Supplemental File to "4K-Resolution Photo Exposure Correction at 125 FPS with ∼8K Parameters"

Yijie Zhou[1] Chao Li[1] Jin Liang[1] Tianyi Xu[1] Xin Liu[2,3,] Jun Xu[1,4,*]
[1]Nankai University [2]Tianjin University [3]Lappeenranta-Lahti University of Technology
[4]Guangdong Provincial Key Laboratory of Big Data Computing, CUHK (Shenzhen)

1. Content

In this supplemental file, we provide more details of our Multi-Scale Linear Transformation (MSLT) networks presented in the main paper. Specifically, we provide

- the detailed implementation of Laplacian Pyramid (LP) decomposition and reconstruction in § 2.
- the channel dimension of the features in our SFE module in § 3.
- the details of coefficient transformation in our bilateral grid network in § 4.
- more details of high-frequency layers correction in § 5.
- the architecture of the Channel-MLP network in our main paper in § 6.
- more ablation studies in § 7.
- more visual comparisons of our MSLTs with the other comparison methods on the ME [1] and SICE datasets [3] in § 8.
- the visual comparisons in ablation studies in § 9.
- the societal impact in § 10.

2. Detailed implementation of Laplacian Pyramid (LP) decomposition and reconstruction

In our MSLT, we deploy the conventional Gaussian kernel for Laplacian Pyramid (LP) [2, 6, 9, 10] decomposition and reconstruction. In decomposition, we first use a fixed 5×5 Gaussian kernel (Eqn. 1) to perform convolution on the input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ with stride = 2, padding = 2 to obtain $\mathbf{G}_1$. Then, we perform the same convolution operation on $\mathbf{G}_i$ ($i = 1, ..., n-1$, note that $n = 4$ in our MSLTs) to generate $\mathbf{G}_{i+1}$. After getting Gaussian pyramid sequence $\{\mathbf{G}_i \in \mathbb{R}^{\frac{H}{2^{i-1}} \times \frac{W}{2^{i-1}} \times 3} | i = 1, ..., n\}$, we upsample the Gaussian pyramid $\mathbf{G}_{i+1}$ ($i = 1, ..., n-1$) by inserting comfortable all-zero vectors between every two rows and between every two columns, which is convolved with the Gaussian kernel (Eqn. 1) and then subtracted from $\mathbf{G}_i$ to obtain the high-frequency layer $\mathbf{H_i}$ of Laplacian pyramids. For $i = n$, we directly treat $\mathbf{G}_n$ as the low-frequency layer $\mathbf{L_n}$. In this way, we obtain the Laplacian pyramids of $\{\mathbf{H}_i | i = 1, ..., n-1\}$ and $\mathbf{L_n}$. In reconstruction, for each layer in the processed Laplacian pyramids, we use the same upsample method used in the decomposition and then add the results to the higher layer. Finally, we obtain the reconstructed image $\mathbf{O} \in \mathbb{R}^{H \times W \times 3}$.

$$\text{Gaussian kernel} = \frac{1}{256} \times \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \quad (1)$$

In our MSLT+ and MSLT++, we introduce learnable 3×3 convolutions with stride = 2 for downsampling in the Laplacian pyramid decomposition, and 3×3 convolutions with stride = 1 followed by bi-linear interpolation for upsampling in the Laplacian pyramid reconstruction.

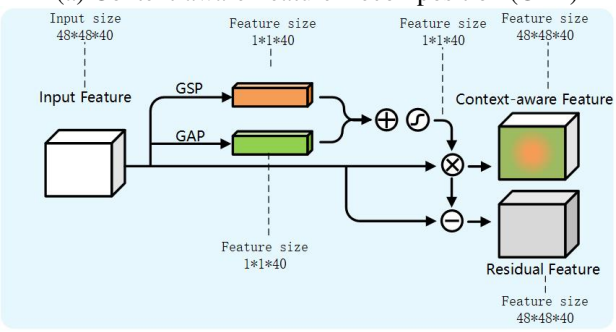
3. Channel dimension of the features in our SFE module

For our Self-modulated Feature Extraction (SFE) module, as shown in Figure 1 (b), we describe the specific numbers of input channels and output channels for the SFE module, which is used in both predicting the guidance map $\mathbf{G}$ in our bilateral grid network and feature extraction in our Hierarchical Feature Decomposition (HFD) module, as shown in Figures 2 and 3 in the main paper. For the guidance map prediction, numbers of channel $\mathbf{C}_1$ and $\mathbf{C}_2$ are 3 and 8, respectively. In order to generate a gray-scale guidance map, we additionally take a 1×1 convolution from 8 channels

*Corresponding author: csjunxu@nankai.edu.cn.

to 1 channel at the end of SFE. For the feature extraction in our HFD, both $\mathbf{C}_1$ and $\mathbf{C}_2$ are equal to 40.

(a) Context-aware Feature Decomposition (CFD)



(b) Self-modulated Feature Extraction (SFE)

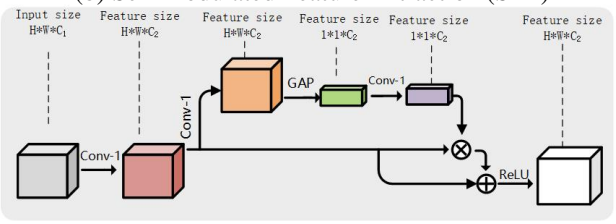


Figure 1. **The detailed structure of our CFD module (a) and our SFE module (b)**. In our MSLTs, the CFD module receives a fixed feature input size of $48 \times 48 \times 40$ in our HFD. But the input and output of SFE module in predicting the guidance map is different from that in HFD module. See § 3 for details.

