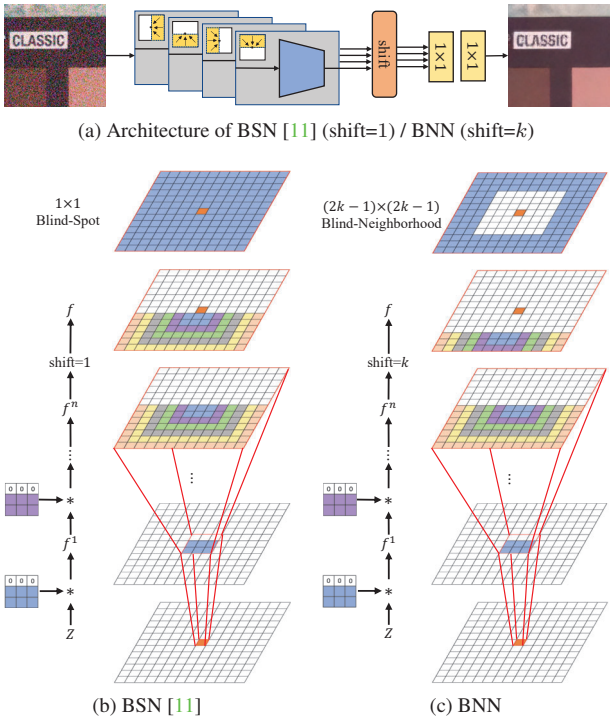
Spatially Adaptive Self-Supervised Learning for Real-World Image Denoising

Supplementary Material

Junyi Li[1], Zhilu Zhang[1], Xiaoyu Liu[1], Chaoyu Feng, Xiaotao Wang, Lei Lei, Wangmeng Zuo[1,2](✉)
[1]School of Computer Science and Technology, Harbin Institute of Technology, China
[2]Peng Cheng Laboratory, China
nagejacob@gmail.com, cszlzhang@outlook.com, liuxiaoyu1104@gmail.com, wmzuo@hit.edu.cn



(a) Architecture of BSN [11] (shift=1) / BNN (shift=k)

(b) BSN [11] (c) BNN

Figure A. Illustration of our BNN. (a) BNN is implemented by adjusting the shift size of the BSN [11]. (b) BSN shifts one pixel to create 1×1 blind-spot. (c) Our BNN shifts k pixels to create $(2k-1)\times(2k-1)$ blind-neighborhood.

A. Content

The content of this supplementary material involves:

- Detailed architectures of BNN and LAN in Sec. B.
- Analysis of model efficiency in Sec. C.
- More analysis of soft coefficients in Sec. D
- Effect of different training strategies in Sec. E.
- Result comparison using different training datasets in Sec. F.
- Comparison between BNN and pixel-shuffle downsampling in Sec. G.
- Additional qualitative results in Sec. H.

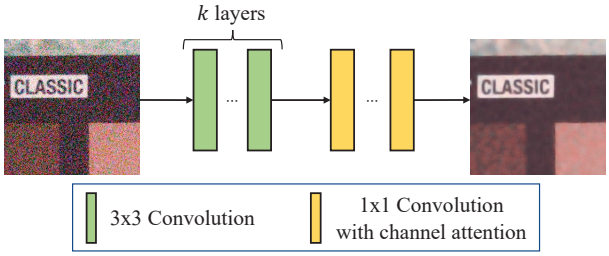


Figure B. Network architecture of LAN. We stack k 3×3 convolution layers to create $(2k+1)\times(2k+1)$ receptive field, then refine the features by several 1×1 convolution blocks with channel attention mechanism [23].

B. Detailed Architectures of BNN and LAN

We show the detailed architectures of the blind-neighborhood network (BNN) in Fig. A and the locally aware network (LAN) in Fig. B.

Blind-neighborhood Network (BNN). Our BNN is modified from the BSN used in HQ-SSL [11]. As shown in Figure A(a), BNN shares the same architecture with BSN [11]. It applies four network branches whose receptive field is restricted in different directions. The output feature of each network branch is further shifted to create the blind-spot or blind-neighborhood. Figure A(b) shows the receptive field of one network branch and the corresponding shift operation for BSN [11]. The network branch extracts feature f^n

Table A. Model efficiency analyses of unpaired and self-supervised methods. #FLOPs and time is measured on denoising a 256×256 image patch.

