DNF: Decouple and Feedback Network for Seeing in the Dark
— *Supplementary Materials* —

Xin Jin[1*] Ling-Hao Han[1*] Zhen Li[1] Chun-Le Guo[1†] Zhi Chai[2] Chongyi Li[3]
[1]VCIP, CS, Nankai University [2]Hisilicon Technologies Co. Ltd.
[3]S-Lab, Nanyang Technological University
{xjin, lhhan}@mail.nankai.edu.cn, zhenli1031@gmail.com, guochunle@nankai.edu.cn,
chaizhi2@huawei.com, chongyi.li@ntu.edu.sg
https://github.com/Srameo/DNF

1. Architecture and Training Details

Architecture Details. We illustrate the detailed network architecture of the proposed DNF in Fig. 1. Our network essentially consists of two UNet-style [9] subnetworks. One subnetwork, which targets to output a denoised RAW image Y_{raw}, contains a RAW encoder E_{raw} and an RAW decoder D_{raw}. The other one, which is responsible for recovering the final sRGB image Y_{rgb}, contains an RAW encoder E'_{raw} equipped with gated fusion modules (GFMs) and an sRGB decoder. Noted that, the local residual shortcuts of the CID block are switched off in E_{raw}, however, switched on in E'_{raw}. Except for GFMs and the shortcuts, the parameters of E'_{raw} are shared with E_{raw}. The connections between the two subnetworks are built by L GFMs for handling multi-stage feedback features. The RAW encoder E_{raw} and decoder D_{raw} both contain L channel-independent denoising (CID) blocks. And the sRGB decoder D_{rgb} is build up with L Matrixed Color Correction (MCC) blocks at different scales. Here, we set L as 4 according to the number of stages. For each stage in the RAW encoder, we use a convolution with the kernel size of 2 and the stride of 2 to perform downsampling. Moreover, we use a transposed convolution for upsampling the feature at the end of each stage in the decoders.

Training Details. The DNF is implemented in PyTorch framework [8]. For all experiments, the model is created with the default parameter initialization and optimized by the Adam optimizer [5] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for 500 epochs, where no weight decay is used. The learning rate will vary with iterations like SGDR [7]. The initial learning rate is set to 2×10^{-4} and the minimum learning rate is set to 2×10^{-5} without restart. During training, we randomly crop the input image with a patch size of 512×512. And the batch size is 1 for the SID [1] dataset and 4 for the MCR [2] dataset. We respectively use one and four NVIDIA GeForce RTX 2080Ti GPUs for training on the SID dataset and MCR dataset. The data augmentations are the same as SID [1], *i.e.*, random flipping and rotation. During inference, the test image is fed into the network with the full resolution without any data augmentation strategy. The code implemented by MindSpore framework is also provided.

2. More Visual Results

In this section, we provide additional visual results on two benchmarks, MCR [2] and SID [1] datasets, to further show the superiority of our proposed DNF. The restoration results of SID [1], SGN [3], RRT [4], EEMEFN [10], LDC [6] and MCR [2] are presented for comparison. As shown in Fig. 2-6, our proposed DNF is able to reconstruct more satisfactory results with less color shifts and clear details. The corresponding PSNR, which also provides quantitative comparisons, is presented in figures. Also, more qualitative results on our proposed residual switch mechanism (RSM) (Sec.3.2 in main paper) can be found in Fig. **??**. The results shows that our RSM enables our encoder to generates noise features during RAW denoising stage, however signal features when color restoration stage.

3. Broader Impacts

The proposed DNF framework redesigns the low-light image enhancement pipeline exclusively for the RAW data. With the great performance achieved, it makes possible for the real-world application to perform the fast and accurate enhancement. Because of the progressive pipeline and the explicit utilization of the domain-specific properties, the generalized framework could be applied to mobile devices and embedded systems in an end-to-end manner, directly from the sensor to the screen. It worth to be noted that the enhanced low-light imaging poses the potential risks of violating personal privacy.

*Equal contribution.
†C. L. Guo is the corresponding author.

