Supplementary Material
Learning Multi-Scale Photo Exposure Correction

Mahmoud Afifi[1,2*] Konstantinos G. Derpanis[1] Björn Ommer[3] Michael S. Brown[1]

[1]Samsung AI Centre (SAIC), Toronto, Canada

[2]York University, Canada [3]Heidelberg University, Germany

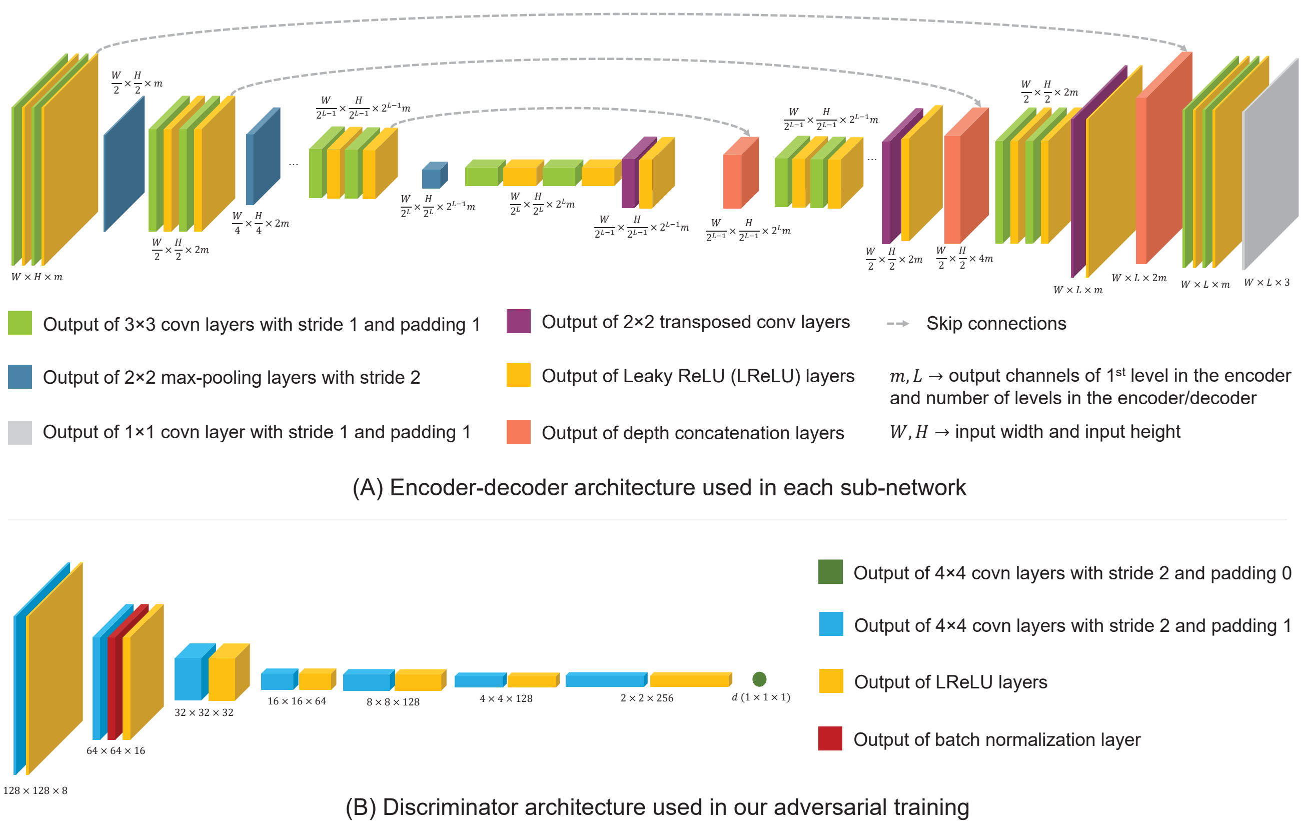


Figure 1: Details of the architectures used in our work. (A) Encoder-decoder architecture [25] used to design our sub-networks in the main network. (B) Discriminator architecture.

Our supplemental material is organized as follows: Sec. 1 provides information on our implementation, including the description of our main network, the discriminator network, and additional training details. Sec. 2 presents ablation studies that we carried out to validate our network design and loss function. Sec. 3 provides additional qualitative results. Sec. 4 concludes with a discussion on potential applications of our proposed method.

