Supplementary Material:Interpretable Neural Networks Decoupling

Yuchao Li[1], Rongrong Ji[1,2⋆] , Shaohui Lin[3], Baochang Zhang[4], Chenqian Yan[1], Yongjian Wu[5], Feiyue Huang[5], Ling Shao[6,7]

[1]Department of Artificial Intelligence, School of Informatics, Xiamen University, China, [2]Peng Cheng Laboratory, Shenzhen, China, [3]National University of Singapore, Singapore, [4]Beihang University, China, [5]BestImage, Tencent Technology (Shanghai) Co.,Ltd, China, [6]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE, [7]Inception Institute of Artificial Intelligence, Abu Dhabi, UAE

A Network Interpretation

A.1 Filter State

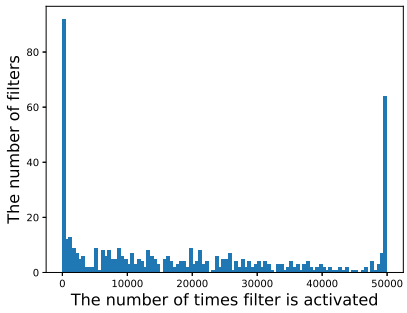


Fig. A. The distribution of the number of times filters activated on ImageNet 2012 validation set. These filters are collected from the last convolutional layer of VGG-16.

As shown in Fig A, we collect filters in the last convolutional layer of VGG16 on ImageNet 2012 after network decoupling and present the results of the number of times they are activated. ImageNet 2012 contains 50,000 validation images, so the number of times each filter is activated falls in the interval $[0, 50000]$. The leftmost bar represents the number of filters that never been activated (*i.e.,* silent filters), and the rightmost bar represents the number of filters that are activated every time (*i.e.,* energetic filters). The middle bars represent the number of filters, which respond to specific inputs (*i.e.,* dynamic filters). For instance, the rightmost bar represents that there are 60 filters, which are activated 50,000 times during evaluating on ImageNet 2012, in the last convolutional layer of

⋆ Corresponding author.

VGG-16 after network decoupling. They represent three different roles played by filters in the network: Silent filters represent the redundant information, dynamic filters are responsible for specific semantic concepts. A special case is the energetic filters, the existence of which attributes to the fact that most networks are limited in width (*i.e.,* the number of filters). the networks need some energetic filters which encode the more semantic concepts rather than a specific one. After that, energetic filters are participated in the calculation path of all input images to improve the network performance.

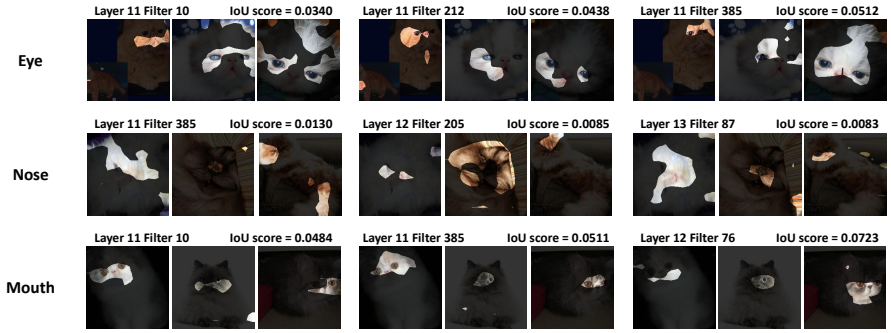


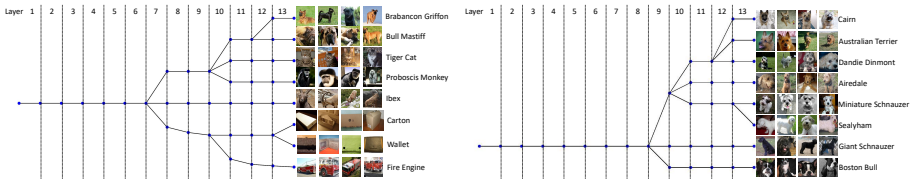
Fig. B. Visualization of the receptive fields of filters which are inactivated because of the lack of semantic feature (*i.e.,* eye, nose and mouth) in images (*i.e.,* cat).

