Global Second-order Pooling Convolutional Networks

Zilin Gao[1,2], Jiangtao Xie[1], Qilong Wang[3], Peihua Li[1*]
[1]Dalian University of Technology [2]DAMO Academy, Alibaba Group [3]Tianjin University
{gzl, jiangtaoxie}@mail.dlut.edu.cn, qlwang@tju.edu.cn, peihuali@dlut.edu.cn

Abstract

Deep Convolutional Networks (ConvNets) are fundamental to, besides large-scale visual recognition, a lot of vision tasks. As the primary goal of the ConvNets is to characterize complex boundaries of thousands of classes in a high-dimensional space, it is critical to learn higher-order representations for enhancing non-linear modeling capability. Recently, Global Second-order Pooling (GSoP), plugged at the end of networks, has attracted increasing attentions, achieving much better performance than classical, first-order networks in a variety of vision tasks. However, how to effectively introduce higher-order representation in earlier layers for improving non-linear capability of ConvNets is still an open problem. In this paper, we propose a novel network model introducing GSoP across from lower to higher layers for exploiting holistic image information throughout a network. Given an input 3D tensor outputted by some previous convolutional layer, we perform GSoP to obtain a covariance matrix which, after nonlinear transformation, is used for tensor scaling along channel dimension. Similarly, we can perform GSoP along spatial dimension for tensor scaling as well. In this way, we can make full use of the second-order statistics of the holistic image throughout a network. The proposed networks are thoroughly evaluated on large-scale ImageNet-1K, and experiments have shown that they outperform non-trivially the counterparts while achieving state-of-the-art results.

1. Introduction

Deep Convolutional Networks (ConvNets) are fundamental to computer vision field, since they are not only paramount for high accuracy of large-scale object recognition, but also play central roles, through means of pre-trained models, in advancing substantially many other computer vision tasks, e.g., object detection [29], semantic segmentation [27] and video classification [35]. Given color images as inputs, the ConvNets can learn progressively the low-level, mid-level and high-level features [42], finally producing global image representations connected to softmax layer for classification. To better characterize complex boundaries of thousands of classes in a very high-dimensional space, one possible solution is to learn higher-order representations for enhancing nonlinear modeling capability of ConvNets.

Recently, modeling of higher-order statistics for more discriminative image representations has attracted great interests in deep ConvNets. The global second-order pooling[1] (GSoP), producing covariance matrices as image representations, has achieved state-of-the-art results in a variety of vision tasks [22, 3, 33, 36] such as object recognition, fine-grained visual categorization, object detection and video classification. The pioneering works, i.e., DeepO$_2$P [18] and bilinear CNN (B-CNN) [26], performed global second-order pooling, rather than the commonly used global average (i.e., first-order) pooling (GAvP) [25], after the last convolutional layers in an end-to-end manner. However, most of the variants of GSoP [7, 1] only focused on small-scale scenarios. In large-scale visual recognition, MPN-COV [23, 22] has shown matrix power normalized GSoP can significantly outperform global average pooling.

Though GSoP plugged at the end of network has proven successful, how to effectively introduce higher-order representation in earlier layers for improving non-linear capability of ConvNets is still an open problem. Several works [24, 37, 43] have made attempts to enhance non-linear modeling capability using quadratic transformation to model feature interactions, instead of only using linear transformation of convolutions. However, performance gains of these methods are limited in large-scale visual recognition. Motivated by Squeeze-and-Excitation (SE) networks [15], we introduce GSoP across from lower to higher layers of deep ConvNets, aiming to learn more discriminative representations by exploiting the second-order statistics of holistic image throughout a deep ConvNet.

At the heart of our global second-order networks is the GSoP block, which can be conveniently plugged into any

*Peihua Li is the corresponding author.
The work was supported by National Natural Science Foundation of China (No. 61471082 and No. 61806140).

[1]To our knowledge the term "second-order pooling" was coined in [2].

location of a deep ConvNet. Given a 3D tensor outputted by some previous convolutional layer, we first perform GSoP to model pairwise channel correlations of the holistic tensor. We then accomplish embedding of the resulting covariance matrix by convolutions and non-linear activations, which is finally used for scaling the 3D tensor along channel dimension. The diagram of our GSoP convolutional network (GSoP-Net) is presented in Figure 1a and the proposed second-order block is illustrated in Figure 1b. The primary differences of the proposed GSoP-Net from existing networks are compared in Table 1, which will be detailed in next section. Our main contributions are threefold. (1) Distinct from the existing methods which can only exploit second-order statistics at network end, we are among the first who introduce this modeling into intermediate layers for making use of holistic image information in earlier stages of deep ConvNets. By modeling the correlations of the holistic tensor, the proposed blocks can capture long-range statistical dependency [35], making full use of the contextual information in the image. (2) We design a simple yet effective GSoP block, which is highly modular with low memory and computational complexity. The GSoP block, which is able to capture global second-order statistics along channel dimension or position dimension, can be conveniently plugged into existing network architectures, further improving their performance with small overhead. (3) On ImageNet benchmark, we perform a thorough ablation study of the proposed networks, analyzing the characteristics and behaviors of the proposed GSoP block. Extensive comparison with the counterparts has shown the competitiveness of our networks.