4. Details of coefficient transformation in our bilateral grid network

Here, we elaborate on the coefficient transformation in the bilateral grid network our MSLT. We use the 3D bilateral grid of affine transformation coefficients $\mathcal{B} \in \mathbb{R}^{16\times16\times72}$ and the guidance map $\mathbf{G} \in \mathbb{R}^{\frac{H}{2^{n-1}}\times\frac{W}{2^{n-1}}}$ for slicing [4]. We compute a 2D grid of coefficients $\mathbf{B} \in \mathbb{R}^{\frac{H}{2^{n-1}}\times\frac{W}{2^{n-1}}}$ using $\mathcal{B}$ and pixel locations from grid $\mathbf{G}$ by tri-linear interpolation [4]:

$$\mathbf{B}[x,y] = \sum_{i,j,k} \tau(g_h x - i)\tau(g_w y - j)\tau(d\cdot\mathbf{G}[x,y] - k)\mathcal{B}[i,j,k], \tag{2}$$

where $\tau(\cdot) = max(1 - |\cdot|, 0)$ is the linear interpolation kernel, g_h and g_w are the spacial shape of grid $\mathcal{B}$. We fix both g_h and g_w to 16 and the depth of $\mathcal{B}$ to d = 6. Each cell of grid $\mathbf{B}$ contains 12 channels. For each pixel of the low-frequency layer $\mathbf{L}_n$, we multiply the three RGB values with the corresponding values of the 1st to the 3rd channels of the corresponding pixel in grid $\mathbf{B}$ and add them together, plus the fourth channel value as a bias to get corrected R channel value of the pixel. Similarly, the G and B channels of this pixel are corrected. More details about the bilateral grid learning based transformation scheme can be found in [4].

5. More details of high-frequency layers correction

For the processing of the high-frequency layers, we deploy a small MLP consisted of two 1×1 convolutional layers with a LeakyReLU [12] between them. For high-frequency layer $\mathbf{H}_{n-1}$, when predicting the mask $\mathbf{M}_{n-1}$, the input is a 9-channel feature map concatenated by $\mathbf{H}_{n-1}$, the upsampled low-frequency layer $\mathbf{L}_n$ and the upsampled corrected low-frequency layer $\overline{\mathbf{L}}_n$ along the channel dimension. So we set the channel numbers of the input and output to the first 1×1 convolutional layer as both 9. We set the channel numbers of the input and output to the second 1×1 convolutional layer as 9 and 3, respectively. For each other high-frequency layer $\mathbf{H}_i$ $(i = n-2, ..., 1)$, we set the channel numbers of the input and output to both 1×1 convolutional layers as 3 to predict the mask $\mathbf{M}_i$.

Additionally, in our MSLT++ network, we directly use the high-frequency layer $\mathbf{H}_1$ for Laplacian pyramid reconstruction rather than that processed by the high-frequency layer correction to accelerate the inference speed. The specific structure of MSLT++ is shown in Figure 3.

6. Architecture of the Channel-MLP network in our main paper

To reduce the parameter amount and computational costs, we employ channel-wise MLP widely in our MSLTs. As a comparison to MLPs, we design a plain Channel-MLP network with 7,683 parameters to perform exposure correction in the Tables 1-3 and Figure 6 of the main paper. As shown in Figure 2, the plain Channel-MLP network contains four sequential 1×1 convolutional layers, each of which followed by a ReLU activation layer.

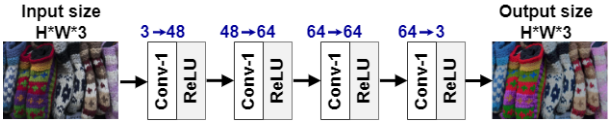


Figure 2. **Architecture of the comparison Channel-MLP**. The numbers on the "Conv-1" box represent the number of input and output channels of the convolution, respectively.

7. More Ablation Studies

In this section, we provide more experimental results to study: 1) how parameter sharing in high-frequency layers correction influences the performance of our MSLT? 2) how the GAP and GSP influence our CFD module? 3) how to design the use of SFE modules in our HFD module? 4) the effect of feature separation order in our CFD module.

1) How parameter sharing in high-frequency layers correction influences the performance of our MSLT? In high-frequency layers correction, we deploy small

MLPs consisted of two 1×1 convolutional layers with a LeakyReLU [12] between them to predict Mask $\{\mathbf{M}_i | i = 1, ..., n-1\}$. As described in § 5, the 1×1 convolutions used to predict Mask $\{\mathbf{M}_i | i = 1, ..., n-2\}$ has 3 input and output channels. Therefore, we design a comparison experiment of whether small MLPs used in different high-frequency layers correction share parameters. As shown in Table 1, whether the small MLPs in high-frequency layers correction share parameters has little effect on the performance of our MSLT. For a lower number of parameters, we choose sharing parameters in our MSLT.

Table 1. **Results of the high-frequency layers correction of our MSLT with the parameters of 1×1 convolutions shared or not.** "not shared" means we deploy independent convolutions between each high-frequency layer. "shared" means small MLPs in different high-frequency layers share convolution parameters.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	# Param.	FLOPs (M)	Speed (ms)
not shared	20.87	0.832	0.1670	7,618	83.45	4.24
shared	21.02	0.835	0.1644	7,594	83.45	4.34

2) How GAP and GSP influences our CFD module? The mean and standard deviation (std) of each channel are used in our CFD module to estimate the 3D bilateral grid of affine transformation coefficients for exposure correction. To demonstrate their combined effect, we replace the addition of GAP and GSP (denoted as "GAP + GSP") in our CFD module with single GAP (denoted as "GAP") or singel GSP (denoted as "GSP") in our CFD module. As shown in Table 2, with similar inference speed, "GAP + GSP" achieves best numerical results, while single GAP performs better than singe GSP. This illustrates that adding the mean and std of each channel in our CFD module is indeed useful. Besides, the mean plays a principal role.

