Deep Learning for Semantic Image Segmentation
-----Deeplabs

Jianping Fan
Dept of Computer Science
UNC-Charlotte

Course Website:
http://webpages.uncc.edu/jfan/itcs5152.html

Deep Learning for Semantic Image Segmentation

- **Deeplab v1, v2, v3**

- **U-nets**

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in ICLR, 2015.

Deeplab series

Topics

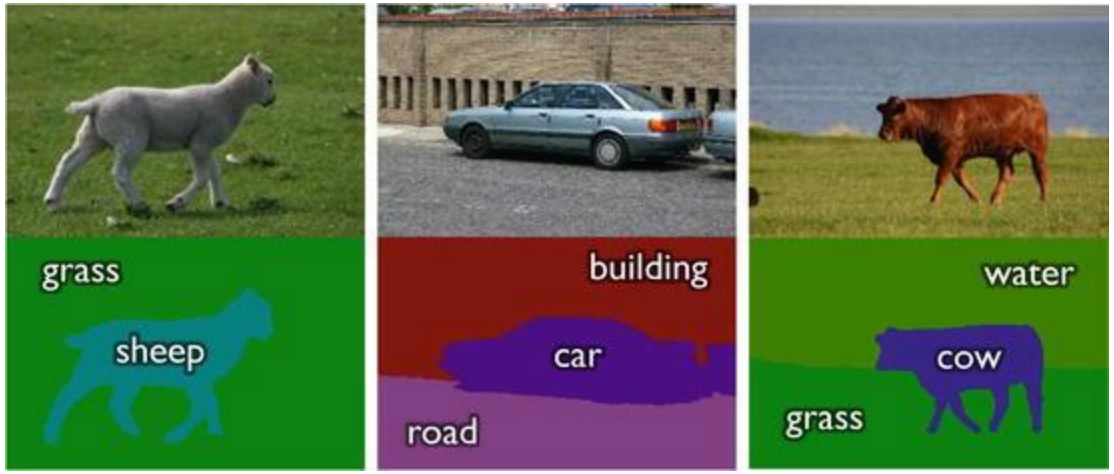
- Introduction
  - Semantic Segmentation
  - DCNN for segmentation
- ‘Holes’ algorithm
- Boundary recovery
  - Probabilistic Graphical Models
  - Fully Connected CRFs

Slides credit to Topaz Gilad, 2016

Introduction

What is semantic image segmentation?

- **Partitioning** an image into **regions** of **meaningful** objects.
- Assign an object **category label**.

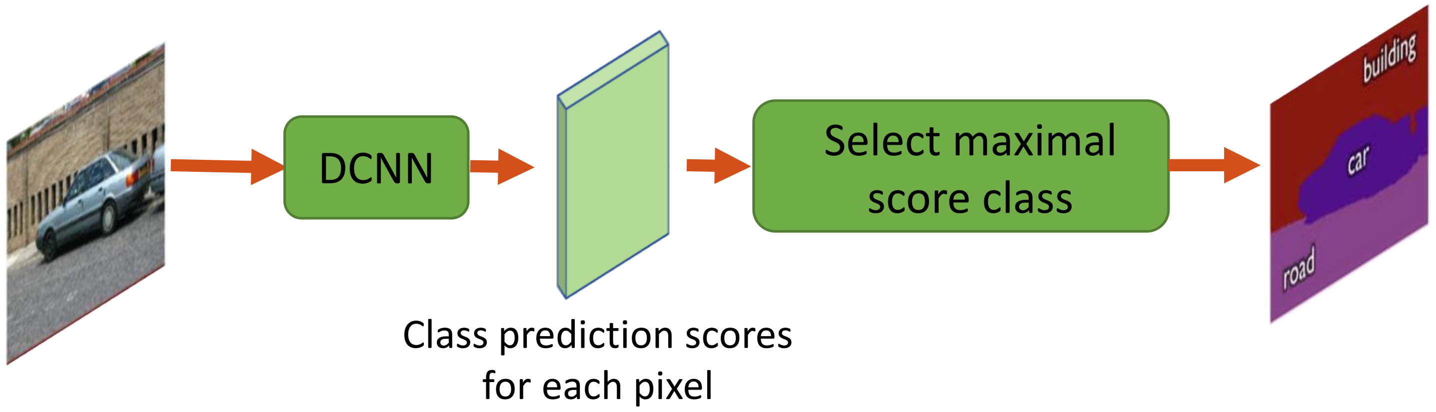


Jamie Shotton and Pushmeet Kohli, [Semantic Image Segmentation](), Computer Vision, pp 713-716, Springer, 2016.

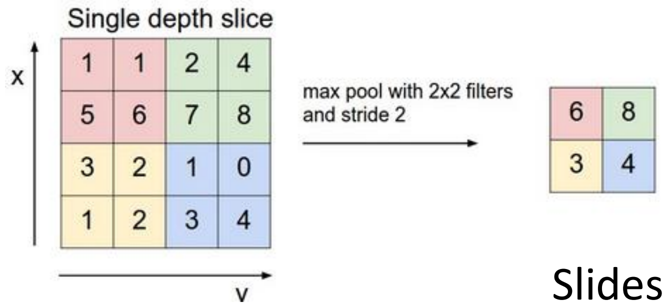
Slides credit to Topaz Gilad, 2016

Introduction

DCNN and image segmentation



- **What happens in each standard DCNN layer?**
  - ➢ Striding
  - ➢ Pooling



Slides credit to Topaz Gilad, 2016

Introduction

DCNN and image segmentation

Pooling advantages:

- ✓ Invariance to small translations of the input.
- ✓ Helps avoid overfitting.
- ✓ Computational efficiency.

Striding advantages:

- ✓ Fewer applications of the filter.
- ✓ Smaller output size.

Slides credit to Topaz Gilad, 2016

Introduction

DCNN and image segmentation

What are the disadvantages for semantic segmentation?

- x **Down-sampling** causes loss of information.
- x The input invariance harms the pixel-perfect accuracy.

DeepLab address those issues by:

- **Atrous** convolution ('Holes' algorithm).
- **CRFs** (Conditional Random Fields).

Slides credit to Topaz Gilad, 2016

Up-Sampling

Addressing the reduced resolution problem

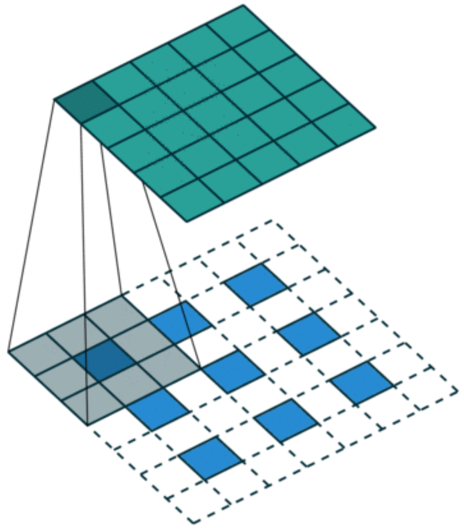
Possible solution:

‘deconvolutional’ layers (backwards convolution).

- x Additional memory and computational time.
- x Learning additional parameters.

Suggested solution:

Atrous (‘Holes’) convolution



Slides credit to Topaz Gilad, 2016

Atrous ('Holes') Algorithm

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

Atrous convolution for 1-D signal:

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

Introduce zeros between filter values

$x[i]$ 1-D input signal

$w[k]$ filter of length K

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

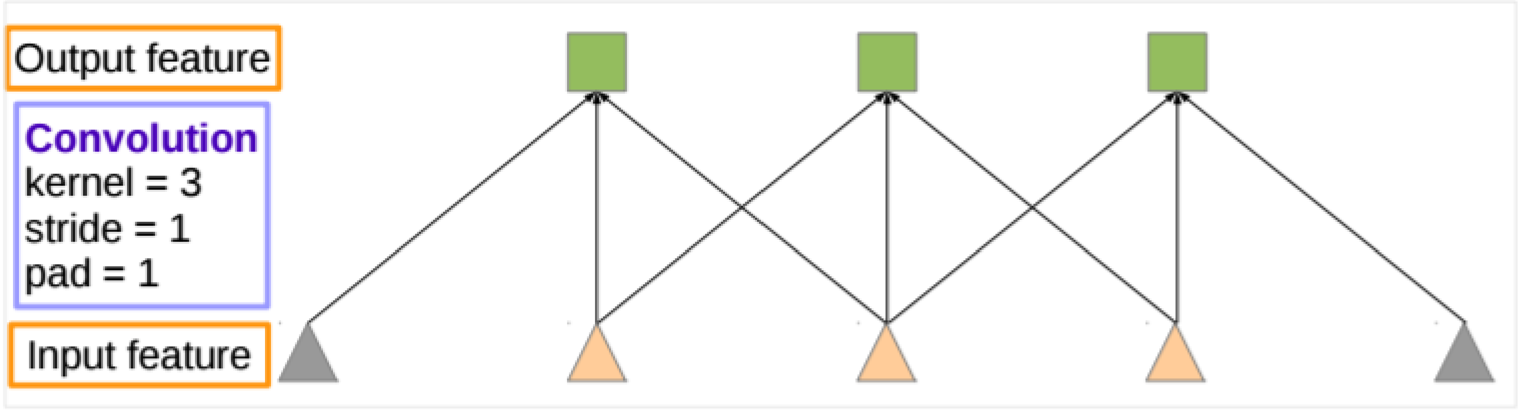
$y[i]$ output of atrous convolution.

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

Chen, Liang-Chieh, et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *arXiv preprint arXiv:1606.00915* (2016).

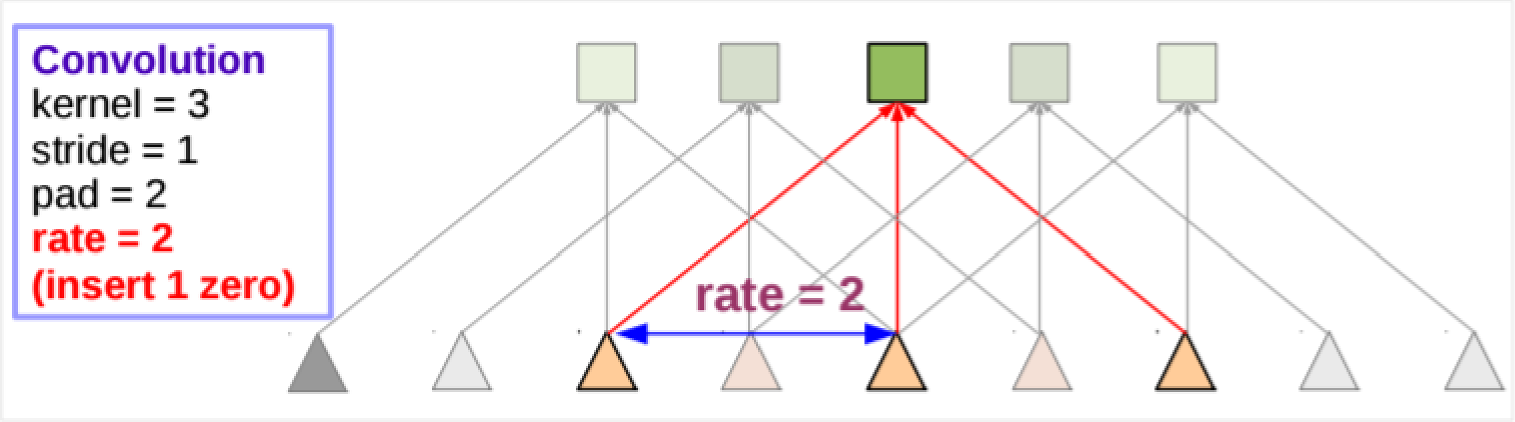
Atrous ('Holes') Algorithm

Standard convolution



(a) Sparse feature extraction

Atrous convolution



(b) Dense feature extraction

Chen, Liang-Chieh, et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *arXiv preprint arXiv:1606.00915* (2016).

Atrous ('Holes') Algorithm

Filters field-of-view

- **Small** field-of-view → accurate **localization**
- **Large** field-of-view → **context** assimilation
- 'Holes': Introduce zeros between filter values.
- **Effective filter size increases** (enlarge the **field-of-view** of filter):

$$k \times k \text{ filter to } k_e = k + (k-1)(r-1)$$

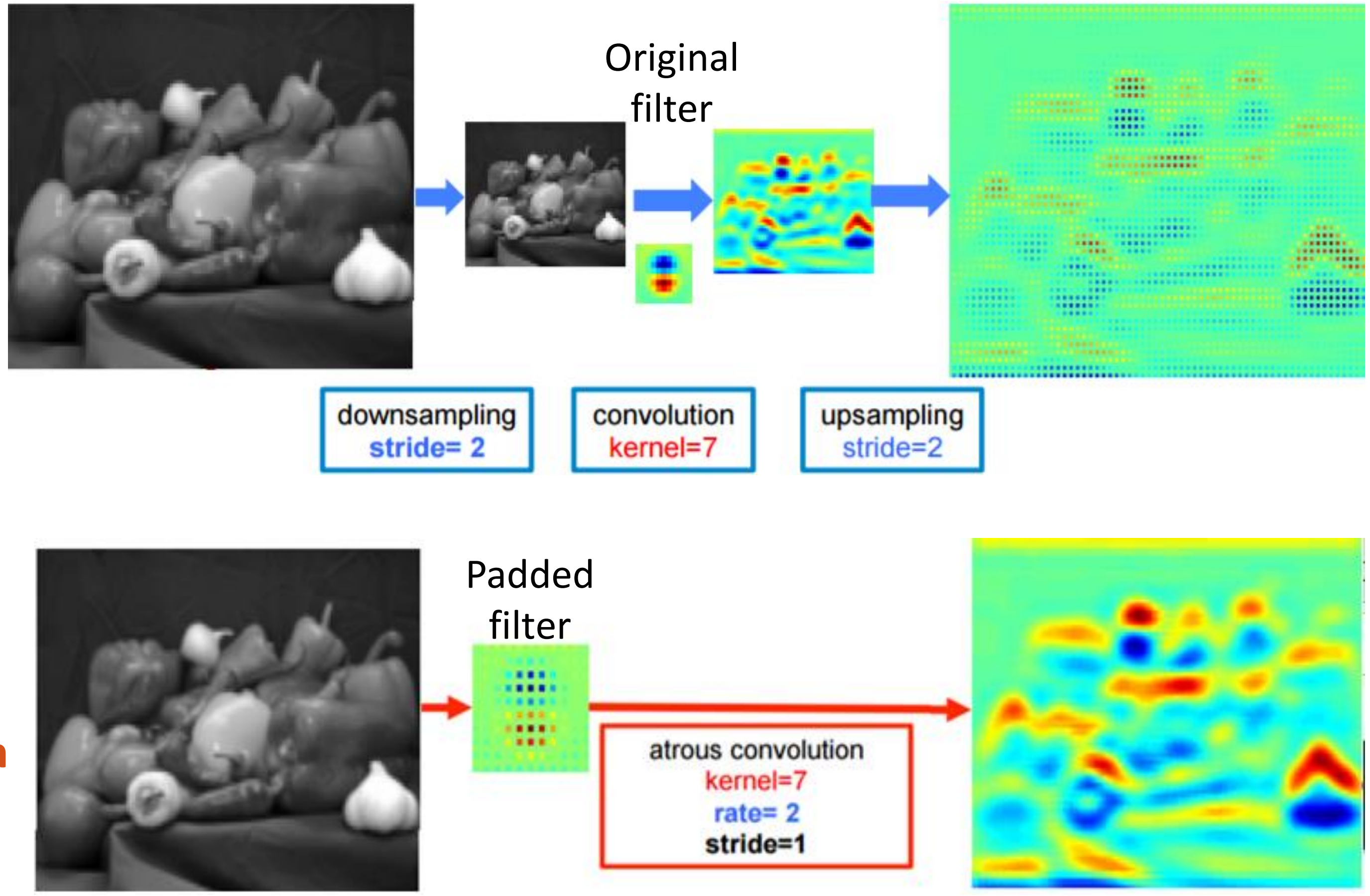
- However, we take into account **only** the **non-zero** filter values:
  - ✓ Number of filter parameters is the same.
  - ✓ Number of operations per position is the same.

