Single-Image HDR Reconstruction by Learning to Reverse the Camera Pipeline Supplementary Material

Yu-Lun Liu[1,2*] Wei-Sheng Lai[3*] Yu-Sheng Chen[1] Yi-Lung Kao[1]
Ming-Hsuan Yang[3,4] Yung-Yu Chuang[1] Jia-Bin Huang[5]
[1]National Taiwan University [2]MediaTek Inc. [3]Google [4]UC Merced [5]Virginia Tech
https://www.cmlab.csie.ntu.edu.tw/~yulunliu/SingleHDR

1. Overview

In this supplementary material, we present additional results to complement the main manuscript. We first illustrate detailed network configurations, model size, and execution speed of the proposed model in Section 3. Second, we show the quantitative evaluation of the *tone-mapped* HDR images in Section 4. Third, we analyze the effect of the noise model, edge, and histogram features, and visualize the linearity of the reconstructed HDR images. Finally, we provide visual comparisons with the alternatives of our Dequantization-Net, Linearization-Net, and Hallucination-Net in Section 5. Our project website also provides comprehensive visual comparisons on all the datasets.

2. Datasets

In this section, we describe the details of the datasets we used for training and evaluation. We summarize the statistics of four datasets, HDR-SYNTH, HDR-REAL, RAISE, and HDR-EYE, in Table 1.

2.1. HDR-SYNTH dataset

We collect a total of 562 HDR images used in prior work [8, 10, 11, 27, 30, 31] as well as from online sources, e.g., Pfstools HDR gallery[1]. We split the dataset into 502 HDR images for training and 60 HDR images for evaluation. For each HDR image, we resize the shorter side to 512 and crop the image into two 512 × 512 images.

We normalize all the HDR images to have a mean value of 0.5 as in [7]. To model the sensor noise, we use a realistic noise model [9] to add a Poisson-Gaussian noise to the clean HDR images. The Poisson-Gaussian noise can be approximated by a heteroscedastic Gaussian with a signal-dependent variance $\sigma^2(I) = I \cdot \sigma_s^2 + \sigma_c^2$ where I is the pixel intensity. Given a clean HDR image $\tilde{H}$, we first determine σ_s and σ_c by uniformly sampling from the range of [0, 0.013] and [0, 0.005], respectively. Then, we generate two noise maps: (1) a stationary noise n_c with the variance σ_c^2, and (2) a signal-dependent noise n_s with the spatially variant variance $\tilde{H} \cdot \sigma_s^2$. Finally, a noisy HDR image H is generated by $H = \tilde{H} + n_s + n_c$, which is the target ground-truth of our model.

We synthesize the LDR images using the camera pipeline in Equation 1 of the main paper. To generate training data, we uniformly sample 600 exposure times t in the $\log_2$ space within $[-3, 3]$ and apply 171 different CRFs from the training set of the CRF dataset [12]. In total, we generate 103,010,400 (= $502 \times 2 \times 600 \times 171$) LDR-HDR pairs for training. To generate test data, we use 7 exposure times, $2^{-3}, \cdots, 2^0, \cdots, 2^3$ and apply 10 different CRFs from the test set of the CRF dataset [12]. This gives a total of 8,400 (= $60 \times 2 \times 10 \times 7$) LDR-HDR pairs for evaluation. As real-world LDR images often contain JPEG compression artifacts, we save the synthesized LDR images in a JPEG format with a quality factor randomly sampled from the range of [85, 100]. This process can be considered as a data augmentation procedure, which has been shown effective to improve the reconstruction quality on real images [6].

Our synthetic LDR images may become too dark or too bright due to the random sampling of exposure time. These images are not common cases and too difficult for a model to generate reasonable results. Therefore, we remove these extreme cases by the following strategy. First, we convert an LDR image to grayscale and compute the number of pixels that are within over-exposed or under-exposed regions. The over-exposed region is defined as the pixel intensity greater than $249/255$, and

[1] http://pfstools.sourceforge.net/hdr_gallery.html

Table 1: **Summary of our datasets.**

Dataset	Training		Testing	
	# HDR images	# LDR images	# HDR images	# LDR images
HDR-Synth	502	103,010,400	60	4,669
HDR-Real	410	3,974	70	919
RAISE	-	-	8154	8154
HDR-Eye	-	-	46	46

the under-exposed region is defined as the pixel intensity smaller than $6/255$. During the training stage, we exclude the LDR images which have more than 25% of pixels within the over-exposed or under-exposed region. We use the same strategy to filter out extreme cases in the test set. After the selection, there are a total of 4,669 testing images in the synthetic dataset.

2.2. HDR-Real dataset

We invite about 600 of amateurs to take HDR images and instruct them to capture the scenes with multiple exposures using a steady tripod. The LDR images are captured from 42 different cameras, including NIKON D90, NEX-5N, Canon EOS 60D, etc. We then use Photomatix[2] to fuse the LDR exposure stacks into HDR images, each from 5 to 20 LDR images. In total, we have 480 HDR images and 4,893 LDR images. We choose 70 HDR images and 882 LDR images for evaluation and use the rest for training. We note that the LDR images in the HDR-Synth dataset are *synthesized* from the image formation pipeline in Equation 1 of the main paper, while the LDR images in the HDR-Real dataset are *real* photos captured using different cameras with unknown pipelines.

2.3. RAISE dataset

As the source HDR images in HDR-Synth and HDR-Real datasets are generated by fusing multiple LDR images, there might exist some biases from the conversion tool (e.g., Photomatix). Therefore, we also evaluate on the RAISE [5] dataset, which contains aligned RAW-TIFF image pairs. We consider the RAW images (12 or 14 bits, depending on the capturing camera) as ground-truth HDR images, which are the *unprocessed* signals from the camera sensors. The TIFF images are 8-bit images processed by real camera pipelines (Nikon D90 or D7000). We then convert the TIFF images into the JPEG format as our LDR input. In total, we obtain 8,154 RAW-JPEG image pairs, which are more realistic images than our HDR-Synth and HDR-Real datasets.

2.4. HDR-Eye dataset

We use 46 HDR-LDR image pairs of HDR-Eye dataset [25] for testing. The HDR-Eye dataset is composed of images taken with Sony DSC-RX100 II, Sony NEX-5N, and Sony α6000. The HDR images are created by combining nine bracketed images with different exposures settings (-2.7, -2, -1.3, -0,7, 0, 0,7, 1,3, 2, 2.7 [EV]). The LDR images are 8-bit images in the TIFF format. We also convert the TIFF images into the JPEG format as our LDR input.

