Generalized Lightness Adaptation using Channel Selective Normalization ——Supplementary Material——

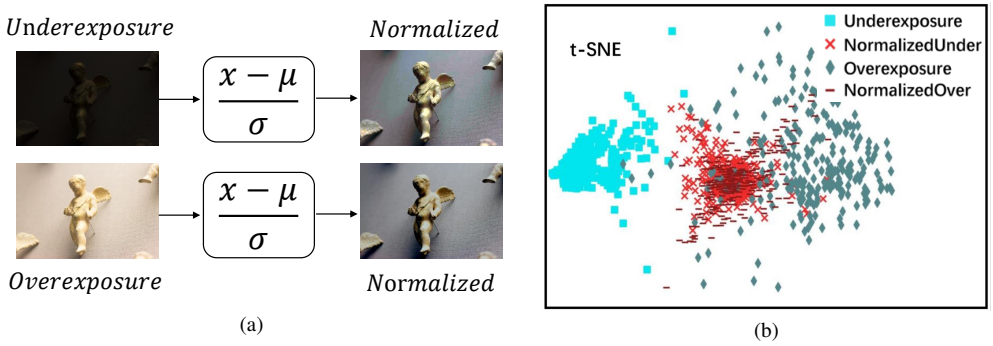
Mingde Yao[1], Jie Huang[1], Xin Jin[2], Ruikang Xu[1], Shenglong Zhou[1], Man Zhou[3], Zhiwei Xiong[1]
[1]University of Science and Technology of China
[2]Eastern Institute of Technology
[3]Nanyang Technological University

The supplementary material is organized as follows:

- Section 1 provides more analyses on normalization.
- Section 2 provides more analyses on channel selection, including additional visualizations of selected channels, the ratio of selected channels, and the robustness of the selection.
- Section 3 provides details on the calculation of the number of parameters.
- Section 4 provides the running times and FLOPs of our CSNorm.
- Section 5 provides more information about datasets and experiments.
- Section 6 provides additional visual results to comprehensively demonstrate the effectiveness of the proposed CSNorm.
- Section 7 discusses the strengths and weaknesses of the proposed CSNorm, as well as possible directions for future work.

1. More analyses on normalization

We visually demonstrate the effectiveness of normalization in extracting lightness-invariant information. In Fig.1(a), we observe that images captured under different exposures exhibit significant differences and discrepancies between them. However, when we normalize the images by subtracting their mean values μ and dividing the standard deviations σ, the normalized images exhibit similar representations, largely eliminating the discrepancies between them. To provide a more comprehensive analysis, we performed a cluster analysis of the overexposed and underexposed images before and after normalization, as shown in Fig.1(b). The results indicate that underexposure and overexposure images tend to overlap after normalization, indicating that normalization effectively extracts lightness-invariant components. These observations provide statistical evidence of the effectiveness of normalization in lightness adaptation.



(a) (b)

Figure 1. The illustration of normalization for bridging the gap of different lightness conditions. (a) images before and after normalization. (b) t-SNE [9] visualization of the images (Huawei dataset [6]) before and after normalization. The underexposure and overexposure images tend to be intersected together after normalization.

2. More analyses on channel selection

First, we provide the visualization results of the channels before and after normalization in Fig. 2 and 3, where the channels are selected by CSNorm. As can be seen, the channels extracted from the same image are inconsistent due to different illumination, which leads to inconsistent outputs. After the selection and normalization of CSNorm, the appearances of these channels are substantially closer, making the network avoid the effects of illumination and obtain better generalization.

Second, we observe that the number of lightness-relevant channels is *nearly* robust in the feature space. Although different images have different contents and lightness, the number of normalized channels is *nearly* the same among different images. For example, the overall gating ratio is about 1/3 channels on the low-light image enhancement task, *e.g.,* about 20/64 channels are selected to be normalized in the SID network and 10/32 channels in the DRBN network. Similar phenomena are observed in the tasks of inverse tone mapping and image retouching.

Third, benefiting from the gating module G, our model can not only adaptively select the lightness-relevant channels but also perform different selections determined by the input image itself. This is not contradictory to the point above, because the change is very small, roughly in one to two-channel changes in our experiments (including low-light image enhancement, image retouching, and inverse tone mapping tasks).

