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

Xingang Pan[1 0000−0002− 82 − 467], 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

Abstract. Convolutional neural networks (CNNs) have achieved great successes in many computer vision problems. Unlike existing works that designed CNN architectures to improve performance on a single task of a single domain and not generalizable, we present IBN-Net, a novel convolutional architecture, which remarkably enhances a CNN's modeling ability on one domain (*e.g.* Cityscapes) as well as its generalization capacity on another domain (*e.g.* GTA5) without finetuning. IBN-Net carefully integrates Instance Normalization (IN) and Batch Normalization (BN) as building blocks, and can be wrapped into many advanced deep networks to improve their performances. This work has three key *contributions*. (1) By delving into IN and BN, we disclose that IN learns features that are invariant to appearance changes, such as colors, styles, and virtuality/reality, while BN is essential for preserving content related information. (2) IBN-Net can be applied to many advanced deep architectures, such as DenseNet, ResNet, ResNeXt, and SENet, and consistently improve their performance without increasing computational cost. [1] (3) When applying the trained networks to new domains, *e.g.* from GTA5 to Cityscapes, IBN-Net achieves comparable improvements as domain adaptation methods, even without using data from the target domain. With IBN-Net, we won the 1st place on the WAD 2018 Challenge Drivable Area track, with an mIoU of 86.18%.

Keywords: Instance Normalization, Invariance, Generalization, CNNs

1 Introduction

Deep convolutional neural networks (CNNs) have improved performance of many tasks in computer vision, such as image recognition [1], object detection [21], and semantic segmentation [1]. However, existing works mainly design network architectures to solve the above problems on a single domain, for example, improving scene parsing on the real images of Cityscape dataset [2, 20]. When these networks are applied to the other domain of this scene parsing task, such as the

[1] **Code and models are available at** *https://github.com/XingangPan/IBN-Net*

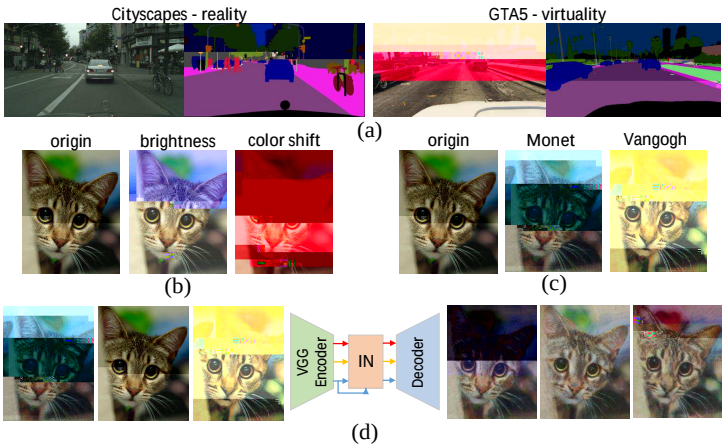


Fig. 1. (a) visualizes two example images (left) and their segmentation maps (right) selected from Cityscapes [2] and GTA5 [22] respectively. These samples have similar categories and scene configurations when comparing their segmentation maps, but their images are from di□erent domains, *i.e.* reality and virtuality. (b) shows simple appearance variations, while those of complex appearance variations are provided in (c). (d) proves that Instance Normalization (IN) is able to filter out complex appearance variance. The style transfer network used here is AdaIN [14]. (Best viewed in color)

virtual images of GTA dataset 22 their performance would drop notably. This is due to the appearance gap between the images of these two datasets as shown in Fig.1 a .

A natural solution to solve the appearance gap is by using transfer learning. For instance by netuning a CNN pretrained on Cityscapes using the data from GTA . we are able to adapt the features learned from Cityscapes to GTA . where accuracy can be increased. But even so the appearance gap is not eliminated because when applying the netuned CNN back to Cityscapes the accuracy would be signi cantly degraded. How to address large diversity of appearances by designing deep architectures. It is a key challenge in computer vision.

The answer is to induce appearance invariance into CNNs. This solution is obvious but non-trivial. For example there are many ways to produce the property of spatial invariance in deep networks such as max pooling 1 deformable convolution 3 which are invariant to spatial variations like poses viewpoints and scales but are not invariant to variations of image appearances. As shown in Fig.1 b. when the appearance variance of two datasets are simple and known beforehand such as lightings and infrared they can be reduced by explicitly augmenting data. However as shown in Fig.1 c. when appearance variance are complex and unknown such as arbitrary image styles and virtuality the CNNs have to learn to reduce them by introducing new component into their deep architectures.

