Supplementary material for Analyzing Filters Toward Efficient ConvNet

Takumi Kobayashi
National Institute of Advanced Industrial Science and Technology, Japan
takumi.kobayashi@aist.go.jp

A. Orthonormal Steerable Filter

We present the practical algorithm for computing the orthonormal steerable filters that are bases for convolution filters (Analysis 1, Sec. 2). The orthonormality is embedded into the steerable filter [3] by means of Gram-Schmidt method as shown in line 6 of Algorithm 1. As described in Sec. 2, our bases of N-th order are based on the steerable filters of *up to* N-th order derivatives, thereby producing $\frac{1}{2}(N+1)(N+2)$ basis filters in total.

Algorithm 1 : Orthonormal steerable basis filters

Input: N: Derivative order, σ: Standard deviation of Gaussian envelope
r: filter (reach) size to produce $\mathbb{D} = \{(x,y) \in \{-r, \cdots, r\} \times \{-r, \cdots, r\}\}$

1: n-th order Gaussian derivative function: $\mathbf{g}^{[n]}(x,y) \triangleq \left(\frac{\partial}{\partial x}\right)^n e^{-\frac{1}{2\sigma^2}(x^2+y^2)}$
2: Initial basis set: $\mathcal{B} = \emptyset$ (empty)
3: **for** $n = 0$ to N **do**
4: n-th order steerable filters: $\tilde{\boldsymbol{b}}_i^{[n]} \xleftarrow{\mathbb{D}} \mathbf{g}^{[n]}(\cos(\theta_i)x + \sin(\theta_i)y, -\sin(\theta_i)x + \cos(\theta_i)y)$, where $\theta_i = \frac{i\pi}{n+1}$, $i \in \{0, \cdots, n\}$, and $\xleftarrow{\mathbb{D}}$ means discretizing a function on $\mathbb{D}$ and then applying L_2-normalization to produce $(2r+1)\times(2r+1)$ filter $\tilde{\boldsymbol{b}}_i^{[n]}$
5: **for** $i = 0$ to n **do**
6: Orthonormalize by $\mathcal{B}$: $\boldsymbol{b}_i^{[n]} = \texttt{orthnorm}_{\mathcal{B}}(\tilde{\boldsymbol{b}}_i^{[n]})$ such that $\boldsymbol{b}_i^{[n]} \perp \mathcal{B}$, $\|\boldsymbol{b}_i^{[n]}\|_F^2 = 1$
7: $\mathcal{B} \leftarrow \mathcal{B} \cup \{\boldsymbol{b}_i^{[n]}\}$
8: **end for**
9: **end for**

Output: $\mathcal{B}$: Orthonormal basis set which contains $|\mathcal{B}| = \frac{1}{2}(N+1)(N+2)$ basis filters

B. Pre-trained Networks For Analyzing Convolution Filters

In Sec. 2, we analyze the optimized convolution filters sampled from the pre-trained ConvNets which are listed in Table A. Those models except for VGG-M$_{9\times9}$ are downloaded from [1]. Here, we construct VGG-M$_{9\times9}$ by replacing the first convolution layer of 7×7 in VGG-M with 9×9 convolution since the filter size of 9×9 is not found in any other pre-trained ConvNet models. The detailed network architecture of VGG-M$_{9\times9}$ and the learning parameters are shown in the next section.

Table A. Number of convolution layers in pre-trained ConvNets which are downloaded from [1] except for VGG-M$_{9\times9}$.

	Convolution size				
ConvNet	3×3	5×5	7×7	9×9	11×11
VGG-F [2]	3	1	0	0	1
VGG-M [2]	3	1	1	0	0
VGG-M$_{9\times9}$	3	1	0	1	0
VGG-S [2]	3	1	1	0	0
AlexNet [7]	3	1	0	0	1
Caffe-reference [6]	3	1	0	0	1
VGG-vd-16 [9]	13	0	0	0	0
VGG-vd-19 [9]	16	0	0	0	0
GoogLeNet [11]	10	9	1	0	0
ResNet-50 [4]	16	0	1	0	0
ResNet-101 [4]	33	0	1	0	0
ResNet-152 [4]	50	0	1	0	0

C. Network Architecture

This section details the ConvNets that we use in the experiments. The ConvNets include AlexNet [7] and VGG-S/M/F [2] as well as the deeper ConvNets of VGG-vd-16/19 [9] and ResNet-50 [4], all of which are trained *from scratch* on ILSVRC2014 training dataset.

All the ConvNets are implemented by using the MatConvNet toolbox[1] [12], and we apply SGD with momentum to train them by following the default leaning parameter values suggested in the toolbox as shown in Table B; the learning rate is decreased constantly in log-scale at every epoch for AlexNet, VGG-S/M/F and VGG-vd-16/19 and at every 30 epochs for ResNet-50.

Table B. Learning parameters. We follow the default parameter values suggested in the MatConvNet toolbox except for the mini-batch size of VGG-vd-16/19. $\lceil\cdot\rceil$ indicates a ceiling function.

