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

Pan et al., ECCV 2018.

Presenter: Yoonki Cho

# Contents

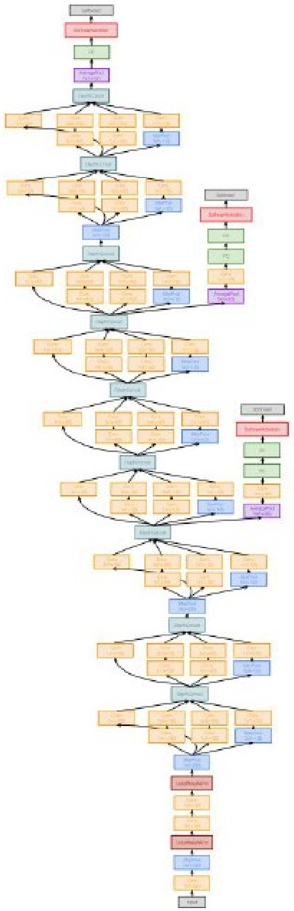
# Introduction

## Motivation & Research Goal

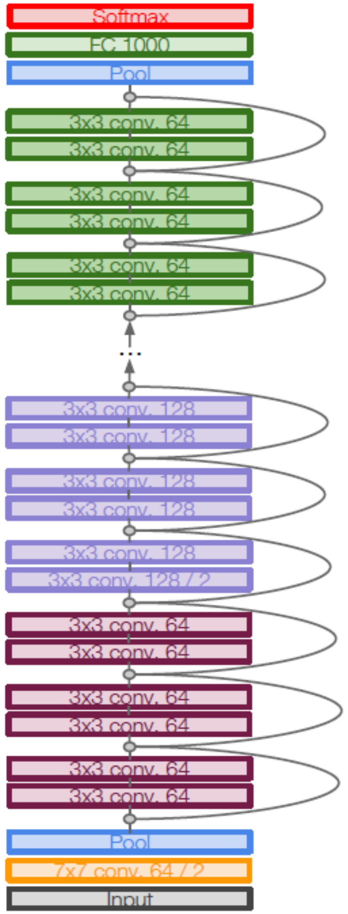
# Problems of Existing CNN Architectures

- Existing studies focus on **improving the performance on a single domain**.


VGG



GoogLeNet




ResNet

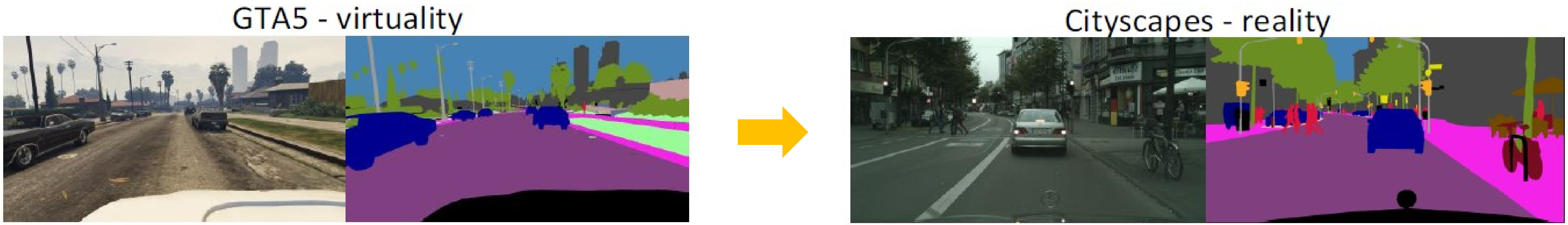




Image credit: © Stanford CS231n: Convolutional Neural Networks for Visual Recognition

# Problems of Existing CNN Architectures

- Can we directly apply the model trained on domain A to domain B?
    - No. Most CNN architectures are not generalizable to unseen domain.
    - There are always domain gaps.



# Problems of Existing CNN Architectures

- To overcome this issue,
  - Transfer Learning.
    - Fine-tune the model on the target domain with labels.
  - Domain Adaptation.
    - Use the statistics of the target domain to facilitate adaptation.
  - Domain Generalization.
    - Train the model on several source domains to improve the generalization.