3. Number of parameters

In this section, we provide an analysis of the parameters in the CSNorm network. These parameters consist of 1) the affine parameters in the Instance Normalization (IN) layer, and 2) the parameters in the gating module G.

The number of parameters in the IN layer is proportional to the number of input channels, with each channel having its own scale and shift parameters. Specifically, the number of parameters in the IN layer is $2N$, where N is the number of input channels.

For the gating module, the parameters are only maintained in the two fully connected layers. Specifically, given a fully connected layer with N input channels and M output channels, the number of parameters in the fully connected layer is $N \times M + M$, where $N \times M$ is the number of weights connecting the input and output channels, and M is the number of bias terms.

The total number of parameters in the CSNorm network can be calculated as follows

$$\begin{aligned} \text{Total Parameters} =& T_{\text{IN}} + T_{\text{FC}} \\ =& 2N + (N \times M + M) + (M \times N + N), \end{aligned} \tag{1}$$

where T_{IN} is the number of parameters in the IN layer, and T_{FC} is the number of parameters in the gating module layers. Note that the gating module's first fully connected layer maps the number of input channels from N to M, and the second fully connected layer map the number of channels from M back to N.

For example, given a feature with 64 channels and $M = 128$, the number of parameters in CSNorm is $2 \times 64 + (64 \times 128 + 128) + (128 \times 64 + 64) = 16640$, which means CSNorm only takes about 16k parameters. In this way, the number of parameters grows linearly with the number of channels (N), when M is a constant.

4. Running times and FLOPs

We also provide the running times and FLOPs of our CSNorm. As shown in Table 1, equipped with our CSNorm, the running time of the model remains almost unchanged. The increase in FLOPs is so small that it is nearly negligible compared to the FLOPs of the base model.

Methods	DRBN [11]	**DRBN-CSNorm**	SID [2]	**SID-CSNorm**	NAFNet [3]	**NAFNet-CSNorm**
FLOPs (G)	41.4157	41.4157 (+0.00)	54.7441	54.7441 (+0.00)	54.5231	54.5231 (+0.00)
Time (s)	0.09	0.10 (+0.01)	0.05	0.05 (+0.00)	0.10	0.11 (+0.01)

Table 1. Comparison of running time and FLOPs.

5. Experimental details

5.1. Datasets

Low light image enhancement To validate the effectiveness of CSNorm on low-light image enhancement, we use two datasets: LOL [10] and Huawei [6]. Specifically, we train our model on 485 pairs of images from the LOL dataset [10] and tested it on 15 pairs. We used 2200 pairs from the Huawei dataset [6] for training, with 280 pairs reserved for testing purposes. The training images were randomly cropped into patches of size 256×256, while the testing images were not cropped and directly fed into the network.

Inverse tone mapping We conducted inverse tone mapping experiments using the HDRTV [4] and Kim *et al.* [8] datasets. The HDRTV dataset consists of 1235 pairs of images for training and 117 pairs for testing, while the Kim *et al.* dataset has 39840 pairs of patches for training and 28 pairs of images for testing. To ensure consistency between the two datasets, we converted the Kim *et al.* dataset from the Ycbcr color space to the RGB color space, aligning it with HDRTV [4] dataset. This conversion process eliminated the color space discrepancy between the two datasets. It is important to note that the conversion functions used for the input SDR images and target HDR images are different, as the input images are standardized with Rec.709 while the target images follow Rec.2020.

Image retouching The MIT-Adobe FiveK [1] dataset contains RAW photos and corresponding retouched versions created by five experts (A/B/C/D/E). We employ the retouched results of expert C as the ground truth following previous methods [5, 7]. We use the same pre-processing steps as in [7]. We randomly choose 200 photographs for testing and the remaining 800 images for training.

5.2. Detail implementations

We provide detailed implementations of our experiments. We train all the networks with their official codes on different tasks and datasets. For base networks equipped with our proposed CSNorm, we insert the CSNorm at the middle blocks of the base networks. All the models are trained and tested on one GTX 3090Ti GPU.