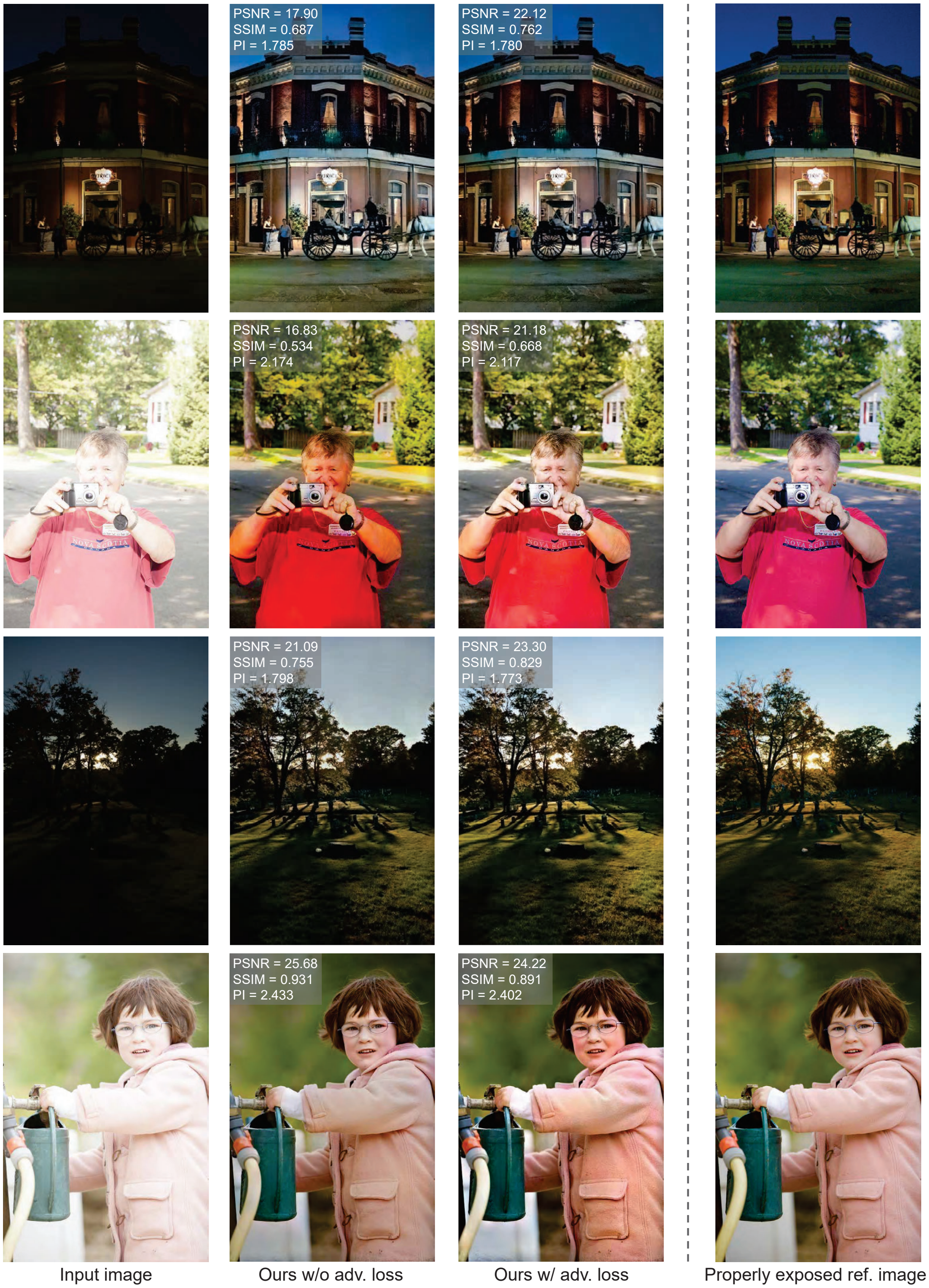


Figure 2: Comparisons between our results with (w/) and without (w/o) the adversarial loss for training. The peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) [30], and perceptual index (PI) [1] are shown for each result. Notice that higher PSNR and SSIM values are better, while lower PI values indicate better perceptual quality. The input images are taken from our test set.

1. Implementation Details

In the main paper, we proposed a coarse-to-fine network to correct exposure errors in photographs. In this section, we provide the implementation details of our network, the discriminator network used in the adversarial training process, and additional training details.

1.1. Main Network

Our main network consists of four sub-networks with ~7M parameters trained in an end-to-end manner. The largest network capacity is dedicated to the first sub-network with decreasing amounts of capacity as we move from coarse-to-fine scales. Each sub-network accepts a different representation of the input image extracted from the Laplacian pyramid decomposition. The first sub-network is a four-layer encoder-decoder network with skip connections (i.e., U-Net-like architecture [25]). The output of the first convolutional (conv) layer has 24 channels. Our first sub-network has ~4.4M learnable parameters and accepts the low-frequency band level of the Laplacian pyramid, i.e., $\mathbf{X}_{(4)}$. The result of the first sub-network is then upscaled using a $2\times2\times3$ transposed conv layer with three output channels and a stride of two. This processed layer is then added to the first mid-frequency band level of the Laplacian pyramid (i.e., $\mathbf{X}_{(3)}$) and is fed to the second sub-network.

The second sub-network is a three-layer encoder-decoder network with skip connections. It has 24 channels in the first conv layer of the encoder, with a total of ~1.1M learnable parameters. The second sub-network processes the upscaled input from the first sub-network and outputs a residual layer, which is then added back to the input to the second sub-network followed by a $2\times2\times3$ transposed conv layer with three output channels and a stride of two. The result is added to the second mid-frequency band level of the Laplacian pyramid (i.e., $\mathbf{X}_{(2)}$) and is fed to the third sub-network, which generates a new residual that is added back again to the input of this sub-network.

The third sub-network has the same design as the second network. Finally, the result is added to the high-frequency band level of the Laplacian pyramid (i.e., $\mathbf{X}_{(1)}$) and is fed to the fourth sub-network to produce the final processed image.

The final sub-network is a three-layer encoder-decoder network with skip connections and has ~482.2K learnable parameters, where the output of the first conv layer in its encoder has 16 channels. We provide the details of the main encoder-decoder architecture of each sub-network in Fig. 1-(A).

1.2. Discriminator Network

In the adversarial training of our network, we use a lightweight discriminator network with ~1M learnable parameters. We provide the details of the discriminator in Fig. 1-(B). Notice that unlike our main network, we resize all input image patches to have 256×256 pixels before being processed by the discriminator. The output of the last layer in our discriminator is a single scalar value which is then used in our loss during the optimization, as described in the main paper.

1.3. Additional Training Details

We use He et al.'s method [15] to initialize the weights of our encoder and decoder conv layers, while the bias terms are initialized to zero. We minimize our loss functions using the Adam optimizer [18] with a decay rate $\beta_1 = 0.9$ for the exponential moving averages of the gradient and a decay rate $\beta_2 = 0.999$ for the squared gradient. We use a learning rate of 10^{-4} to update the parameters of our main network and a learning rate of 10^{-5} to update our discriminator's parameters.

We train our network on patches with different dimensions. Training begins without the adversarial loss, $\mathcal{L}_{\text{adv}}$, then $\mathcal{L}_{\text{adv}}$ is added to fine-tune the results of our initial training [22]. Specifically, we begin our training without $\mathcal{L}_{\text{adv}}$ on 176,590 patches with dimensions of 128×128 pixels extracted randomly from our training images for 40 epochs. The mini-batch size is set to 32. The learning rate is decayed by a factor of 0.5 after the first 20 epochs. Then, we continue training on another 105,845 patches with dimensions of 256×256 pixels for 30 epochs (15 epochs without $\mathcal{L}_{\text{adv}}$ and 15 epochs with $\mathcal{L}_{\text{adv}}$) with a mini-batch size of eight. The learning rates for the main network and the discriminator network are decayed by a factor of 0.5 every 10 epochs. Finally, we fine-tune the trained networks on another 69,515 training patches with dimensions of 512×512 pixels for 20 epochs with a mini-batch size of four and a learning rate decay of 0.5 applied every five epochs.

We discard training patches that have an average intensity less than 0.02 or higher than 0.98. We also discard homogeneous patches that have an average gradient magnitude less than 0.06. We randomly left-right flip training patches for data augmentation.