Chen, Liang-Chieh, et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *arXiv preprint arXiv:1606.00915* (2016).

Atrous ('Holes') Algorithm

Standard convolution

Atrous convolution



Chen, Liang-Chieh, et al. "[DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs]." *arXiv preprint arXiv:1606.00915* (2016).

Boundary recovery

- DCNN trade-off:
  Classification accuracy ↔ Localization accuracy

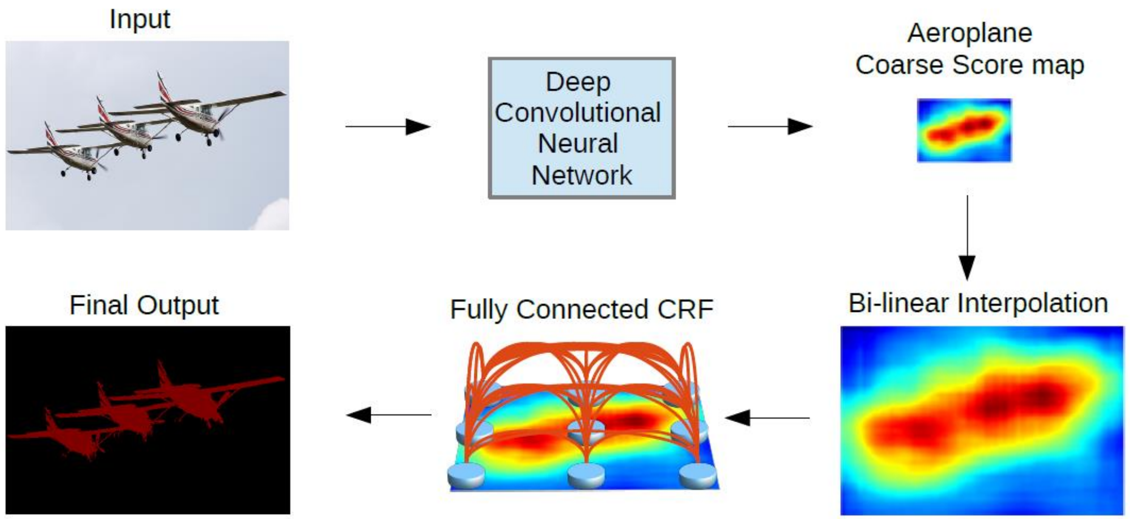
✓ DCNN score maps **successfully** predict **classification** and rough position.

x **Less effective** for **exact outline**.

Chen, Liang-Chieh, et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *arXiv preprint arXiv:1606.00915* (2016).

Boundary recovery

- **Possible solution:** super-pixel representation.
- **Suggested Solution:** fully connected CRFs.



L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in ICLR, 2015.
https://www.researchgate.net/figure/225069465_fig1_Fig-1-Images-segmented-using-SLIC-into-superpixels-of-size-64-256-and-1024-pixels

Slides credit to Topaz Gilad, 2016

Conditional Random Fields

Problem statement

- $\mathbf{X}$ - Random field of input observations (images) of size N.
- $\mathbf{L} = \{l_1, \ldots, l_M\}$ - Set of labels.
- $\mathbf{Y}$ - Random field of pixel labels.
- X_j - color vector of pixel j.
- Y_j - label assigned to pixel j.

CRFs are usually used to model connections between different images.

Here we use them to **model connection between image pixels**!

P. Krahenbuhl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in NIPS, 2011.

Probabilistic Graphical Models

- ## Graphical Model

Factorization - a distribution over many variables represented as a product of local functions, each depends on a smaller subset of variables.

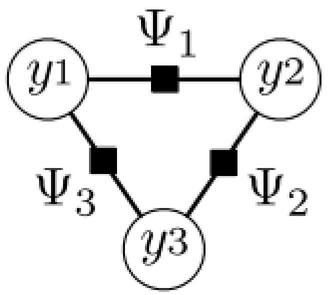
$$p(\mathbf{x},\mathbf{y}) = Z^{-1} \prod_{a \in F} \psi_a \left(x_{N(a)}, y_{N(a)} \right)$$

Probabilistic Graphical Models

- Undirected vs. Directed

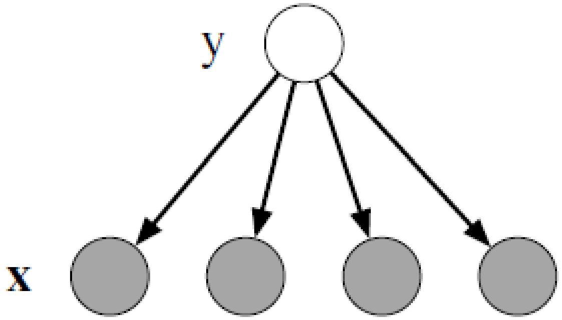
G(V, F, E)

Undirected



$$p\left(y_1,y_2,y_3\right)=\Psi_1\left(y_1,y_2\right)\Psi_1\left(y_2,y_3\right)\Psi_1\left(y_1,y_3\right)$$

Directed



$$p\left(y|\mathbf{x}\right)=p\left(y\right)\prod_{k=1}^{4}p\left(x_k|y\right)$$

C. Sutton and A. McCallum, "An introduction to Conditional Random Fields", Foundations and Trends in Machine Learning, vol. 4, No. 4 (2011) 267–373

Conditional Random Fields

Fully connected CRFs

Definition:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \prod_{a=1}^{A} \psi_a(\mathbf{Y}_a \mid \mathbf{X})$$

- Z(X) - is an input-dependent normalization factor.

Factorization (energy function):

$$E(\mathbf{y} \mid \mathbf{X}) = \sum_{i=1}^{N} \psi_i(y_i \mid \mathbf{X}) + \sum_{i \neq j} \psi_{i,j}(y_i, y_j \mid \mathbf{X})$$

- y - is the label assignment for pixels.

P. Krahenbuhl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in NIPS, 2011.

C. Sutton and A. McCallum, "An introduction to Conditional Random Fields", Foundations and Trends in Machine Learning, vol. 4, No. 4 (2011) 267–373

Conditional Random Fields

Potential functions in our case

$$\psi_i\left(y_i \mid \mathbf{X}\right) = -\log\left(p\left(y_i \mid \mathbf{X}\right)\right)$$

- $p(y_i|\boldsymbol{X})$ - is the label assignment probability for pixel *i* computed by DCNN.

$$\psi_{i,j}\left(y_i, y_j \mid \mathbf{X}\right) = \mathbf{1}_{y_i \neq y_j} \cdot \left[\underbrace{\theta_1 \exp\left(-\frac{\left\| s_i - s_j \right\|^2}{2\sigma_a^2} - \frac{\left\| x_i - x_j \right\|^2}{2\sigma_b^2} \right)}_{\text{‘bilateral’ kernel}} + \underbrace{\theta_2 \exp\left(-\frac{\left\| s_i - s_j \right\|^2}{2\sigma_\gamma^2} \right)}_{\text{smoothness kernel}} \right]$$

- s_i - position of pixel *i*.
- x_i - intensity (color) vector of pixel *i*.
- θ_1, θ_2 - learned parameters (weights).
- $\sigma_a^2, \sigma_b^2, \sigma_\gamma^2$ - hyper parameters (what is considered “near” / “similar”).

Chen, Liang-Chieh, et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *arXiv preprint arXiv:1606.00915* (2016).

Conditional Random Fields

Potential functions in our case

$$\psi_{i,j}\left(y_i, y_j \mid \mathbf{X}\right) = \mathbf{1}_{y_i \neq y_j} \cdot \left[\underbrace{\theta_1 \exp\left(-\frac{\left\| s_i - s_j \right\|^2}{2\sigma_a^2} - \frac{\left\| x_i - x_j \right\|^2}{2\sigma_b^2} \right)}_{\text{‘bilateral’ kernel}} + \underbrace{\theta_2 \exp\left(-\frac{\left\| s_i - s_j \right\|^2}{2\sigma_\gamma^2} \right)}_{\text{smoothness kernel}} \right]$$

Pixels “nearness”

Pixels color similarity

- **Bilateral kernel** – nearby pixels with similar color are likely to be in the same class.
- σ_a^2, σ_b^2 - what is considered “near” / “similar”).

Chen, Liang-Chieh, et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *arXiv preprint arXiv:1606.00915* (2016).

Conditional Random Fields

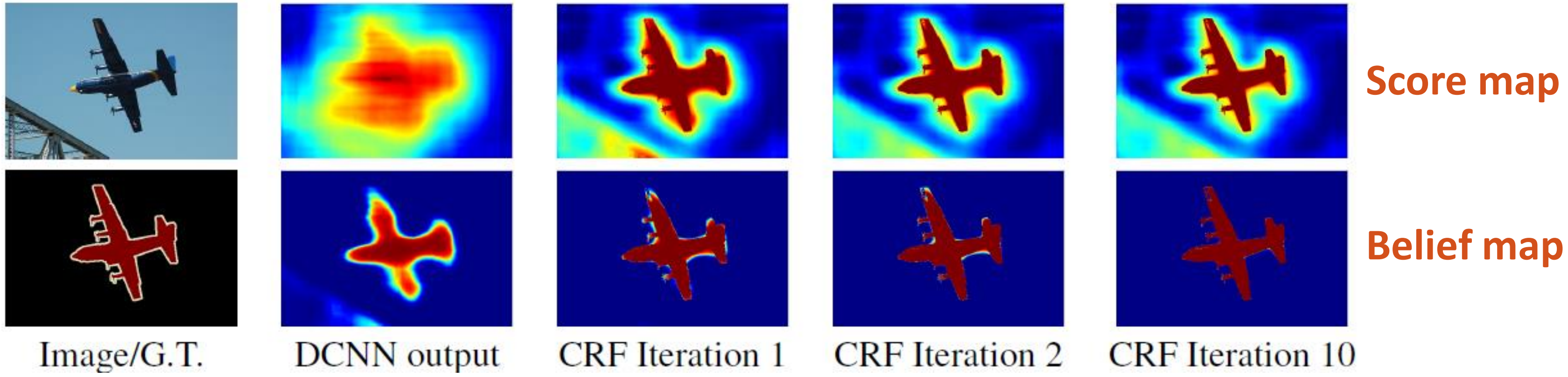
Potential functions in our case

$$\psi_{i,j}\left(y_i, y_j \mid \mathbf{X}\right) = \boxed{\mathbf{1}_{y_i \neq y_j}} \cdot \left[\underbrace{\theta_1 \exp\left(-\frac{\left\| s_i - s_j \right\|^2}{2\sigma_a^2} - \frac{\left\| x_i - x_j \right\|^2}{2\sigma_b^2} \right)}_{\text{'bilateral' kernel}} + \underbrace{\theta_2 \exp\left(-\frac{\left\| s_i - s_j \right\|^2}{2\sigma_\gamma^2} \right)}_{\text{smoothness kernel}} \right]$$

- $\mathbf{1}_{y_i \neq y_j}$ – uniform penalty for nearby pixels with different labels.
  - x Insensitive to compatibility between labels!

P. Krahenbuhl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in NIPS, 2011.

Boundary recovery



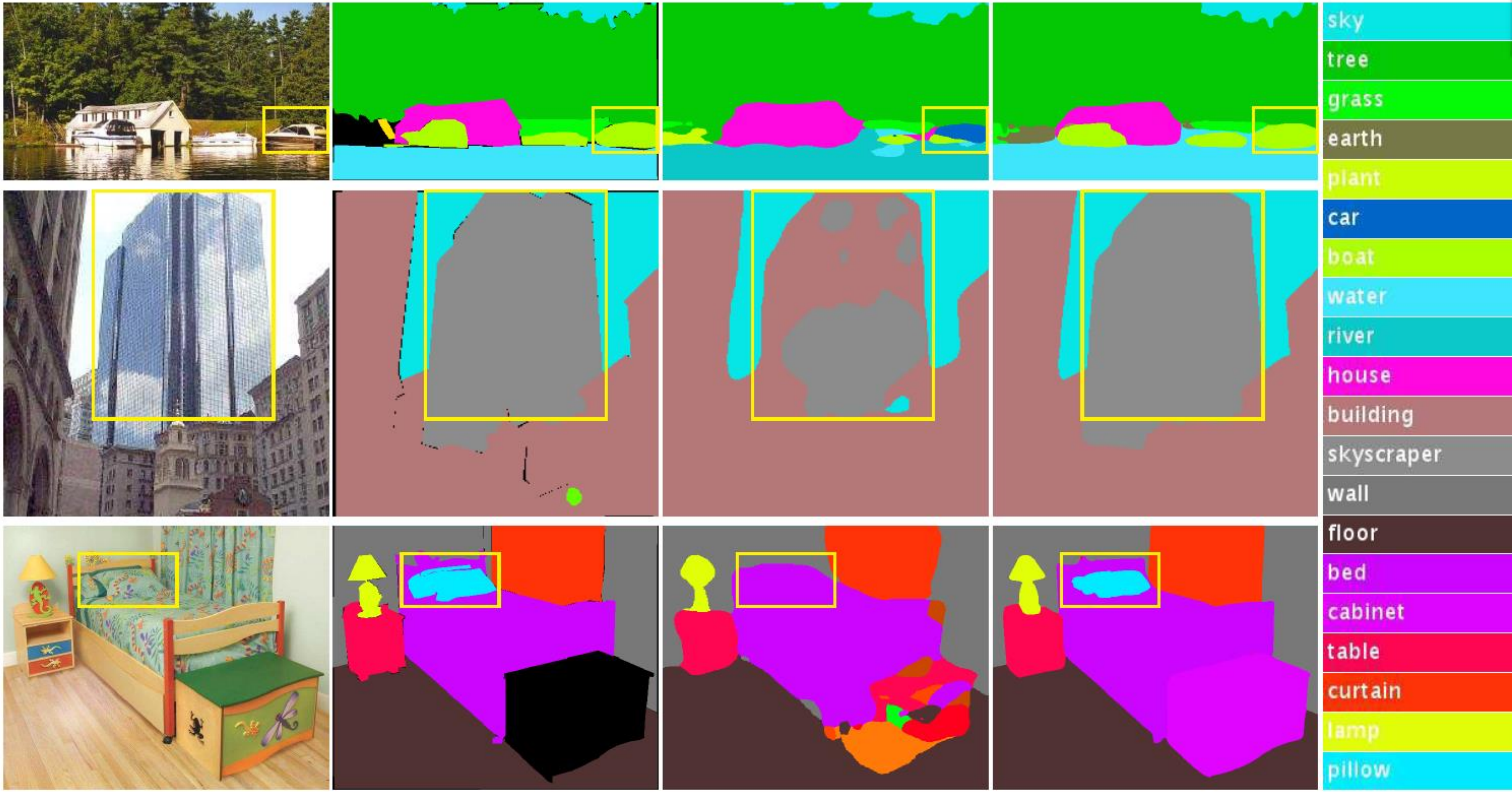
L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in ICLR, 2015.

