Supplementary Materials of "Fast and High Quality Image Denoising via Malleable Convolution"

Yifan Jiang*[1], Bartlomiej Wronski[2], Ben Mildenhall[2], Jonathan T. Barron[2], Zhangyang Wang[1], and Tianfan Xue[2]

[1] University of Texas at Austin
[2] Google Research

1 Architecture and Training Details

The proposed MalleNet-families include MalleNet-S, -M, -L, and -Xl. We start from constructing MalleNet-S. MalleNet-S consists of four levels, the top two levels use 16-channel intermediate features and bottom two levels use 32-channel intermediate features. We adopt Inverse Bottleneck Block [3] as the basic operator for each level, whose detailed structure is shown in Fig. 2 right and we use a fixed expansion ratio of 3 in the depth-wise convolution. We use two blocks for each of the top two levels and six blocks for each of the bottom two levels, the and inject one 3×3 MalleConv layer in the middle of each level.

To construct MalleNet-M, -L, -XL, we simply modify MalleNet-S by growing the number of channels of each stage from {16, 16, 32, 32} to {32, 32, 64, 64}, {64, 64, 128, 128}, {144, 144, 288, 288} respectively. For MalleNet-XL, we further increase the expansion ratio from 3 to 5 to increase its capability. For real-world noise benchmark SIDD dataset, we construct MalleNet-R by adjusting the base channel number to 64 and adopt ResNet Block as basic operator (as shown in Fig. 2 left).

To better illustrate how to adopt MalleConv on existing popular backbones, we describe the detailed injecting position of MalleConv in the ablation study on Sec. 5.2 of our main manuscripts. Concretely speaking, we inject MalleConv in the middle layer of DnCNN [4] and UNet, and the last layer of Residual Dense Block (RDN). Furthermore, we present the details of the proposed efficient predictor network, as shown in Fig. 3. The predictor network is constructed with several stacked ResNet Blocks and MaxPooling layer. We apply the standard ResNet blocks for all variants of MalleNet-families.

To mitigate training instability, we will optimize the static convolutional kernel only for the first 10 iterations on each backbone, and then include the Malleable Convolution (dynamic kernel) in the forward and backward process. This is observed to avoid NaN loss in most cases.

* This work was performed while Yifan Jiang worked at Google.

2 Visual Comparison

We include the visual comparisons between the proposed MalleNet and previous state-of-the-art approaches on both simulated benchmarks and real-world benchmark (SIDD [1]), as shown in Fig. 4 and Fig. 5. Since SIDD only provide validation set with 256×256 patches as noisy/gt paired data, we report the PSNR/SSIM and visual results on validation set. As shown in Fig. 4, we randomly pick several visual examples generated from MalleNet-R and the best competitor HINet [2], and observe similar performance between these two approaches, although quantitative score provided by MalleNet-R slightly behinds the HINet. This might because the diminishing marginal utility of PSNR when it reaches an enough high value. Moreover, regarding to the efficiency, MalleNet-R saves up to $\times$**2.42** runtime latency and $\times$**5.86** FLOPs costs.

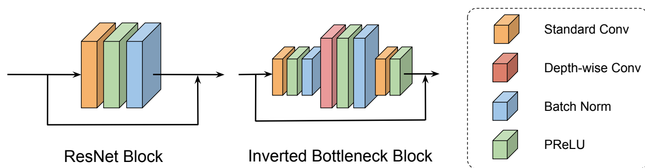

Fig. 1. Detailed illustration of "ResNet Block" and "Inverted Bottleneck Block" used in the proposed MalleNet architecture.

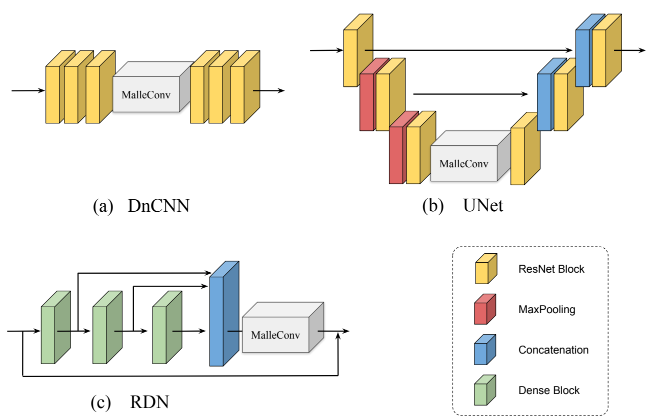

Fig. 2. Detailed injected position of Malleable Convolution on different backbones.

References

1. Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. *CVPR*, 2018.

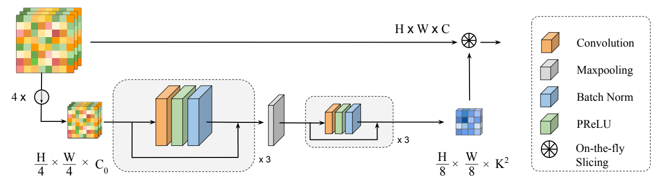

Fig. 3. Detailed architecture of the proposed predictor network.