In the adversarial training, we optimize both the main network and the discriminator in an iterative manner. At each optimization step, the learnable parameters of each network are updated to minimize its own loss function. Our main network's loss function is described in the main paper. The discriminator is trained to minimize the following loss function [10]:

$$\mathcal{L}_{\text{dsc}} = r\left(\mathbf{T}\right) + c\left(\mathbf{Y}\right), \tag{1}$$

where $r\left(\mathbf{T}\right)$ refers to the discriminator loss of recognizing the properly exposed reference image $\mathbf{T}$, while $c\left(\mathbf{Y}\right)$ refers to the discriminator loss of recognizing our corrected image $\mathbf{Y}$. The $r\left(\mathbf{T}\right)$ and $c\left(\mathbf{Y}\right)$ loss functions are given by the

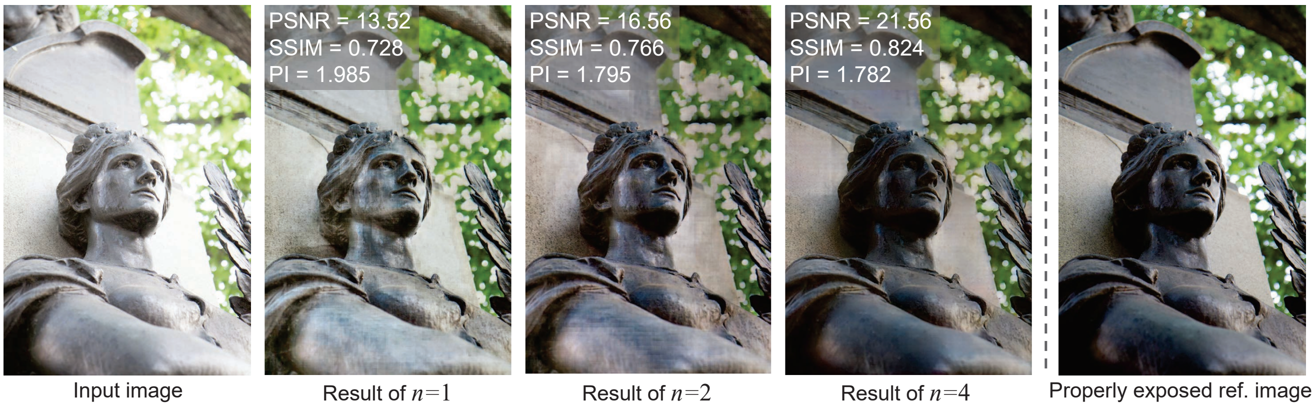


Figure 3: Comparison of results by varying the number of Laplacian pyramid levels. The peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) [30], and perceptual index (PI) [1] are shown for each result. Notice that higher PSNR and SSIM values are better, while lower PI values indicate better perceptual quality. The input image is taken from our validation set.

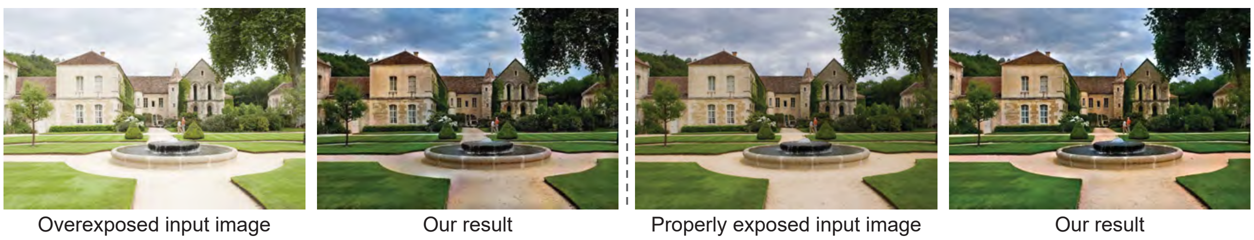


Figure 4: Our framework can deal with both improperly and properly exposed input images producing compelling results. The input images are taken from our test set.

following equations:

$$r\left(\mathbf{T}\right) = -\log\left(\mathcal{S}\left(\mathcal{D}\left(\mathbf{T}\right)\right)\right), \tag{2}$$

$$c\left(\mathbf{Y}\right) = -\log\left(1 - \mathcal{S}\left(\mathcal{D}\left(\mathbf{Y}\right)\right)\right), \tag{3}$$

where $\mathcal{S}$ denotes the sigmoid function and $\mathcal{D}$ is the discriminator network described in Fig. 1-(B).

2. Ablation Studies

This section presents details on the ablation studies that were performed to validate the architecture and loss function used in the main paper.

2.1. Loss Function

Our loss function (Eq. 1 in the main paper) includes three main terms. The first term is the standard reconstruction loss (i.e., L_1 loss). The second and third terms consist of the pyramid and adversarial losses, respectively, which are introduced to further improve the reconstruction and perceptual quality of the output images. In the following, we discuss the effect of these loss terms.

Table 1: Results of our ablation study on 500 images randomly selected from our validation set. We show the effects of: (i) the pyramid loss, $\mathcal{L}_{\text{pyr}}$, and (ii) the number of Laplacian levels, n, in the main network. For each experiment, we show the values of the peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) [30]. The best PSNR/SSIM values are indicated with bold for each experiment.

	Pyramid loss $\mathcal{L}_{\text{pyr}}$		Number of levels n		
	w/o	w/	$n=1$	$n=2$	$n=4$
PSNR	18.041	**18.385**	16.984	17.442	**18.385**
SSIM	0.746	**0.749**	0.723	0.734	**0.749**

2.1.1 Pyramid Loss Impact

In Fig. 5 of the main paper, we show the output of each sub-network when we train our model with and without the pyramid loss. We observe that the pyramid loss helps to provide additional supervision to guide each sub-network to follow a coarse-to-fine reconstruction. In this ablation study, we aim to quantitatively evaluate the effect of the pyramid loss on our final results.