	Method	#Param (M)	#FLOPs (G)	Time (ms)
Unpaired	GCBD [3]	**0.56**	73.1	6.1
	UIDNet [8]	**0.56**	73.1	6.1
	C2N [9]	217.26	2978.5	154.0
	Wu *et al.* [17]	24.93	137.2	17.9
Self-Supervised	Noise2Void [10]	7.18	73.7	6.9
	Noise2Self [2]	**0.56**	73.1	6.1
	NAC [18]	0.78	51.2	9.7
	R2R [14]	**0.56**	73.1	6.1
	CVF-SID [13]	1.19	155.7	14.8
	AP-BSN+R^3 [12]	3.66	3788.1	418.6
	Ours	1.08	**35.0**	**4.8**

with one direction receptive field from input Z, while the center pixel is in the receptive field. To exclude the center pixel from the receptive field, the shift operation is applied on f^n to generate f:

$$f(i,j) = f^n(i, j-1) \tag{1}$$

where (i, j) is the spatial position for each pixel. The shifted features of four network branches are fused to a whole receptive field with blind-spot. From Figure A(c), our BNN should exclude not only the center pixel but also its neighboring pixels from the receptive field, so we enlarge the shift size from 1 to k:

$$f(i,j) = f^n(i, j-k) \tag{2}$$

The fusion of four network branches of BNN results in a whole receptive field with $(2k-1) \times (2k-1)$ blind-neighborhood.

Locally Aware Network (LAN). LAN learns the supervision for textured areas. We stack 3×3 convolution layers to create the local receptive field, specifically, k 3×3 layers can make up $(2k+1) \times (2k+1)$ receptive field. In order to further refine the color information, we additionally add several 1×1 convolution blocks with channel attention mechanism [23].

C. Analysis of Model Efficiency

We focus on developing a novel self-supervised denoising framework, rather than a denoising network architecture. Thus, we use a common and efficient network architecture, *i.e.*, U-Net [16] as our denoising network. Most methods also use representative denoising networks, such as DnCNN [22] and U-Net [16]. But the performance of some methods highly depends on the denoising network capacity or complicated post-processing. Table A shows the model efficiency of several unpaired and self-supervised methods. We achieve lowest #FLOPs and inference time among the competing methods. Among these, the performance of C2N [9] is achieved with DIDN [19], which costs 154ms to denoise a 256×256 patch. AP-BSN [12] has the the closest performance to ours, but costs 418.6ms due to the time-consuming random-replacing refinement (R^3) strategy. The #FLOPs and inference time of our method are only $\sim 1\%$ that of AP-BSN [12]. In short, our method is not only effective but also efficient.

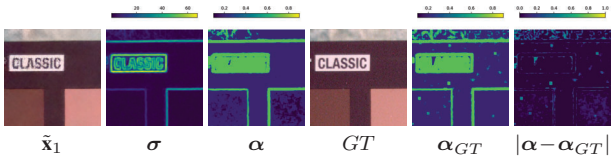


Figure C. Visual analysis of standard deviation map $\boldsymbol{\sigma}$ and soft coefficient $\boldsymbol{\alpha}$. $\boldsymbol{\alpha}_{GT}$ is computed from clean images.

Table B. PSNR of different thresholds in Eqn. (3).

Lower / Upper bound	0.5 / 5	2 / 5	1 / 3	1 / 7	1 / 5
PSNR	37.32	37.26	37.18	37.33	37.39

Table C. Result comparison on SIDD validation dataset [1] of different training strategies.