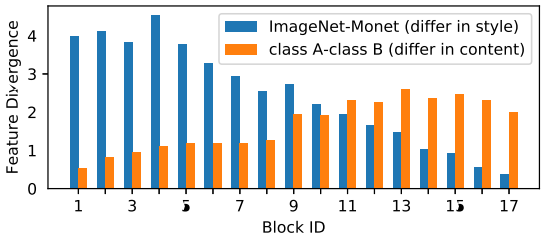


Fig. 2. (a) Feature divergence calculated from image sets with appearance difference (blue) and content difference (orange). We show the results of the 17 features after the residual blocks of ResNet50. The detailed definition of feature divergence is given in Section 4.3. The orange bars are enlarged 10 times for better visualization.

To this end, we present IBN-Net, a novel convolutional architecture, which learns to **capture** and **eliminate** appearance variance, while **maintains** discrimination of the learned features. IBN-Net carefully integrates Instance Normalization (IN) and Batch Normalization (BN) as building blocks, enhancing both its learning and generalization capacity. It has two appealing benefits that previous deep architectures do not have.

First, different from previous CNN structures that isolate IN and BN, IBN-Net unifies them by delving into their learned features. For example, many recent advanced deep architectures employed BN as a key component to improve their

BN layers to IN for a half of features and BN for the other half. These give rise to our IBN-Net.

Our **contributions** can be summarized as follows.

(1) A novel deep structure, IBN-Net, is proposed to **improve both** learning and generalization capacities of deep networks. For example, IBN-Net-50 achieves 22.37/6.32 and 51.71/28.11 top1/top5 errors on the original validation set of ImageNet [4] and a new validation set after style transformation respectively, outperforming ResNet-50 by 1.39/0.75 and 2.11/2.27, when they have similar numbers of parameters and computational cost.

(2) By delving into IN and BN, we disclose the key characteristics of their learned features, where IN provides visual and appearance **invariance**, while BN accelerates training and preserves **discriminative** features. This finding is important to understand them, and helpful to design the architecture of IBN-Net, where IN is preferred in shallow layers to remove appearance variations, whereas

presented simple appearance variations such as color or brightness shift could simply be eliminated by normalizing each RGB channel of an image with its mean and standard deviation. For more complex appearance transforms such as style transformations recent studies have found that such information could be encoded in the mean and variance of the hidden feature maps [14]. Therefore the instance normalization (IN [2]) layer shows potential to eliminate such appearance differences.

CNN Architectures. Since CNNs have shown compelling modeling capacity over traditional methods their architectures have gone through a number of developments. Among them one of the most widely used is the residual network (ResNet

and adversarial loss [2, 11]. Besides, [23] and [10] use generative adversarial networks (GAN) to transfer images between two domains to help adaptation, but required independent models for the two domains. There are two main limitations in transfer learning and domain adaptation. First, in real applications it is difficult to obtain the statistics of the target domain. It is also difficult to collect data that covers all possible scenarios in the target domain. Second, most state-of-the-art methods employ different model weights for the source and target domains in order to improve performance. But the ideal case is that one model could adapt to all domains.

Another paradigm towards this problem is domain generalization, which aims to acquire knowledge from a number of related source domains and apply it to a new target domain whose statistics is unknown during training. Existing methods typically design algorithms to learn domain agnostic representations or design models that capture common aspects from the domains, such as [1, 1, 1]. However, for real applications it is often hard to acquire data from a number of related source domains, and the performance highly depends on the series of source domains.

In this work, we increase the modeling capacity and generalization ability across domains by designing a new CNN architecture, IBN-Net. The benefit is that we do not require either target domain data or related source domains, unlike existing domain adaptation and generalization methods. The improvement of generalization across domains is achieved by designing architectures with built-in appearance invariance. Our method is extremely useful for the situations that the target domain data are unobtainable, where traditional domain adaptation cannot be applied. For more detailed comparison of our method with related works, please refer to our supplementary material.

3 Method

3.1 Background

Batch normalization [1] enables larger learning rate and faster convergence by reducing the internal covariate shift during training CNNs. It uses the mean and variance of a mini-batch to normalize each feature channels during training, while in inference phase, BN uses the global statistics to normalize features. Experiments have shown that BN significantly accelerates training, and could improve the final performance meanwhile. It has become a standard component in most prevalent CNN architectures like Inception [2], ResNet [6], DenseNet [13], etc.

