[Review] DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution

Taekyung Ki

February 1, 2021

연세대학교 수학계산학부

Outline

Introduction

Contibution

Exploring a backbone for object detector that looks at the images twice or more.

- Recursive Feature Pyramid (RFP)
- Switchable Atrous Convolution (SAC)
- DetectoRS by combining RFP and SAC

Table: A glimpse of the improvements of the box and mask AP by DetectoRS on COCO test-dev.

Method	Backbone	AP_{box}	AP_{mask}	FPS
HTC [4]	ResNet-50	43.6	38.5	4.3
DetectoRS	ResNet-50	**51.3**	**44.4**	**3.9**

Feature Pyramid Network (FPN) [2]

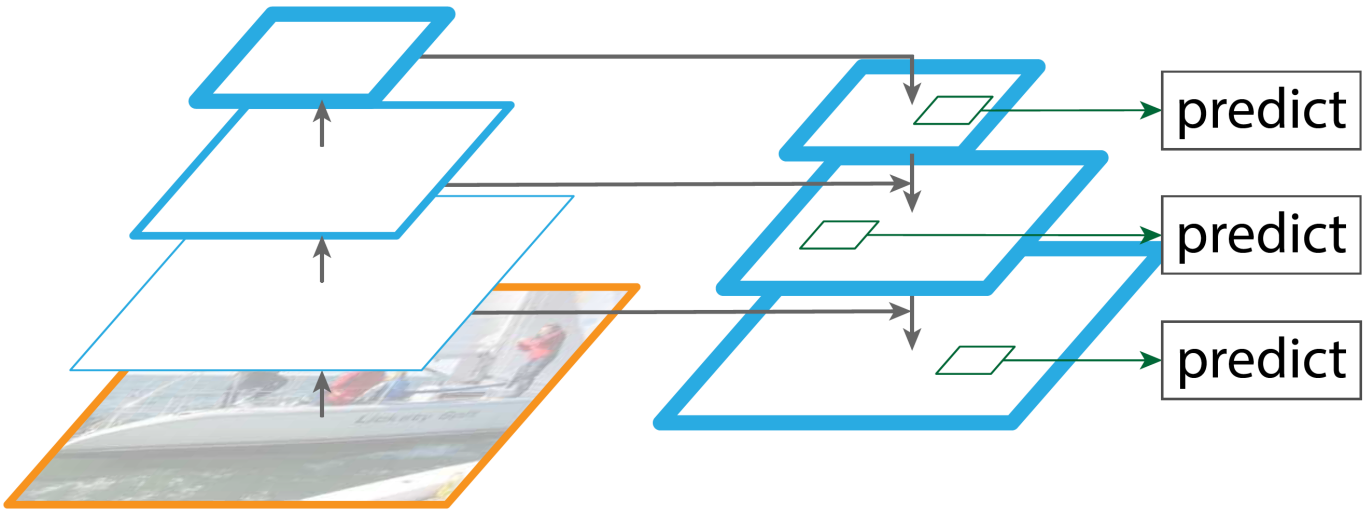


Figure: Feature pyramid network (FPN) [2].

Recursive formula for FPN

Let $\mathbf{B}_i$ denote the i-th stage of the bottom-up backbone, and $\mathbf{F}_i$ denote the i-th top-down FPN operation. The backbone equipped with FPN outputs a set of feature maps $\{\mathbf{f}_i | i = 1, \cdots, S\}$, where S is the number of the stages.

$$\forall i = 1, \cdots, S, \quad \mathbf{f}_i = \mathbf{F}_i(\mathbf{f}_{i+1}, \mathbf{x}_i) \quad \text{and} \quad \mathbf{x}_i = \mathbf{B}_i(\mathbf{x}_{i-1}).$$

Feature Pyramid Network (FPN) [2]

