Supplementary Material of Simple Baselines for Image Restoration

Liangyu Chen*, Xiaojie Chu*, Xiangyu Zhang, and Jian Sun

MEGVII Technology, Beijing, CN
{chenliangyu,chuxiaojie,zhangxiangyu,sunjian}@megvii.com

In this document, we provide additional details and qualitative results of the baseline and NAFNet.

1 Other Details

1.1 Inverted Bottleneck

Following [4] we adopt inverted bottleneck design in the baseline and NAFNet. In the baseline, the channel width within the first skip connection is always consistent with the input, its computational cost could be approximated by:

$$H \times W \times c \times c + H \times W \times c \times k \times k + H \times W \times c \times c, \tag{1}$$

where H, W represent the spatial size of the feature map, c indicates the input dimension, and k is the kernel size of the depthwise convolution (3 in our experiments). In practice, $c \gg k \times k$, thus Eqn. (1) $\approx 2 \times H \times W \times c \times c$. The hidden dimension within the second skip connection is twice the input dimension, its computational cost is:

$$H \times W \times c \times 2c + H \times W \times 2c \times c, \tag{2}$$

notations following Eqn. (1). As a result, the overall computational cost of one baseline block $\approx 6 \times H \times W \times c \times c$.

As for NAFNet's block, the SimpleGate module shrinks the channel width by half. We double the hidden dimension in the first skip connection, and its computational cost could be approximated by:

$$H \times W \times c \times 2c + H \times W \times 2c \times k \times k + H \times W \times c \times c, \tag{3}$$

notations following Eqn. (1). And the hidden dimension in the second skip connection follows baseline. Its computational cost is:

$$H \times W \times c \times 2c + H \times W \times c \times c. \tag{4}$$

As a result, the overall computational cost of one NAFNet's block $\approx 6 \times H \times W \times c \times c$, which is consistent with the baseline's block. The advantage of this is that the baseline and NAFNet can share hyperparameters, such as the number of blocks, learning rate, etc.

It should be noted that the above discussion omits the computation of some modules, e.g. layer normalization, GELU, channel attention, and etc., as their computational cost is negligible compared to convolution.

* Equally contribution.

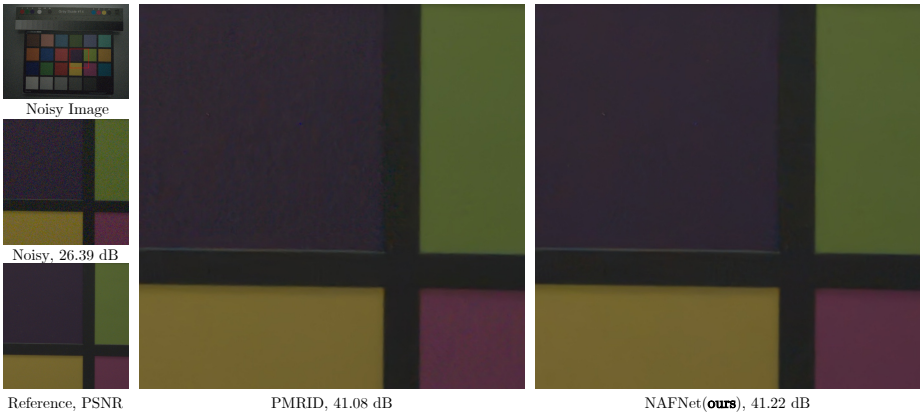

Fig. 1: Additional qualitatively comparison of raw image denoising results with PMRID[6]. Zoom in to see details

1.2 Channel Attention and Simplified Channel Attention

For a feature map with width of c, the channel attention module shrinks it by a factor of r and then project it back into c (by fully-connect layer). The computational cost could be approximated by $c \times c/r + c/r \times c$. As to the simplified channel attention module, its computational cost is $c \times c$. For a fair comparison, we choose $r = 2$ so that their computational costs are consistent in our experiments.

1.3 Feature Fusion

There are skip connections from the encoder block to the decoder block, and there are several ways to fuse the features of encoder/decoder. In [2], the encoder features are transformed by a convolution and then concatenate with the decoder features. In [8], features are concatenated first and then transformed by a convolution. Differently, we simply element-wise add the encoder and decoder features as the feature fusion approach.

1.4 Downsample/Upsample Layer

For the downsample layer, we use the convolution with a kernel size of 2 and a stride of 2. This design choice is inspired by [1]. For the upsample layer, we double the channel width by a pointwise convolution first, and then follows a pixel shuffle module[5].

2 More Visualization Results

We provide additional visualization results of raw image denoising, image deblurring, RGB image denoising tasks, as we shown in Figure 1, 2, and 3. Our

Fig. 2: Additional qualitative comparison of image deblurring methods

baselines can restore more fine details compare to other methods. It is recommended to zoom in to compare the details in the red box.