DeepLab

- **Group:**
  - CCVL (Center for Cognition, Vision, and Learning).

- **Basis networks** (pre-trained for ImageNet)**:**
  - VGG-16 (Oxford Visual Geometry Group, ILSVRC 2014 1st).
  - ResNet-101 (Microsoft Research Asia, ILSVRC 2015 1st).

- **Code:** https://bitbucket.org/deeplab/deeplab-public/

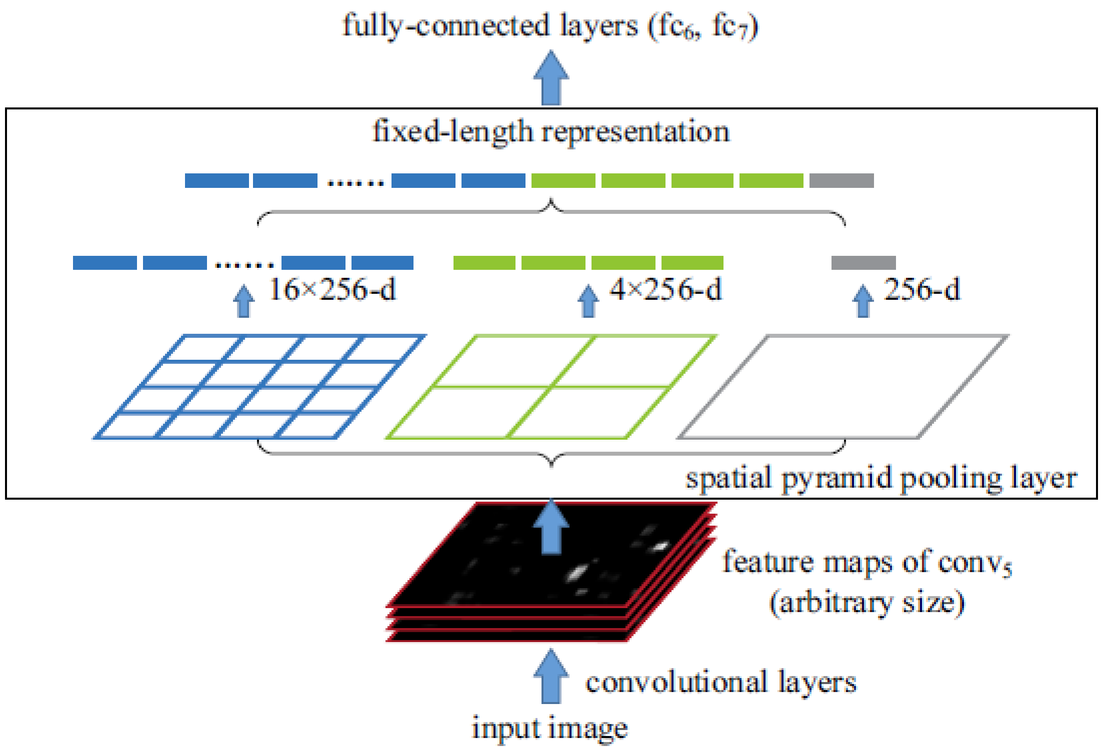


(a) Image (b) Ground Truth (c) FCN (d) PSPNet (e) ColorMap

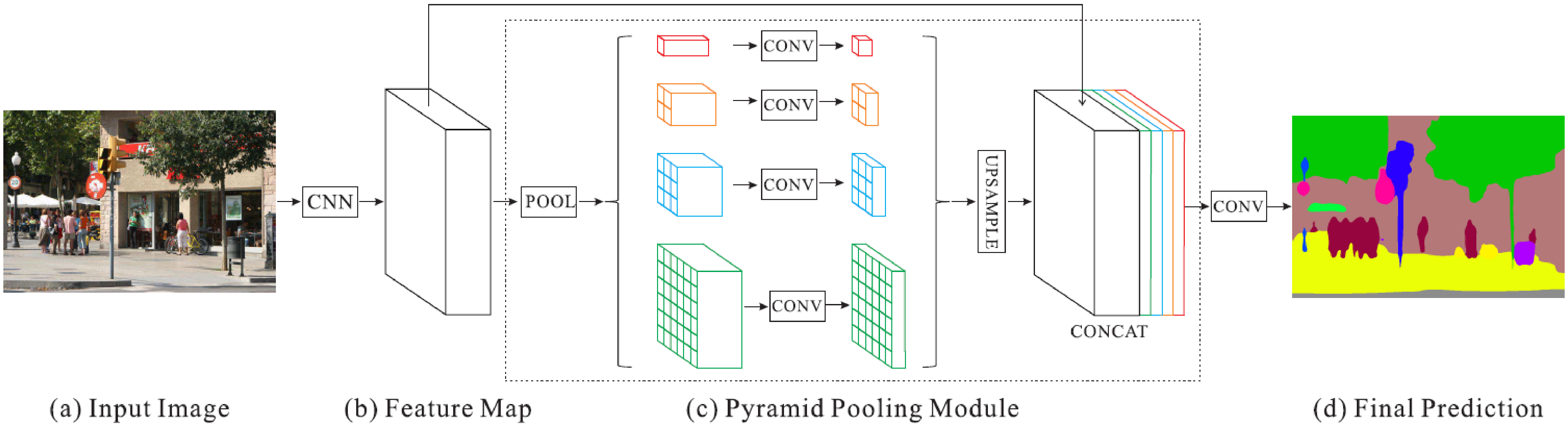
Pyramid Pooling Model

Spatial Pyramid Pooling:

- Feature map: a×a
- Pyramid level: n×n
- Window size: $\lceil a/n \rceil$
- Stride: $\lfloor a/n \rfloor$



Pyramid Pooling Model

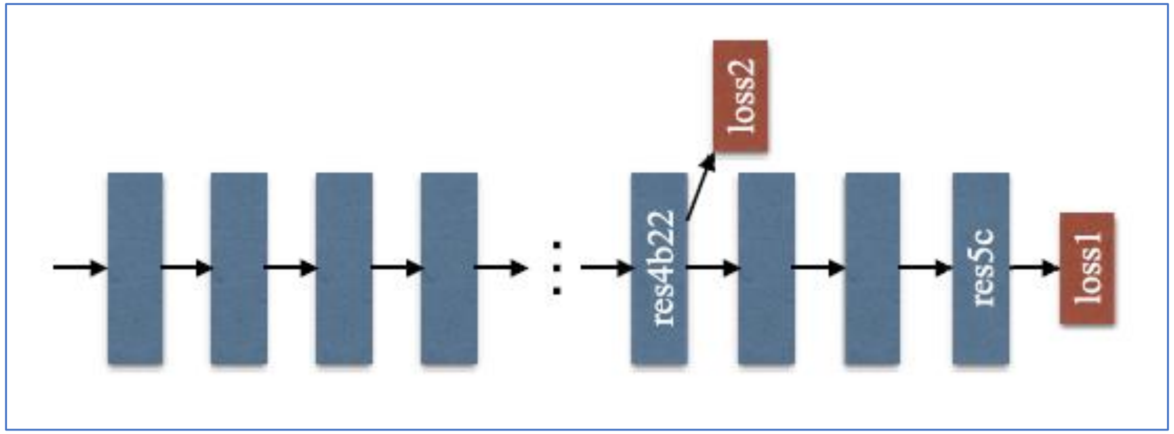


Auxiliary Loss

Slide credit to Kaicheng Wang

Ablation Study

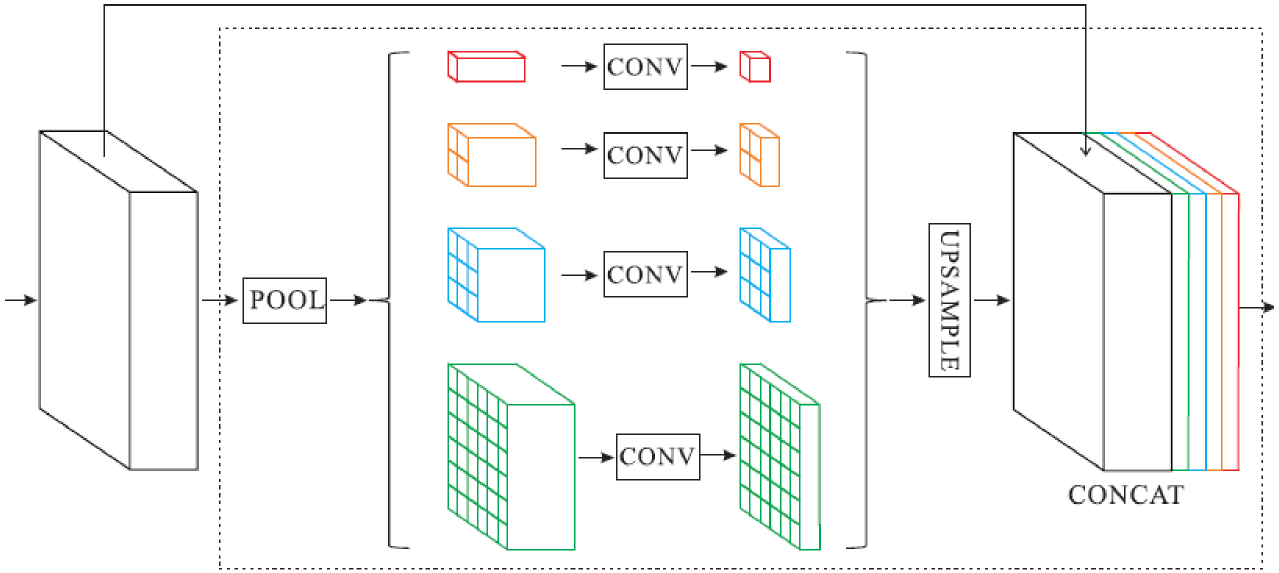
- Auxiliary Loss



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

Ablation Study

- PSPNet:



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**

Slide credit to Kaicheng Wang

Experiment Details

- "Poly" learning rate as DeepLabv2
  - rate = $(1 - \frac{iter}{\max\_iter})^{power}$,with power=0.9
  - (start_lr – end_lr)*rate + end_lr
- Augmentation
  - Mirror
  - Resize between 0.5 – 2
  - Gaussian blur

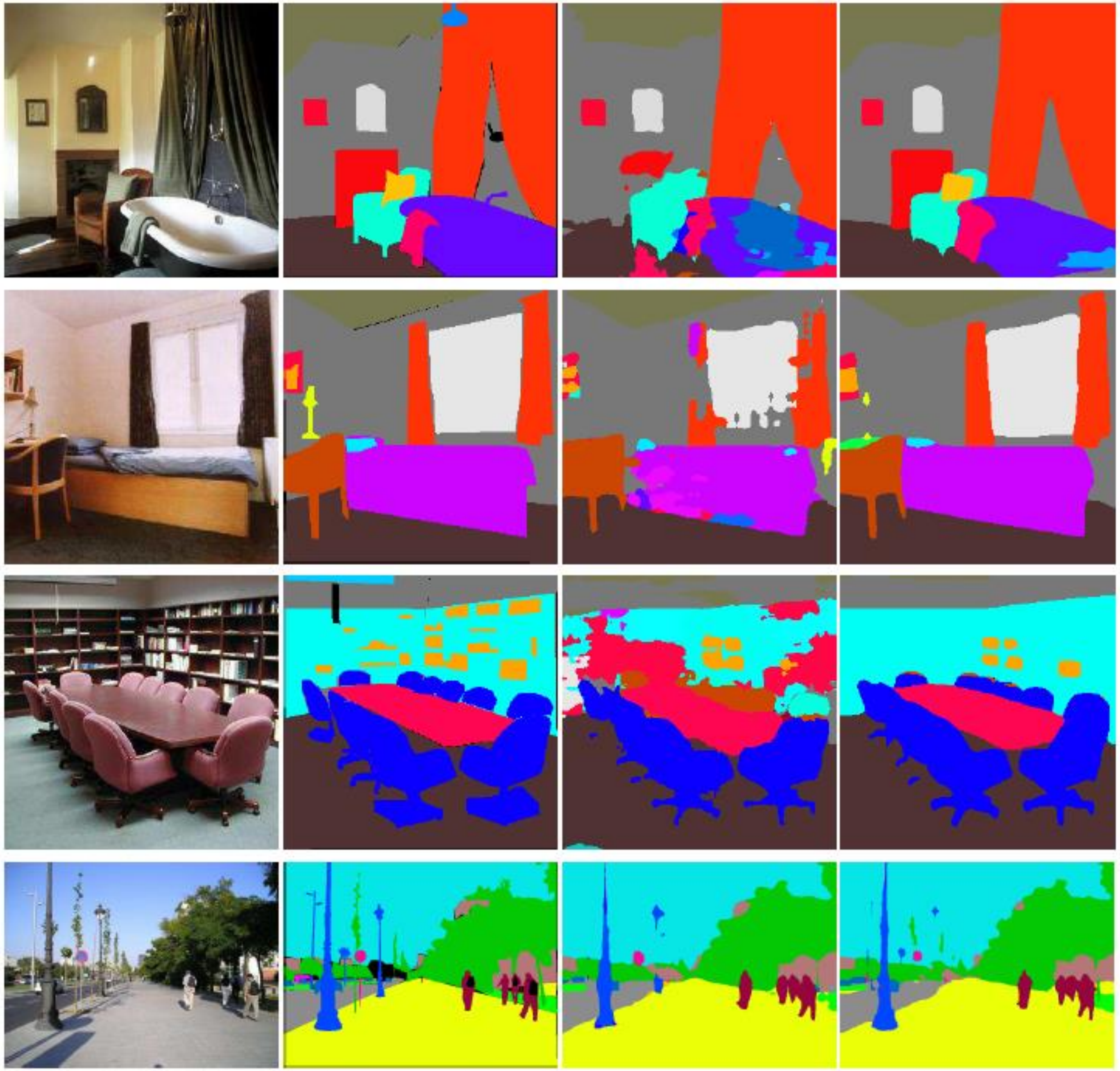
Slide credit to Kaicheng Wang

Validation on ADE20K

Method	Mean IoU(%)	Pixel Acc.(%)
FCN [26]	29.39	71.32
SegNet [2]	21.64	71.00
DilatedNet [40]	32.31	73.55
CascadeNet [43]	34.90	74.52
ResNet50-Baseline	34.28	76.35
ResNet50+DA	35.82	77.07
ResNet50+DA+AL	37.23	78.01
ResNet50+DA+AL+PSP	**41.68**	**80.04**
ResNet269+DA+AL+PSP	43.81	80.88
ResNet269+DA+AL+PSP+MS	**44.94**	**81.69**