A.2 Semantic Concepts

To explore the relationship between the characterization of different semantic concepts in the network, we also visualize the different semantic concepts for the same category of images, as shown in Fig. B. The different parts of the cats (*i.e.,* eye, nose and mouth) are occluded by black blocks, and then the filters becomes inactivated due to the lack of semantic features have been collected. As shown in the Fig. B, the 385-th filter in the 11-th convolutional layer of VGG16 is always inactivated due to the lack of some features of cat. This demonstrates that the semantic concepts detected by this filter are covering the entire cat face, including eye, nose and mouth. Other filters are only responsible for a single semantic concept. For example, the 76-th and 205-th filters in the 12-th convolutional layer only detects the mouth and nose of the cats, respectively.

A.3 Decision-Making Process of a Network.

To investigate the decision-making process of a network and the functional process of its intermediate layers, we collect the calculation paths of a decoupled VGG-16 from eight different categories of images, which contain different kinds of



(a) Different kinds of objects. (b) Fine-grained dogs.

Fig. C. Visualization of the decision-making process of VGG-16 based on eight different categories of images.

artifacts and animals in Fig. F, and fine-grained dogs in Fig. 3(b). We first collect the architecture encoding vectors layer-by-layer and then compute their Hopkins Statistic [4] to analyze whether the inputs have different calculation paths in this layer. If yes, we divide inputs into two subclasses by k-means. In contrast, we keep the inputs in the same class and turn to the next layer. The results show that the bottom layers in the network are responsible for general features, thus all inputs share the same calculation path. As shown in Fig. F, the decoupled VGG-16 cannot distinguish the difference between artifacts and animals until reaching the 7-th convolutional layer. Moreover, the network distinguishes the difference in the fine-grained dogs after reaching the 9-th convolutional layer, as shown in Fig. 3(b). As the layers become deeper, the network gradually distinguishes the different objects, and similar objects are distinguished in the last layers.

Network	λ_m	λ_k	λ_s	R
ResNet-56	0.01	1	0.00015	0
VGGNet	0.04	1	0.0002	0
GoogleNet	0.006	1	0.00005	0
ResNet-18	0.005	1	0.01	0.6
VGG-16	0.01	1	0.01	0.5/0.8

Table A. Hyper-parameter settings on network acceleration.

A.4 Visualization of Network Architecture

As shown in Fig F, we visualize the calculation paths of different categories for ResNet-20 on CIFAR-10. Each path for the specific class (*e.g.*, airplane) is obtained by the statistic of all images labelled by this class. Our method can successfully decouple the network architecture, and effectively distinguish the "hard" and " easy" classes. For example, The images belonging to "automobile" or "truck" require much more filters (*i.e.*, more complex path), compared to a thinner path based on the "bird" category.

Model	Top-1 Acc(%)	Top-5 Acc(%)	FLOPs Reduction	CPU Time Reduction
Perforated CNNs [1]	-	88.8	-	2.00×
RunTime Neural Pruning [5]	-	87.58	3.00×	-
ThiNet [7]	69.80	89.53	3.23×	-
Global and Dynamic Filter Pruning [6]	68.80	88.77	2.42×	1.62×
Feature Boosting and Suppression [2]	-	89.86	3.00×	**2.97**×
Decoupling	**71.51**	**90.32**	**3.23**×	2.44×

Table B. Results of VGG-16 on ImageNet2012. The baseline in our method has 15.48B FLOPs and an average 1220 ms testing on CPU based an image by running the whole of the validation dataset.

