Real-time Image Enhancer via Learnable Spatial-aware 3D Lookup Tables: Supplementary Material

Tao Wang[*], Yong li[*], Jingyang Peng[*], Yipeng Ma, Xian Wang, Fenglong Song[†], Youliang Yan[†]
Huawei Noah's Ark Lab
{wangtao10,liyong156,pengjingyang,mayipeng,wangxian10,songfenglong,yanyouliang}@huawei.com

Contents

The structure of this supplementary material is as follows. Section 1 introduces the structure of our network and spatial-aware trilinear interpolation . Section 2 gives more details on the definition of the loss function. More comparison results and additional analysis are demonstrated in Section 3.

1. Methodology

Self-adaptive two-head weight predictor. Our self-adaptive two-head weight predictor adopts a UNet-style structure, consisting of an encoder for features compression and a decoder for pixel-wise category prediction. Skip connections concatenate output features from encoder layers to corresponding decoder layers, and the detailed structure is illustrated in Figure 1. Additionally, we also introduce another predictor for image-leval scenes categorization, whose input is compressed global features from the encoder. Its architecture is listed in Table 1, where feature size $C \times H \times W$ denotes the corresponding layers produce features with C channels of shape $H \times W$ and T is the number of image-leval scenes.

Layer	Feature Size
Global Feature	$256 \times 1 \times 1$
FC with LeakyRelu	$128 \times 1 \times 1$
Instance Norm	$128 \times 1 \times 1$
FC with LeakyRelu	$64 \times 1 \times 1$
Instance Norm	$64 \times 1 \times 1$
FC with LeakyRelu	$T \times 1 \times 1$

Table 1: Network architecture of scenes category predictor in the self-adaptive two-head weight predictor.

Spatial-aware trilinear interpolation. The core idea of 3D LUT is to retouch input images according to some compressed parameters(i.e., LUTs), which means that one element in 3D Lattice may correspond to multiple neighborhood elements. So an additional operation is needed to improve the smoothness of the enhanced results. Considering the efficiency the performance, trilinear based interpolation is used in our method.

Let $\mathrm{X}^{h,w} = \{X^{h,w,r}, X^{h,w,g}, X^{h,w,b}\}$ be a pixel in input image at location (h, w). Whatever RGB values it has, there would be eight nearest neighbours when $X^{h,w}$ is mapped into a 3D LUT. The minimum coordinate for eight neighbours (i, j, k) in a 3D LUT is obtained through a lookup operation with its RGB value (I^r, I^g, I^b) as follows:

$$\begin{aligned} i' = \frac{X^{h,w,r}}{\Delta}, j' = \frac{X^{h,w,g}}{\Delta}, k' = \frac{X^{h,w,b}}{\Delta} \\ i = \lfloor i' \rfloor, j = \lfloor j' \rfloor, k = \lfloor k' \rfloor \end{aligned} \tag{1}$$

where $\Delta = C_{max}/(N-1)$, C_{max} is the maximum color value and $\lfloor \cdot \rfloor$ is the floor function. Distance between its exact coordinate and the minimum neighbour coordinate are defined as d_i^r, d_j^g, d_k^b.

$$\begin{aligned} d_i^r = i' - i, d_j^g = j' - j, d_k^b = k' - k \\ d_{i+1}^r = 1 - d_i^r, d_{j+1}^g = 1 - d_j^g, d_{k+1}^b = 1 - d_k^b \end{aligned} \tag{2}$$

Combining the Equation 2(image level scenario adaptation) and Equation 3(pixel-wise category fusion) in our paper, the interpolated output $\{Y^{h,w,c} | c \in \{r, g, b\}\}$ at location (h, w) can be obtained as follows. Owning to the spatial-aware attribute of the pixel-wise category weight map $\alpha_m^{h,w}$, the interpolation is defined as spatial-aware trilinear interpolation.

$$\begin{aligned} Y^{h,w,c} &= \sum_{t=0}^{T-1} \omega_t * (\sum_{ii=i}^{i+1} \sum_{jj=j}^{j+1} \sum_{kk=k}^{k+1} d_{ii}^r d_{jj}^g d_{kk}^b O_{(ii,jj,kk)}^{h,w,c}) \\ &= \sum_{t=0}^{T-1} \sum_{ii=i}^{i+1} \sum_{jj=j}^{j+1} \sum_{kk=k}^{k+1} \omega_t d_{ii}^r d_{jj}^g d_{kk}^b \sum_{m=0}^{M-1} \alpha_m^{h,w} O_{(ii,jj,kk)}^{m,c} \end{aligned} \tag{3}$$

[*] Authors contributed equally
[†] Corresponding author

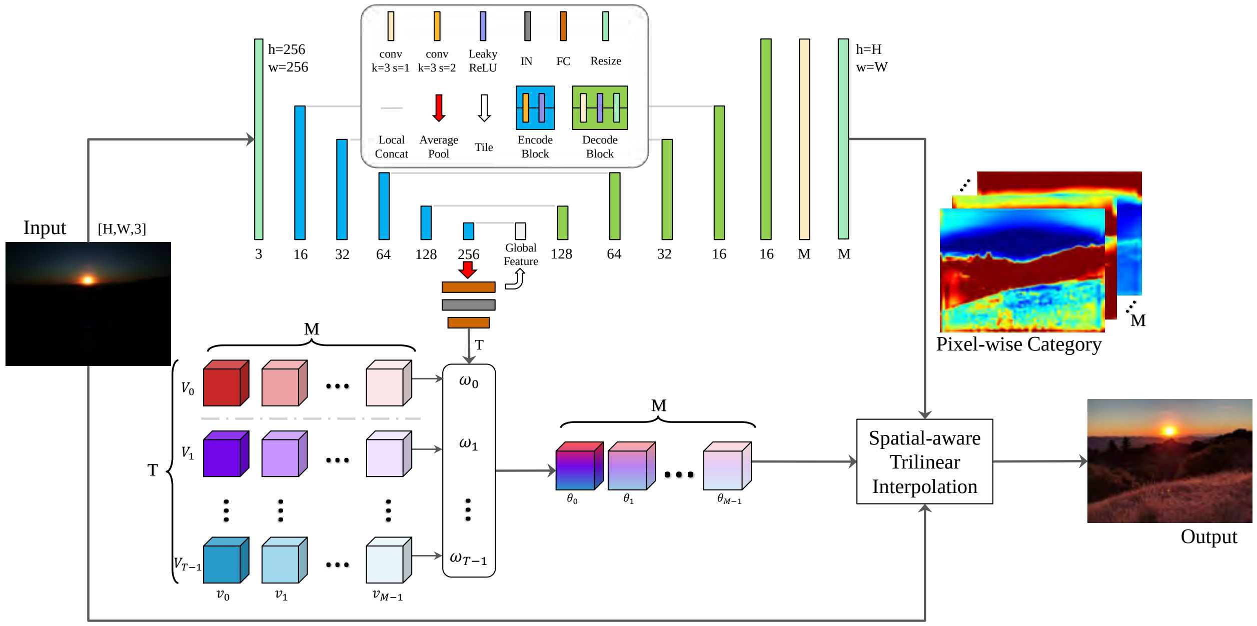


Figure 1: Network architecture of self-adaptive two-head weight predictor.

2. Loss Function

We train our network on a pair-wise dataset $\mathbf{D} = \{(\mathrm{X}_s, \mathrm{Y}_s)|s \in \mathbb{I}_0^{S-1}\}$ with supervised method, where X_s is an input image and Y_s is the corresponding target image. S is the number of image pairs and $s \in \mathbb{I}_0^{S-1}$ is short for $s = 0, 1, \ldots, S-1$. Color Difference Loss L_c and Perception Loss L_p are introduced in more detail in the following.

Color Difference Loss. To measue the color distance and encourage the color in the enhanced image to match that in the corresponding learning target, we use CIE94 in LAB color space as our color loss. Let $(\hat{L}, \hat{a}, \hat{b}), (L, a, b)$ denote predicted and target image in LAB color space, respectively. Then L_c can be defined as Equation 4. Detailed descriptions about it can be found in [5].

$$\begin{aligned}
C_1, C_2 &= \sqrt{\hat{a}^2 + \hat{b}^2 + \epsilon}, \sqrt{a^2 + b^2 + \epsilon} \\
S_C, S_H &= 1 + 0.0225 * (C_1 + C_2), 1 + 0.0075 * (C_1 + C_2) \\
\Delta a,\ \Delta b &= \hat{a} - a,\ \hat{b} - b \\
\Delta C, \Delta L &= C_1 - C_2, \hat{L} - L \\
\Delta H &= \sqrt{\Delta a^2 + \Delta b^2 - \Delta C^2 + \epsilon} \\
L_c &= \sqrt{\Delta L^2 + \left(\frac{\Delta C}{S_C}\right)^2 + \left(\frac{\Delta H}{S_H}\right)^2 + \epsilon} \qquad (4)
\end{aligned}$$

Perception Loss. To improve the perceptual quality of the enhanced image, a widely used LPIPS loss [9] is chosen. It is defined as weighted norms of L_2-distance between features of ground truth images and enhanced images on a pre-trained AlexNet:

$$L_p = \sum_l \frac{1}{H^l W^l} \sum_{h=1,w=1}^{H^l, W^l} \beta_l \times \left\| \hat{y}^l_{h,w} - y^l_{h,w} \right\|_2^2 \qquad (5)$$

where l is the layer chosen to calculate the LPIPS loss, β_l is the weight for the layer l, and $\hat{y}^l, y^l$ is the corresponding ground truth features and enhanced features. By default, we choose outputs from the first five ReLU layers from AlexNet, and all β_l are set to 1.

