Blind2Unblind: Self-Supervised Image Denoising with Visible Blind Spots

Zejin Wang[1,2] Jiazheng Liu[1,3] Guoqing Li[1] Hua Han[1,3,*]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences

[2]School of Artificial Intelligence, University of Chinese Academy of Sciences

[3]School of Future Technology, University of Chinese Academy of Sciences

A. Training Framework for Blind2Unblind

The training framework for Blind2Unblind is shown in Algorithm 1.

Algorithm 1: Blind2Unblind

Input: A set of noisy images $Y = \{\mathbf{y}_i\}_{i=1}^{n}$;
Denoising network $f_\theta(\cdot)$;
Hyper-parameters η, λ;

while *not converged* **do**

- Sample a noisy image $\mathbf{y}$;
- Generate a global masker $\mathbf{\Omega}_{(\cdot)}$;
- Derive a masked volume $\mathbf{\Omega_y}$, where $\mathbf{\Omega_y}$ is the network input, and $\mathbf{y}$ is the network target;
- For the network input $\mathbf{\Omega_y}$, derive the denoised volume $f_\theta(\mathbf{\Omega_y})$;
- Global mask mapper $h_{(\cdot)}$ samples the denoised volume $f_\theta(\mathbf{\Omega_y})$ at blind spots, then obtain a blind denoised image $h(f_\theta(\mathbf{\Omega_y}))$;
- For the original noisy image $\mathbf{y}$, derive the visible denoised image $\hat{f}_\theta(\mathbf{y})$ without gradients;
- Calculate re-visible loss $\mathcal{L}_{rev} = \|h(f_\theta(\mathbf{\Omega_y})) + \lambda \hat{f}_\theta(\mathbf{y}) - (\lambda+1)\mathbf{y}\|_2^2$;
- Calculate regularization $\mathcal{L}_{reg} = \|h(f_\theta(\mathbf{\Omega_y})) - \mathbf{y}\|_2^2$;
- Update network parameters θ by minimizing the regularized re-visible loss $\mathcal{L}_{rev} + \eta \cdot \mathcal{L}_{reg}$;

end

B. Details of Interpolation from Neighbors

Figure 1 shows the workflow of interpolation from neighbors. The workflow can be divided into the following three steps: 1) The mask is generated by random masking each 2×2 cells in image $\mathbf{y}$. The kernel convolves image $\mathbf{y}$ with stride 1 and padding 1 to produce $\mathbf{y}_c$. Then, $\mathbf{y}_m$ is obtained via Hadamard product $\mathbf{y}_c \circ mask$. 2) We perform Hadamard product $\mathbf{y} \circ (1 - mask)$ to gernerate $\mathbf{y}_{inv}$. 3) Sum by $\mathbf{y}_m$ and $\mathbf{y}_{inv}$, we finally obtain the masked image $\mathbf{\Omega_y}$.

*Corresponding author

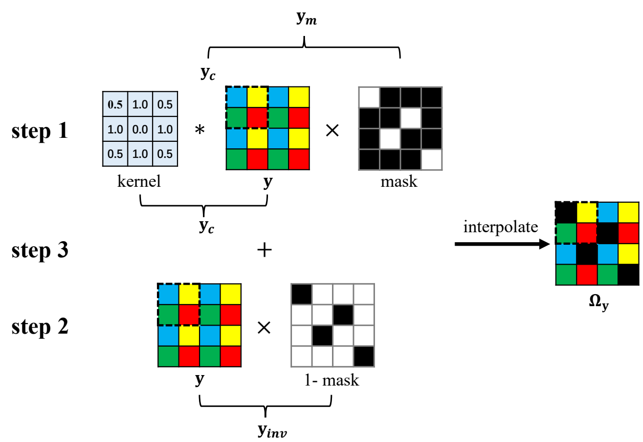

Figure 1. Details of interpolation from neighbors.

C. Details of Random Mask Strategy

The illustration of the random mask strategy is presented in Figure 2. The image y is divided into several blocks with 2x2 cells. A specific pixel in each cell is randomly set as a blind spot. Namely, there are four ways for random masking of 2x2 cells. After random masking, the masked image $\mathbf{\Omega_y}$ is fed into the denoising network to generate the denoised image $f_\theta(\mathbf{\Omega_y})$.