2. Related Works

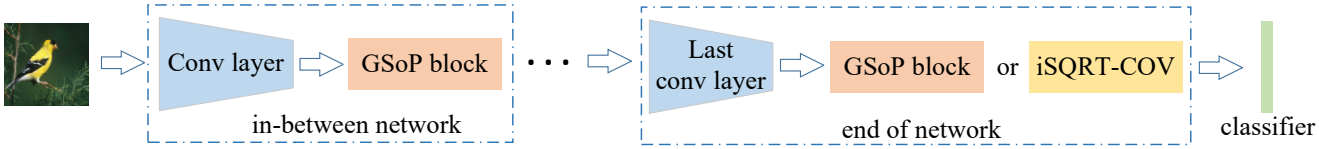
GAvP (1^{st}–order) In-between Network. Global average pooling plugged at the end of network [25], which summarizes the first-order statistics (i.e., mean vector) as image representations, has been widely used in most deep ConvNets such as ResNet [11], Inception [31] and DenseNet [17]. For the first time, SE-Net [15] introduced GAvP in-between network for making use of holistic image context at earlier stages, reporting significant improvement over its network-end counterparts. The SE-Net consists of two modules: a squeeze module accomplishing global average pooling followed by convolution and non-linear activations for capturing channel dependency, and an excitation module scaling channel for data recalibration. Besides GAvP along channel dimension, CBAM [38] extends the idea of SE-Net, combining GAvP along channel dimension as well as spatial dimension for accomplishing self-attention. Compared to SE-Net and CBAM which use only first-order statistics (mean) of the holistic image, our GSoP-Net exploits second-order statistics (correlations), having stronger modeling capability.

	in-between network		end of network	
	global pool	method	global pool	method
AlexNet [20] VGG [30]	×	N/A	×	N/A
ResNet [11] Inception [31] DenseNet [17]	×	N/A	√	1^{st}–order
SE-Net [15] CBAM [38]	√	1^{st}–order	√	1^{st}–order
DeepO$_2$P [18] B-CNN [26] MPN-COV [23, 22] G^{2}DeNet [34]	×	N/A	√	2^{nd}–order
GSoP-Net (ours)	√	2^{nd}–order	√	2^{nd}–order

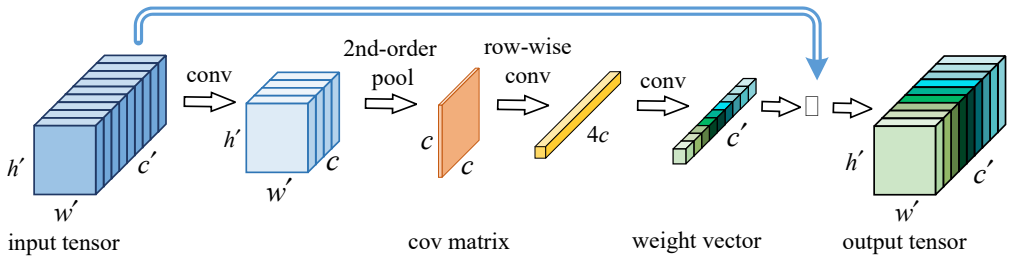
Table 1: Summary of ConvNet models in terms of global statistical pooling. Different from existing networks, we introduce global second-order pooling into intermediate layers of deep ConvNets. So we can make full use of second-order statistics to effectively capture holistic image information throughout a network.

GSoP (2^{nd}–order) at Network Net. The global second-order pooling, plugged at network end and trainable in an end-to-end manner, has received great interests, achieving significant performance improvement [3, 23, 22]. Several researchers [7, 3, 1] have shown close connections between higher-order pooling with kernel machines, based on which they proposed explicit mapping functions as kernel approximation for compactness of covariance representations. Wang et al. [34] proposed a global Gaussian distribution embedding network (G^{2}DeNet), where one multivariate Gaussian, identified as a symmetric positive definite matrix of covariance matrix and mean vector [21], is plugged at network end. MoNet [39] proposed a sub-matrix square-root layer, enabling G^{2}DeNet to have compact representation. In [4], the first-order information is combined with the second-order one which achieves consistent improvements over the standard bilinear networks on texture recognition. In all the aforementioned works, second-order modeling are only exploited at the end of deep networks.

Quadratic Transformation Network. The conventional network depends heavily on linear convolution operations. Several researchers take a step further to explore higher-order transformation for enhancing non-linear modeling capability of deep networks. The second-order Response Transform (SORT) [37] develops a two-branch network module to combine responses of two convolutional blocks and multiplication of the responses. They perform element-wise square root for normalizing the second-order term. In [24], a factorized bilinear network (FBN) is proposed to model the pairwise feature interaction. By constraining the rank of quadratic transformation matrix, FBN can introduce



(a) Overview of GSoP-Net. The proposed global second-order pooling (GSoP) block can be conveniently inserted after any convolutional layer in-between network. We propose to use, at the network end, GSoP block followed by common global average pooling producing compact image representations (GSoP-Net1), or matrix power normalized covariance [23] outputting covariance matrices as image representations (GSoP-Net2).



(b) GSoP block. Given an input tensor, after dimension reduction, the GSoP block starts with covariance matrix computation, followed by two consecutive operations of a linear convolution and non-linear activation, producing the output tensor which is scaling (multiplication) of the input one along the channel dimension.