Unlike batch normalization, **instance normalization** [2] uses the statistics of an individual sample instead of mini-batch to normalize features. Another important difference between IN and BN is that IN applies the same normalize procedure for both training and inference. Instance normalization has been mainly used in the style transfer field [2, , 14]. The reason for IN's success in style transfer and similar tasks is that, these tasks trying to change image appearance while preserving content, and IN allows to filter out instance-specific

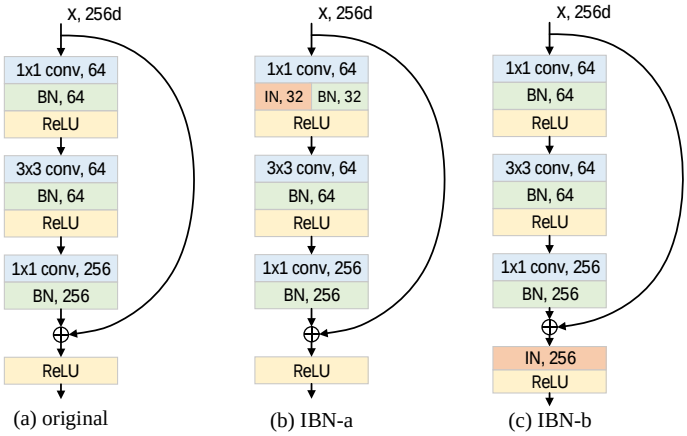
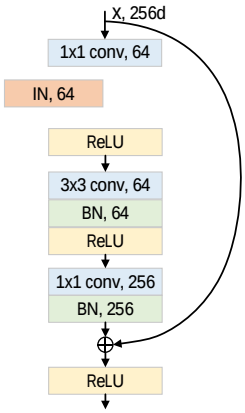


Fig. 3. Instance-batch normalization (IBN) block.

contrast information from the content. Despite these successes IN has not shown bene ts for high-level vision tasks like image classi cation and semantic segmentation. Ulyanov


x, 256d
1x1 conv, 64
IN, 64
ReLU
3x3 conv, 64
BN, 64
ReLU
1x1 conv, 256
BN, 256
ReLU

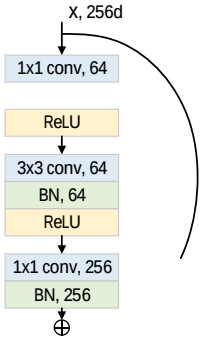

x, 256d
1x1 conv, 64
ReLU
3x3 conv, 64
BN, 64
ReLU
1x1 conv, 256
BN, 256

Table 1. Results on ImageNet validation set with appearance transforms. The performance drops are given in brackets.

appearance transform	ResNet50 [8] top1/top5 err.	IBN-Net50-a top1/top5 err.	IBN-Net50-b top1/top5 err.
origin	24.27/7.08	22.54/6.32	23.64/6.86
RGB+50	28.22/9.64 (3.94/2.56)	25.54/8.03 (3.00/1.71)	23.82/6.96 (0.18/0.10)
R+50	27.53/8.78 (3.26/1.70)	25.20/7.56 (2.66/1.24)	25.10/7.43 (1.46/0.57)
std $\times 1.5$	40.01/19.08 (15.74/12.00)	35.97/16.22 (13.43/9.90)	23.64/6.86 (0.00/0.00)
Monet	54.51/29.32 (30.24/22.24)	51.57/27.15 (29.03/20.83)	50.45/25.22 (26.81/18.36)

4 Experiments

We evaluate IBN-Net on both classi cation and semantic segmentation tasks

Table 2. Results of IBN-Net over other CNNs on ImageNet validation set. The performance gains are shown in the brackets. More detailed descriptions of these IBN-Nets are provided in the supplementary material.

help improve performance but loses generalization meanwhile. The combination of IBN-Net a and d makes little difference with d showing that the effects of INs on the main path of ResNet would dominate eliminating the effects of those on the residual path. Finally adding additional IBN layers to IBN-Net-a brings no good a moderate amount of IN features would suffice.

On the amount of IN and BN. Here we study IBN-Nets with different amount of IN layers added. Table.4 gives performance of IBN-Net 0-a with IN layers added to different amount of residual groups. It can be seen that the performance is improved with more IN layers added to shallow layers but decreased when IN layers are added to the last residual group. This indicates that IN in shallow layers help to improve modelling capacity while in deep layers BN should be kept to preserve important content information. Furthermore we study the effects of IN-BN ratio on the performance as shown in Table. . Again the best performance is achieved at a moderate ratio 0.2 -0. . demonstrating the trade-off relationship between IN and BN.

