Supplementary Materials: Identity Enhanced Residual Image Denoising

Saeed Anwar[1,2], Cong Phuoc Huynh[2], Fatih Porikli[2]
[1]Data61-CSIRO, [2]The Australian Natinal University
saeed.anwar@data61.csiro.au

In this section of the paper, we provide performance on the synthetic datasets and more results on real-image denoising.

1. Identity Module vs. Residual module

Figure 1 shows the difference between the our network blocks and resent blocks. It is to be noted here, that resnet employs different block structures in the same network.

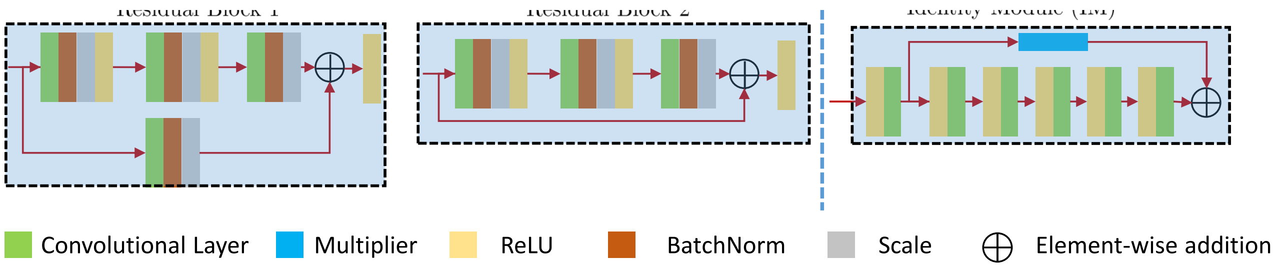

Figure 1. The comparison between our network block and the Resnet blocks. Our network block is simple than resent which employs several different layers.

As compared to our method, ResNet [5] contains elements such as branched convolutions, strides, batch normalization, flattening, a high number of feature maps, pooling, fully convolutional layer and post-ReLU. Here, it is to be noted that ResNet [5] in its original form is not suitable for image denoising. So, we experimented with a modified version of a ResNet, by setting all the convolution strides to one and removing pooling and fully convolutional layer to make it suitable for denoising purposes. Furthermore, we reduce the depth of the network to 20 layers to make it comparable to our model, DnCNN, and other state-of-the-art networks. The PSNR on BSD68 ($\sigma = 25$) for the modified ResNet network is 16.25 dB without dilation and 16.90 dB with a dilation of three, keeping all the other parameters and training details as ours. This result is very low even compared with decade old methods.

2. inference time and memory cost

The number of parameters in RIDNet is 1.5M (having a model size of 6.0MB) as compared to IERD's 1M parameter (with 2.7MB model size). Similarly, the inference time taken by our method is 11.42s for the BSD68 dataset, while the time taken by RIDNet is 15.42s. It should be noted here that the time reported also contains loading images, displaying statistics, and saving the results.

3. Datasets

We performed experimental validation on the widely used publicly available three synthetically generated noisy datasets described below.

- **Classical images:** As a tradition, we first provide a comparison on 12 classical images. The noise of standard deviations (std) of $\sigma_n = 15, 25, 50, 70$ are added to the image.