Figure 1: Our global second-order pooling network (GSoP-Net). Figure 1a gives an overview of GSoP-Net and the proposed GSoP block is presented in Figure 1b. We introduce global second-order pooling into intermediate layers of deep ConvNets, which goes beyond the existing works where GSoP can only be used at network end. By modeling higher-order statistics of holistic images at earlier stages, our network can enhance capability of non-linear representation learning of deep networks.

bilinear pooling into intermediate layers. Zoumpourlis et al. [43] introduce Volterra kernel-based convolutions, which can model first-, second- or higher-order interactions of data, serving as approximations of non-linear functionals. All the works above are concerned with non-linear filters, applied only to local neighborhood, just like linear convolution. In contrast, our GSoP networks collect the second-order statistics of the holistic image for enhancing non-linear capability of deep networks.

3. Global Second-order Pooling Network

We illustrate the proposed GSoP-Net in Figure 1a. Note that the second-order pooling block we designed can be conveniently inserted after any convolutional layer. By introducing this block in intermediate layers, we can model high-order statistics of the holistic image at early stages, having ability to enhance non-linear modeling capability of deep ConvNets.

In practice, we build two network architectures. With GSoP blocks in-between network and at the end of network, we can use GSoP block as well which is followed by the common global average pooling, producing the mean vector as compact image representation, which we call GSoP-Net1. Alternatively, at the end of network, we can adopt matrix power normalized covariance matrices as image representations [23], called GSoP-Net2, which is more discriminative yet is high-dimensional.

3.1. GSoP Block

Figure 1b shows the diagram of the key module of our network, i.e., GSoP block. Similar to [15], the block consists of two modules, i.e., squeeze module and excitation module. The squeeze module aims to model the second-order statistics along the channel dimension of the input tensor. We are given a 3D tensor of $h' \times w' \times c'$ as an input, where h' and w' are spatial height and width and c' is the number of channels. First, we use 1×1 convolution reducing the number of channels from c' to c ($c < c'$) to decrease the computational cost of the following operations. For the $h' \times w' \times c$ tensor of reduced dimensionality, we compute pairwise channel correlations, obtaining one $c \times c$ covariance matrix. The resulting covariance matrix has clear physical meaning, i.e., its i^{th} row indicates statistical dependency of channel i with all channels. As the quadratic operations involved change the order of data, we perform row-wise normalization for the covariance matrix, respecting the inherent structural information. In contrast, the SE-Net uses global first-order pooling, which can only summarize the mean of individual channels, having limited statistical modeling capability.

In the excitation module, prior to channel scaling, we perform two consecutive operations of convolution plus

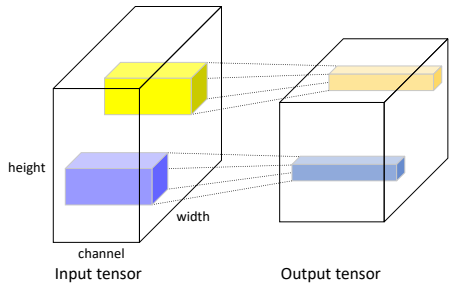


Figure 2: Classical convolutional operations fail to capture holistic dependency of 3D tensor due to limited receptive field size. For example, the data in small blue tensor cannot interact with that of yellow tensor at distant position due to limited receptive filed size. Our GSoP-Net addresses this by modeling pairwise correlations of the holistic tensor.

non-linear activation for covariance matrix embedding. To maintain the structural information, we perform row-wise convolution for the covariance matrix by regarding each row as a group in group convolution [20]. Then we perform the second convolution and this time we use the sigmoid function as a nonlinear activation, outputting a $c \times 1$ weight vector. We finally multiply each channel of input tensor by the corresponding element in the weight vector. Individual channels are thus emphasized or suppressed in a soft manner in terms of the weights.

3.2. Extension to Spatial Position

In previous section, we describe global second-order pooling along channel dimension, which we call *channel-wise GSoP*. We can extend it to spatial position, called *position-wise GSoP*, capturing pairwise feature correlations of the holistic tensor for position-wise feature scaling. The design philosophy of the position-wise GSoP Block is very similar to that of the channel-wise one. We also use 1×1 convolution for reducing the number of channels. Furthermore, as we are to compute pairwise correlations of features at all spatial positions, we adopt downsampling, decreasing the spatial size to fixed $h \times w$. So we obtain a position-wise covariance matrix of $hw \times hw$. Row i of the covariance matrix, where $i = 1, \ldots, hw$ enumerate all spatial positions, indicates statistical correlation of the i^{th} feature with all features. The position-wise covariance matrix is also fed to two consecutive operations, i.e., row-wise convolution and convolution followed by sigmoid. After appropriate reshaping, we can obtain an $h \times w$ weight matrix which encodes nonlinear pair-wise dependency among features at all positions. At last, the weight matrix is upsampled to $h' \times w'$ and then multiplied position-wise with spatial features.

3.3. Mechanism of GSoP Block

In classical deep ConvNets, restricted by limited receptive field size, the convolution operations can only process a local neighborhood of 3D tensor. The data at distant position cannot interact, e.g., the small blue tensor and the small yellow one as shown in Figure 2. The long-range dependencies can only be captured by larger receptive fields produced by deep stacking of convolutional operations. This leads to several downsides such as optimization difficulty and modeling difficulty of multi-hop dependency [35].

