LIDIA: Lightweight Learned Image Denoising with Instance Adaptation (Supplementary material)

Gregory Vaksman
Department of Computer Science,
Technion Institute of Technology
Technion City, Haifa 32000, Israel
grishav@campus.technion.ac.il

Michael Elad
Google Research
Mountain-View, California
melad@google.com

Peyman Milanfar
Google Research
Mountain-View, California
milanfar@google.com

1. Multi-Scale Patches

In Section 2.2 of our paper we describe the two scales used in handling the image patches. A visualization of the corresponding 1st and 2nd scale patches is depicted in Figure (1) , showing how the two are aligned.

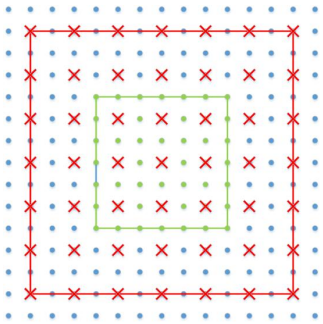


Figure 1: Visualization of the corresponding 1st and 2nd scale patches: Blue dots are the processed image pixels; the green square is a 1st scale patch, while the red square is its corresponding 2nd scale patch. Both are of size 7×7 pixels.

2. Network Architecture

Section 2.3 of our paper presents the architecture of the proposed filtering network. In this section we bring visualization of some elements of its architecture.

Figure 2 schematically shows concatenation of two TRT blocks. Since concatenation of two SL layers can be replaced by a single effective SL, due to their linearity, we remove one SL layer in any concatenation of two TRT-s. The TRT component without the second transform is denoted by TR, and when concatenating k TRT-s, the first $k-1$ blocks should be replaced by TR-s.

Another component of the filtering network, Aggregation block (AGG), is depicted in Figure 3. This block

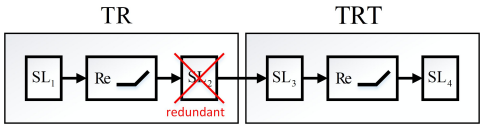


Figure 2: Concatenation of two TRTs: One SL layer is removed due to linearity, converting the first TRT into TR.

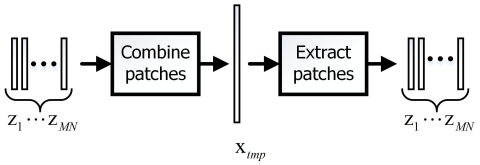


Figure 3: Aggregation (AGG) block.

imposes consistency between overlapping patches $\{\mathbf{z}_i\}$ by combining them to a temporary image $\mathbf{x}_{tmp}$ using plain averaging, and extracting them back from the obtained image. Sizes of the network layers are listed in Tables 1 and 2.

3. Denoising with Known Noise Level

In this section we bring additional examples of denoising results. Figures 6, 7, 8 and 9 present examples of denoising results. Since our architecture is related to both BM3D and NLNet, we focus on qualitative comparisons with these algorithms. For all noise levels our results are significantly sharper, contain less artifacts and preserve more details than those of BM3D. In comparison to NLNet, our method is significantly better in high noise levels due to our multi-scale treatment, recovering large and repeating elements, as shown in Figure 6p. In fact our algorithm manages to recover repeating elements better than all methods presented in Figure 6 except NLRN. In addition, in cases of high noise levels, the multi-scale treatment allows handling smooth areas with less artifacts than NLNet, as one can see from the results in Figure 7p and 7o. In medium noise levels, our al-

	TR_0 / $TR_0^{(2)}$	TBR_1 / $TBR_1^{(2)}$	T_{pre} / $T_{pre}^{(2)}$	TR_{post} / $TR_{post}^{(2)}$	TBR_2	TBR_3	T_4
W_1	49×64	64×64	64×49	49×64	64×64	64×64	64×49
W_2	14×14	14×14	14×1	1×14	56×56	56×56	56×1

Table 1: Size of the network layers for grayscale image denoising

	TR_0 / $TR_0^{(2)}$	TBR_1 / $TBR_1^{(2)}$	T_{pre} / $T_{pre}^{(2)}$	TR_{post} / $TR_{post}^{(2)}$	TBR_2	TBR_3	T_4
W_1	75×80	80×80	80×75	75×80	80×80	80×80	80×75
W_2	14×14	14×14	14×1	1×14	56×56	56×56	56×1

Table 2: Size of the network layers for color image denoising

gorithm recovers more details, while NLNet tends to oversmooth the recovered image. For example, see the Elephant skin in Figure 8 and the mountain glacier in Figure 9.

