arXiv , 2017

Rethinking atrous convolution for semantic image segmentation

Liang-Chieh, Chen George, Papandreou Florian, Schroff Hartwig
Google Inc

- **Problem/Objective**
  - Segmenting objects at multiple scales

- **Contribution/Key Idea**
  - Propose ‘DeepLabv3’ system.
    1. Module with atrous convolution laid out in cascade.
    2. Module with atrous convolution laid out in parallel.

전유진

Rethinking atrous convolution for semantic image segmentation

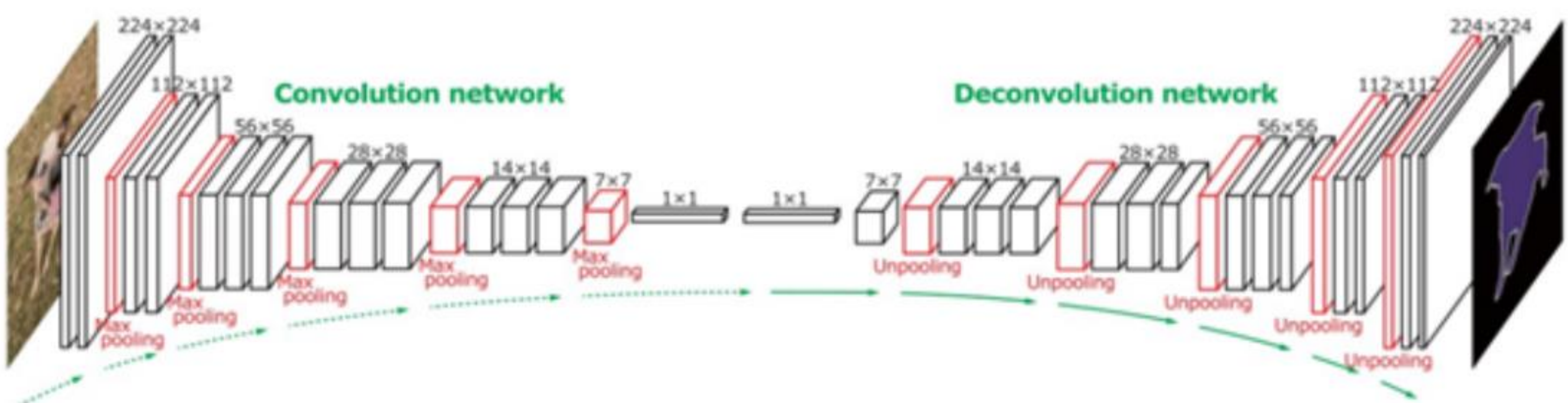
- Atrous convolution 도입 배경.

- Problem.

Reduced feature resolution caused by consecutive pooling, convolution striding operations.

- Solution.

Use of Atrous convolution.



전유진

Rethinking atrous convolution for semantic image segmentation

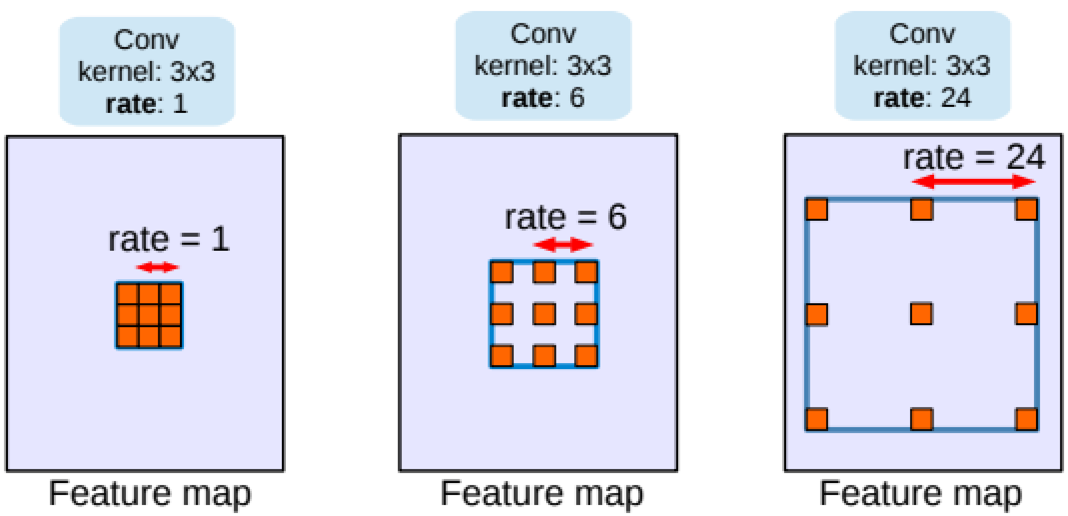
- Atrous convolution이란?

