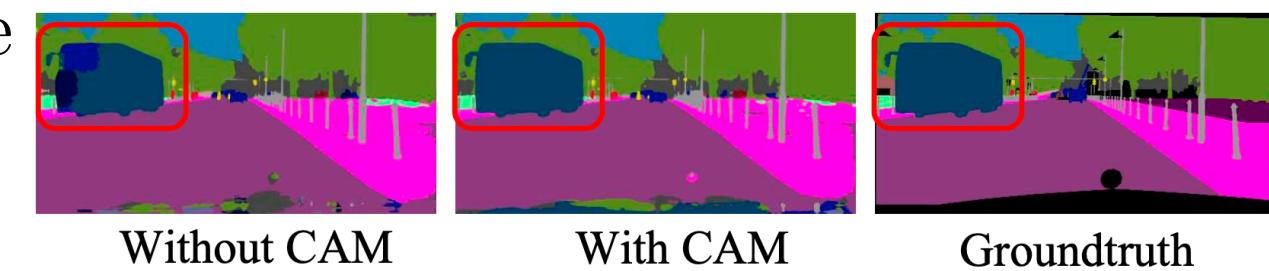
$X = \text{softmax}(AA^T) \in \mathbb{R}^{C \times C}$

$$x_{ji} = \frac{\exp(A_i \cdot A_j)}{\sum_{i=1}^C \exp(A_i \cdot A_j)}$$

$$X^T A \in \mathbb{R}^{C \times H \times W} \rightarrow \text{after reshape}$$

$$E_j = \beta \sum_{i=1}^C x_{ji} A_i + A_j$$

Method	BaseNet	PAM	CAM	Mean IoU%
Dilated FCN	Res50			70.03
DANet	Res50	✓		75.74
DANet	Res50		✓	74.28
DANet	✓	✓		76.34
Dilated FCN	Res101			72.54
DANet	Res101	✓		77.03
DANet	Res101		✓	76.55
DANet	Res101	✓	✓	77.57





Boulder



Questions?

[YouTube Playlist](#)