	AlexNet	VGG-F	VGG-M	VGG-S	VGG-vd-16/19	ResNet-50
mini-batch size	256	256	196	128	64	256
number of epoch	20					90
learning rate	$10^{-\frac{16+3t}{19}}$, $t\in\{1,\cdots,20\}$					$10^{-\lceil\frac{t}{30}\rceil}$, $t\in\{1,\cdots,90\}$
momentum	0.9					
weight decay	0.0005					0.0001

The architectures of the ConvNets are detailed in Table C∼L. The moderately deep ConvNets of AlexNet [7] and VGG-S/M/F [2] are shown in Table C, while the deeper ConvNets of VGG-vd-16/19 [9] and ResNet-50 [4] are respectively shown in Table E and Table G. Note that in AlexNet (Table C), we apply BatchNormalization (BN) [5] instead of DropOut [10] and thereby remove the layers of local response normalization. Additionally, VGG-M$_{9\times9}$ and VGG-M$_{11\times11}$, the variants of VGG-M, are also detailed in Table D. In those tables, we indicate by bold fonts the convolution layers to which Analysis 1 can be applied; the ConvNets are re-parameterized at those convolution layers by replacing the filter weights with the coefficients of the bases as trainable parameters. The fully-connected layer (fc6) that Analysis 2 focuses on is also highlighted by gray cell color. Table F&H∼L show the improved ConvNets by applying our methods.

[1] http://www.vlfeat.org/matconvnet/, vertion 1.0-beta23.

Table C. Architectures of moderately deep ConvNets [7, 2]. The conventional layer names are shown in the left-most column with underline. The first row shows the input image sizes, while the others indicate the parameters either of convolution or max-pooling. The convolution layer (Conv) is followed by BatchNormalization (BN) [5] and rectified linear unit (ReLU) [8]. We highlight by bold fonts the convolution to which Analysis 1 is applied, and indicate by gray cell color the fully-connected layer devoted to Analysis 2.

		AlexNet [7]	VGG-F [2]	VGG-M [2]	VGG-S [2]
	input	$[227\times227\times3]$	$[224\times224\times3]$	$[224\times224\times3]$	$[224\times224\times3]$
conv1	**Conv** BN ReLU	$[\mathbf{11}\times\mathbf{11}\times3\times96]$ stride:4, pad:0	$[\mathbf{11}\times\mathbf{11}\times3\times64]$ stride:4, pad:0	$[\mathbf{7}\times\mathbf{7}\times3\times96]$ stride:2, pad:0	$[\mathbf{7}\times\mathbf{7}\times3\times96]$ stride:2, pad:0
	max-pool	$[3\times3]$ stride:2, pad:0	$[3\times3]$ stride:2, pad:[0,1,0,1][b]	$[3\times3]$ stride:2, pad:0	$[3\times3]$ stride:3, pad:[0,2,0,2][b]
conv2	**Conv** BN ReLU	$[\mathbf{5}\times\mathbf{5}\times48\times256]$[a] stride:1, pad:2	$[\mathbf{5}\times\mathbf{5}\times64\times256]$ stride:1, pad:2	$[\mathbf{5}\times\mathbf{5}\times96\times256]$ stride:2, pad:1	$[\mathbf{5}\times\mathbf{5}\times96\times256]$ stride:1, pad:0
	max-pool	$[3\times3]$ stride:2, pad:0	$[3\times3]$ stride:2, pad:0	$[3\times3]$ stride:2, pad:[0,1,0,1][b]	$[2\times2]$ stride:2, pad:[0,1,0,1][b]
conv3	**Conv** BN ReLU	$[\mathbf{3}\times\mathbf{3}\times256\times384]$ stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times256\times256]$ stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times256\times512]$ stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times256\times512]$ stride:1, pad:1
conv4	**Conv** BN ReLU	$[\mathbf{3}\times\mathbf{3}\times192\times384]$[a] stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times256\times256]$ stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times512\times512]$ stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times512\times512]$ stride:1, pad:1
conv5	**Conv** BN ReLU	$[\mathbf{3}\times\mathbf{3}\times192\times256]$[a] stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times256\times256]$ stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times512\times512]$ stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times512\times512]$ stride:1, pad:1
	max-pool	$[3\times3]$ stride:2, pad:0	$[3\times3]$ stride:2, pad:0	$[3\times3]$ stride:2, pad:0	$[3\times3]$ stride:3, pad:[0,1,0,1][b]
fc6	Conv BN ReLU	$[6\times6\times256\times4096]$ stride:1, pad:0	$[6\times6\times256\times4096]$ stride:1, pad:0	$[6\times6\times512\times4096]$ stride:1, pad:0	$[6\times6\times512\times4096]$ stride:1, pad:0
fc7	Conv BN ReLU	$[1\times1\times4096\times4096]$ stride:1, pad:0	$[1\times1\times4096\times4096]$ stride:1, pad:0	$[1\times1\times4096\times4096]$ stride:1, pad:0	$[1\times1\times4096\times4096]$ stride:1, pad:0
fc8	Conv	$[1\times1\times4096\times1000]$ stride:1, pad:0	$[1\times1\times4096\times1000]$ stride:1, pad:0	$[1\times1\times4096\times1000]$ stride:1, pad:0	$[1\times1\times4096\times1000]$ stride:1, pad:0
	SoftMax				

[a] The convolution filter covers subset of input channels; refer to [7]. [b] It means the padding on [left, right, top, bottom].