By computing all pairwise feature correlations (or inner product), the non-local operation can capture dependency of features at distant positions. As a result, the non-local operation can excite significant features, which is consistent with self-attention machinery [32]. Our *position-wise GSoP* multiplies each feature with one weight, which encodes nonlinear correlations of this feature with features at all positions. As such, our position-wise GSoP can also model long-range dependency of features, functioning as a kind of spatial self-attention. Beyond that, our *channel-wise GSoP* can capture long-range dependency along channel dimension, steering self-attention to significant channels. Note that SE-Net can capture long-range channel dependency as well, which, however, can model only the first-order statistical dependency, having limited representation capability.

3.4. Block Implementation

Our blocks can be conveniently inserted into ResNet architecture. The ResNet contains 4 residual stages, i.e., conv2_x, . . ., conv5_x, each containing stacks of bottleneck blocks. The exception is the first stage (i.e., conv1) which only contains one single convolutional layer, without bottleneck structure. To simplify block design and to tradeoff between computational complexity and classification accuracy, we adopt fixed size covariance matrices for all residual stages. In practice, we reduce the number of channel to 128 for both channel-wise and position-wise GSoP; in addition, we set the size of spatial covariance matrix to 64 (i.e., h=w=8). We note that the value of covariance matrix size is evaluated in Section 4.1.

After the 1×1 convolution for dimensionality reduction of channels, we perform downsampling for position-wise GSoP to obtain feature maps of fixed size (i.e., 8×8). By reshaped to a 3D tensor with first dimension being singleton, the $d\times d$ covariance matrix can be seen as $1\times d$ feature map with d channels, and so row-wise BN and row-wise group convolutions [20] can be easily accomplished. The channel number after the row convolution is raised to $4d$ and $4hw$ for channel-wise pooling and position-wise pooling, respectively. The size of weight vector for channel-wise pooling or weight matrix for position-wise pooling, should match the input tensor size. We mention that after the proposed blocks, we also use a shortcut connection, adding the input tensor to the scaled, output one. In Table 2, we present implementation of GSoP block for conv4_x.

	channel-wise GSoP		position-wise GSoP	
layers	3D filter	output tensor	3D filter	output tensor
conv + BN + ReLU	1×1×1024 G=1	14×14×128	1×1×1024 G=1	14×14×128
down sampling	–	–	–	8×8×128
COV pool+BN	–	128×128→ 1×128×128	–	64×64→ 1×64×64
row-wise conv	1×128×1 G=128	1×1×512	1×64×1 G=64	1×1×256
conv + sigmoid	1×1×512 G=1	1×1×1024	1×1×256 G=1	1×1×64→ 8×8×1
up sampling	–	–	–	14×14×1
scaling	–	14×14×1024	–	14×14×1024
parameters (M)	0.72		0.16	
MFLOPs	28.1		26.2	

Table 2: GSoP blocks for conv4_x. 'G' indicates #group convolutions [20], in which G=1 indicates common convolution (no group); gray text indicates reshape operation. Shortcut connections are added after GSoP blocks.

4. Experiments

In this section, we first conduct ablation analysis of the proposed GSoP-Nets. We then make comparison with the competing methods as well as state-of-the-arts on ImageNet. We finally evaluate generalization capability of our network to small-scale classification. All of our program are implemented under the PyTorch framework, and runs on four workstations each of which is equipped with 2 GTX 1080Ti GPUs and an Intel i7-4790K@4GHz CPU.

Datasets. Our experiments are mainly conducted on ImageNet-1K [5] benchmark. The ImageNet-1K contains 1.28M training images and 50K validation images from 1,000 classes. In Section 4.1, for the purpose of faster ablation study, we build a small subset of ImageNet-1K by randomly selecting 250 classes, including 320K/12.5K images for training/validation, which we call ImageNet-$\frac{1}{4}$K. For comparison with state-of-the-art networks, we adopt standard ImageNet-1K in Section 4.2. To evaluate the generalization capability of our network, we also make experiments on CIFAR-100 benchmark [19], which contains 60K color images of 32x32 pixels from 100 categories, with 50K images for training and 10K images for testing.

Experimental Setting. During training from scratch with ResNet architecture on ImageNet, we follow [11] for data augmentation involving scale, color and flip jittering. The weights are initialized as in [10]. We randomly crop 224×224 images from the rescaled images with per-channel mean subtraction. The networks are optimized using stochastic gradient descent (SGD) with a weight decay of 1e-4, a momentum of 0.9 and a mini-batch of 160. The initial learning rate is set to 0.1, divided by 10 every 30 epochs until 100 epochs, unless specified otherwise. During testing stage, we evaluate the error on the single 224×224 center crop from an image whose shorter size is 256.

