Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net

Supplementary Material

Xingang Pan[1], Ping Luo[1], Jianping Shi[2], and Xiaoou Tang[1]

[1] CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong
{px117,pluo,xtang}@ie.cuhk.edu.hk
[2] SenseTime Group Limited
shijianping@sensetime.com

This document provides (1) a table comparing our method with related works, (2) detailed descriptions of some specific networks we use in the experiments, and (3) per-category results for Table 6 of the original paper.

1 Comparison with Related Works

Table 1. This table summarises the comparisons of our work with related works. "T" and "S" donote target and source respectively.

	invariance			model	reduce	cross	cost		
	spatial	appear	noise	cap.	overfit	domain	param	time	data label
Maxpool [12]	✓				✓				
Deformable [1]	✓			✓					
Dropout [17]			✓		✓				
IN [21]		✓			✓	✓			
BN [10]			✓		✓				
ResNet [4]	✓		✓	✓	✓				
ResNeXt [22]	✓		✓	✓	✓			↑	
SENet [7]	✓		✓	✓	✓		↑	↑	
DenseNet [8]	✓		✓	✓	✓				
StyleTransfer [21,2,9]	✓	✓	✓		✓	✓			w/T w/o
IBN-Net (ours)	✓	✓	✓	✓	✓	✓	→	→	w/o w/o
Finetuning						✓			w/T w/T
Domain Adapt [20,13,18,19,6,23,15,5]						✓			w/T w/o
Domain Gen. [11,14,3]						✓			w/S w/o

Table 1 summarises the comparisons of our work with related works in three different aspects, including operations that are invariant to specific variations, architectures of deep networks, and methods that improve performance across domains. They are summarized in multiple aspects, including invariance to which kind of variations, whether improving modeling capacity or reducing overfitting within one domain, whether improving generalization across domains, whether

increasing network parameters (#param) and running time, and whether requiring additional data or labels for generalizing across domains.

Our method shows two main advantages over related works. Firstly, IBN-Net is the only CNN architecture that improves both modeling and generalization capacities. Unlike ResNeXt and SENet, IBN-Net does not increase either model parameters or running time. Secondly, compared with other methods that improve cross-domain performance, our approach does not require additional data or labels, thus being more generic.

2 Details of Network Architectures

Table 2. Architectures for the original ResNet50 and its IBN-Net versions. Here "IBN-a" and "IBN-b" are as shown in Fig.3 of the original paper.

layer name	configuration	normalization type		
		ResNet50	IBN-Net50-a	IBN-Net50-b
conv1	7×7, 64, stride 2	BN	BN	IN
pool1	3×3 max pool, stride 2	–	–	–
conv2_x	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	BN	IBN-a	1-2nd blocks: BN 3rd block: IBN-b
conv3_x	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	BN	IBN-a	1-3rd blocks: BN 4th block: IBN-b
conv4_x	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	BN	IBN-a	BN
conv5_x	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	BN	BN	BN
classifier	average pool, 1000-d fc	–	–	–

In section 3.2 of the original paper, we have described the IBN-Net version of ResNet [4]. Here we summarize their architectures in Table 2 for clearer illustration. For ResNeXt [22] and SENet [7], the IBN-Net version naturally follows the same modification as for ResNet. Note that we have also warped IBN-Net to DenseNet [8], the architectures are shown in Table 3.

Table 3. Architectures for the original DenseNet121 [8] and its IBN-Net version. Here 'IBN' denotes a normalization layer with IN for 40% channels and BN for the rest 60% channels.

Layers	Output Size	DenseNet-121	DenseNet-121-IBN-a
Convolution	112 × 112	7 conv, stride 2	
Pooling	56 × 56	3 max pool, stride 2	
Dense Block (1)	56 × 56	[BN, 1 × 1 conv; BN, 3 × 3 conv] × 6	[IBN, 1 × 1 conv; BN, 3 × 3 conv] for the 1, 4th blocks, [BN, 1 × 1 conv; BN, 3 × 3 conv] for the rest
Transition Layer (1)	56 × 56	1 conv	
	28 × 28	2 average pool, stride 2	
Dense Block (2)	28 × 28	[BN, 1 × 1 conv; BN, 3 × 3 conv] × 12	[IBN, 1 × 1 conv; BN, 3 × 3 conv] for the 1, 4, 7, 10th blocks, [BN, 1 × 1 conv; BN, 3 × 3 conv] for the rest
Transition Layer (2)	28 × 28	1 conv	
	14 × 14	2 average pool, stride 2	
Dense Block (3)	14 × 14	[BN, 1 × 1 conv; BN, 3 × 3 conv] × 24	[IBN, 1 × 1 conv; BN, 3 × 3 conv] for the 1, 4, ..., 22th blocks, [BN, 1 × 1 conv; BN, 3 × 3 conv] for the rest
Transition Layer (3)	14 × 14	1 conv	
	7 × 7	2 average pool, stride 2	
Dense Block (4)	7 × 7	[BN, 1 × 1 conv; BN, 3 × 3 conv] × 16	[BN, 1 × 1 conv; BN, 3 × 3 conv] × 16
Classification Layer	1 × 1	7 global average pool	
		1000D fully-connected, softmax	

3 Per-category Results for Cityscapes-GTA5 Datasets

Table 4. Per-category results for Table 6 of the original paper.