Table D. ConvNet architectures for the variants of VGG-M [2]. These ConvNets are constructed by replacing only the first 7×7 convolution of VGG-M with 9×9 and 11×11, respectively, for producing 9×9 and 11×11 convolution filters used in Sec. 2.1 and Sec. 2.2.

		VGG-M$_{9\times9}$	VGG-M$_{11\times11}$
input		$[224\times224\times3]$	$[224\times224\times3]$
conv1	**Conv** BN ReLU	$[\mathbf{9}\times\mathbf{9}\times3\times96]$ stride:2, pad:1	$[\mathbf{11}\times\mathbf{11}\times3\times96]$ stride:2, pad:2
	max-pool	$[3\times3]$ stride:2, pad:0	$[3\times3]$ stride:2, pad:0
conv2	**Conv** BN ReLU	$[\mathbf{5}\times\mathbf{5}\times96\times256]$ stride:2, pad:1	$[\mathbf{5}\times\mathbf{5}\times96\times256]$ stride:2, pad:1
	max-pool	$[3\times3]$ stride:2, pad:[0,1,0,1]	$[3\times3]$ stride:2, pad:[0,1,0,1]
conv3	**Conv** BN ReLU	$[\mathbf{3}\times\mathbf{3}\times256\times512]$ stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times256\times512]$ stride:1, pad:1
conv4	**Conv** BN ReLU	$[\mathbf{3}\times\mathbf{3}\times512\times512]$ stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times512\times512]$ stride:1, pad:1
conv5	**Conv** BN ReLU	$[\mathbf{3}\times\mathbf{3}\times512\times512]$ stride:1, pad:1	$[\mathbf{3}\times\mathbf{3}\times512\times512]$ stride:1, pad:1
	max-pool	$[3\times3]$ stride:2, pad:0	$[3\times3]$ stride:2, pad:0
fc6	Conv BN ReLU	$[6\times6\times512\times4096]$ stride:1, pad:0	$[6\times6\times512\times4096]$ stride:1, pad:0
fc7	Conv BN ReLU	$[1\times1\times4096\times4096]$ stride:1, pad:0	$[1\times1\times4096\times4096]$ stride:1, pad:0
fc8	Conv	$[1\times1\times4096\times1000]$ stride:1, pad:0	$[1\times1\times4096\times1000]$ stride:1, pad:0
	SoftMax		

Table E. Deeper ConvNet architectures of VGG-vd-16/19 [9]. "$\{\ \sim\ \} \times n$" means n-times repeat of the block $\{\ \sim\ \}$, and "{ 〃 }" in VGG-vd-19 indicates the same block as that of VGG-vd-16 shown in the left.

	VGG-vd-16 [9]	VGG-vd-19 [9]
input	$[224\times224\times3]$	
conv1-1	**Conv**-BN-ReLU: $[\mathbf{3}\times\mathbf{3}\times3\times64]$, stride:1, pad:1	
conv1-2	**Conv**-BN-ReLU: $[\mathbf{3}\times\mathbf{3}\times64\times64]$, stride:1, pad:1	
	max-pool: $[2\times2]$, stride:2, pad:0	
conv2-1	**Conv**-BN-ReLU: $[\mathbf{3}\times\mathbf{3}\times64\times128]$, stride:1, pad:1	
conv2-2	**Conv**-BN-ReLU: $[\mathbf{3}\times\mathbf{3}\times128\times128]$, stride:1, pad:1	
	max-pool: $[2\times2]$, stride:2, pad:0	
conv3-1	**Conv**-BN-ReLU: $[\mathbf{3}\times\mathbf{3}\times128\times256]$, stride:1, pad:1	
conv3-⋆	{ **Conv**-BN-ReLU: $[\mathbf{3}\times\mathbf{3}\times256\times256]$, stride:1, pad:1 }$\times2$	{ 〃 }$\times3$
	max-pool: $[2\times2]$, stride:2, pad:0	
conv4-1	**Conv**-BN-ReLU:$[\mathbf{3}\times\mathbf{3}\times256\times512]$, stride:1, pad:1	
conv4-⋆	{ **Conv**-BN-ReLU: $[\mathbf{3}\times\mathbf{3}\times512\times512]$, stride:1, pad:1 }$\times2$	{ 〃 }$\times3$
	max-pool: $[2\times2]$, stride:2, pad:0	
conv5-⋆	{ **Conv**-BN-ReLU: $[\mathbf{3}\times\mathbf{3}\times512\times512]$, stride:1, pad:1 }$\times3$	{ 〃 }$\times4$
	max-pool: $[2\times2]$, stride:2, pad:0	
fc6	Conv-BN-ReLU: $[7\times7\times512\times4096]$, stride:1, pad:0	
fc7	Conv-BN-ReLU: $[1\times1\times4096\times4096]$, stride:1, pad:0	
fc8	Conv: $[1\times1\times4096\times1000]$, stride:1, pad:0	
	SoftMax	