B Network Acceleration

Method	Top1-Acc(%)	FLOPs	dynamic filters	slient filters	energetic filters
ResNet-56+ACM	93.17	118M	0.30%	13.25%	86.45%
ResNet-56+ACM+$\mathcal{L}_s$	92.94	63M	0.24%	59.60%	40.16%
ResNet-56+ACM+$\mathcal{L}_s$+$\mathcal{L}_{kl}$	92.81	117M	0.00%	15.22%	84.78%
ResNet-56+ACM+$\mathcal{L}_s$+$\mathcal{L}_{mi}$	92.99	67M	29.00%	32.55%	38.45%
ResNet-56+ACM+$\mathcal{L}_s$+$\mathcal{L}_{mi}$+$\mathcal{L}_{kl}$	93.08	63M	30.27%	35.24%	34.49%

Table C. Effect of the loss. ACM represents the architecture controlling module.

For VGG-16 on ImageNet 2012, to achieve the best trade-off between accuracy and speed, we follow the [7] to set $R = 0.5$ in the first ten layers and $R = 0.8$ in the last three layers. As shown in Table B, we obtain 90.32% Top-5 accuracy with a 2.44× real CPU running speedup and 3.23× reduction in FLOPs, which is better than static pruning [1, 7, 6] and dynamic pruning [5, 2].

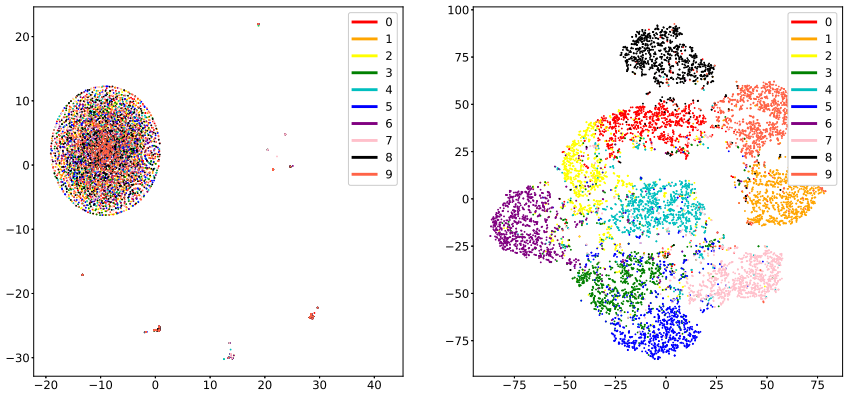
The detailed of hyper-parameter settings in our experiments are shown in Table A. The λ_m, λ_k and λ_s control the influence of corresponding losses. And the R represents the target compression ratio.

C Adversarial Sample Detection

C.1 Adversarial Attack Analysis

We use the FGSM [3] to attack ResNet-20 on CIFAR-10. As shown in Fig. G, the images labelled by "airplane" are attacked to be other classes. Compared to the original calculation path in Fig. F, the adversarial samples always confuse the network starting from the bottom layers. Furthermore, we compare the

calculation path of different categories that are attacked to be the same one in Fig. H,. For the images attacked to be the same category, they almost share the same calculation path. Thus, the adversarial samples can be effectively detected using our architecture decoupling method by comparing f calculation path between the un-attacked/original images (*e.g.*, dog in Fig. F) and the adversarial samples.



(a) ResNet-56 without our losses. (b) ResNet-56 with our losses.

Fig. D. Visualization of the distribution of the whole of calculation path in ResNet-56 on CIFAR-10 with or without losses proposed by our method.

D Ablation Study

D.1 Effect of the Loss

We train the ResNet-56 on CIFAR-10 with or without the losses proposed in our method to analyze the effect of the each loss. As shown in Table C, the combination of three losses achieves the best trade-off between accuracy and FLOPs. The lack of $\mathcal{L}_{mi}$ results in that the network tends to use static pruning to compress itself. Meanwhile, compared with only using $\mathcal{L}_s$, using $\mathcal{L}_{kl}$ makes the filters respond to the all objects, which leads to the higher probability to generate the energetic filters. Furthermore, as shown in Fig. D, after the training with the combination of these losses, we can decouple ResNet-56 successfully.