We train two light-weight models of our main network with and without our pyramid loss term. Each model has

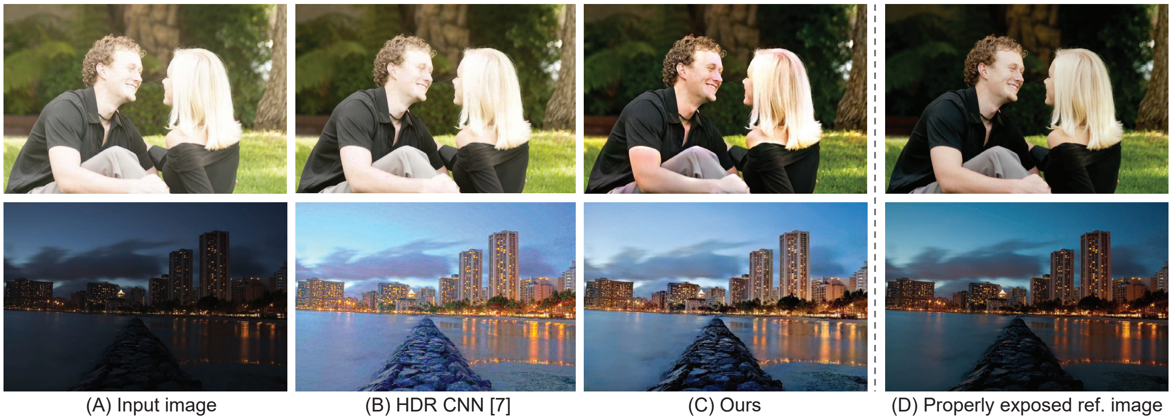


Figure 5: Additional qualitative results. (A) Input images. (B) Results of HDR CNN [7] with Adobe Photoshop's HDR tool [6]. (C) Our results. (G) Properly exposed reference images. The input images are taken from our test set.

four 3-layer U-Nets with a total of $\sim$4M learnable parameters, where the number of output channels of the first encoder in each U-Net is set to 24.

The training is performed on a sub-set of our training data for $\sim$150,000 iterations on 80,000 128×128 patches, $\sim$100,000 iterations on 40,000 256×256 patches, and $\sim$25,000 iterations on 25,000 512×512 patches. Table 1 shows the results on 500 randomly selected images from our validation set. The results show that the pyramid loss not only helps in providing a better interpretation of the task of each sub-network but also improves the final results.

2.1.2 Adversarial Loss Impact

In the main paper, we show quantitative results of our method with and without the adversarial loss term. Our trained model with the adversarial loss term achieves better perceptual quality (i.e., lower perceptual index (PI) values [1]) than training without the adversarial loss term. Note that there is a fundamental trade-off between the perceptual quality (i.e., PI) and the pixel-wise similarity (i.e., PSNR and SSIM) [2]. That is, w/o adversarial loss the pixel-wise similarity is improved rather than the perceptual quality and vice-versa.

Fig. 2 shows qualitative comparisons of our results with and without the adversarial loss. As shown, the network trained without the adversarial training tends to produce darker images with slightly unrealistic colors in some cases, while the adversarial regularization improves the perceptual quality of our results.

2.2. Number of Laplacian Pyramid Levels

We repeat the same experimental setup described in Sec. 2.1.1 with a varying number of Laplacian pyramid levels (sub-networks). Specifically, we train a network with $n = 1$ levels—this network is equivalent to a vanilla U-Net-like architecture [25]. Additionally, we train another network with $n = 2$ (i.e., two sub-networks).

For a fair comparison, we fix the total number of parameters in each model by changing the number of filters in the conv layers. Specifically, we set the number of output channels of the first layer in the encoder to 48 for the trained model with $n = 1$, while we decrease it to 34 for the two-sub-net model (i.e., $n = 2$) to have approximately the same number of learnable parameters. Thus, the trained model in Sec. 2.1.1, used to study the pyramid loss impact, and the additional two trained models have approximately the same number of parameters.

Table 1 shows the results obtained by each model on the same random validation image subset used to study the pyramid loss impact in Sec. 2.1.1. Fig. 3 shows a qualitative comparison. As can be seen, the best quantitative and qualitative results are obtained using the four-sub-net model (i.e., $n = 4$ levels).

3. Additional Results and Comparisons

In this section, we provide additional qualitative results. Fig. 4 shows our results when the input image has no exposure errors. As can be seen, our method produces consistent output images regardless of the exposure setting of the input image. Additional qualitative comparisons with other methods on our testing set are shown in Fig. 5–9.

Generalization We provide additional results on images that are outside our training/testing sets. Fig. 10 shows qualitative comparisons with the methods of Yuan and Sun [27] and Guo et al. [12], which were designed to correct overexposure errors in photographs. The source code of these methods is not available. Thus, the presented input images and corresponding results by the methods of Yuan and Sun [27] and Guo et al. [12] are taken from the original papers [12, 27]. As shown in Fig. 10, our method produces