Training Strategy	PSNR / SSIM	Training Time
Joint Training	37.12 / 0.928	120h
Multi-Stage Training	37.39 / 0.934	54h

D. More Analysis of Soft Coefficients

To evaluate the accuracy of $\boldsymbol{\alpha}$, we make a comparison between $\boldsymbol{\alpha}$ (computed from BNN outputs) and $\boldsymbol{\alpha}_{GT}$ (computed from clean images). Like $\boldsymbol{\alpha}_{GT}$, $\boldsymbol{\alpha}$ can well detect the texture regions and edges (see Fig. C). When replacing $\boldsymbol{\alpha}$ with $\boldsymbol{\alpha}_{GT}$ for the network training, we obtain similar denoising performance. From Fig. C, $\boldsymbol{\sigma}$ is a reliable indicator that it's usually higher than 5 in the textured areas (*e.g.*, edges, texts) and lower than 1 in flat areas. To generate the coefficient $\boldsymbol{\alpha}$ that indicating flatness, we convert $\boldsymbol{\sigma}$ to $\boldsymbol{\alpha}$ with piecewise sigmoid function in Eqn. (3) and set the threshold empirically. Tab. B shows the effect of thresholds, and the sensitivity to thresholds is acceptable.

E. Effect of Different Training Strategies

Our method consists of three networks, *i.e.* BNN, LAN and U-Net, where BNN and LAN learn spatially adaptive supervisions for U-Net. These three networks can be

Table D. Result comparison using different training datasets. We compare the results of different models trained with the same dataset (*i.e.* SIDD Medium, SIDD Benchmark, or DND Benchmark).

(a) Training on SIDD Medium.

Method	SIDD Benchmark PSNR↑ / SSIM↑	DND Benchmark PSNR↑ / SSIM↑
CVF-SID [13]	34.43 / 0.912	36.31 / 0.923
AP-BSN+R^3 [12]	35.97 / 0.925	37.98 / 0.938
Ours	37.41 / 0.934	38.34 / 0.941

(b) Training on SIDD Benchmark.

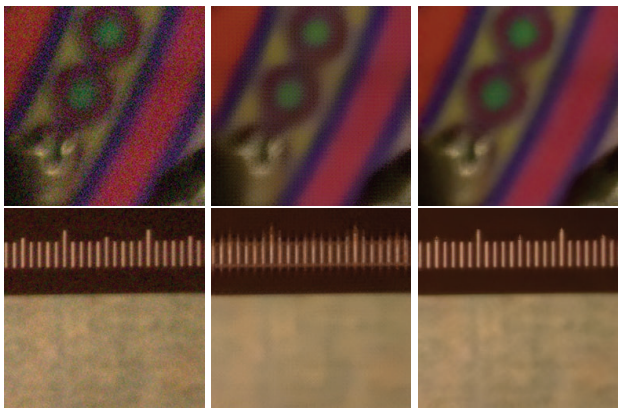
Method	SIDD Benchmark PSNR↑ / SSIM↑
CVF-SID [13]	34.51 / 0.916
AP-BSN+R^3 [12]	36.91 / 0.931
Ours	37.37 / 0.929

(c) Training on DND Benchmark.

Method	DND Benchmark PSNR↑ / SSIM↑
CVF-SID [13]	36.49 / 0.924
AP-BSN+R^3 [12]	38.09 / 0.937
Ours	38.58 / 0.936

Table E. Quantitative comparison between BNN and the PD strategy on SIDD validation dataset [1]. PD_5 denotes pixel-shuffle downsampling with downsampling factor 5. 'PSNR of Flat Areas' is calculated on the flat areas of PD_5+DBSN or BNN output, which are detected from ground-truth clean images. 'PSNR of Final Denoising' is calculated on the denoising results of U-Net [16], which is adaptively supervised by our LAN output as well as PD_5+DBSN or our BNN output.

Supervision for Flat Areas	PD_5+DBSN [17]	BNN
PSNR of Flat Areas	51.36	54.34
PSNR of Final Denoising	34.32	37.39



(a) Noisy (b) PD_5+DBSN [17] (c) BNN

Figure D. Visual comparison between our BNN and PD_5+DBSN output on SIDD Validation dataset [1].

trained simultaneously or successively, which are named joint training strategy and multi-stage training strategy, respectively. The joint training strategy updates the parameters of all three networks at every iteration, while the multi-stage training strategy only updates the parameters of one network at each stage. We compare the denoising performance and training time of the two training strategies on SIDD validation dataset [1]. From Table C, the multi-stage training strategy has better quantitative results and less than half the training time of joint training one. Thus we choose multi-stage training as our training strategy.