3. Additional Analysis

In this section, we compare our model with several SOTA methods on two datasets with different resolution. Visualizations show that our model outcompetes other methods on both 480p datasets and on high resolution dataset.

3.1. Comparison on 480p MIT-Adobe FiveK Dataset (released by [8])

We directly use the released dataset and nothing is changed. To be clear, it contains 4500 training pairs and 498 pairs for testing. More visual comparisons can be found in Figure 2 and Figure 3. Each figure shows results from eight methods and their corresponding error maps.

3.2. Comparison on Full Resolution MIT-Adobe FiveK Dataset (Ours)

In this subsection, we train and test the proposed model on our constructed full-resolution MIT-Adobe FiveK dataset. More visual comparisons can be found in Figure 4 and Figure 5.

3.3. Comparison on 480p HDR+ Burst Photography Dataset (released by [8])

We test performance on the open source HDR+ burst Photography dataset released by [8]. More visual comparisons can be found in Figure 6 and Figure 7.

3.4. Comparison on 480p HDR+ Burst Photography Dataset (Ours)

We also test performance on our constructed 480p HDR+ burst Photography dataset, with post-processing mentioned in our paper. More visual comparisons can be found in Figure 8, Figure 9 and Figure 10.

3.5. Comparison on Full Resolution HDR+ Burst Photography Dataset (Ours)

We then test our algorithm on our constructed full resolution HDR+ burst photography dataset. The dataset is post-processed generally the same as what we mentioned in section 3.4. The only difference is that all image pairs are kept as their original size, and no resize is applied. Results show that our model work well on high resolution images. More visual comparisons can be found in Figure 11, Figure 12 and Figure 13.

3.6. Failure Cases

We further show some test cases where our model fails to produce visually pleasant enhanced images. Results show that our model sometimes may suffer from artifacts. In this section, we directly used the full resolution HDR+ burst photography dataset as mentioned in section 3.5. More visual comparisons can be found in Figure 14 and Figure 15.

References

[1] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017. 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

[2] Jie Huang, Zhiwei Xiong, Xueyang Fu, Dong Liu, and Zheng-Jun Zha. Hybrid image enhancement with progressive laplacian enhancing unit. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1614–1622, 2019. 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

[3] Jie Huang, Pengfei Zhu, Mingrui Geng, Jiewen Ran, Xingguang Zhou, Chen Xing, Pengfei Wan, and Xiangyang Ji. Range scaling global u-net for perceptual image enhancement on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

[4] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3277–3285, 2017. 4, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

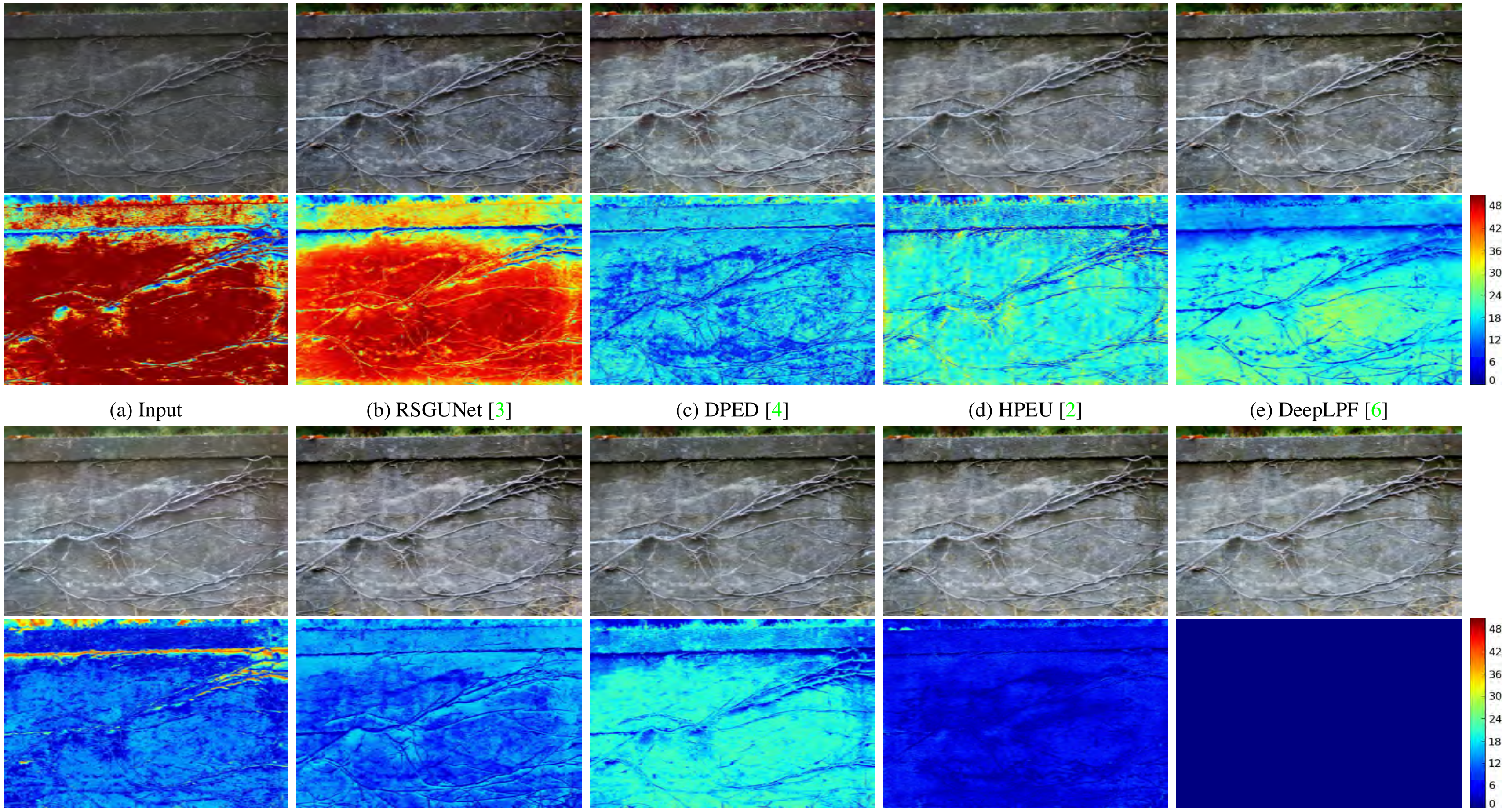
[5] Bruce Justin Lindbloom. *Delta E (CIE 1994)*, 2017 (accessed November 10, 2020). `http://www.brucelindbloom.com/index.html?Eqn_DeltaE_CIE94.html`. 2

[6] Sean Moran, Pierre Marza, Steven McDonagh, Sarah Parisot, and Gregory Slabaugh. Deeplpf: Deep local parametric filters for image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12826–12835, 2020. 4, 5, 8, 9, 10, 11, 12

[7] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6849–6857, 2019. 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

