Co-Evolutionary Compression for Unpaired Image Translation
Supplementary Material

Abstract

In the main body of the proposed method, we have developed a novel approach for compressing generative networks using a co-evolutionary algorithm. In this document, we will further provide more visualization results, statistics of compressed models, and the quantitative evaluation to illustrate its superiority.

Architectures of Compressed Generators. As mentioned above, we utilize a co-evolutionary approach two simultaneously excavate redundancy in two generators for transferring images from different domains. Thus, we further observe architectures of compressed generators after applying the proposed method to have an explicit understanding. Table 1 and Table 2 report the detailed information of compressed generators on cityscapes-A2B and cityscapes-B2A tasks, respectively, as discussed in the main body.

It is obvious that, the compressed model in Table 2 has more parameters and calculations than those of the model in Table 1, which demonstrates that the model for conducting a harder task has less redundancy in the unpaired image translation.

Comparison with Conventional Pruning Method. Besides the conventional ThiNet [4], here we also compare the proposed method with other methods to illustrate its superiority on the unpaired image translation task, including network Trimming [1] and Slimming [3]. Figure 1 illustrates images generated by exploiting conventional pruning methods and the proposed method, where models generated by these methods are tuned to have similar models sizes as those illustrated in the main body. Not surprisingly, similar to images generated using ThiNet, results of Trimming and Slimming cannot preserve rich texture and color information in the desired domain. Since these method are designed for compressing CNNs to minimize reconstruction errors of recognition results and cannot simultaneously excavate redundancy in two generators, which are no longer suitable for conducting the unpaired image translation experiments.

Visualization Results. Moreover, we have provided some of example images generated using generators compressed by exploiting the proposed method on different datasets. Here we illustrate more visualization results to better illustrate the superiority of our approach, as shown in Figure 2. Compared with images generated using original generators, images produced by exploiting compressed generators using the proposed co-evolutionary method can maintain clear texture and color information with high visual quality. Thus, portable generators provided by the proposed method can be used for replacing original heavy generators for applying on mobile devices.

Quantitative Evaluation. In order to evaluate the performance of compressed generators, we have conducted the quantitative evaluation of the proposed method on the cityscapes dataset in the main body using "FCN-scores" [2]. Moreover, the segmentation results are shown in Figure 3. Images generated using the network compressed by our method obtained similar results to those of using the original model.

References

[1] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016. 1, 2

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 1

[3] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *ICCV*, pages 2755–2763, 2017. 1, 2

[4] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *ICCV*, pages 5058–5066, 2017. 1, 2

[5] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 2

Table 1. Compression statistics for the generator from street view to segmentation map.

	Layer/Stride	Original Filter Shape	Compressed Filter Shape	Memory(MB)	FLOPs(M)	r^c / r^s
	Conv1/s1	64×3×7×7	38×3×7×7	0.021	368	1.68×
	Conv2/s2	128×64×3×3	38×72×3×3	0.094	404	2.99×
	Conv3/s2	256×128×3×3	115×72×3×3	0.284	306	3.96×
×9	ResBlock-Conv1/s1	256×256×3×3	115×115×3×3	0.454	488	4.96×
	ResBlock-Conv2/s1	256×256×3×3	115×115×3×3	0.454	488	
	ConvTrans1/s2	256×128×3×3	69×115×3×3	0.272	1,171	4.13×
	ConvTrans2/s2	64×128×3×3	38×69×3×3	0.090	1,549	3.12×
	Conv4/s1	3×64×7×7	3×38×7×7	0.021	366	1.68×

Table 2. Compression statistics for the generator from segmentation map to street view.

	Layer/Stride	Original Filter Shape	Compressed Filter Shape	Memory(MB)	FLOPs(M)	r^c / r^s
	Conv1/s1	64×3×7×7	32×3×7×7	0.018	310	2.00×
	Conv2/s2	128×64×3×3	32×63×3×3	0.069	298	4.06×
	Conv3/s2	256×128×3×3	136×63×3×3	0.294	316	3.82×
×9	ResBlock-Conv1/s1	256×256×3×3	136×136×3×3	0.635	682	3.54×
	ResBlock-Conv2/s1	256×256×3×3	136×136×3×3	0.635	682	
	ConvTrans1/s2	256×128×3×3	68×136×3×3	0.318	1,365	3.54×
	ConvTrans2/s2	64×128×3×3	37×68×3×3	0.086	1,486	3.25×
	Conv4/s1	3×64×7×7	3×38×7×7	0.021	357	1.73×