Table F. ConvNet architectures of improved VGG-M by introducing BoW-based representation (Sec. 3.3) into the fully-connected layer (fc6) in Table C. It should be noted that bow layer (gray cell color) is introduced in bow models and the max-pooling layer after conv5 is removed in the dense-bow models. The same modification is applied to AlexNet, VGG-S/F and VGG-vd-16/19.

		VGG-M bow(avg)	VGG-M dense-bow(avg)	VGG-M bow(max)	VGG-M dense-bow(max)
input		$[224\times224\times3]$			
conv1	**Conv** BN ReLU	$[\mathbf{7}\times\mathbf{7}\times3\times96]$ stride:2, pad:0			
	max-pool	$[3\times3]$ stride:2, pad:0			
conv2	**Conv** BN ReLU	$[\mathbf{5}\times\mathbf{5}\times96\times256]$ stride:2, pad:1			
	max-pool	$[3\times3]$ stride:2, pad:[0,1,0,1]			
conv3	**Conv** BN ReLU	$[\mathbf{3}\times\mathbf{3}\times256\times512]$ stride:1, pad:1			
conv4	**Conv** BN ReLU	$[\mathbf{3}\times\mathbf{3}\times512\times512]$ stride:1, pad:1			
conv5	**Conv** BN ReLU	$[\mathbf{3}\times\mathbf{3}\times512\times512]$ stride:1, pad:1			
	max-pool	$[3\times3]$ stride:2, pad:0	——	$[3\times3]$ stride:2, pad:0	——
	bow	Conv: $[1\times1\times512\times4096]$, stride:1, pad:0 BatchNorm ReLU avg-pool:$[6\times6]$, stride:1, pad:0	Conv: $[1\times1\times512\times4096]$, stride:1, pad:0 BatchNorm ReLU avg-pool:$[13\times13]$	Conv: $[1\times1\times512\times4096]$, stride:1, pad:0 BatchNorm max-pool:$[6\times6]$, stride:1, pad:0 ReLU	Conv: $[1\times1\times512\times4096]$, stride:1, pad:0 BatchNorm max-pool:$[13\times13]$ ReLU
mlp1	Conv BN ReLU	$[1\times1\times4096\times4096]$ stride:1, pad:0	$[1\times1\times4096\times4096]$ stride:1, pad:0	$[1\times1\times4096\times4096]$ stride:1, pad:0	$[1\times1\times4096\times4096]$ stride:1, pad:0
mlp2	Conv	$[1\times1\times4096\times1000]$ stride:1, pad:0	$[1\times1\times4096\times1000]$ stride:1, pad:0	$[1\times1\times4096\times1000]$ stride:1, pad:0	$[1\times1\times4096\times1000]$ stride:1, pad:0
	SoftMax				

Table G. Basic architecture of ResNet-50 [4] used in Table 5(*a,c*) of Sec. 4.3. "$\{\mathcal{A}|\mathcal{B}\}$" means the parallel paths of the process $\mathcal{A}$ and $\mathcal{B}$ sharing the same input, and "Identical Mapping" indicates passing the input to the output as it is. As described in Sec. 4.3, we focus on 3×3 convolutions without operating on `conv1`.

ResNet-50 [4] with configuration (*a,c*)