Train	Test	Model	road	sidewalk	building	vall	fence	pole	t. light	t. sign
Cityscapes	Cityscapes	ResNet50	95.2	77.9	86.8	37.2	45.0	52.6	56.3	67.9
		IBN-Net50-a	96.5	**80.2**	**89.1**	40.5	**52.3**	**55.5**	**61.5**	**71.6**
		IBN-Net50-b	**96.9**	79.3	88.8	**43.7**	47.1	55.4	59.5	70.1
	GTA5	ResNet50	68.6	22.5	60.6	21.2	13.8	27.4	29.7	16.7
		IBN-Net50-a	74.3	26.0	53.0	21.8	18.3	33.1	34.0	**28.0**
		IBN-Net50-b	**79.3**	**27.9**	**68.4**	**31.7**	**20.0**	**35.6**	**40.2**	22.7

vegetation	terrain	sky	person	rider	car	truck	bus	train	motor	bicycle	**mIoU**
90.2	56.8	87.5	75.3	49.2	91.5	43.9	61.7	33.2	45.3	72.4	64.5
91.1	**60.5**	**91.8**	**78.0**	**54.2**	**92.9**	**51.8**	**70.5**	**47.4**	**54.3**	**73.4**	**69.1**
90.6	57.3	91.5	77.5	53.8	91.9	46.1	64.5	37.4	47.9	73.1	67.0
47.8	17.4	**85.1**	25.8	20.4	43.6	15.2	**10.5**	0.6	19.0	**12.5**	29.4
59.4	**29.5**	72.1	30.6	25.8	60.2	15.7	10.1	0.4	19.6	7.7	32.5
66.4	29.1	73.9	**53.6**	**29.4**	**67.0**	**25.9**	10.0	**0.9**	**27.6**	11.2	**37.9**

Train	Test	Model	road	sidewalk	building	vall	fence	pole	t. light	t. sign
GTA5	GTA5	ResNet50	94.5	77.0	83.7	43.0	26.4	39.8	38.2	40.1
		IBN-Net50-a	94.5	77.3	**86.1**	**48.5**	34.0	43.2	43.0	**46.9**
		IBN-Net50-b	**94.6**	**77.6**	85.7	47.5	**36.8**	**43.4**	**43.9**	46.5
	Cityscapes	ResNet50	20.1	17.2	47.1	4.9	13.9	20.2	**22.9**	8.5
		IBN-Net50-a	30.1	13.1	**57.8**	10.3	16.5	15.4	21.1	10.0
		IBN-Net50-b	**43.2**	**27.2**	49.8	**17.3**	**19.7**	**20.6**	16.9	**10.9**

vegetation	terrain	sky	person	rider	car	truck	bus	train	motor	bicycle	**mIoU**
77.3	55.6	93.7	66.1	45.9	85.5	75.3	72.8	55.3	52.3	37.2	61.0
79.7	**59.0**	**94.4**	**69.2**	**51.9**	86.8	**80.2**	**83.3**	**62.1**	53.3	37.4	**64.8**
79.3	57.4	94.3	67.0	50.8	**86.9**	79.0	80.6	54.5	**54.4**	**40.4**	64.2
73.6	6.3	52.6	47.9	**9.4**	43.3	5.8	12.2	0.2	8.5	**6.7**	22.2
81.8	12.0	**74.8**	50.4	1.2	65.2	9.8	8.7	0.0	**11.2**	4.2	26.0
81.8	**26.3**	63.2	**51.0**	3.1	**76.7**	**20.2**	**22.3**	**1.2**	7.0	4.7	**29.6**

Table 4 gives per-category IoU results for the Table 6 of the original paper. When train and evaluate on the same domain, IBN-Net50-a achieves best results for most object categories. And under cross evaluation setting, IBN-Net50-b performs better.

References

1. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. ICCV (2017)
2. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. ICLR (2017)
3. Ghifary, M., Bastiaan Kleijn, W., Zhang, M., Balduzzi, D.: Domain generalization for object recognition with multi-task autoencoders. ICCV (2015)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CVPR (2016)
5. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. arXiv preprint arXiv:1711.03213 (2017)
6. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)
7. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. arXiv preprint arXiv:1709.01507 (2017)
8. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. CVPR (2017)
9. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. ICCV (2017)
10. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML (2015)
11. Khosla, A., Zhou, T., Malisiewicz, T., Efros, A.A., Torralba, A.: Undoing the damage of dataset bias. ECCV (2012)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. NIPS (2012)
13. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. ICML (2015)
14. Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. ICML (2013)
15. Sankaranarayanan, S., Balaji, Y., Jain, A., Lim, S.N., Chellappa, R.: Unsupervised domain adaptation for semantic segmentation with gans. arXiv preprint arXiv:1711.06969 (2017)
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
17. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research (2014)
18. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. ECCV (2016)
19. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. CVPR (2017)
20. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)
21. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. CVPR (2017)
22. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. CVPR (2017)

23. Zhang, Y., David, P., Gong, B.: Curriculum domain adaptation for semantic segmentation of urban scenes. ICCV (2017)