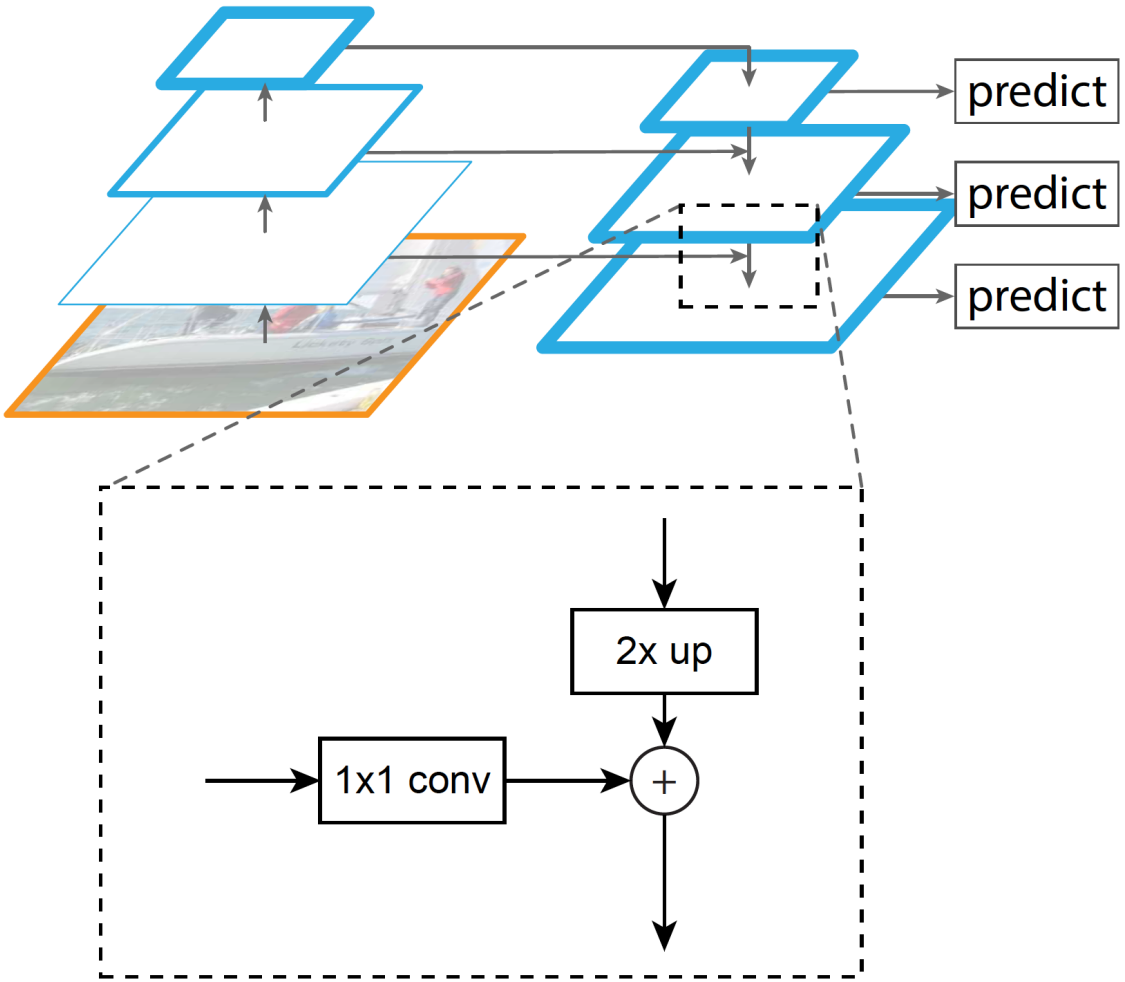


Figure: Feature pyramid network (FPN). A building block illustrating the lateral connection and the top-down pathway, merged by addition [2].

Atrous Convolution

Atrous Convolution

Consider one-dimensional case. The output $\mathbf{y}[i]$ of atrous convolution of a 1-D input $\mathbf{x}[i]$ with a filter $\mathbf{w}[k]$ of length K is defined as:

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

where r is *atrous rate* parameter.

- Note that if the original kernel size is $k \times k$ and with the rate parameter r, the kernel of atrous convolution is $k + (k-1)(r-1)$.

Atrous Convolution

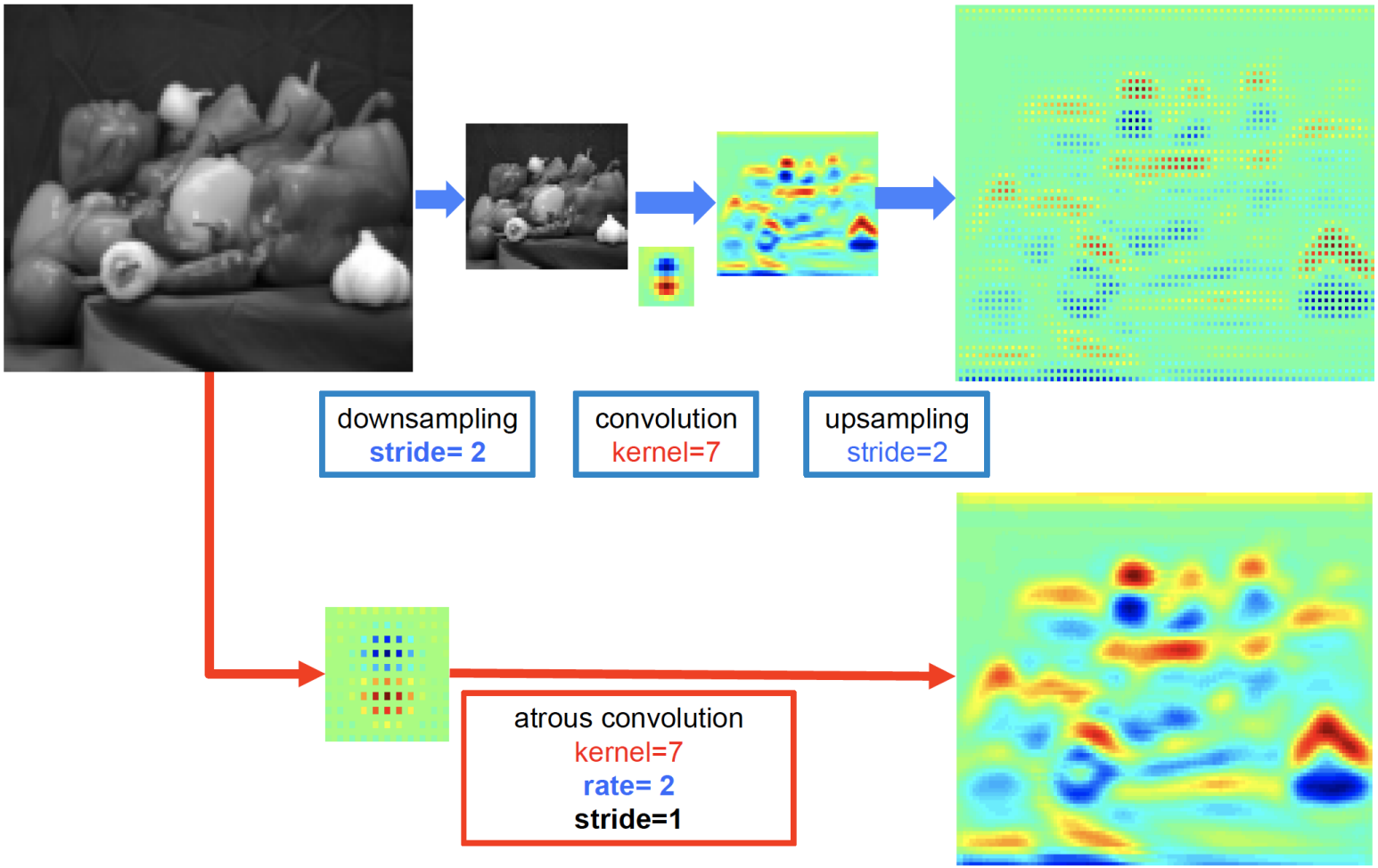


Figure: Illustration of Atrous Convolution in 2-D. The kernel here is the vertical Gaussian derivative [3].