Our decoupling method has three hyper-parameters (*i.e.*, λ_m, λ_k and λ_s) to control the network decoupling. As shown in Fig. E, we calculate the percentage of different filter states (*i.e.,* energetic, silent and dynamic) in the network with different λ_m, λ_k and λ_s on ResNet-56. We set $\lambda_m = 0.01$, $\lambda_k = 1$ and $\lambda_s = 0.0001$ when they are not being measured. We find that λ_m controls the number of dynamic filters, which means the network architecture can be decoupled as λ_m

increases. Meanwhile, λ_s controls the number of filters that participate in the network inference, and λ_k controls the number of energetic filters.

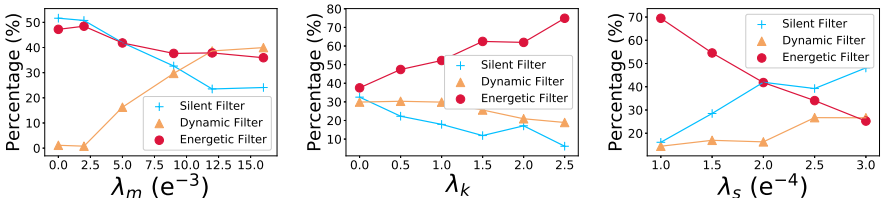


Fig. E. Percentages of different filter states for different λ_m, λ_k and λ_s on ResNet-56.

References

1. Figurnov, M., Ibraimova, A., Vetrov, D.P., Kohli, P.: Perforatedcnns: Acceleration through elimination of redundant convolutions. NeurIPS (2016)
2. Gao, X., Zhao, Y., Dudziak, L., Mullins, R., Xu, C.z.: Dynamic channel pruning: Feature boosting and suppression. ICLR (2018)
3. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. ICLR (2015)
4. Hopkins, B., Skellam, J.G.: A new method for determining the type of distribution of plant individuals. Annals of Botany (1954)
5. Lin, J., Rao, Y., Lu, J., Zhou, J.: Runtime neural pruning. NeurIPS (2017)
6. Lin, S., Ji, R., Li, Y., Wu, Y., Huang, F., Zhang, B.: Accelerating convolutional networks via global & dynamic filter pruning. IJCAI (2018)
7. Luo, J.H., Wu, J., Lin, W.: Thinet: A filter level pruning method for deep neural network compression. ICCV (2017)