- Atrous convolution

= convolve the input x with upsampled filters produced by inserting r-1 zeros b/w two consecutive filter values along each spatial dimension.

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

i : location.
r : atrous rate.
w : filter.
x : input feature map.
y : output.



전유진

Rethinking atrous convolution for semantic image segmentation

- Modules with Atrous convolution in cascade/parallel 제안 배경.

- Advantages of Atrous convolution

1. Adaptively modify filter's field-of-view by changing the rate value.
2. Explicitly control how densely to compute feature responses in FCN.

→ Be possible to capture multi-scale context by adopting multiple atrous rates.

↓

- Problem.

Existence of objects at multiple scales.

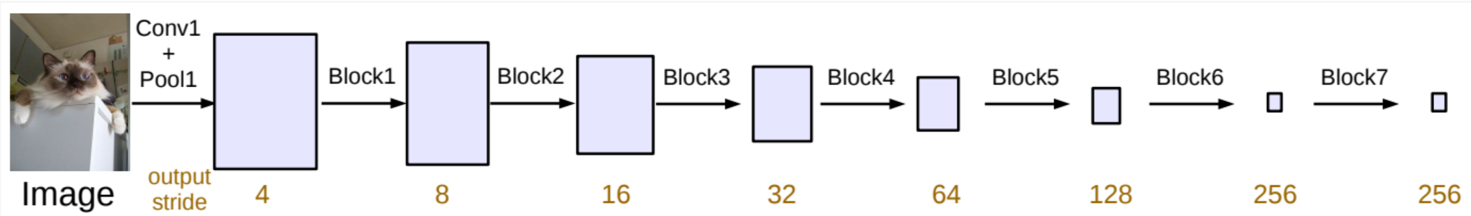
- Solution.

Design modules which employ Atrous convolution in cascade/parallel.

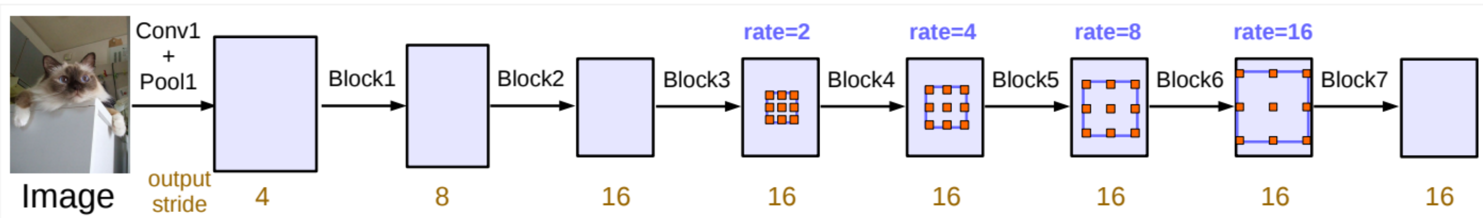
전유진

Rethinking atrous convolution for semantic image segmentation

- Module with Atrous convolution laid out in cascade



(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.

- Problem.

Detail info is decimated due to consecutive striding.

- Solution.

Apply atrous convolution with rates determined by the desired output_stride value.

- Figure3.

Cascade block5, block6, block7 as replicas of block4, adopting different atrous rates.

- Advantage.

Easy to capture long range info in the deeper blocks.

pooling

전유진

- Different atrous rates

Final atrous rate = unit rate * corresponding rate.

$$Multi\_Grid = (r_1, r_2, r_3)$$: unit rates.

When output_stride = 16 & *Multi_Grid* = (1,2,4),
final atrous rates = 2*(1,2,4)=(2,4,8) for block4, 5, 6.