# Research Goal

- This paper aims to develop the CNN architecture to improve the learning and generalization capacities.

- It does not require either target domain data or related source domains.

# Backgrounds

Batch & Instance Normalization

# Batch Normalization (BN)

- It enables faster convergence of CNN architecture.
  - o Reduce the internal covariate shift.
  - o Reduce the effects of noise in the mini-batch sampling.
- Nowadays, it is the standard component in recent CNN architectures.

**Input:** Values of $x$ over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: $\gamma$, $\beta$

**Output:** $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

# Instance Normalization (IN)

- It normalize the mean&std of individual data sample in mini-batch.

- IN is often used in the style transfer.
    - It can control the style (appearance) of images.
    - It filter out the instance-specific contrast information.

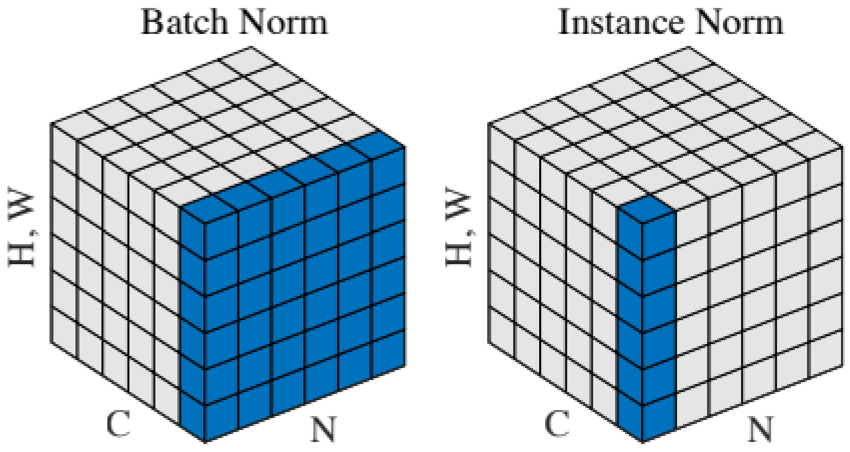





Image credit: © Wu and He, Group Normalization, ECCV 2018.

# Method

Instance-Batch Normalization Networks (IBN-Net)

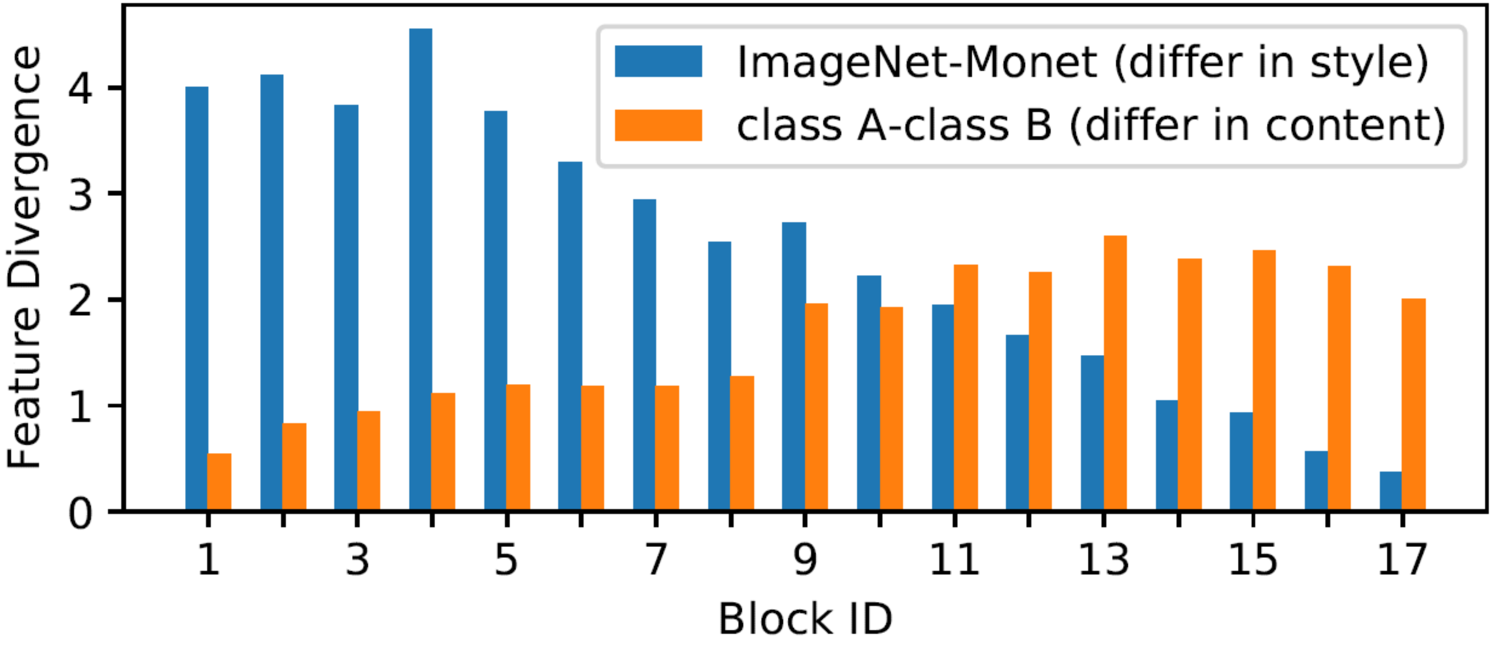
# Instance-Batch Normalization Networks (IBN-Net)