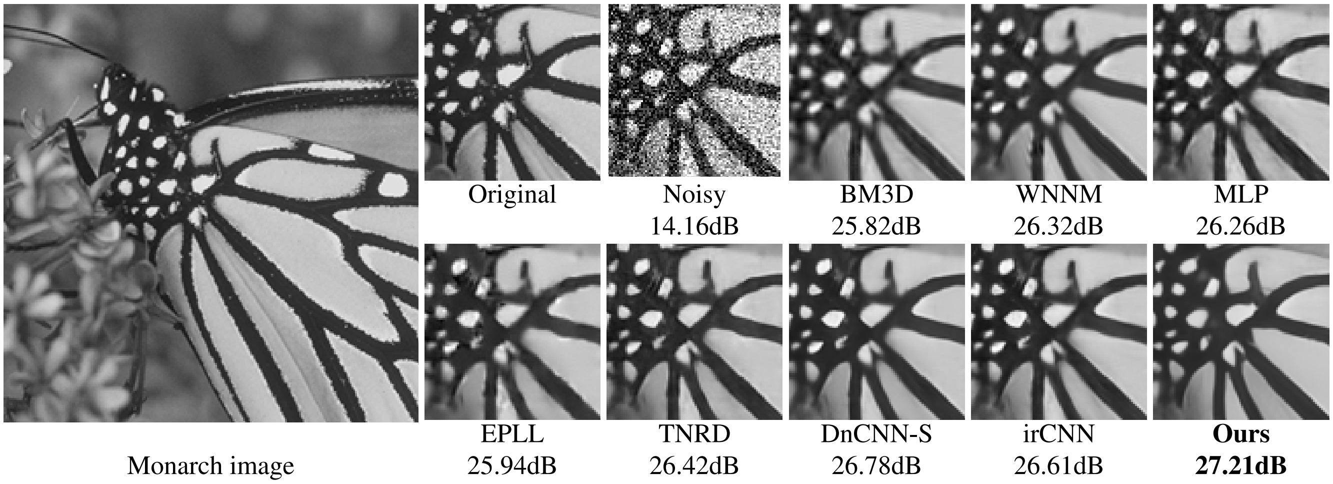

Figure 2. Denoising quality comparison on a sample image with strong edges and texture, selected from classical image set for noise level $\sigma_n = 50$. The visual quality, *i.e.* sharpness of the edges on the wings and small textures reproduced by our method is the best among all.

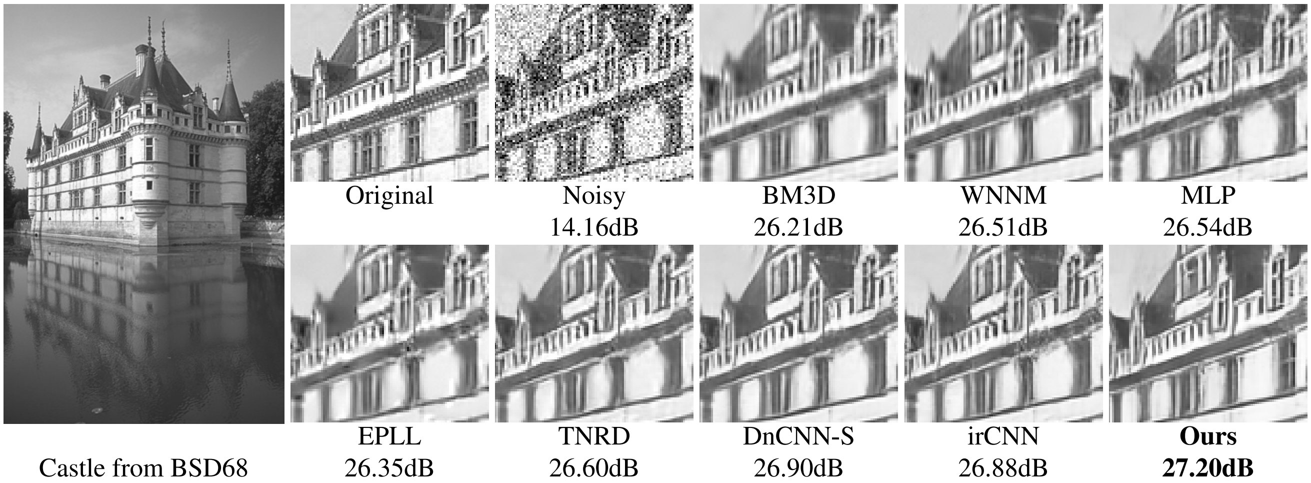

Figure 3. Comparison on a sample image from BSD68 dataset for $\sigma_n = 50$. Our network is able to recover fine textures on the castle