Slide credit to Kaicheng Wang

Validation on ADE20K



(a) Image (b) Ground Truth (c) Baseline (d) PSPNet

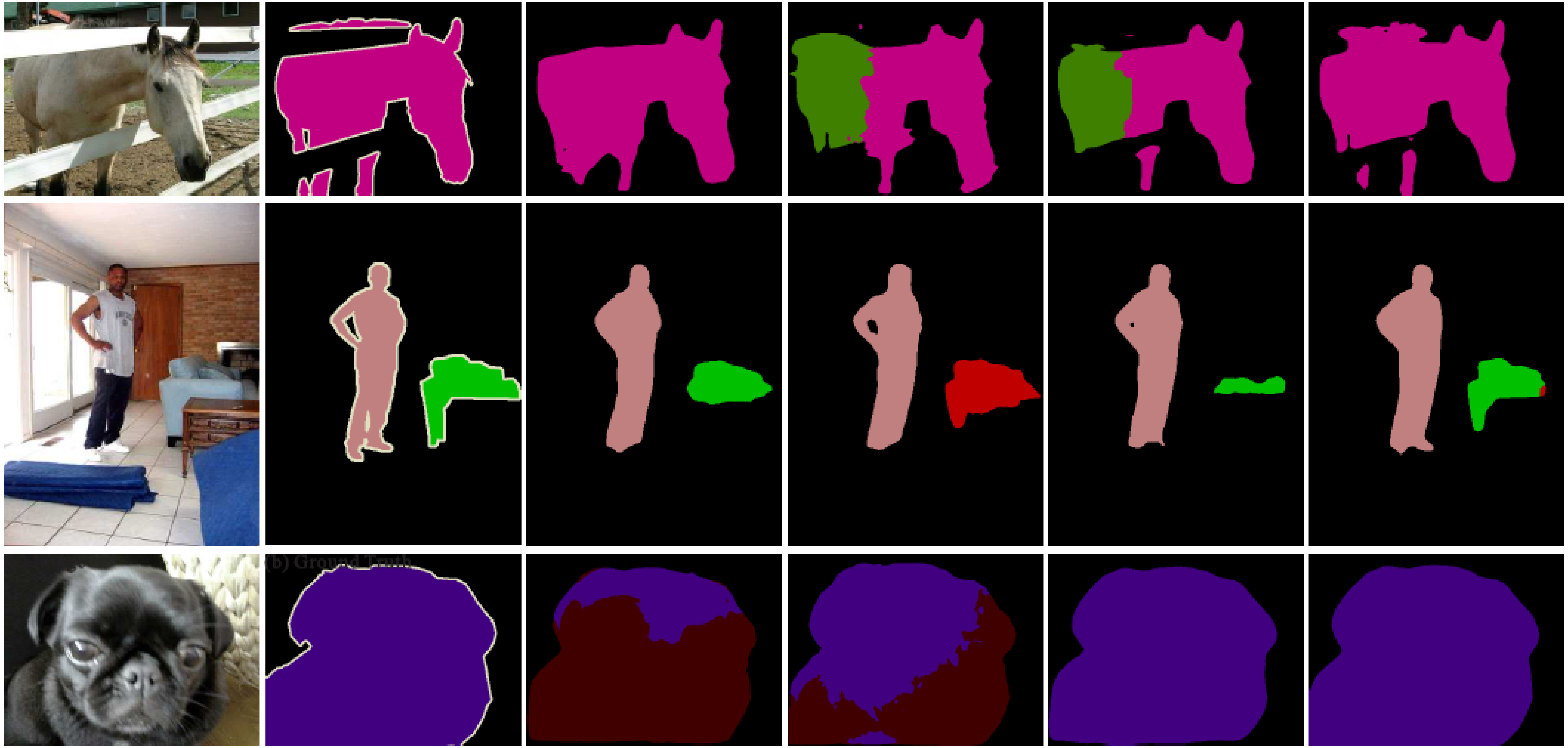
Slide credit to Kaicheng Wang

PASCAL VOC2012

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
FCN [26]	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
Zoom-out [28]	85.6	37.3	83.2	62.5	66.0	85.1	80.7	84.9	27.2	73.2	57.5	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	63.3	69.6
DeepLab [3]	84.4	54.5	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	59.7	82.2	50.4	73.1	63.7	71.6
CRF-RNN [41]	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1	72.0
DeconvNet [30]	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	72.5
GCRF [36]	85.2	43.9	83.3	65.2	68.3	89.0	82.7	85.3	31.1	79.5	63.3	80.5	79.3	85.5	81.0	60.5	85.5	52.0	77.3	65.1	73.2
DPN [25]	87.7	59.4	78.4	64.9	70.3	89.3	83.5	86.1	31.7	79.9	62.6	81.9	80.0	83.5	82.3	60.5	83.2	53.4	77.9	65.0	74.1
Piecewise [20]	90.6	37.6	80.0	67.8	74.4	92.0	85.2	86.2	39.1	81.2	58.9	83.8	83.9	84.3	84.8	62.1	83.2	58.2	80.8	72.3	75.3
PSPNet	**91.8**	**71.9**	**94.7**	**71.2**	**75.8**	**95.2**	**89.9**	**95.9**	**39.3**	**90.7**	**71.7**	**90.5**	**94.5**	**88.8**	**89.6**	**72.8**	**89.6**	**64.0**	**85.1**	**76.3**	**82.6**
CRF-RNN† [41]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
BoxSup† [7]	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	75.2
Dilation8† [40]	91.7	39.6	87.8	63.1	71.8	89.7	82.9	89.8	37.2	84.0	63.0	83.3	89.0	83.8	85.1	56.8	87.6	56.0	80.2	64.7	75.3
DPN† [25]	89.0	61.6	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	77.5
Piecewise† [20]	94.1	40.7	84.1	67.8	75.9	93.4	84.3	88.4	42.5	86.4	64.7	85.4	89.0	85.8	86.0	67.5	90.2	63.8	80.9	73.0	78.0
FCRNs† [38]	91.9	48.1	93.4	69.3	75.5	94.2	87.5	92.8	36.7	86.9	65.2	89.1	90.2	86.5	87.2	64.6	90.1	59.7	85.5	72.7	79.1
LRR† [9]	92.4	45.1	94.6	65.2	75.8	**95.1**	89.1	92.3	39.0	85.7	70.4	88.6	89.4	88.6	86.6	65.8	86.2	57.4	85.7	77.3	79.3
DeepLab† [4]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5	79.7
PSPNet†	**95.8**	**72.7**	**95.0**	**78.9**	**84.4**	94.7	**92.0**	**95.7**	**43.1**	**91.0**	**80.3**	**91.3**	**96.3**	**92.3**	**90.1**	**71.5**	**94.4**	**66.9**	**88.8**	**82.0**	**85.4**

PASCAL VOC2012

Methoc	mIoU
FCN [2	62.2
Zoom-(	69.6
DeepL	71.6
CRF-R	72.0
Deconv	72.5
GCRF	73.2
DPN [2	74.1
Piecew	75.3
PSPNe	**82.6**
CRF-R	74.7
BoxSu	75.2
Dilatio	75.3
DPN†	77.5
Piecew	78.0
FCNs	79.1
LRR†	79.3
DeepL	79.7
PSPNe	**85.4**



(a) Image (b) Ground Truth (c) FCN (d) DPN (e) DeepLab (f) PSPNet

Slide credit to Kaicheng Wang

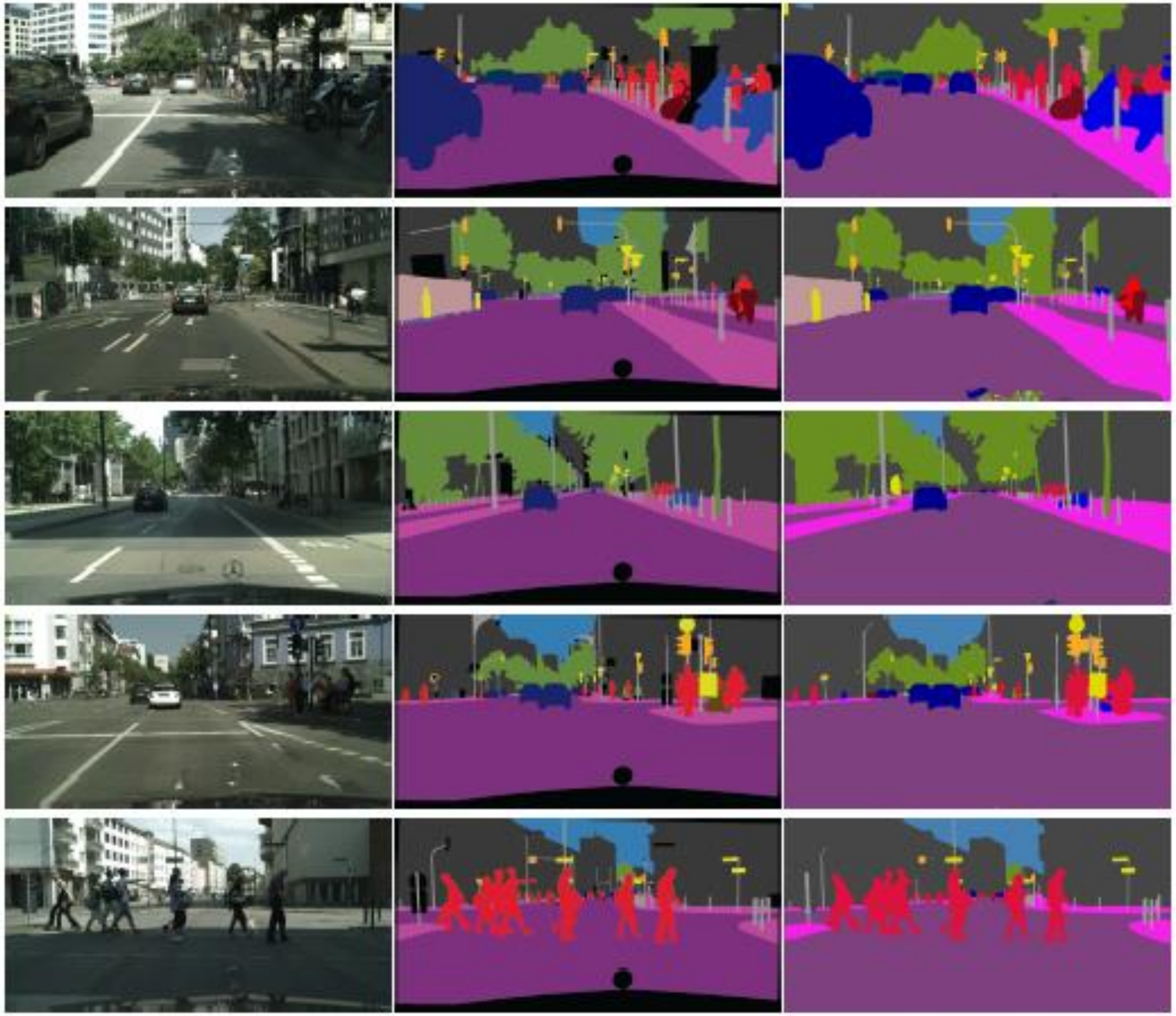
Cityscapes

Method	road	swalk	build.	wall	fence	pole	tlight	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
CRF-RNN [41]	96.3	73.9	88.2	47.6	41.3	35.2	49.5	59.7	90.6	66.1	93.5	70.4	34.7	90.1	39.2	57.5	55.4	43.9	54.6	62.5
FCN [26]	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65.0	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8	65.3
SiCNN+CRF [16]	96.3	76.8	88.8	40.0	45.4	50.1	63.3	69.6	90.6	67.1	92.2	77.6	55.9	90.1	39.2	51.3	44.4	54.4	66.1	66.3
DPN [25]	97.5	78.5	89.5	40.4	45.9	51.1	56.8	65.3	91.5	69.4	94.5	77.5	54.2	92.5	44.5	53.4	49.9	52.1	64.8	66.8
Dilation10 [40]	97.6	79.2	89.9	37.3	47.6	53.2	58.6	65.2	91.8	69.4	93.7	78.9	55.0	93.3	45.5	53.4	47.7	52.2	66.0	67.1
LRR [9]	97.7	79.9	90.7	44.4	48.6	58.6	68.2	72.0	92.5	69.3	94.7	81.6	60.0	94.0	43.6	56.8	47.2	54.8	69.7	69.7
DeepLab [4]	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8	70.4
Piecewise [20]	98.0	82.6	90.6	44.0	50.7	51.1	65.0	71.7	92.0	72.0	94.1	81.5	61.1	94.3	61.1	65.1	53.8	61.6	70.6	71.6
PSPNet	**98.6**	**86.2**	**92.9**	**50.8**	**58.8**	**64.0**	**75.6**	**79.0**	**93.4**	**72.3**	**95.4**	**86.5**	**71.3**	**95.9**	**68.2**	**79.5**	**73.8**	**69.5**	**77.2**	**78.4**
LRR‡ [9]	97.9	81.5	91.4	50.5	52.7	59.4	66.8	72.7	92.5	70.1	95.0	81.3	60.1	94.3	51.2	67.7	54.6	55.6	69.6	71.8
PSPNet‡	**98.6**	**86.6**	**93.2**	**58.1**	**63.0**	**64.5**	**75.2**	**79.2**	**93.4**	**72.1**	**95.1**	**86.3**	**71.4**	**96.0**	**73.5**	**90.4**	**80.3**	**69.9**	**76.9**	**80.2**

Slide credit to Kaicheng Wang

Cityscap

Method	road	swal	...	us	train	mbike	bike	mIoU
CRF-RNN [41]	96.3	73.		.5	55.4	43.9	54.6	62.5
FCN [26]	97.4	78.		.6	46.5	51.6	66.8	65.3
SiCNN+CRF [16]	96.3	76.		.3	44.4	54.4	66.1	66.3
DPN [25]	97.5	78.		.4	49.9	52.1	64.8	66.8
Dilation10 [40]	97.6	79.		.4	47.7	52.2	66.0	67.1
LRR [9]	97.7	79.		.8	47.2	54.8	69.7	69.7
DeepLab [4]	97.9	81.		.5	57.5	57.7	68.8	70.4
Piecewise [20]	98.0	82.		.1	53.8	61.6	70.6	71.6
PSPNet	**98.6**	**86.**		**.5**	**73.8**	**69.5**	**77.2**	**78.4**
LRR‡ [9]	97.9	81.		.7	54.6	55.6	69.6	71.8
PSPNet‡	**98.6**	**86.**		**.4**	**80.3**	**69.9**	**76.9**	**80.2**



(a) Image (b) Ground Truth (c) PSPNet

Slide credit to Kaicheng Wang

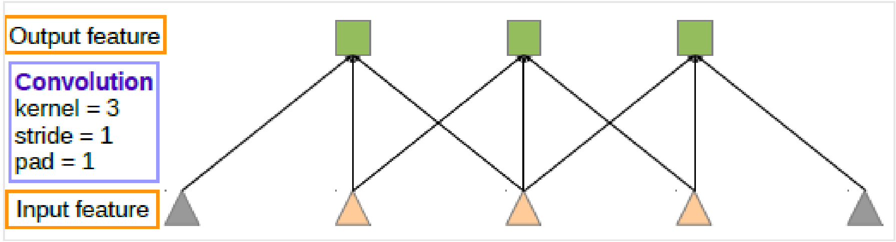
Dilated Convolution

- Innovation: Tiny represented feature map
- Removing striding → Receptive field decreases
- Solution: Dilated convolution
  - Resolution + Receptive Field