input	$[224 \times 224 \times 3]$
`conv1`	Conv-BN-ReLU: $[7 \times 7 \times 3 \times 64]$, stride:2, pad:3
	max-pool: $[3 \times 3]$, stride:2, pad:1
`conv2-1`	$\left\{ \begin{array}{l\|l} \text{Conv-BN: } [1 \times 1 \times 64 \times 256], & \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 64 \times 64], & \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 64 \times 256] & \text{stride:1, pad:0} \end{array} \right\}$
	Sum + ReLU
`conv2-*`	$\left\{ \text{Identity Mapping} \left\| \begin{array}{l\|l} \text{Conv-BN-ReLU: } [1 \times 1 \times 256 \times 64], & \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 64 \times 64], & \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 64 \times 256] & \text{stride:1, pad:0} \end{array} \right. \right\} \times 2$
	Sum + ReLU
`conv3-1`	$\left\{ \begin{array}{l\|l} \text{Conv-BN: } [1 \times 1 \times 256 \times 512], & \text{stride:2, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 128 \times 128], & \text{stride:2, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 128 \times 512] & \text{stride:1, pad:0} \end{array} \right\}$
	Sum + ReLU
`conv3-*`	$\left\{ \text{Identity Mapping} \left\| \begin{array}{l\|l} \text{Conv-BN-ReLU: } [1 \times 1 \times 512 \times 128], & \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 128 \times 128], & \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 128 \times 512] & \text{stride:1, pad:0} \end{array} \right. \right\} \times 3$
	Sum + ReLU
`conv4-1`	$\left\{ \begin{array}{l\|l} \text{Conv-BN: } [1 \times 1 \times 512 \times 1024], & \text{stride:2, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 256 \times 256], & \text{stride:2, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 256 \times 1024] & \text{stride:1, pad:0} \end{array} \right\}$
	Sum + ReLU
`conv4-*`	$\left\{ \text{Identity Mapping} \left\| \begin{array}{l\|l} \text{Conv-BN-ReLU: } [1 \times 1 \times 1024 \times 256], & \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 256 \times 256], & \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 256 \times 1024] & \text{stride:1, pad:0} \end{array} \right. \right\} \times 5$
	Sum + ReLU
`conv5-1`	$\left\{ \begin{array}{l\|l} \text{Conv-BN: } [1 \times 1 \times 1024 \times 2048], & \text{stride:2, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 512 \times 512], & \text{stride:2, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 512 \times 2048] & \text{stride:1, pad:0} \end{array} \right\}$
	Sum + ReLU
`conv5-*`	$\left\{ \text{Identity Mapping} \left\| \begin{array}{l\|l} \text{Conv-BN-ReLU: } [1 \times 1 \times 2048 \times 512], & \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 512 \times 512], & \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 512 \times 2048] & \text{stride:1, pad:0} \end{array} \right. \right\} \times 2$
	Sum + ReLU
	avg-pool: $[7 \times 7]$, stride:1, pad:0
	Conv: $[1 \times 1 \times 2048 \times 1000]$, stride:1, pad:0
	SoftMax

Table H. Improved architecture of ResNet-50 used in Table 5(*b,d*) by introducing BoW-based representation (Sec. 3.3) according to the analysis in Sec. 4.3. The `bow` layer is applied to the *concatenated* features at `conv4-6` of which dimensionality is reduced like PCA in the BoW framework, while the layers of `conv5-*` in Table G are removed; this model is actually composed of 43 convolution layers.

ResNet-50 [4] with configuration (*b,d*)

input	$[224 \times 224 \times 3]$
`conv1`	Conv-BN-ReLU: $[7 \times 7 \times 3 \times 64]$, stride:2, pad:3
	max-pool: $[3 \times 3]$, stride:2, pad:1
`conv2-1`	$\left\{ \text{Conv-BN: } [1 \times 1 \times 64 \times 256],\ \text{stride:1, pad:0} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 64 \times 64],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 64 \times 64],\ \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 64 \times 256]\ \text{stride:1, pad:0} \end{array} \right\}$
	Sum + ReLU
`conv2-*`	$\left\{ \text{Identity Mapping} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 256 \times 64],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 64 \times 64],\ \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 64 \times 256]\ \text{stride:1, pad:0} \end{array} \right\} \times 2$
	Sum + ReLU
`conv3-1`	$\left\{ \text{Conv-BN: } [1 \times 1 \times 256 \times 512],\ \text{stride:2, pad:0} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 256 \times 128],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 128 \times 128],\ \text{stride:2, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 128 \times 512]\ \text{stride:1, pad:0} \end{array} \right\}$
	Sum + ReLU
`conv3-*`	$\left\{ \text{Identity Mapping} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 512 \times 128],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 128 \times 128],\ \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 128 \times 512]\ \text{stride:1, pad:0} \end{array} \right\} \times 3$
	Sum + ReLU
`conv4-1`	$\left\{ \text{Conv-BN: } [1 \times 1 \times 512 \times 1024],\ \text{stride:2, pad:0} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 512 \times 256],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 256 \times 256],\ \text{stride:2, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 256 \times 1024]\ \text{stride:1, pad:0} \end{array} \right\}$
	Sum + ReLU
`conv4-*`	$\left\{ \text{Identity Mapping} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 1024 \times 256],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 256 \times 256],\ \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 256 \times 1024]\ \text{stride:1, pad:0} \end{array} \right\} \times 4$
	Sum + ReLU
`conv4-6`	$\left\{ \text{Identity Mapping} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 1024 \times 256],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 256 \times 256],\ \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 256 \times 1024]\ \text{stride:1, pad:0} \end{array} \right\}$
	Concatenation + ReLU
Dimensionality reduction	Conv-BN-ReLU: $[1 \times 1 \times 2048 \times 1024]$, stride:1, pad:0
`bow`	Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 2048]$, stride:1, pad:0 avg-pool: $[7 \times 7]$, stride:1, pad:0
	Conv: $[1 \times 1 \times 2048 \times 1000]$, stride:1, pad:0
	SoftMax

Table I. Improved architecture of ResNet-50 used in Table 5(e) by enlarging the filter size in conv4 layers.