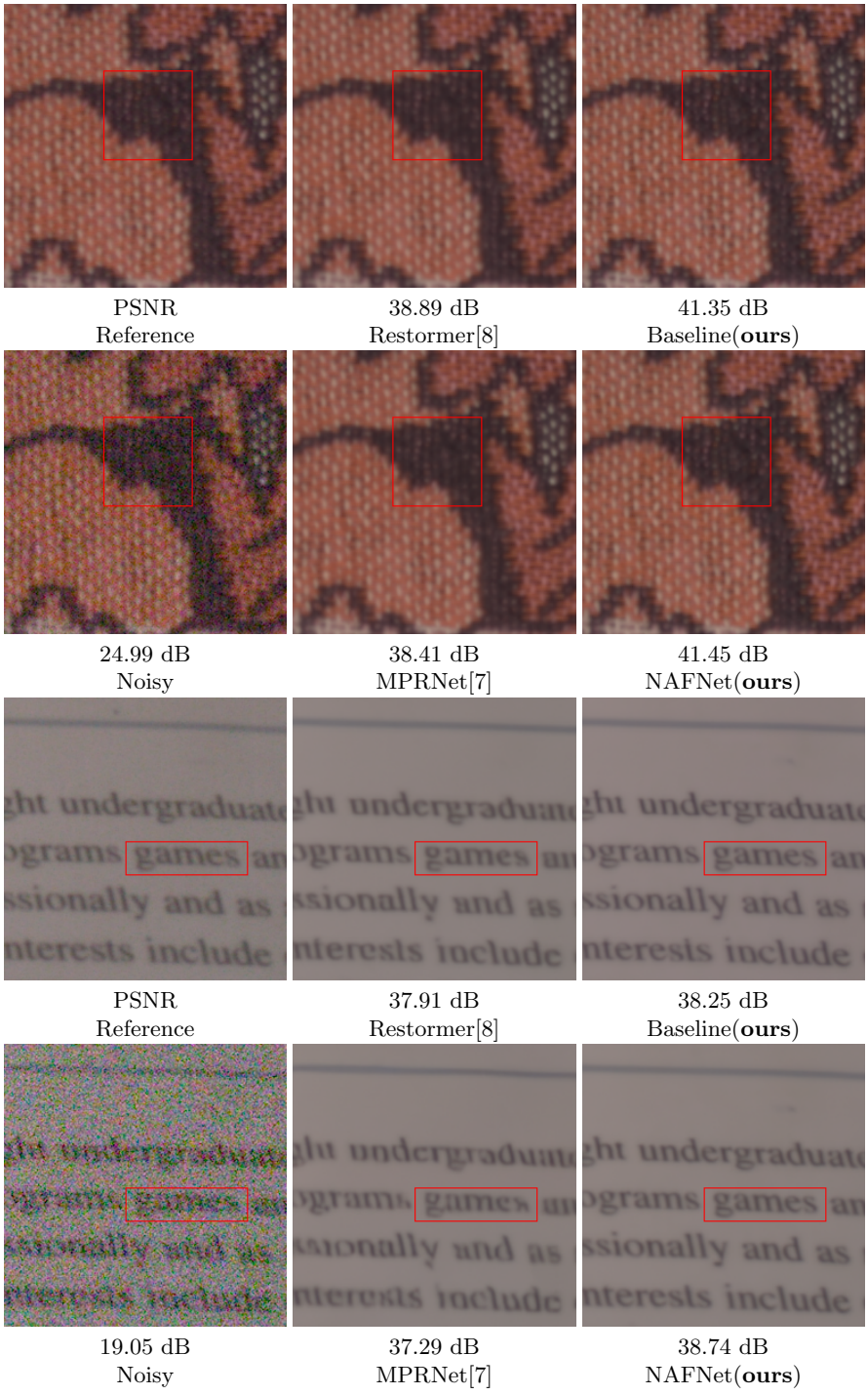

Fig. 3: Additional qualitative comparison of image denoising methods

References

1. Alsallakh, B., Kokhlikyan, N., Miglani, V., Yuan, J., Reblitz-Richardson, O.: Mind the pad–cnns can develop blind spots. arXiv preprint arXiv:2010.02178 (2020)
2. Chen, L., Lu, X., Zhang, J., Chu, X., Chen, C.: Hinet: Half instance normalization network for image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 182–192 (2021)
3. Chu, X., Chen, L., , Chen, C., Lu, X.: Improving image restoration by revisiting global information aggregation. arXiv preprint arXiv:2112.04491 (2021)
4. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022)
5. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016)
6. Wang, Y., Huang, H., Xu, Q., Liu, J., Liu, Y., Wang, J.: Practical deep raw image denoising on mobile devices. In: European Conference on Computer Vision. pp. 1–16. Springer (2020)
7. Waqas Zamir, S., Arora, A., Khan, S., Hayat, M., Shahbaz Khan, F., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. arXiv e-prints pp. arXiv–2102 (2021)
8. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. arXiv preprint arXiv:2111.09881 (2021)