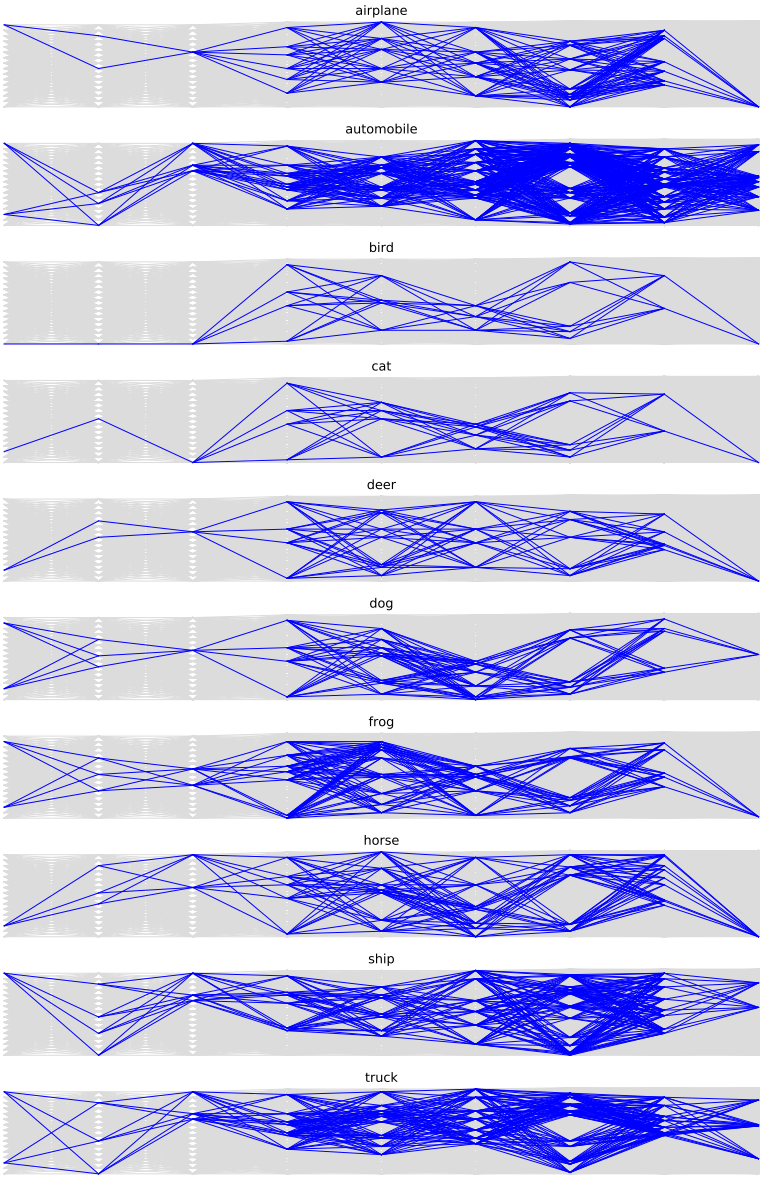


Fig. F. The calculation path of different categories on ResNet-20.