Table 2. **Results of only using GAP or GSP in our CFD module.** "GAP" (or "GSP") means we use only "GAP" (or "GSP") in our CFD module. "GAP + GSP" means we use the method of adding the "GAP" and "GSP" in our CFD module.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	# Param.	FLOPs (M)	Speed (ms)
GAP	20.71	0.829	0.1688	7,594	83.73	4.32
GSP	20.47	0.826	0.1670	7,594	83.17	4.33
GAP+GSP	21.02	0.835	0.1644	7,594	83.45	4.34

3) How to design and use SFE module in HFD?. To study this question, we remove SFE modules in HFD or keep only one convolution and ReLU in SFE, denoted as "w/o SFEs" and "w/ Conv-1", respectively. As shown in Table 3, although removing the SFE module or part of it can reduce parameters and computational costs, the PSNR, SSIM [15] and LPIPS [16] are not as good as keeping our SFE module.
4) Effect of feature decomposition order in CFD. Our CFD module decompose the context-aware feature and the residual feature by feature subtraction. Here, we contrast the cases either the context-aware feature or the residual feature used as inputs to the next SFE, respectively. As shown in Table 4, our model performs comparably when the SFE module is fed with the context-aware feature or the residual feature. We conclude that the feature decomposition order in CFD module does not affect the performance of the HFD module.

Table 3. **Results of how the SFE modules are present in the HFD module.** "w/o SFEs" ("w/ SFEs") means whether we remove the SFE modules in the HFD. "w/ Conv-1" means we replace SFE in HFD module with a simple 1×1 convolutional layer and a ReLU layer.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	# Param.	FLOPs (M)	Speed (ms)
w/o SFEs	20.18	0.823	0.1845	2,672	60.77	3.85
w/ Conv-1	20.64	0.830	0.1740	4,321	72.11	3.88
w/ SFEs	21.02	0.835	0.1644	7,594	83.45	4.34

Table 4. **Results of whether the Context-aware feature output by CFDs is input to SFE or Residual feature is input to SFE in HFD module**. "Context-aware feature" means we feed the context-aware feature into SFE module and "Residual feature" means we feed the residual feature feature into SFE module in our CFD module.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	# Param.	FLOPs (M)	Speed (ms)
Context-aware feature	20.81	0.827	0.1694	7,594	83.45	4.35
Residual feature	21.02	0.835	0.1644	7,594	83.45	4.34

8. More visual comparisons of our MSLTs with the other comparison methods

Here, we present more visual comparison results with other competing methods on the ME dataset [1] and the SICE [3] dataset here. For the ME dataset, we present two sets of comparison images for each of the five relative exposure values of $\{-1.5, -1, 0, +1, +1.5\}$ in Figures 5-9. For the SICE dataset, we present three sets of comparison images each for under and over exposed inputs in Figures 10 and 11. All these results demonstrate that our MSLT networks (MSLT, MSLT+, and MSLT++) achieve comparable or even better visual quality on the exposure corrected images than the competing methods with larger parameter amount and computational costs.

9. Visual comparisons in ablation studies

In this section, we will provide visual comparisons of ablation studies in our paper and this supplementary file. Figures 12-16 represent the 1st-5th ablation study in our paper and Figures 17-20 represent the 1st-4th ablation study in this supplementary file, respectively. For simplicity, we randomly select one image from the two datasets for comparison in each ablation study.

Specifically, Figure 12 shows the visual results of our MSLT with different number of Laplacian pyramid levels on one over-exposure image. Figures 13 and 14 show the visual results of our MSLT with different variants of CFD module and different number of CFD modules in HFD. Figure 15 shows visual results of our MSLT with different variants of HFD module in the developed Bilateral Grid Network. Figure 16 shows the visual results of our MSLT and

MSLT+ with some high-frequency layers in Laplacian pyramid unprocessed by MSLT/MSLT+. Figure 17 shows visual results of our MSLT with the parameters of 1×1 convolutions shared or not. Figure 18 shows visual results of our MSLT which handles whether or not GAP and GSP are used in CFD module. Figure 19 shows visual results of our MSLT which handles SFE modules differently. Figure 20 shows visual results of our MSLT with with different inputs to our SFE module.

As we can see, our MSLT/MSLT+ can better restore the brightness and color of the images than the other methods in all these ablation studies.

10. Societal Impact

This work has the potential to be applied to enhance the user experience of taking photos in real-time, and enjoys much positive societal impact.

References

[1] Mahmoud Afifi, Konstantinos G Derpanis, Bjorn Ommer, and Michael S Brown. Learning multi-scale photo exposure correction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9157–9167, 2021. 1, 3, 6, 7, 8, 9, 10

[2] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in Computer Vision*, pages 671–679. Elsevier, 1987. 1

[3] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Transactions on Image Processing*, 27(4):2049–2062, 2018. 1, 3, 9, 10

[4] Jiawen Chen, Andrew Adams, Neal Wadhwa, and Samuel W Hasinoff. Bilateral guided upsampling. *ACM Transactions on Graphics*, 35(6):1–8, 2016. 2

[5] Ziteng Cui, Kunchang Li, Lin Gu, Shenghan Su, Peng Gao, Zhengkai Jiang, Yu Qiao, and Tatsuya Harada. You only need 90k parameters to adapt light: A light weight transformer for image enhancement and exposure correction. In *British Machine Vision Conference*, 2022. 6, 7, 8, 9, 10

[6] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. *Advances in Neural Information Processing Systems*, 28, 2015. 1

[7] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1780–1789, 2020. 6, 7, 8, 9, 10

[8] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 11

[9] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 624–632, 2017. 1

[10] Jie Liang, Hui Zeng, and Lei Zhang. High-resolution photo-realistic image translation in real-time: A laplacian pyramid translation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9392–9400, 2021. 1, 6, 7, 8, 9, 10

