PSConv: Squeezing Feature Pyramid into One Compact Poly-Scale Convolutional Layer

Duo Li[1,2], Anbang Yao[2(✉)], and Qifeng Chen[1(✉)]

[1] The Hong Kong University of Science and Technology, Kowloon, Hong Kong
duo.li@connect.ust.hk, cqf@ust.hk
[2] Intel Labs China, Beijing, China
anbang.yao@intel.com

Abstract. Despite their strong modeling capacities, Convolutional Neural Networks (CNNs) are often scale-sensitive. For enhancing the robustness of CNNs to scale variance, multi-scale feature fusion from different layers or filters attracts great attention among existing solutions, while the more granular kernel space is overlooked. We bridge this regret by exploiting multi-scale features in a finer granularity. The proposed convolution operation, named Poly-Scale Convolution (PSConv), mixes up a spectrum of dilation rates and tactfully allocates them in the individual convolutional kernels of each filter regarding a single convolutional layer. Specifically, dilation rates vary cyclically along the axes of input and output channels of the filters, aggregating features over a wide range of scales in a neat style. PSConv could be a drop-in replacement of the vanilla convolution in many prevailing CNN backbones, allowing better representation learning without introducing additional parameters and computational complexities. Comprehensive experiments on the ImageNet and MS COCO benchmarks validate the superior performance of PSConv. Code and models are available at https://github.com/d-li14/PSConv.

Keywords: Convolutional kernel · Multi-scale feature fusion · Dilated convolution · Categorization and detection

1 Introduction

With the booming development of CNNs, dramatic progress has been made in the field of computer vision. As an inherent feature extraction mechanism, CNNs naturally learn coarse-to-fine hierarchical image representations. To mimic human visual systems that could process instances and stuff concurrently, it is of

D. Li—Indicates intern at Intel Labs China.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58589-1_37) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12366, pp. 615–632, 2020.
https://doi.org/10.1007/978-3-030-58589-1_37

vital importance for CNNs to gather diverse information from objects of various sizes and understand meaningful contextual backgrounds. However, streamlined CNNs usually have fixed-sized receptive fields, lacking the ability to tackle this kind of issue. Such a deficiency restricts their performance on visual recognition tasks, especially scale-sensitive dense prediction problems. The advent of FCN [25] and Inception [35] demonstrates the privilege of multi-scale representation to perceive heterogeneous receptive fields with impressive performance improvement. Motivated by these pioneering works, follow-up approaches explore and upgrade multi-scale feature fusion with more intricate skip connections or parallel streams. However, we notice that most of the existing works capture these informative multi-scale features in a layer-wise or filter-wise style, laying emphasis on the architecture engineering of the entire network or their composed building blocks.

From a brand new perspective, we shift the focus of design from macro- to micro-architecture towards the target of easily exploiting multi-scale features without touching the overall network architecture. Expanding kernel sizes and extending the sampling window sizes via increasing dilation rates are two popular techniques to enlarge the receptive fields inside one convolution operation. Compared to large kernels that bring about more parameter storage and computational consumption, dilated convolution is an alternative to cope with objects in an array of scales without introducing extra computational complexities. In this paper, we present Poly-Scale Convolution (PSConv), a novel convolution operation, extracting multi-scale features from the more granular convolutional kernel space. PSConv respects two design principles: firstly, regarding one single convolutional filter, its constituent kernels use a group of dilation rates to extract features corresponding to different receptive fields; secondly, regarding all convolutional filters in one single layer, the group of dilation rates corresponding to each convolutional filter alternates along the axes of input and output channels in a cyclic fashion, extracting diverse scale information from the incoming features and mapping them into outgoing features in a wide range of scales. Through these atomic operations on individual convolutional kernels, we effectively dissolve the aforementioned deficiency of standard convolution and push the multi-scale feature fusion process to a much more granular level. This proposed approach tiles the *kernel lattice*[1] with hierarchically stacked pyramidal features defined in the previous methodologies [22]. Each specific feature scale in one pyramid layer can be grasped with a collection of convolutional kernels in a PSConv operation with the same corresponding dilation rate, thus the whole feature pyramid can be represented in a condensed fashion using one compact PSConv layer with a spectrum of dilation rates. *Poly-Scale* Convolution extends the conventional *mono-scale* convolution living on a homogeneous dilation space of kernel lattice, hence the name of this convolution form. In our PSConv, scale-aware features located in different channels collaborate as a unity to deal with

[1] *kernel lattice* refers to the two-dimensional flattened view of convolutional filters where the kernel space is reduced while the channel space is retained, thus each cell in the lattice represents an individual kernel (see Fig. 2 for intuitive illustration).

scale variance problems, which is critical for handling a single instance with a non-rigid shape or multiple instances with complex scale variations. For scale-variant stimuli, PSConv is capable of learning self-adaptive attention for different receptive fields following a dynamic routing mechanism, improving the representation ability without any additional parameters or memory cost.

Thanks to its plug-and-play characteristic, our PSConv can be readily used to replace the vanilla convolution of arbitrary state-of-the-art CNN architectures, *e.g.*, ResNet [11], giving rise to PS-ResNet. We also build PS-ResNeXt featuring group convolutions to prove the universality of PSConv. These models are comprehensively evaluated on the ImageNet [7] dataset and show consistent gains over the baseline of plain CNN counterparts. More experiments on (semi-)dense prediction tasks, *e.g.*, object detection and instance segmentation on the MS COCO dataset, further demonstrate the superiority of our proposed PSConv over the standard ones under the circumstances with severe scale variations. It should be noted that PSConv is also independent of other macro-architectural choices and thus orthogonal and complementary to existing multi-scale network designs at a coarser granularity, leaving extra room to combine them together for further performance enhancement.