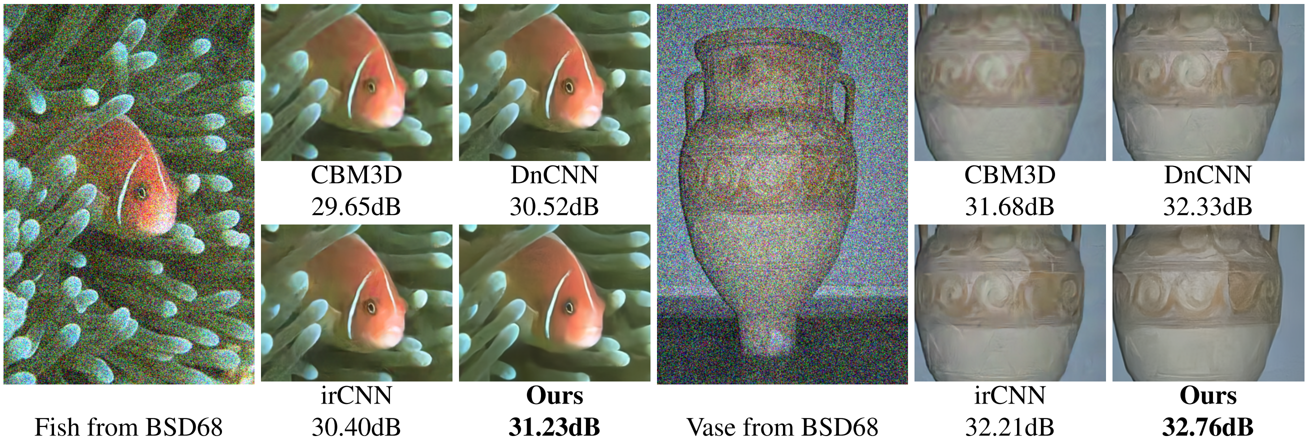

Figure 4. Denoising performance for state-of-the-art versus the proposed method on sample color images from the dataset in [8], where the noise standard deviation σ_n is 50. The image we recover is more natural, contains less contrast artifacts and is closest to the ground-truth.

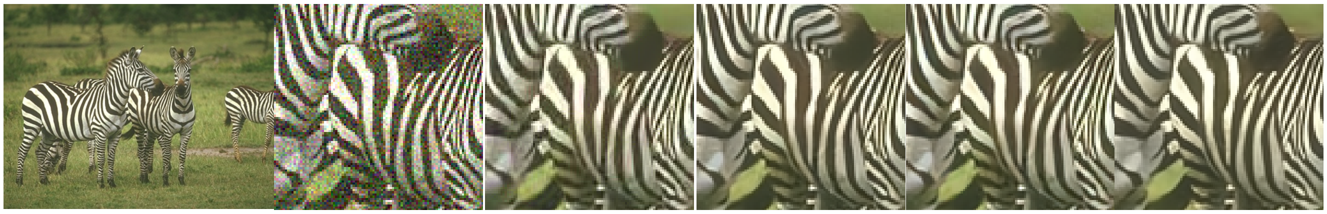
Original Input (20.18 dB) CBM3D (29.37 dB)DnCNN (30.89 dB) irCNN (30.60 dB) Ours (**31.04** dB)

Figure 5. A sample color image with rich textures, selected from [8]'s BSD68 dataset for $\sigma_n = 25$.

Table 1. Performance comparison between image denoising algorithms on widely used classical images, in terms of PSNR (in dB). The best results are highlighted with red color while the blue color represents the second best denoising results.