Atrous Spatial Pyramid Pooling (ASPP) [3]

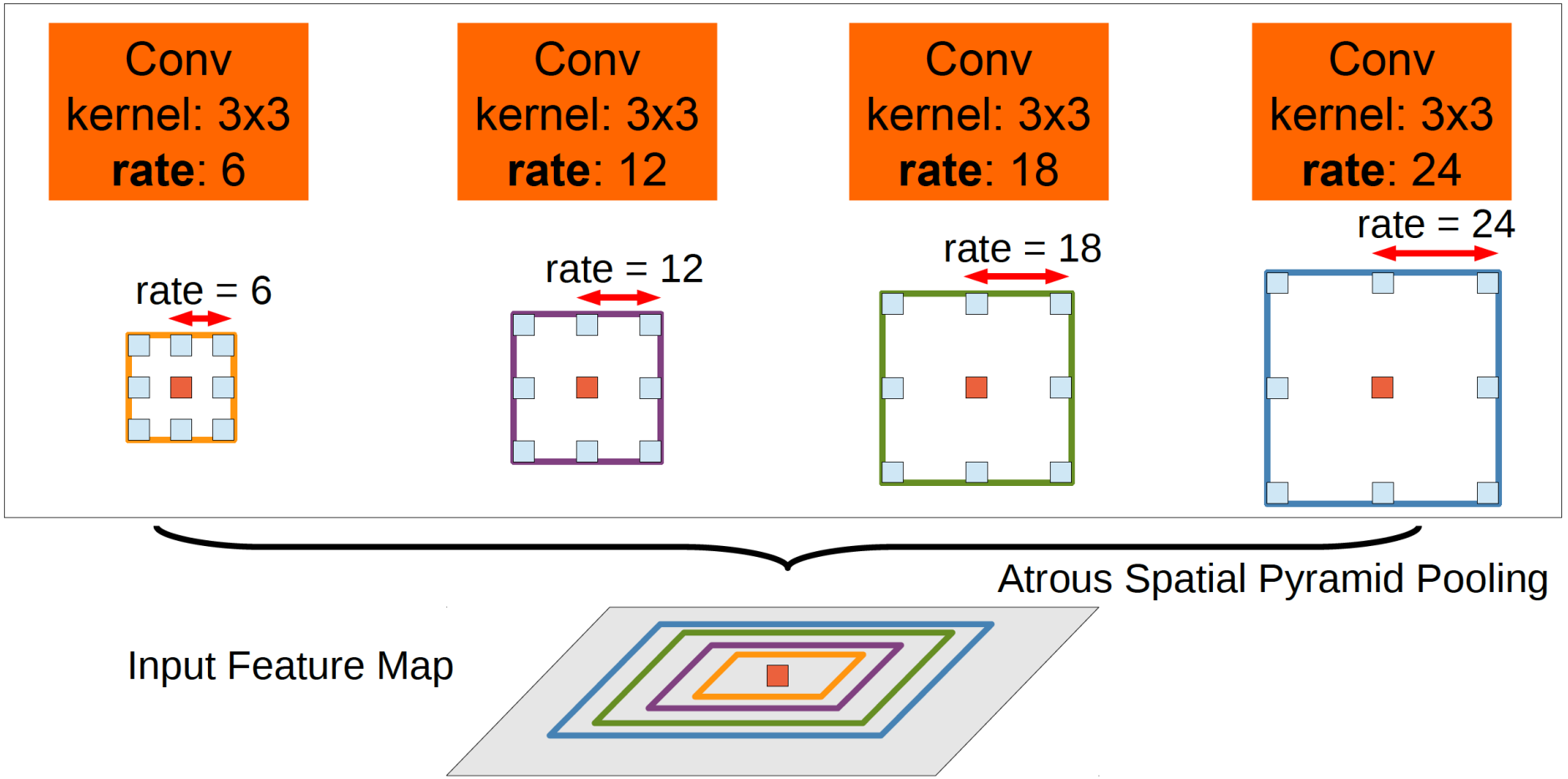


Figure: Illustration of ASPP in 2-D [3].

Recursive Feature Pyramid (RFP)