Our core contributions are summarized as follows:

- ❏ We extend the scope of the conventional mono-scale convolution operation by developing our Poly-Scale Convolution, which effectively and efficiently aggregates multi-scale features via arranging a spectrum of dilation rates in a cyclic manner inside the kernel lattice.
- ❏ We investigate the multi-scale network design through the lens of kernel engineering instead of network engineering, which avoids the necessity of tuning network structure or layer configurations while achieves competitive performance, when adapted to existing CNN architectures.

2 Related Work

We briefly review previous relevant network and modular designs and clarify their similarities and differences compared to our proposed approach.

Multi-scale Network Design. Early works like AlexNet [18] and VGGNet [32] learn multi-scale features in a data-driven manner, which are naturally equipped with a hierarchical representation by the inherent design of CNNs. The shallow layers seek finer structures in the images like edges, corners, and texture, while deep layers abstract semantic information, such as outlines and categories. In order to break the limitation of fixed-sized receptive fields and enhance feature representation, many subsequent works based on explicit multi-scale feature fusion are presented. Within this scope, there exists a rich literature making innovations on **skip connection** and **parallel stream**.

The **skip connection** structure exploits features with multi-size receptive fields from network layers at different depths. The representative FCN [25] adds up feature maps from multiple intermediate layers with the skip connection.

Analogous techniques have also been applied to the field of edge detection, presented by HED [40]. In the prevalent encoder-decoder architecture, the decoder network could be a symmetric version of the encoder network, with skip connections over some mirrored layers [26] or concatenation of feature maps [31]. DLA [42] extends the peer-to-peer skip connections into a tree structure, aggregating features from different layers in an iterative and hierarchical style. FishNet [34] stacks an upsampling body and a downsampling head upon the backbone tail, refining features that compound multiple resolutions.

The **parallel stream** structure generates multi-branch features conditioned on a spectrum of receptive fields. Though too numerous to list in full, recent research efforts often attack conventional designs via either maintaining a feature pyramid virtually from bottom to top or repeatedly stacking split-transform-merge building blocks. The former pathway of design includes several exemplars like Multigrid [16] and HRNet [33], which operate on a stack of features with different resolutions in each layer. Similarly, Octave Convolution [5] decomposes the standard convolution into two resolutions to process features at different frequencies, removing spatial redundancy by separating scales. The latter pathway of design is more crowded with the following works. The Inception [13,35,36] family utilizes parallel pathways with various kernel sizes in one Inception block. BL-Net [2] is composed of branches with different computational complexities, where the features at the larger scale pass through fewer blocks to spare computational resources and the features from different branches at distinct scales are merged with a linear combination. Res2Net [8] and OSNet [45] construct a group of hierarchical residual-like connections or stacked Lite 3×3 layers along the channel axis in one single residual block. ELASTIC [38] and ScaleNet [20] learn a soft scaling policy to allocate weights for different resolutions in the paratactic branches. Despite distinct with respect to detailed designs, these works all extensively use down-sampling or up-sampling to resize the features to 2^n times and inevitably adjust the original architecture via the selection of new hyperparameters and layer configurations when plugged in. On the contrary, our proposed PSConv can be a straightforwardly drop-in replacement of the vanilla convolution, leading a trend towards more effective and efficient multi-scale feature representation. Conventionally, features with multi-size receptive fields are integrated via channel concatenation, weighted summation or attention models. In stark contrast, we suggest to explore multi-scale features in a finer granularity, encompassed in merely one single convolutional layer.

In addition to the aforementioned networks designed to enhance image classification, scale variance poses more challenges in (semi-)dense prediction tasks, *e.g.*, object detection and semantic segmentation. Faster R-CNN [9] uses predefined anchor boxes of different sizes to address this issue. DetNet [21], RFBNet [24] and TridentNet [19] apply dilated convolutions to enlarge the receptive fields. DeepLab [4] and PSPNet [44] construct feature pyramid in a parallel fashion. FPN [22] is designed to fuse features at multiple resolutions through top-down and lateral connections and provides anchors specific to different scales.

Dynamic Convolution. All approaches above process multi-scale information without drilling down into the pure single convolutional layer. Complementarily, another line of research concentrates on injecting scale modules into the original network directly and handling various receptive fields in an automated fashion. STN [14] explicitly learns a parametric manipulation of the feature map conditioned on itself to improve the tolerance to spatial geometric transformations. ACU [15] and DCN [6,46] learn offsets at each sampling position of the convolutional kernel or the feature map to permit a flexible shape deformation during the convolution process. SAC [43] inserts an extra regression layer to densely infer the scale coefficient map and applies an adaptive dilation rate to the convolutional kernel at each spatial location of the feature map. POD [28] predicts a globally continuous scale and then converts the learned fractional scale to a channel-wise combination of integer scales for fast deployment. We respect the succinctness of these plugged-in modules and follow these approaches in their form. In this spirit, we formulate a novel convolution representation through cyclically alternating dilation rates along both input and output channel dimensions to address the scale variations. We also note that some of the aforementioned modules are designed specifically for (semi-)dense prediction problems, *e.g.*, SAC, DCN, and POD and others do not scale to large-scale classification benchmarks like ImageNet, *e.g.*, STN for MNIST and SVHN, ACU for CIFAR. In contrast, our proposed PSConv focuses on backbone engineering, empirically shows its effectiveness on ImageNet and generalizes well to other complicated tasks on MS COCO. Furthermore, while the offsets are learned efficiently in some methods (ACU, SAC, and DCN), the inference is time-consuming due to the dynamic grid sampling and the bilinear interpolation at each position. Aligning to the tenet of POD [28], it is unnecessary to permit too much freedom with floating-point offsets at each spatial location as DCN [6] and learning in such an aggressive manner places an extra burden on the inference procedure. We opt for a better accuracy-efficiency trade-off by constraining dilation rates in the integer domain and organizing them into repeated partitions. Last but not least, the recently proposed MixConv [37] may be the most related scale module compared to PSConv, which will be discussed at the end of the next section.