For training from scratch on CIFAR-100, following [12], we use standard data augmentation of horizontal flip and random translation. The networks are trained within 110 epochs with the initial learning rate of 0.25, which is reduced to 0.025 and 0.0025 at the 80^{th} and 95^{th} epoch, respectively. The weight decay and momentum are same with those on ImageNet while the mini-batch size is 128.

	output	layer
conv1	112×112	conv, 7×7, 64, Stride=2
pool1	56×56	max pool, 3×3, Stride=2
conv2_x		$\begin{bmatrix}\text{conv}, 1\times1, 64\\ \text{conv}, 1\times1, 64\\ \text{conv}, 1\times1, 256\end{bmatrix}\times2$ GSoP Block
conv3_x	28×28	$\begin{bmatrix}\text{conv}, 1\times1, 128\\ \text{conv}, 1\times1, 128\\ \text{conv}, 1\times1, 512\end{bmatrix}\times2$ GSoP Block
conv4_x	14×14	$\begin{bmatrix}\text{conv}, 1\times1, 256\\ \text{conv}, 1\times1, 256\\ \text{conv}, 1\times1, 1024\end{bmatrix}\times2$ GSoP Block
conv5_x	14×14	$\begin{bmatrix}\text{conv}, 1\times1, 512\\ \text{conv}, 1\times1, 512\\ \text{conv}, 1\times1, 2048\end{bmatrix}\times2$
	1×1	GSoP block+GAvP, 2K or iSQRT-COV [22], 32K
	1×1	FC + softmax

Table 3: GSoP-Net with ResNet-26 architecture.

4.1. Ablation Analysis on GSoP-Nets

We develop a lightweight residual network of 26 layers (i.e., ResNet-26) as our baseline architecture, where every residual stage contains two bottlenecks. For conv2_x∼conv4_x, we insert one GSoP block per residual stage. For *GSoP-Net1* we insert one GSoP block after the last residual stage, followed by global average pooling, outputting a 2K-dimensional image representation fully connected to softmax layer; for *GSoP-Net2*, instead, we use matrix power normalized covariance pooling, producing 32K-dimensional image representation. As in [23, 22], we do not perform downsampling at the last residual stage to alleviate the problem of small sample size. Table 3 presents the architecture of our GSoP-Nets.

Impact of Covariance Size. The covariance matrices, produced by the second-order pooling blocks, encode the statistical correlation of the holistic tensors, playing a central role in our networks. So we first evaluate impact of covariance matrix size on the proposed networks. Table 4a summarizes the results, in which the top and middle panel

		top-1 err/top-5 err	
		GSoP-Net1	GSoP-Net2
channel-wise cov size c	64×64	18.00/4.99	16.84/4.58
	128×128	**17.42/4.53**	**16.68/4.36**
	256×256	17.61/4.64	16.67/4.18
position-wise cov size hw	36×36	19.21/5.46	17.34/4.80
	64×64	**18.37/5.05**	**17.18/4.80**
	144×144	18.41/5.08	17.51/4.63
vanilla network		19.18/5.62	

(a) Impact of covariance matrix size.

		top-1 err/top-5 err	
		GSoP-Net1	GSoP-Net2
channel-wise pool		**17.42/4.53**	**16.68/4.36**
position-wise pool		18.37/5.05	17.18/4.80
fusion	average	17.90/4.73	16.77/4.36
	maximum	**17.48/4.52**	16.80/4.39
	concatenation	17.58/4.61	**16.49/4.35**

(b) Comparison of fusion schemes.

[S2, S3, S4, S5]	top-1 err	top-5 err
[−, −, −, −]	19.18	5.62
[C, −, −, −]	18.45	5.22
[−, C, −, −]	18.72	5.33
[−, −, C, −]	18.85	5.24
[−, −, −, C]	18.33	5.12
[C, C, C, C]	17.42	4.53
[−, −, −, √]	17.43	4.71
[C, C, C, √]	**16.68**	**4.36**

(c) Single block performance.

2 blocks	error	3 blocks	error	#blocks:4→1	error
[C, C, −, −]	18.05/5.22	[C, C, C, −]	17.54/4.67	[C, C, C, C]	**17.42/4.53**
[−, C, C, −]	18.29/4.86	[−, C, C, C]	17.54/4.79	[−, C, C, C]	17.54/4.79
[−, −, C, C]	18.09/4.81	[C, −, C, C]	17.64/4.89	[−, −, C, C]	18.09/4.81
[C, −, −, C]	18.01/4.99	[C, C, −, C]	17.90/4.97	[−, −, −, C]	18.33/5.12

(d) Top-1/top-5 errors (%) of varying number of blocks.

Table 4: Ablation results of our GSoP-Nets with ResNet-26 architecture on ImageNet-$\frac{1}{4}$K.

shows the impacts using channel-wise (cov size: $c \times c$) and position-wise pooling (cov size: $hw \times hw$), respectively. We first observe that, whatever the second-order pooling, the proposed networks improve over vanilla ResNet-26, demonstrating that our holistic modeling methods in earlier stages are beneficial in enhancing the network's discriminative capability. For channel-wise second-order pooling, relative to varying values of c, GSoP-Net1 achieves the best results with $c = 128$. The errors of GSoP-Net2 consistently decline as c gets larger and the lowest error is obtained with $c = 256$. For position-wise second-order pooling, GSoP-Net1 with $hw = 64$ produces the lowest errors. Notably, for either channel-wise or position-wise pooling, it is clear that GSoP-Net2 performs much better than GSoP-Net1, which suggests that image representation of covariance matrix is superior to that of mean vector by average pooling.

Fusion of Channel- and Position-wise Pooling. The channel-wise and position-wise second-order pooling capture statistical correlations from different dimensions of 3D tensor. They can be combined for holistic image modeling. Given an input tensor, we independently perform second-order pooling along the channel dimension and spatial dimension, producing two output tensors. We can fuse the two output tensors by the commonly used operations of average/maximum and concatenation. As concatenation operation increases tensor size, we use one convolutional layer for maintaining the original tensor size.