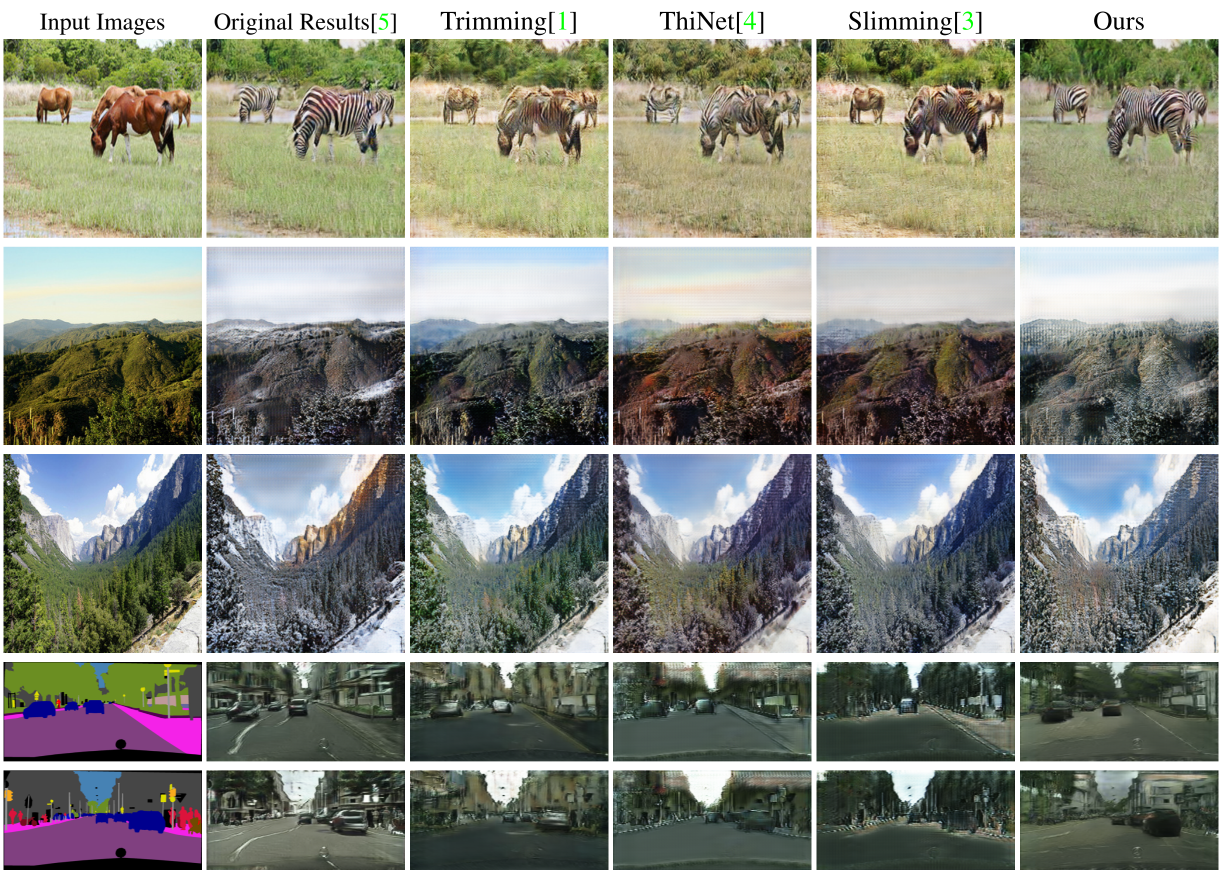

Figure 1. Images produced by generators compressed by exploiting different approaches.