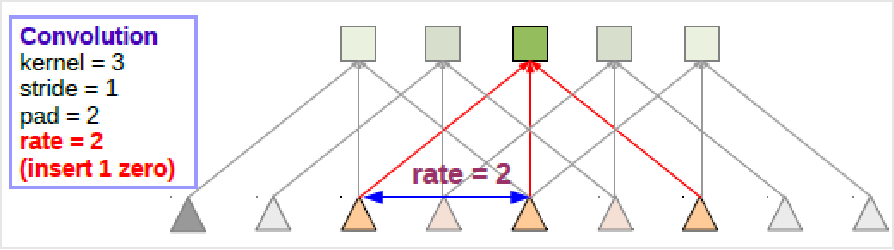
Atrous Convolution

- Convolution with holes
- Also called dilated convolution

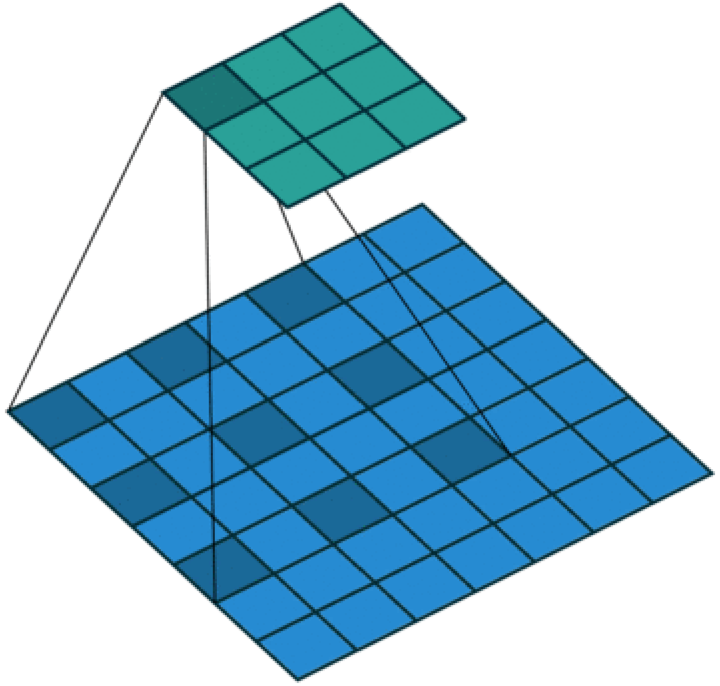
$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k] w[k]$$



(a) Sparse feature extraction

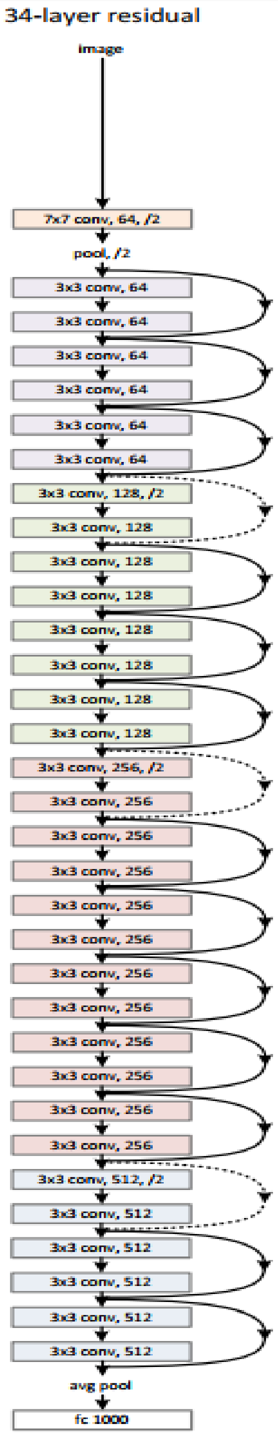


(b) Dense feature extraction

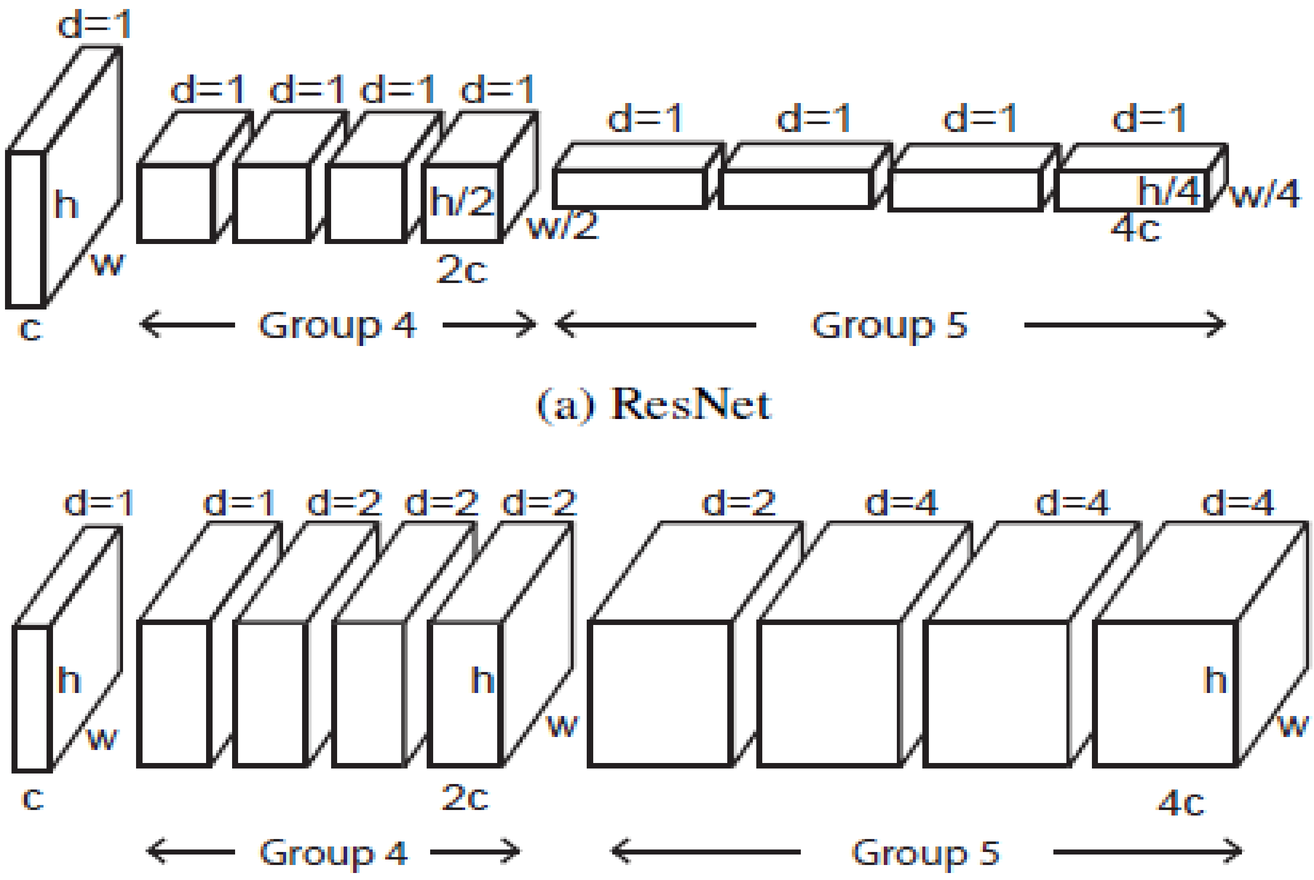


ResNet Base

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9



DRN structure



(a) ResNet

(b) DRN

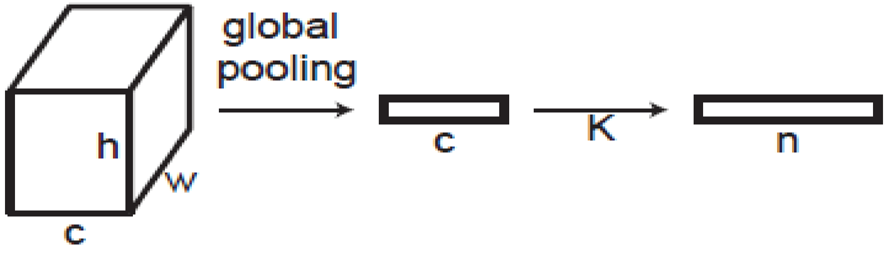
$\mathcal{G}_1^4$、$\mathcal{G}_1^5$ striding removed