The results of fusion methods are presented in Table 4b. For GSoP-Net1, the average scheme performs worse than the other two, while the maximum scheme is slightly better than the concatenation one. For GSoP-Net2, the concatenation scheme is a little superior to the other two schemes. However, compared to separate channel-wise pooling, with any fusion scheme, combination of position-wise pooling brings little improvement. These results suggest that the two kinds of second-order pooling methods are not complementary, though the two proposed networks individually have obvious improvement over the vanilla network.

Performance of Single Second-order Block. In this part, we analyze the performance of single channel-wise block separately added to different residual stage. We make no analysis on position-wise pooling as it is inferior to the channel-wise one. Table 4d presents the results, where Si denotes residual stage i, $i = 2, \ldots, 5$; −, C and √ denote no second-order block, one second-order block and iSQRT-COV meta layer [23] inserted at the corresponding residual stage, respectively. It can be seen that insertion of single block into any residual stage brings comparable improvement over the vanilla network, suggesting that the second-order block at different stage makes similar contribution to the overall GSoP-Net1. The iSQRT-COV, which inserts a matrix normalized covariance matrix at residual stage 5 as the final image representation, is a strong baseline, even achieving comparable result with GSoP-Net1. The GSoP-Net2, which essentially amounts to insertion of global second-order pooling at intermediate stages of iSQRT-COV network, leading to further, non-trivial improvement. This suggests the benefit of introducing second-order statistics in earlier layers of networks.

Results of Varying Number of Second-order Blocks. Table 4d shows the results of varying number of channel-wise second-order blocks inserted at different residual stages. It can be seen that overall the networks with identical number of second-order blocks produce comparable results.

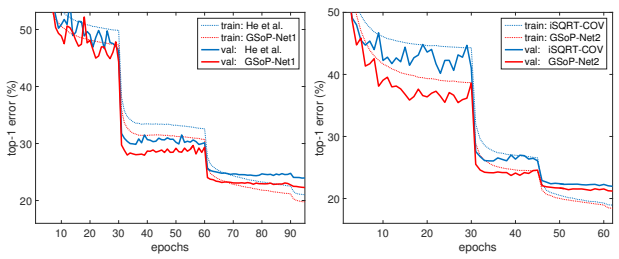


Figure 3: Convergence curves of our GSoP-Nets under ResNet-50 architecture. Left: GSoP-Net1 vs vanilla network [11]; right: GSoP-Net2 vs iSQRT-COV [22].

The performance consistently improves as the number of second-order blocks increase. With initial 4 second-order blocks, gradual block removal from higher stage to lower stage results in consistent performance decline; similar phenomenon can be observed for block removal from opposite direction and the corresponding results are not presented due to page limit.

4.2. Results on ImageNet-1K

In this subsection, we further evaluate our proposed GSoP-Nets on standard ImageNet-1K under ResNet-50 architecture. We insert a GSoP block for residual stage 2, 3 and 4, respectively. For GSoP-Net1, we insert one GSoP block for residual stage 5, followed by the commonly used global average pooling; for GSoP-Net2, instead of the GSoP block, the meta-layer of iSQRT-COV [22] is inserted.

4.2.1 Convergence and Network Complexity

Convergence. Figure 3 illustrates the convergence curves of our GSoP-Net. For GSoP-Net1, though second-order statistical modeling is exploited, it is for tensor (convolutional features) scaling while the image representation is first order, just like the original ResNet-50. As shown in the left figure, the convergence behavior of GSoP-Net1 is similar to that of ResNet-50, but consistently has lower validation error throughout the training process. Different from iSQRT-COV, for GSoP-Net2 we introduce second-order blocks for residual stages 1,2 and 3. From the right figure, we can see that GSoP-Net2 inherits fast convergence property of iSQRT-COV, while steadily performs better. We attribute the improvement of our networks over their counterparts to the holistic modeling of second-order statistics introduced in earlier stages.

Network Complexity. Table 5 shows comparison of parameter and computation. The number of parameters of GSoP-Net1 is comparable to that of the vanilla ResNet-50, while GSoP-Net2 has nearly doubled the number of parameters. The increased parameters in GSoP-Net2 are mainly due to FC layer, in which dimensionality of image representation is 32K, accounting for most increase of the total

	description	top-1	top-5	params/GFLOPs
He et al. [11]	Baseline network	23.85	7.13	25.5M/3.86
FBN [24]	Quadratic transformations	24.0	7.1	–
SORT [37]		23.82	6.72	–
MPN-COV [23]	GSoP at network end	22.74	6.54	2.2×/1.6×
iSQRT-COV [22]		22.14	6.22	2.2×/1.6×
SE-Net [15]	GAvP across network	23.29	6.62	1.1×/1.0×
GENet [13]		21.88	5.80	1.3×/1.0×
CBAM [38]		22.66	6.31	1.1×/1.0×
GSoP-Net1 (ours)	GSoP across network	22.02	5.88	1.1×/1.6×
GSoP-Net2 (ours)		**21.19**	**5.64**	2.3×/1.7×
ResNeXt [40]	Modified architectures upon ResNet	22.11	5.90	1.0×/1.0×
DropBlock [8]		21.87	5.98	1.0×/1.0×
DRN-A-50 [41]		22.94	6.57	1.0×/4.9×