	Cman	House	Peppers	Starfish	Monar	Airpl	Parrot	Lena	Barbara	Boat	Man	Couple	Average
$\sigma_n = 15$													
BM3D	31.91	34.93	32.69	31.14	31.85	31.07	31.37	34.26	33.10	32.13	31.92	32.10	32.372
WNNM	32.17	35.13	32.99	31.82	32.71	31.39	31.62	34.27	33.60	32.27	32.11	32.17	32.696
EPLL	31.85	34.17	32.64	31.13	32.10	31.19	31.42	33.92	31.38	31.93	32.00	31.93	32.138
CSF	31.95	34.39	32.85	31.55	32.33	31.33	31.37	34.06	31.92	32.01	32.08	31.98	32.318
TNRD	32.19	34.53	33.04	31.75	32.56	31.46	31.63	34.24	32.13	32.14	32.23	32.11	32.502
DnCNNS	32.61	34.97	33.30	32.20	33.09	31.70	31.83	34.62	32.64	32.42	32.46	32.47	32.859
DnCNNB	32.10	34.93	33.15	32.02	32.94	31.56	31.63	34.56	32.09	32.35	32.41	32.41	32.680
IrCNN	32.55	34.89	33.31	32.02	32.82	31.70	31.84	34.53	32.43	32.34	32.40	32.40	32.769
Our-agnostic	32.11	35.10	33.28	32.31	33.07	31.58	31.80	34.67	32.48	32.42	32.40	32.50	32.812
Our-specific	32.61	35.21	33.21	32.35	33.33	31.77	32.01	34.69	32.74	32.44	32.50	32.52	32.950
$\sigma_n = 25$													
BM3D	29.45	32.85	30.16	28.56	29.25	28.42	28.93	32.07	30.71	29.90	29.61	29.71	29.969
WNNM	29.64	33.22	30.42	29.03	29.84	28.69	29.15	32.24	31.24	30.03	29.76	29.82	30.257
EPLL	29.26	32.17	30.17	28.51	29.39	28.61	28.95	31.73	28.61	29.74	29.66	29.53	29.692
MLP	29.61	32.56	30.30	28.82	29.61	28.82	29.25	32.25	29.54	29.97	29.88	29.73	30.027
CSF	29.48	32.39	30.32	28.80	29.62	28.72	28.90	31.79	29.03	29.76	29.71	29.53	29.837
TNRD	29.72	32.53	30.57	29.02	29.85	28.88	29.18	32.00	29.41	29.91	29.87	29.71	30.055
DnCNNS	30.18	33.06	30.87	29.41	30.28	29.13	29.43	32.44	30.00	30.21	30.10	30.12	30.436
DnCNNB	29.94	33.05	30.84	29.34	30.25	29.09	29.35	32.42	29.69	30.20	30.09	30.10	30.362
IrCNN	30.08	33.06	30.88	29.27	30.09	29.12	29.47	32.43	29.92	30.17	30.04	30.08	30.384
Our-agnostic	29.87	33.34	30.94	29.68	30.39	29.08	29.38	32.65	30.17	30.27	30.08	30.20	30.505
Our-specific	30.26	33.44	30.87	29.77	30.62	29.23	29.61	32.66	30.29	30.30	30.18	30.24	30.624
$\sigma_n = 50$													
BM3D	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.22	26.78	26.81	26.46	26.722
WNNM	26.45	30.33	26.95	25.44	26.32	25.42	26.14	29.25	27.79	26.97	26.94	26.64	27.052
EPLL	26.10	29.12	26.80	25.12	25.94	25.31	25.95	28.68	24.83	26.74	26.79	26.30	26.471
MLP	26.37	29.64	26.68	25.43	26.26	25.56	26.12	29.32	25.24	27.03	27.06	26.67	26.783
TNRD	26.62	29.48	27.10	25.42	26.31	25.59	26.16	28.93	25.70	26.94	26.98	26.50	26.812
DnCNNS	27.03	30.00	27.32	25.70	26.78	25.87	26.48	29.39	26.22	27.20	27.24	26.90	27.178
DnCNNB	27.03	30.02	27.39	25.72	26.83	25.89	26.48	29.38	26.38	27.23	27.23	26.91	27.206
IrCNN	26.88	29.96	27.33	25.57	26.61	25.89	26.55	29.40	26.24	27.17	27.17	26.88	27.136
Our-agnostic	27.03	30.48	27.57	26.01	27.03	25.84	26.53	29.77	26.89	27.28	27.29	27.06	27.398
Our-specific	27.25	30.70	27.54	26.05	27.21	26.06	26.53	29.65	26.62	27.36	27.26	27.24	27.457
$\sigma_n = 70$													
BM3D	24.62	27.91	25.07	23.56	24.24	23.75	24.49	27.57	25.47	25.40	25.56	25.00	25.221
WNNM	24.86	28.59	25.25	23.78	24.62	24.00	24.64	27.85	26.17	25.58	25.68	25.18	25.517
EPLL	24.60	27.32	25.03	23.52	24.19	23.72	24.44	27.11	23.20	25.27	25.50	24.80	24.891
DnCNNS	25.37	28.22	25.50	23.97	25.10	24.34	24.98	27.85	23.97	25.76	25.91	25.31	25.523
Our-specific	25.83	29.19	25.90	24.28	25.66	24.59	25.12	28.25	25.06	26.00	26.02	25.78	25.974