→ Dilated convolution

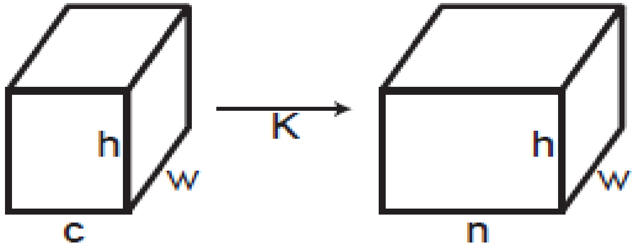
Slide credit to Kaicheng Wang

Prediction Model

- Instantaneously used for classification and localization



(a) Classification output

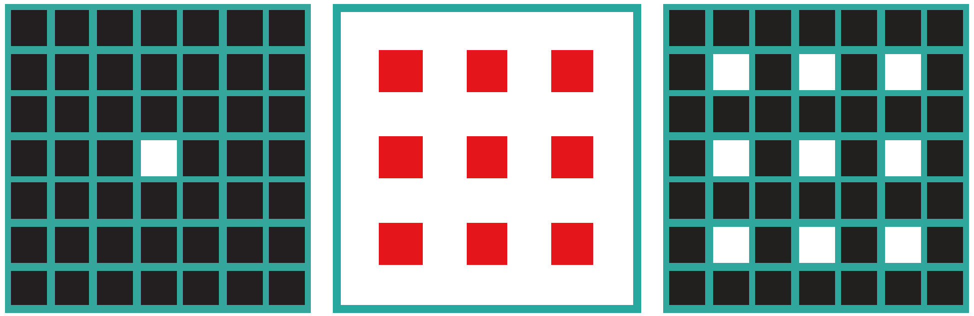


(b) Localization output

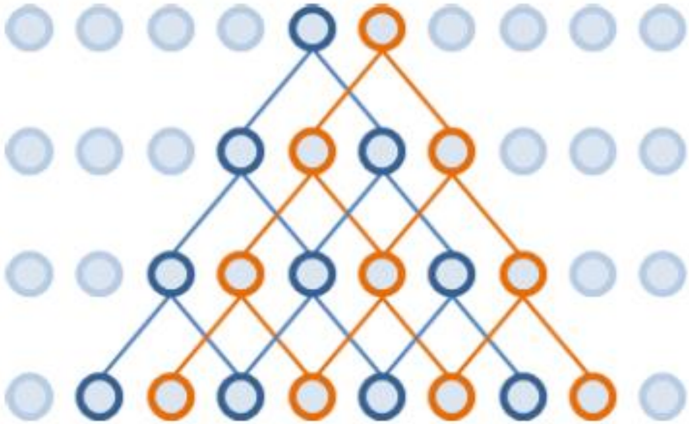
Slide credit to Kaicheng Wang

Dilated Defect

- Gridding artifacts
  - Nearby pixels receive information from different grid

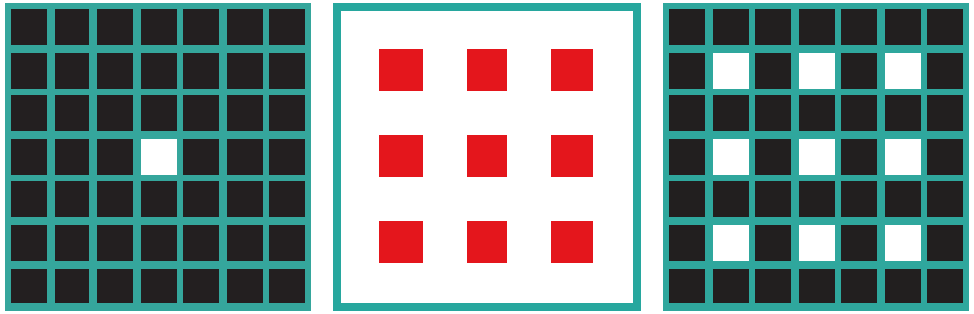


(a) Input (b) Dilation 2 (c) Output



Dilated Defect

- Gridding artifacts
  - Nearby pixels receive information from different grid



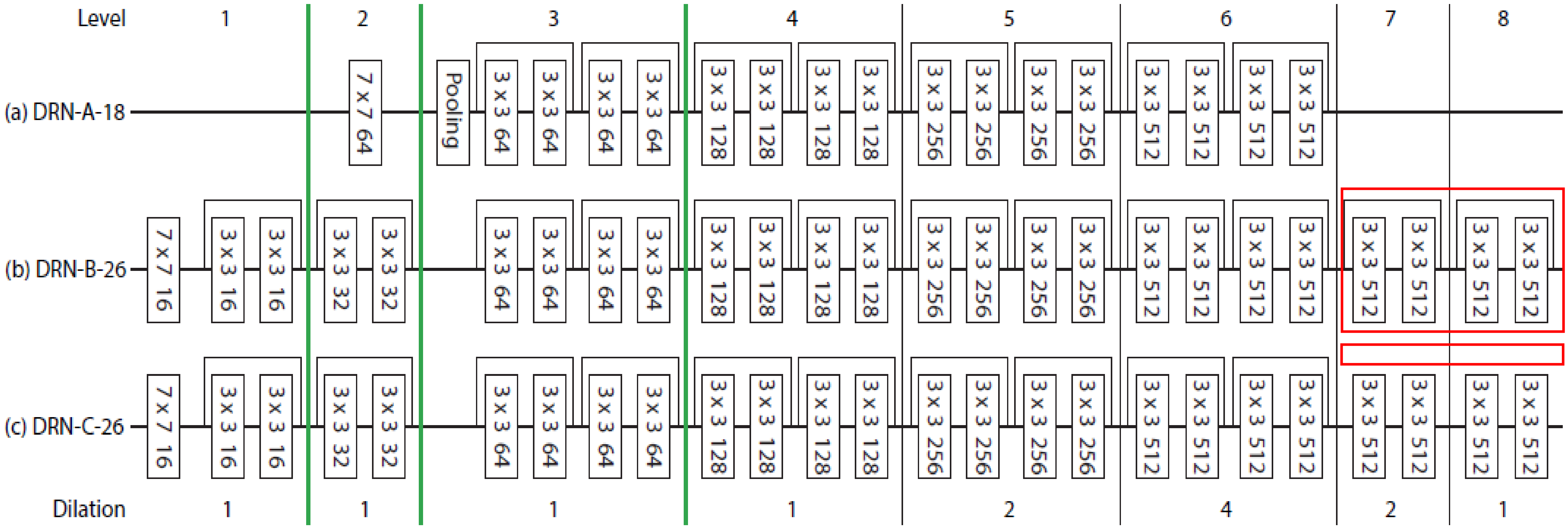
(a) Input (b) Dilation 2 (c) Output

(a) Input (b) ResNet-18 (c) DRN-A-18

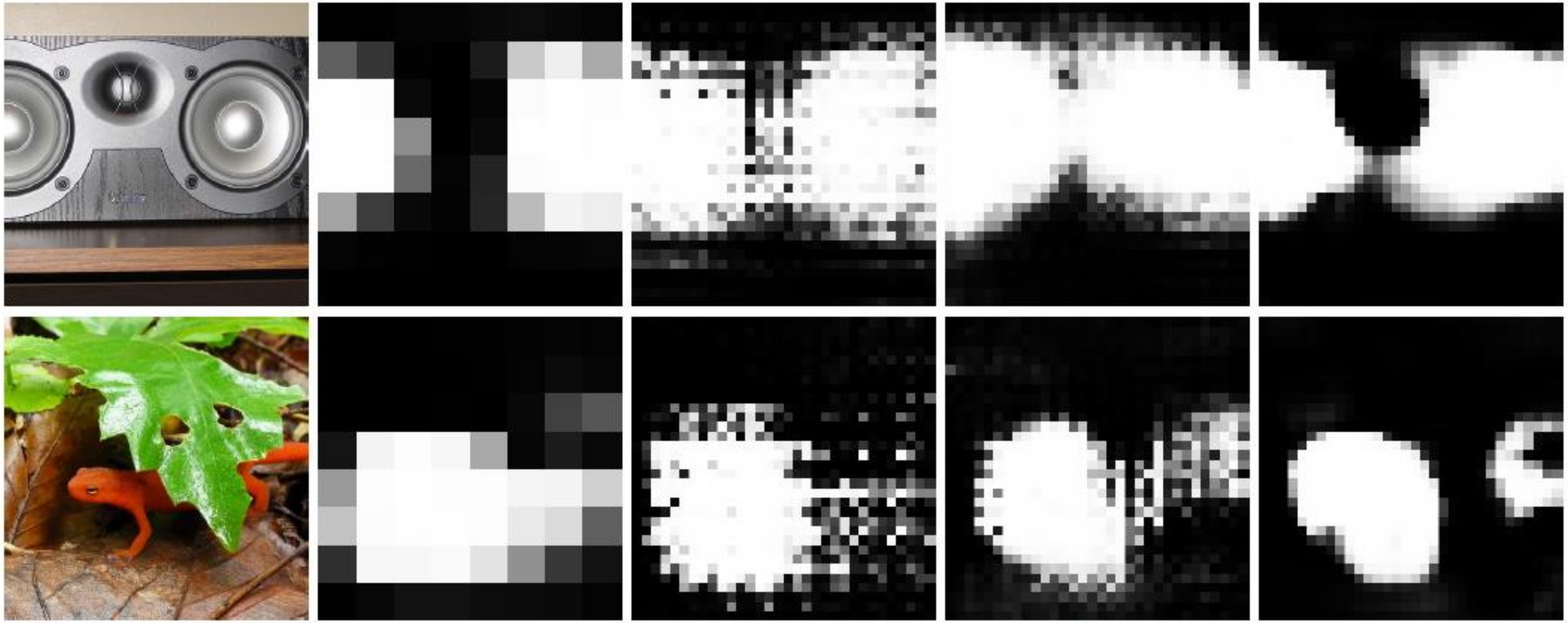
Dilated Defect

- Removing max pooling
- Adding layers
- Removing residual connections

Dilated Defect



Dilated Defect



(a) Input (b) ResNet-18 (c) DRN-A-18 (d) DRN-B-26 (e) DRN-C-26

Experimental result

- Classification :
  - ImageNet 2012

Model	1 crop		10 crops		P
	top-1	top-5	top-1	top-5	
ResNet-18	30.43	10.76	28.22	9.42	11.7M
DRN-A-18	28.00	9.50	25.75	8.25	11.7M
DRN-B-26	25.19	7.91	23.33	6.69	21.1M
DRN-C-26	24.86	7.55	22.93	6.39	21.1M
ResNet-34	27.73	8.74	24.76	7.35	21.8M
DRN-A-34	24.81	7.54	22.64	6.34	21.8M
DRN-C-42	22.94	6.57	21.20	5.60	31.2M
ResNet-50	24.01	7.02	22.24	6.08	25.6M
DRN-A-50	22.94	6.57	21.34	5.74	25.6M
ResNet-101	22.44	6.21	21.08	5.35	44.5M

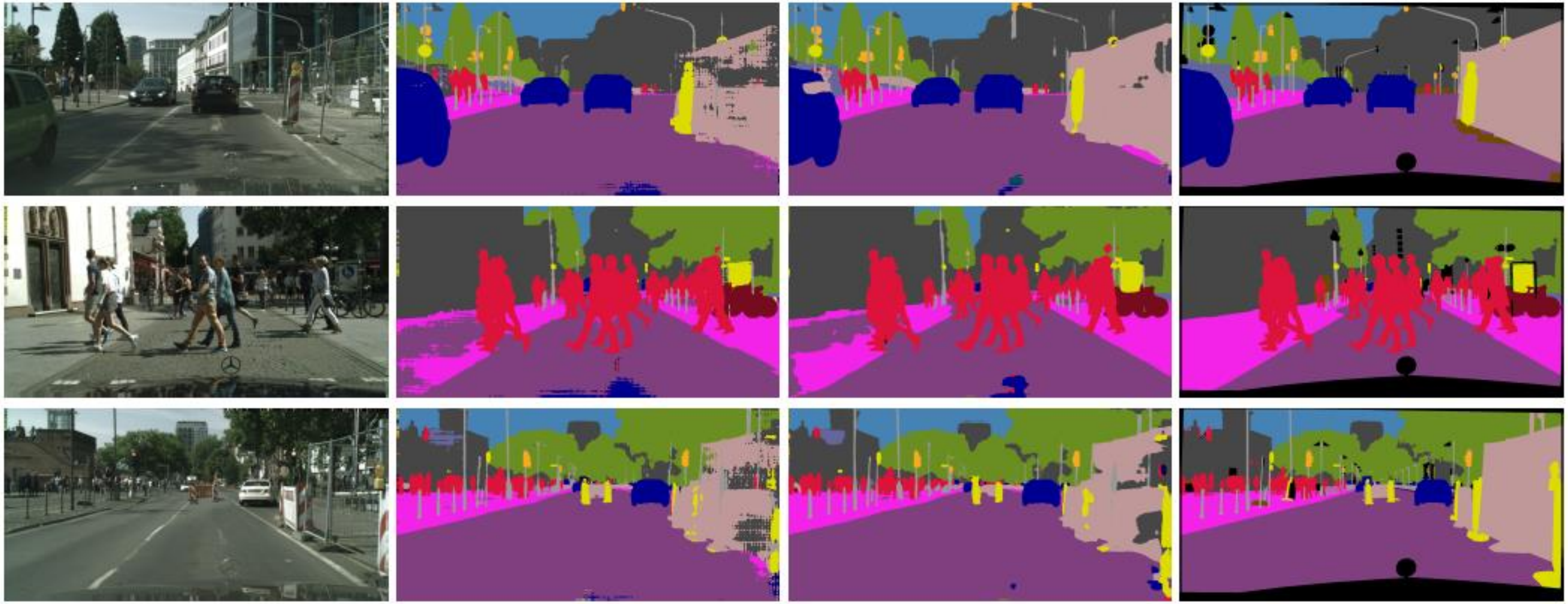
Table 1: Image classification accuracy (error rates) on the ImageNet 2012 validation set. Lower is better. P is the number of parameters in each model.

Experimental result

- Cityscapes

	Road	Sidewalk	Building	Wall	Fence	Pole	Light	Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	mean IoU
DRN-A-50	96.9	77.4	90.3	35.8	42.8	59.0	**66.8**	74.5	91.6	57.0	93.4	78.7	55.3	92.1	43.2	59.5	36.2	**52.0**	**75.2**	67.3
DRN-C-26	97.4	80.7	90.4	36.1	47.0	56.9	63.8	73.0	91.2	**57.9**	93.4	77.3	53.8	92.7	45.0	70.5	48.4	44.2	72.8	68.0
DRN-C-42	**97.7**	**82.2**	**91.2**	**40.5**	**52.6**	**59.2**	66.7	**74.6**	**91.7**	57.7	**94.1**	**79.1**	**56.0**	**93.6**	**56.0**	**74.3**	**54.7**	50.9	74.1	**70.9**

Experimental result



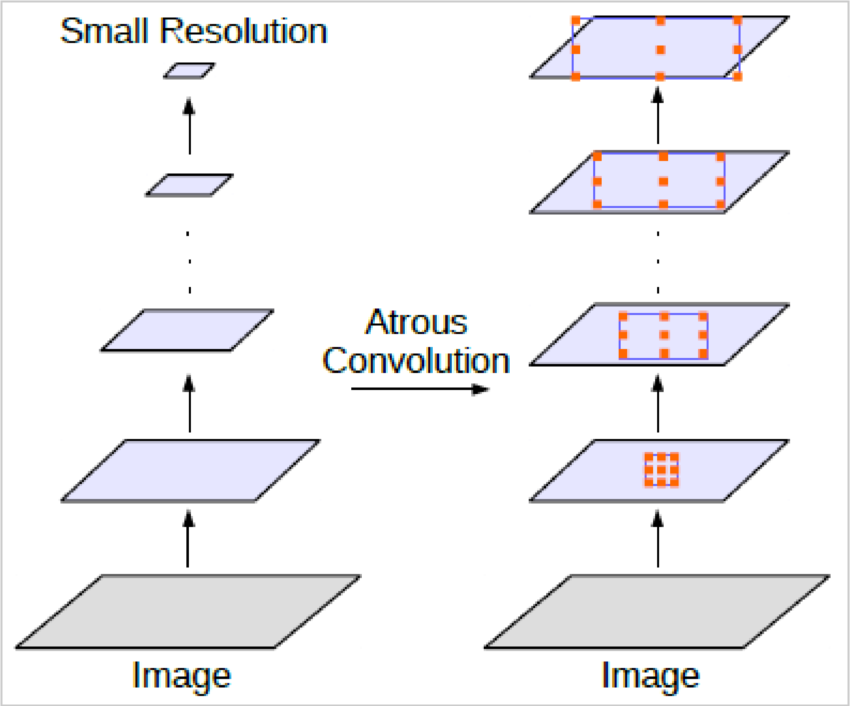
(a) Input (b) DRN-A-50 (c) DRN-C-26 (d) Ground truth

DeepLabv3

- Introducing
  - Multigrid
  - Image-level features encoding

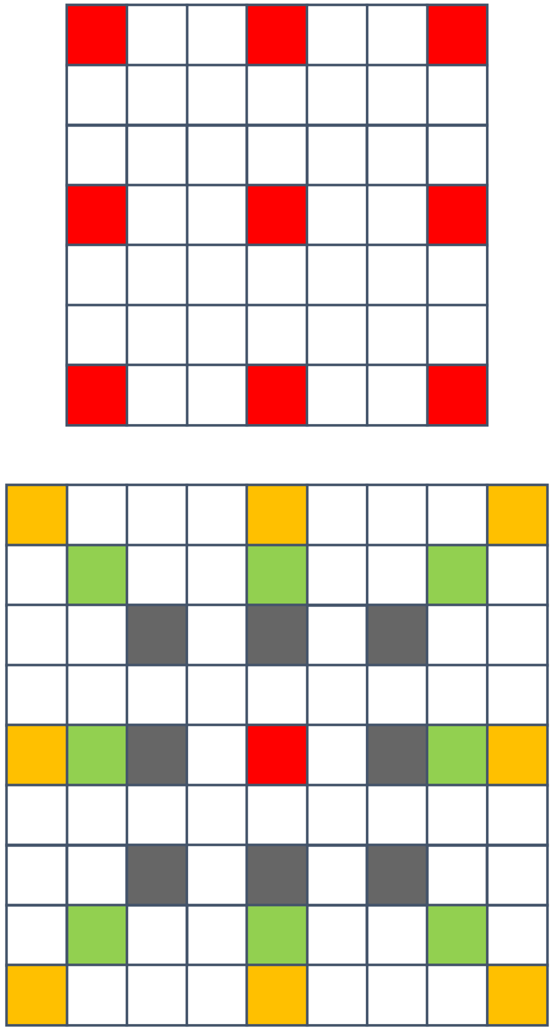
    global context

  → Global average pooling



Multigrid

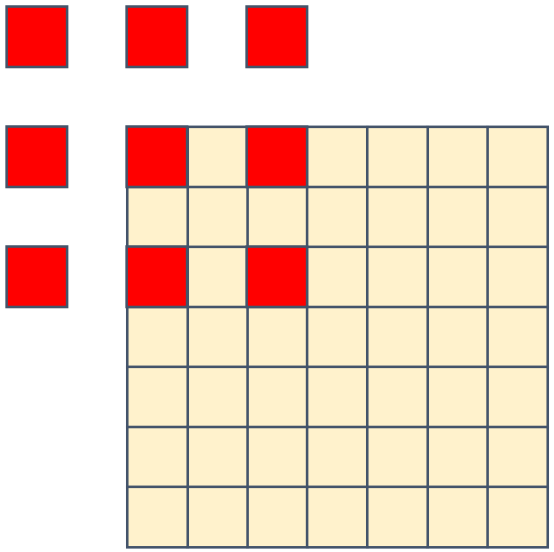
- Dilated defect
  - Local information missing
- Solution
  - Different dilation rates

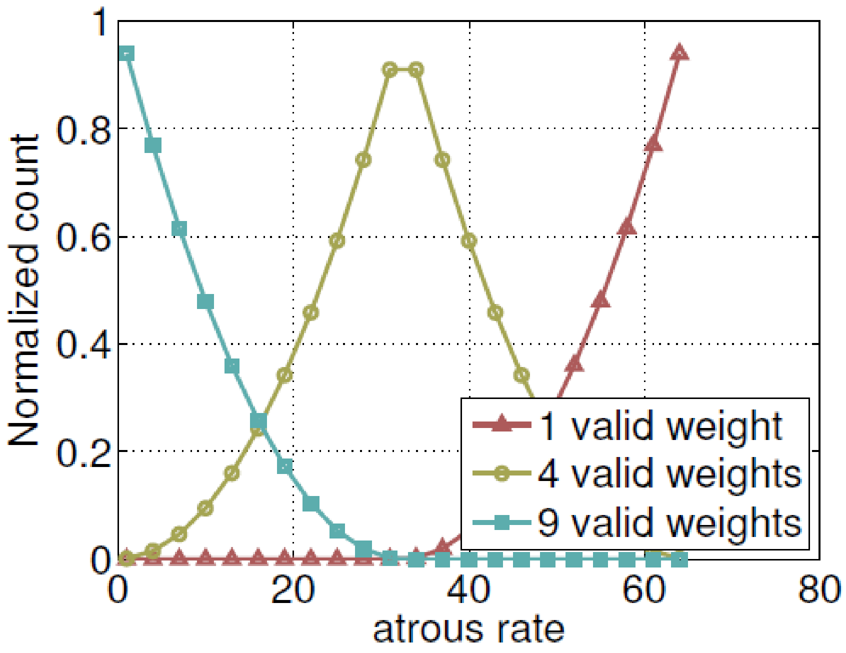


Global Average Pooling

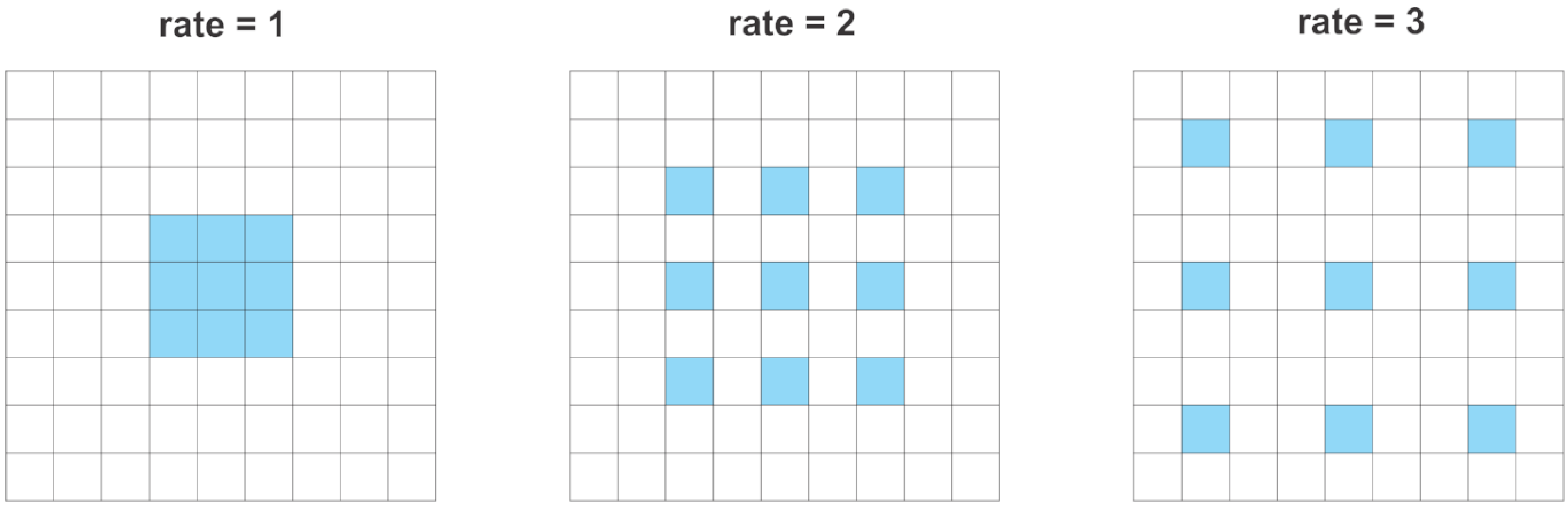
- Valid weight decreases as sampling rate increases