Table 5: Comparison (%) of different methods with ResNet-50 architecture on ImageNet-1K.

parameters, just like MPN-COV [23] and iSQRT-COV [22]. Note that advances on model compression, e.g., [6, 28, 9], has potential to significantly reduce the number of parameters, particularly in FC layer, while maintaining the performance. In practice, we can exploit such techniques to reduce parameters. Analogous to [23, 22], the GFLOPs of our networks are 1.58x of the number of vanilla ResNet. The computations increased are attributed to removal of downsampling in the last residual stage, so that feature map size doubles. This operation is helpful for robust covariance estimation by alleviating the problem of small sample and high dimensionality [23]. This somewhat slowdowns the training, however, while making little difference for inference. With a single GTX 1080Ti GPU with CUDA 9.0 and CuDNN 7.1, the inference time (ms) per image are 2.52 vs 2.68/2.84 (vanilla ResNet-50 vs GSoP-Net1/GSoP-Net2).

4.2.2 Comparison with Competing Networks.

Table 5 compares classification errors between our GSoP-Nets and the competing networks on ImageNet-1K.

Comparison with FBN and SORT. The two works [24, 37] are among the first which introduce quadratic transformation, instead of just linear convolutions, throughout a network. However, compared to the vanilla network, their performance gains are not significant. In contrast, our networks are much better, achieving over 2.8% and 2.6% higher accuracies than FBN and SORT. This comparison demonstrates that, by making favorable use of higher-order information, we can greatly improve the network performance.

Comparison with Global Cov Pool at Network End. Here we compare our GSoP-Net2 with several methods where global second-order pooling is inserted only at the end of network. All of them estimate covariance matrices of the last convolutional features as image representa-

tions. DeepO$_2$P computes matrix logarithm for covariance matrix while B-CNN performs element-wise power normalization plus ℓ_2 normalization. As DeepO$_2$P and B-CNN are not competitive for large-scale visual recognition [23], here we do not compare with them. MPN-COV uses structured normalization by matrix square root, and iSQRT-COV is a faster version of MPN-COV, in which matrix square root is based on iterative algorithm, rather than GPU unfriendly SVD. Our GSoP-Net2 outperforms MPN-COV by 1.55% in top-1 error (0.90% in top-5 error). Compared to iSQRT-COV, the GSoP-Net2 achieves 0.95%/0.58% lower top-1/top-5 error rates, while resulting in negligible overhead. We note that the iSQRT-COV is a strong baseline and our improvement is nontrivial. The comparison between our GSoP-Net2 and MPN-COV/iSQRT-COV indicates that introducing higher-order statistics in earlier stages can enhance representational learning capability of deep ConvNets.

Comparison with Global Avg Pool across Network. From Table 5, we can see that our GSoP-Net1 performs 1.3%/0.7% better than SE-Net in top-1/top-5 errors. As an extension of SE-Net, CBAM combines global average and max pooling along both channel dimensional and spatial dimension. Nevertheless, the error rates of GSoP-Net1 are lower than CBAM. Building upon SE-Net, GENet [13] proposes gather and excitation operations for exploiting context information. Our GSoP-Net2 outperforms GENet by a non-trivial margin. These comparisons between our networks and SE-Net and its variants show that higher-order modeling is able to capture richer statistics than the first-order modeling, leading to more discriminative representation. Notably, we do not insert GSoP block after each bottleneck structure; instead, we only insert the GSoP block per residual stage. As a result, we only add no more than 4 GSoP blocks, and more GSoP blocks may further improve the performance of our network.

Comparison with State-of-the-arts. Finally, we compare with several state-of-the-art networks which modify upon ResNet-50 architecture. Compared to ResNet, ResNeXt [40] considerably increases network width, which, however, keeps parameters and computation almost unchanged through an extensive use of group convolutions [20]. DRN-A-50 [41] removes downsampling in residual stage 3 and 4, and meanwhile uses dilated convolution to maintain the receptive size. DropBlock [8] extends dropout technique to convolution; by drop blocks of feature map randomly, it maintains the context integrity during training. As shown in Table 5, these modified networks performs much better than ResNet-50. Nevertheless, our GSoP-Net2 outperforms all of them by a non-trivial margin. It is noteworthy to mention that, if built upon the modified networks above, the performance of our network may improve further.

model	top-1 err (%)	params	GFLOPs
He et al [12]	24.33	1.7M	0.25
SE-Net [14]	21.31	1.9M	0.29
CMPE [16]	22.35	2.0M	N/A
iSQRT-COV [22]	19.95	2.5M	0.52
GSoP-Net1 (ours)	20.86	2.9M	0.55
GSoP-Net2 (ours)	**18.58**	3.6M	0.58

Table 6: Comparison (%) of our networks with the counterparts on CIFAR-100.

4.3. Results on CIFAR-100

This section conducts experiments on CIFAR-100 [19] to evaluate the generalization capability of the proposed GSoP-Net. The backbone network is pre-activation ResNet-164 [12], containing 3 residual stages each of which contains 18 bottlenecks. In GSoP-Net1, we insert 18 GSoP blocks into the backbone network uniformly, and in GSoP-Net2 the last GSoP block is replaced by a meta-layer of iSQRT-COV. Downsampling is not performed in the last residual stage. The final dimension of image representation in GSoP-Net2 is 8K and a dropout layer (dropout rate=0.5) is used for FC layer. The covariance size is 64×64 in both GSoP-Net1 and GSoP-Net2.