[8] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

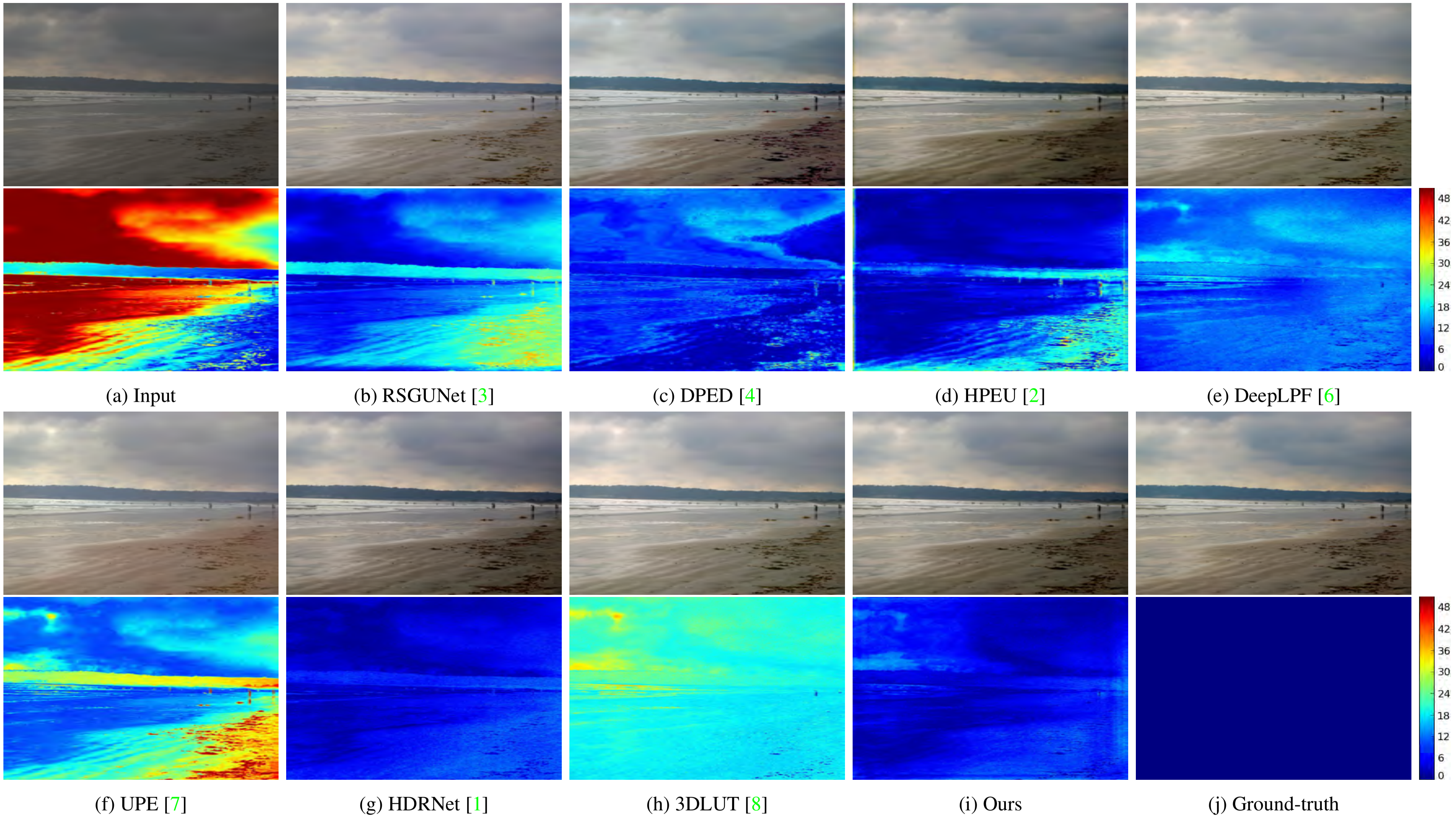
[9] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 2



(a) Input (b) RSGUNet [3] (c) DPED [4] (d) HPEU [2] (e) DeepLPF [6]

(f) UPE [7] (g) HDRNet [1] (h) 3DLUT [8] (i) Ours (j) Ground-truth

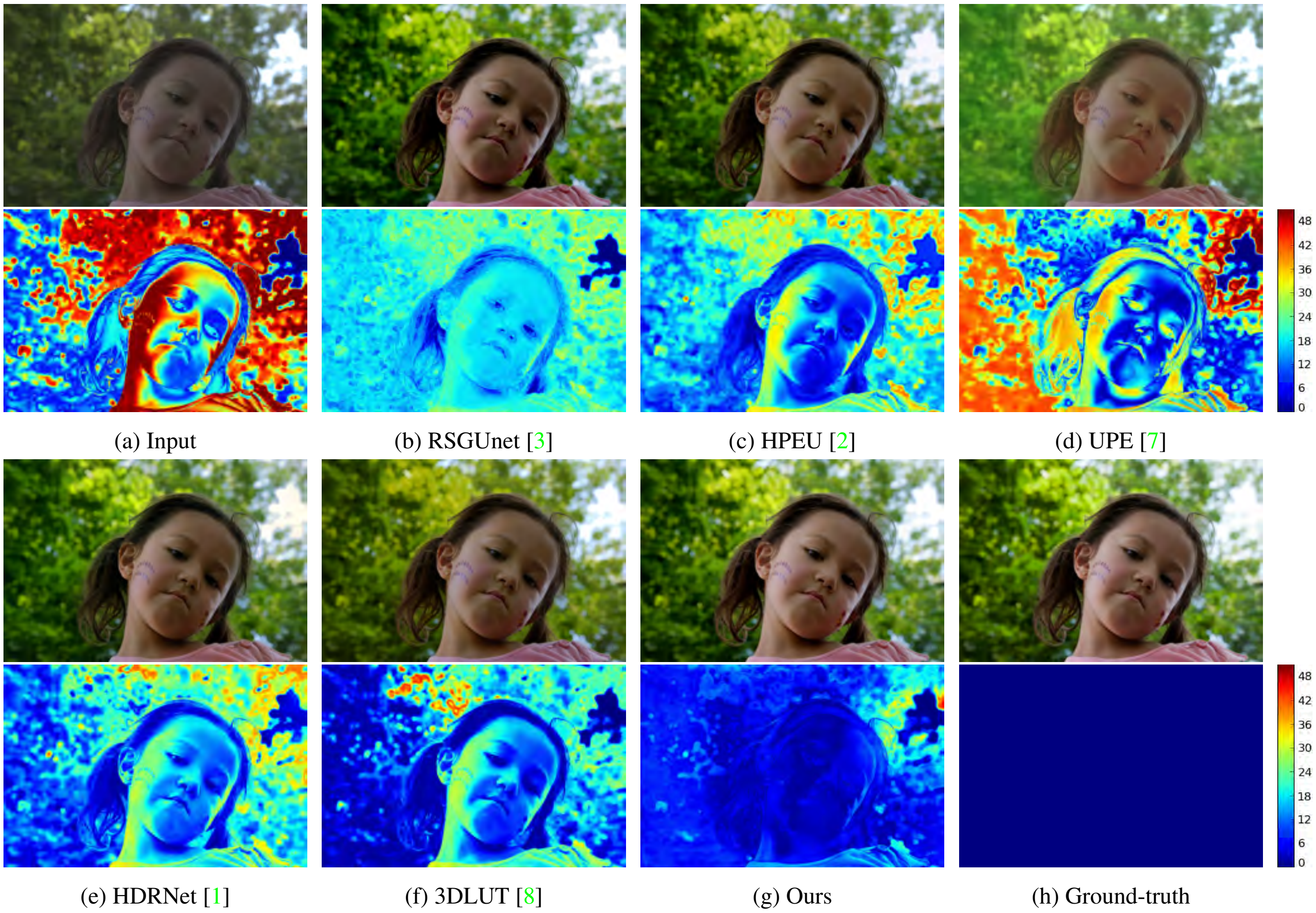
Figure 2: Results comparison on ‘a4050’ of 480p MIT-Adobe FiveK dataset, and corresponding error maps. For each pair of results, the upper image is an enhanced image, and the image below is an error map between the enhanced image and the ground-truth.



(a) Input (b) RSGUNet [3] (c) DPED [4] (d) HPEU [2] (e) DeepLPF [6]

(f) UPE [7] (g) HDRNet [1] (h) 3DLUT [8] (i) Ours (j) Ground-truth

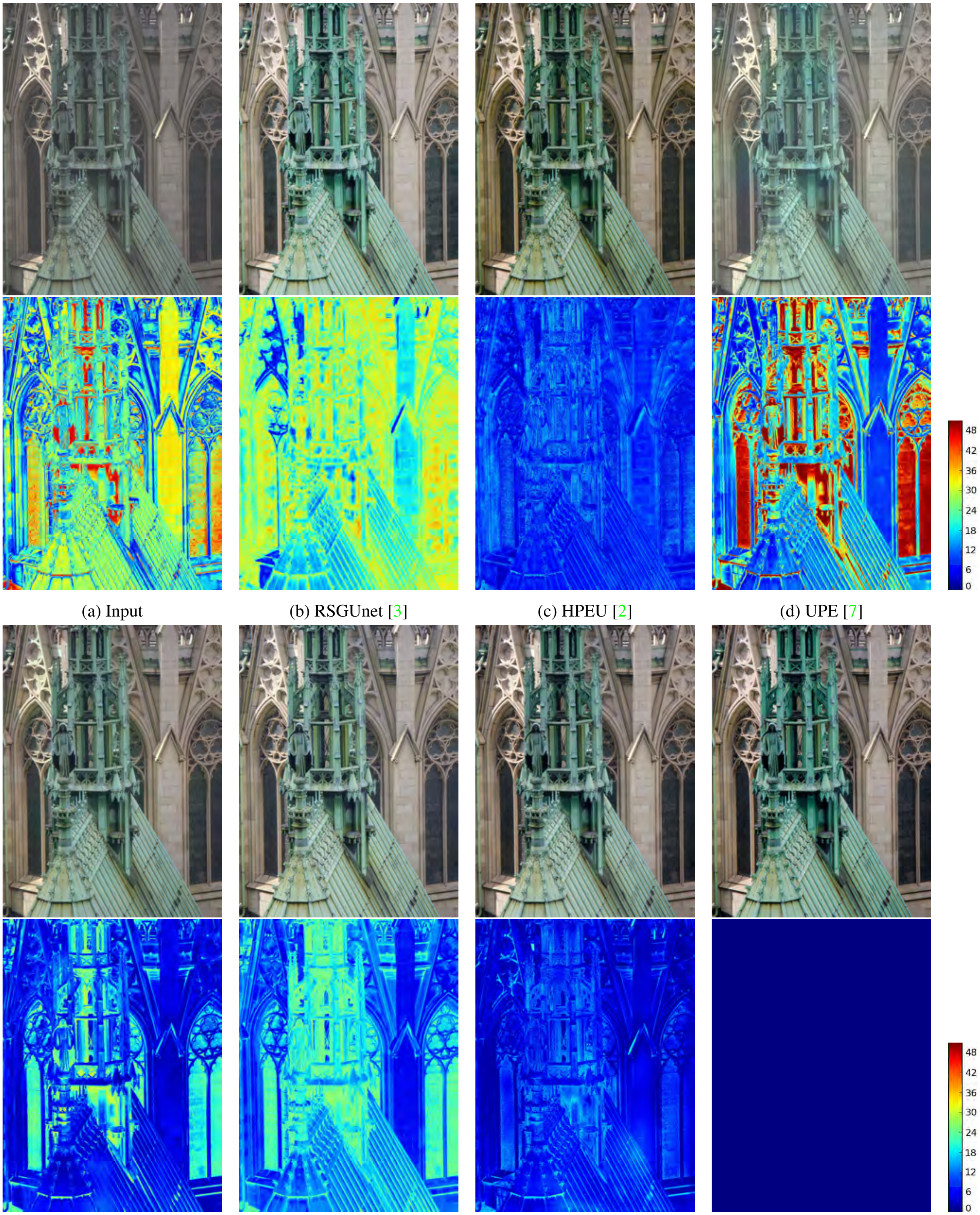
Figure 3: Results comparison on ‘a4816’ of 480p MIT-Adobe FiveK dataset, and corresponding error maps.