Figures 10, 11 and 12 present examples of denoising results for color images that show that our method, C-LIDIA, handles low frequency regions better than CBM3D and CNLNet, due to its multi-scale treatment.

4. Comparison between LIDIA and LIDIA-S

This section presents visual comparison between LIDIA (our regular architecture) and LIDIA-S (a further simplified version of our architecture) denoising results. Examples depicted in Figure 13 show that the visual quality gap between the two is marginal.

5. Network Adaptation

Figure 4 presents a graph of the PSNR (denoising performance) vs. the number of adaptation training epochs. This graph corresponds to the text image adaptation experiment using a single training image. As can be seen, after 3 epochs we get a 2.8dB improvement. Continuation of this training leads to more than 4dB performance boost.

Still on the same topic, Figure 5 shows the training images used for the external adaptation experiments, and Figure 14 presents example of external adaptation for an astronomical image. Table 3 contains values of minimum, maximum and median improvement achieved by internal adaptation on Urban100 and BSD68 image sets.

Code that reproduces results reported in the article will be realised soon.

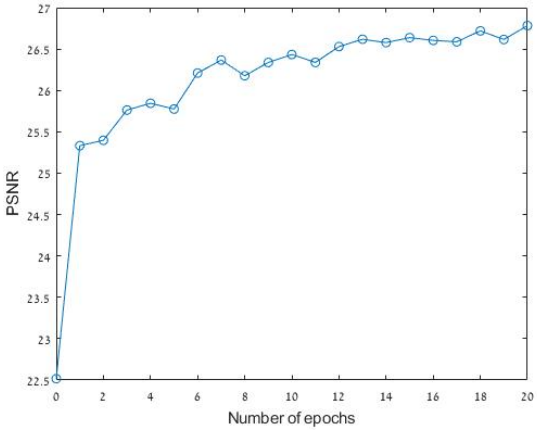


Figure 4: PSNR vs. number of epochs for the text image adaptation experiment.

Image set	Max	Min	Average	Median
Urban100	1.09	0.01	0.29	0.21
BSD68	0.35	-0.02	0.05	0.04

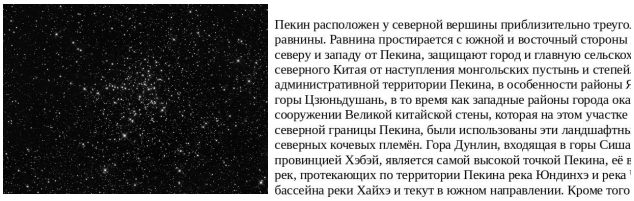
Table 3: PSNR improvement obtained as a result of applying internal adaptation.

References

[1] Yunjin Chen, Wei Yu, and Thomas Pock. On learning optimized reaction diffusion processes for effective image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5261–5269, 2015. 4, 5, 6, 7

[2] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In *2007 IEEE International Conference on Image Processing*, volume 1, pages I–313. IEEE, 2007. 8, 9, 10

[3] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and



(a) (b)

Figure 5: Training images for the network adaptation experiments: (a) astronomical (800×570), (b) text (703×354).