[11] Long Ma, Tengyu Ma, Risheng Liu, Xin Fan, and Zhongxuan Luo. Toward fast, flexible, and robust low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5637–5646, 2022. 6, 7, 8, 9, 10

[12] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning*, volume 30, page 3. Atlanta, Georgia, USA, 2013. 2, 3

[13] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 11

[14] Haoyuan Wang, Ke Xu, and Rynson WH Lau. Local color distributions prior for image enhancement. In *Eur. Conf. Comput. Vis.*, pages 343–359, 2022. 6, 7, 8, 9, 10

[15] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. 3

[16] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 586–595, 2018. 3

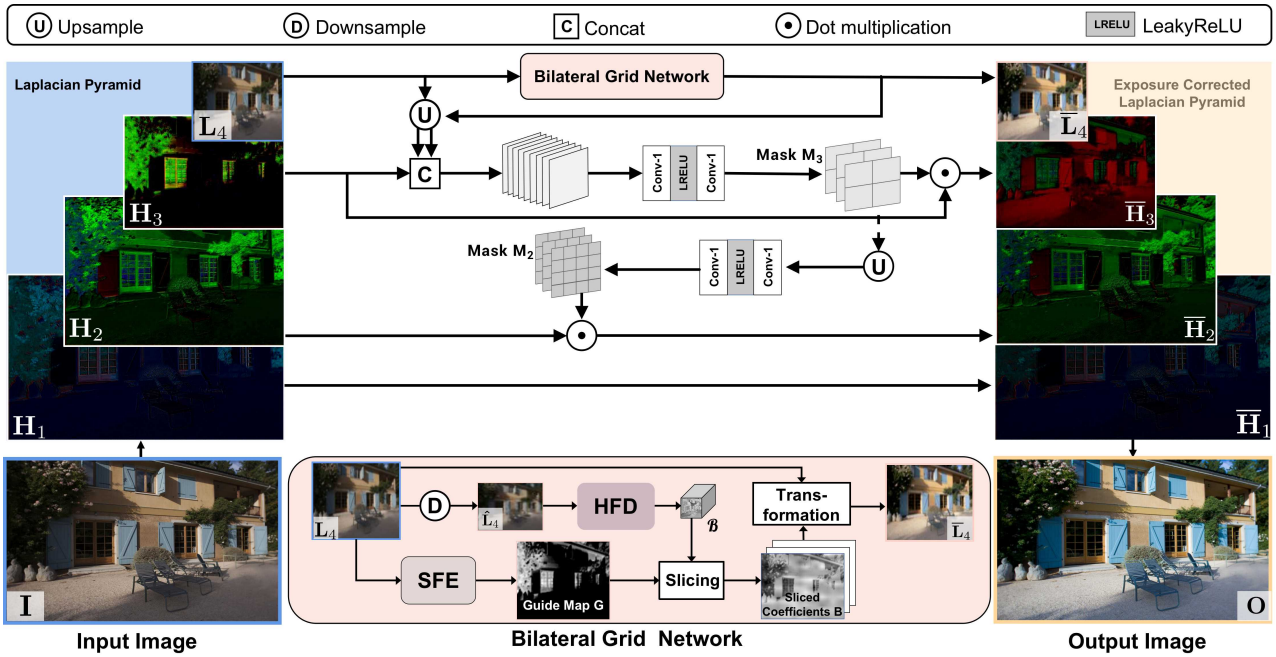
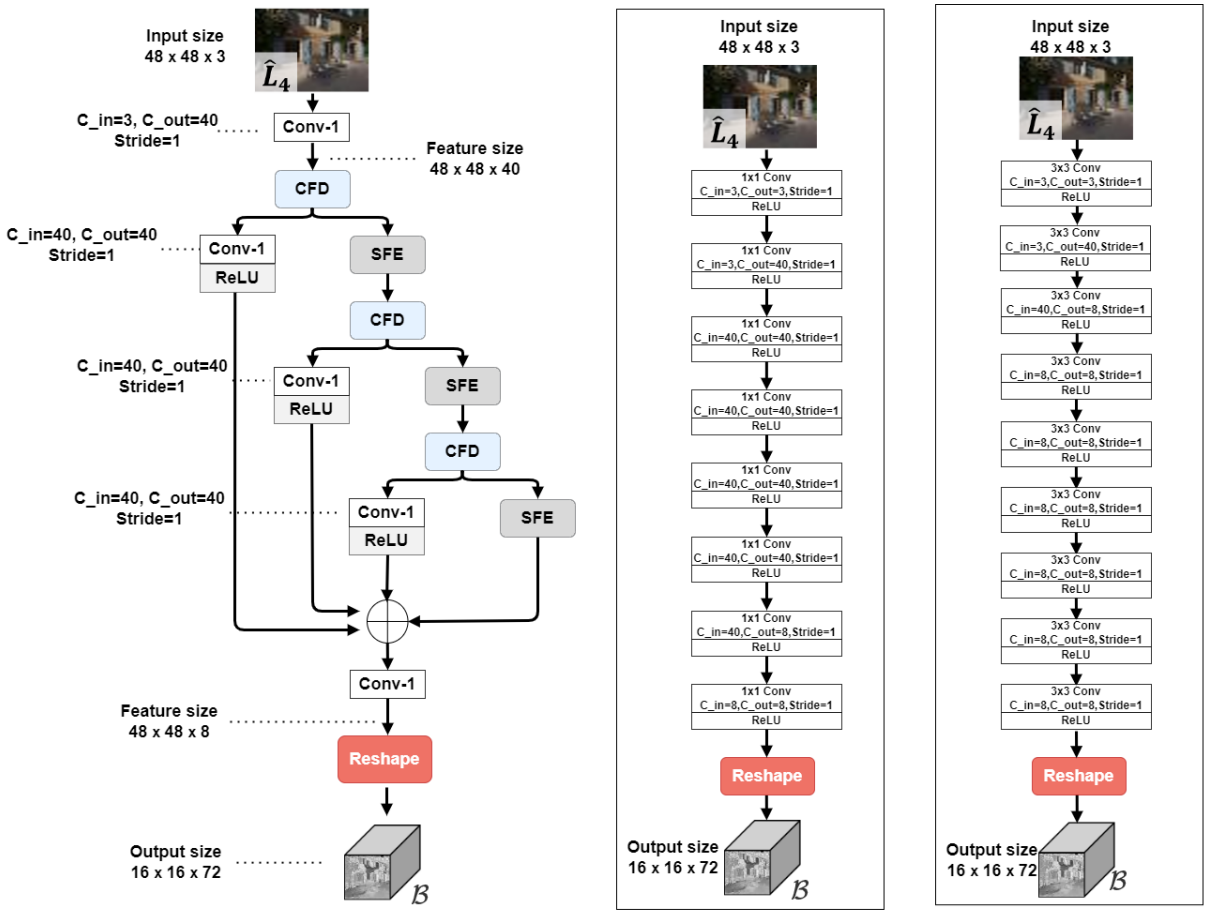