3. Network Architectures and Implementation

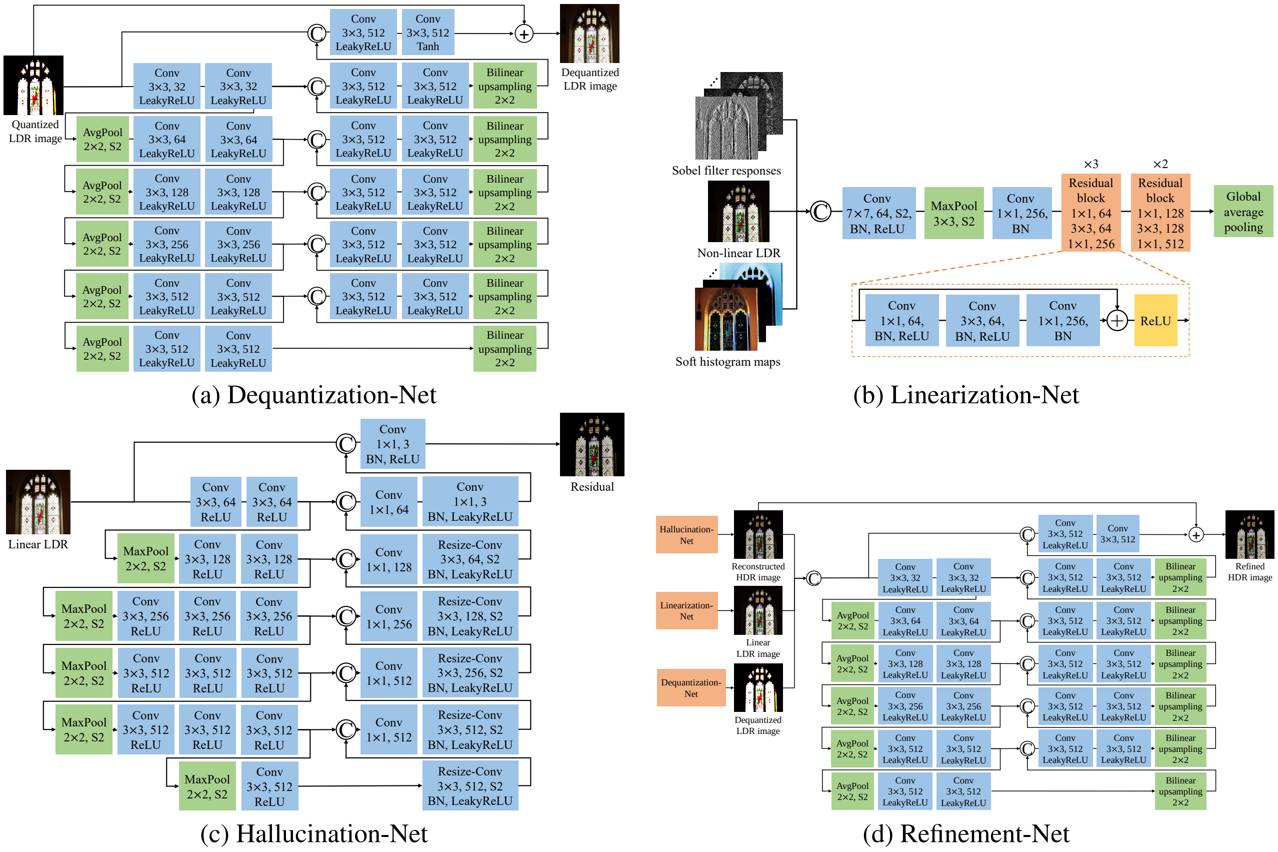
3.1. Histogram feature

. Our histogram layer preserves the spatial information and is fully differentiable. We compute the histogram features for each RGB channel and concatenate the features together, thereby obtaining a total of $3B$ channels of histogram features. In addition, we compute histogram features for $B = 4, 8, 16$ to capture information from different resolutions of the histogram bins. As a result, we obtain a total $84 = 3 \cdot (4 + 8 + 16)$ histogram maps. Along with the non-linear input image (3 channels) and edge responses (6 channels), the input to the Linearization-Net is a 93-channel feature.

3.2. Implementation details

We show the detailed architectures of our Dequantization-Net, Linearization-Net, Hallucination-Net, and Refinement-Net in Figure 1. We implement the proposed model with TensorFlow [1] and use the Adam optimizer [15] with a learning rate of 10^{-4} to train the Dequantization-Net, Linearization-Net, and Hallucination-Net for 500k iterations. We then jointly fine-tune the entire model for 100k iterations by setting the learning rate to be 10^{-5}. We use a batch of 8 for training individual

[2] https://www.hdrsoft.com/



(a) Dequantization-Net

(b) Linearization-Net

(c) Hallucination-Net

(d) Refinement-Net

Figure 1: **Architectures of the sub-networks in the proposed model.** Ⓒ: Concatenate along the feature channel.

Table 2: **Model sizes and execution time.** We compute the number of network parameters and evaluate the execution time of state-of-the-art CNN-based approaches. We also provide the breakdown of individual sub-networks in the proposed model.

Method	Implementation	#Parameters (million)	Time (seconds)
HDRCNN [6]	TensorFlow	29.44	0.334
DrTMO [7]	Chainer	48.09	1.489
ExpandNet [23]	PyTorch	0.45	0.063
Ours	TensorFlow	29.01	0.520
Dequantization-Net	TensorFlow	1.99	0.055
Linearization-Net	TensorFlow	1.20	0.111
Hallucination-Net	TensorFlow	24.56	0.305
Refinement-Net	TensorFlow	1.26	0.049

networks and 4 for jointly fine-tuning the entire model. Finally, we add the Refinement-Net and fine-tune the entire model for 100k iterations with a 10^{-5} learning rate using HDR-SYNTH and HDR-REAL. The entire training process takes about 7 days to converge on a single NVIDIA Titan Xp GPU.

3.3. Model size and speed

We compare the model size (i.e., number of network parameters) of the CNN-based approaches [6, 7, 23] and the proposed model in Table 2. Our Dequantization-Net and Linearization-Net use 7% and 6% of the overall parameters, respectively. While our Hallucination adopts the same architecture as the HDRCNN [6], replacing the transposed convolution (4 × 4 kernels) with the resize convolution (3 × 3 kernels) reduces 17% of parameters. Our overall model size is slightly smaller than HDRCNN and is 41% smaller than DrTMO.

Table 3: **Quantitative comparison on HDR images with existing methods.** * represents that the model is re-trained on our synthetic training data and + is fine-tuned on both synthetic and real training data. **Red** text indicates the best and blue text indicates the best performing state-of-the-art method.

Method	Training dataset	HDR-SYNTH	HDR-REAL	RAISE	HDR-EYE
AEO [2]	-	54.67 ± 9.22	48.47 ± 8.27	56.54 ± 4.00	49.20 ± 5.81
HPEO [14]	-	51.08 ± 8.10	46.29 ± 6.70	55.16 ± 5.40	44.12 ± 4.75
KOEO [16]	-	54.93 ± 7.56	50.19 ± 7.42	54.33 ± 4.72	49.62 ± 6.06
MEO [24]	-	55.02 ± 9.07	48.50 ± 8.33	56.30 ± 3.72	49.28 ± 6.09
HDRCNN [6]	Pre-trained model of [6]	56.94 ± 7.99	49.74 ± 7.82	57.08 ± 3.77	50.80 ± 5.79
HDRCNN* [6]	HDR-SYNTH	58.72 ± 6.57	50.02 ± 6.96	54.72 ± 3.84	50.57 ± 5.07
HDRCNN+ [6]	HDR-SYNTH + HDR-REAL	55.51 ± 6.64	51.38 ± 7.17	56.51 ± 4.33	51.08 ± 5.84
DrTMO [7]	Pre-trained model of [7]	57.15 ± 6.95	50.55 ± 7.25	57.69 ± 4.08	51.80 ± 5.93
DrTMO* [7]	HDR-SYNTH	56.74 ± 7.54	50.20 ± 7.57	57.65 ± 3.62	51.47 ± 5.88
DrTMO+ [7]	HDR-SYNTH + HDR-REAL	56.41 ± 7.20	50.77 ± 7.78	57.92 ± 3.69	51.26 ± 5.94
ExpandNet [23]	Pre-trained model of [23]	53.55 ± 4.98	48.67 ± 6.46	54.62 ± 1.99	50.43 ± 5.49
Deep chain HDRI [17]	Pre-trained model of [17]	-	-	-	49.80 ± 5.97
Deep recursive HDRI [18]	Pre-trained model of [18]	-	-	-	48.85 ± 4.91
Ours*	HDR-SYNTH	**60.11 ± 6.10**	51.59 ± 7.42	58.80 ± 3.91	52.66 ± 5.64
Ours+	HDR-SYNTH + HDR-REAL	59.52 ± 6.02	**53.16 ± 7.19**	**59.21 ± 3.68**	**53.16 ± 5.92**

Table 4: **Quantitative comparison on tone-mapped images with existing methods.** * represents that the model is re-trained on our training data. We generate tone-mapped HDR image with four local tone-mapping operators in Photomatix and measure the average PSNR, SSIM, and LPIPS scores. The proposed model performs well on all three datasets.

Method	HDR-SYNTH Dataset			HDR-REAL Dataset			RAISE Dataset			HDR-EYE Dataset		
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
AEO [2]	21.26	0.9010	0.1278	19.60	0.8680	0.1784	17.09	0.8058	0.2137	17.56	0.7201	0.2640
HPEO [14]	21.50	0.9134	0.1253	20.23	0.8827	0.1719	18.08	0.8272	0.2063	17.63	0.7416	0.2547
KOEO [16]	23.00	0.9220	0.1289	22.04	0.8953	0.1678	19.23	0.8270	0.2208	18.87	0.7528	0.2436
MEO [24]	21.48	0.9016	0.1286	19.65	0.8673	0.1781	17.51	0.8095	0.2119	16.03	0.7093	0.2702
HDRCNN [6]	22.64	0.9118	0.1168	20.30	0.8797	0.1651	17.47	0.8119	0.2092	16.92	0.7304	0.2494
HDRCNN* [6]	24.04	0.9257	0.1092	20.73	0.8604	0.1752	17.40	0.7794	0.2431	17.96	0.7391	0.2470
HDRCNN+ [6]	22.28	0.8925	0.1312	21.95	0.8974	0.1550	18.23	0.8246	0.2061	18.69	0.7551	0.2351
DrTMO [7]	23.25	0.9371	0.1291	21.43	0.9082	0.1812	21.48	0.8651	0.1617	21.60	0.8140	0.1550
DrTMO* [7]	23.06	0.9350	0.1385	21.14	0.9092	0.1854	21.65	0.8757	0.1516	22.15	0.8234	0.1598
DrTMO+ [7]	23.30	0.9332	0.1259	21.65	0.9068	0.1755	21.57	0.8593	0.1611	21.50	0.8092	0.1681
ExpandNet [23]	23.14	0.9241	0.1181	21.25	0.8875	0.1729	19.88	0.8418	0.1740	19.71	0.7759	0.2044
Ours	25.27	0.9333	**0.0948**	20.48	0.8802	0.1809	22.17	0.8839	0.1313	20.09	0.7870	0.2011
Ours+	**25.44**	**0.9449**	0.0987	**24.32**	**0.9248**	**0.1188**	**24.19**	**0.9037**	**0.1128**	**22.33**	**0.8270**	**0.1502**

We measure the execution time on an NVIDIA Tesla K80 GPU card with an image with a resolution of 512 × 512 pixels. Our Dequantization-Net, Linearization-Net, and Hallucination-Net take 66%, 9%, and 25% of execution time. As the Dequantization-Net does not contain any downsampling or pooling layers, all the convolutional layers are operated in the full-resolution space.