F. Result Comparison Using Different Training Datasets

Self-supervised denoisers [12, 13] are suitable for situations when the training images are scarce. They can be trained solely on the testing dataset (*e.g.*, SIDD Benchmark with 40 images [1], DND Benchmark with 50 images [15]) rather than a training dataset (*e.g.*, SIDD Medium with 320 images [1]). Training on the testing images may also benefit the denoising performance, where the networks are better fitted to the noise distribution of the testing images.

In Table 1 of the main text, the results of CVF-SID [13] and AP-BSN [12] are measured on the models trained with corresponding testing datasets, while ours are measured on the model only trained with SIDD Medium. For a fairer comparison, here we report the results of different models trained with the same dataset (*i.e.* SIDD Medium, SIDD Benchmark, or DND Benchmark) in Table D. It can be seen that no matter which dataset is used for training, our model can outperform CVF-SID [13] and AP-BSN [12] well. In addition, on the DND Benchmark dataset, our model trained on the testing images shows 0.24dB improvement over the model trained on SIDD Medium dataset.

G. Comparison between BNN and Pixel-Shuffle Downsampling

In this section, we demonstrate BNN can provide better supervision for flat areas than the pixel-shuffle downsampling (PD) strategy. Pixel-shuffle downsampling strategy breaks the spatial correlation of noise, then a spatially independent denoiser can be applied to denoise the sub-images. However, as mentioned in Sec. 3.1 in the main text, PD destroys the high-frequency information [5] and leads to aliasing artifacts.

We conduct experiments on the PD strategy. We first apply downsampling factor 5 (PD_5) to cover the noise correlation range, then we utilize recent DBSN [17] to remove the noise of sub-sampled images. As shown in Figure D, PD_5+DBSN [17] indeed removes noise, but additionally introduces aliasing artifacts. BNN does not suffer from aliasing artifacts and achieves better quantitative results, as BNN operates on the original resolution. From Table E, BNN

achieves better quantitative results in flat areas, thus provides better supervision for flat areas than PD_5+DBSN [17]. The performance of the denoising U-Net also demonstrates the superiority of BNN.

H. Additional Qualitative Results

The additional visual comparison on SIDD [1] and DND [15] dataset can be seen in Fig. E and Fig. F, respectively.