(a) Input (b) RSGUnet [3] (c) HPEU [2] (d) UPE [7]

(e) HDRNet [1] (f) 3DLUT [8] (g) Ours (h) Ground-truth

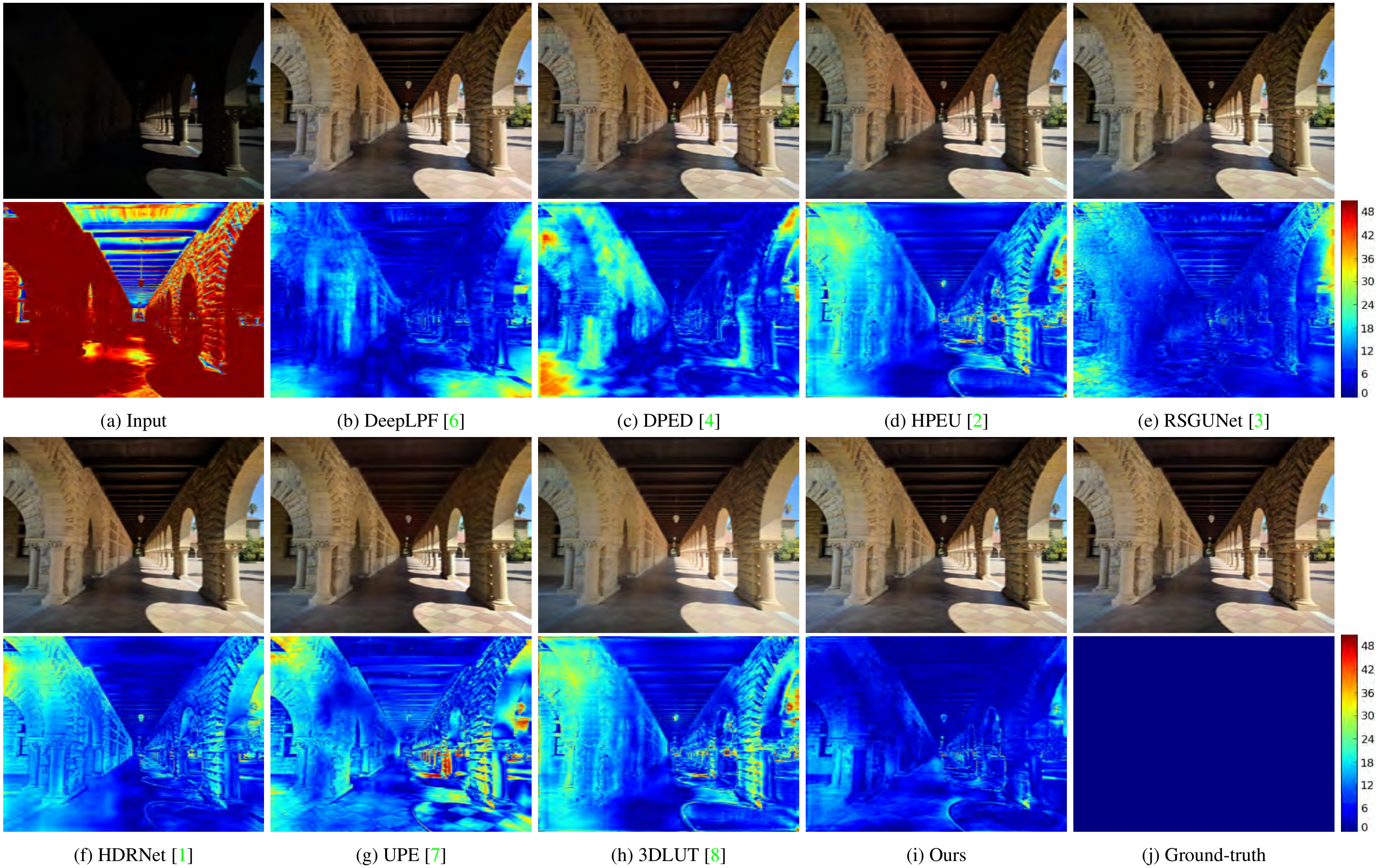
Figure 4: Results comparison on ‘a4163’ of full-resolution MIT-Adobe FiveK dataset, and corresponding error maps. For each pair of results, the upper image is an enhanced image, and the image below is an error map between the enhanced image and the ground-truth.



(a) Input (b) RSGUnet [3] (c) HPEU [2] (d) UPE [7]

(e) HDRNet [1] (f) 3DLUT [8] (g) Ours (h) Ground-truth

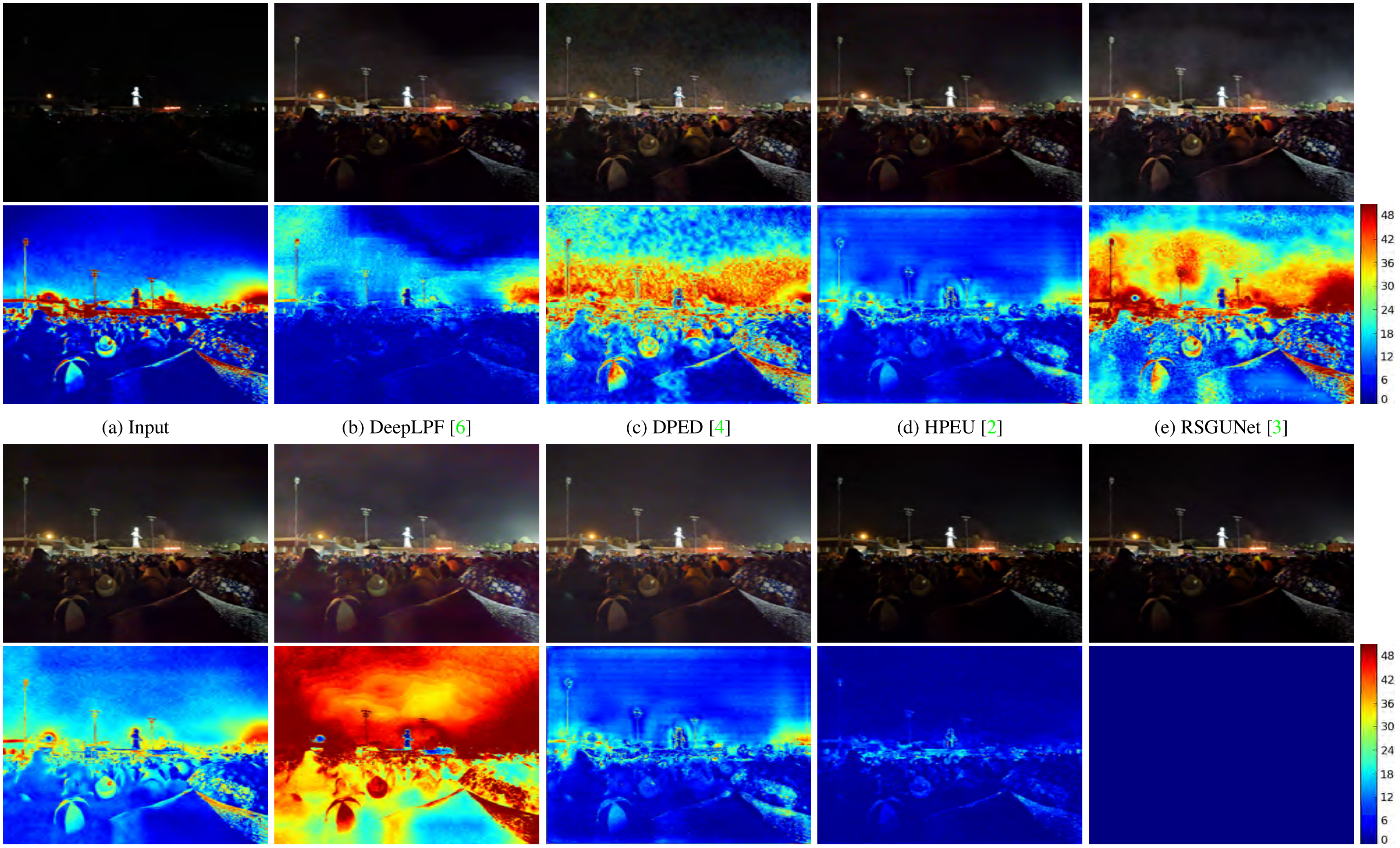
Figure 5: Results comparison on ‘a1544’ of full-resolution MIT-Adobe FiveK dataset, and corresponding error maps.