3 Method

Compared to previous multi-scale feature fusion solutions in a coarse granularity, we seek an alternative design with the finer granularity and stronger feature extraction ability, while maintaining a similar computational load.

3.1 Sketch of Convolution Operations

We initiate from elaborating the vanilla (dilated) convolution process to make the definition of our proposed PSConv self-contained. For a single convolutional layer, let the tensor $\mathcal{F} \in \mathbb{R}^{C_{in} \times H \times W}$ denotes its input feature map with the shape of $C_{in} \times H \times W$, where C_{in} is the number of channels, H and W are the height and

width respectively. A set of C_{out} filters with the kernel size $K \times K$ are convolved with the input tensor individually to obtain the desired output feature map with C_{out} channels, where each filter has C_{in} kernels to match those channels in the input feature map. Denote the above filters as $\mathcal{G} \in \mathbb{R}^{C_{out} \times C_{in} \times K \times K}$, then the vanilla convolution operation can be represented as

$$\mathcal{H}_{c,x,y} = \sum_{k=1}^{C_{in}} \sum_{i=-\frac{K-1}{2}}^{\frac{K-1}{2}} \sum_{j=-\frac{K-1}{2}}^{\frac{K-1}{2}} \mathcal{G}_{c,k,i,j} \mathcal{F}_{k,x+i,y+j}, \tag{1}$$

where $\mathcal{H}_{c,x,y}$ is one element in the output feature map $\mathcal{H} \in \mathbb{R}^{C_{out} \times H \times W}$, $c = 1, 2, \cdots, C_{out}$ is the index of an output channel, $x = 1, 2, \cdots, H$ and $y = 1, 2, \cdots, W$ are indices of spatial positions in the feature map.

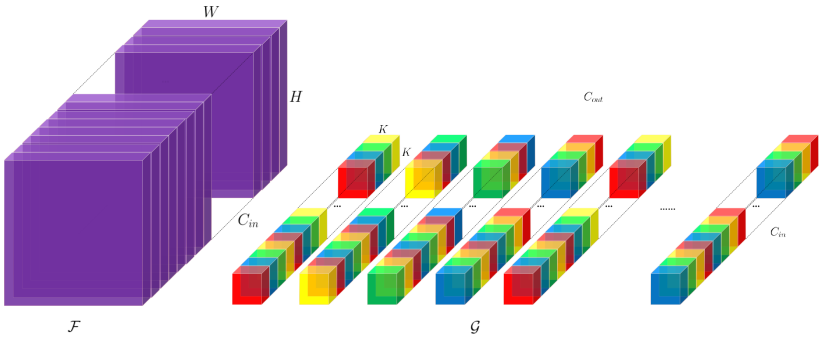

Fig. 1. Schematic illustration of our proposed PSConv operation. $\mathcal{F}$ represents the input feature map and $\mathcal{G}$ represents C_{out} convolutional filters in a set. Convolutional kernels with the same dilation rates in the set of filters $\mathcal{G}$ are rendered with the same color. Best viewed in color.

Dilated Convolution [41] enlarges sampling intervals in the spatial domain to cover objects of larger sizes. A dilated convolution with the dilation rate d can be represented as

$$\mathcal{H}_{c,x,y} = \sum_{k=1}^{C_{in}} \sum_{i=-\frac{K-1}{2}}^{\frac{K-1}{2}} \sum_{j=-\frac{K-1}{2}}^{\frac{K-1}{2}} \mathcal{G}_{c,k,i,j} \mathcal{F}_{k,x+id,y+jd}. \tag{2}$$

Noticing that a combination of dilation rates is conducive to extract both global and local information, we propose a new convolution form named Poly-Scale Convolution (PSConv) which scatters organized dilation rates over different kernels inside one convolutional filter. Furthermore, our PSConv integrates multi-scale features in a one-shot manner and brings characteristics of the dilated convolution into full play, thus without introducing additional computational cost. To gather multi-scale information from different input channels via a linear

summation, dilation rates are varied at different kernels in one convolutional filter. To process an input channel with various receptive fields, dilation rates are also varied in different filters for a certain channel. It is written as

$$\mathcal{H}_{c,x,y} = \sum_{k=1}^{C_{in}} \sum_{i=-\frac{K-1}{2}}^{\frac{K-1}{2}} \sum_{j=-\frac{K-1}{2}}^{\frac{K-1}{2}} \mathcal{G}_{c,k,i,j} \mathcal{F}_{k,x+iD_{(c,k)},y+jD_{(c,k)}}, \tag{3}$$

where $D \in \mathbb{R}^{C_{out} \times C_{in}}$ is a matrix composed of channel-wise and filter-wise dilation rates in two orthogonal dimensions. An element $D_{(c,k)}$ is associated with a specific channel in one filter to support $\mathcal{G}_{c,k,:,:}$ as a unique convolutional kernel, thus the whole matrix D can be interpreted as a mathematical representation of the kernel lattice in its subspace of dilation rate.

3.2 Design Details

As stated above, our major work is to reformulate the dilation rate patterns in the subspace of kernel lattice. We ensure that each row and column of the matrix D have non-identical elements to achieve the desired properties of multi-scale feature fusion. On the contrary, if we avoid and retain identical elements in one row, then we would not collect multi-scale information to produce a new output channel in this operation, and it can be boiled down to multi-stream transformation before concatenation; if the similar event occurs in one column, the corresponding input channel would not have necessarily diverse receptive fields covered, and it reduces to the split-transform-summation design of multi-scale networks. These are both suboptimal according to our ablative experiments in Table 5. The illustration diagrams of these two simplified cases are provided in the supplementary materials.