4. Additional Quantitative Evaluation and Analysis

4.1. Quantitative comparisons on HDR images

We compare the proposed model with conventional methods, AOE [2], HPEO [14], KOEO [16], MEO [24], and recent CNN-based models, HDRCNN [6], DrTMO [7], ExpandNet [23], Deep chain HDRI [17], and Deep recursive HDRI [18]. As the source code of HDRCNN and DrTMO is available, we also re-train their models on our HDR-SYNTH training set (denoted by $*$) and fine-tune on our HDR-REAL training set (denoted by $+$) for fair comparisons. Table 3 shows that the proposed model performs favorably against existing approaches on all four datasets.

4.2. Quantitative comparisons on tone-mapped images

We use the local (i.e., spatially varying) tone-mapping operators to generate the tone-mapped images and measure the PSNR, SSIM, LPIPS scores. Table 4 shows the average scores from four tone-mapping operators, e.g., Balanced, Smooth, Enhanced, and Soft, in Photomatix. The proposed model performs favorably against existing methods on all the datasets.

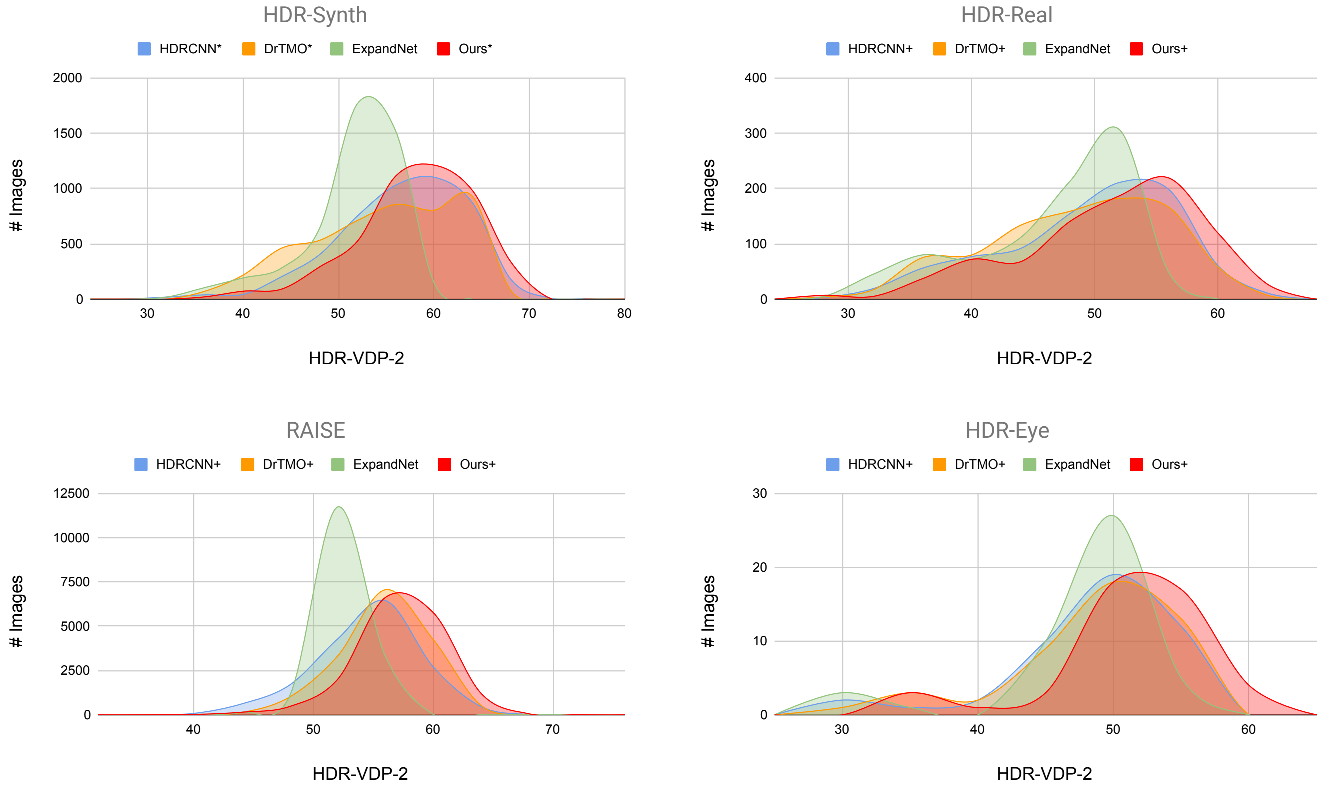


Figure 2: **Histogram of HDR-VDP-2 scores.** Ours results is distributed to the right (i.e., more images with higher scores) of the HDRCNN*[6], DrTMO*[7], and ExpandNet [23].

Table 5: **Effect of noise modeling.** We demonstrate that adding realistic Poisson-Gaussian noise [9] to the training data improves the generalization of our model on real LDR images (e.g., JPEG images produced by a real camera).

Dataset	HDR-Real	RAISE
w/o noise	49.22 ± 7.02	58.39 ± 3.53
Gaussian noise	49.49 ± 7.45	58.81 ± 3.46
Poisson-Gaussian noise (ours)	$\mathbf{49.66 \pm 7.50}$	$\mathbf{58.98 \pm 2.97}$

4.3. Histogram of HDR-VDP-2 scores

In Figure 2, we plot the distributions of the HDR-VDP-2 scores on the HDR-Synth, HDR-Real, RAISE, and HDR-Eye test sets for the HDRCNN*[6], DrTMO*[7], ExpandNet [23], and our model (Ours*). These plots clearly show that our results are distributed toward the right (i.e., more images with higher HDR-VDP-2 scores) of other approaches.

4.4. Effect of the noise model

To validate the effect of the realistic Poisson-Gaussian noise model [9], we further train the proposed model on the data synthesized with Gaussian noise and without noise. We evaluate the performance on the HDR-Real and RAISE test sets. Table 5 shows that the model trained with realistic noise performs well on both datasets. Note that the models in Table 5 are trained on the HDR-Synth training set and are not fine-tuned on the HDR-Real training set. The results also demonstrate that our model generalizes well to unseen LDR images generated from real camera pipelines.

4.5. Effect of edge and histogram features

Our Linearization-Net uses additional input features to utilize edge and histogram cues. To understand the effect of these input features on existing methods, we re-train the HDRCNN [6] and DrTMO [7] models by taking the LDR image, Sobel

Table 6: **Effect of additional input features used in the proposed Linearization-Net.** Directly adopting the edge and histogram features in existing models, HDRCNN [6] and DrTMO [7] only obtains a marginal improvement.

Method	HDR-Synth	HDR-Real
HDRCNN*	55.07 ± 7.63	49.57 ± 7.22
HDRCNN* + edge and histogram features	55.14 ± 7.59	49.61 ± 7.24
DrTMO*	53.82 ± 7.84	48.48 ± 7.52
DrTMO* + edge and histogram features	53.90 ± 7.93	48.31 ± 7.21
Ours	$\mathbf{57.10 \pm 7.32}$	$\mathbf{49.91 \pm 7.47}$

Table 7: **Comparison on linearization.** * represents that the model is re-trained on our training data. The proposed Linearization-Net reconstructs more accurate inverse CRFs than existing approaches.