- The authors insist that the appearance variance (e.g., style) causes the domain gap.
  - Instance Normalization can control the style variance between images.

- This paper present a novel CNN architecture named IBN-Net.
  - Carefully integrates Instance Normalization (IN) and Batch Normalization (BN) in network layers.

# Experimental observation

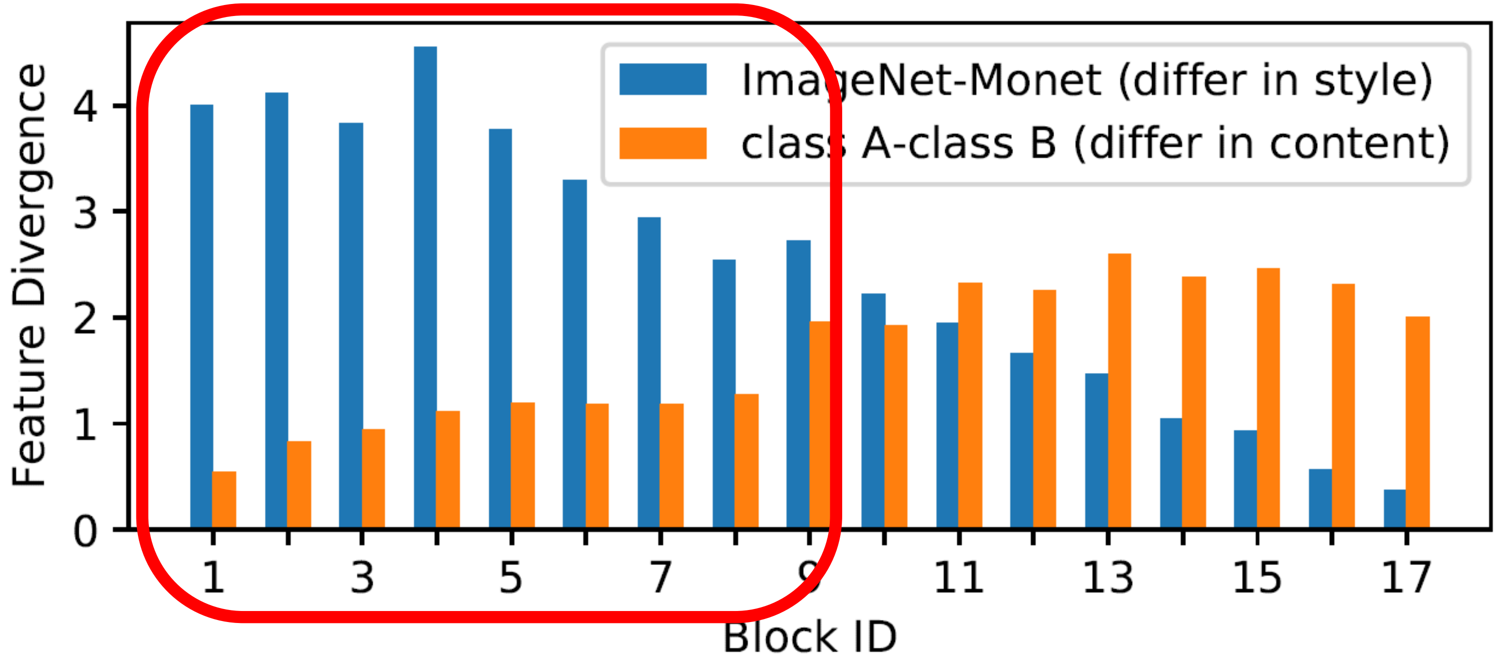

- Feature divergence between style and content variance.




**Depth** ↑

# Experimental observation

- Feature divergence between style and content variance.