(a) Input (b) DeepLPF [6] (c) DPED [4] (d) HPEU [2] (e) RSGUNet [3]

(f) HDRNet [1] (g) UPE [7] (h) 3DLUT [8] (i) Ours (j) Ground-truth

Figure 6: Results comparison on ‘1671’ of HDR+ burst photography dataset released by [8], and corresponding error maps. We can see that our method gets less error and artifacts in the sky than other methods.



(a) Input

(b) DeepLPF [6]

(c) DPED [4]

(d) HPEU [2]

(e) RSGUNet [3]

(f) HDRNet [1]

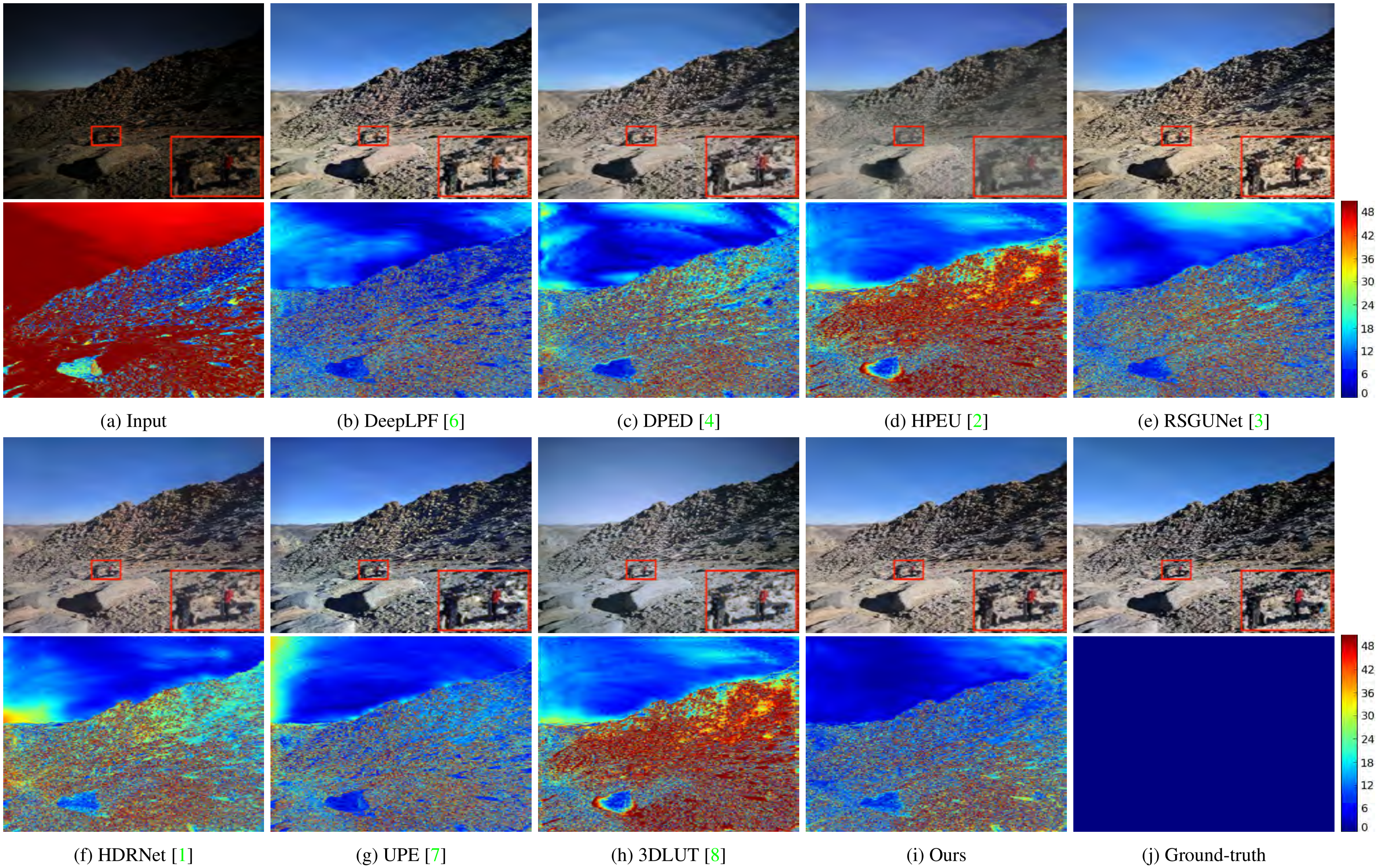
(g) UPE [7]

(h) 3DLUT [8]

(i) Ours

(j) Ground-truth

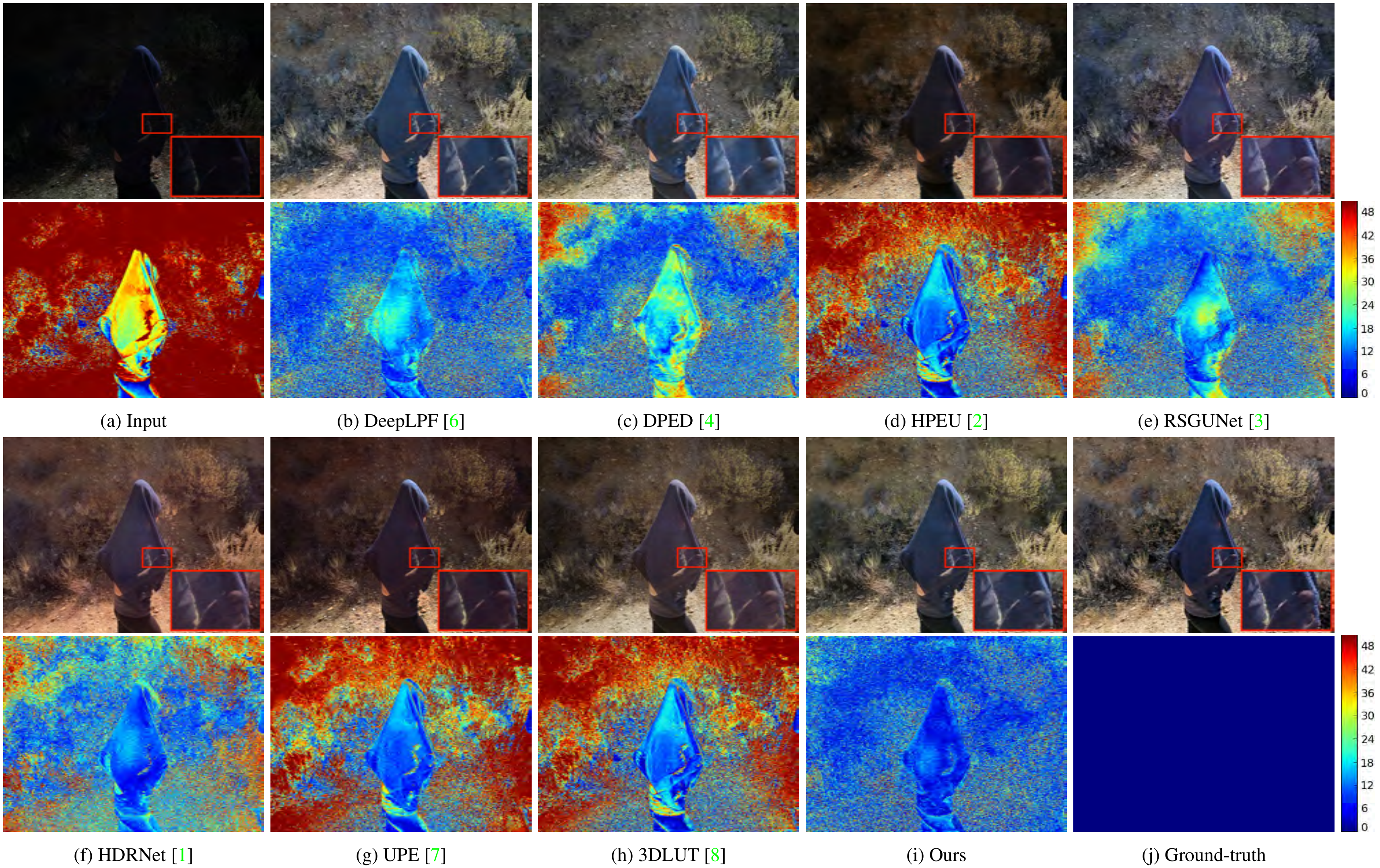
Figure 7: Results comparison on ‘1517’ of HDR+ burst photography dataset released by [8], and corresponding error maps. We can see that our method gets less error in both the foreground and the background areas than the competing approaches.