Figure 3. **Overview of our MSLT++ network**. Based on MSLT+ network, we remove the mask prediction MLP in correcting the high-frequency layer $\mathbf{H}_1$ in MSLT+, and directly using the $\mathbf{H}_1$ together with other corrected layers $\{\overline{\mathbf{L}}_4, \overline{\mathbf{H}}_3, \overline{\mathbf{H}}_2\}$ for final LP reconstruction.



(a) Our HFD module (b) "Conv-1" (c) "Conv-3"

Figure 4. **The detailed structure of (a) our HFD module, (b) "Conv-1" and (c) "Conv-3" in Section 4.3 (4) of our paper.** All these three networks take a feature map of $48 \times 48 \times 3$ as input and output a 3D bilateral grid of affine coefficients $\mathcal{B} \in \mathbb{R}^{16\times16\times72}$. C_in and C_out denote the number of input and output channels of convolutions, respectively.

Figure 5. **Visual quality comparison of exposure corrected images from ME dataset [1] for 0 exposure value**.

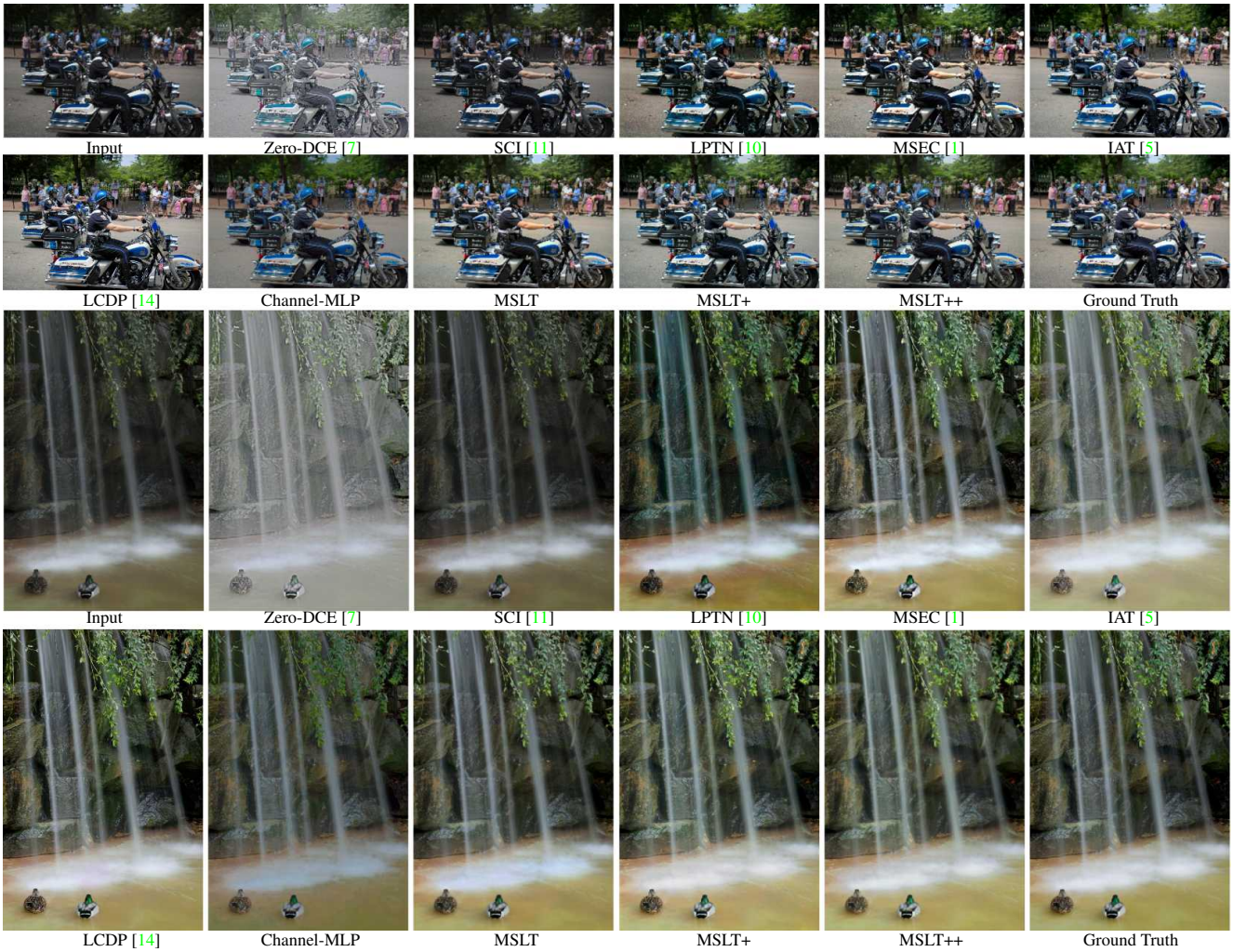


Figure 6. **Visual quality comparison of exposure corrected images from ME dataset [1] for -1 exposure value**.