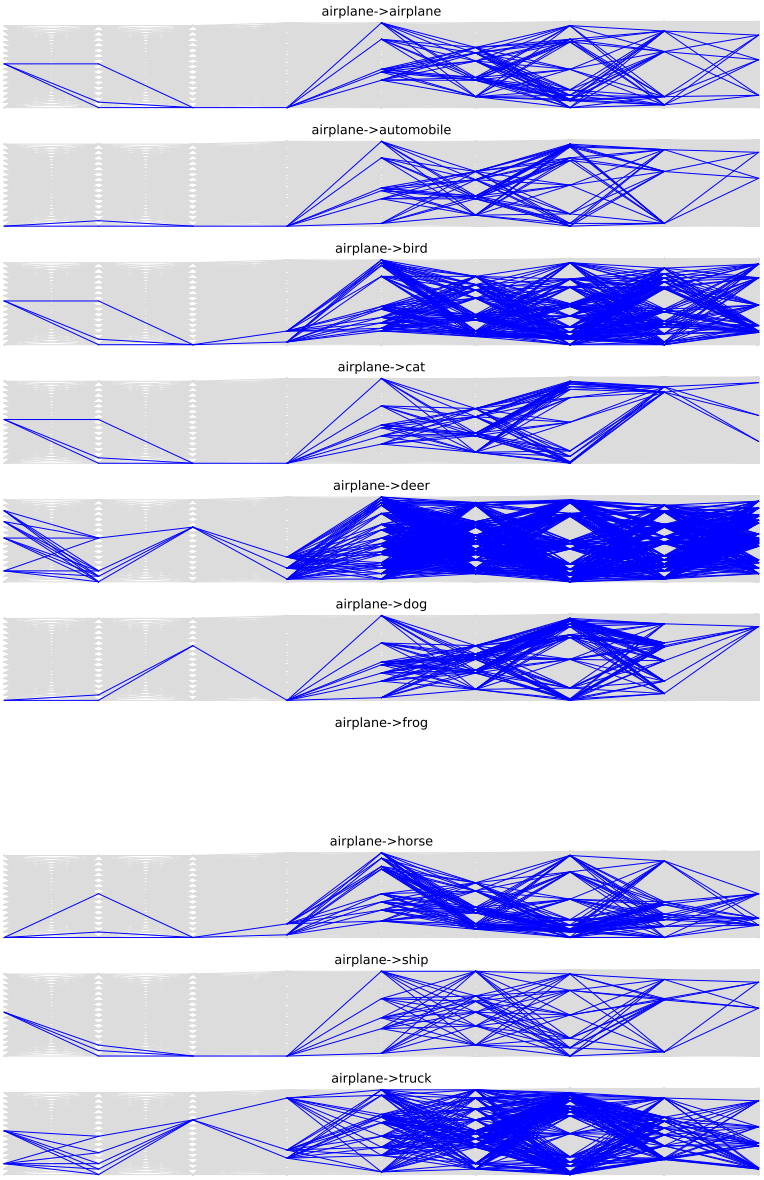


Fig. G. The calculation paths of same category images that are classified as other categories after being attacked on ResNet-20. The blank line represents we do not have this adversarial sample.

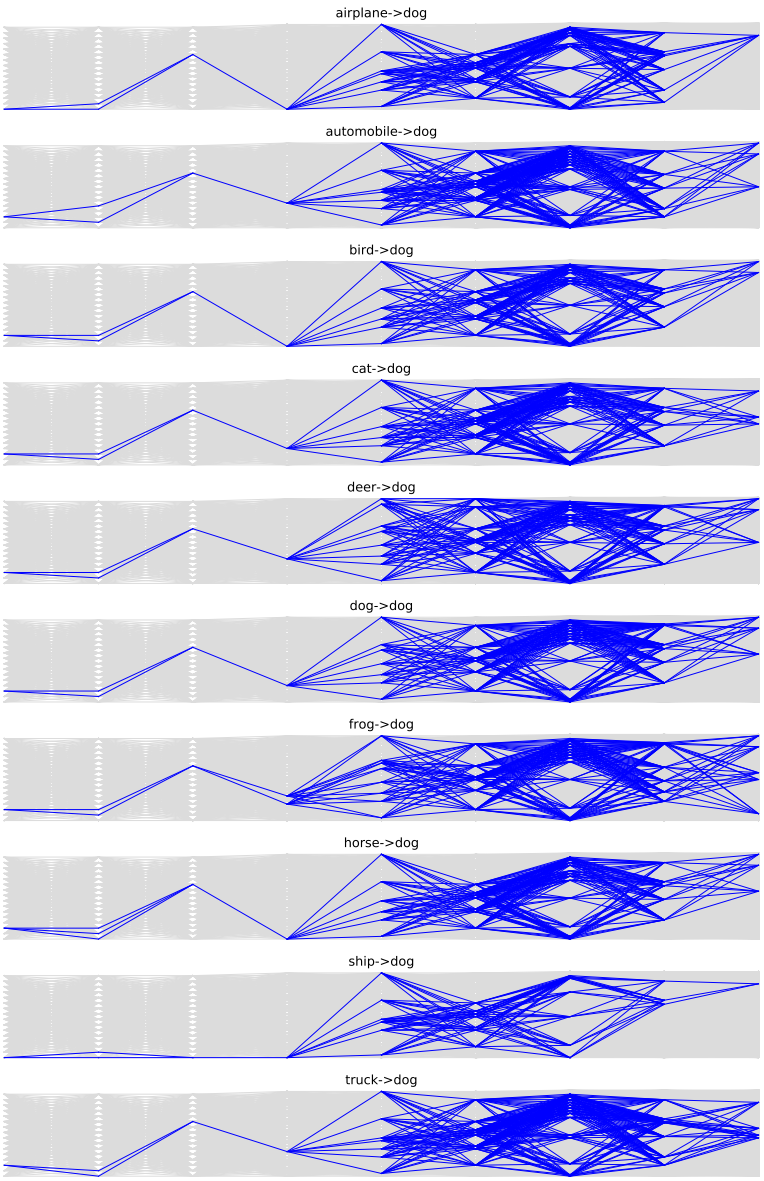


Fig. H. The calculation paths of different categories images that are classified as same category after being attacked on ResNet-20.