(a) Input (b) DeepLPF [6] (c) DPED [4] (d) HPEU [2] (e) RSGUNet [3]

(f) HDRNet [1] (g) UPE [7] (h) 3DLUT [8] (i) Ours (j) Ground-truth

Figure 8: Results comparison on '0043_20161006_162052_490' of HDR+ burst photography dataset (Ours), and corresponding error maps. We can see that our method gets less error and artifacts in the sky than other methods.



(a) Input

(b) DeepLPF [6]

(c) DPED [4]

(d) HPEU [2]

(e) RSGUNet [3]

(f) HDRNet [1]

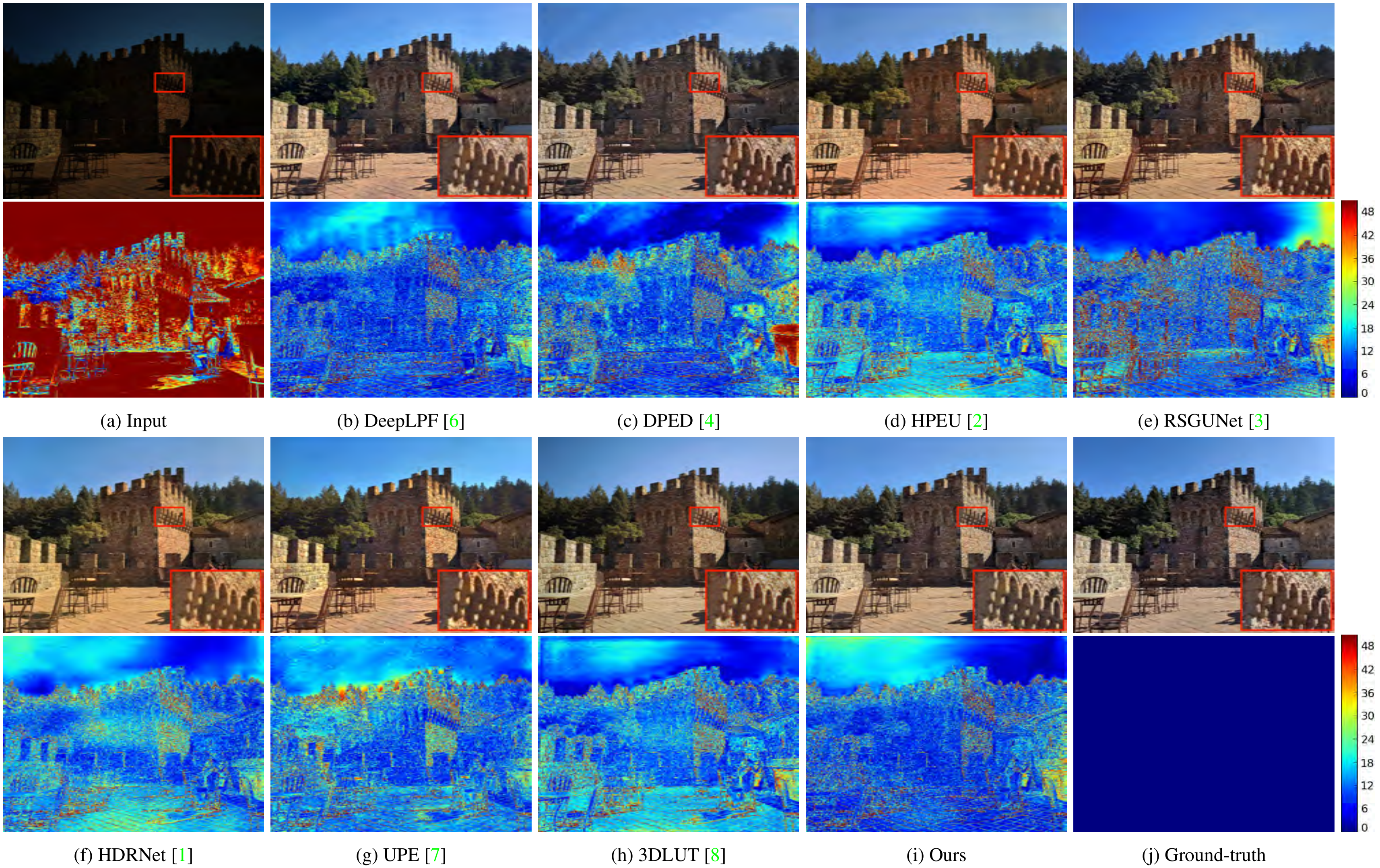
(g) UPE [7]

(h) 3DLUT [8]

(i) Ours

(j) Ground-truth

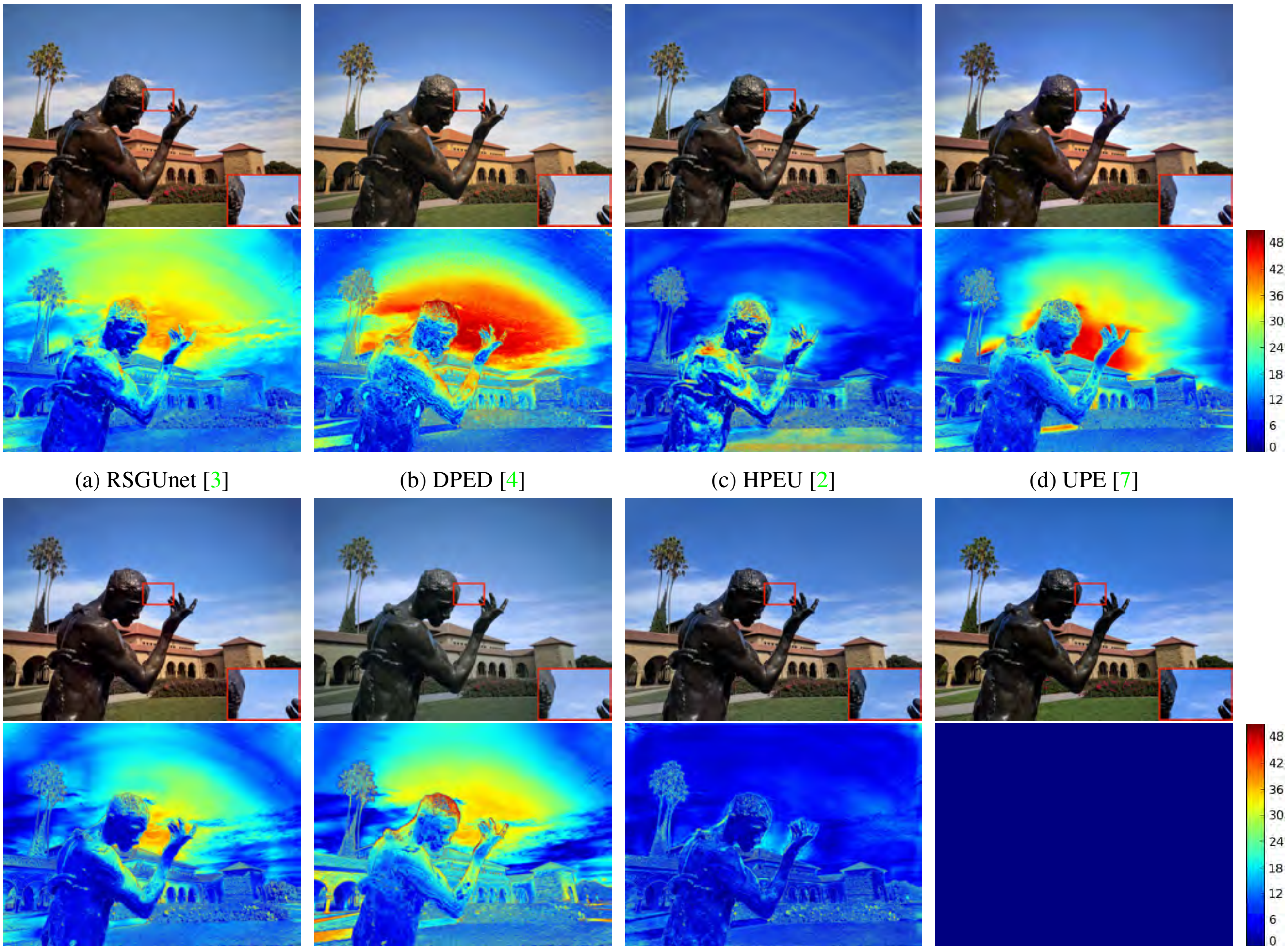
Figure 9: Results comparison on '0006_20160726_110609_666' of HDR+ burst photography dataset (Ours), and corresponding error maps. We can see that our method gets less error in both the foreground and the background areas than the competing approaches.



(a) Input (b) DeepLPF [6] (c) DPED [4] (d) HPEU [2] (e) RSGUNet [3]

(f) HDRNet [1] (g) UPE [7] (h) 3DLUT [8] (i) Ours (j) Ground-truth

Figure 10: Results comparison on ‘5a9e_20150403_162152_482’ of HDR+ burst photography dataset (Ours), and corresponding error maps. HPEU, HDRNet, UPE and 3DLUT get large errors in both building and sky areas. RSGUNet performs better on sky areas, though the right side of the sky area has larger error. Artifacts occur in the sky area of the DPED method, and some building details are lost. DeepLPF shows equal quality compared with our method, but it is not a 4K real-time algorithm.