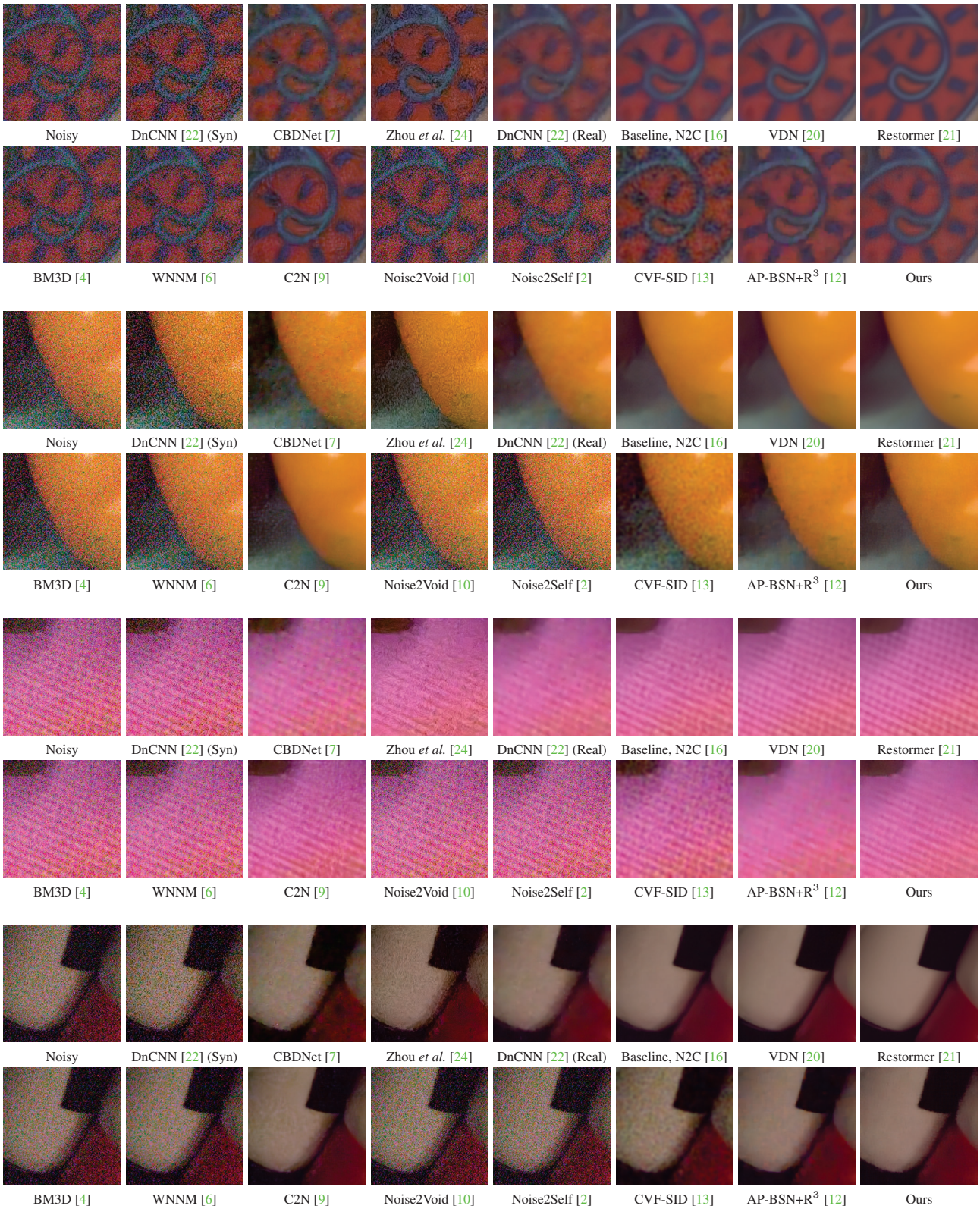


Figure E. Visual comparison on SIDD validation dataset [1].