Method	L2 error (↓) of inverse CRF	PSNR (↑) of linear image
Pre-defined x^2 [6]	11.64 ± 12.47	24.81 ± 6.47
Pre-defined $x^{2.2}$	9.06 ± 10.74	25.82 ± 6.04
Average inverse CRF	7.36 ± 7.03	25.24 ± 4.82
CRF-Net [19]	4.23 ± 4.37	30.61 ± 6.82
CRF-Net* [19]	2.71 ± 4.10	32.84 ± 6.85
Linearization-Net (ours)	$\mathbf{1.56 \pm 2.52}$	$\mathbf{34.64 \pm 6.73}$

edge response, and our soft histogram features as input. Table 6 shows that the HDRCNN and DrTMO obtain only marginal improvement with the soft histogram features. Instead, we adopt these features to estimate the inverse CRF, which have been shown effective for radiometric calibration [20]. This highlights the importance of our network designs for reversing the LDR image formation pipeline as incorporating these additional features to other models does not lead to substantial performance improvement.

4.6. Comparison with existing radiometric calibration approaches

Next, we compare the proposed Linearization-Net with existing approaches, including pre-defined curves $\mathcal{G} = x^2$ and $\mathcal{G} = x^{2.2}$, the average inverse CRF in the CRF dataset [12], and the CRF-Net [19]. We also re-train the CRF-Net on our training data for fair comparisons (marked with *). As shown in Table 7, the proposed Linearization-Net obtains the lowest reconstruction error of inverse CRFs as well as the highest PSNR values. More results of the reconstructed inverse CRFs and linear images are in the supplementary material.

4.7. Linearity of reconstructed HDR images

We measure the linearity of reconstructed HDR images on an LDR image with a color-checker (Macbeth chart) containing patches of known reflectance [3]. We then analyze the gray row of the color checker. Since there exists a scale ambiguity on the reconstructed HDR images, we normalize the intensity of the gray row to $[0, 1]$ and measure the coefficient of determination R^2 of each reconstructed result by fitting the intensity to the line $y = 1 - x$. Figure 3 shows the results of four different scenes. We note that the R^2 of the ground-truth is not a perfect 1 as there exists noise in the ground-truth image. Compared to other approaches, our model achieves a higher value of R^2, which demonstrates that our reconstructed result accurately captures the scene irradiance.

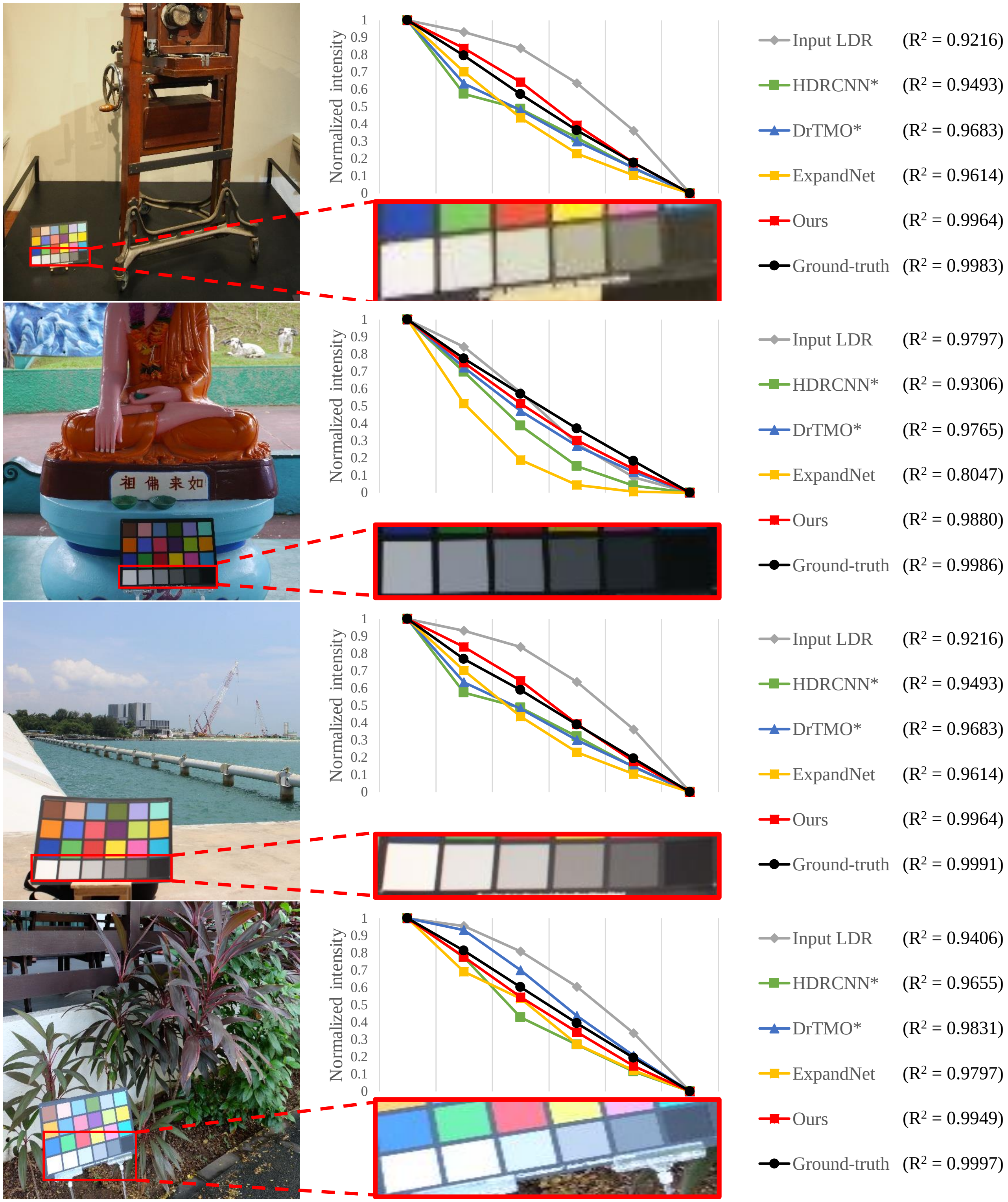
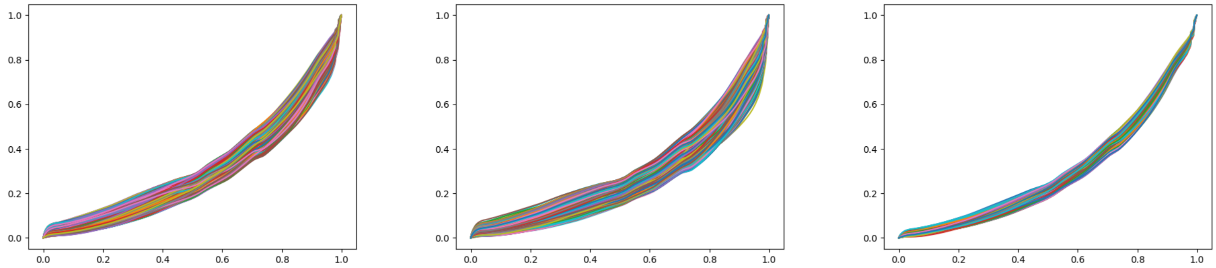


Figure 3: **Comparison on linearity.** Our model restores the gray row of the color checker well, which shows a better linearity than other approaches.

4.8. Consistency of the estimated CRFs

As the images in the RAISE dataset are mainly captured by three different cameras (Nikon D40, D90, and D7000), we can evaluate the consistency of the recovered inverse CRFs from the same camera. The results are shown in Figure 4 and Figure 5. Our method generates more consistent inverse-CRF prediction comparing to CRF-Net [19].

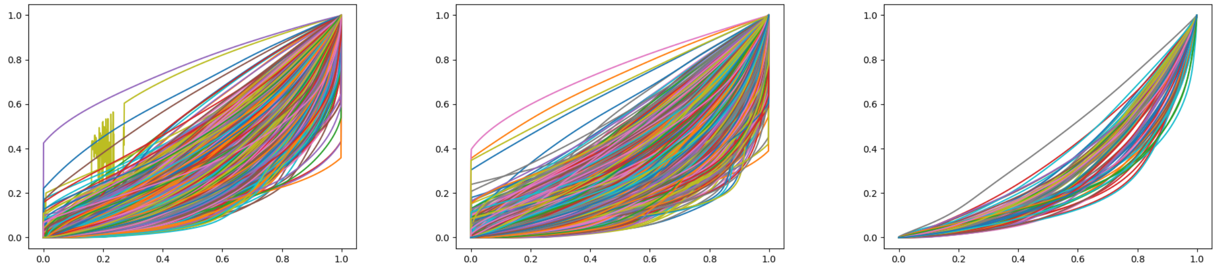


(a) Nikon D90 (2,276 images)

(b) Nikon D7000 (5,802 images)

(c) Nikon D40 (76 images)

Figure 4: **Estimated inverse CRFs by our method.**



(a) Nikon D90 (2,276 images)

(b) Nikon D7000 (5,802 images)

(c) Nikon D40 (76 images)

Figure 5: **Estimated inverse CRFs by CRF-Net [19].**

4.9. Evaluation on low-light images

Our methods are not designed specifically for handling noisy low-light images. We can still evaluate our method on low-light image dataset [29]. This evaluation would be helpful in understanding how well our method can handle large image noise. As shown in Figure 6, the tone-mapped images of our method still suffer from severe noises as we do not design any specific component for reducing camera noise.