(a) RSGUnet [3]

(b) DPED [4]

(c) HPEU [2]

(d) UPE [7]

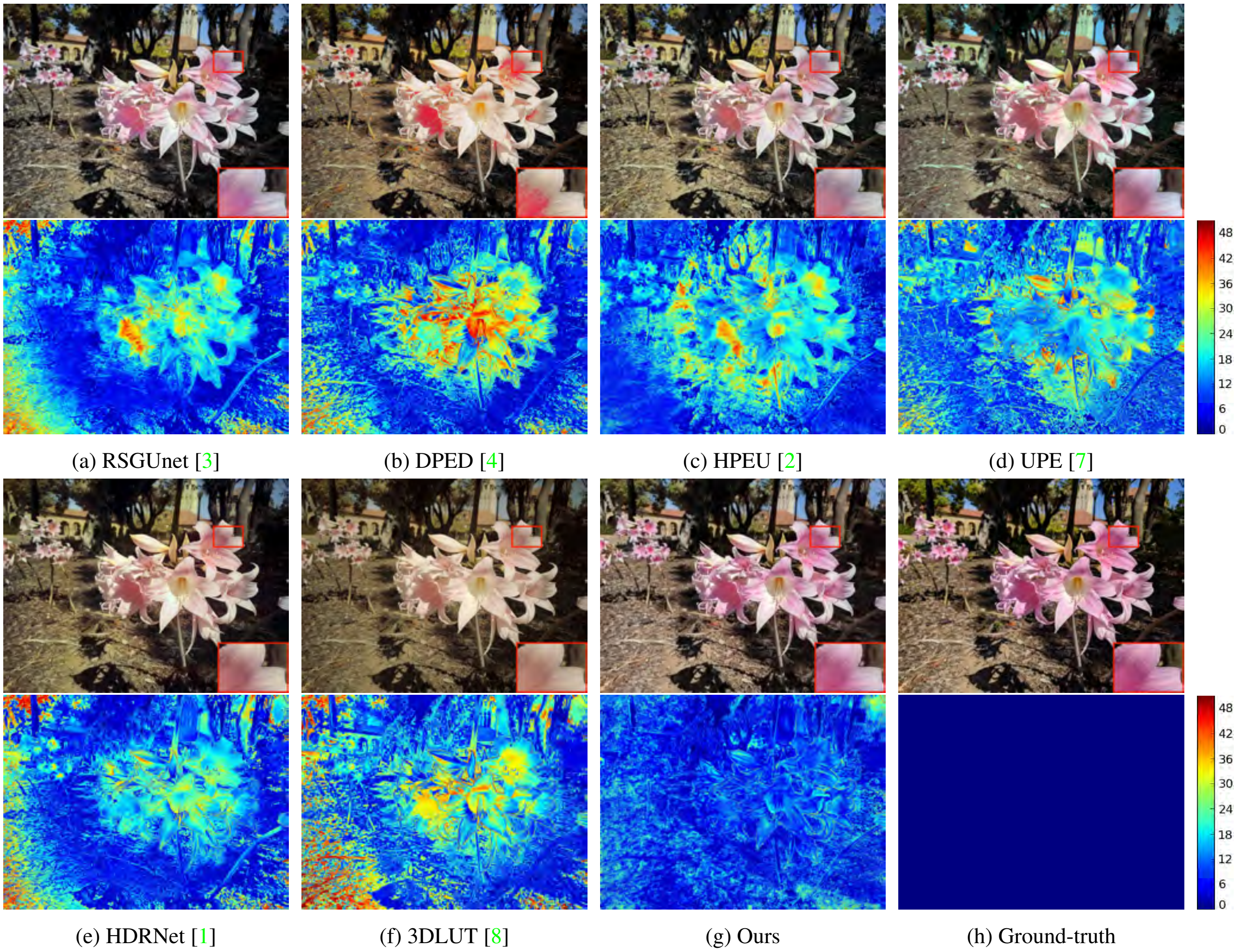
(e) HDRNet [1]

(f) 3DLUT [8]

(g) Ours

(h) Ground-truth

Figure 11: Results comparison on '0382_20150924_100900_333' of full resolution HDR+ burst photography dataset, and corresponding error maps. Our result is the closest to ground truth in color and perception. DPED and HPEU show visible banding artifacts in sky, while RSGUNet, DeepUPE, HDRNet and original 3DLUT suffer from color bias in both sky and grass. Our spatial-aware 3DLUT has the smallest error to ground truth, and the most pleasant visual perception in local contrast.



(a) RSGUnet [3]

(b) DPED [4]

(c) HPEU [2]

(d) UPE [7]

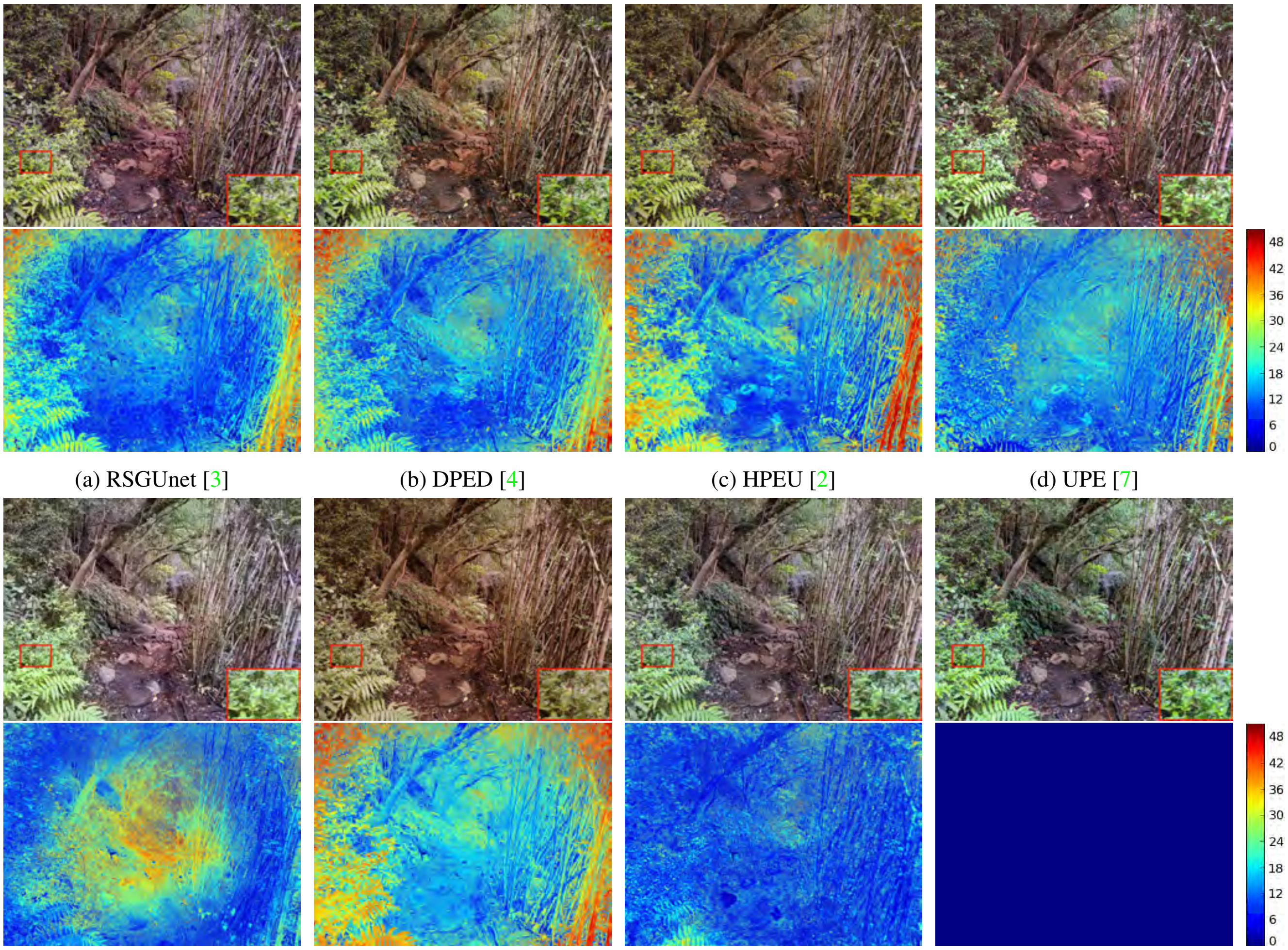
(e) HDRNet [1]

(f) 3DLUT [8]