Following the above analysis, the design philosophy of PSConv could be decomposed into two coupled ingredients. Firstly, we concentrate on a single filter. In order to constrain the number of different dilation rates in a reasonable range, we heuristically arrange them inside one filter with a cyclic layout, *i.e.*, dilation rates vary in a periodical manner along the axis of input channels. Specifically speaking, a total of C_{in} input channels are divided into P partitions. For each partition, $t = \lceil \frac{C_{in}}{P} \rceil$ channels are accommodated and a fixed pattern of dilation rates $\{d_1, d_2, \cdots, d_t\}$ is filled in to construct a row of the matrix D. Secondly, we broaden our horizons to all filters. In order to endow different filters with capacities to gather different kinds of scale combinations of input features, we adopt a shift-based strategy for dilation rates to flip the former filter to the latter one, *i.e.*, the pattern of dilation rates regarding a convolutional filter is shifted by one channel to build its adjacent filter. In the illustrative example of Fig. 2, $C_{in} = C_{out} = 16$ and the partition number P is set to 4, hence there leaves a blank of 4 dilation rates to be determined in the pattern $\{d_1, d_2, d_3, d_4\}$, where a specific colorization distinguishes one type of dilation rate from others. It is noted that viewed from the axis of output channels, dilation rates also present periodical variation. In other words, all types of dilation rates occur alternately along the vertical and horizontal axes in the trellis.

Furthermore, a comparison diagram is shown in Fig. 2, to achieve better intuitive comprehension about different convolution operations. The filters of PSConv are exhibited from the vertical view of $\mathcal{G}$ (with appropriate rotate transformation) in Fig. 1, where each tile in the grid represents a kernel of $K \times K$ shape and the grid corresponds to the dilation rate matrix D. The filters of (dilated) convolution and group convolution are likewise displayed. In the conventional filters, if a dilation rate is applied, it will dominate the whole kernel lattice, while our PSConv has clear distinctions compared with them. We claim that merely varying dilation rates in the axis of output channels equals to using split-transform-merge units spanning a spectrum of dilation rates in the different streams. Our method takes one step further to spread the scale information along both input and output channel axes, pushing the selection of scale-variant features into the entire kernel space. To the best of our knowledge, **it is the first attempt to mix up multi-scale information simultaneously in two orthogonal dimensions and leverage the complementary multi-scale benefits from such a fine granularity in the kernel space**.

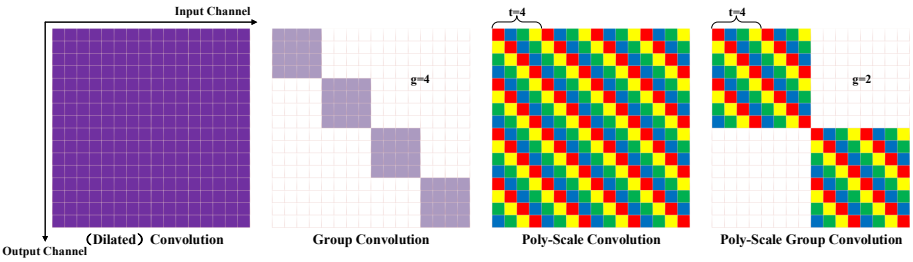

Fig. 2. Comparison between dilation space of kernel lattice in different convolution operations. Kernels of standard convolution (with or without dilation) are showcased in the leftmost, where each kernel is located at one cell in the lattice. Group convolution (group number $g = 4$) extensively utilized in the efficient network design is also included for reference. Poly-Scale convolution (cyclic interval $t = 4$) and Poly-Scale group convolution (group number $g = 2$ and cyclic interval $t = 4$) in the right show significant differences from the former two. Best viewed in color.

It is noteworthy that PSConv is a generalized form of dilated convolution: since the cyclic interval t decides how many types of dilation rates are contained in one partition, all kernels may share the same dilation rate once the partition number equals to that of input channels and then it degenerates into vanilla dilated convolution. The PSConv can also be applied to the group-wise convolution form by injecting the shared cyclic pattern into each group, as illustrated in the rightmost of Fig. 2. Owing to the interchangeability of channel indices, grouping channels with the same dilation rate together leads to an equivalent but efficient implementation, which is depicted in the supplementary materials.

The recently proposed MixConv [37] might be similar to PSConv at the first glimpse. However, they are distinct regarding both the design principle and the

application focus. On the one hand, MixConv integrates multiple kernel sizes for different patterns of resolutions which inevitably increases the parameters and computational budget, while PSConv mixes up a spectrum of dilation rates with a unified kernel size to economically condense multi-scale features within one convolution operation. Thus, for these two convolution forms, the manipulations on the kernel lattice are shaped from orthogonal perspectives. On the other hand, MixConv is dedicatedly developed for depthwise convolution (DWConv), while PSConv is versatile to both standard and group convolution. Due to the inherent constraint of DWConv, each individual channel in a MixConv operation exploits feature representation of a certain scale. However, in our PSConv, multi-scale representations are scattered along both input and output channels in a periodical manner. Hence, an individual channel could gather multifarious feature resolutions from the view of either input or output channels. We attach a more in-depth discussion around their differences and an illustration of the DWConv-based variant of PSConv in the supplementary materials.

4 Experiments

We conduct extensive experiments from conceptual to dense prediction tasks on several large-scale visual recognition benchmarks. Experimental results empirically validate the effectiveness and efficiency of our proposed convolution form. All experiments are performed with the PyTorch [27] library.

4.1 ILSVRC 2012

ImageNet [7] is one of the most challenging datasets for image classification, which is served as the benchmark of the ILSVRC2012 competition. It includes 1,281,167 training images and 50,000 validation images, and each image is manually annotated as one of the 1,000 object categories.