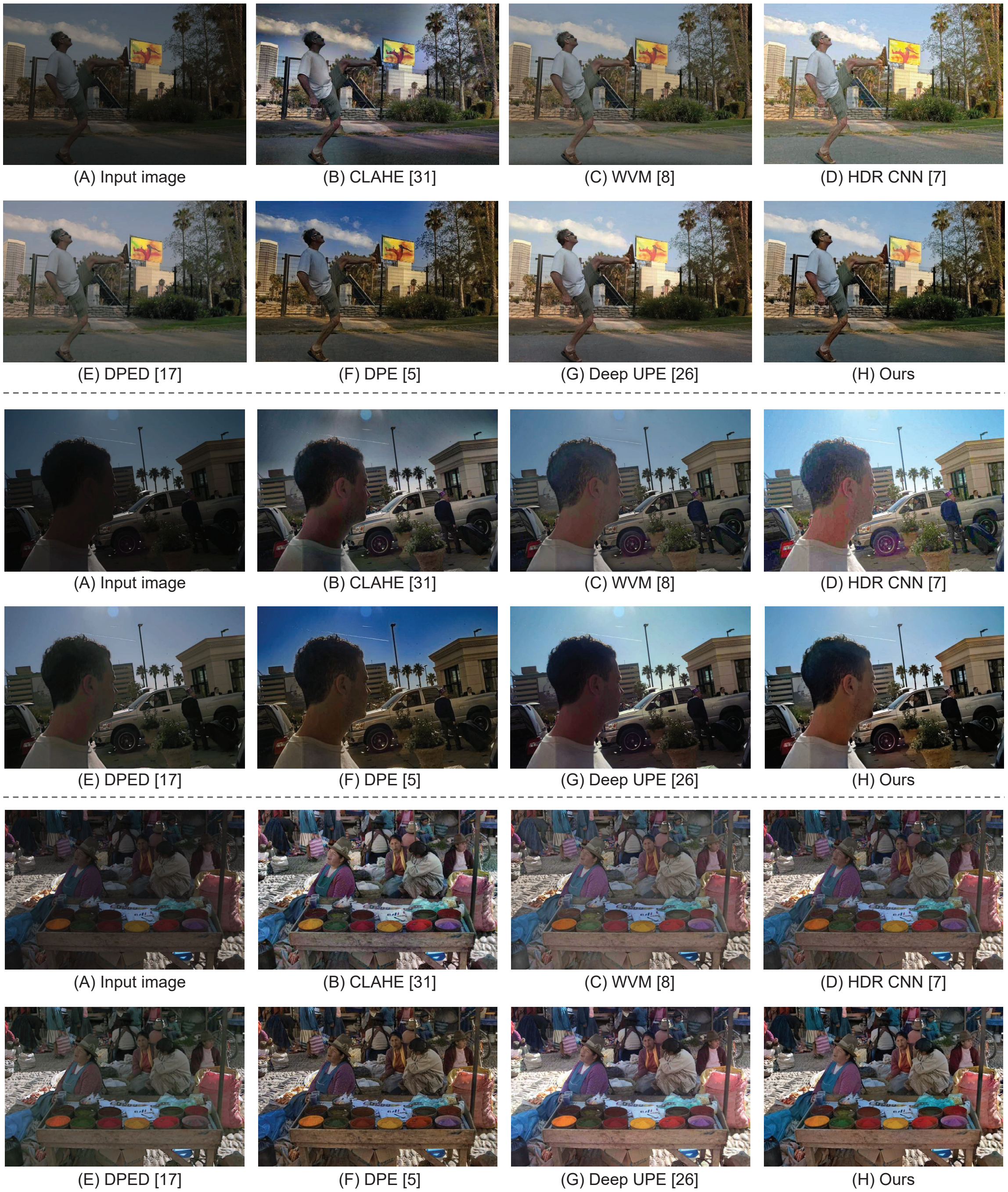


Figure 6: Additional qualitative comparisons with other methods in correcting underexposed images. (A) Input images. (B) Results of CLAHE [31]. (C) Results of WVM [8]. (D) Results of HDR CNN [7] with Adobe Photoshop's HDR tool [6]. (E) Results of DPED [17]. (F) Results of DPE [5]. (G) Results of Deep UPE [26]. (H) Our results. The input images are taken from our test set.

compelling results.

Fig. 11 shows a qualitative comparison using the DICM image set. Fig. 12 shows a qualitative comparison using the SID dataset [4]. In the shown example, we rendered the raw-RGB images provided in the SID dataset to 8-bit JPEG compressed sRGB image. This 8-bit compressed format is more challenging compared to dealing with the 12-bit linear raw images as used by prior work. Though our method