**Shallow layers are more sensitive to style variance!**

# Instance-Batch Normalization Networks (IBN-Net)

- From the observation,
  - They used IN and BN simultaneously in shallow layers (Style invariance).
  - To prevent the content discrimination, they did not add INs in deeper layers (Content maintenance).
- The authors propose the IBN-block for style invariance & content maintenance.
  - They replace the first three convolution groups (blocks) to IBN-blocks.

# IBN-block

- IBN-block for ResNet-like architecture.

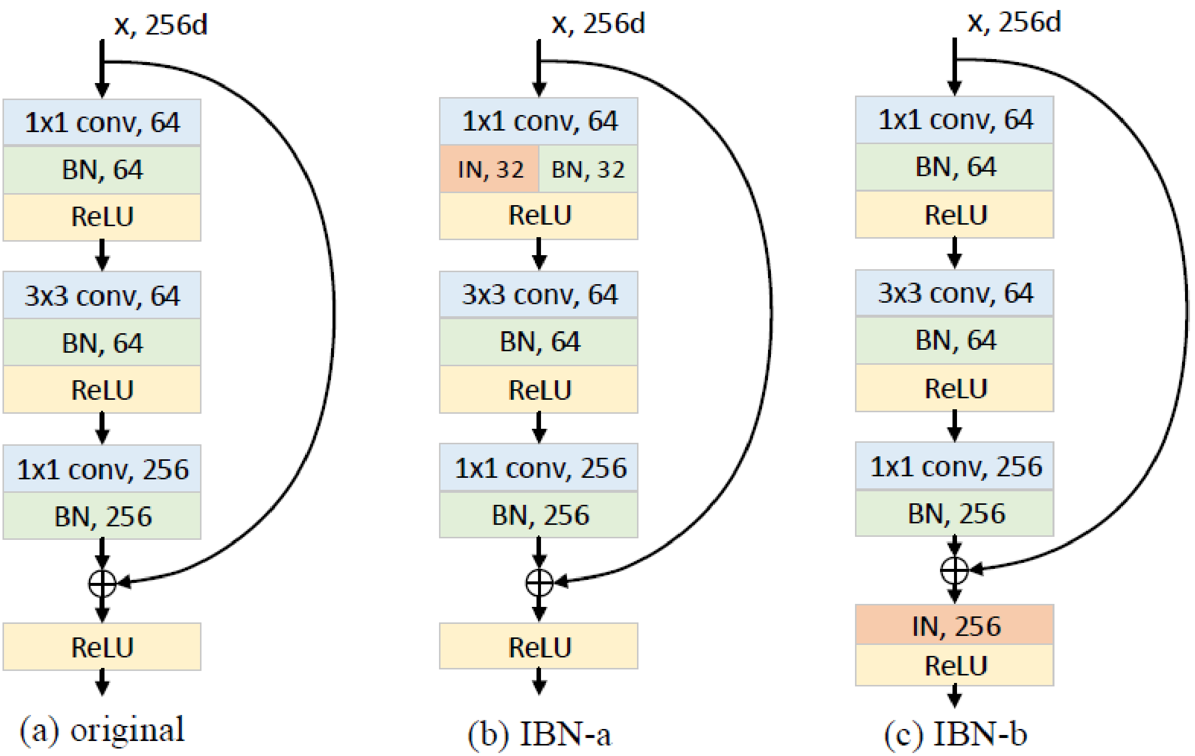



**Fig. 3.** Instance-batch normalization (IBN) block.

# IBN-block

- It can be extended in various versions.