→ Global average pooling



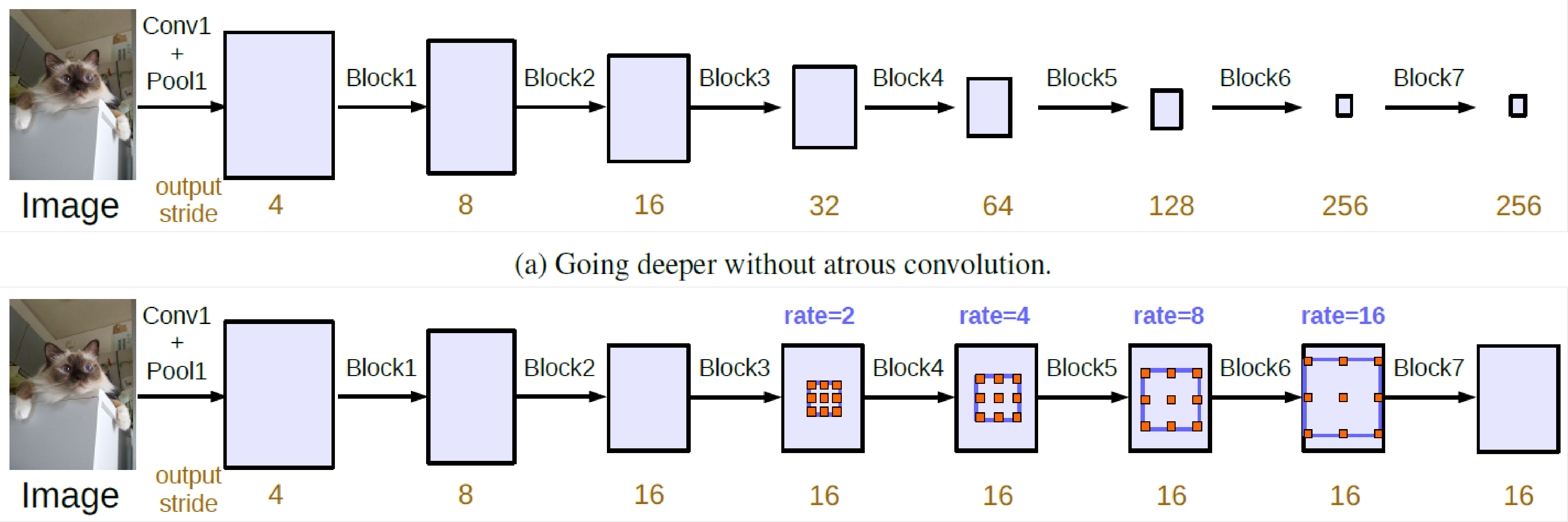


Atrous (or dilated) convolutions are regular convolutions with a factor that allows us to expand the filter's field of view.
Consider a *3x3* convolution filter for instance. When the dilation rate is equal to 1, it behaves like a standard convolution. But, if we set the dilation factor to 2, it has the effect of enlarging the convolution kernel.



Cascaded with Atrous

- Duplicating with atrous



(a) Going deeper without atrous convolution.

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

Figure 3. Cascaded modules without and with atrous convolution.

Parallel ASPP

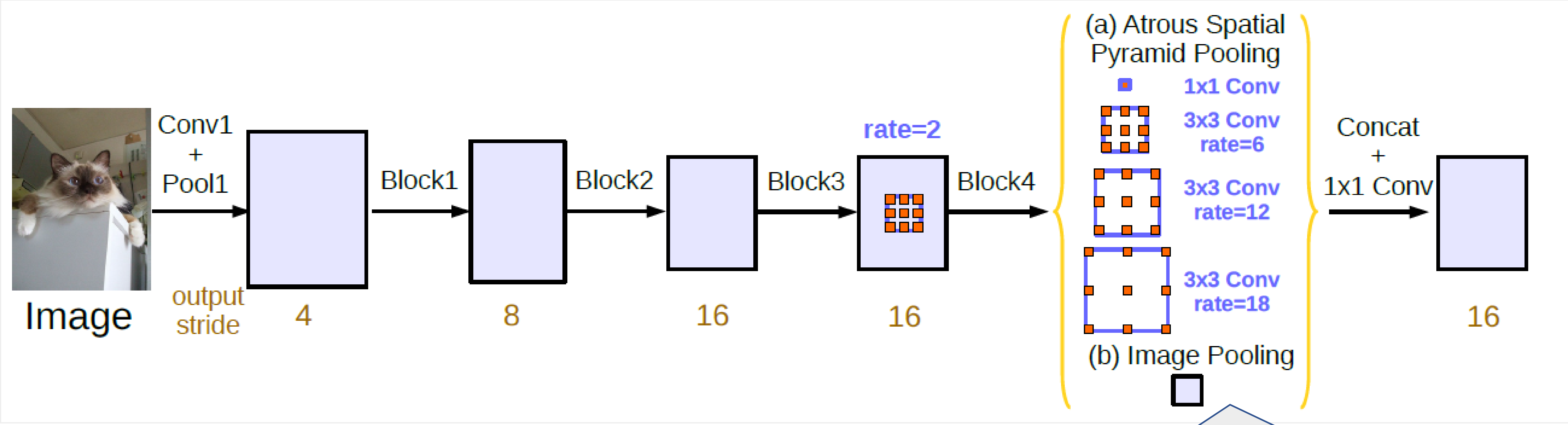


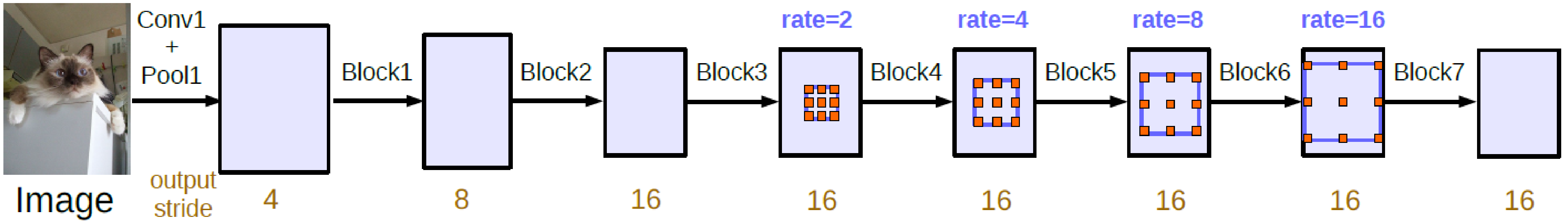
Figure 5. Parallel modules with atrous convolution (ASPP), … evel features.

Global Context
Information
(PSPnet)

Training Protocol

- Learning rate
  - Poly, same as DeepLabv2
- Crop size
  - Large crop size needed for large rate
- Batch normalization
  - Large batch needed

Going Deeper



Network	block4	block5	block6	block7
ResNet-50	64.81	72.14	74.29	73.88
ResNet-101	68.39	73.21	75.34	75.76

Table 2. Going deeper with atrous convolution when employing ResNet-50 and ResNet-101 with different number of cascaded blocks at *output_stride* = 16. Network structures 'block4', 'block5', 'block6', and 'block7' add extra 0, 1, 2, 3 cascaded modules respectively. The performance is generally improved by adopting more cascaded blocks.

Multigrid

- Different rate within block4 to block7

Multi-Grid	block4	block5	block6	block7
(1, 1, 1)	68.39	73.21	75.34	75.76
(1, 2, 1)	70.23	75.67	76.09	**76.66**
(1, 2, 3)	73.14	75.78	75.96	76.11
(1, 2, 4)	73.45	75.74	75.85	76.02
(2, 2, 2)	71.45	74.30	74.70	74.62

Table 3. Employing multi-grid method for ResNet-101 with different number of cascaded blocks at *output_stride* $= 16$. The best model performance is shown in bold.

PASCAL VOC 2012

Method	mIOU
Adelaide_VeryDeep_FCN_VOC [73]	79.1
LRR_4x_ResNet-CRF [20]	79.3
DeepLabv2-CRF [10]	79.7
CentraleSupelec Deep G-CRF [7]	80.2
HikSeg_COCO [68]	81.4
Deep Layer Cascade (LC) [43]	82.7
TuSimple [72]	83.1
Large_Kernel_Matters [58]	83.6
Multipath-RefineNet [45]	84.2
ResNet-38_MS_COCO [74]	84.9
PSPNet [80]	85.4
DeepLabv3	85.7

Table 7. Performance on PASCAL VOC 2012 *test* set.

PASCAL VOC 2012

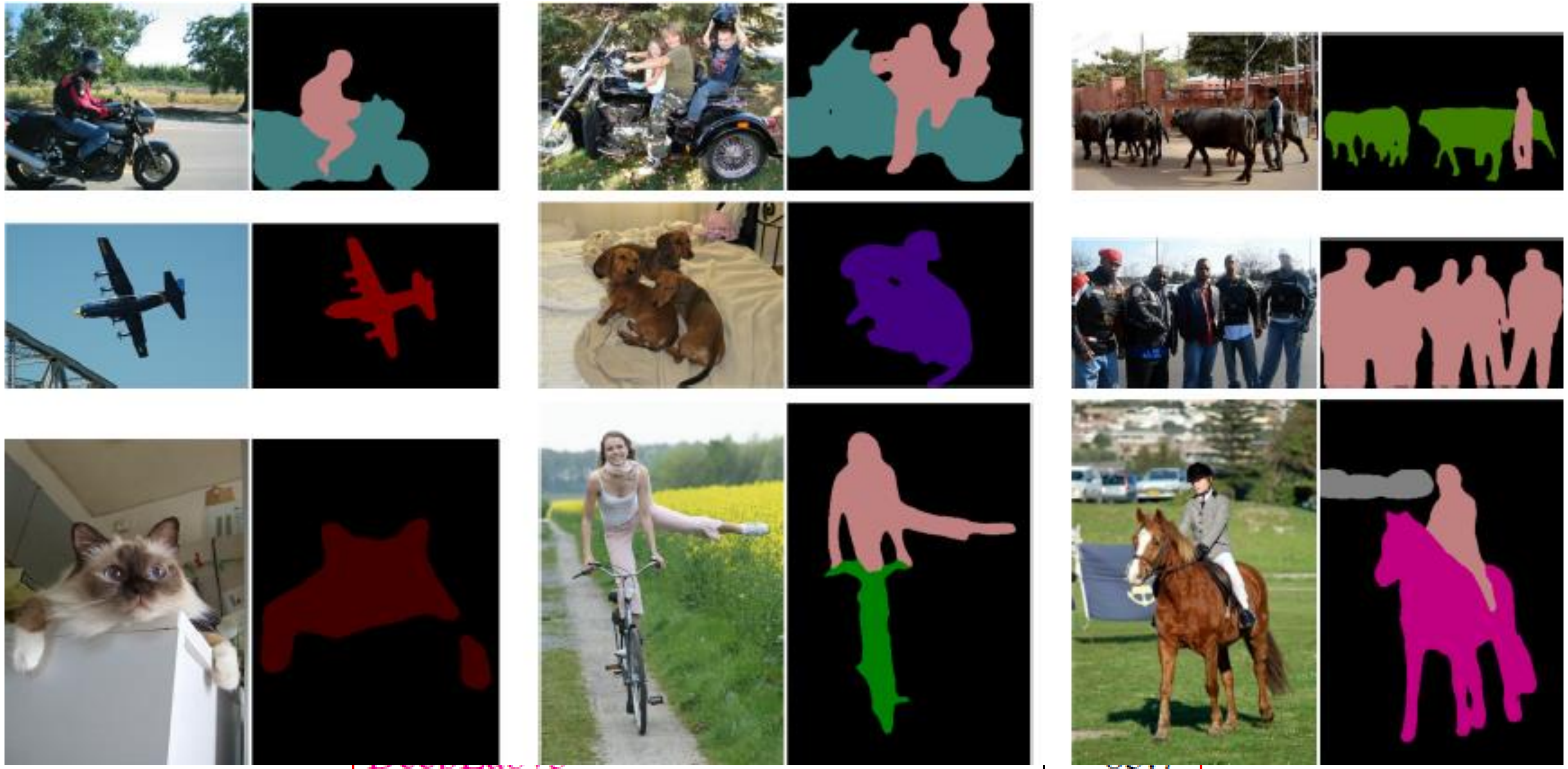


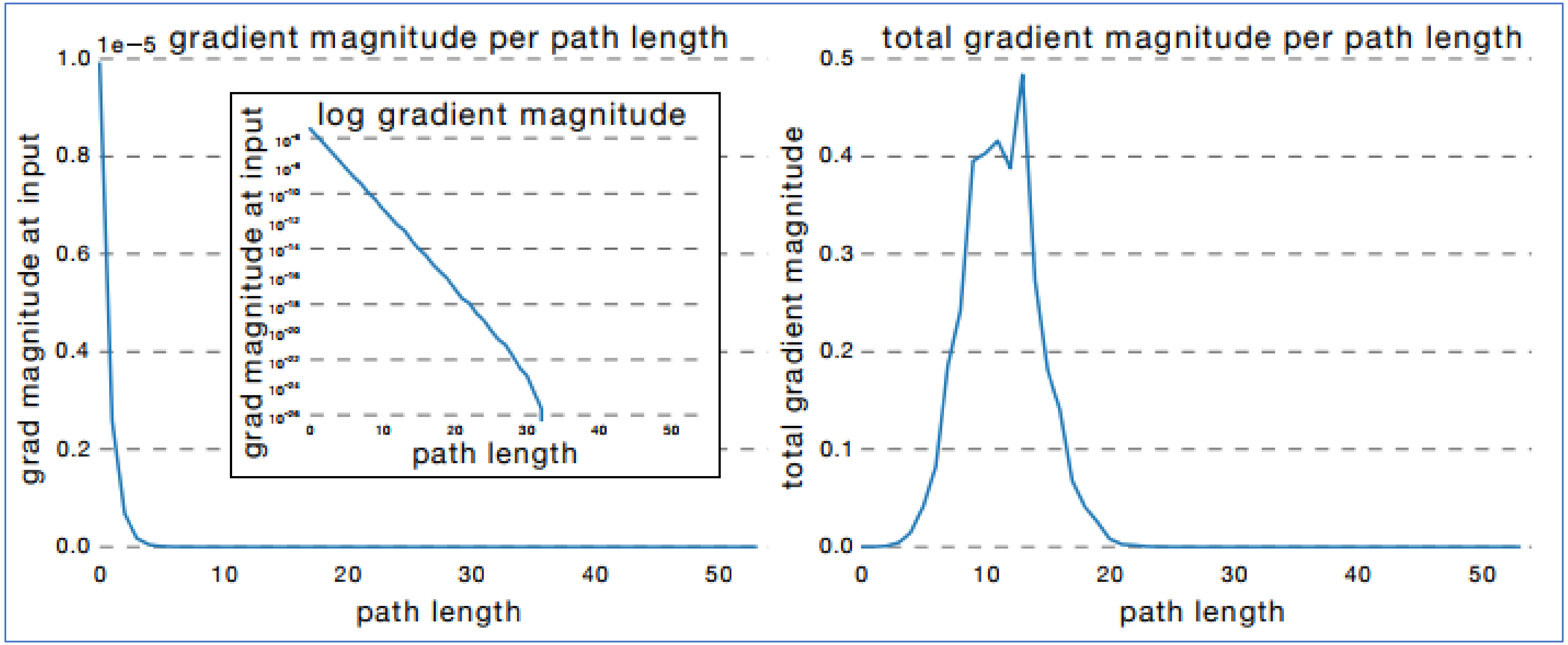
Table 7. Performance on PASCAL VOC 2012 *test* set.