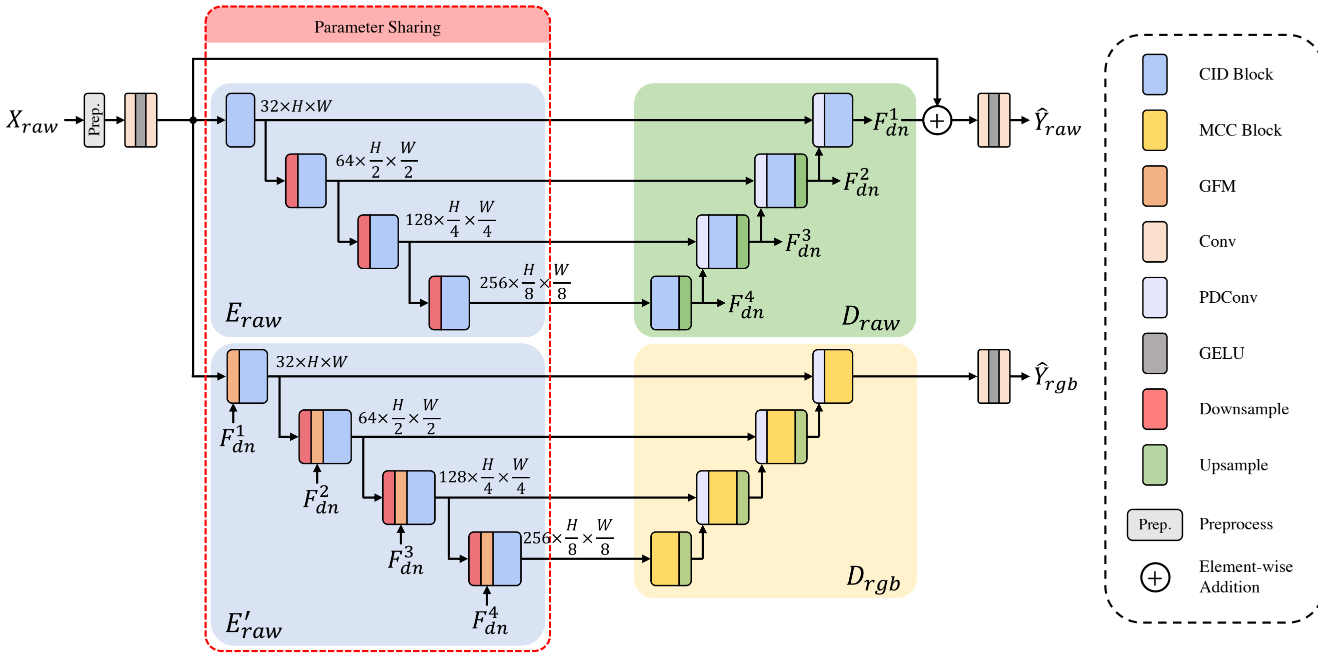

Figure 1. Detailed architecture for our proposed DNF. The X_{raw} denotes the input low-light RAW image. F^i_{dn} means the denoising feature extracted from the channel-independent denoising (CID) block in the i-th stage. $\hat{Y}_{raw}$ and $\hat{Y}_{rgb}$ represent an output RAW image as well as an output sRGB image, respectively. A point-wise convolution followed by a depth-wise convolution is denoted as `PDConv` in the legend. The $\hat{C} \times \hat{H} \times \hat{W}$ formatted expression on the arrow indicates the feature size for the corresponding stage. $H \times W$ is the input resolution. $E_{raw}, D_{raw}, D_{rgb}$ represents RAW encoder, RAW decoder, and sRGB decoder, respectively. The RAW encoder equipped with four GFMs is denoted as E'_{raw}. Also, the local residual shortcut of the CID block is switched on in the E'_{raw}, however, switched off in the E_{raw}.

References

[1] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *CVPR*, 2018. 1, 4, 5, 6, 7

[2] Xingbo Dong, Wanyan Xu, Zhihui Miao, Lan Ma, Chao Zhang, Jiewen Yang, Zhe Jin, Andrew Beng Jin Teoh, and Jiajun Shen. Abandoning the bayer-filter to see in the dark. In *CVPR*, 2022. 1, 3, 6, 7

[3] Shuhang Gu, Yawei Li, Luc Van Gool, and Radu Timofte. Self-guided network for fast image denoising. In *ICCV*, 2019. 1, 4, 5, 6, 7