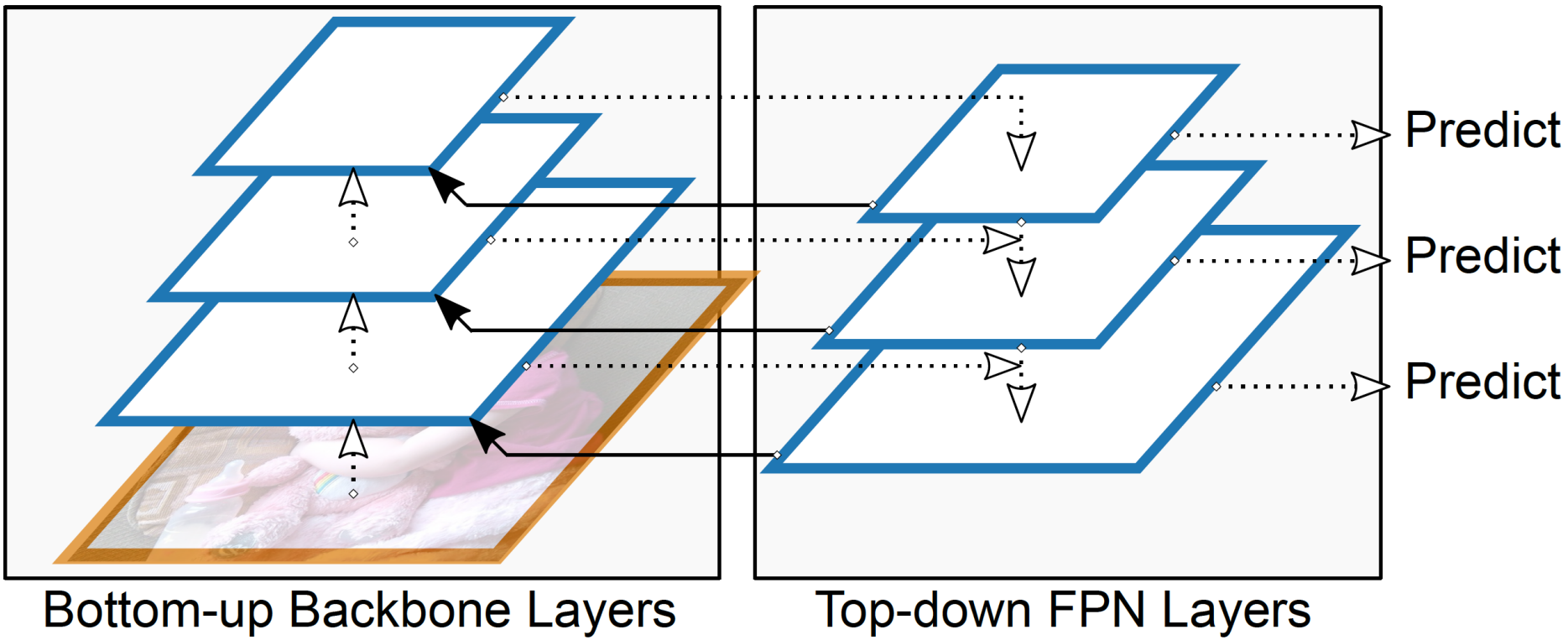


Figure: Recursive feature pyramid (RFP) which incorporates feedback connections into FPN.

Recursive Feature Pyramid (RFP)

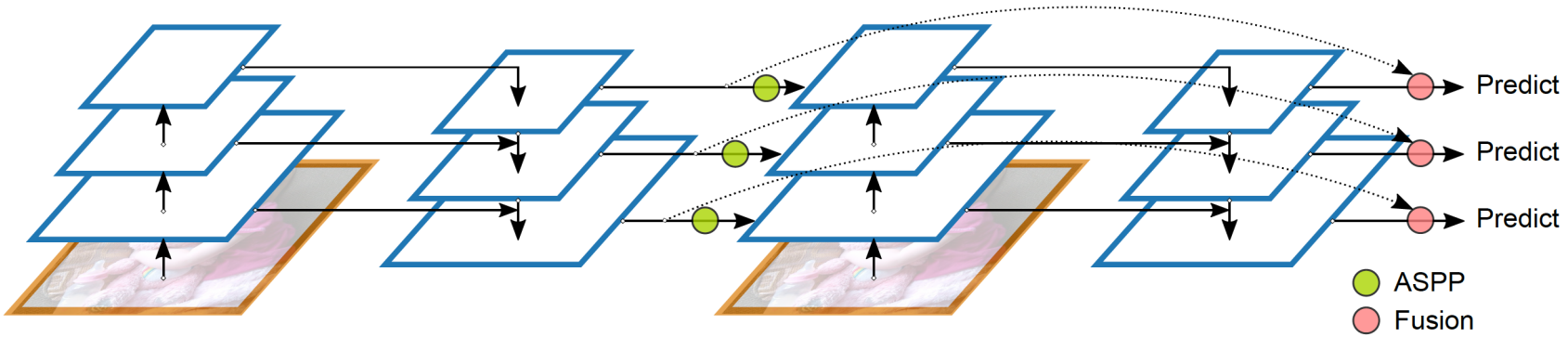


Figure: Recursive feature pyramid unrolled to a 2-step sequential network.

Recursive Formula for RFP

Let $\mathbf{R}_i$ denote the feature transformers before connecting them back to the bottom-up backbone. Then $\forall i = 1, \cdots, S$, $\forall t = 1, \cdots, T$,

$$\mathbf{f}_i^t = \mathbf{F}_i^t(\mathbf{f}_{i+1}^t, \mathbf{x}_i^t), \quad \mathbf{x}_i^t = \mathbf{B}_i^t(\mathbf{x}_{i-1}^t, \mathbf{R}_i^t(\mathbf{f}_i^{t-1})),$$

where T is the number of unrolled iterations, and we use superscript t to denote operations and features at the step t.

Recursive Feature Pyramid (RFP)