6. Additional visual comparisons of final results

We provide more visualization results on low-light image enhancement, inverse tone mapping, and image retouching tasks, as shown in Fig. 4–8. All the models are trained and tested on different datasets (or different lightness conditions), to validate the generalization ability of the proposed CSNorm. It can be seen that, the base models have a poor generalization ability across different lightness conditions. However, when they equip our CSNorm, which is lightweight and plug-and-play, they obtain the generalization ability to perform well on other unknown lightness conditions.

7. More discussions

In this paper, we introduce CSNorm, a novel normalization method for generalized lightness adaptation. As CSNorm does not need to collect extra training datasets under different lightness conditions, its application scenarios could be much broader than both previous models. Besides, the proposed CSNorm has the potential to be utilized in other tasks to enhance the generalization capability. It can be plugged into any existing network that supports the normalization layer. Additionally, CSNorm could facilitate the development of the specific community, such as smartphone photography. However, the proposed method still faces the problem of being unable to handle extreme lighting conditions, and the effect on noise is neglected in the method design. We will address these issues in our future work. Overall, we believe that CSNorm will have a significant positive impact on both computer vision industries and academia.

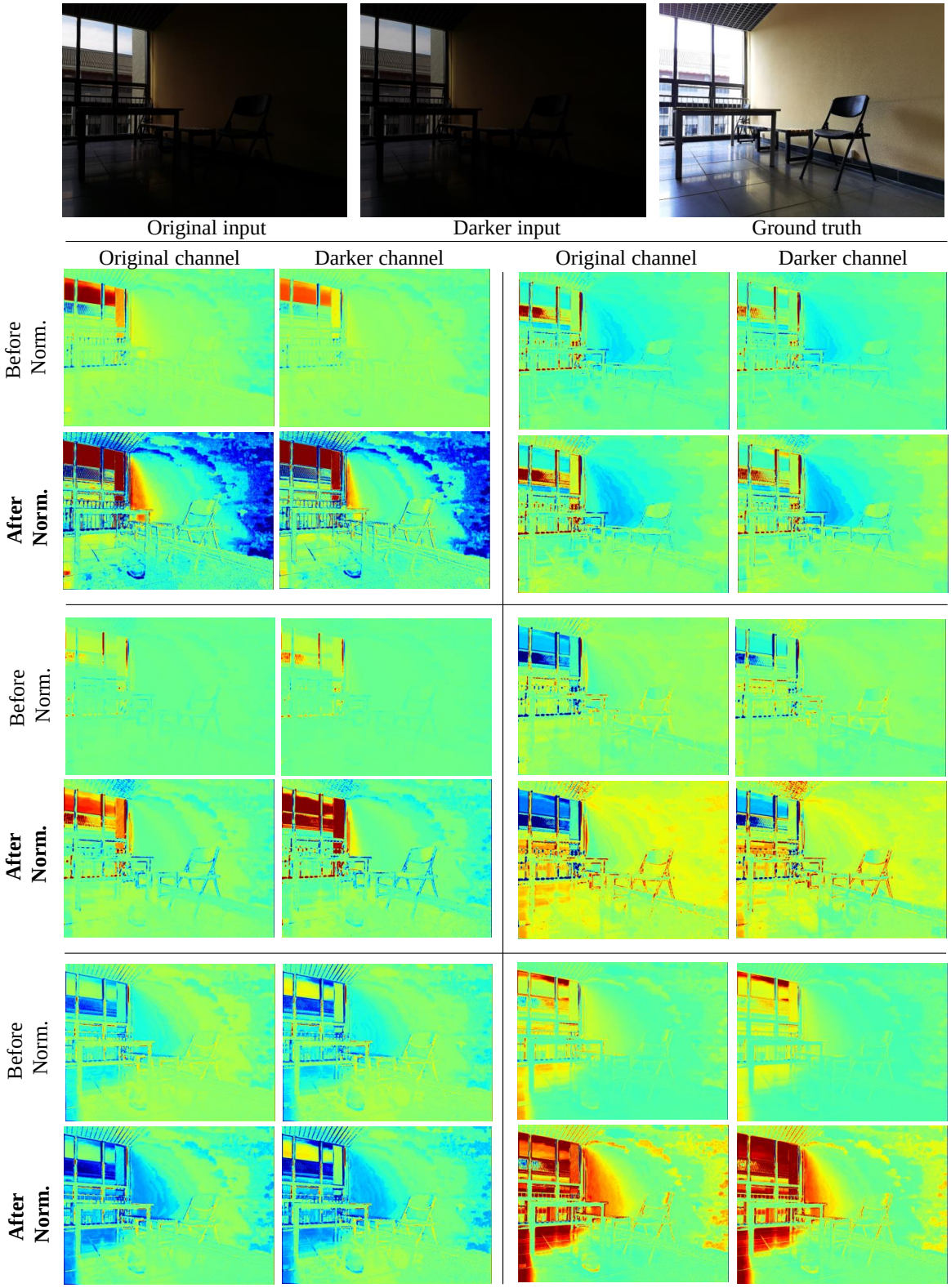


Figure 2. Channels selected by CSNorm before and after normalization. The base network is SID [2] which is trained on the LOL [10] and tested on the Huawei [6] dataset.