We incorporate our PSConv layer into various state-of-the-art convolutional neural networks, including ResNet [11], ResNeXt [39] and SE-ResNet [12]. The training procedure is performed on the ImageNet training set by the SGD optimizer with the momentum of 0.9 and the weight decay of 1e-4. The mini-batch size is set to 256 and the optimization process lasts for a period of 120 epochs to achieve full convergence. The learning rate initiates from 0.1 and decays to zero following a half cosine function shaped schedule, the same as [2] and [5]. We adopt random scale and aspect ratio augmentation together with random horizontal flipping to process each training sample prior to feeding it into neural networks. We select the best-performing model along the training trajectory and report its performance on the ImageNet validation set. As is the common practice, we first resize the shorter side of validation images to 256 pixels and then crop the central region of 224×224 size for evaluation.

As shown in Table 1, network models equipped with PSConv layers demonstrate consistent improvement over counterpart baseline models mostly with over 1% gains of the top-1 error. We replace all the 3×3 standard convolutional layers in the middle of bottleneck blocks with our PSConv layers. In all of our main experiments, the cyclic interval is set to 4 and the dilation rate pattern is fixed as $\{d_1, d_2, d_3, d_4\} = \{1, 2, 1, 4\}$ which are determined by ablation studies, as detailed in the next subsection. It is observed that the PS-ResNet-50 model achieves 21.126% top-1 error, which is comparable to the vanilla ResNet-101 model with almost half of the trainable parameter storage and computational resource consumption. The PS-ResNeXt-50 (32×4d) model even achieves superior performance over the vanilla 101-layer ResNeXt model, which demonstrates the wide applicability of our PSConv in boosting both standard and group convolution. Furthermore, we integrate PSConv into the modern SE-ResNet models and obtain performance margins again, which showcases the compatibility of our proposed convolution operation to other advanced atomic operations such as the channel-attention modules. Notably, all the above gains are obtained without theoretically introducing any additional computational cost.

Table 1. Recognition error comparisons on the ImageNet validation set. The standard metrics of top-1/top-5 errors are measured using single center crop evaluation. The baseline results are re-implemented by ourselves.

Architecture	Conv Type	Top-1/Top-5 Err.(%)	Architecture	Conv Type	Top-1/Top-5 Err.(%)
ResNet-50	Standard	22.850/6.532	ResNet-101	Standard	21.102/5.696
	PSConv	**21.126/5.724**		PSConv	**19.954/5.052**
ResNeXt-50 (32×4d)	Standard	21.802/6.084	ResNeXt-101 (32×4d)	Standard	20.502/5.390
	PSConv	**20.378/5.296**		PSConv	**19.498/4.724**
SE-ResNet-50	Standard	22.192/6.040	SE-ResNet-101	Standard	20.732/5.406
	PSConv	**20.814/5.578**		PSConv	**19.786/4.924**

For horizontal comparison, we give a brief synopsis of some recent multi-scale networks in Table 2 for reference. Despite that discrepancies in model profiles and training strategies could lead to no apple-to-apple comparisons in most cases, our PS-ResNet-50 achieves competitive accuracy compared to other ResNet-50-based architectures under the similar level of parameters and computational complexities. Specifically, two variants of Dilated Residual Networks (DRN) increase the computation cost to a large extent due to the removed strides in the last two residual stages, but only achieves inferior or comparable performance.

Table 2. Performance comparison with state-of-the-art multi-scale network architectures on the ImageNet validation set.

Network	Params	GFLOPs	LR decay schedule	Top-1/Top-5 Err.(%)
ResNet-50 [11]	25.557M	4.089	Cosine (120 epoch)	22.850/6.532
PS-ResNet-50 (ours)	25.557M	4.089	Cosine (120 epoch)	**21.126/5.724**
DRN-A-50 [41]	25.557M	19.079	Stepwise (120 epoch)	22.9/6.6
DRN-D-54 [41]	35.809M	28.487	Stepwise (120 epoch)	21.2/5.9
FishNet-150 [34]	24.96M	6.45	Stepwise (100 epoch)	21.86/6.05
FishNet-150 [34]	24.96M	6.45	Cosine (200 epoch) w/ label smoothing	20.65/5.25
HRNet-W18-C [33]	21.3M	3.99	Stepwise (100 epoch)	23.2/6.6
OctResNet-50 [5] ($\alpha = 0.5$)	25.6M	2.4	cosine (110 epoch)	22.6/6.4
bL-ResNet-50 [2] ($\alpha = 2, \beta = 4$)	26.69M	2.85	cosine (110 epoch)	22.69/-
Res2Net-50 [8] ($26w \times 4s$)	25.70M	4.2	Stepwise (100 epoch)	22.01/6.15
ScaleNet-50 [20]	31.483M	3.818	Stepwise (100 epoch)	22.02/6.05

4.2 Ablation and Analysis

We first systematically probe the impact of partition numbers and dilation rate patterns in one cycle. We next assess the ability of PSConv to generalize to another classification benchmark beyond ImageNet, namely CIFAR-100.

Partition Number. On the one hand, provided that channels are divided into too many partitions, there leaves limited room for varied dilation rates within one partition and it frequently alternates around certain values. In the extreme case that the partition number equals to the number of channels, PSConv degenerates into the vanilla dilated convolution with a shared dilation rate. On the other hand, if there are too few partitions, each partition can accommodate a large number of heterogeneous dilation rates, which may have contradictory effects on extracting diverse features, hence we initially constrain the dilation rate in one basic pattern to toggle between 1 and 2 in this set of ablation experiments. Specifically, we set the dilation rate in one slot of a cycle to 2 and the other slots to 1. Under this constraint, features corresponding to large receptive fields will infrequently emerge with the growing cyclic interval, which may still impede the full utilization of multi-scale features.

Table 3. Performance comparison of PS-ResNet-50 with varied cyclic intervals on the ImageNet validation set. The best result is highlighted in **bold**, the same hereinafter.