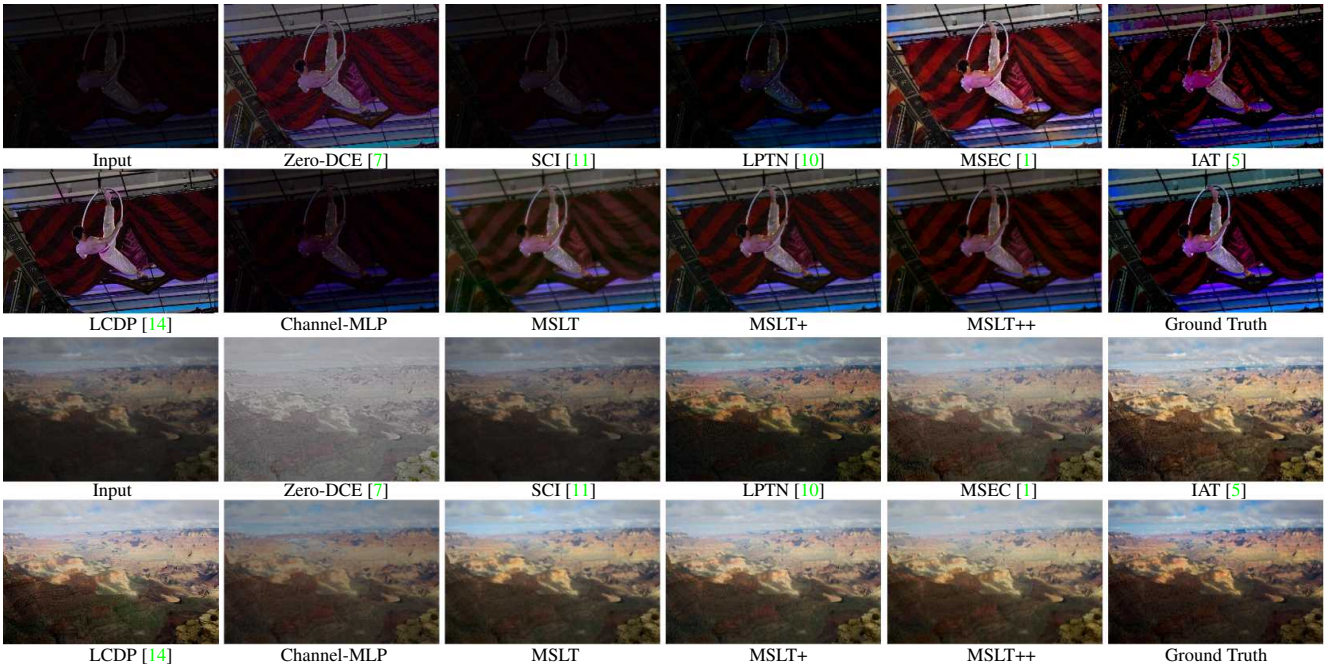


Figure 7. **Visual quality comparison of exposure corrected images from ME dataset [1] for -1.5 exposure value**.

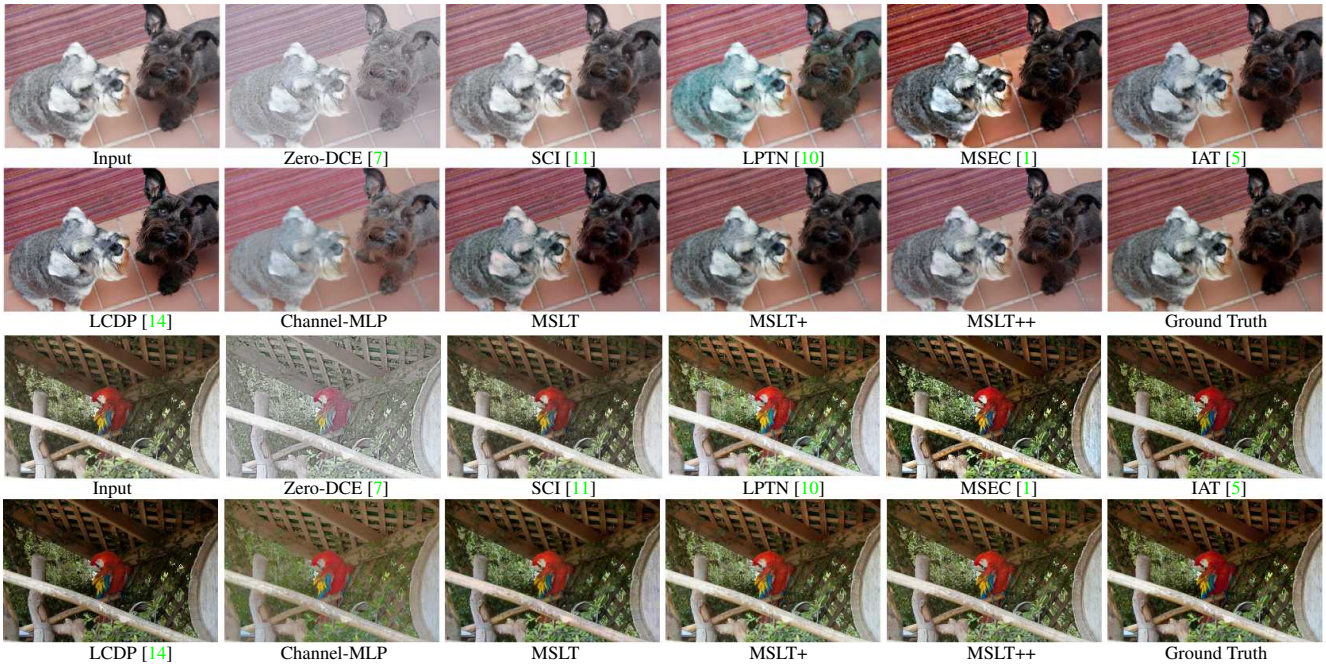


Figure 8. **Visual quality comparison of exposure corrected images from ME dataset [1] for +1 exposure value**.