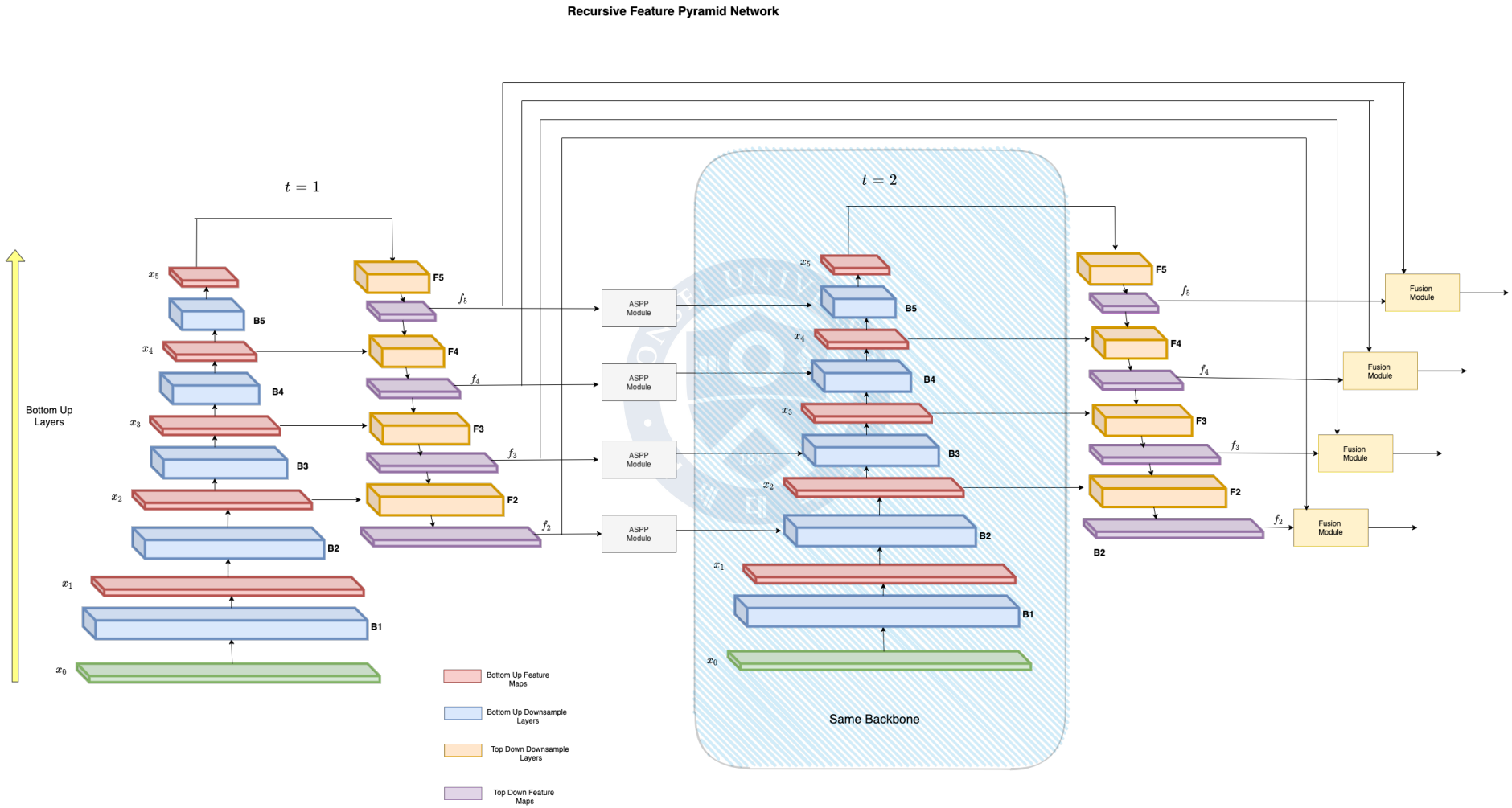


Figure: Recursive feature pyramid unrolled to a 2-step sequential network (detail).

ASPP as the Connecting Module

Connecting Module

- Authors make changes to the ResNet backbone **B** to allow it to take both **x** and **R**(**f**) as its input.
- They only make changes to the first block of each ResNet stage by combining RFP features.

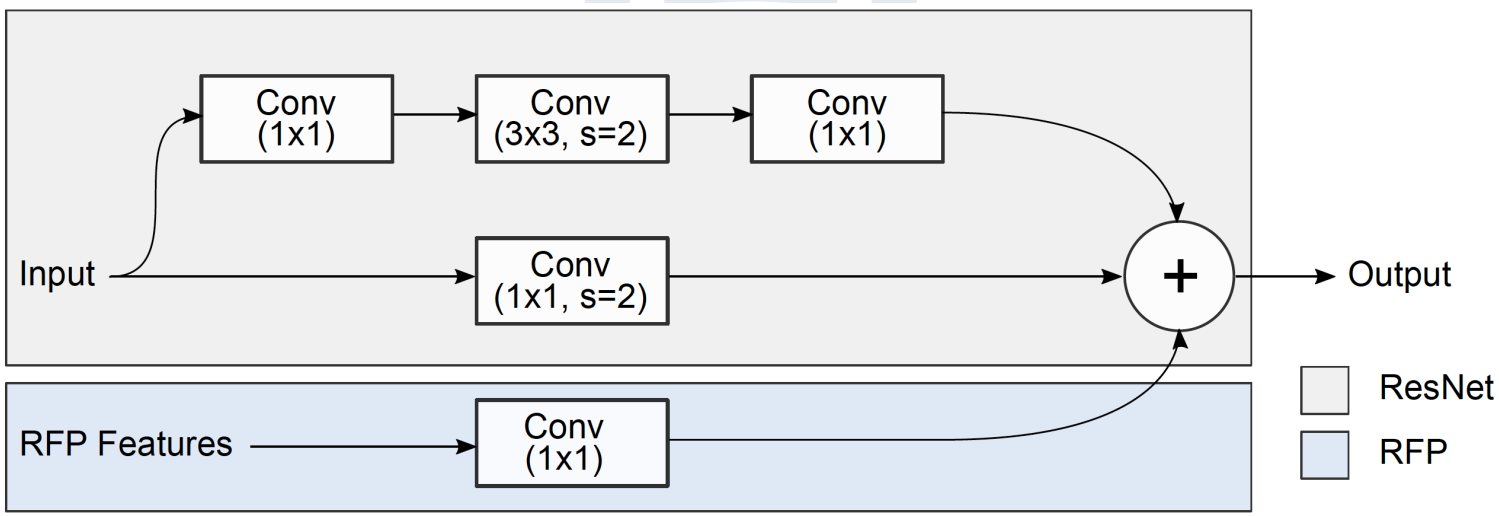


Figure: ASPP as the connecting module.