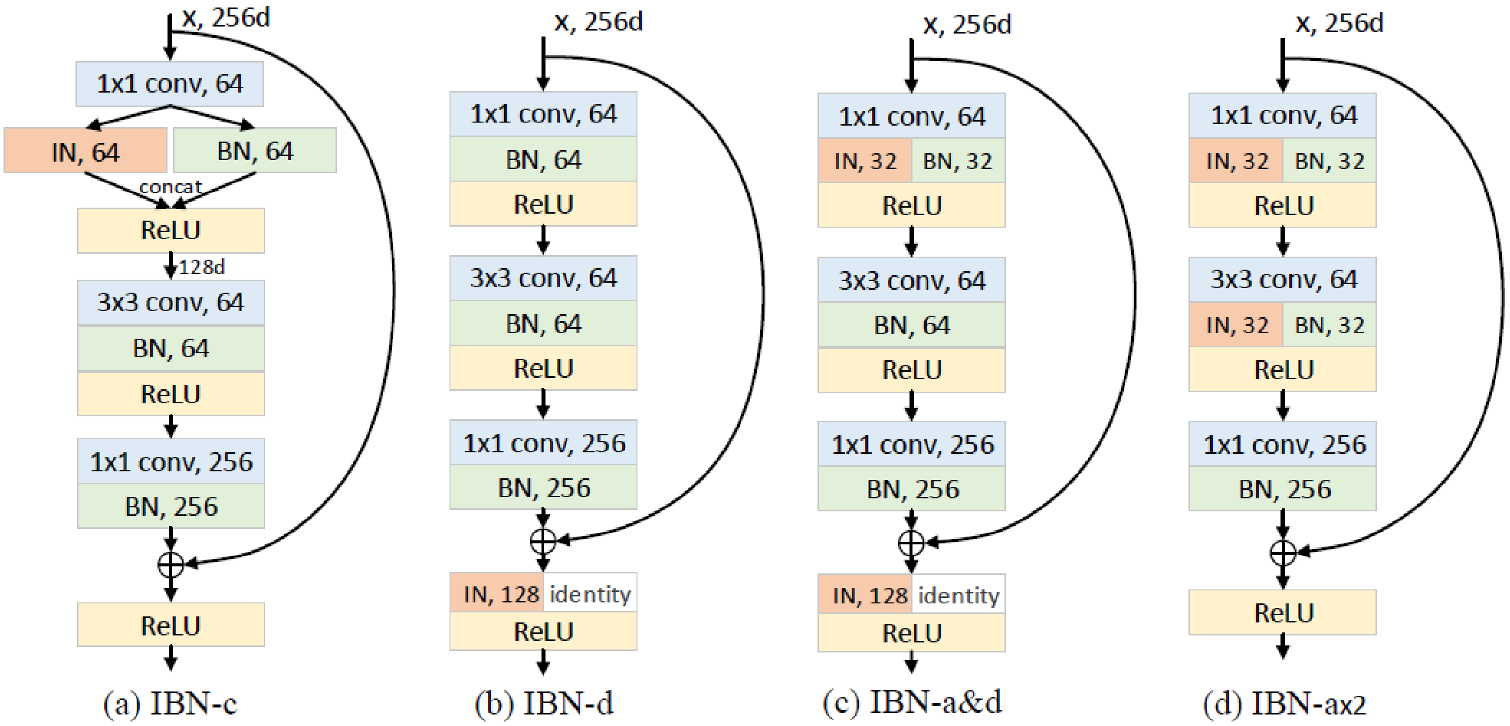



**Fig. 4.** Variants of IBN block.

# Experiments

# Results on ImageNet

- ImageNet results according to various appearance changes.