Architecture	ResNet-50	PS-ResNet-50		
Cyclic Interval	$t = 1$ (baseline)	$t = 2$	$t = 4$	$t = 8$
Top-1/Top-5 Err.(%)	22.850/6.532	21.948/5.978	**21.476/5.720**	21.634/5.816

Table 4. Performance comparison of PS-ResNet-50 with various dilation patterns on the ImageNet validation set.

Dilation Pattern	$\{1, 1, 1, 1\}$ (**baseline**)	$\{1, 2, 1, 1\}$	$\{1, 4, 1, 1\}$	$\{1, 2, 1, 2\}$	$\{1, 2, 1, 4\}$ (**default**)
Top-1/Top-5 Err.(%)	22.850/6.532	22.368/6.214	22.754/6.470	21.948/5.978	**21.126/5.724**

We use the ResNet-50 model on the ImageNet dataset for experiments and tune the partition numbers, giving rise to a spectrum of cyclic intervals. The corresponding results shown in Table 3 empirically support our speculation above. The PS-ResNet-50 ($t = 4$) achieves better performance when the cyclic interval increases from 2 to 4. The accuracy tends to decline when its cyclic interval gets further increment. Thus we set $t = 4$ as the default value in our main experiments. In each case, PS-ResNet-50 with a specific cyclic setting outperforms the vanilla ResNet-50 baseline result.

Pattern of Dilation Rates. Let the cyclic interval be 4. Noticing that the dilation rate pattern is an unordered set, we initially set any one of the dilation rate to a larger numeric value. For example, $\{d_1, d_2, d_3, d_4\}$ is set to $\{1, 2, 1, 1\}$, where the unique large dilation rate is placed in the second slot without loss of generality owing to its unordered nature. Next we assume that further increasing this large dilation rate (e.*g.*, setting $\{d_1, d_2, d_3, d_4\} = \{1, 4, 1, 1\}$) would lead to intra-group separation of these two dilation rates and unsmoothed transition of the receptive fields. Then we tend to inject another large dilation rate into this pattern. Considering that the setting of $\{d_1, d_2, d_3, d_4\} = \{1, 2, 1, 2\}$ is equivalent to $t = 2$ in the above experiments, we change the pattern to $\{d_1, d_2, d_3, d_4\} = \{1, 2, 1, 4\}$ for the sake of perceiving larger receptive fields and interspacing the two different large dilation rates. This consequent PS-ResNet-50 achieves 21.126% top-1 error in the ImageNet evaluation, which is exactly the one reported in Table 1. For further exploration, we tentatively incorporate larger dilation rate to compose the combination of $\{d_1, d_2, d_3, d_4\} = \{1, 2, 4, 8\}$, but it shows much inferior performance (over 5% drop). We attribute this failure to the exclusively aggressive dilation rate arrangement, since inappropriately enlarging the receptive field can involve irrelevant pixels into spatial correlation (Table 4).

Apart from the static setting of dilation rates, we develop a learnable binary mask to distinguish the large dilation rate from the small one. This binary mask is decomposed via the Kronecker product, where the STE (Straight-Through Estimator) [29] technique is utilized to solve the discrete optimization problem. As a consequence, the dynamic version of PS-ResNet-50 with optional dilation rates of 1 and 2 reduces the top-1 error to 21.138%, that is close to the best-performing static PS-ResNet-50 ($t = 4$, $\{d_1, d_2, d_3, d_4\} = \{1, 2, 1, 4\}$) involving larger dilation rates in its PSConv pattern. Although extra parameters and computational complexity result in no fair comparison, it opens up a promising perspective deserving future research development.

Table 5. Performance comparison of PS-ResNet-50 on the ImageNet validation set, with the variation of dilation rates along different axes of kernel lattice.

Input Channel Axis	Output Channel Axis	Top-1/Top-5 Err.(%)
✓	✗	21.658/5.832
✗	✓	22.056/6.174
✓	✓	**21.126/5.724**

Table 6. Top-1 error comparisons on the CIFAR-100 test set. Our results were obtained by computing mean and standard deviation over 5 individual runs (denoted by mean ± std. in the table).

Architecture	Conv Type	Top-1 Error(%)	Architecture	Conv Type	Top-1 Error(%)
ResNeXt-29 (8 × 64d)	Standard (*official*)	17.77	ResNeXt-29 (16 × 64d)	Standard (*official*)	17.31
	Standard (*self impl.*)	18.074 ± 0.130		Standard (*self impl.*)	17.538 ± 0.094
	PSConv	**17.138 ± 0.286**		PSConv	**16.528 ± 0.353**

Following the searched optimal setting of $\{d_1, d_2, d_3, d_4\} = \{1, 2, 1, 4\}$, we remove the shift strategy among different filters, which means the variation of dilation rates only exists in the axis of input channels. In this setup, we observe a drop of around 1% regarding the top-1 validation accuracy. Symmetrically, we only vary the dilation rates in the axis of output channels with the same setting of $\{d_1, d_2, d_3, d_4\} = \{1, 2, 1, 4\}$, which indicates no cyclic operations inside each individual filter. As shown in Table 5, it also achieves inferior performance.

Beyond ImageNet. CIFAR-100 [17] is another widely-adopted benchmark for image classification, which consists of 50,000 training images and 10,000 test images. Each colorful image in the dataset is of 32×32 size and drawn from 100 classes, hence it is more challenging than CIFAR-10 with similar image qualities but a coarser taxonomy. We choose the high-performing ResNeXt [39] architecture as a strong baseline, and replace all the 3×3 convolutional layers in every bottleneck block with PSConv layers to build our PS-ResNeXt models for comparison. The data augmentation is the same as the preprocessing method in [11,39], utilizing sequential zero padding, random cropping and standardization. The whole training regime strictly follows the original paper to isolate the contribution of our PSConv. For evaluation, we perform five independent runs of training the same architecture with different initialization seeds and report the mean top-1 error as well as the standard deviation.