ASPP as the Connecting Module

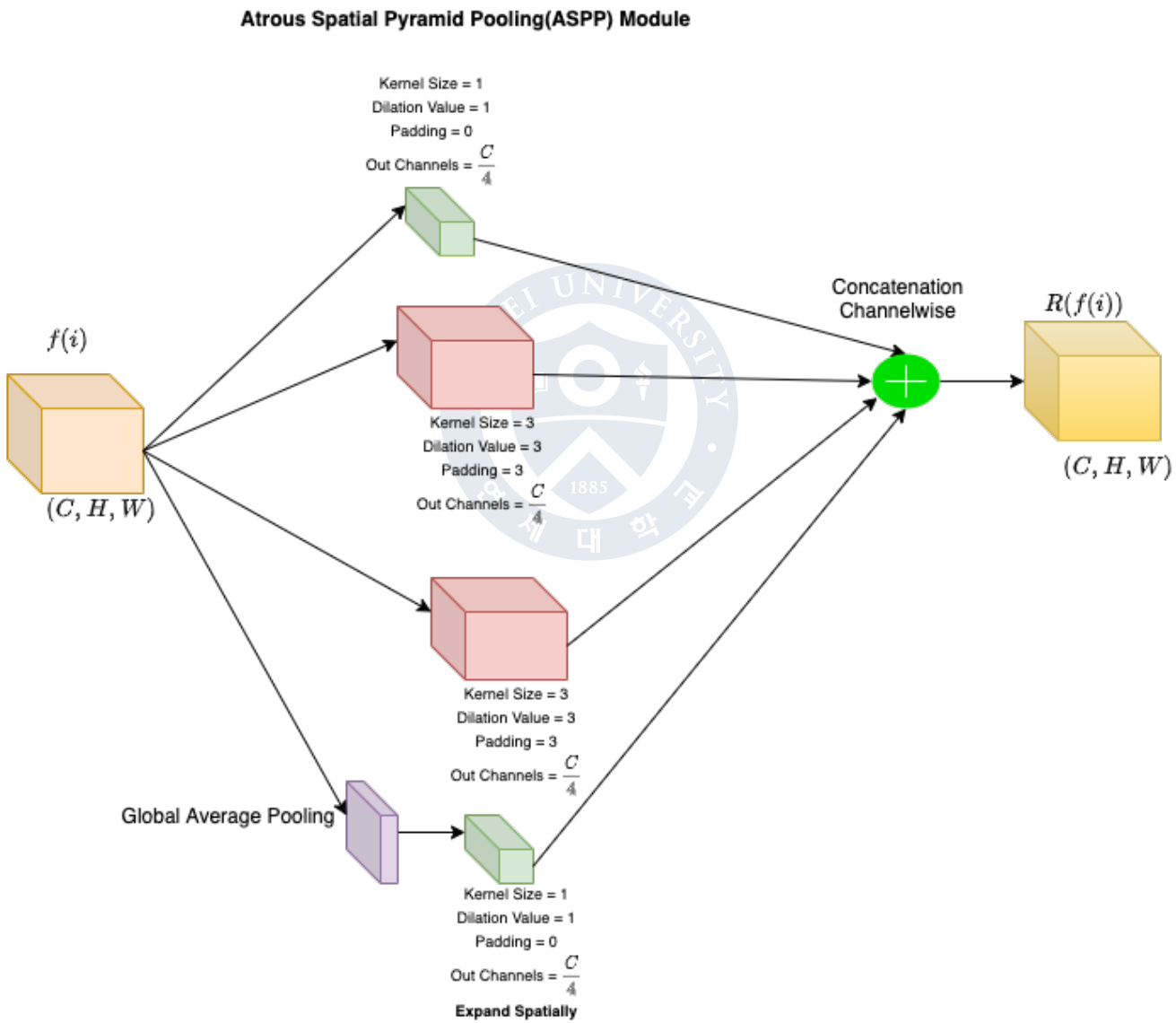
Connecting Module

- Authors make changes to the ResNet backbone **B** to allow it to take both **x** and **R**(**f**) as its input.
- They only make changes to the first block of each ResNet stage by combining RFP features.

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

Figure: Structure summary of ResNet [5].

ASPP as the Connecting Module [6]



Fusion Module

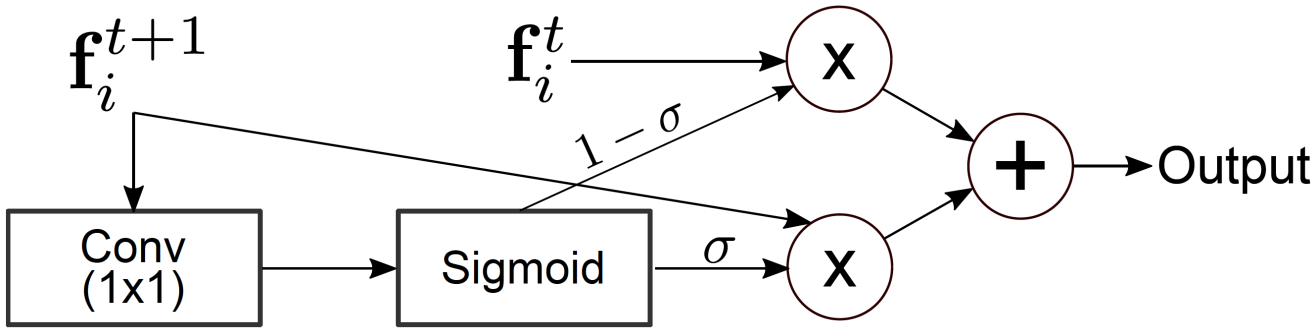
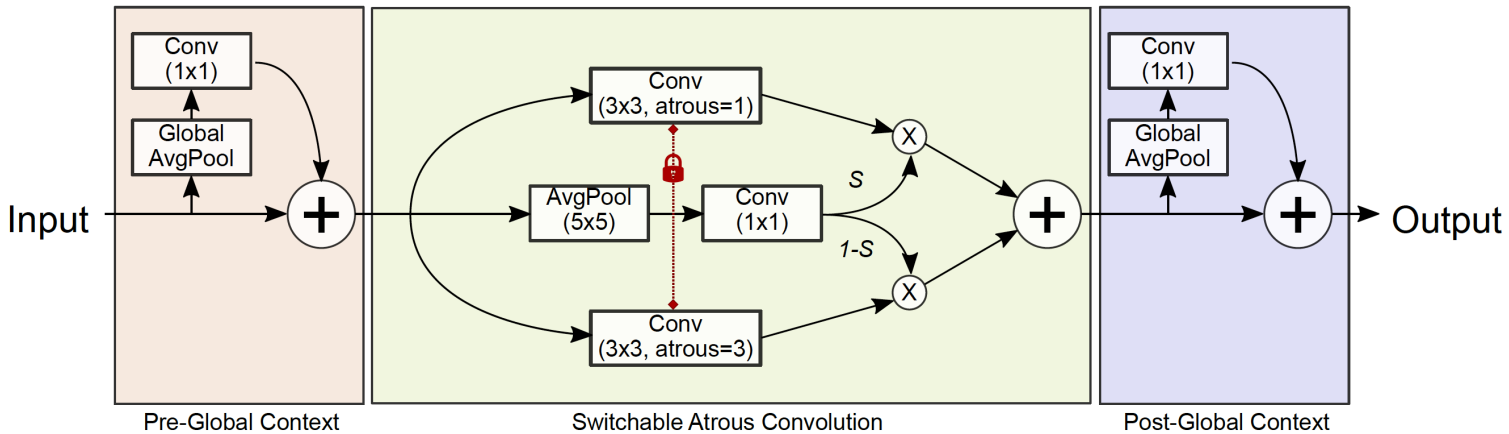


Figure: Fusion module in RFP.