**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 |
| **Color** | 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)** |
| **Brightness** | 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)** |
| **Contrast** | std ×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)** |
| **Style** | 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)** |

# Results on ImageNet

- Comparison over other CNN architectures.

**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.

| Model | original | re-implementation | IBN-Net-a |
|---|---|---|---|
| | top1/top5 err. | top1/top5 err. | top1/top5 err. |
| DenseNet121 [13] | 25.0/- | 24.96/7.85 | 24.47/7.25 (0.49/0.60) |
| DenseNet169 [13] | 23.6/- | 24.02/7.06 | 23.25/6.51 (0.79/0.55) |
| ResNet50 [8] | 24.7/7.8 | 24.27/7.08 | 22.54/6.32 (1.73/0.76) |
| ResNet101 [8] | 23.6/7.1 | 22.48/6.23 | 21.39/5.59 (1.09/0.64) |
| ResNeXt101 [31] | 21.2/5.6 | 21.31/5.74 | 20.88/5.42 (0.43/0.32) |
| SE-ResNet101 [12] | 22.38/6.07 | 21.68/5.88 | 21.25/5.51 (0.43/0.37) |

# Results on ImageNet

- Comparison with different design choices.

**Table 4.** Comparison of IBN-Net50-a with IN layers added to different amount of residual groups.

| Residual groups | none | 1 | 1-2 | 1-3 | 1-4 |
|---|---|---|---|---|---|
| top1 err. | 24.27 | 23.58 | 22.94 | **22.54** | 22.96 |
| top5 err. | 7.08 | 6.72 | 6.40 | **6.32** | 6.49 |

**Table 5.** Effects of the ratio of IN channels in the IBN layers. 'full' denotes ResNet50 with all BN layers replaced by IN.

| IN ratio | 0 | 0.25 | 0.5 | 0.75 | 1 | full |
|---|---|---|---|---|---|---|
| top1 err. | 24.27 | **22.49** | 22.54 | 23.11 | 23.44 | 28.56 |
| top5 err. | 7.08 | 6.39 | **6.32** | 6.57 | 6.94 | 9.83 |

# Results on Semantic Segmentation

- Results on cross-domain semantic segmentation.

**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 | **64.8** | **92.5** |
| | | IBN-Net50-b | 64.2 | 92.4 |
| | Cityscapes | ResNet50 | 22.2 | 53.5 |
| | | IBN-Net50-a | 26.0 | 60.9 |
| | | IBN-Net50-b | **29.6** | **66.8** |