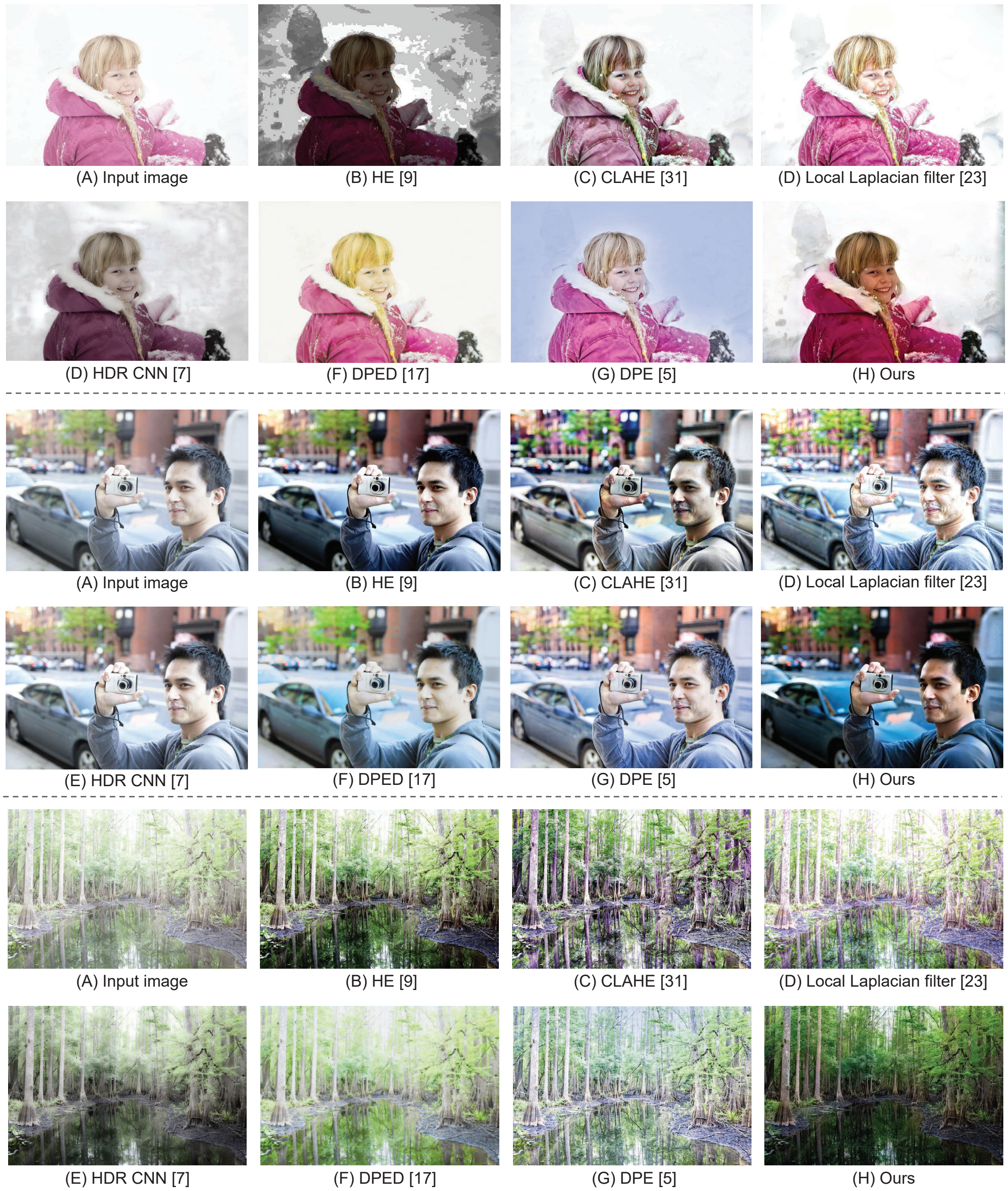


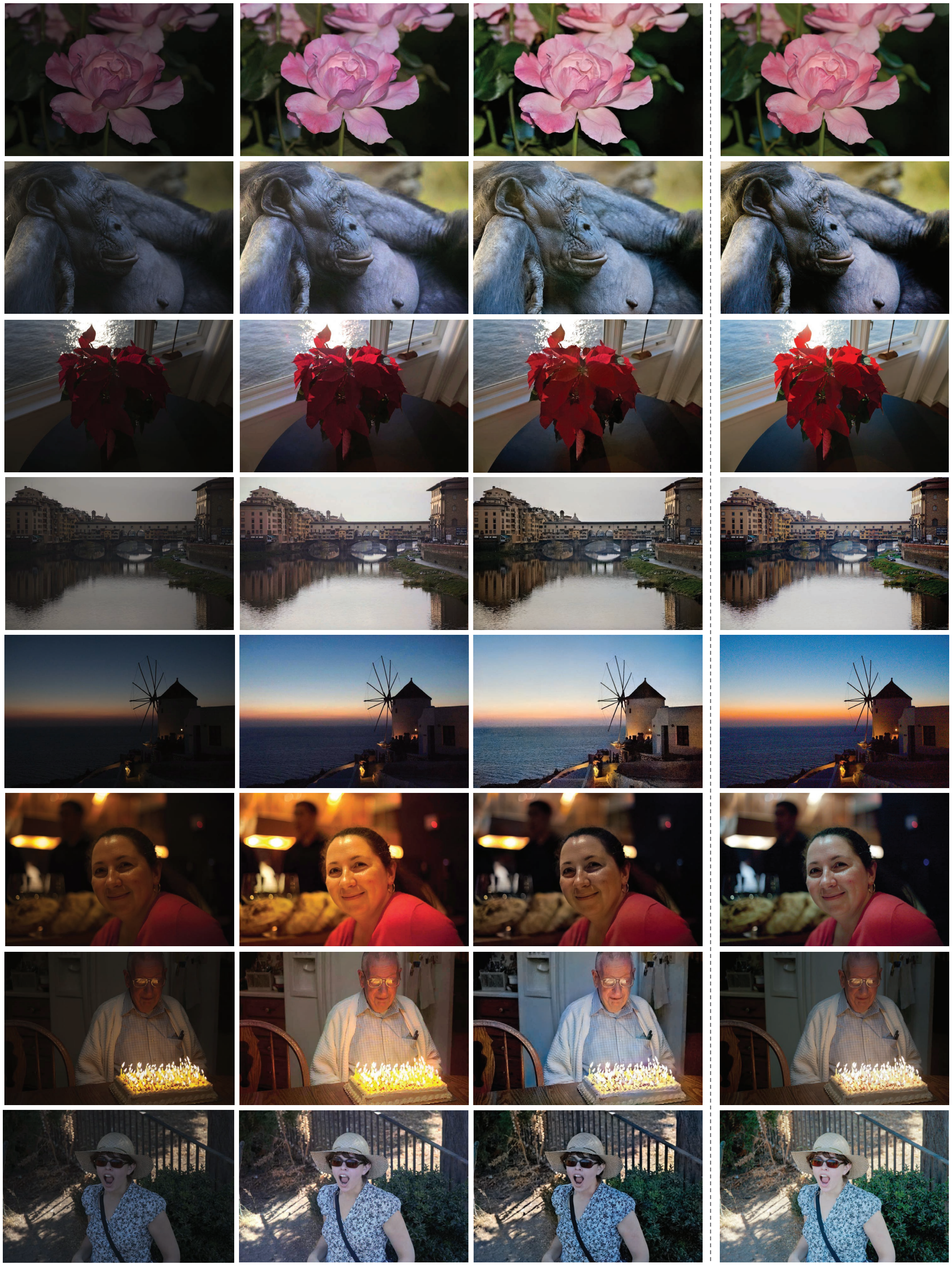
Figure 7: Additional qualitative comparisons with other methods in correcting overexposed images. (A) Input images. (B) Results of histogram equalization (HE) [9]. (C) Results of the contrast-limited adaptive histogram equalization (CLAHE) [31]. (D) Results of the local Laplacian filter [23]. (E) Results of HDR CNN [7] with Adobe Photoshop's (PS) HDR tool [6]. (F) Results of the DSLR Photo Enhancement dataset (DPED) trained model [17]. (G) Results of deep photo enhancer (DPE) [5]. (H) Our results. The input images are taken from our test set.

is not targeting this kind of "dark" scenes, it is arguable that our result is visually on par with the recently proposed method for low-light image enhancement—namely, the Zero-DCE method [11].

We further examined our model on the testing set used in [26]. This set has no overlap with our training ex-

(A) Input image (B) DPED [17] (C) Ours (D) Properly exposed ref. image

Figure 8: Additional qualitative results of correcting overexposed images. (A) Input images. (B) Results of DPED [17]. (C) Our results. (G) Properly exposed reference images. The input images are taken from our test set.



(A) Input image (B) Deep UPE [26] (C) Ours (D) Properly exposed ref. image

Figure 9: Additional qualitative results of correcting underexposed images. (A) Input images. (B) Results of Deep UPE [26]. (C) Our results. (G) Properly exposed reference images. The input images are taken from our test set.

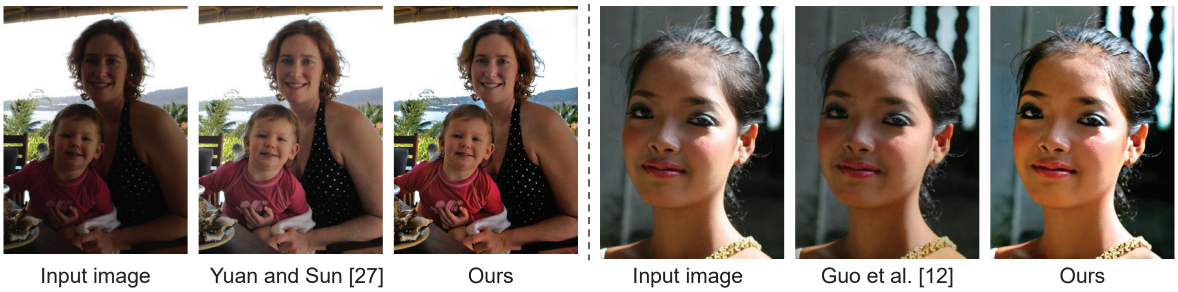


Figure 10: Qualitative comparison with the methods of Yuan and Sun [27] and Guo et al. [12]. The input images are taken from [27] and [12], respectively.