Fusion Module

- FRP uses a fusion module to combine $\mathbf{f}_i^t$ and $\mathbf{f}_i^{t+1}$ to update the values of $\mathbf{f}_i$ at the unrolled stage $t+1$.
- The fusion module uses the feature $\mathbf{f}_i^{t+1}$ to compute an attention map by a convolutional layer follwed by a Sigmoid operation.
- The resulting attention map is used to compute the weighted sum of $\mathbf{f}_i^t$ and $\mathbf{f}_i^{t+1}$ to form an update $\mathbf{f}_i$.

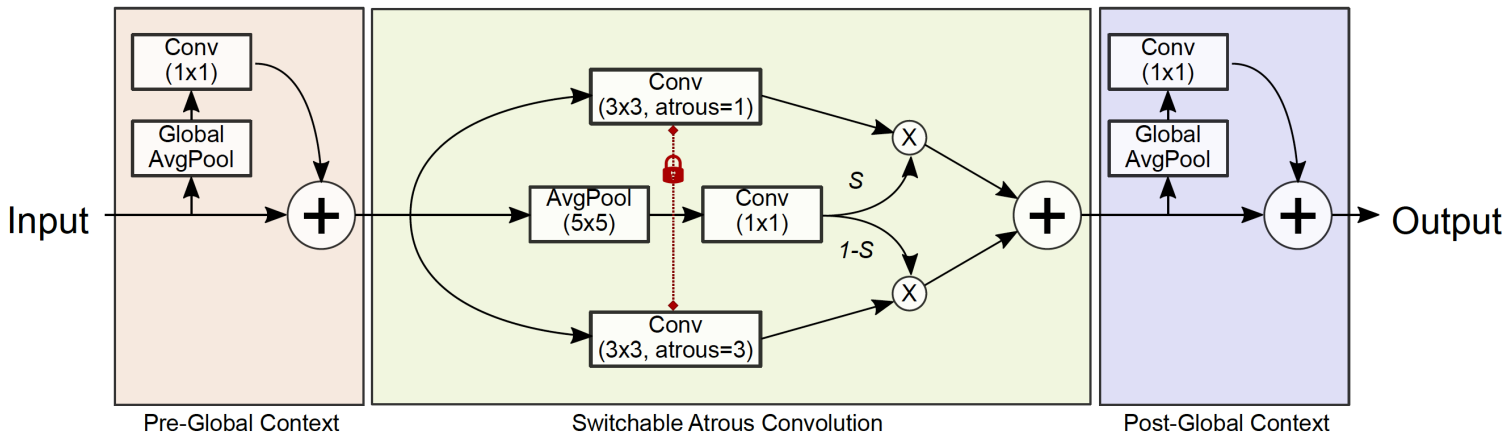
Switchable Atrous Convolution (SAC)



Switchable Atrous Convolution (SAC)

- Authors convert every 3×3 convolutional layer in the backbone to SAC.
- Two global context modules are added before and after the SAC component.

Switchable Atrous Convolution (SAC)

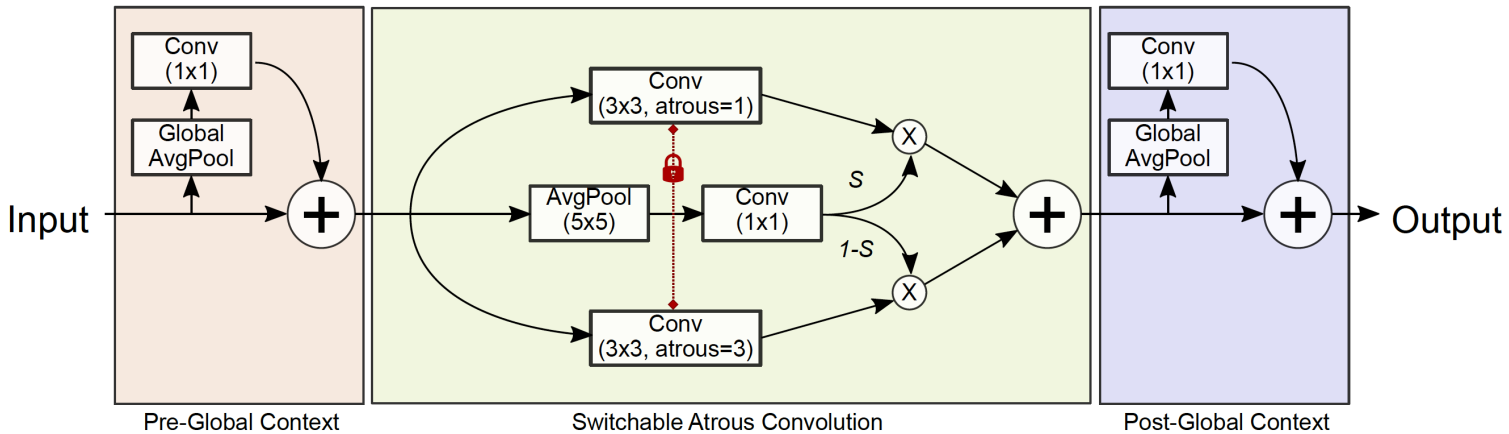


Switchable Atrous Convolution (SAC)

$$\mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1) \xrightarrow[\text{SAC}]{\text{Convert}} \mathbf{S}(\mathbf{x}) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1) + (1 - \mathbf{S}(\mathbf{x})) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w} + \triangle\mathbf{w}, r),$$

where $\mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1)$ be the convolutional operation with weight $\mathbf{w}$ and atrous rate r which takes $\mathbf{x}$ as its input and output $\mathbf{y}$.