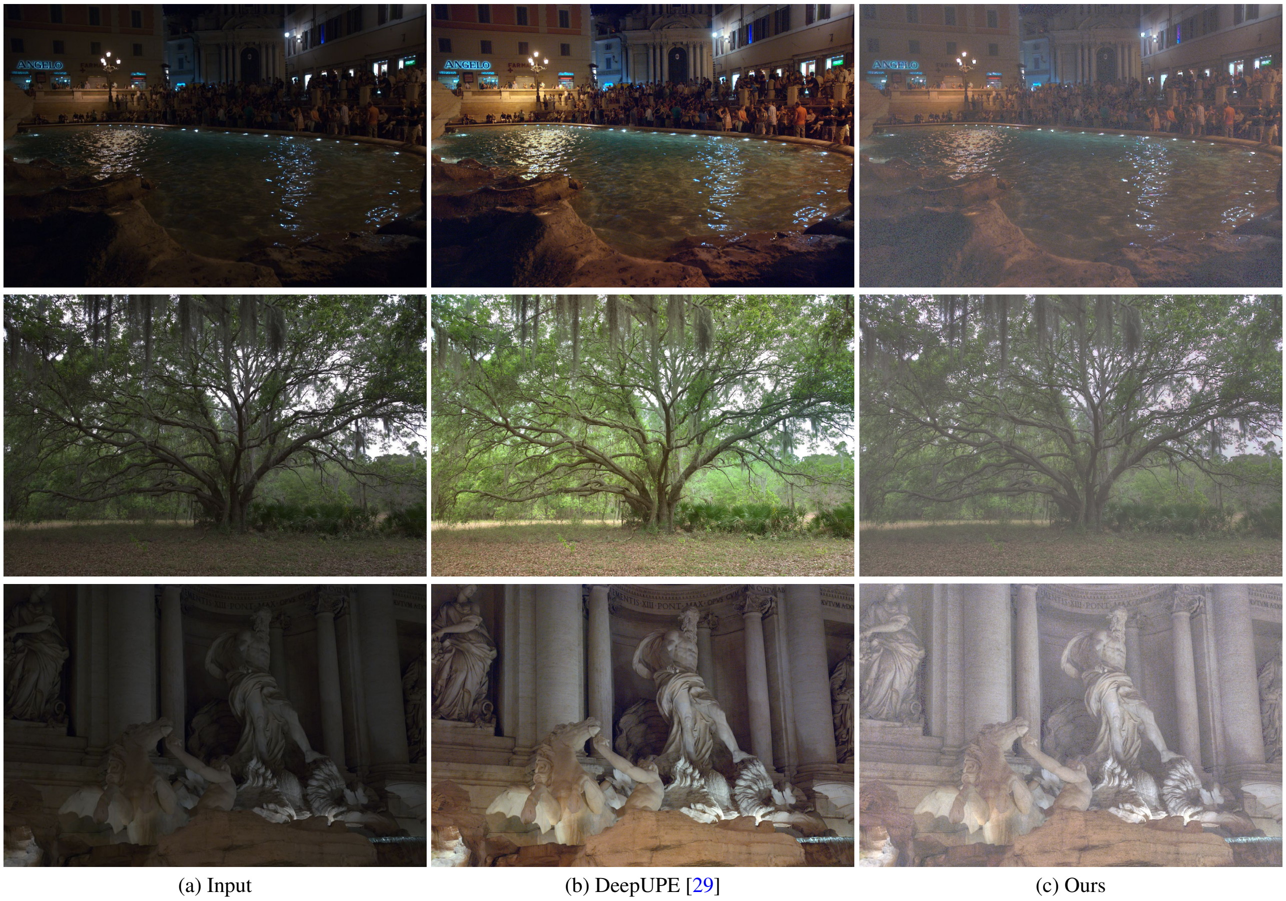

(a) Input (b) DeepUPE [29] (c) Ours

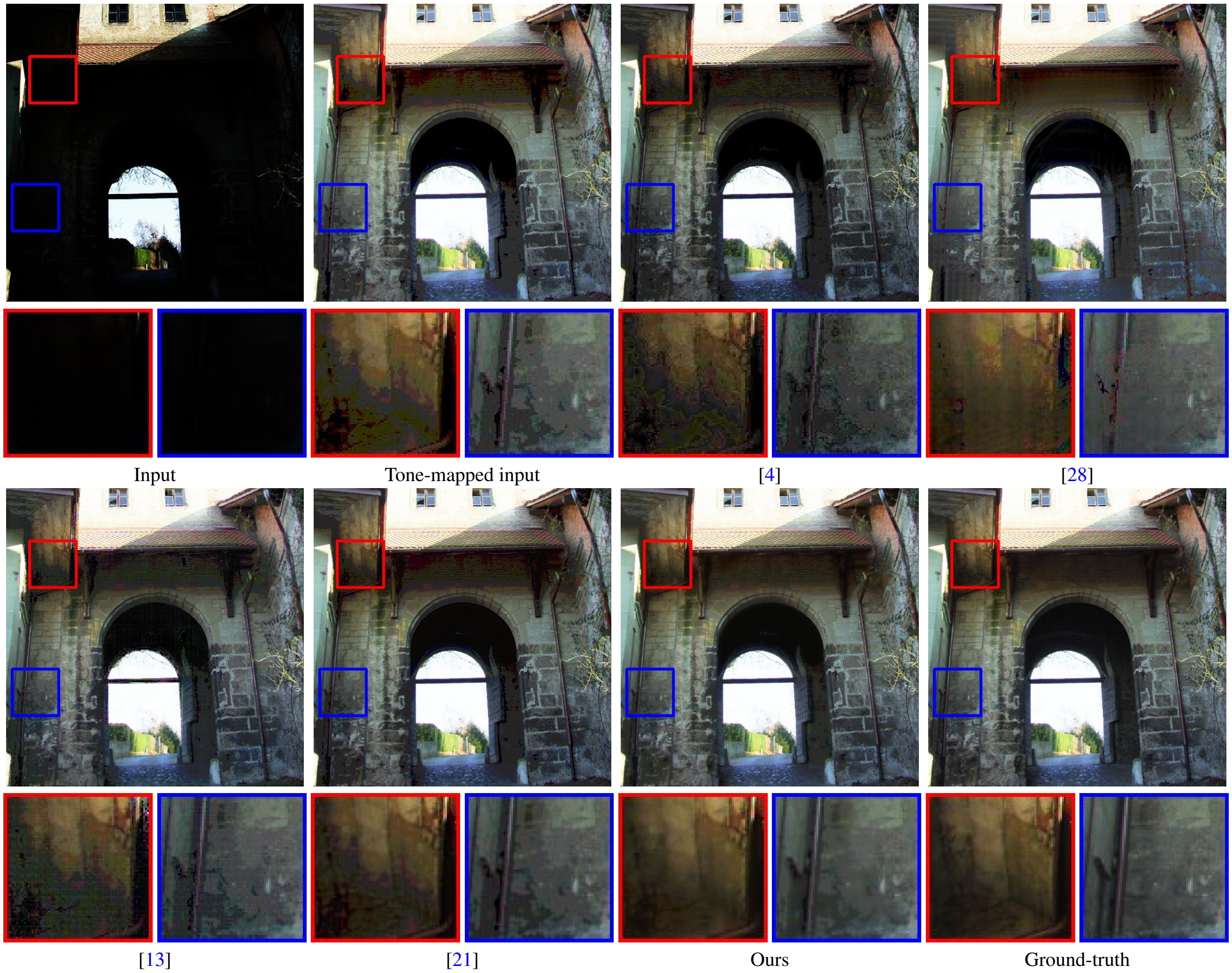
Figure 6: **Results on low-light images.**

5. Additional Visual Comparisons

We provide more visual results to compare the proposed Dequantization-Net, Linearization-Net, Hallucination-Net with the alternatives.