ResNet-50 [4] with configuration (e)	
input	$[224 \times 224 \times 3]$
conv1	Conv-BN-ReLU: $[7 \times 7 \times 3 \times 64]$, stride:2, pad:3
	max-pool: $[3 \times 3]$, stride:2, pad:1
conv2-1	$\left\{ \text{Conv-BN: } [1 \times 1 \times 64 \times 256],\ \text{stride:1, pad:0} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 64 \times 64],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 64 \times 64],\ \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 64 \times 256]\ \text{stride:1, pad:0} \end{array} \right\}$
	Sum + ReLU
conv2-$\star$	$\left\{ \text{Identity Mapping} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 256 \times 64],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 64 \times 64],\ \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 64 \times 256]\ \text{stride:1, pad:0} \end{array} \right\} \times 2$
	Sum + ReLU
conv3-1	$\left\{ \text{Conv-BN: } [1 \times 1 \times 256 \times 512],\ \text{stride:2, pad:0} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 256 \times 128],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 128 \times 128],\ \text{stride:2, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 128 \times 512]\ \text{stride:1, pad:0} \end{array} \right\}$
	Sum + ReLU
conv3-$\star$	$\left\{ \text{Identity Mapping} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 512 \times 128],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{3} \times \mathbf{3} \times 128 \times 128],\ \text{stride:1, pad:1} \\ \text{Conv-BN: } [1 \times 1 \times 128 \times 512]\ \text{stride:1, pad:0} \end{array} \right\} \times 3$
	Sum + ReLU
conv4-1	$\left\{ \text{Conv-BN: } [1 \times 1 \times 512 \times 1024],\ \text{stride:2, pad:0} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 512 \times 256],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{5} \times \mathbf{5} \times 256 \times 256],\ \text{stride:2, pad:2} \\ \text{Conv-BN: } [1 \times 1 \times 256 \times 1024]\ \text{stride:1, pad:0} \end{array} \right\}$
	Sum + ReLU
conv4-$\star$	$\left\{ \text{Identity Mapping} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 1024 \times 256],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{5} \times \mathbf{5} \times 256 \times 256],\ \text{stride:1, pad:2} \\ \text{Conv-BN: } [1 \times 1 \times 256 \times 1024]\ \text{stride:1, pad:0} \end{array} \right\} \times 4$
	Sum + ReLU
conv4-6	$\left\{ \text{Identity Mapping} \;\middle\vert\; \begin{array}{l} \text{Conv-BN-ReLU: } [1 \times 1 \times 1024 \times 256],\ \text{stride:1, pad:0} \\ \textbf{Conv}\text{-BN-ReLU: } [\mathbf{5} \times \mathbf{5} \times 256 \times 256],\ \text{stride:1, pad:2} \\ \text{Conv-BN: } [1 \times 1 \times 256 \times 1024]\ \text{stride:1, pad:0} \end{array} \right\}$
	Concatenation + ReLU
Dimensionality reduction	Conv-BN-ReLU: $[1 \times 1 \times 2048 \times 1024]$, stride:1, pad:0
bow	Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 2048]$, stride:1, pad:0 avg-pool: $[7 \times 7]$, stride:1, pad:0
	Conv: $[1 \times 1 \times 2048 \times 1000]$, stride:1, pad:0
	SoftMax

Table J. Improved architecture of ResNet-50 used in Table 5(*f*) by adding 5×5 convolution in the residual path at `conv4` layers. The number of channels which the 5×5 convolution receives is half of that in the corresponding 3×3 convolution.

ResNet-50 [4] with configuration (*f*)	
input	$[224 \times 224 \times 3]$
`conv1`	Conv-BN-ReLU: $[7 \times 7 \times 3 \times 64]$, stride:2, pad:3
	max-pool: $[3 \times 3]$, stride:2, pad:1
`conv2-1`	{ Conv-BN: $[1 \times 1 \times 64 \times 256]$, stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 64 \times 64]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 64 \times 64]$, stride:1, pad:1 / Conv-BN: $[1 \times 1 \times 64 \times 256]$ stride:1, pad:0 }
	Sum + ReLU
`conv2-*`	{ Identity Mapping | Conv-BN-ReLU: $[1 \times 1 \times 256 \times 64]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 64 \times 64]$, stride:1, pad:1 / Conv-BN: $[1 \times 1 \times 64 \times 256]$ stride:1, pad:0 } $\times 2$
	Sum + ReLU
`conv3-1`	{ Conv-BN: $[1 \times 1 \times 256 \times 512]$, stride:2, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 256 \times 128]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 128 \times 128]$, stride:2, pad:1 / Conv-BN: $[1 \times 1 \times 128 \times 512]$ stride:1, pad:0 }
	Sum + ReLU
`conv3-*`	{ Identity Mapping | Conv-BN-ReLU: $[1 \times 1 \times 512 \times 128]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 128 \times 128]$, stride:1, pad:1 / Conv-BN: $[1 \times 1 \times 128 \times 512]$ stride:1, pad:0 } $\times 3$
	Sum + ReLU
`conv4-1`	{ Conv-BN: $[1 \times 1 \times 512 \times 1024]$, stride:2, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 512 \times 256]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 256 \times 256]$, stride:2, pad:1 / Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 512 \times 128]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 128 \times 128]$, stride:2, pad:2 / Conv-BN: $[1 \times 1 \times 128 \times 1024]$ stride:1, pad:0 }
	Sum + ReLU
`conv4-*`	{ Identity Mapping | Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 256]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 256 \times 256]$, stride:1, pad:1 / Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 128]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 128 \times 128]$, stride:1, pad:2 / Conv-BN: $[1 \times 1 \times 128 \times 1024]$ stride:1, pad:0 } $\times 4$
	Sum + ReLU
`conv4-6`	{ Identity Mapping | Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 256]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 256 \times 256]$, stride:1, pad:1 / Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 128]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 128 \times 128]$, stride:1, pad:2 / Conv-BN: $[1 \times 1 \times 128 \times 1024]$ stride:1, pad:0 }
	Concatenation + ReLU
Dimensionality reduction	Conv-BN-ReLU: $[1 \times 1 \times 3072 \times 1024]$, stride:1, pad:0
`bow`	Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 2048]$, stride:1, pad:0 / avg-pool: $[7 \times 7]$, stride:1, pad:0
	Conv: $[1 \times 1 \times 2048 \times 1000]$, stride:1, pad:0
	SoftMax