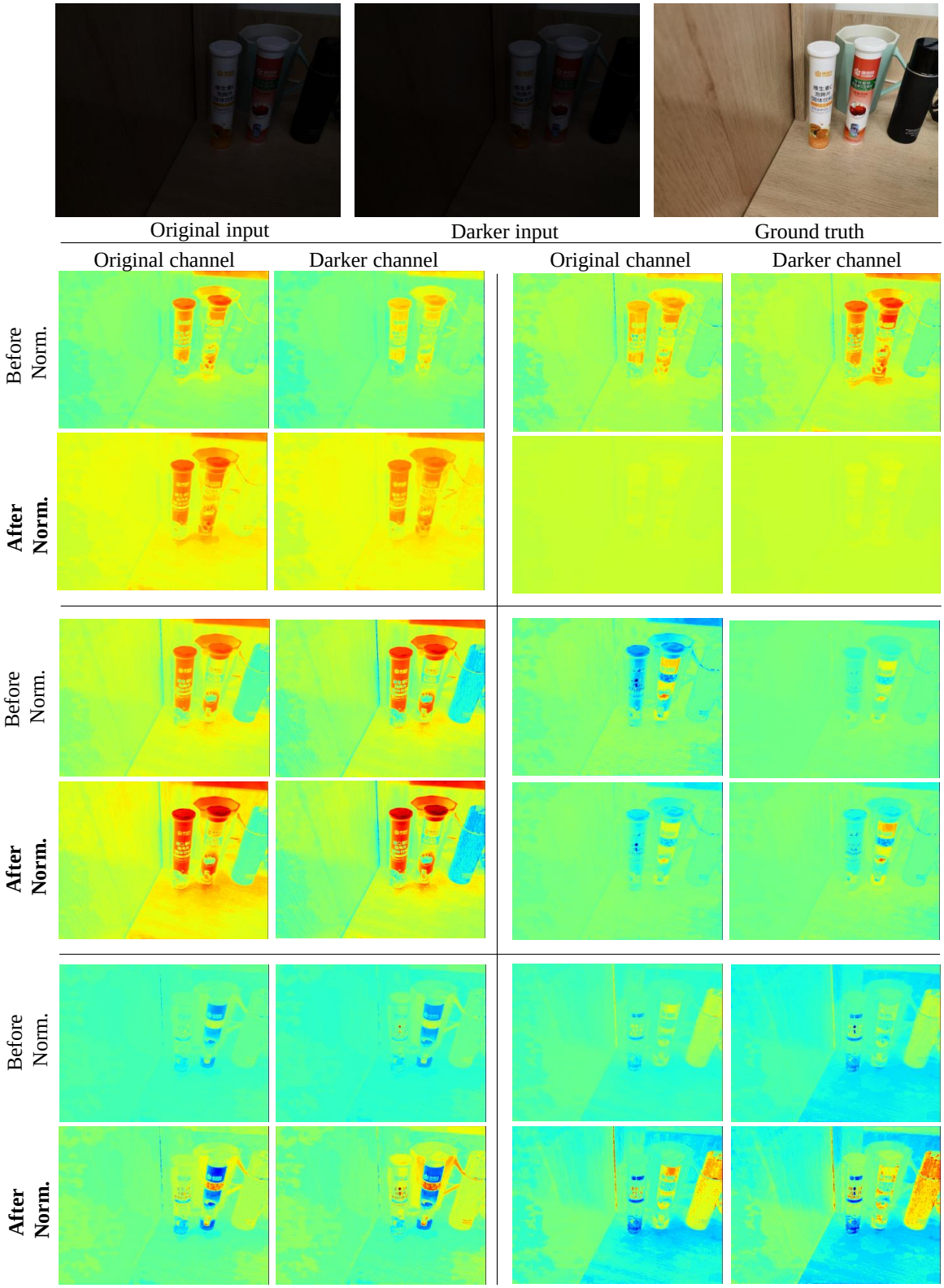


Figure 3. Channels selected by CSNorm before and after normalization. The base network is SID [2] which is trained on the LOL [10] and tested on the Huawei [6] dataset.

Figure 4. Visual results of generalized low-light image enhancement on the Huawei [6] dataset. The models are trained on the LOL [10] dataset and tested on the Huawei [6] dataset. Equipped with our CSNorm, the generalization