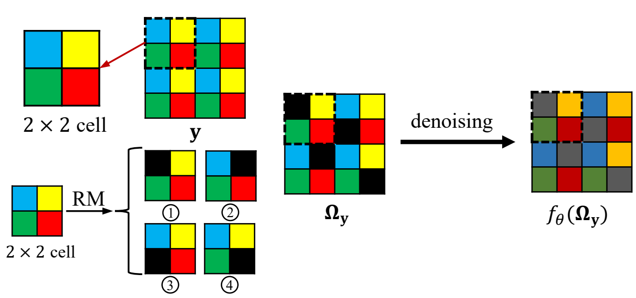

Figure 2. Details of random mask strategy.

D. More Experimental Results

Figure 3 illustrates the steps of our proposed method while denoising sRGB images in the setting of $\sigma = 25$. Figure 4 shows the visual comparison of denoising raw-RGB images in the challenging SIDD benchmark.

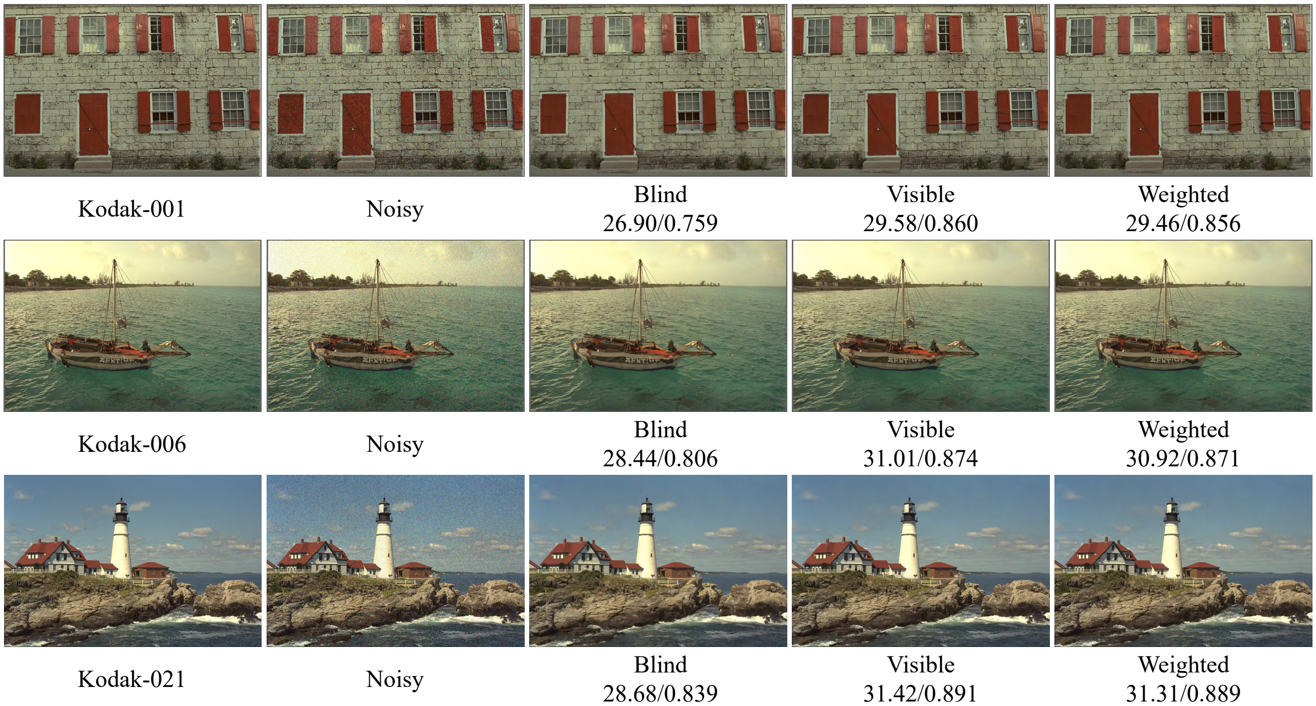

Figure 3. The steps of our proposed method while denoising sRGB images in the setting of $\sigma = 25$. Blind denotes $h(f_\theta(\mathbf{\Omega_y}))$, Visible denotes $\hat{f}_\theta(\mathbf{y})$, and Weighted denotes $\frac{h(f_\theta^*(\mathbf{\Omega_y}))+\lambda \hat{f}_\theta^*(\mathbf{y})}{\lambda+1}$.