Table K. Improved architecture of ResNet-50 used in Table 5(g) by adding 5×5 convolution at conv2∼conv4 layers. The number of channels which the 5×5 convolution receives is half of that in the corresponding 3×3 convolution.

ResNet-50 [4] with configuration (g)	
input	$[224 \times 224 \times 3]$
conv1	Conv-BN-ReLU: $[7 \times 7 \times 3 \times 64]$, stride:2, pad:3
	max-pool: $[3 \times 3]$, stride:2, pad:1
conv2-1	{ Conv-BN: $[1 \times 1 \times 64 \times 256]$, stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 64 \times 64]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 64 \times 64]$, stride:1, pad:1 / Conv-BN: $[1 \times 1 \times 64 \times 256]$ stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 64 \times 32]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 32 \times 32]$, stride:1, pad:2 / Conv-BN: $[1 \times 1 \times 32 \times 256]$ stride:1, pad:0 }
	Sum + ReLU
conv2-*	{ Identity Mapping | Conv-BN-ReLU: $[1 \times 1 \times 256 \times 64]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 64 \times 64]$, stride:1, pad:1 / Conv-BN: $[1 \times 1 \times 64 \times 256]$ stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 256 \times 32]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 32 \times 32]$, stride:1, pad:2 / Conv-BN: $[1 \times 1 \times 32 \times 256]$ stride:1, pad:0 } $\times 2$
	Sum + ReLU
conv3-1	{ Conv-BN: $[1 \times 1 \times 256 \times 512]$, stride:2, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 256 \times 128]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 128 \times 128]$, stride:2, pad:1 / Conv-BN: $[1 \times 1 \times 128 \times 512]$ stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 256 \times 64]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 64 \times 64]$, stride:2, pad:2 / Conv-BN: $[1 \times 1 \times 64 \times 512]$ stride:1, pad:0 }
	Sum + ReLU
conv3-*	{ Identity Mapping | Conv-BN-ReLU: $[1 \times 1 \times 512 \times 128]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 128 \times 128]$, stride:1, pad:1 / Conv-BN: $[1 \times 1 \times 128 \times 512]$ stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 512 \times 64]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 64 \times 64]$, stride:1, pad:2 / Conv-BN: $[1 \times 1 \times 64 \times 512]$ stride:1, pad:0 } $\times 3$
	Sum + ReLU
conv4-1	{ Conv-BN: $[1 \times 1 \times 512 \times 1024]$, stride:2, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 512 \times 256]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 256 \times 256]$, stride:2, pad:1 / Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 512 \times 128]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 128 \times 128]$, stride:2, pad:2 / Conv-BN: $[1 \times 1 \times 128 \times 1024]$ stride:1, pad:0 }
	Sum + ReLU
conv4-*	{ Identity Mapping | Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 256]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 256 \times 256]$, stride:1, pad:1 / Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 128]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 128 \times 128]$, stride:1, pad:2 / Conv-BN: $[1 \times 1 \times 128 \times 1024]$ stride:1, pad:0 } $\times 4$
	Sum + ReLU
conv4-6	{ Identity Mapping | Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 256]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 256 \times 256]$, stride:1, pad:1 / Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0 | Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 128]$, stride:1, pad:0 / **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 128 \times 128]$, stride:1, pad:2 / Conv-BN: $[1 \times 1 \times 128 \times 1024]$ stride:1, pad:0 }
	Concatenation + ReLU
Dimensionality reduction	Conv-BN-ReLU: $[1 \times 1 \times 3072 \times 1024]$, stride:1, pad:0
bow	Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 2048]$, stride:1, pad:0 / avg-pool: $[7 \times 7]$, stride:1, pad:0
	Conv: $[1 \times 1 \times 2048 \times 1000]$, stride:1, pad:0
	SoftMax

Table L. Improved architecture of ResNet-50 used in Table 5(*h*) by adding 5×5 convolution at conv2∼conv4 layers. The number of channels which the 5×5 convolution receives is the *same* as that in the corresponding 3×3 convolution.