Switchable Atrous Convolution (SAC)



Switchable Atrous Convolution (SAC)

$$\mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1) \xrightarrow[\text{SAC}]{\text{Convert}} \mathbf{S}(\mathbf{x}) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w}, 1) + (1 - \mathbf{S}(\mathbf{x})) \cdot \mathbf{Conv}(\mathbf{x}, \mathbf{w} + \triangle\mathbf{w}, r),$$

where r is a hyper-parameter of SAC, $\triangle\mathbf{w}$ is a trainable weight, and the switch function $\mathbf{S}(\cdot)$ is implemented as an average pooling layer with a 5×5 kernel followed by a 1×1 convolutional layer.

Conclusions

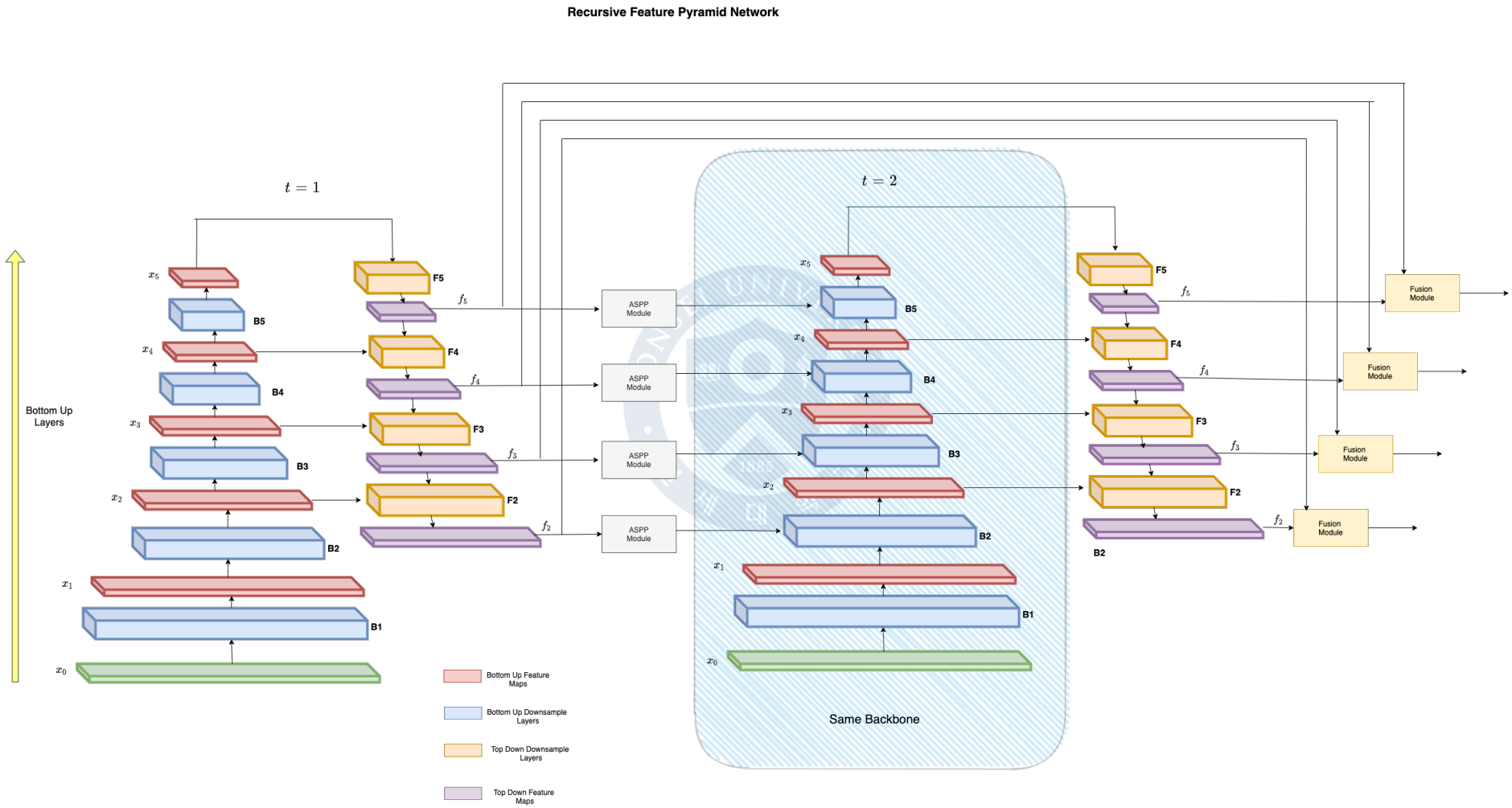


Figure: Recursive feature pyramid unrolled to a 2-step sequential network (detail) [6].

Conclusions

Conclusions

- RFP implements **thinking twice at the macro level**, where the outputs of FPN are brought back to each stage of the bottom-up backbone through feedback connections.
- SAC instantiates **looking twice at the micro level**, where the inputs are convolved with two different atrous rates.

References

[1] Siyuan Qiao and Liang-Chieh Chen and Alan Yuille, *DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution*, preprint, 2020, arXiv preprint arXiv:2006.02334

[2] Tsung-Yi Lin and Piotr Dollár and Ross Girshick and Kaiming He and Bharath Hariharan and Serge Belongie, *Feature Pyramid Networks for Object Detection*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2117–2125, 2017

[3] Liang-Chieh Chen and George Papandreou and Iasonas Kokkinos and Kevin Murphy and Alan L. Yuille, *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 40, 4, 834–848, 2017

[4] Kai Chen and Jiangmiao Pang and Jiaqi Wang and Yu Xiong and Xiaoxiao Li and Shuyang Sun and Wansen Feng and Ziwei Liu and Jianping Shi and Wanli Ouyang and others, *Hybrid Task Cascade for Instance Segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4974–4983, 2019

[5] Kaiming He and Xiangyu Zhang and Shaoquing Ren and Jian Sun, *Deep Residual Learning for Image Recognition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778, 2016

[6] https://medium.com/visionwizard/detectors-state-of-the-art-object-detector-from-google-research-e0b89abdd1fc