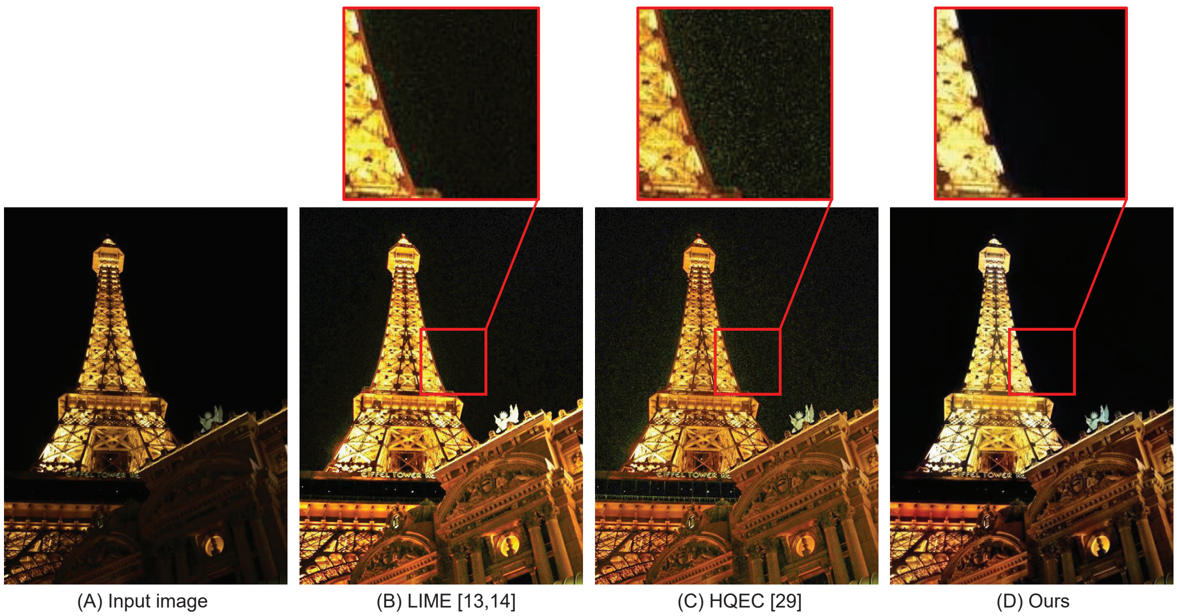


Figure 11: Additional qualitative results of correcting overexposed images. (A) Input image. (B) Result of LIME [13, 14]. (C) Result of HQEC [29]. (D) Our result. The input image is taken from the DICM image set [19].

amples taken from the MIT-Adobe FiveK dataset [3] and its input images were processed using a different rendering/degradation procedure, as described in [26]. Fig. 13 shows a qualitative comparison between our method and the recent Zero-DCE method [11] for low-light image enhancement. The quantitative results using the testing set used in [26], are reported in Table 2.

As can be seen, our method achieves on par, sometimes better, results compared to the state-of-the-art methods designed specifically to deal with underexposure errors. Unlike these methods, our method can effectively deal with both under- and overexposure errors, as discussed in the main paper. Note that we did not fine-tune our method on either the SID dataset or the set used in [26], before reporting our results. Additional qualitative comparisons using images taken from Flickr are shown in Fig. 14.

Table 2: Comparison with other methods for low-light image enhancement using the test set used in [26].

Method	PSNR
White-Box [16]	18.57
Distort-and-Recover [24]	20.97
Deep UPE [26]	**23.04**
Zero-DCE [11]	15.455
Ours	21.02

4. Potential Applications

In this section, we highlight two potential applications of our method: (i) photo editing and (ii) image preprocessing.

Photo Editing The main potential application of the proposed method is to post-capture correct exposure errors in images. This correction process can be performed in a fully

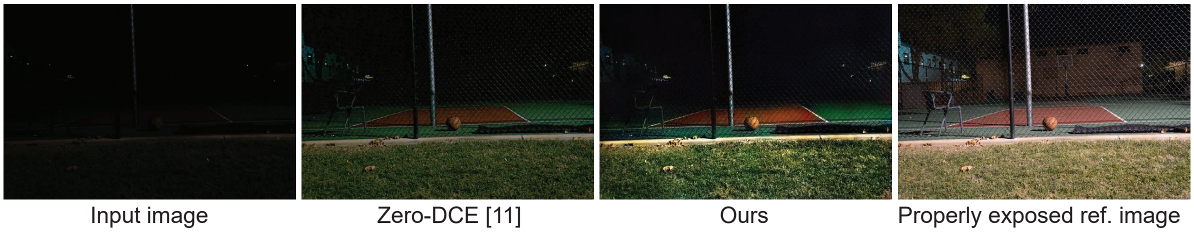


Figure 12: Qualitative example from the SID dataset [4]. We compare our result with the recent Zero-DCE method [11].

Figure 13: Qualitative comparison with the recent Zero-DCE method [11] on the testing set, used in [26].

automated way (as described in the main paper) or can be performed in an interactive way with the user. Specifically, we introduce a scale vector $\mathbf{S} = [S_1, S_2, S_3, S_4]^\top$ that can be used to independently scale each level in the pyramid $\mathbf{X}$ in the inference stage. The scale vector $\mathbf{S}$ is introduced to produce different visual effects in the final result $\mathbf{Y}$. In particular, this scaling operation is performed as a pre-processing of each level in the pyramid $\mathbf{X}$ as follows: $\mathbf{S}_{(l=i)}\mathbf{X}_{(l=i)}$, s.t. $i \in \{1, 2, 3, 4\}$. The values of the scale vector $\mathbf{S}$ can be interactively controlled by the user to edit our network results. Fig. 15 shows different results obtained by our network in an interactive way through our graphical user interface (GUI). Our GUI can be used as a photo editing tool to apply different visual effects and filters on the input images. Note that we used $\mathbf{S} = [1.8, 1.8, 1.8, 1.12]^\top$ in our experiments in the main paper, as we found it gives the best compelling results (see Fig. 16).

Image Preprocessing Our method can also improve the results of computer vision tasks by using it as a pre-processing step to correct exposure errors in input images. Fig. 17 shows example applications. In these examples, we show results of face and facial landmark detection of the work in [28] and image semantic segmentation results obtained by the work in [20, 21]. As shown, the results of face detection and semantic segmentation are improved by pre-processing the input images using our method.