ResNet-50 [4] with configuration (*h*)			
input	$[224 \times 224 \times 3]$		
conv1	Conv-BN-ReLU: $[7 \times 7 \times 3 \times 64]$, stride:2, pad:3		
	max-pool: $[3 \times 3]$, stride:2, pad:1		
conv2-1	Conv-BN: $[1 \times 1 \times 64 \times 256]$, stride:1, pad:0	Conv-BN-ReLU: $[1 \times 1 \times 64 \times 64]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 64 \times 64]$, stride:1, pad:1 Conv-BN: $[1 \times 1 \times 64 \times 256]$ stride:1, pad:0	Conv-BN-ReLU: $[1 \times 1 \times 64 \times 64]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 64 \times 64]$, stride:1, pad:2 Conv-BN: $[1 \times 1 \times 64 \times 256]$ stride:1, pad:0
	Sum + ReLU		
conv2-*	Identity Mapping	Conv-BN-ReLU: $[1 \times 1 \times 256 \times 64]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 64 \times 64]$, stride:1, pad:1 Conv-BN: $[1 \times 1 \times 64 \times 256]$ stride:1, pad:0	Conv-BN-ReLU: $[1 \times 1 \times 256 \times 64]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 64 \times 64]$, stride:1, pad:2 Conv-BN: $[1 \times 1 \times 64 \times 256]$ stride:1, pad:0
	Sum + ReLU		$\times 2$
conv3-1	Conv-BN: $[1 \times 1 \times 256 \times 512]$, stride:2, pad:0	Conv-BN-ReLU: $[1 \times 1 \times 256 \times 128]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 128 \times 128]$, stride:2, pad:1 Conv-BN: $[1 \times 1 \times 128 \times 512]$ stride:1, pad:0	Conv-BN-ReLU: $[1 \times 1 \times 256 \times 128]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 128 \times 128]$, stride:2, pad:2 Conv-BN: $[1 \times 1 \times 128 \times 512]$ stride:1, pad:0
	Sum + ReLU		
conv3-*	Identity Mapping	Conv-BN-ReLU: $[1 \times 1 \times 512 \times 128]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 128 \times 128]$, stride:1, pad:1 Conv-BN: $[1 \times 1 \times 128 \times 512]$ stride:1, pad:0	Conv-BN-ReLU: $[1 \times 1 \times 512 \times 128]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 128 \times 128]$, stride:1, pad:2 Conv-BN: $[1 \times 1 \times 128 \times 512]$ stride:1, pad:0
	Sum + ReLU		$\times 3$
conv4-1	Conv-BN: $[1 \times 1 \times 512 \times 1024]$, stride:2, pad:0	Conv-BN-ReLU: $[1 \times 1 \times 512 \times 256]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 256 \times 256]$, stride:2, pad:1 Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0	Conv-BN-ReLU: $[1 \times 1 \times 512 \times 256]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 256 \times 256]$, stride:2, pad:2 Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0
	Sum + ReLU		
conv4-*	Identity Mapping	Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 256]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 256 \times 256]$, stride:1, pad:1 Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0	Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 256]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 256 \times 256]$, stride:1, pad:2 Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0
	Sum + ReLU		$\times 4$
conv4-6	Identity Mapping	Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 256]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{3} \times \mathbf{3} \times 256 \times 256]$, stride:1, pad:1 Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0	Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 256]$, stride:1, pad:0 **Conv**-BN-ReLU: $[\mathbf{5} \times \mathbf{5} \times 256 \times 256]$, stride:1, pad:2 Conv-BN: $[1 \times 1 \times 256 \times 1024]$ stride:1, pad:0
	Concatenation + ReLU		
Dimensionality reduction	Conv-BN-ReLU: $[1 \times 1 \times 3072 \times 1024]$, stride:1, pad:0		
bow	Conv-BN-ReLU: $[1 \times 1 \times 1024 \times 2048]$, stride:1, pad:0 avg-pool: $[7 \times 7]$, stride:1, pad:0		
	Conv: $[1 \times 1 \times 2048 \times 1000]$, stride:1, pad:0		
	SoftMax		

References

[1] MatConvNet pre-trained models. http://www.vlfeat.org/matconvnet/pretrained/. Accessed: 2017-1-19.

[2] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *BMVC*, 2014.

[3] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Journal of Machine Learning Research*, 37:448–456, 2015.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv,* 1408.5093, 2014.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[8] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[11] 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*, pages 1–9, 2015.

[12] A. Vedaldi and K. Lenc. MatConvNet – convolutional neural networks for matlab. In *ACM MM*, 2015.