- **BSD68:** Berkely Segmentation Dataset abbreviated as BSD68 ([12]) is composed of 68 images. We provide qualitative and quantitative results for both grayscale images. To generate noisy test images, we corrupt the images by additive white Gaussian noise with standard deviations (std) of $\sigma_n = 15, 25, 50, 70$.

Table 2. Performance comparison between our method and existing algorithms on the grayscale version of the BSD68 dataset ([8]). The missing denoising results, indicated by “-”, occurs when the method is not trained to deal with the input noisy images.

Noise	Methods												
Level	BM3D [3]	WNNM [4]	EPLL [16]	TNRD [2]	DenoiseNet [11]	DnCNN [14]	IrCNN [15]	NLNet [6]	MWCNN [13]	NLRN [7]	N3Net [10]	Ours-Agnostic	Ours-Specific
15	31.08	31.32	31.19	31.42	31.44	31.73	31.63	31.52	31.86	31.88	-	31.68	31.81
25	28.57	28.83	28.68	28.92	29.04	29.23	29.15	29.03	29.41	29.41	29.30	29.18	29.34
50	25.62	25.83	25.67	26.01	26.06	26.23	26.19	26.07	26.53	26.47	26.39	26.31	26.40
70	24.44	-	24.43	-	-	24.90	-	-	-	-	25.14	-	25.13

Table 3. The similarity between the denoised color images and the ground-truth color images of BSD68 dataset for our network and existing algorithms measured by PSNR (in dB) reported for noise levels of σ=15, 25, and 50.

Noise	Methods							
Levels	CBM3D	MLP	TNRD	DnCNN	IrCNN	CNLNet	Ours-agnostic	Ours-specific
15	33.50	-	31.37	33.89	33.86	33.69	33.96	34.12
25	30.69	28.92	28.88	31.33	31.16	30.96	31.32	31.42
50	27.37	26.00	25.94	27.97	27.86	27.64	28.05	28.19