[4] Haofeng Huang, Wenhan Yang, Yueyu Hu, Jiaying Liu, and Ling-Yu Duan. Towards low light enhancement with raw images. *IEEE TIP*, 2022. 1, 4, 5

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 1

[6] Chongyi Li, Chunle Guo, and Change Loy Chen. Learning to enhance low-light image via zero-reference deep curve estimation. *TPAMI*, 2021. 1, 4, 5, 6, 7

[7] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *ICLR*, 2016. 1

[8] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Workshops*, 2017. 1

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *ICML*, 2015. 1

[10] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. EEMEFN: Low-light image enhancement via edge-enhanced multi-exposure fusion network. In *AAAI*, 2020. 1, 4, 5, 6, 7

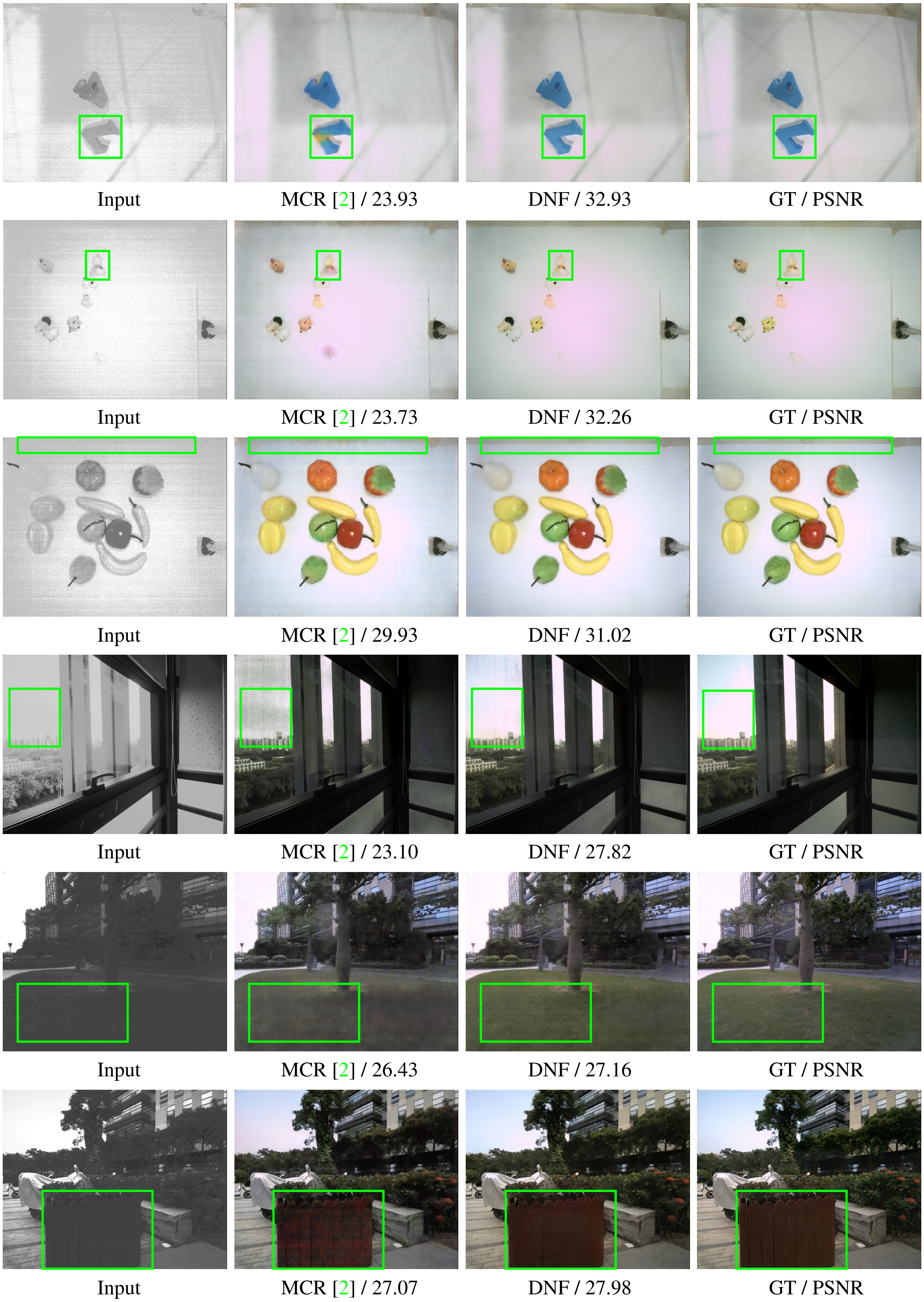

Figure 2. Visual results comparisons on the MCR [2] Dataset (*Zoom-in for best view*). We amplified the input with a pre-defined amplification ratio.