abilities of base networks (DRBN [11] and NAFNet [3]) are significantly improved.

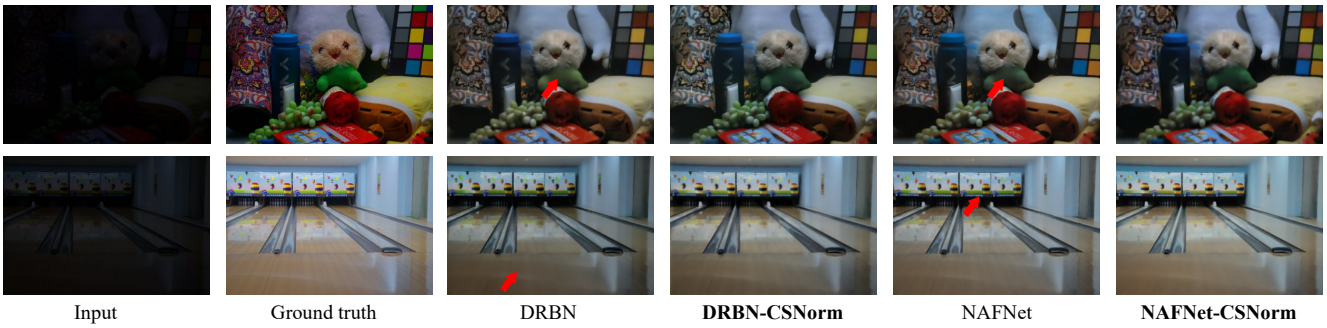


Figure 5. Visual results of generalized low-light image enhancement on the LOL [10] dataset. The models are trained on the Huawei [6] dataset and tested on the LOL [10] dataset. Equipped with our CSNorm, the generalization abilities of base networks (DRBN [11] and NAFNet [3]) are significantly improved.