5.1. Dequantization

We show the visual comparisons of our Dequantization-Net with existing dequantization methods in Figure 7 and Figure 8. The contouring artifacts are clearly visible in the tone-mapped input LDR images. The results of [4, 13, 21] still contain the contouring artifacts. The filter-based method [28] makes the image too smooth. In contrast, our Dequantization-Net effectively reduces the contouring artifacts and scattered noise while maintaining the edges and structures of the input images.



Input

Tone-mapped input

[4]

[28]

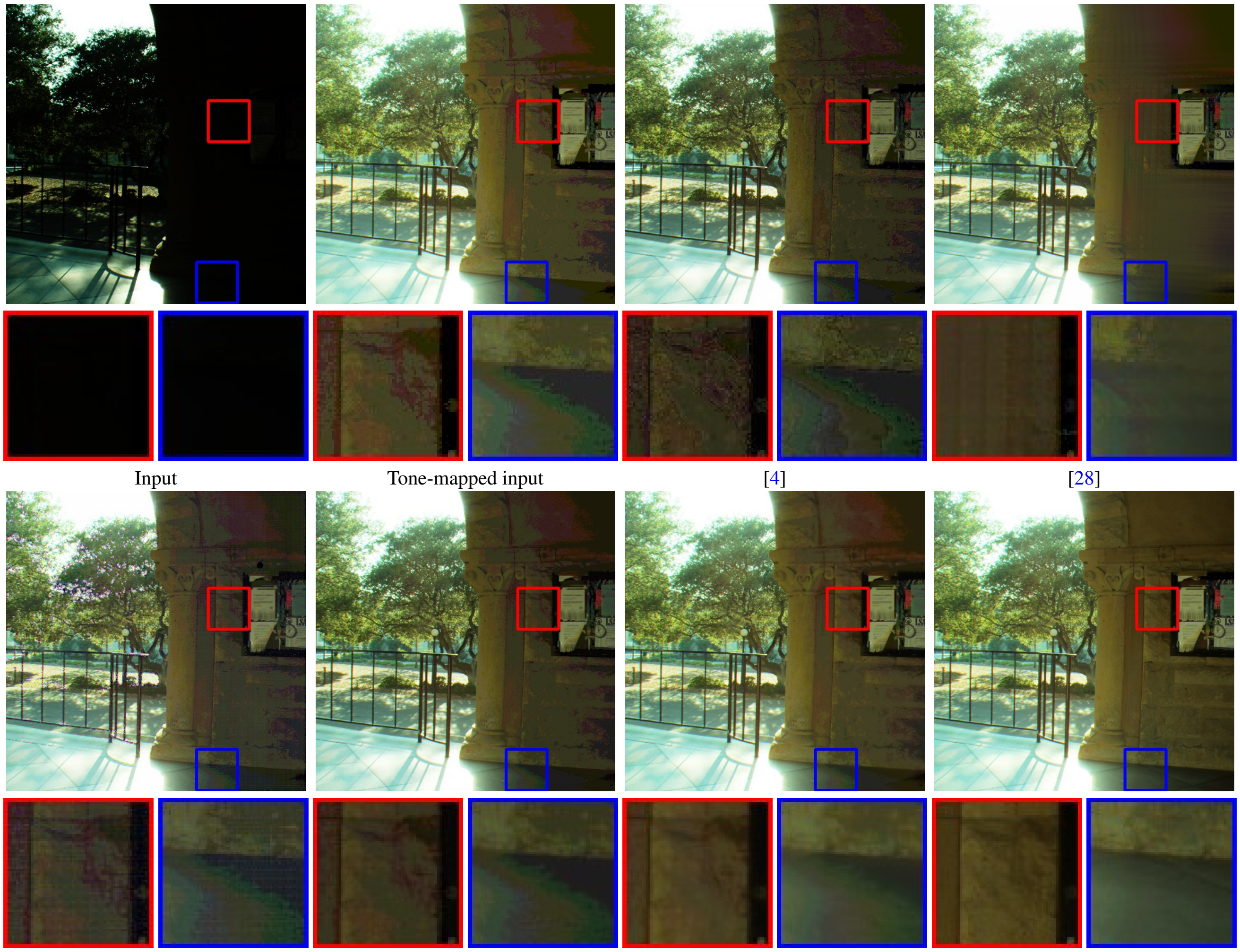
[13]

[21]

Ours

Ground-truth

Figure 7: **Effect of Dequantization.** Our Dequantization-Net effectively reduces the contouring artifacts and scattered noise while maintaining the edges and structures of the input images.



Input

Tone-mapped input

[4]

[28]

[13]

[21]

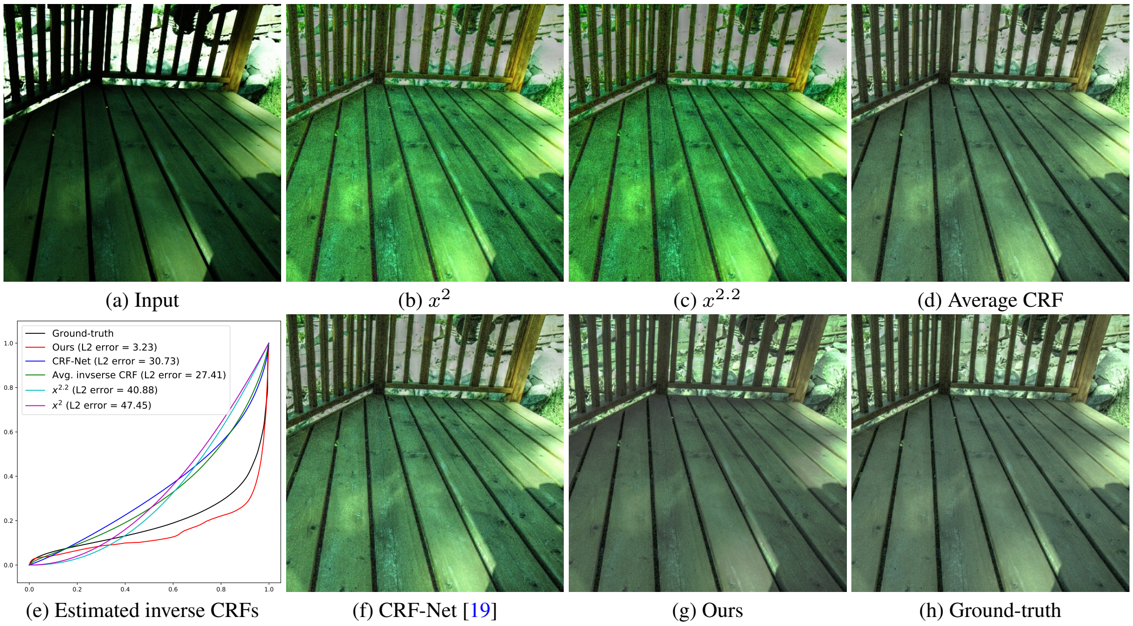
Ours

Ground-truth

Figure 8: **Effect of Dequantization.** Our Dequantization-Net effectively reduces the contouring artifacts and scattered noise while maintaining the edges and structures of the input images.

5.2. Linearization

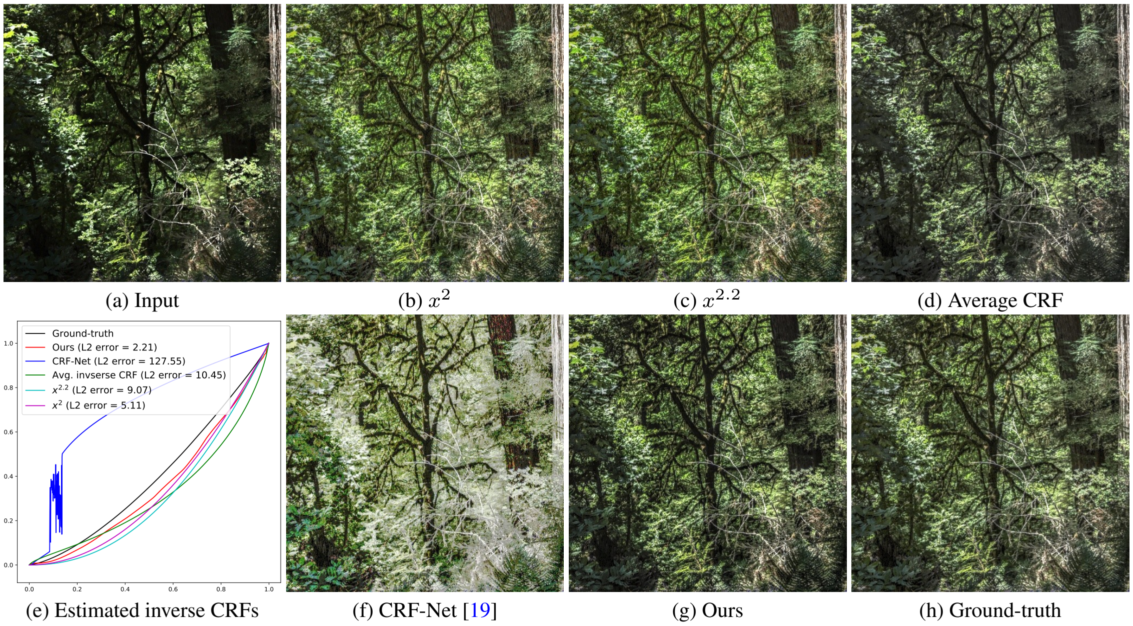
In Figure 9 and 10, we visualize the reconstructed inverse CRFs and the linear images. We note that the HDRCNN [6] uses a fixed $\mathcal{G} = x^2$ for any input images, which may not fit some uncommon curves (as shown in Figure 9). The prediction of the CRF-Net [19] is not guaranteed to be monotonically increasing (as shown in Figure 10) and, therefore, may not restore the correct linear irradiance of a scene. In contrast, the proposed Linearization-Net adopts the edge and histogram features as well as a monotonically increasing constraint to reconstruct more accurate CRFs, recovering the faithful color of the scene.