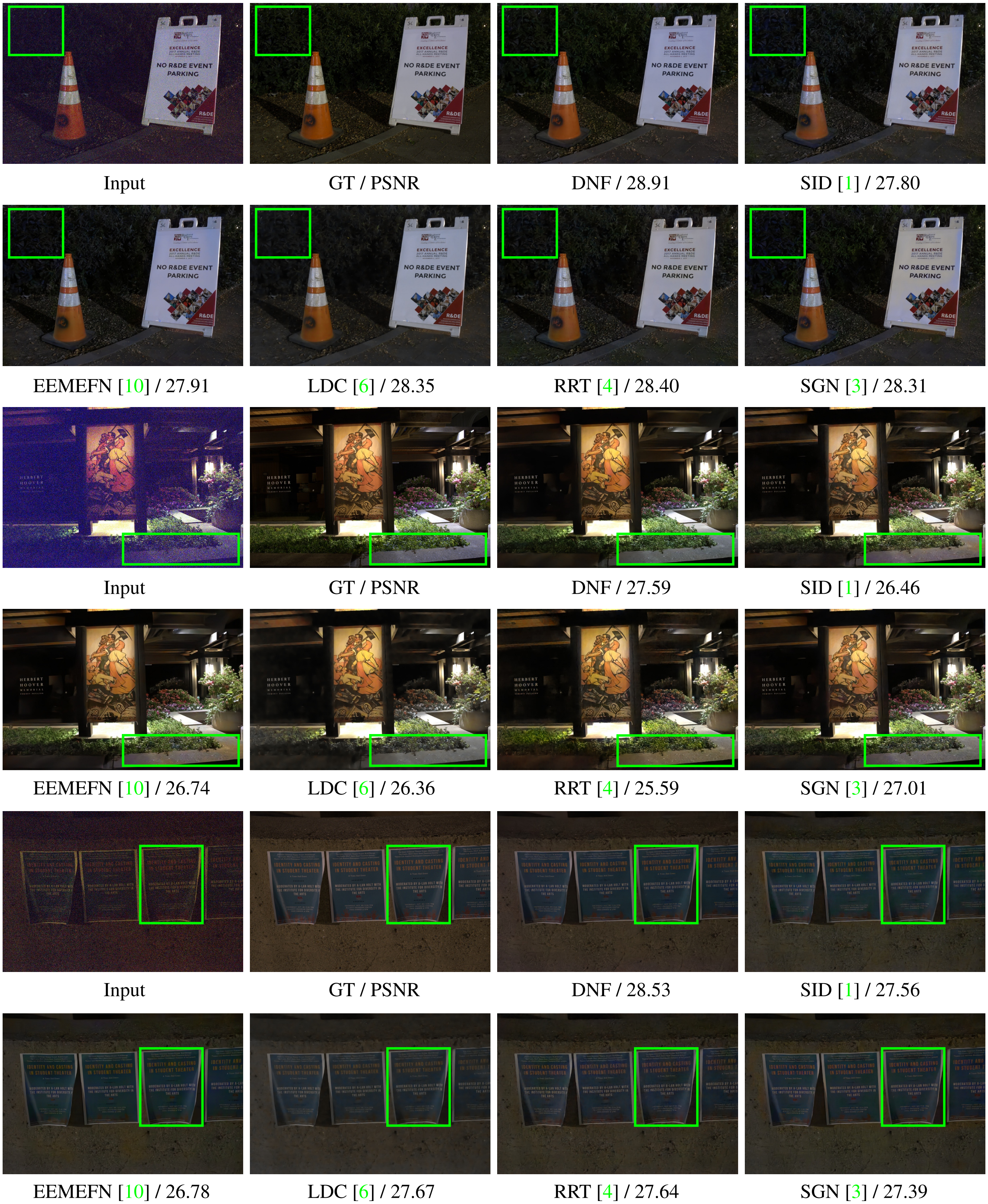

Figure 3. Visual results comparisons on the SID [1] Fuji Dataset (*Zoom-in for best view*). We amplified and post-processed the input images with an ISP for visualization like [1].