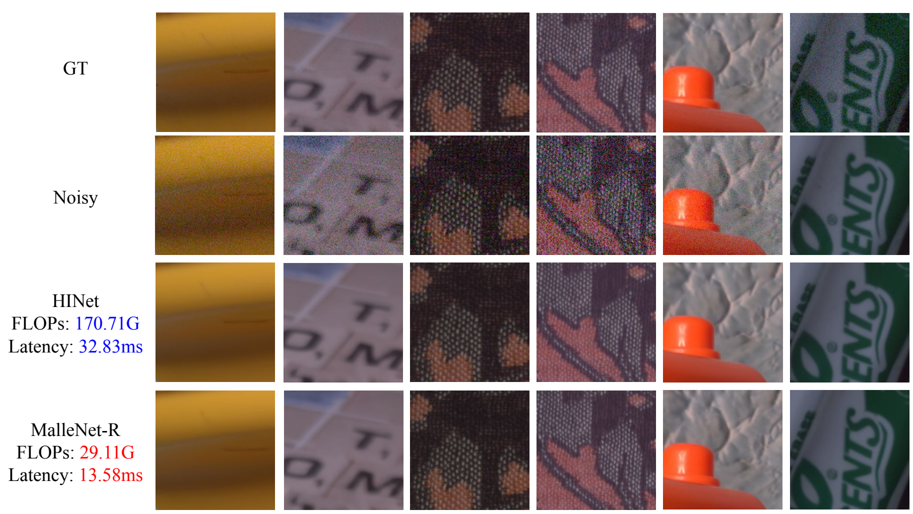

Fig. 4. Randomly picked testing patches from MalleNet and previous state-of-the-art methods on real-world noise benchmark (SIDD [1]). **Best viewed in color and zoomed in.**

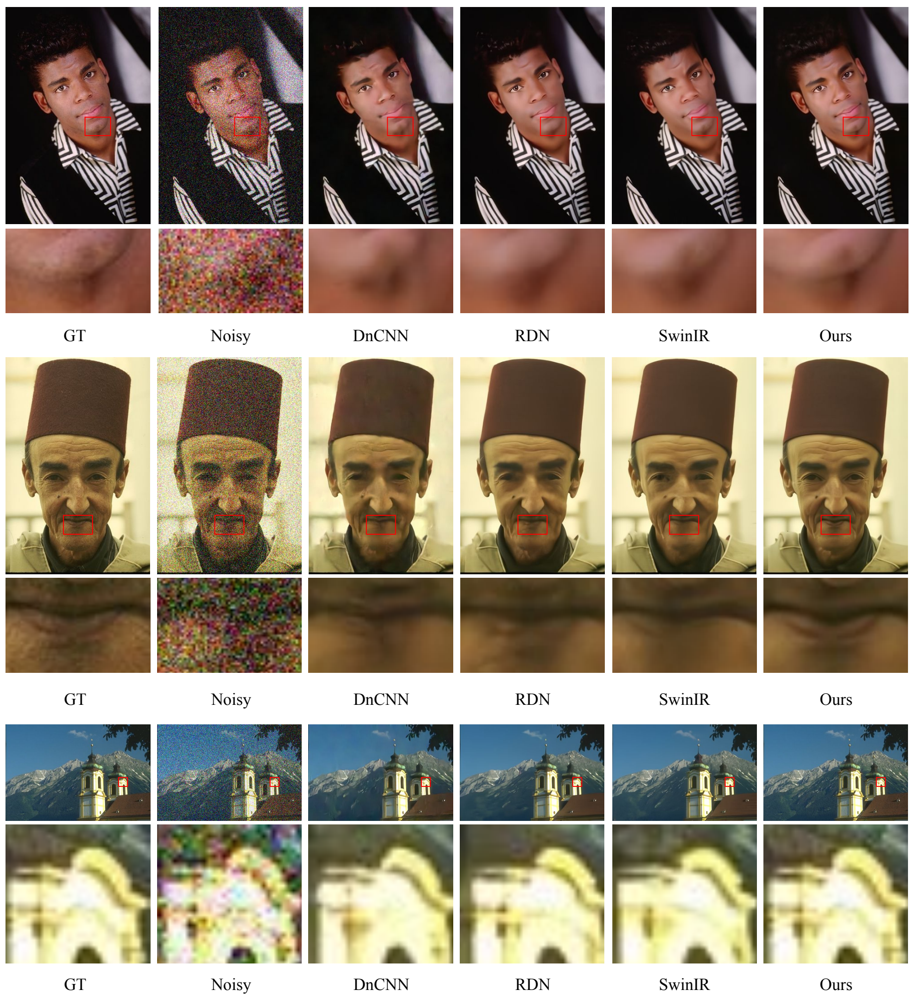

Fig. 5. Visual Comparison between MalleNet and previous state-of-the-art methods on simulated dataset ($\sigma = 50$).

2. Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. *CVPR*, 2021.
3. Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *CVPR*, 2018.
4. Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 2017.