We summarize the comparison results in Table 6. The performance of our reproduced ResNeXt-29 is slightly degraded, thus we list results from both the official release and our implementation, annotated as *official* and *self impl.* with the standard convolution respectively. It is evident that PS-ResNeXt-29 (8 × 64d) and PS-ResNeXt-29 (16 × 64d) outperform the original ResNeXts by around 1% accuracy gains. Even compared to the results from the original author, absolute gains of 0.632% and 0.782% are achieved using our PSConv neural networks. It is observed that using networks with various cardinalities on

datasets with distinct characteristics (like thumbnails), PSConv could still yield satisfactory performance gains.

Speed Benchmark. For an input tensor with the size of $(N, C, H, W) = (200, 64, 56, 56)$, a standard 3×3 convolutional layer with 64 output channels takes 4.85ms to process on a single Titan X GPU, using CUDA v9.0 and cuDNN v7.0 as the backend. The dilated convolution with a dilation rate of 2 consumes 2.99 times of above and the inference time of our PSConv is 1.14× of dilated convolution. There exist a similar trend in the comparison of their group convolution based counterparts. Thus, improved performance of inference speed can be achieved by optimizing vanilla dilated convolutions on GPU/CPU inference. The further optimized results for practical deployment are provided in the supplementary materials.

Scale Allocation. We dive into the PSConv kernels to analyze the law of scale-relevant feature distributions by dissecting the weight proportion with respect to different dilation rates, as is shown in the supplementary materials.

4.3 MS COCO 2017

To further demonstrate the generality of our proposed convolution, we apply the PSConv-based backbones to object detection and instance segmentation frameworks and finetune the PSConv-based detectors on the 2017 version of Microsoft COCO [23] benchmark. This large-scale dataset including 118,287 training images and 5,000 validation images is considered highly challenging owing to the huge number of objects within per image and large variation among these instances, which is suitable for inspecting the superiority of our PSConv models.

We use the popular MMDetection [3] toolbox to conduct experiments. ResNet-50/101 and ResNeXt-101 (32 × 4d) along with FPN [22] necks are selected as the backbone networks. For object detection and instance segmentation tasks, we adopt the main-stream Faster R-CNN [30] and Mask R-CNN [10] as the basic detectors respectively. We replace all the 3×3 convolutional layers in the pre-trained backbone network by PSConv layers, while the convolution layers in the FPN neck are kept as standard convolutions[2]. Then we finetune these detectors on the training set following the 1× learning rate schedule, which indicates a total of 12 epochs with the learning rate divided by 10 at the epoch of 8^{th} and 11^{st} respectively. During this transfer learning process, we maintain the same data preparation pipeline and hyperparameter settings for our models as the baseline detectors. During evaluation, we test on the validation set and report the COCO-style Average Precision (AP) under IOU thresholds ranging from 0.5 to 0.95 with an increment of 0.05. We also keep track of scores for small, medium and large objects. These metrics comprehensively assess the qualities of detection and segmentation results from various views of different scales.

[2] Actually we have preliminary experiments by also replacing these layers with PSConv layers, but it achieves marginal benefit. For instance, AP^{bbox} of Faster R-CNN with ResNet-50 and FPN only increases from 38.4% to 38.6%.

Table 7. Bounding-box and mask Average Precision (AP) comparison on the COCO 2017 validation set for the instance segmentation track with different backbones.

Detector	Architecture	Conv Type	Box AP						Mask AP					
			AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN	R50	Standard	37.3	59.0	40.2	21.9	40.9	48.1	34.2	55.9	36.2	15.8	36.9	50.1
		PSConv	$39.4_{(+\mathbf{2.1})}$	61.3	42.8	24.1	43.1	51.3	$35.6_{(+\mathbf{1.4})}$	57.9	37.9	17.2	38.4	52.4
	R101	Standard	39.4	60.9	43.3	23.0	43.7	51.4	35.9	57.7	38.4	16.8	39.1	53.6
		PSConv	$41.6_{(+\mathbf{2.2})}$	63.4	45.1	24.7	45.6	54.4	$37.4_{(+\mathbf{1.5})}$	60.0	39.8	17.8	40.4	55.1
	X101-32 × 4d	Standard	41.1	62.8	45.0	24.0	45.4	52.6	37.1	59.4	39.7	17.7	40.5	53.8
		PSConv	$42.4_{(+\mathbf{1.3})}$	64.4	46.1	25.4	46.5	55.7	$38.0_{(+\mathbf{0.9})}$	60.8	40.5	18.6	41.0	55.8
Cascade Mask R-CNN	R50	Standard	41.2	59.1	45.1	23.3	44.5	54.5	35.7	56.3	38.6	16.4	38.2	52.6
		PSConv	$42.9_{(+\mathbf{1.7})}$	61.7	46.9	24.2	46.5	57.2	$36.9_{(+\mathbf{1.2})}$	58.4	39.5	17.1	39.4	54.6
	R101	Standard	42.6	60.7	46.7	23.8	46.4	56.9	37.0	58.0	39.9	16.7	40.3	54.6
		PSConv	$44.6_{(+\mathbf{2.0})}$	63.2	48.6	25.9	48.7	59.6	$38.4_{(+\mathbf{1.4})}$	60.5	41.2	18.6	41.5	56.8
	X101-32 × 4d	Standard	44.4	62.6	48.6	25.4	48.1	58.7	38.2	59.6	41.2	18.3	41.4	55.6
		PSConv	$45.3_{(+\mathbf{0.9})}$	64.2	49.5	27.0	49.2	60.0	$38.9_{(+\mathbf{0.7})}$	61.1	41.8	19.4	41.9	56.6

The comparison results of Mask R-CNN are shown in Table 7 (similar comparisons of Faster R-CNN are provided in the supplementary materials), where the baseline results with standard backbone networks are extracted from the model zoo of MMDetection, and absolute gains of AP concerning our PSConv models are indicated in the parentheses. Since our ImageNet pre-trained backbones in Sect. 4.1 are trained using the cosine learning rate annealing, we would have an unfair accuracy advantage against the pre-trained backbones in MMDetection. In order to pursue fair comparison to its published baseline results, we first retrain backbones of ResNet and ResNeXt following the conventional stepwise learning rate annealing strategy [11] and then load these backbones to the detectors[3]. It is evident that PSConv brings consistent and considerable performance gains over the baseline results, across different tasks and various backbones. In addition, we introduce the Cascade (Mask) R-CNN [1] as a stronger baseline detector and reach the conclusion that our PSConv operation can benefit both basic detectors and more advanced cascade detectors.