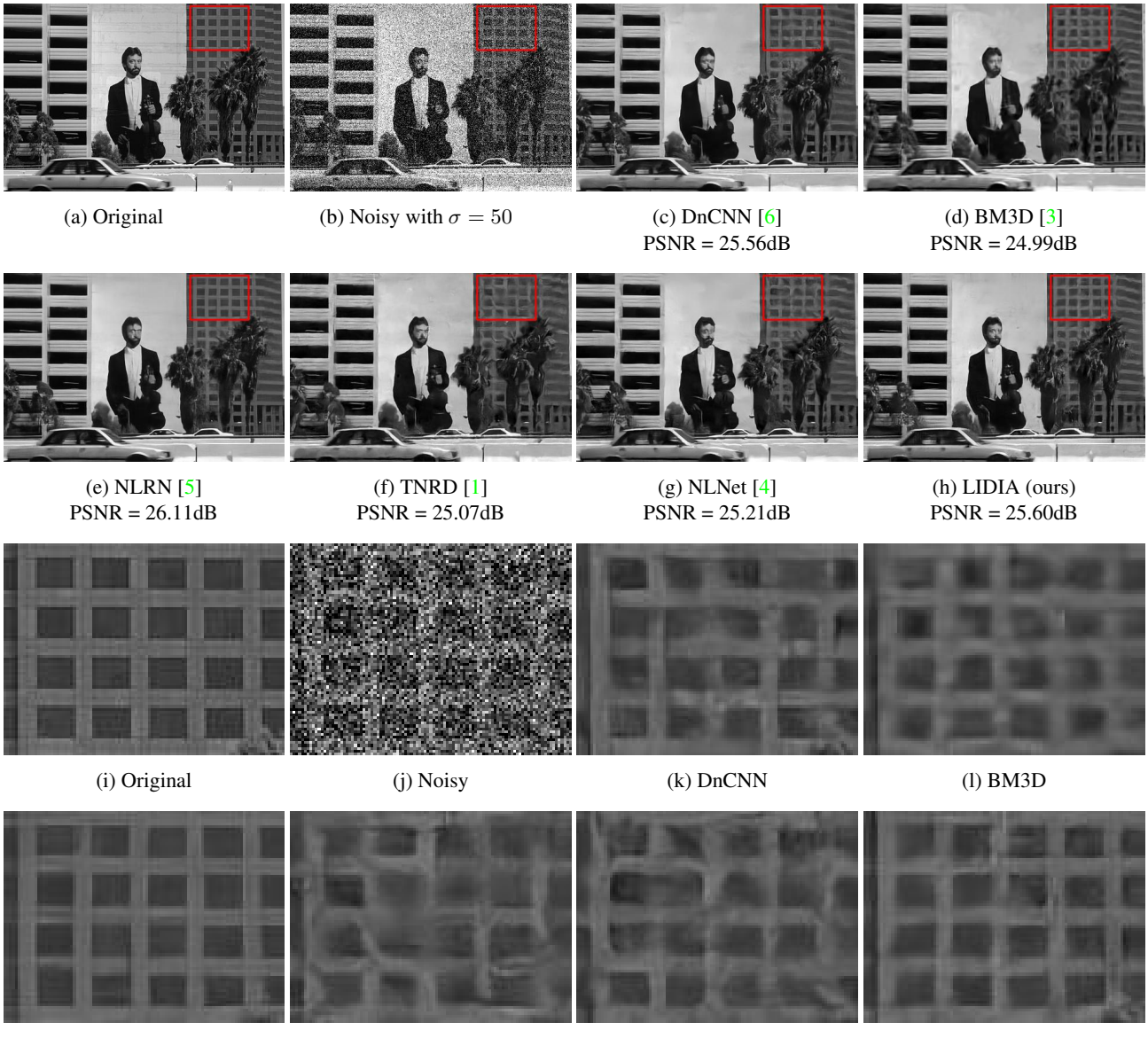
Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 4, 5, 6, 7

[4] Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3587–3596, 2017. 4, 5, 6, 7, 8, 9, 10

[5] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems*, pages 1673–1682, 2018. 4, 5, 6, 7

[6] 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. 4, 5, 6, 7, 8, 9, 10, 11