(a) Input (b) x^2 (c) $x^{2.2}$ (d) Average CRF

(e) Estimated inverse CRFs (f) CRF-Net [19] (g) Ours (h) Ground-truth

Figure 9: **Effect of Linearization.**



(a) Input (b) x^2 (c) $x^{2.2}$ (d) Average CRF

(e) Estimated inverse CRFs (f) CRF-Net [19] (g) Ours (h) Ground-truth

Figure 10: **Effect of Linearization.**

5.3. Hallucination

To reduce the checkerboard artifacts, we use the resize-convolution layers in the decoder of our Hallucination-Net instead of the transposed convolutional layers. Figure 11 demonstrates that the resize-convolution layers effectively alleviate the checkerboard artifacts exhibited in the results with the transposed convolutional layers.

In addition, we add the perceptual loss in the Hallucination-Net for synthesizing more realistic details. Figure 12 provides results without and with the perceptual loss and shows that the perceptual loss renders sharper details as clearly seen in the insets.

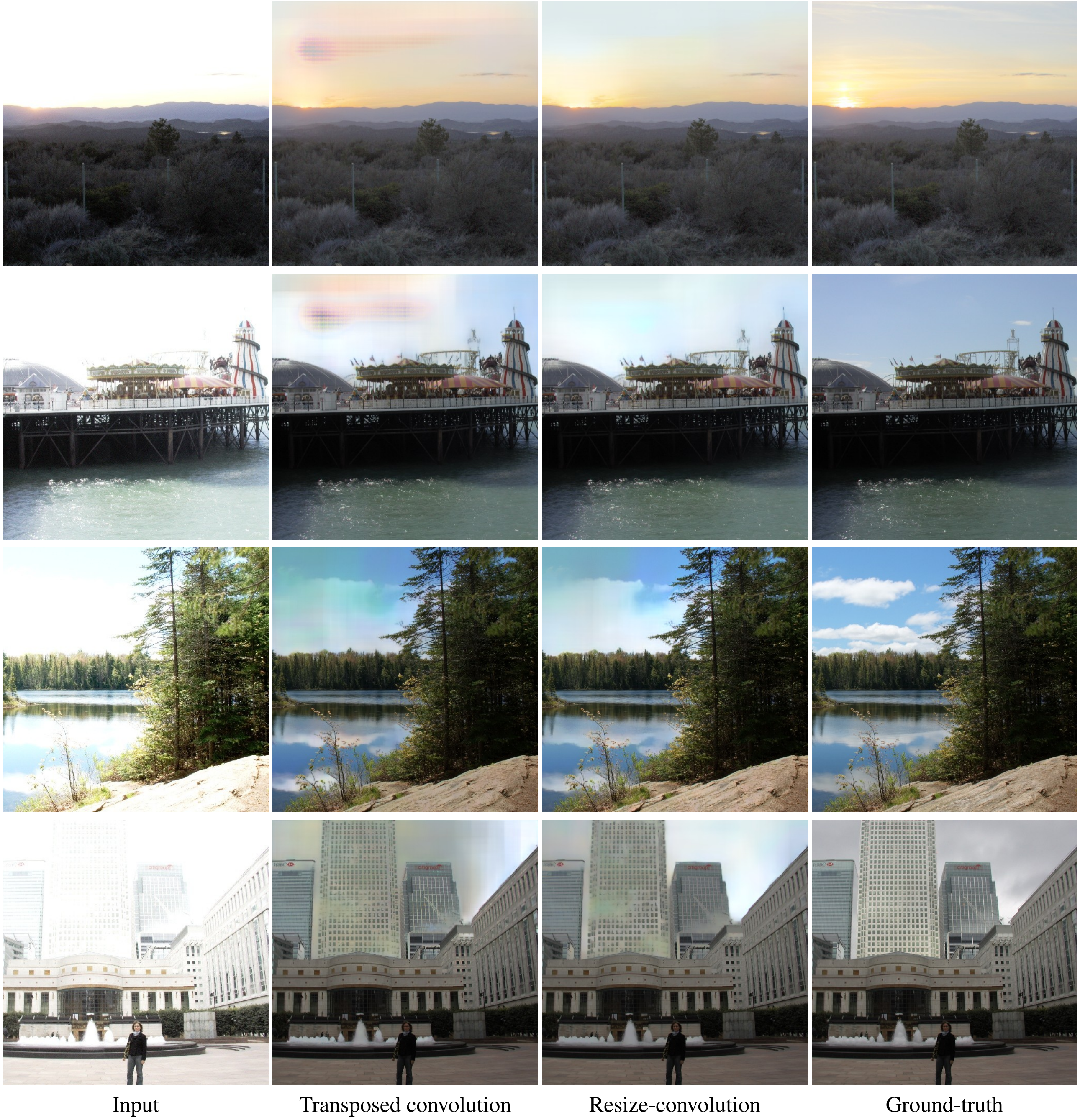


Figure 11: **Effect of resize-convolution.** The resize-convolution effectively reduce the checkerboard artifacts caused by the transposed convolution in the decoder of our Hallucination-Net.