The experimental results on CIFAR-100 are presented in Table 6. Compared with the vanilla network, GSoP-Net1 and GSoP-Net2 obtain gains of 3.47% and 5.75%, respectively, improving the performance by a large margin. CMPE [16] implements channel-wise excitation operation by establishing the correlation of the channel-wise representation between two nearby bottlenecks, which can be considered as a cross-block version of SE-Net. GSoP-Net1 performs better than SE-Net and CMPE by 0.45% and 1.49% respectively. iSQRT-COV is very competitive, outperforming SE-Net by $\sim$1.36%. By introducing second-order statistics in earlier stages, our GSoP-Net2 makes further improvement ($\uparrow$ 1.37%) over iSQRT-COV.

5. Conclusion

We presented a simple yet effective method for capturing holistic statistical correlations throughout a deep convolutional neural network. By exploiting global second-order statistics at earlier stages, the proposed method can learn more discriminative representations. As far as we know, our work is among the first which introduce higher-order pooling into intermediate layers of deep networks. Our proposed networks performs better than SE-Net [15], i.e., the first-order counterpart, while non-trivially improves state-of-the-art iSQRT-COV [22] which plugged global covariance pooling as image representation only at network end. The proposed GSoP blocks can be conveniently plugged into other deep architectures, e.g., Inception [31] and DenseNet [17], which will be our future work.

References

[1] S. Cai, W. Zuo, and L. Zhang. Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. In *ICCV*, 2017. 1, 2

[2] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 1

[3] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie. Kernel pooling for convolutional neural networks. In *CVPR*, 2017. 1, 2

[4] X. Dai, J. Yue-Hei Ng, and L. S. Davis. FASON: First and second order information fusion network for texture recognition. In *CVPR*, 2017. 2

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[6] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*, 2014. 7

[7] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. In *CVPR*, 2016. 1, 2

[8] G. Ghiasi, T.-Y. Lin, and Q. V. Le. Dropblock: A regularization method for convolutional networks. In *NIPS*, 2018. 7, 8

[9] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In *NIPS*, 2015. 7

[10] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *ICCV*, 2015. 5

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 5, 7

[12] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 5, 8

[13] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *NIPS*, 2018. 7, 8

[14] J. Hu, L. Shen, A. Samuel, S. Gang, and W. Enhua. Squeeze-and-excitation networks. *arXiv:1709.01507v3*, 2018. 8

[15] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 1, 2, 3, 7, 8

[16] Y. Hu, G. Wen, M. Luo, D. Dai, and M. Jiajiong. Competitive inner-imaging squeeze and excitation for residual network. *arXiv:1807.08920*, 2018. 8

[17] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2, 8

[18] C. Ionescu, O. Vantzos, and C. Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *ICCV*, 2015. 1, 2

[19] A. Krizhevsky. Learning multiple layers of features from tiny images. *Tech. Rep.*, 2009. 5, 8

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 2, 4, 5, 8

[21] P. Li, Q. Wang, H. Zeng, and L. Zhang. Local Log-Euclidean multivariate Gaussian descriptor and its application to image classification. *IEEE TPAMI*, 2017. 2

[22] P. Li, J. Xie, Q. Wang, and Z. Gao. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *CVPR*, 2018. 1, 2, 5, 7, 8

[23] P. Li, J. Xie, Q. Wang, and W. Zuo. Is second-order information helpful for large-scale visual recognition? In *ICCV*, Oct 2017. 1, 2, 3, 5, 6, 7, 8

[24] Y. Li, N. Wang, J. Liu, and X. Hou. Factorized bilinear models for image recognition. In *ICCV*, 2017. 1, 2, 7

[25] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014. 1, 2

[26] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear CNN models for fine-grained visual recognition. In *ICCV*, 2015. 1, 2

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[28] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, 2016. 7

[29] J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. In *CVPR*, 2017. 1

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2, 8

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*. 2017. 4

[33] H. Wang, Q. Wang, M. Gao, P. Li, and W. Zuo. Multi-scale location-aware kernel representation for object detection. In *CVPR*, 2018. 1

[34] Q. Wang, P. Li, and L. Zhang. G^2DeNet: Global Gaussian distribution embedding network and its application to visual recognition. In *CVPR*, 2017. 2

[35] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, 2018. 1, 2, 4

[36] Y. Wang, M. Long, J. Wang, and P. S. Yu. Spatiotemporal pyramid network for video action recognition. In *CVPR*, 2017. 1

[37] Y. Wang, L. Xie, C. Liu, S. Qiao, Y. Zhang, W. Zhang, Q. Tian, and A. Yuille. SORT: Second-order response transform for visual recognition. In *ICCV*, 2017. 1, 2, 7

[38] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 2, 7

[39] H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang. Monet: Deep motion exploitation for video object segmentation. In *CVPR*, 2018. 2

[40] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 7, 8

[41] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. In *CVPR*, 2017. 7, 8

[42] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 1

[43] G. Zoumpourlis, A. Doumanoglou, N. Vretos, and P. Daras. Non-linear convolution filters for CNN-based learning. In *ICCV*, 2017. 1, 3