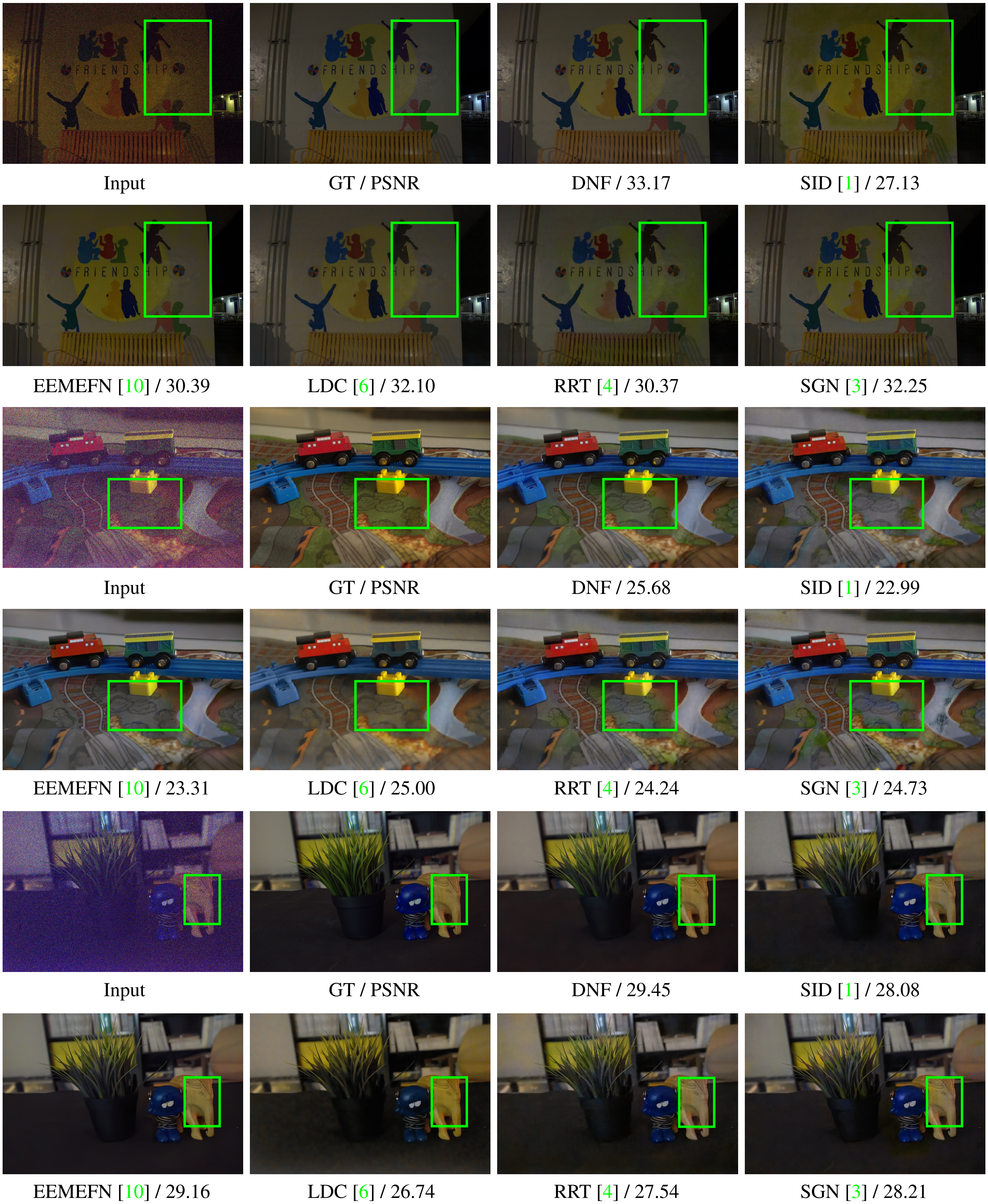

Figure 4. Visual results comparisons on the SID [1] Fuji Dataset (*Zoom-in for best view*). We amplified and post-processed the input images with an ISP for visualization like [1].