Figure 9. **Visual quality comparison of exposure corrected images from ME dataset [1] for +1.5 exposure value**.

Figure 10. **Visual quality comparison of under exposure corrected images from SICE dataset [3].**

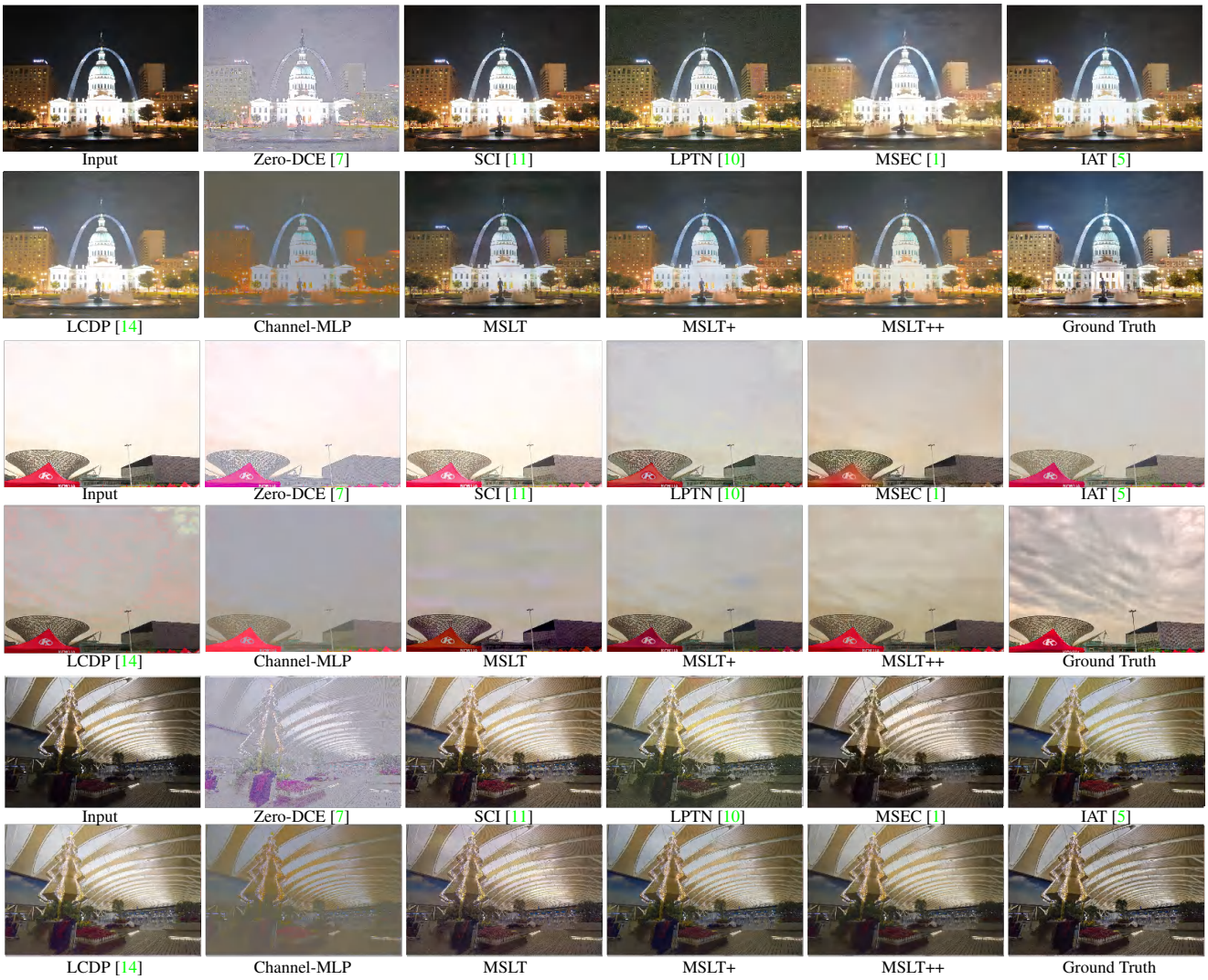


Figure 11. **Visual quality comparison of over exposure corrected images from SICE dataset [3].**

Figure 12. **Visual quality comparison of exposure corrected images processed by our MSLT with different number of Laplacian pyramid levels**. "w/o LP" means we do not use Laplacian pyramid.

Figure 13. **Visual quality comparison of exposure corrected images processed by our MSLT with different variants of CFD module in our HFD module**. "CFD": Context-aware Feature Decomposition. "IN": Instance Normalization [13] with feature decomposition. "CA": Channel Attention [8] with feature decomposition.

