Supplementary Material for "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks"

Qilong Wang[1], Banggu Wu[1], Pengfei Zhu[1], Peihua Li[2], Wangmeng Zuo[3], Qinghua Hu[1,*]
[1] Tianjin Key Lab of Machine Learning, College of Intelligence and Computing, Tianjin University, China
[2] Dalian University of Technology, China [3] Harbin Institute of Technology, China

Method	CNNs	#.Param.	GFLOPs	Top-1	Top-5
ResNet [1]	R-18	**11.148M**	**1.699**	70.40	89.45
SENet [2]		11.231M	1.700	70.59	89.78
CBAM [3]		11.234M	1.700	70.73	89.91
ECA-Net (Ours)		**11.148M**	1.700	**70.78**	**89.92**
ResNet [1]	R-34	**20.788M**	**3.427**	73.31	91.40
SENet [2]		20.938M	3.428	73.87	91.65
CBAM [3]		20.943M	3.428	74.01	91.76
ECA-Net (Ours)		**20.788M**	3.428	**74.21**	**91.83**

Table A. Comparison of different methods using ResNet-18 (R-18) and ResNet-34 (R-34) on ImageNet in terms of network parameters (#.Param.), floating point operations per second (FLOPs), and Top-1/Top-5 accuracy (in %).

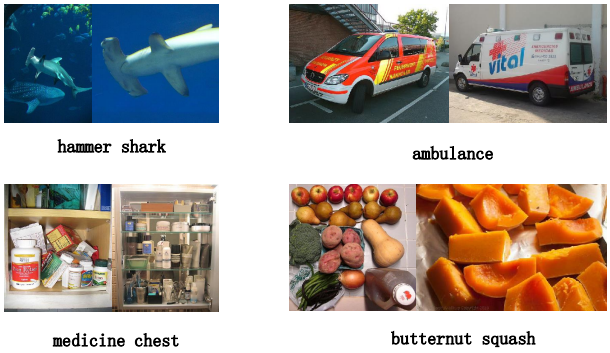


Figure A. Example images of four random sampled classes on ImageNet, including *hammerhead shark*, *ambulance*, *medicine chest* and *butternut squash*.

A. Comparison of Different Methods using ResNet-18 and ResNet-34 on ImageNet

Here, we compare different attention methods using ResNet-18 and ResNet-34 on ImageNet. The results are listed in Table A, where the results of ResNet, SENet and CBAM are duplicated from [3], and we train ECA-Net using the settings of hyper-parameters with [3]. From Table A, we can see that our ECA-Net improves the original ResNet-18 and ResNet-34 over 0.38% abd 0.9% in Top-1 accuracy, respectively. Comparing with SENet and CBAM, our ECA-Net achieves better performance using less model complexity, showing the effectiveness of the proposed ECA module.

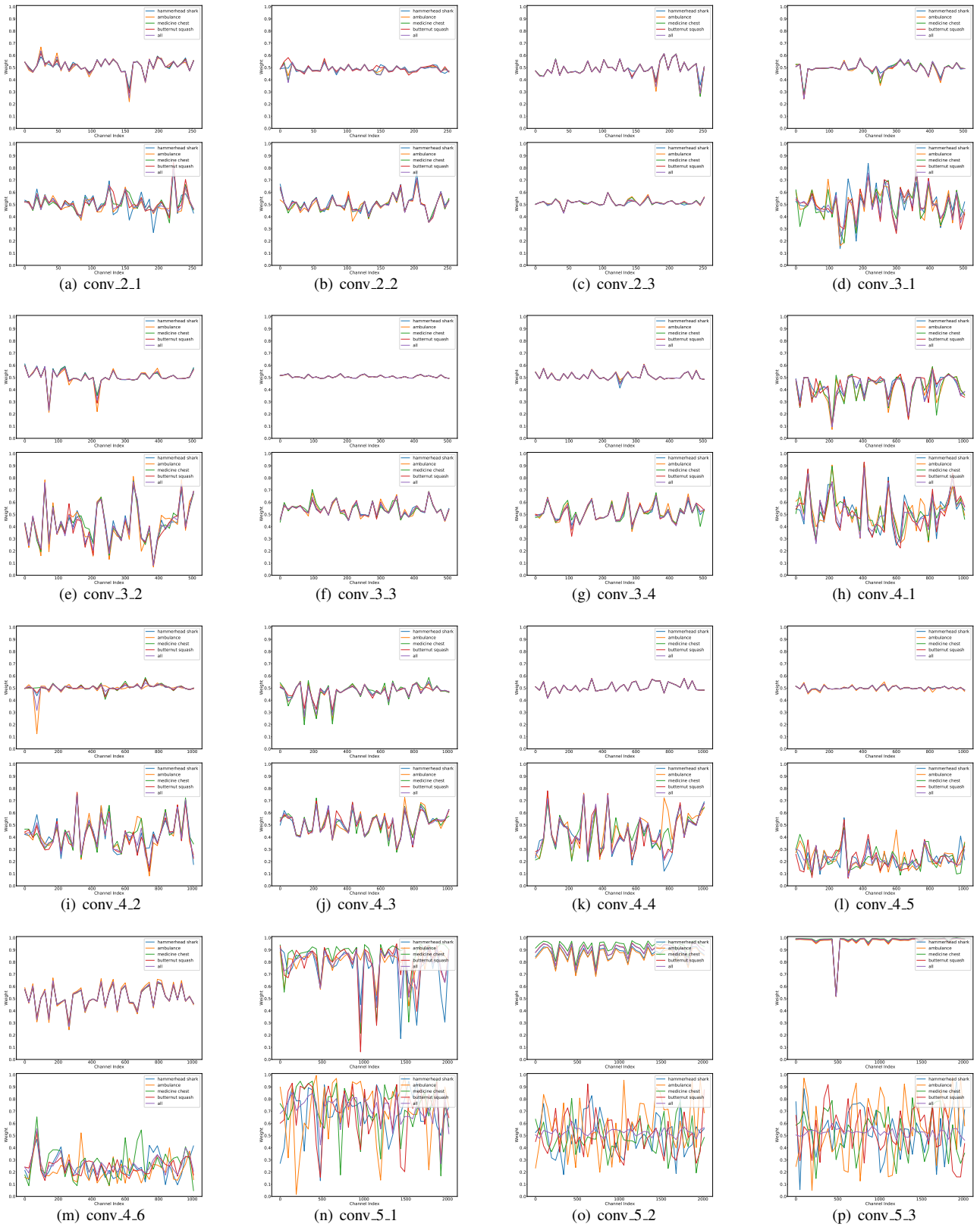
B. Stacking More 1D Convolutions in ECA Module