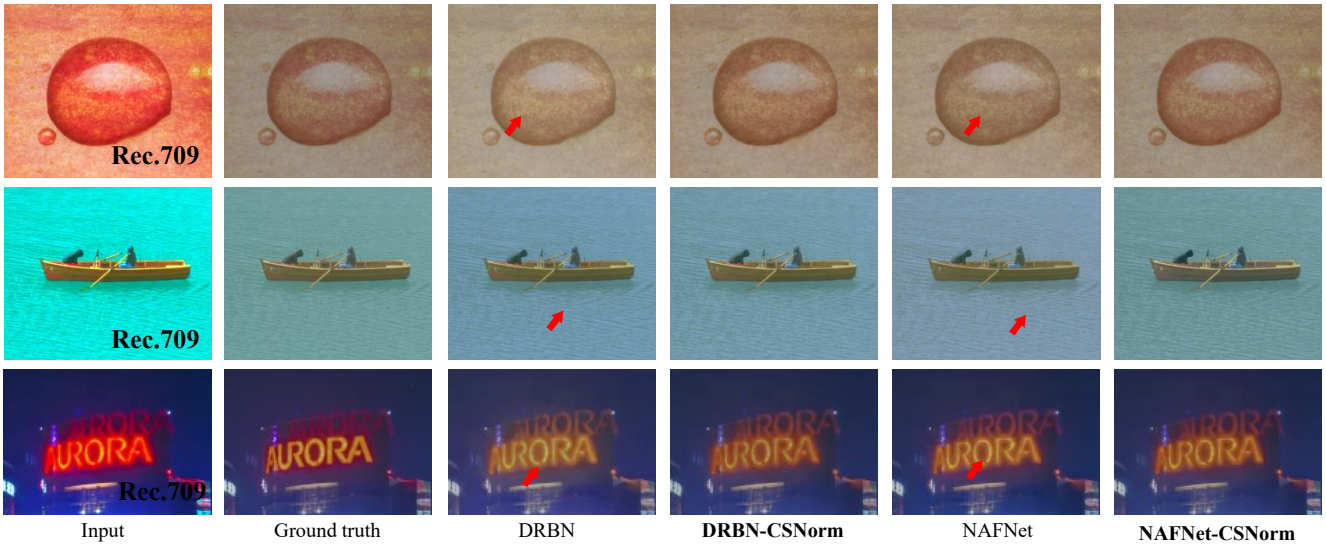


Figure 6. Visual results of the generalized inverse tone mapping on the HDRTV [4] dataset. The models are trained on the Kim *et al.* [8] dataset and tested on the HDRTV [4] dataset. The colors of results seem light-colored since they are visualized in the standard Rec.2020 color space.

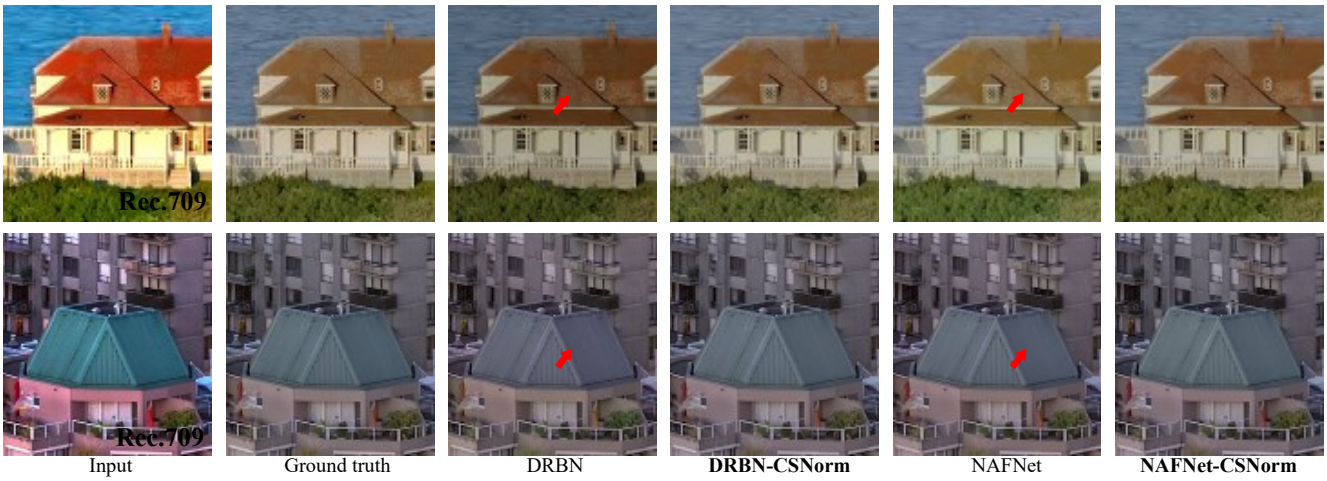


Figure 7. Visual results of the generalized inverse tone mapping on the Kim *et al.* [8] dataset. The models are trained on the HDRTV [4] dataset and tested on the Kim *et al.* [8] dataset. The colors of results seem light-colored since they are visualized in the standard Rec.2020 color space.