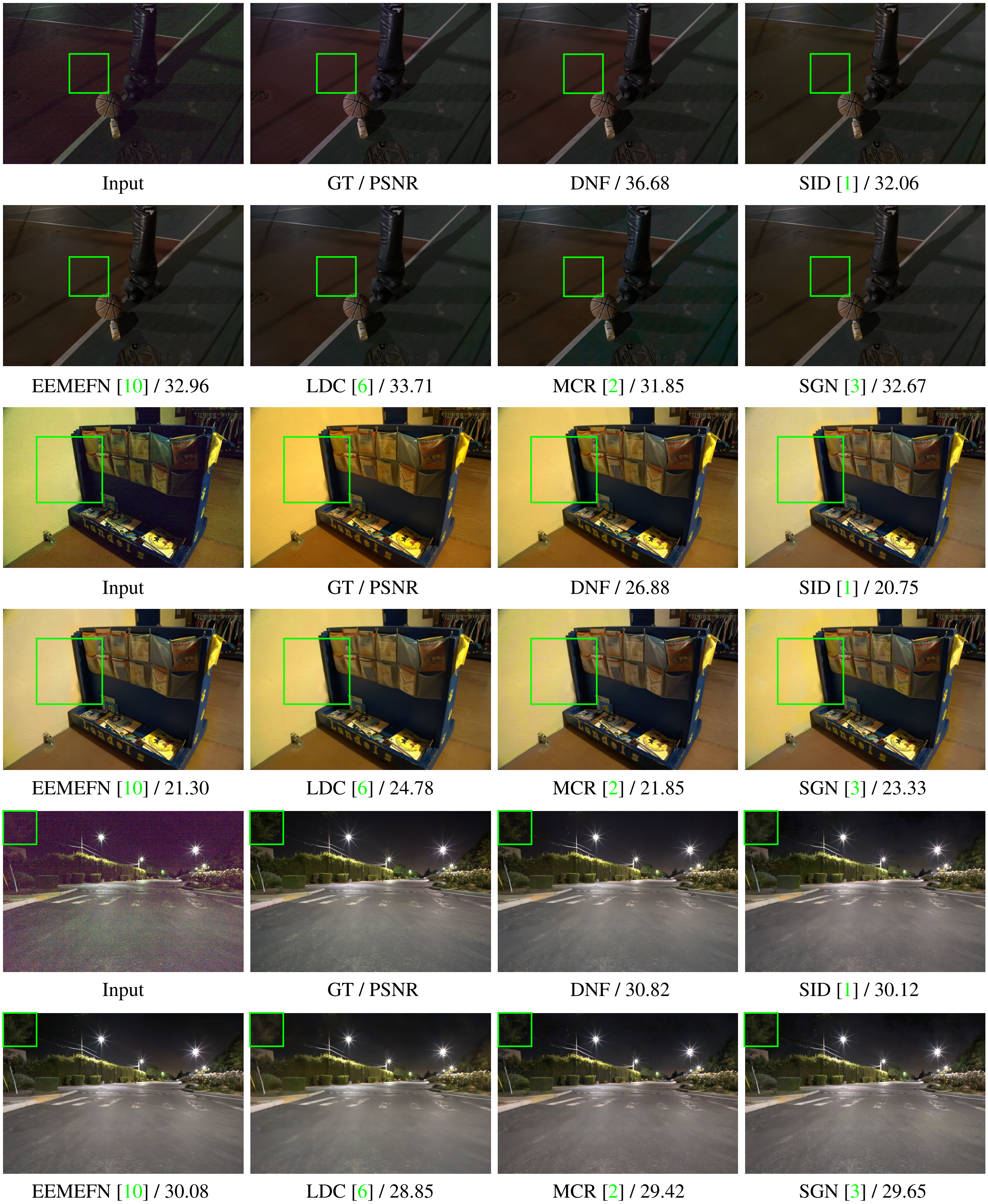

Figure 5. Visual results comparisons on the SID [1] Sony Dataset (*Zoom-in for best view*). We amplified and post-processed the input images with an ISP for visualization like [1].

Figure 6. Visual results comparisons on the SID [1] Sony Dataset (*Zoom-in for best view*). We amplified and post-processed the input images with an ISP for visualization like [1].