Rethinking atrous convolution for semantic image segmentation

- Module with atrous convolution laid out in parallel : ASPP

- DeepLabv2

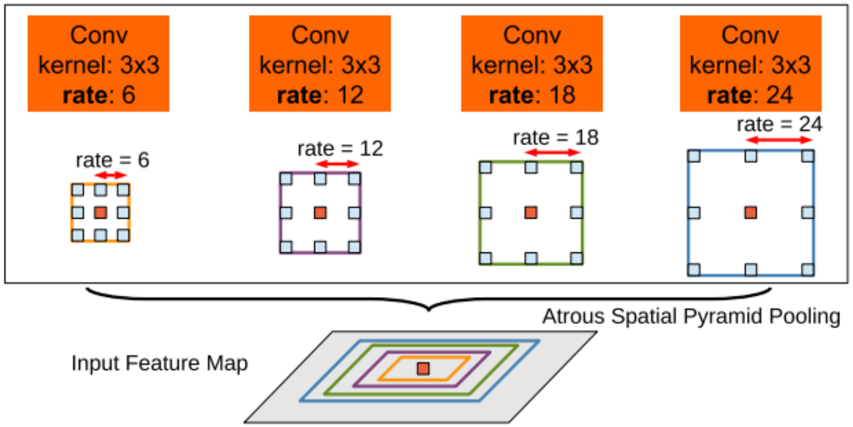


Fig. 4. Atrous Spatial Pyramid Pooling (ASPP). To classify the center pixel (orange), ASPP exploits multi-scale features by employing multiple parallel filters with different rates. The effective Field-Of-Views are shown in different colors.

- DeepLabv3

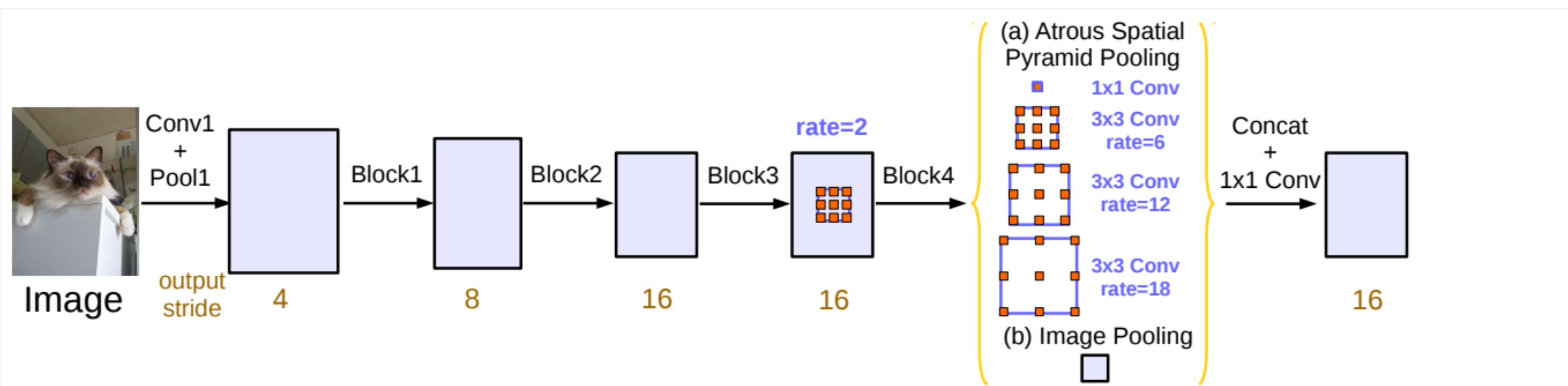


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

전유진

Rethinking atrous convolution for semantic image segmentation

- 2 Improvements from DeepLabv2 to DeepLabv3.

1. Include batch normalization within ASPP.
2. Adopt image-level features.

- 1. Include batch normalization within ASPP.

Batch normalization
= method of normalizing layer inputs to address 'internal covariate shift' problem.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

전유진

2. Adopt image-level features.

- Problem.

As the sampling rate becomes larger, the number of valid filter weights becomes smaller.

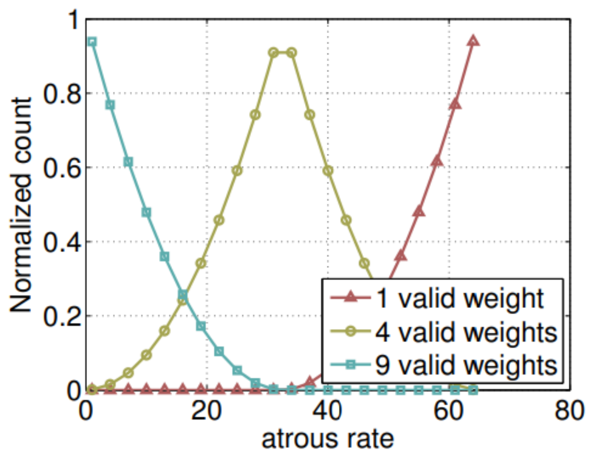


Figure 4. Normalized counts of valid weights with a 3×3 filter on a 65×65 feature map as atrous rate varies. When atrous rate is small, all the 9 filter weights are applied to most of the valid region on feature map, while atrous rate gets larger, the 3×3 filter degenerates to a 1×1 filter since only the center weight is effective.