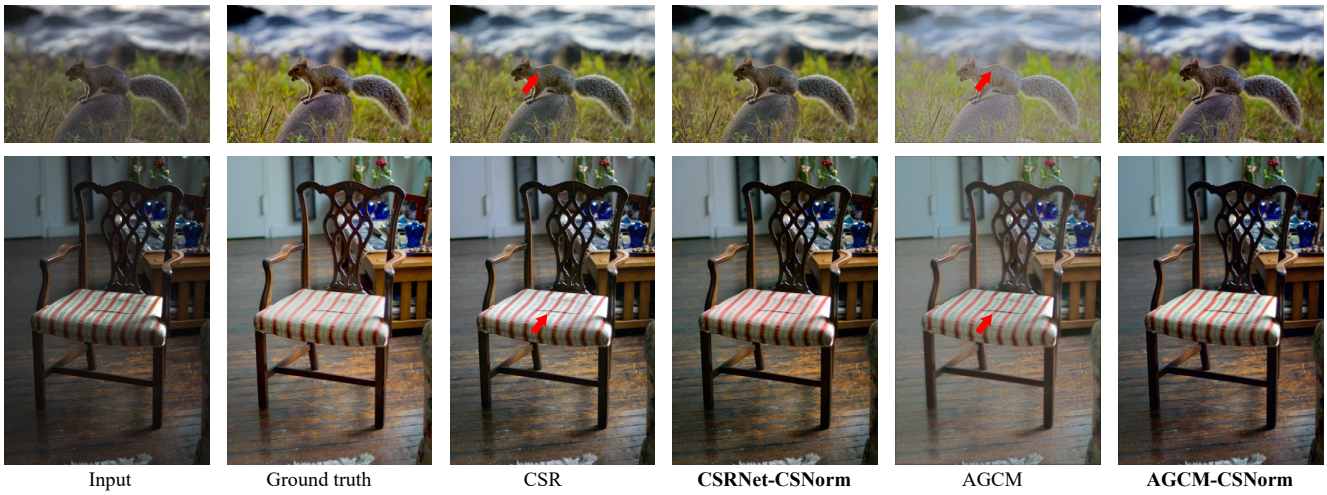


Figure 8. Visual comparisons of the generalized image retouching on the MIT-Adobe FiveK [1] dataset. The models are trained on the original dataset and tested on the scaled lightness condition.

References

[1] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR 2011*, pages 97–104. IEEE, 2011. 3, 7

[2] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3291–3300, 2018. 2, 4, 5

[3] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 17–33. Springer, 2022. 2, 6

[4] Xiangyu Chen, Zhengwen Zhang, Jimmy S Ren, Lynhoo Tian, Yu Qiao, and Chao Dong. A new journey from sdrtv to hdrtv. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4500–4509, 2021. 3, 6, 7

[5] Yu-Sheng Chen, Yu-Ching Wang, Man-Hsin Kao, and Yung-Yu Chuang. Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6306–6314, 2018. 3

[6] Jiang Hai, Zhu Xuan, Ren Yang, Yutong Hao, Fengzhu Zou, Fang Lin, and Songchen Han. R2rnet: Low-light image enhancement via real-low to real-normal network. *arXiv preprint arXiv:2106.14501*, 2021. 1, 3, 4, 5, 6

[7] Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. Exposure: A white-box photo post-processing framework. *ACM Transactions on Graphics (TOG)*, 37(2):1–17, 2018. 3

[8] Soo Ye Kim, Jihyong Oh, and Munchurl Kim. Deep sr-itm: Joint learning of super-resolution and inverse tone-mapping for 4k uhd hdr applications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3116–3125, 2019. 3, 6, 7

[9] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 1

[10] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *arXiv preprint arXiv:1808.04560*, 2018. 3, 4, 5, 6

[11] Wenhan Yang, Shiqi Wang, Yuming Fang, Yue Wang, and Jiaying Liu. From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3063–3072, 2020. 2, 6