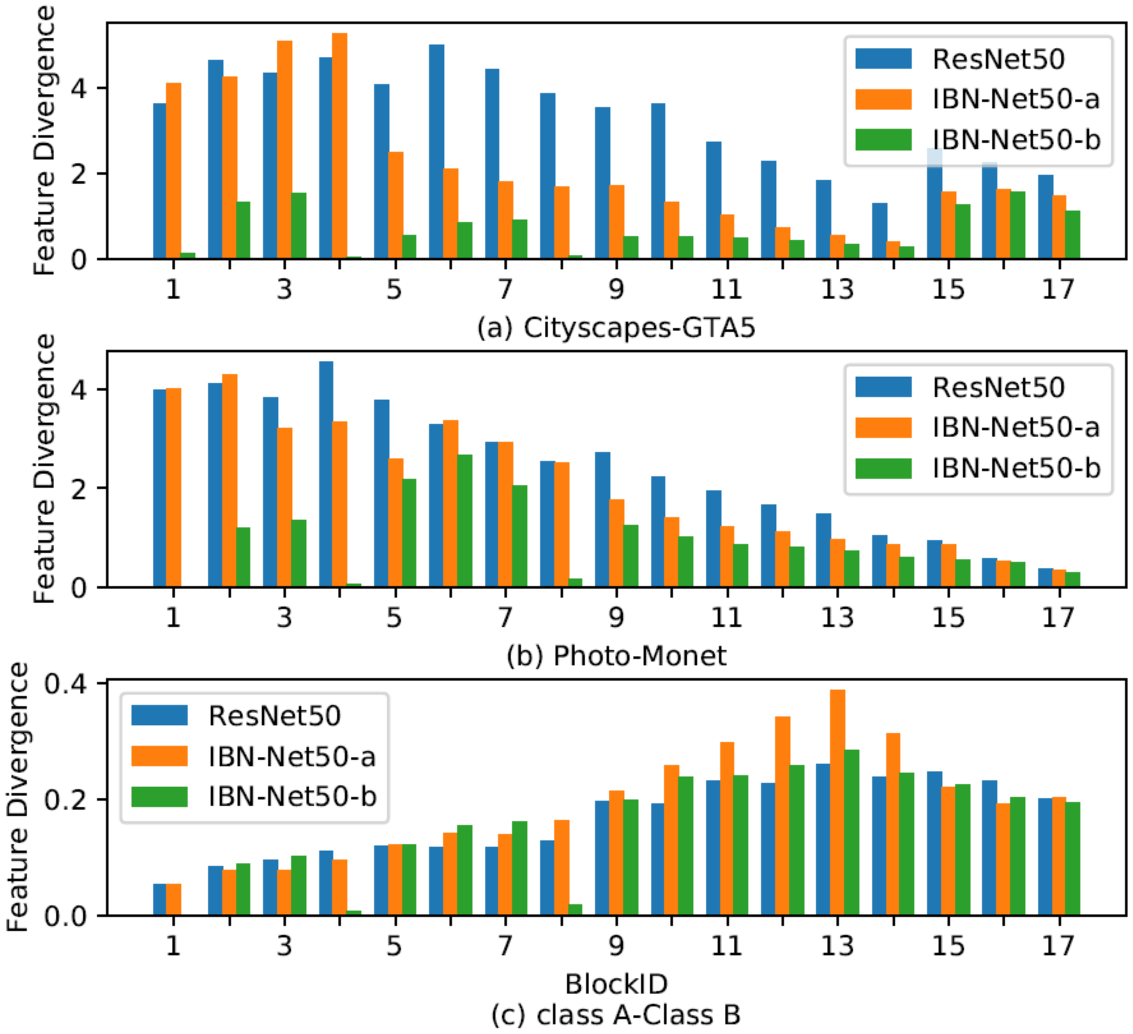
# Results on Semantic Segmentation

- Comparison with domain adaptation methods.

**Table 7.** Comparison with domain adaptation methods. Note that our method does not use target data to help adaptation.

| Method | mIoU | mIoU gain | Target data |
|---|---|---|---|
| Source only [11] | 21.2 | 5.9 | w/ |
| FCN wild [11] | 27.1 | | |
| Source only [32] | 22.3 | 6.6 | w/ |
| Curr. DA [32] | 28.9 | | |
| Source only [24] | 29.6 | **7.5** | w/ |
| GAN DA [24] | 37.1 | | |
| Ours - Source only | 22.17 | **7.5** | **w/o** |
| Ours - IBN - Source only | 29.64 | | |

# Feature Divergence Results



# Summary & Conclusion

- The authors propos a novel CNN architecture called IBN-Net.
- IBN-Net achieves competitive results compared to recent CNN models.
- IBN-Net shows strong generalization capacities.
- IBN-Net can easily applied to the existing CNN model.

Thank you for the listening!

# Appendix

**Table 3.** Results of IBN-Net variants on ImageNet validation set and Monet style set.

| Model | origin | Monet |
| --- | --- | --- |
| | top1/top5 err. | top1/top5 err. |
| ResNet50 | 24.26/7.08 | 54.51/29.32 (30.24/22.24) |
| IBN-Net50-a | **22.54/6.32** | 51.57/27.15 (29.03/20.83) |
| IBN-Net50-b | 23.64/6.86 | **50.45/25.22 (26.81/18.36)** |
| IBN-Net50-c | 22.78/**6.32** | 51.83/27.09 (29.05/20.77) |
| IBN-Net50-d | 22.86/6.48 | 50.80/26.16 (27.94/19.68) |
| IBN-Net50-a&d | 22.89/6.48 | 51.27/26.64 (28.38/20.16) |
| IBN-Net50-a×2 | 22.81/6.46 | 51.95/26.98 (29.14/20.52) |