Detectors with the ResNet-101 backbone consistently show larger margins among different tasks and frameworks compared to the other two backbones. Compared to ResNet-50, the 101-layer network almost quadruples the depth of the conv4_x stage, guaranteeing a higher capacity for performance amelioration. In addition, we come up with the hypothesis that its ResNeXt counterpart has already efficiently deployed the model capacity through adjusting the dimension of cardinality beyond network depth and width, leaving a bottleneck for further performance improvement in both classification and detection tasks. It is observed that the most significant improvement of Faster R-CNN and Mask R-CNN locates in the metric of AP_L among various object sizes, speaking to the theoretically enlarged receptive fields. Finally, representative visualization results of predicted bounding-boxes and masks are attached in the supplementary materials to raise the qualitative insight of our method.

5 Conclusion

In this paper, we have proposed a novel convolution operation named PSConv, which cyclically alternates dilation rates along the axes of input and output channels. PSConv permits to aggregate multi-scale features from a granular perspective and efficiently allocates weights to a collection of scale-specific features through dynamic execution. It is amenable to be plugged into arbitrary state-of-the-art CNN architectures in-place, demonstrating its superior performance on various vision tasks compared to the counterparts with regular convolutions.

References

1. Cai, Z., Vasconcelos, N.: Cascade R-CNN: delving into high quality object detection. In: CVPR (2018)

[3] If we adopt those unfair backbones pre-trained using cosine learning rate decay in Sect. 4.1, we can get even larger performance margins (*e.g.* 2.6% instead of 2.0% for Faster R-CNN with ResNet-50 and FPN).

2. Chen, C.F.R., Fan, Q., Mallinar, N., Sercu, T., Feris, R.: Big-little net: an efficient multi-scale feature representation for visual and speech recognition. In: ICLR (2019)
3. Chen, K., et al.: MMDetection: open MMLab detection toolbox and benchmark. arXiv e-prints arXiv:1906.07155 (Jun 2019)
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
5. Chen, Y., et al.: Drop an octave: reducing spatial redundancy in convolutional neural networks with octave convolution. In: ICCV (2019)
6. Dai, J., et al.: Deformable convolutional networks. In: ICCV (2017)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR (2009)
8. Gao, S.H., Cheng, M.M., Zhao, K., Zhang, X.Y., Yang, M.H., Torr, P.: Res2net: a new multi-scale backbone architecture. IEEE TPAMI, 1 (2019)
9. Girshick, R.: Fast R-CNN. In: ICCV (2015)
10. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
12. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR (2018)
13. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
14. Jaderberg, M., Simonyan, K., Zisserman, A., kavukcuoglu, k.: Spatial transformer networks. In: NIPS (2015)
15. Jeon, Y., Kim, J.: Active convolution: learning the shape of convolution for image classification. In: CVPR (2017)
16. Ke, T.W., Maire, M., Yu, S.X.: Multigrid neural architectures. In: CVPR (2017)
17. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto (2009)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
19. Li, Y., Chen, Y., Wang, N., Zhang, Z.: Scale-aware trident networks for object detection. In: ICCV (2019)
20. Li, Y., Kuang, Z., Chen, Y., Zhang, W.: Data-driven neuron allocation for scale aggregation networks. In: CVPR (2019)
21. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: Detnet: design backbone for object detection. In: ECCV (2018)
22. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
23. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
24. Liu, S., Huang, D., Wang, Y.: Receptive field block net for accurate and fast object detection. In: ECCV (2018)
25. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
26. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: ICCV (2015)
27. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: NeurIPS (2019)

28. Peng, J., Sun, M., Zhang, Z., Yan, J., Tan, T.: POD: practical object detection with scale-sensitive network. In: ICCV (2019)
29. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: XNOR-Net: imagenet classification using binary convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 525–542. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_32
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
31. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
33. Sun, K., et al.: High-resolution representations for labeling pixels and regions. arXiv e-prints arXiv:1904.04514, April 2019
34. Sun, S., Pang, J., Shi, J., Yi, S., Ouyang, W.: Fishnet: a versatile backbone for image, region, and pixel level prediction. In: NeurIPS (2018)
35. Szegedy, C., et al.: Going deeper with convolutions. In: CVPR (2015)
36. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
37. Tan, M., Le, Q.V.: Mixconv: Mixed depthwise convolutional kernels. In: BMVC (2019)
38. Wang, H., Kembhavi, A., Farhadi, A., Yuille, A.L., Rastegari, M.: ELASTIC: improving cnns with dynamic scaling policies. In: CVPR (2019)
39. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR (2017)
40. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV (2015)
41. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: CVPR (2017)
42. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: CVPR (2018)
43. Zhang, R., Tang, S., Zhang, Y., Li, J., Yan, S.: Scale-adaptive convolutions for scene parsing. In: ICCV (2017)
44. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
45. Zhou, K., Yang, Y., Cavallaro, A., Xiang, T.: Omni-scale feature learning for person re-identification. In: ICCV (2019)
46. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: more deformable, better results. In: CVPR (2019)