Densely Connected Convolutional Network

- Feature reuse → Parameter Saving
- Alleviate vanishing gradient

Densely Connected Convolutional Network

- Feature reuse → Parameter Saving
- Alleviate vanishing gradient



[Residual Networks Behave Like Ensembles of Relatively Shallow Networks]

Densely Connected Convolutional Network

DenseNet

$x_l = H_l(\ [x_0, x_1, \dots, x_{l-1}]\ )$

ResNet

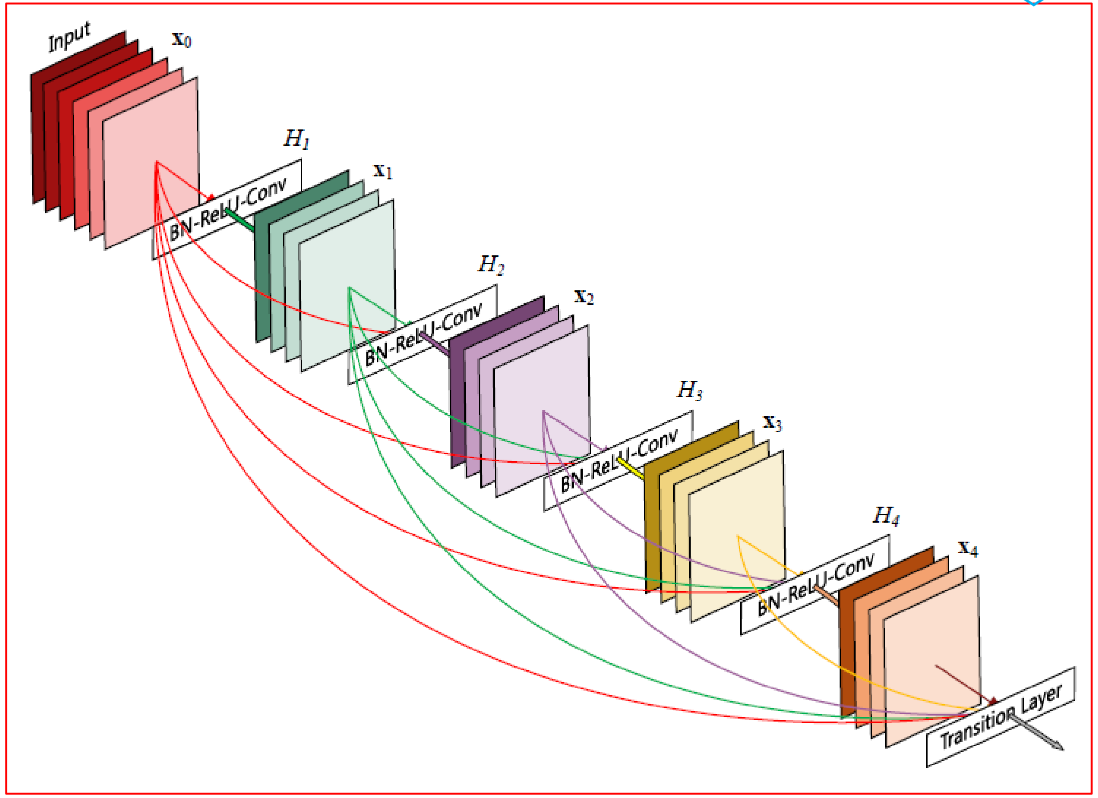
$x_l = H_l(x_{l-1}) + x_{l-1}$

Densely Connected Convolutional Network

DenseNet

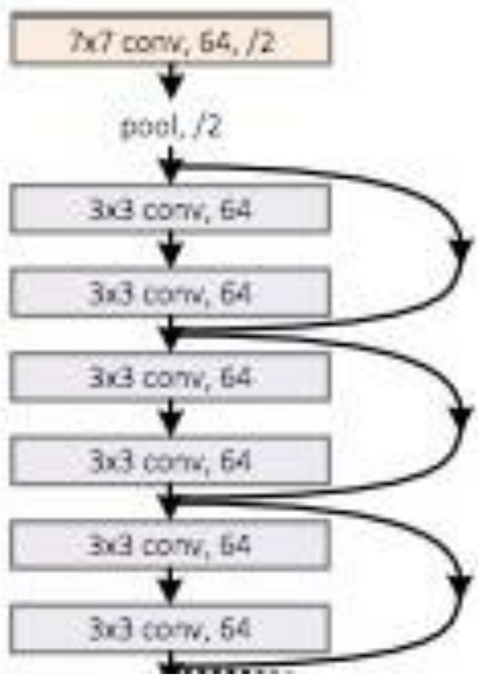
$$x_l = H_l(\,[x_0, x_1, \ldots, x_{l-1}]\,)$$

Dense Block



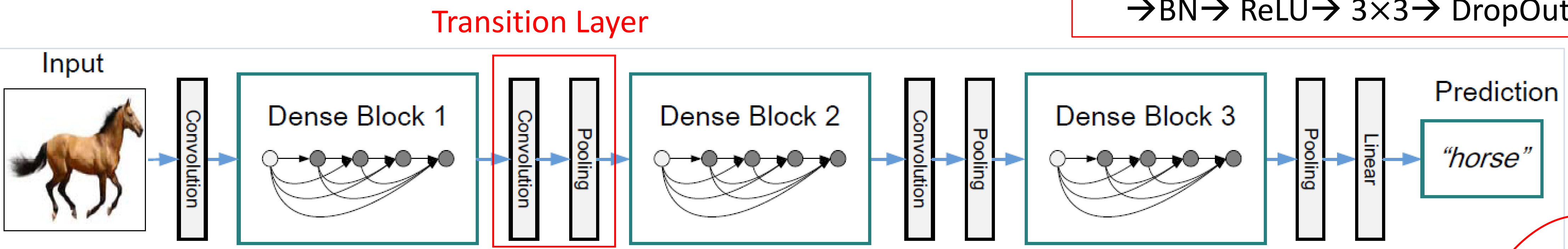
ResNet

$$x_l = H_l(x_{l-1}) + x_{l-1}$$



Framework

→ BN→ ReLU→ 1×1→ DropOut
→BN→ ReLU→ 3×3→ DropOut→



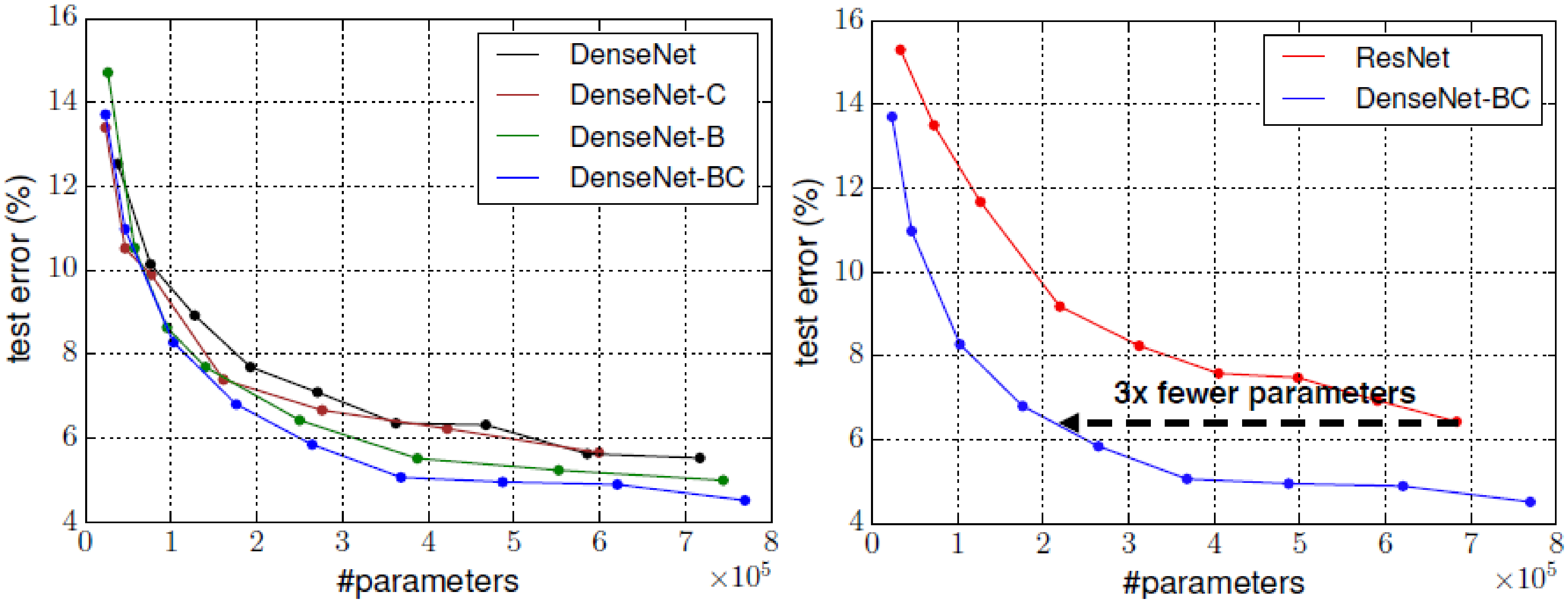
Layers	Output Size	DenseNet-121 $(k = 32)$	DenseNet-169 $(k = 32)$	DenseNet-201 $(k = 32)$	DenseNet-161 $(k = 48)$
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

DenseNet-BC

- DenseNet-B
  - → BN-ReLU-Conv(1×1) → BN-ReLU-Conv(3×3) →
  - Reduce to 4k feature maps

- DenseNet-C
  - Reducing feature maps at transition layers

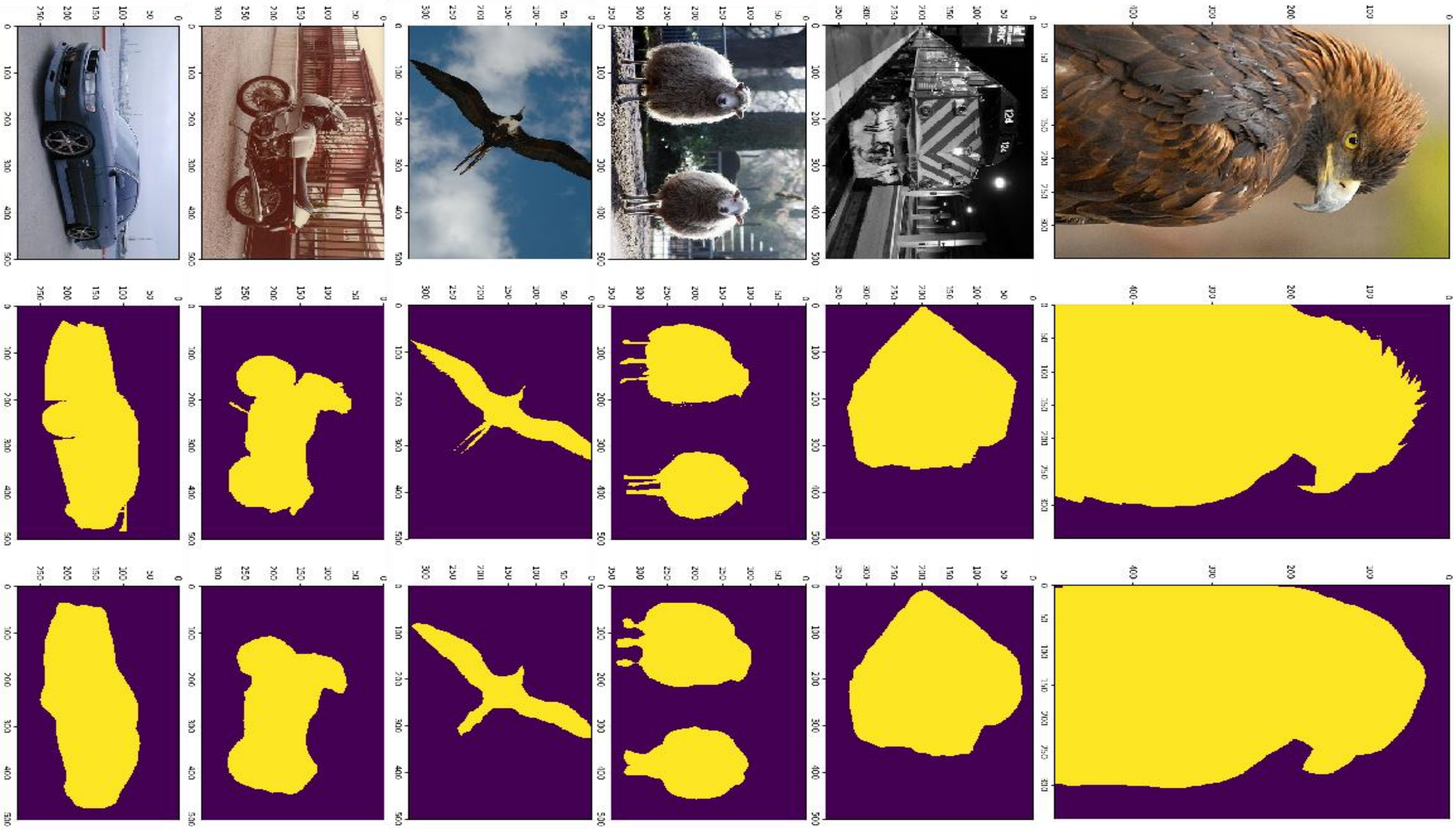
Experimental Result



Experimental Result

Method	Depth	Params	C10	C10+	C100	C100+	SVHN
ResNet [11]	110	1.7M	-	6.61	-	-	-
DenseNet ($k = 12$)	40	1.0M	**7.00**	5.24	**27.55**	24.42	1.79
DenseNet ($k = 12$)	100	7.0M	**5.77**	**4.10**	**23.79**	**20.20**	1.67
DenseNet ($k = 24$)	100	27.2M	**5.83**	**3.74**	**23.42**	**19.25**	**1.59**
DenseNet-BC ($k = 12$)	100	0.8M	**5.92**	4.51	**24.15**	22.27	1.76
DenseNet-BC ($k = 24$)	250	15.3M	**5.19**	**3.62**	**19.64**	**17.60**	1.74
DenseNet-BC ($k = 40$)	190	25.6M	-	**3.46**	-	**17.18**	-

DeepLab V3 for Semantic Image Segmentation



Reference

- ParseNet: Looking wider to see better
- Pyramid Scene Parsing Network
- Dilated Residual Network
- DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs
- Rethinking Atrous Convolution for Semantic Image Segmentation
- Understanding Convolution for Semantic Segmentation
- Spatial pyramid pooling in deep convolutional networks for visual recognition
- Residual Networks Behave Like Ensembles of Relatively Shallow Networks
- Densely Connected Convolutional Networks