Intuitively, more 1D convolutions stacked in ECA module may bring further improvement, due to increase of modeling capability. Actually, we found that one extra 1D convolution brings trivial gains ($\sim$0.1%) at the cost of slightly increasing complexity, but more 1D convolutions degrade performance, which may be caused by that more 1D convolutions make gradient backpropagation more difficult. Therefore, our final ECA module contains only one 1D convolution.

C. Visualization of Weights Learned by ECA Modules and SE Blocks

To further analyze the effect of our ECA module on learning channel attention, we visualize the weights learned by ECA modules and compare with SE blocks. Here, we employ ResNet-50 as backbone model, and illustrate weights of different convolution blocks. Specifically, we randomly sample four classes from ImageNet dataset, which are *hammerhead shark*, *ambulance*, *medicine chest* and *butternut squash*, respectively. Some example images are illustrated in Figure A. After training the networks, for all images of each class collected from validation set of ImageNet, we compute the channel weights of convolution blocks on average. Figure B visualizes the channel weights of conv_i_j, which indicates j-th convolution block in i-th stage. Besides the visualization results of four random sampled classes, we also give the distribution of the aver-

*Qinghua Hu is the corresponding author.
Email: {qlwang, wubanggu, huqinghua}@tju.edu.cn. The work was supported by the National Natural Science Foundation of China (Grant No. 61806140, 61876127, 61925602, 61971086, U19A2073, 61732011), Major Scientific Research Project of Zhejiang Lab (2019DB0ZX01). Q. Wang was supported by National Postdoctoral Program for Innovative Talents.



(a) conv_2_1 (b) conv_2_2 (c) conv_2_3 (d) conv_3_1

(e) conv_3_2 (f) conv_3_3 (g) conv_3_4 (h) conv_4_1

(i) conv_4_2 (j) conv_4_3 (k) conv_4_4 (l) conv_4_5

(m) conv_4_6 (n) conv_5_1 (o) conv_5_2 (p) conv_5_3

Figure B. Visualization the channel weights of conv_i_j, where i indicate i-th stage and j is j-th convolution block in i-th stage. The channel weights learned by ECA modules and SE blocks are illustrated in bottom and top of each row, respectively. Better view with zooming in.

age weights across $1K$ classes as reference. The channel weights learned by ECA modules and SE blocks are illustrated in bottom and top of each row, respectively.

From Figure B we have the following observations. Firstly, for both ECA modules and SE blocks, the distributions of channel weights for different classes are very similar at the earlier layers (i.e., ones from conv_2_1 to conv_3_4), which may be by reason of that the earlier layers aim at capturing the basic elements (e.g., boundaries and corners) [4]. These features are almost similar for different classes. Such phenomenon also was described in the extended version of [2][1]. Secondly, for the channel weights of different classes learned by SE blocks, most of them tend to be the same (i.e., 0.5) in conv_4_2 $\sim$ conv_4_5 while the differences among various classes are not obvious. On the contrary, the weights learned by ECA modules are clearly different across various channels and classes. Since convolution blocks in 4-th stage prefer to learn semantic information, so the weights learned by ECA modules can better distinguish different classes. Finally, convolution blocks in the final stage (i.e., conv_5_1, conv_5_2 and conv_5_3) capture high-level semantic features and they are more class-specific. Obviously, the weights learned by ECA modules are more class-specific than ones learned by SE blocks. Above results clearly demonstrate that the weights learned by our ECA modules have better discriminative ability.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[2] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[3] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *ECCV*, 2018.

[4] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014.

[1] https://arxiv.org/abs/1709.01507