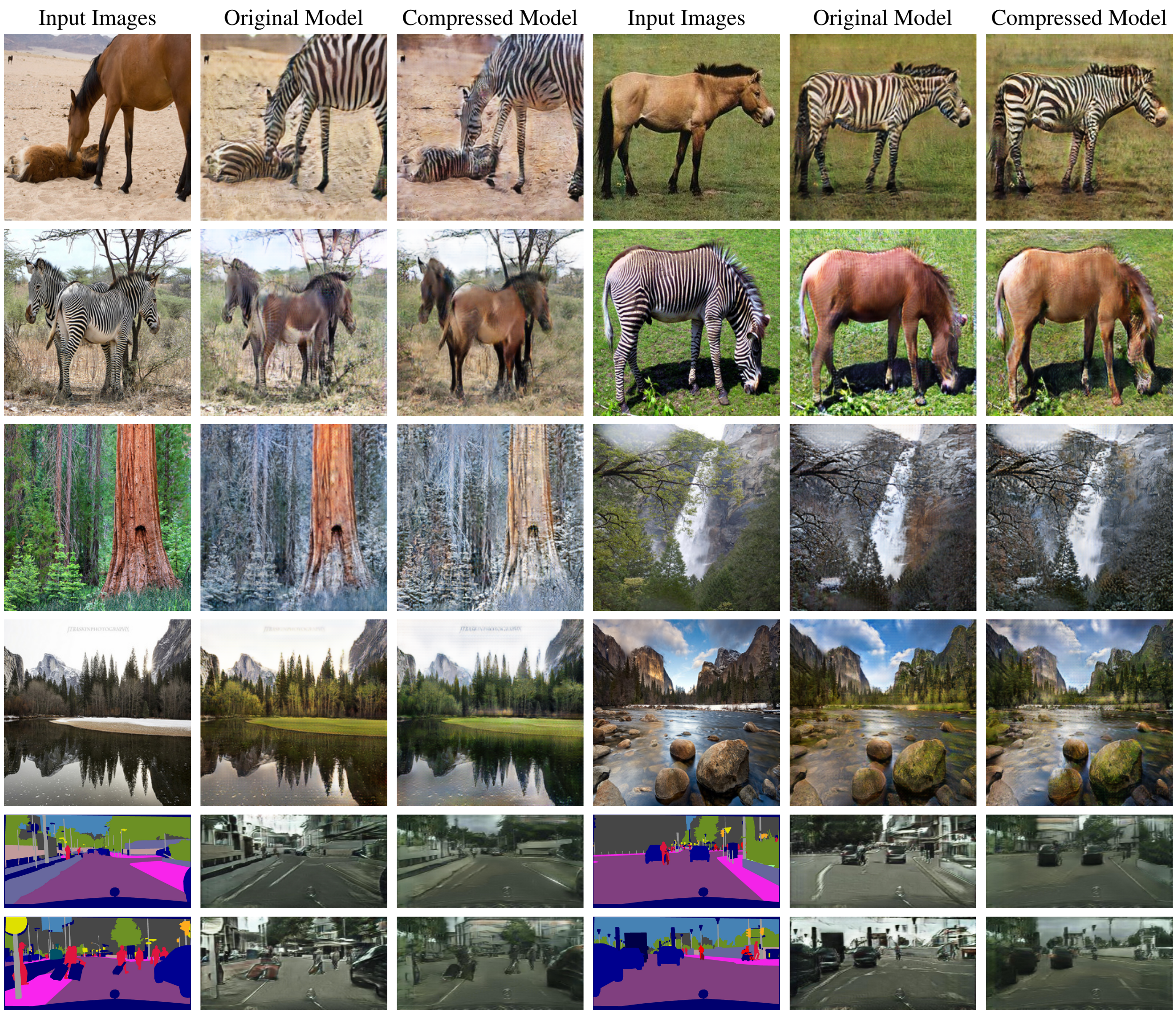

Figure 2. Images produced by original and compressed generators using the proposed method.

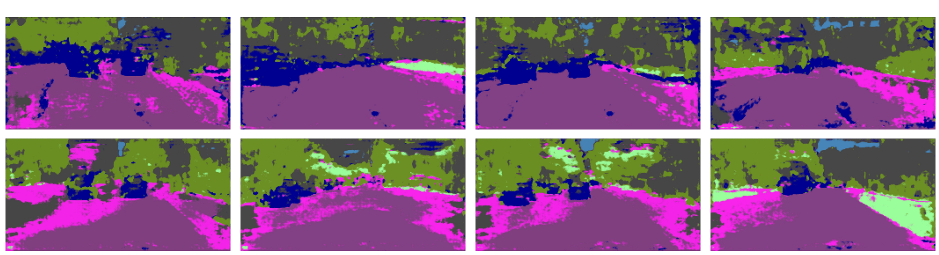

Figure 3. Semantic segmentation results on generated images using FCN. The top line shows the results using the original generator, and the bottom line illustrates the results using the compressed model.