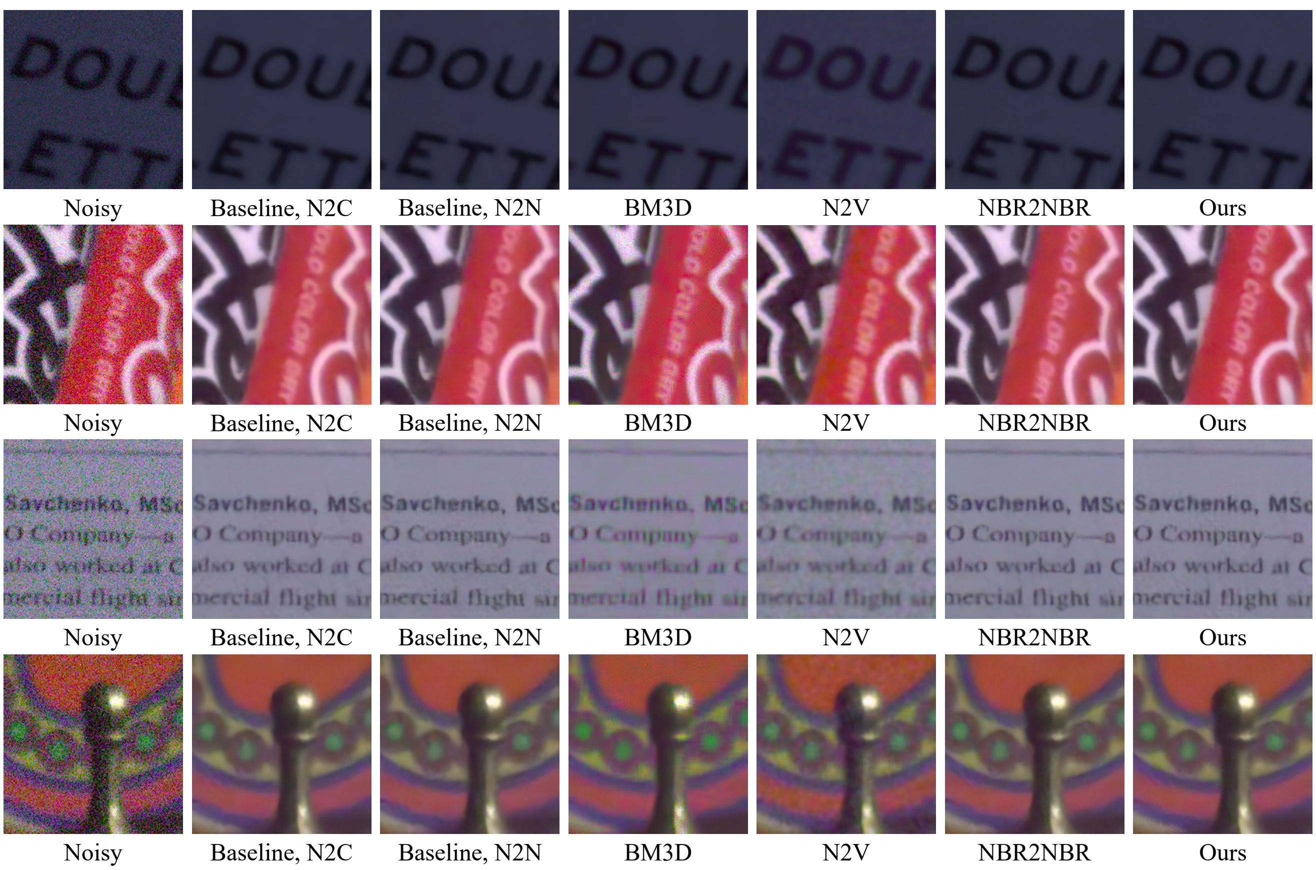

Figure 4. Visual comparison of denoising raw-RGB images in the challenging SIDD benchmark.