4.2 Cross Domain Experiments

If models trained with synthetic data could be applied to the real world it would save much effort for data collection and labelling. In this section we study our model s capacity to generalize across real and synthetic domains on Cityscapes and GTA datasets.

Cityscapes 2] is a traffic scene dataset collect from a number of European cities. It contains high resolution 204 $\times$ 1024 images with pixel level annotations of 34 categories. The dataset is divided into 2 for training 00 for validation and 1 2 for testing.

GTA5 22] is a similar street view dataset generated semi-automatically from the realistic computer game Grand Theft Auto V GTA . It has 12403 training images 3 2 validation images and 1 1 testing images of resolution 1 14 $\times$ 10 2 and the labels have the same categories as in Cityscapes.

Implementation. During training we use random scale aspect ratio and mirror for data augmentation. We apply random crop on full resolution images for Cityscapes and 1024 $\times$ 3 resized images for GTA . because this leads to better performance for both datasets. We use the 'poly' learning rate policy with base learning rate set to 0.01 and power set to 0. . We train the models

Table 6. Results on Cityscapes-GTA dataset. Mean IoU for both within domain evaluation and cross domain evaluation is reported.

Train	Test	Model	mIoU(%)	Pixel Acc.(%)
Cityscapes	Cityscapes	ResNet50	64.5	93.4
		IBN-Net50-a	69.1	94.4
		IBN-Net50-b	67.0	94.3
	GTA5	ResNet50	29.4	71.9
		IBN-Net50-a	32.5	71.4
		IBN-Net50-b	37.9	78.8
GTA5	GTA5	ResNet50	61.0	91.5
		IBN-Net50-a		

and labels. The initial learning rate and the number of epochs is set to 0.003 and 0 respectively. As Table. shows with only 30 of Cityscapes training data IBN-Net 0-a outperforms resnet 0 netuned on all the data.

4.3 Feature Divergence Analysis

In order to understand how IBN-Net achieves better generalization we analyse the feature divergence caused by domain bias in this section. Our metric for feature divergence is as follows. For the output feature of a certain layer in a CNN we denote the mean value of a channel as F which basically describes how much this channel is activated. We assume a Gaussian distribution of F with mean μ and variance σ^2. Then the symmetric KL divergence of this channel between domain A and B would be

$$F_A||F_B \qquad L\ F_A||F_B \qquad L\ F_B||F_A \qquad 1$$

$$L\ F_A||F_B \qquad log\frac{\sigma_A}{\sigma_B}$$ nce FL0980231]TJ/R228 9.9626 Tf 6.49023 1.49414 Td 8

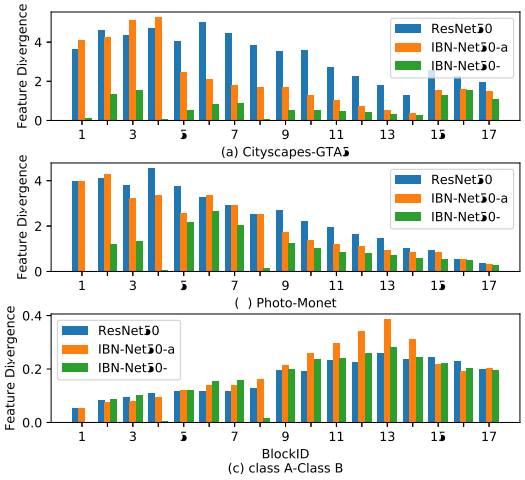


Fig. 5. Feature divergence caused by (a) real-virtual appearance gap, (b) style gap, (c) object class di□erence.

they could work in a manner that helps to lter out the appearance variance within features. In this way the models robustness to appearance transforms is improved as shown in our experiments.

Note that generalization and modelling capacity are not uncorrelated prop-

References

1. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI (2017)
2. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
3. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV (2017)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
5. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. In: ICLR (2017)
6. Ghifary, M., Bastiaan Kleijn, W., Zhang, M., Balduzzi, D.: Domain generalization for object recognition with multi-task autoencoders. In: ICCV (2015)
7. Gross, S., Wilber, M.: Training and investigating residual nets. https://github.com/ facebook/fb.resnet.torch (2016)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
9. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: ECCV (2016)
10. Ho□man, 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)
11. Ho□man, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)
12. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. arXiv preprint arXiv:1709.01507 (2017)
13. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely49828(u)-1.002(,)-494.497(Z)-0.0980222(.)-n

23. 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)
24. 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)
25. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: E0980222(o)-0CV (2016)