(g) Ours

(h) Ground-truth

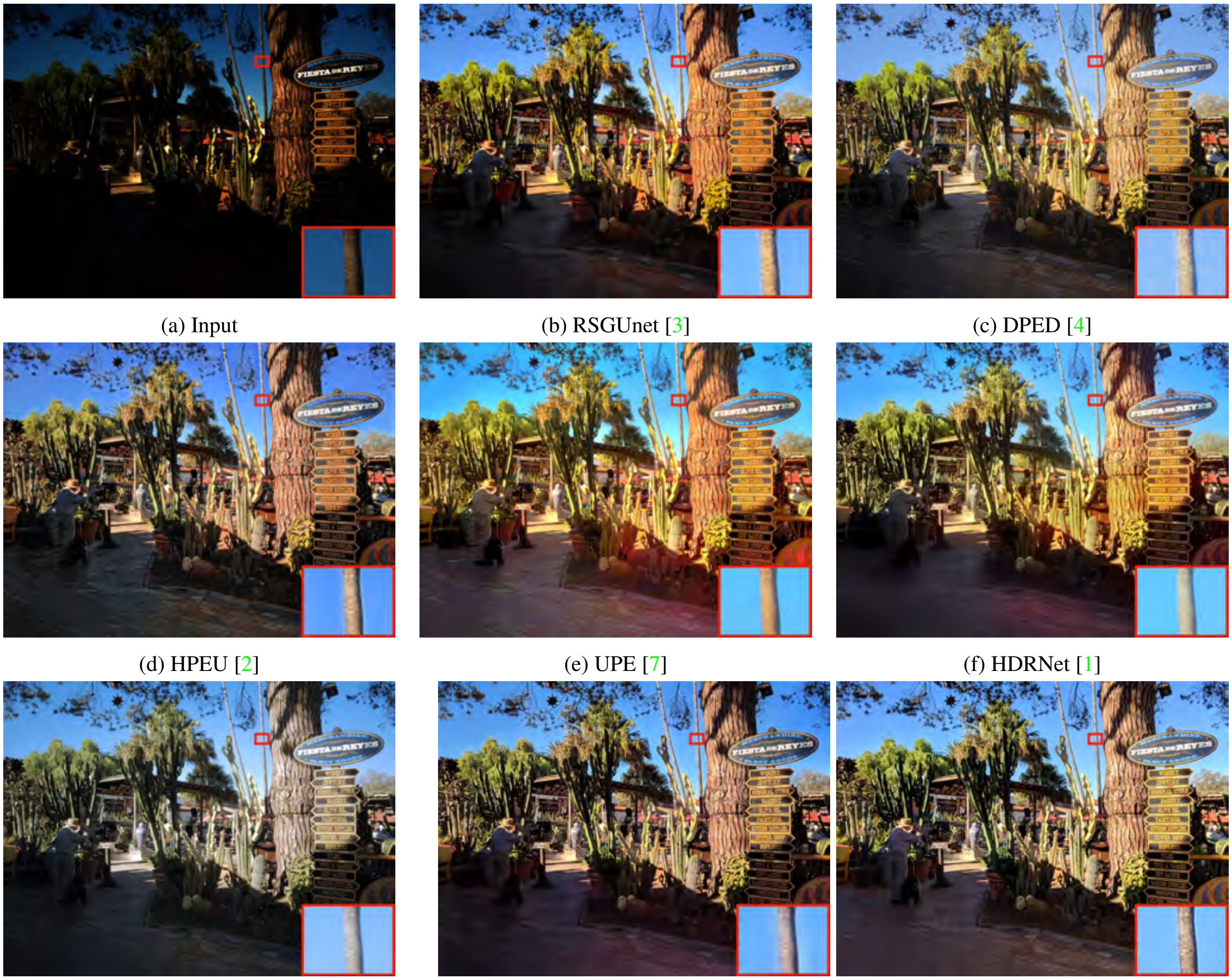
Figure 12: Results comparison on '0919_20150910_150832_572' of full resolution HDR+ burst photography dataset, and corresponding error maps.



(a) RSGUnet [3] (b) DPED [4] (c) HPEU [2] (d) UPE [7]

(e) HDRNet [1] (f) 3DLUT [8] (g) Ours (h) Ground-truth

Figure 13: Results comparison on '1125_20151229_192447_145' of full resolution HDR+ burst photography dataset, and corresponding error maps.

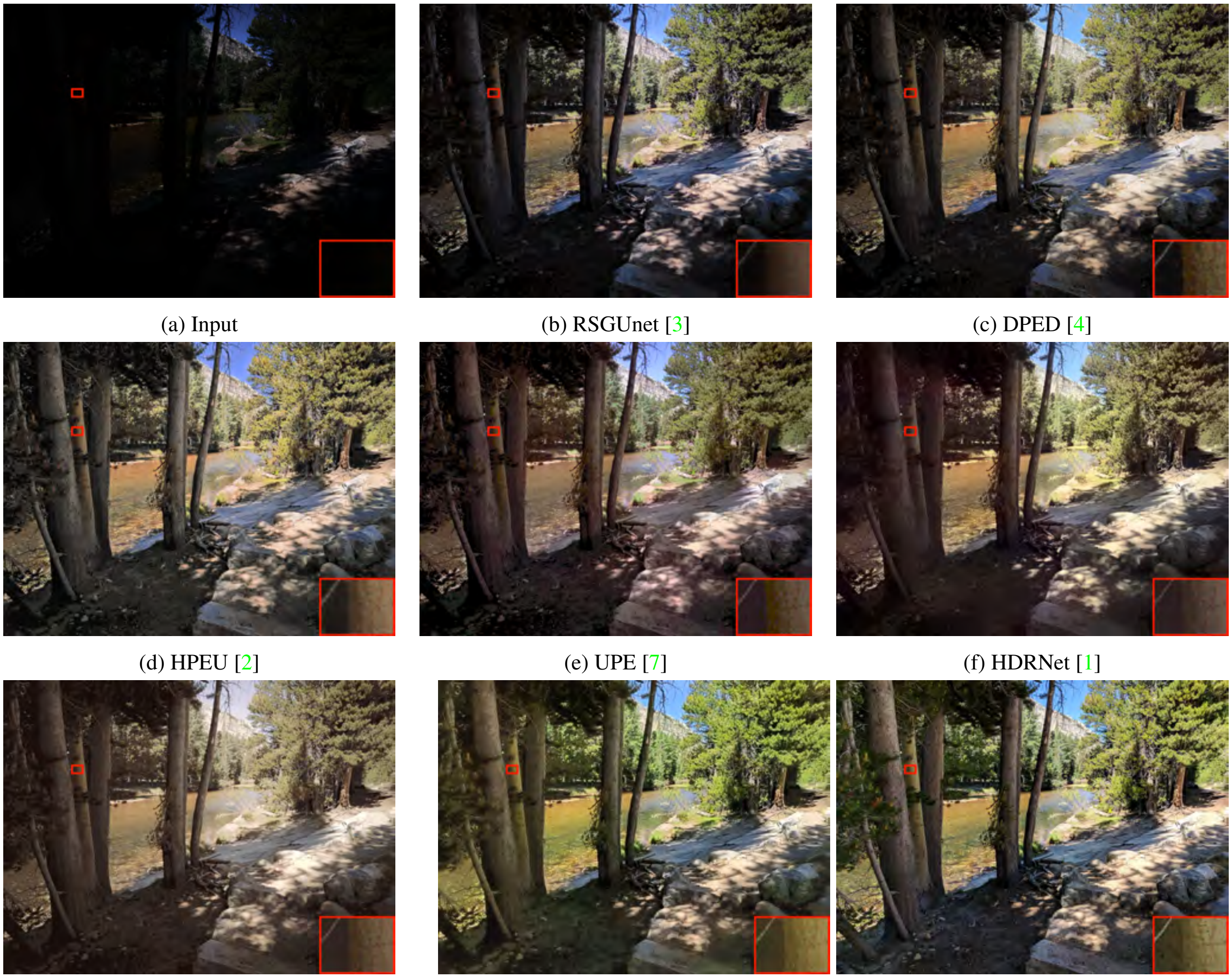


(a) Input (b) RSGUnet [3] (c) DPED [4]

(d) HPEU [2] (e) UPE [7] (f) HDRNet [1]

(g) 3DLUT [8] (h) Ours (i) Ground-truth

Figure 14: One failure case on ‘5a9e_20141005_162240_437’ of full resolution HDR+ burst photography dataset. Although our model is the closest to ground truth in most areas, halo artifacts are sometimes visible in our results where color changes sharply (e.g., around trunks). We believe this is caused by the unsmooth pixel-wise category weights generated by our two-head weight predictor (i.e., category weights change un-smoothly). Thus, such drawback can potentially be solved by introducing additional smooth loss as introduced in our paper.



(a) Input

(b) RSGUnet [3]

(c) DPED [4]

(d) HPEU [2]

(e) UPE [7]

(f) HDRNet [1]

(g) 3DLUT [8]

(h) Ours

(i) Ground-truth

Figure 15: One failure case on '0006_20160722_101752_239' of full resolution HDR+ burst photography dataset. Our model shows the best local contrast in most cases, however, all methods suffer from banding artifacts in extremely dark regions where Signal-to-Noise Ratio is poor. Since our inputs are of 8-bit, too little useful information is carried in extremely dark areas and noises dominate signals. Possible solutions including cooperating with other denoising preprocessing, and replacing low-precision 8-bit inputs with some high-precision 16-bit images.