References

[1] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 PIRM challenge on perceptual image super-resolution. In *ECCV Workshops*, 2018. 2, 4, 5

[2] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *CVPR*, 2018. 5

[3] 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. 10

[4] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *CVPR*, 2018. 6, 11

[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 *CVPR*, 2018. 6, 7

[6] Lisa DaNae Dayley and Brad Dayley. *Photoshop CS5 Bible*. John Wiley & Sons, 2010. 5, 6, 7

[7] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafa Mantiuk, and Jonas Unger. HDR image reconstruction from a single exposure using deep CNNs. *ACM Transactions on Graphics (TOG)*, 36(6):178:1–178:15, 2017. 5, 6, 7

[8] Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping Zhang, and Xinghao Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In *CVPR*, 2016. 6

[9] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., 2001. 7

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 3

[11] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong. Zero-reference deep curve estimation for low-light image enhancement. In *CVPR*, 2020. 7, 10, 11, 12

[12] Dong Guo, Yuan Cheng, Shaojie Zhuo, and Terence Sim. Correcting over-exposure in photographs. In *CVPR*, 2010. 5, 10

[13] Xiaojie Guo. LIME: A method for low-light image enhancement. In *ACM MM*, 2016. 10

Input image | Zero-DCE [11] | Ours

Figure 14: Comparison with the recent Zero-DCE method [11] using images taken from Flickr.

(A) Input image | (B) Our results using different settings

Figure 15: Our GUI photo editing tool. (A) Input image. (B) Our results using different pyramid level scaling settings set by the user in an interactive way. The input image is taken from Flickr.

[14] Xiaojie Guo, Yu Li, and Haibin Ling. LIME: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2):982–993, 2017. 10

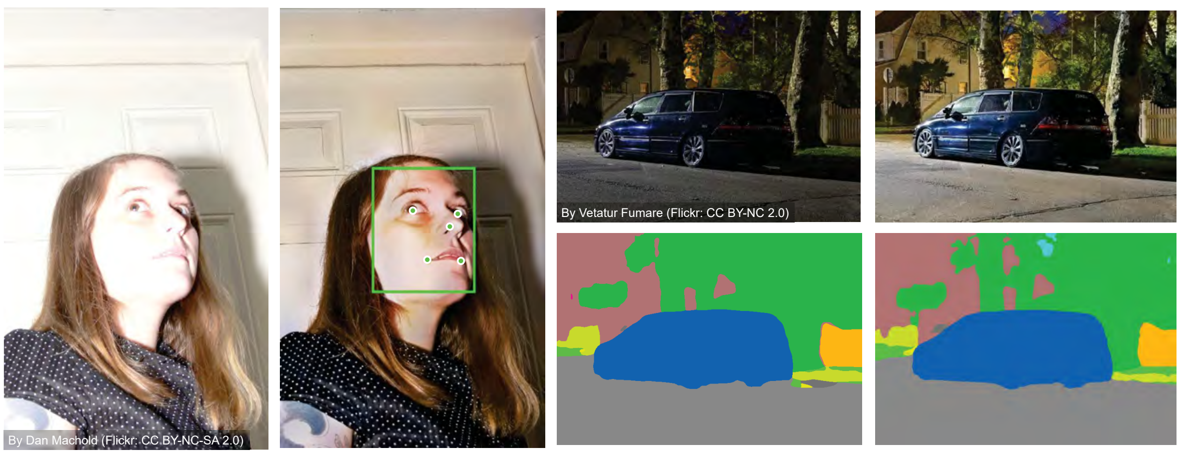
[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *ICCV*, 2015. 3

[16] 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):26:1–26:17, 2018. 10

[17] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. DSLR-quality photos on mobile devices with deep convolutional networks. In *ICCV*, 2017. 6, 7, 8

(A) Input image (B) $S = [1.0, 1.0, 1.0, 1.0]^{\mathrm{T}}$ (C) $S = [1.0, 1.0, 1.0, 1.3]^{\mathrm{T}}$ (D) $S = [1.8, 1.8, 1.8, 1.1]^{\mathrm{T}}$

Figure 16: The effect of the scale vector $\mathbf{S}$ on our final results. (A) Input images. (B-D) Our results using different scale values, $\mathbf{S}$. The shown input images are taken from our validation set.



(A) Failure case of face and facial landmark detection (B) Face and facial landmark detection after our correction (C) Semantic segmentation result on original image (D) Semantic segmentation result on our corrected image

Figure 17: Applying our method as a pre-processing step can improve results of different computer vision tasks. (A) False negative result of face and facial landmark detection due to the overexposure error in the input image. (B) Our corrected image and the results of face and facial landmark detection. (C) Underexposed input image and its semantic segmentation mask. (D) Our corrected image and its semantic segmentation mask. We use the cascaded convolutional network proposed in [28] for face and facial landmark detection. For image semantic segmentation, we use RefineNet [20, 21]. The input images are taken from Flickr.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 3

[19] Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation. In *ICIP*, 2012. 10

[20] Guosheng Lin, Fayao Liu, Anton Milan, Chunhua Shen, and Ian Reid. RefineNet: Multi-path refinement networks for dense prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(5):1228–1242, 2020. 11, 13

[21] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 11, 13

[22] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NeurIPS*, 2017. 3

[23] Sylvain Paris, Samuel W Hasinoff, and Jan Kautz. Local Laplacian filters: Edge-aware image processing with a Laplacian pyramid. *Communications of the ACM*, 58(3):81–91, 2015. 7

[24] Jongchan Park, Joon-Young Lee, Donggeun Yoo, and In So Kweon. Distort-and-recover: Color enhancement using deep reinforcement learning. In *CVPR*, 2018. 10

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1, 3, 5

[26] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhance-

ment using deep illumination estimation. In *CVPR*, 2019. 6, 7, 9, 10, 11

[27] Lu Yuan and Jian Sun. Automatic exposure correction of consumer photographs. In *ECCV*, 2012. 5, 10

[28] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 11, 13

[29] Qing Zhang, Ganzhao Yuan, Chunxia Xiao, Lei Zhu, and Wei-Shi Zheng. High-quality exposure correction of underexposed photos. In *ACM MM*, 2018. 10

[30] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 2, 4

[31] Karel Zuiderveld. Contrast limited adaptive histogram equalization. In *Graphics Gems IV*, pages 474–485, 1994. 6, 7