[7] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 8, 9, 10



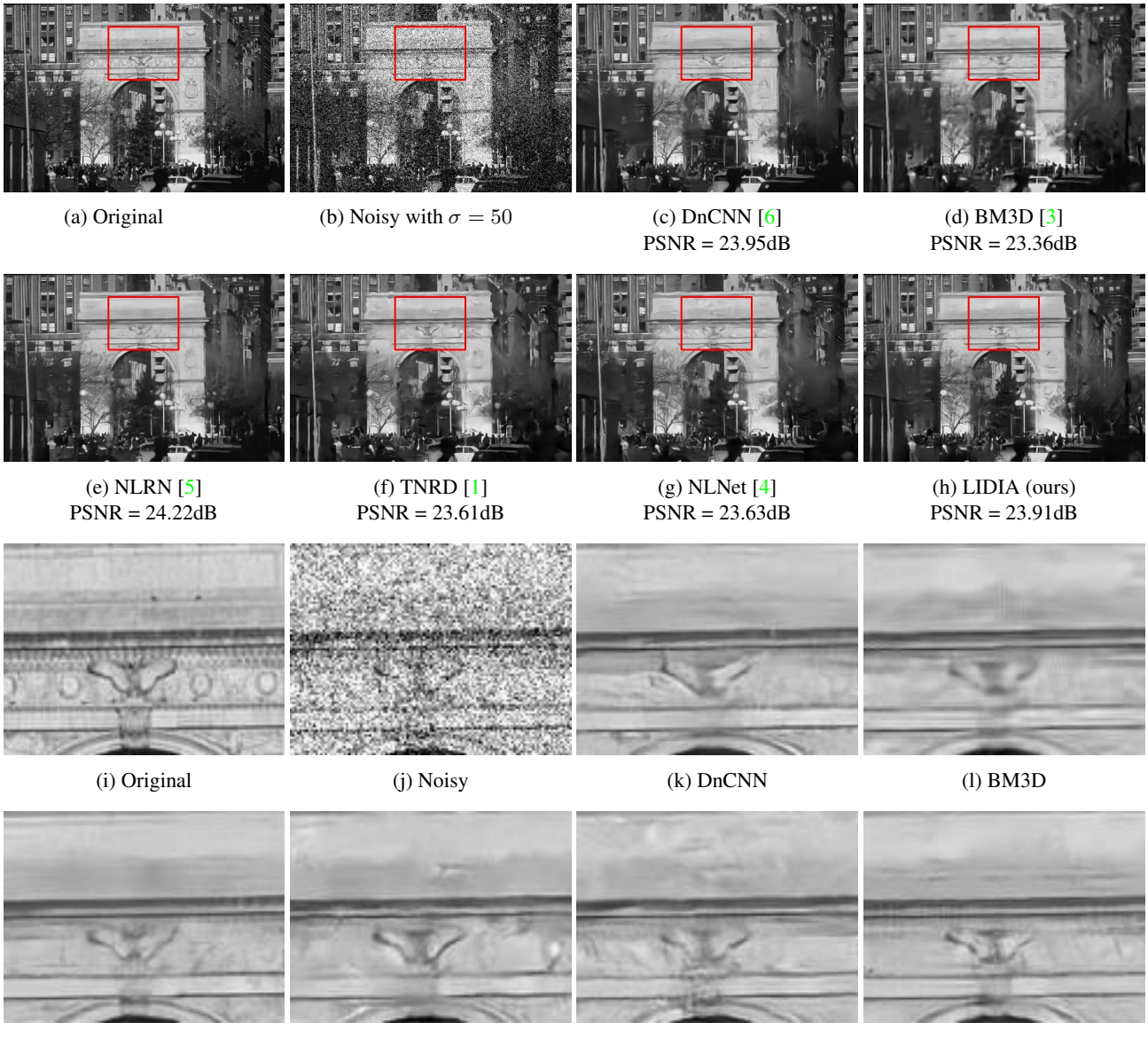
(a) Original (b) Noisy with $\sigma = 50$ (c) DnCNN [6] PSNR = 25.56dB (d) BM3D [3] PSNR = 24.99dB

(e) NLRN [5] PSNR = 26.11dB (f) TNRD [1] PSNR = 25.07dB (g) NLNet [4] PSNR = 25.21dB (h) LIDIA (ours) PSNR = 25.60dB

(i) Original (j) Noisy (k) DnCNN (l) BM3D

(m) NLRN (n) TNRD (o) NLNet (p) LIDIA (ours)

Figure 6: Denoising example with $\sigma = 50$.



(a) Original (b) Noisy with $\sigma = 50$ (c) DnCNN [6] PSNR = 23.95dB (d) BM3D [3] PSNR = 23.36dB

(e) NLRN [5] PSNR = 24.22dB (f) TNRD [1] PSNR = 23.61dB (g) NLNet [4] PSNR = 23.63dB (h) LIDIA (ours) PSNR = 23.91dB

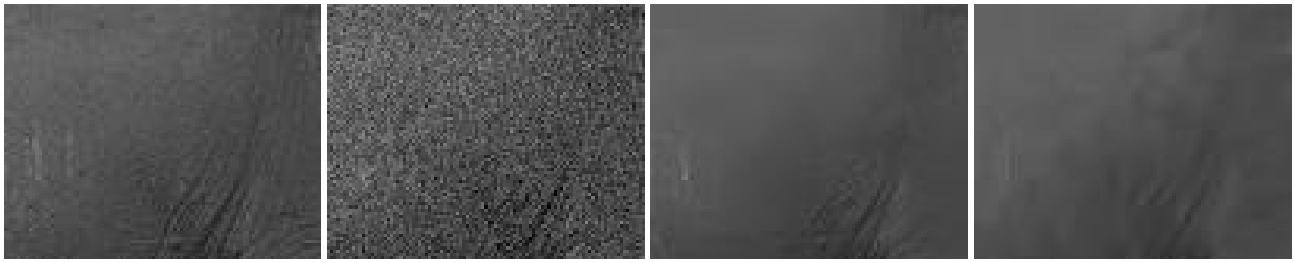
(i) Original (j) Noisy (k) DnCNN (l) BM3D

(m) NLRN (n) TNRD (o) NLNet (p) LIDIA (ours)

Figure 7: Denoising example with $\sigma = 50$.