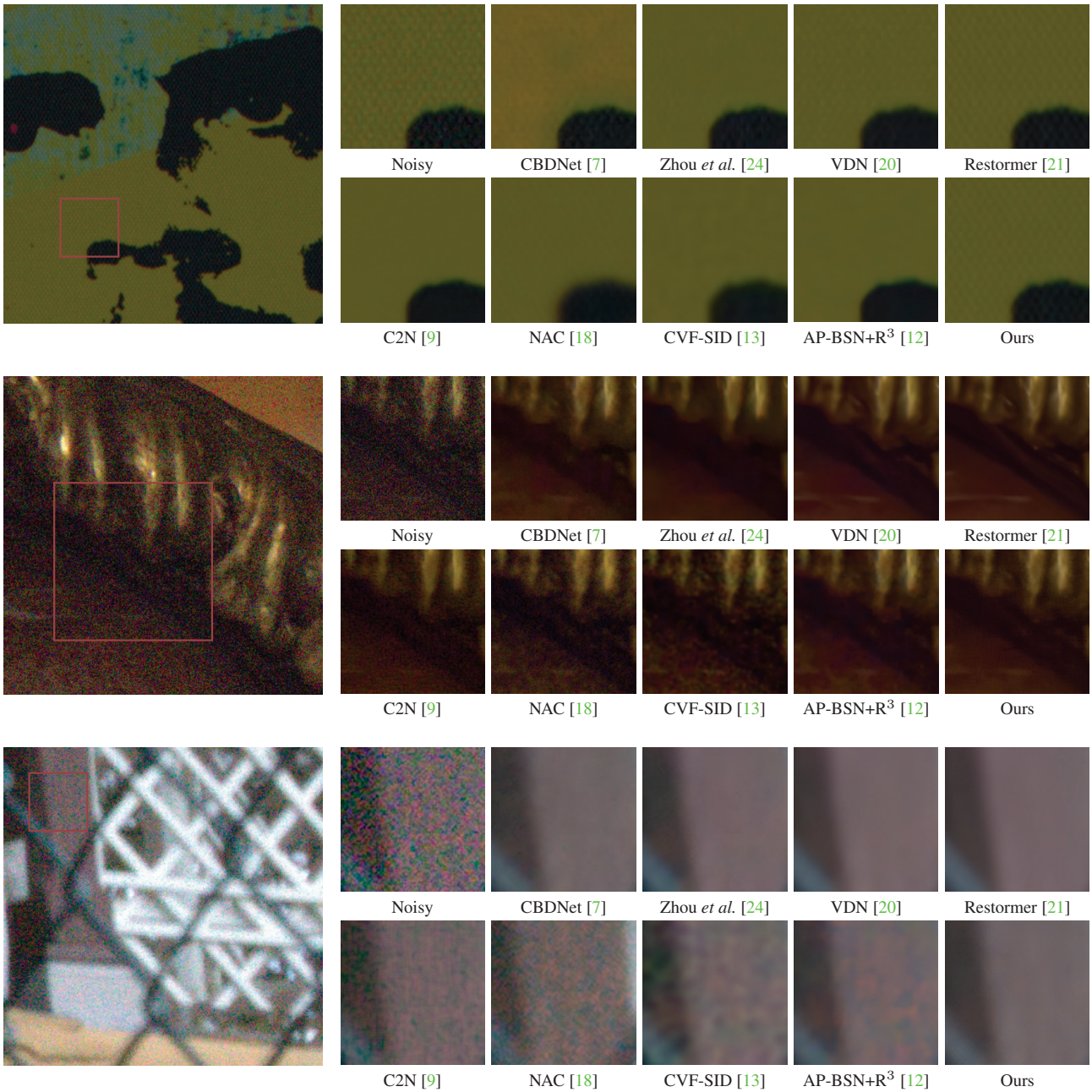


Figure F. Visual comparison on DND benchmark testing dataset [15].

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018. 2, 3, 4, 5

[2] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *International Conference on Machine Learning*, pages 524–533. PMLR, 2019. 2, 5

[3] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3155–3164, 2018. 2

[4] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 5

[5] Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009. 3

[6] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014. 5

[7] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1712–1722, 2019. 5, 6

[8] Zhiwei Hong, Xiaocheng Fan, Tao Jiang, and Jianxing Feng. End-to-end unpaired image denoising with conditional adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4140–4149, 2020. 2

[9] Geonwoon Jang, Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. C2n: Practical generative noise modeling for real-world denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2350–2359, 2021. 2, 5, 6

[10] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2129–2137, 2019. 2, 5

[11] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[12] Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. Ap-bsn: Self-supervised denoising for real-world images via asymmetric pd and blind-spot network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17725–17734, 2022. 2, 3, 5, 6

[13] Reyhaneh Neshatavar, Mohsen Yavartanoo, Sanghyun Son, and Kyoung Mu Lee. Cvf-sid: Cyclic multi-variate function for self-supervised image denoising by disentangling noise from image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17583–17591, 2022. 2, 3, 5, 6

[14] Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-recorrupted: unsupervised deep learning for image denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2043–2052, 2021. 2

[15] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2017. 3, 4, 6

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2, 3, 5

[17] Xiaohe Wu, Ming Liu, Yue Cao, Dongwei Ren, and Wangmeng Zuo. Unpaired learning of deep image denoising. In *European conference on computer vision*, pages 352–368. Springer, 2020. 2, 3, 4

[18] Jun Xu, Yuan Huang, Ming-Ming Cheng, Li Liu, Fan Zhu, Zhou Xu, and Ling Shao. Noisy-as-clean: Learning self-supervised denoising from corrupted image. *IEEE Transactions on Image Processing*, 29:9316–9329, 2020. 2, 6

[19] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up cnn for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[20] Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. *Advances in neural information processing systems*, 32, 2019. 5, 6

[21] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022. 5, 6

[22] 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*, 26(7):3142–3155, 2017. 2, 5

[23] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018. 1, 2

[24] Yuqian Zhou, Jianbo Jiao, Haibin Huang, Yang Wang, Jue Wang, Honghui Shi, and Thomas Huang. When awgn-based denoiser meets real noises. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13074–13081, 2020. 5, 6