- Solution.

Apply global average pooling on the last feature map of the model.
Feed the resulting image-level features to 1x1 convolution with 256 filters.
Bilinearly upsample the feature to the desired spatial dimension.

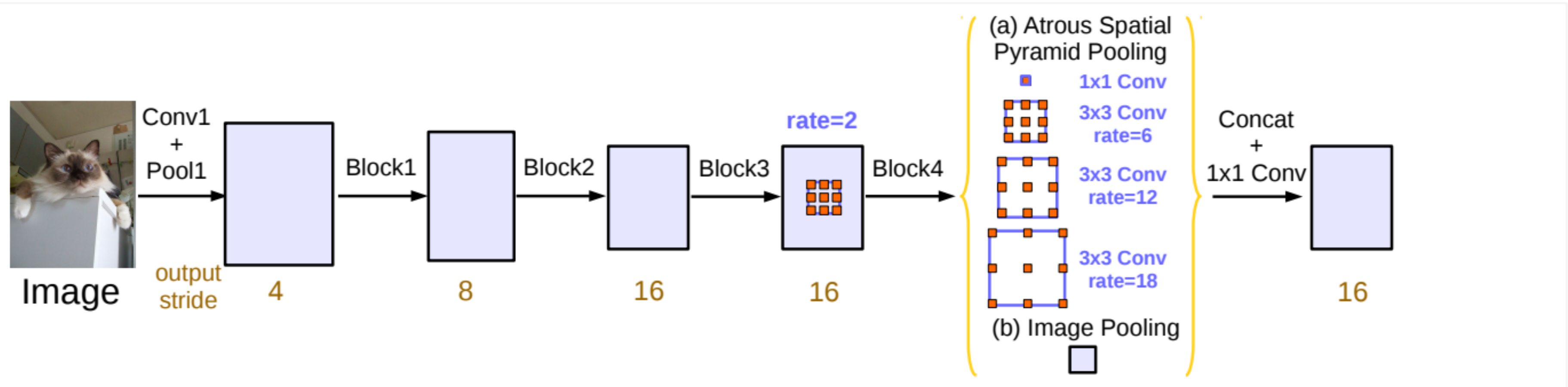


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

Results

- Module with atrous convolution laid out in cascade

output_stride	8	16	32	64	128	256
mIOU	75.18	73.88	70.06	59.99	42.34	20.29

Table 1. Going deeper with atrous convolution when employing ResNet-50 with block7 and different *output_stride*. Adopting *output_stride* = 8 leads to better performance at the cost of more memory usage.

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.

Rethinking atrous convolution for semantic image segmentation

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.

Method	OS=16	OS=8	MS	Flip	mIOU
block7 +	✓				76.66
MG(1, 2, 1)		✓			78.05
		✓	✓		78.93
		✓	✓	✓	79.35

Table 4. Inference strategy on the *val* set. **MG**: Multi-grid. **OS**: *output_stride*. **MS**: Multi-scale inputs during test. **Flip**: Adding left-right flipped inputs.

전유진

- Module with atrous convolution laid out in parallel : ASPP

Multi-Grid			ASPP		Image Pooling	mIOU
(1, 1, 1)	(1, 2, 1)	(1, 2, 4)	(6, 12, 18)	(6, 12, 18, 24)		
✓			✓			75.36
	✓		✓			75.93
		✓	✓			76.58
		✓		✓		76.46
		✓	✓		✓	77.21

Table 5. Atrous Spatial Pyramid Pooling with multi-grid method and image-level features at *output_stride* = 16.

Method	OS=16	OS=8	MS	Flip	COCO	mIOU
MG(1, 2, 4) +	✓					77.21
ASPP(6, 12, 18) +		✓				78.51
Image Pooling		✓	✓			79.45
		✓	✓	✓		79.77
		✓	✓	✓	✓	82.70

Table 6. Inference strategy on the *val* set: **MG**: Multi-grid. **ASPP**: Atrous spatial pyramid pooling. **OS**: *output_stride*. **MS**: Multi-scale inputs during test. **Flip**: Adding left-right flipped inputs. **COCO**: Model pretrained on MS-COCO.

Rethinking atrous convolution for semantic image segmentation

Method	mIOU
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
DeepLabv3-JFT	86.9

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

전유진