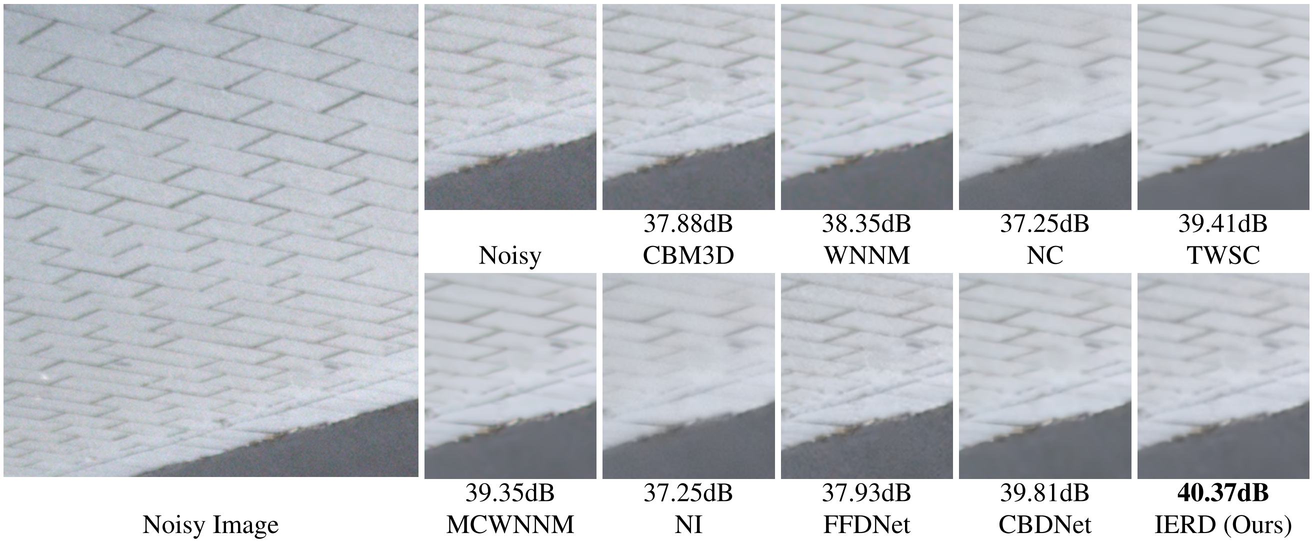

Figure 6. A real noisy example from DND dataset ([9]) for comparison of our method against the state-of-the-art algorithms.

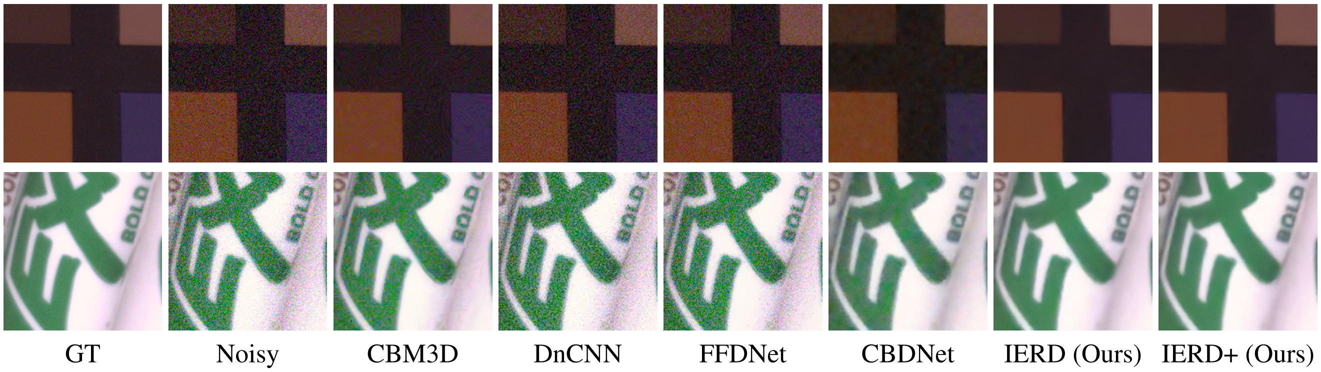

Figure 7. A few challenging examples from SSID dataset ([1]). Our method can restore true colors and remove noise.

- **CBSD68:** Color Berkely Segmentation Dataset abbreviated as CBSD68 ([12]) is composed of 68 images. We provide qualitative and quantitative results for both color images. To generate noisy test images, we corrupt the images by additive white Gaussian noise with standard deviations (std) of $\sigma_n = 15, 25, 50, 70$.

4. Synthetic Grayscale Image Denoising

In this section, first we demonstrate how our method performs on classical images and then report results on the BSD68 and real datasets.

4.1. Classical Images

For completeness, we compare our algorithm to several state-of-the-art denoising methods using grayscale classical images shown in Figure 2 and reported in Table 1.

In Table 1, we present the average PSNR for the denoised images. Our network is the best performer for almost all classical images except "Barbara". The reason for this may be the repetitive structures in the mentioned image, which makes it easy for BM3D ([3]) and WNNM ([4]) to find and employ patches with high similarity to the noisy input, hence providing better results.

Subsequently, we depict an example from the classical images. The visual quality of our recovered images, as shown in Figure 2, is better than all others. This also illustrates that our network restores aesthetically pleasing textures. Small and noticeable features restored by our network include the sharpness and the clarity of the subtle textures around the fore and hind wings, mouth, and antennas of the butterfly. Furthermore, a magnified view of the results in Figures 2 for methods such as [3, 15] and [6] show artifacts and failures in the smooth areas.