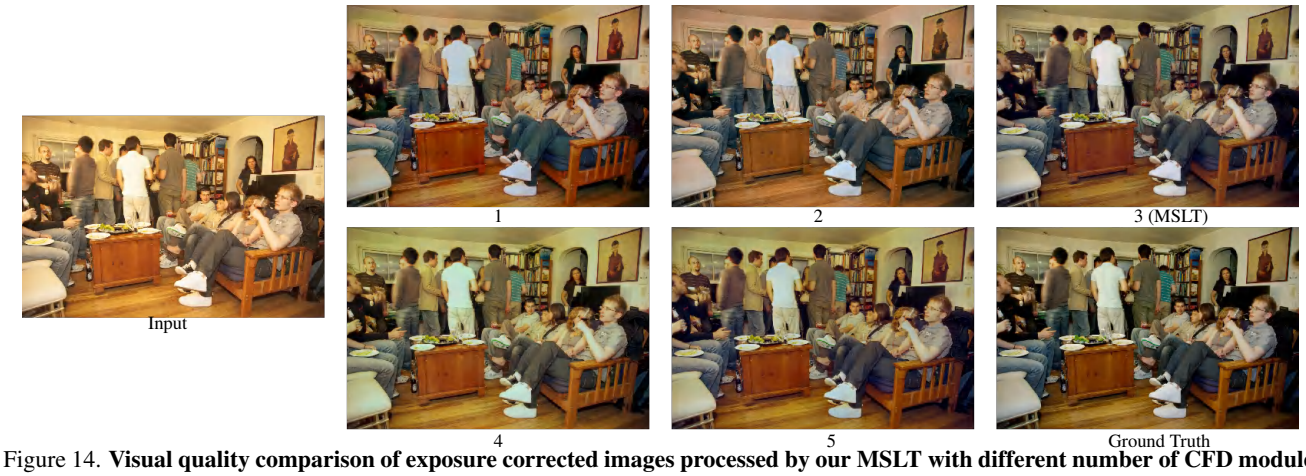


Figure 14. **Visual quality comparison of exposure corrected images processed by our MSLT with different number of CFD modules** in the proposed HFD module.

Figure 15. **Visual quality comparison of exposure corrected images processed by our MSLT with different variants of HFD module** in the developed Bilateral Grid Network. "Conv-1" (or "Conv-3"): the network consisting of multiple 1×1 (or 3×3) convolutional layers and ReLU activation function(see 4). "HFD": our Hierarchical Feature Decomposition module.

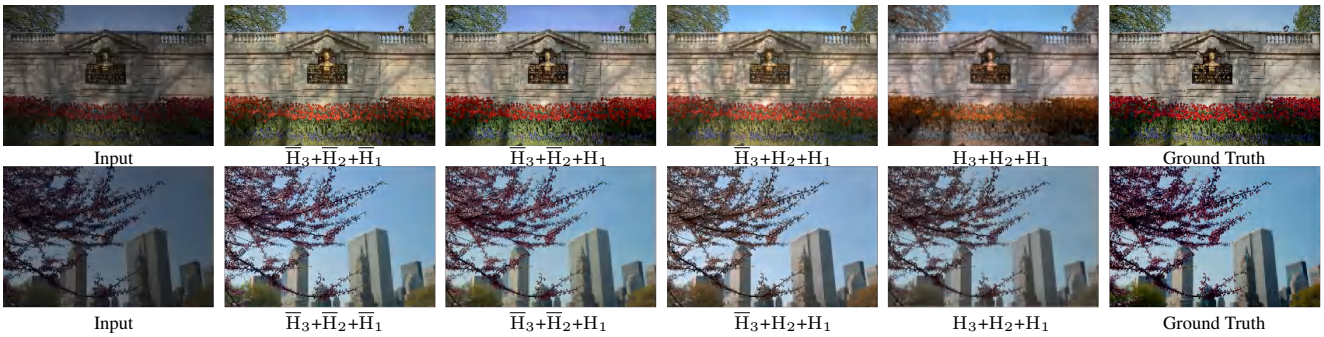
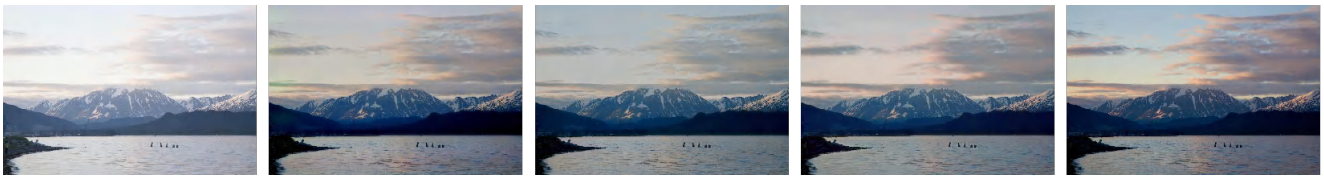


Figure 16. **Visual quality comparison of exposure corrected images processed by our MSLT(1st row) and MSLT+(2nd row) with some high-frequency layers in Laplacian pyramid unprocessed by MSLT/MSLT+**. "H_i": the unprocessed high-frequency layer. "$\overline{H}_i$": the exposure-corrected high-frequency layer.

Input not shared shared (MSLT) Ground Truth

Figure 17. **Visual quality comparison of exposure corrected images processed by our MSLT with the parameters of 1×1 convolutions shared or not.** "not shared": deploy independent convolutions between each high-frequency layer. "shared": small MLPs in different high-frequency layers share convolution parameters.



Input GAP GSP GAP+GSP (MSLT) Ground Truth

Figure 18. **Visual quality comparison of exposure corrected images processed by our MSLT which handles whether or not GAP and GSP are used in CFD moudle.** "GAP" ("GSP"): use only "GAP" ("GSP") in our CFD module. "GAP + GSP": use the method of adding the"GAP" and "GSP" in our CFD module.

Input w/o SFEs w/ Conv-1 w/ SFEs (MSLT) Ground Truth

Figure 19. **Visual quality comparison of exposure corrected images processed by our MSLT which handles SFE modules differently.** "w/o SFEs": SFE moudles are removed from HFD. "w/ Conv-1": only one convolution and ReLU are left in HFD. " w/ SFEs": our MSLT.

Input Context-aware feature Residual feature (MSLT) Ground Truth

Figure 20. **Visual quality comparison of exposure corrected images processed by our MSLT with different inputs to SFE**. "Context-aware feature": the context-aware feature is fed into SFE. "Residual feature": residual feature is fed into SFE.