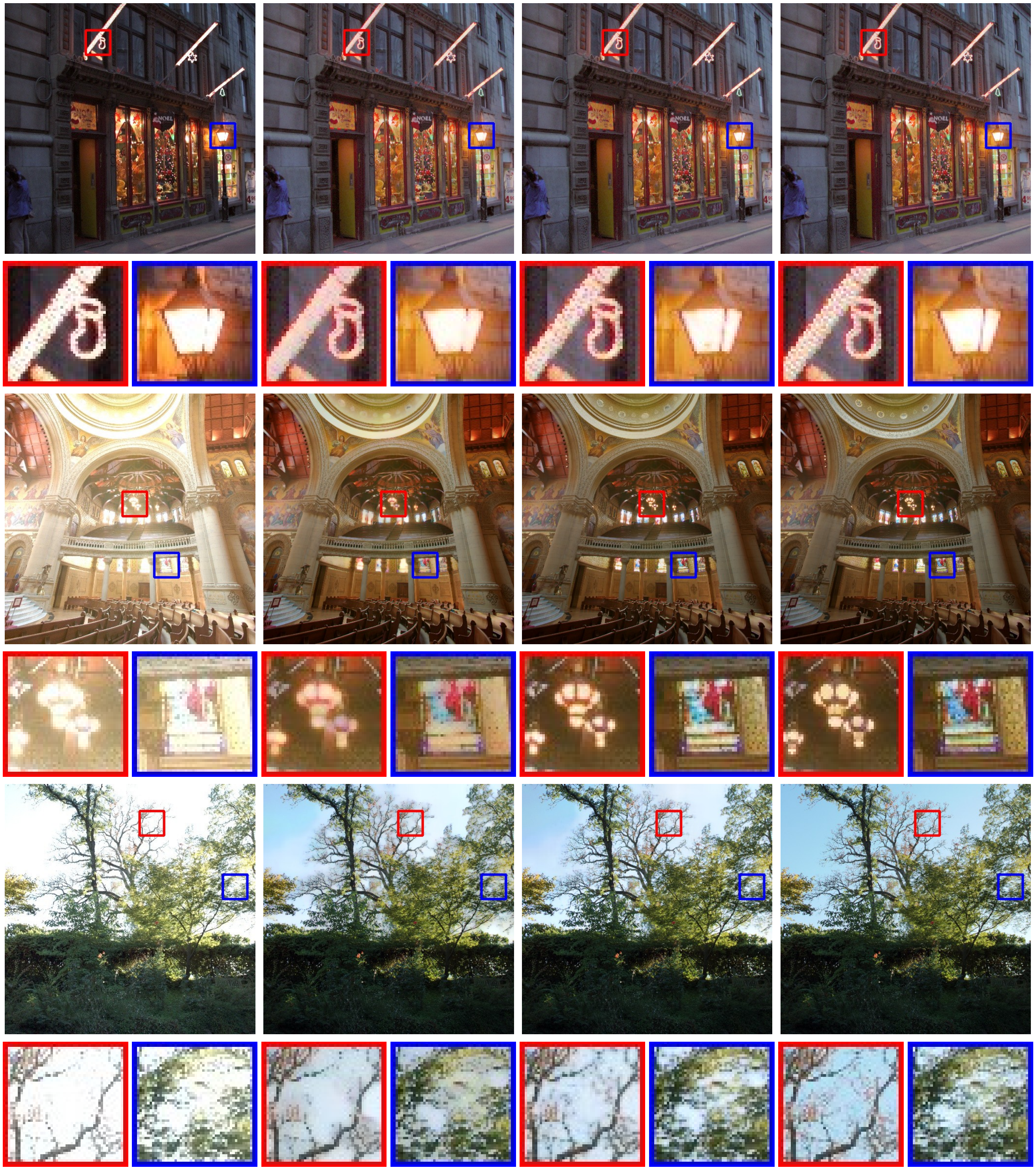
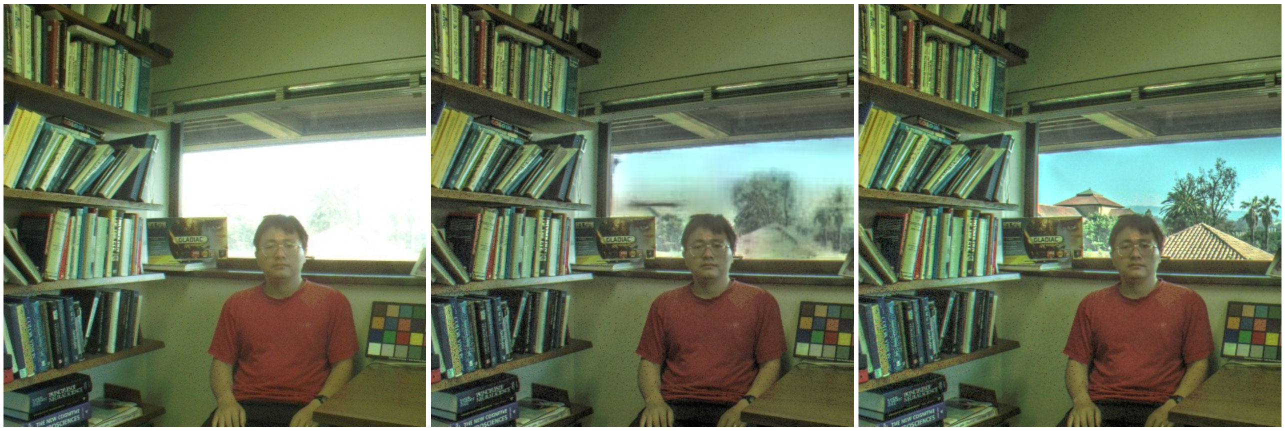


Figure 12: **Effect of perceptual loss.** By optimizing our Hallucination-Net with the perceptual loss, the reconstruction images looks sharper and contains more details.

6. Discussion and Future Work

We see several important future directions. First, our predictions in over-exposed regions may produce over-smoothed contents. As shown in Figure 13, existing algorithms typically fail to generate plausible content in such large saturated regions. Incorporating recent advancement in deep generative models [26] and image completion techniques [32, 22] in our model is a promising direction to address this issue. Second, while our work demonstrates the advantages of explicitly modeling the camera pipeline, our image formation model may be over-simplistic without careful modeling of chromatic aberration and image compression artifacts. We believe that more detailed and accurate modeling can further improve performance. Third, applying our method independently on each frame in a video is likely to result in temporal flicking artifacts. Addressing the temporal coherence issue can open up new opportunities in enhancing existing videos without using special cameras capturing alternating exposures.



(a) Input LDR

(b) Ours

(c) Ground truth HDR

Figure 13: **Failure case.** The scene outside the window is severely over-exposed. Existing methods and our model cannot reconstruct plausible content.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. 2

[2] Ahmet Oğuz Akyüz, Roland Fleming, Bernhard E Riecke, Erik Reinhard, and Heinrich H Bülthoff. Do hdr displays support ldr content?: A psychophysical evaluation. *ACM TOG*, 2007. 4

[3] Dongliang Cheng, Dilip K Prasad, and Michael S Brown. Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *JOSA A*, 2014. 6

[4] Scott J Daly and Xiaofan Feng. Decontouring: Prevention and removal of false contour artifacts. In *Human Vision and Electronic Imaging IX*, 2004. 10, 11

[5] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *ACM MM*, 2015. 2

[6] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K. Mantiuk, and Jonas Unger. HDR image reconstruction from a single exposure using deep CNNs. *ACM TOG*, 2017. 1, 3, 4, 5, 6, 12

[7] Yuki Endo, Yoshihiro Kanamori, and Jun Mitani. Deep reverse tone mapping. *ACM TOG*, 2017. 1, 3, 4, 5, 6

[8] Mark D. Fairchild. The HDR photographic survey. In *Color and Imaging Conference*, 2007. 1

[9] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *TIP*, 2008. 1, 5

[10] Brian Funt and Lilong Shi. The effect of exposure on MaxRGB color constancy. In *Human Vision and Electronic Imaging XV*, 2010. 1

[11] Brian Funt and Lilong Shi. The rehabilitation of MaxRGB. In *Color and Imaging Conference*, 2010. 1

[12] Michael D. Grossberg and Shree K. Nayar. What is the space of camera response functions? In *CVPR*, 2003. 1, 6

[13] Xianxu Hou and Guoping Qiu. Image companding and inverse halftoning using deep convolutional neural networks. *arXiv*, 2017. 10, 11

[14] Yongqing Huo, Fan Yang, Le Dong, and Vincent Brost. Physiological inverse tone mapping based on retina response. *The Visual Computer*, 2014. 4

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 2

[16] Rafael P Kovaleski and Manuel M Oliveira. High-quality reverse tone mapping for a wide range of exposures. In *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*, 2014. 4

[17] Siyeong Lee, Gwon Hwan An, and Suk-Ju Kang. Deep chain hdri: Reconstructing a high dynamic range image from a single low dynamic range image. *IEEE Access*, 2018. 4

[18] Siyeong Lee, Gwon Hwan An, and Suk-Ju Kang. Deep recursive hdri: Inverse tone mapping using generative adversarial networks. In *ECCV*, 2018. 4

[19] Han Li and Pieter Peers. Crf-net: Single image radiometric calibration using CNNs. In *European Conference on Visual Media Production*, 2017. 6, 8, 12

[20] Stephen Lin, Jinwei Gu, Shuntaro Yamazaki, and Heung-Yeung Shum. Radiometric calibration from a single image. In *CVPR*, 2004. 6

[21] Chang Liu, Xiaolin Wu, and Xiao Shu. Learning-based dequantization for image restoration against extremely poor illumination. *arXiv*, 2018. 10, 11

[22] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018. 15

[23] Demetris Marnerides, Thomas Bashford-Rogers, Jonathan Hatchett, and Kurt Debattista. ExpandNet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content. In *EG*, 2018. 3, 4, 5

[24] Belen Masia, Ana Serrano, and Diego Gutierrez. Dynamic range expansion based on image statistics. *Multimedia Tools and Applications*, 2017. 4

[25] Hiromi Nemoto, Pavel Korshunov, Philippe Hanhart, and Touradj Ebrahimi. Visual attention in ldr and hdr images. In *9th International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*, 2015. 2

[26] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 15

[27] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High Dynamic Range Imaging: Acquisition, Display, and Image-based Lighting*. Morgan Kaufmann, 2010. 1

[28] Qing Song, Guan-Ming Su, and Pamela C Cosman. Hardware-efficient debanding and visual enhancement filter for inverse tone mapped high dynamic range images and videos. In *ICIP*, 2016. 10, 11

[29] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *CVPR*, 2019. 9

[30] Greg Ward. High dynamic range image encodings. 2006. 1

[31] Feng Xiao, Jeffrey M. DiCarlo, Peter B. Catrysse, and Brian A. Wandell. High dynamic range imaging of natural scenes. In *Color*

and Imaging Conference, 2002. 1

[32] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *ICCV*, 2019. 15