4.2. BSD68 Dataset

We present the average PSNR scores for the estimated denoised images in Table 2. The IRCNN ([15]) and DnCNN ([14]) network structures are similar, hence produce nearly similar results. On the other hand, our method reconstructs the images accurately, achieving higher PSNR, then completing methods on all four levels of noise. Furthermore, the difference in PSNR between our method and the state-of-the-art techniques at higher noise levels is higher.

For a comprehensive evaluation, we demonstrate the visual results on a selected grayscale image from [12] dataset in Figure 3. In our results; the image details are more similar to the ground-truth details, and our quantitative results are numerically higher than the others. Our method outperforms the second-best method by several orders of magnitude (PSNR is computed in the logarithmic scale). Also, note that the denoising results of other CNN based algorithms are comparable to each other.

4.3. Synthetic Color Image Denoising

For noisy color images, we train our network with the noisy RGB input patches with the corresponding clean ground-truth patches. We only modify the first and last convolution layer of the grayscale network to input and output three channels instead of one channel, keeping all other parameters same as the grayscale network.

We present the quantitative results in Table 3 and qualitative results in Figures 4 and 5 against benchmark methods including the latest CNN based state-of-the-art color image denoising techniques. It can be observed that our algorithm attains an improved average PSNR on all three different noise levels for the color version of the BSD68 dataset ([12]). As shown, our method restores true colors closer to their authentic values, while others fail and induce false colorizations in certain image regions. Furthermore, a close look reveals that our network reproduces the local texture with much fewer artifacts and sufficiently sharp details.

5. Real Image Denoising

We present more example on real image denoising from SSID [1] and DnD [9].

5.1. DnD

Figure 6 shows an image with bricks. The close shows that all other methods have some artifacts while our methods artifacts, which can be seen in the right bottom corner. It can be noted that our PSNR is higher among the competing methods.

5.2. SSID

Our second real image denoising example contains two pictures, as shown in Figure 7. Our IERD and IERD+ have no visual inconsistency and have smoothed the noise while the artifacts are visible for all other methods. The granular structures are present for state-of-the-art methods such as CBDNet and FFDNet, which employ specific techniques to remove the real noise.

6. Summary

In synthetic images case, we have provided ample examples and have shown that our network outperforms classical state-of-the-art denoising algorithms that are intended for use on natural images. Furthermore, we have compared against the current convolutional neural networks, both visually and numerically. Our network gain is about **0.1**dB on BSD68 dataset ([8]) and results are visually pleasing.

For the time being, our approach is only applicable to Gaussian noise removal. However, we would like to train our model with different noise types such as Poisson, astronomical *etc*. and examine its performance on these specific noise types. It should be noted that other state-of-the-art methods, for example, BM3D ([3]), WNNM ([4]) are only applicable to Gaussian noise and may not be readily adapted to handle different noise types.

All CNN approaches performance on images with regular and repeating structures such as "Barbara" is relatively less in terms of PSNR compared to classical denoising methods. This phenomenon is due to the design of traditional denoising methods to exploit the regular and repeating structures. To overcome this issue, either block-matching scheme can be incorporated into our CNN approach or relying on consolidating the outcome of various denoising algorithms with our CNN approach.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 4, 5

[2] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *TPAMI*, pages 1256–1272, 2017. 4

[3] Kostadin Dabov, Alessandro F., Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. pages 2080–2095, 2007. 4, 5, 6

[4] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, pages 2862–2869, 2014. 4, 5, 6

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[6] Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. *CVPR*, 2016. 4, 5

[7] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *NIPS*, pages 1673–1682, 2018. 4

[8] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 2, 3, 4, 6

[9] Tobias Plötz and Stefan Roth. Benchmarking denoising algorithms with real photographs. *CVPR*, 2017. 4, 5

[10] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *NIPS*, 2018. 4

[11] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein. Deep class-aware image denoising. In *International Conference on Sampling Theory and Applications*, 2017. 4

[12] Stefan Roth and Michael J Black. Fields of experts. *IJCV*, 2009. 3, 4, 5

[13] Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *ICCV*, 2017. 4

[14] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017. 4, 5

[15] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. *CVPR*, 2017. 4, 5

[16] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486, 2011. 4