(a) Original (b) Noisy with $\sigma = 15$ (c) DnCNN [6] PSNR = 32.32dB (d) BM3D [3] PSNR = 31.70dB

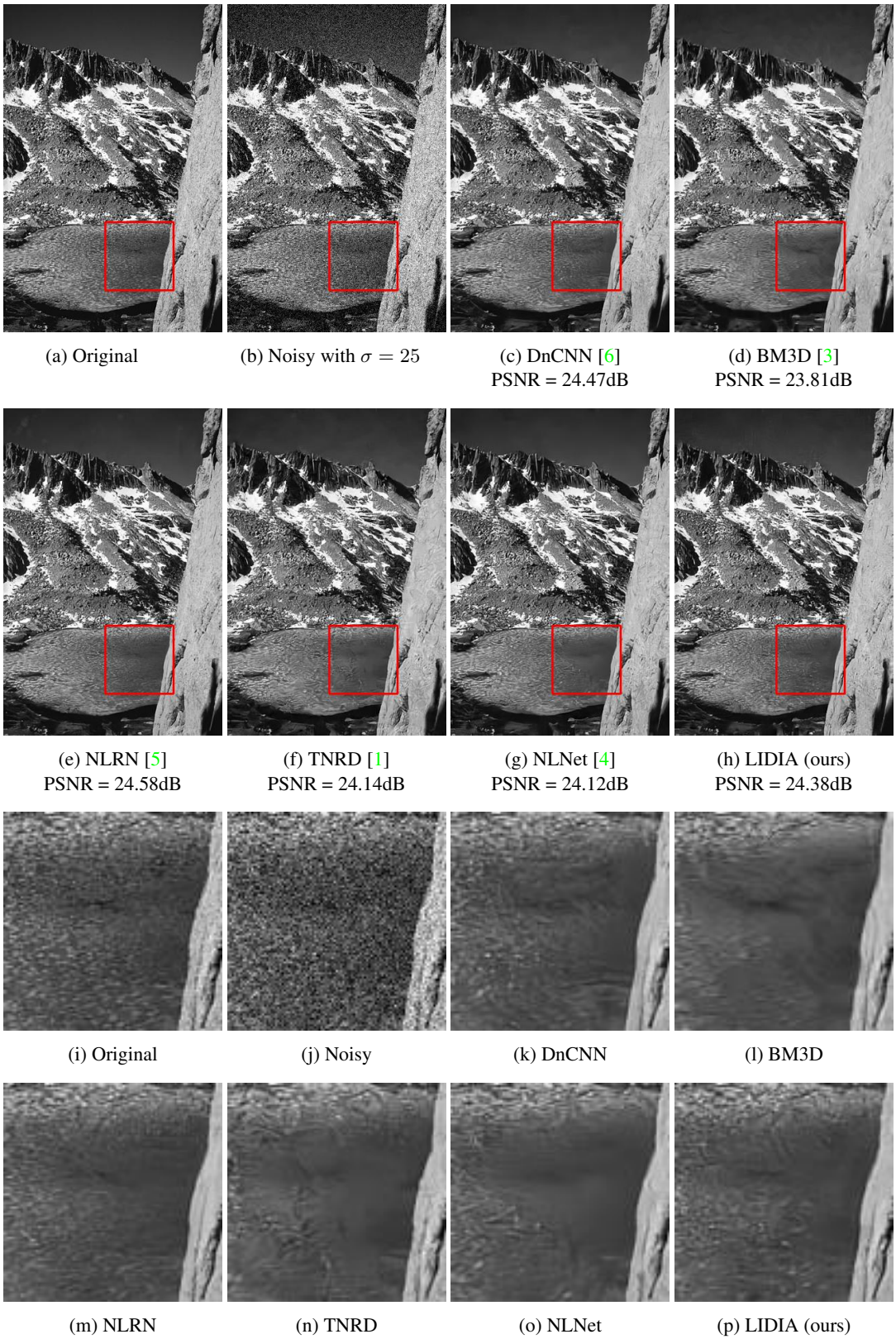
(e) NLRN [5] PSNR = 32.47dB (f) TNRD [1] PSNR = 32.05dB (g) NLNet [4] PSNR = 32.14dB (h) LIDIA (ours) PSNR = 32.30dB



(i) Original (j) Noisy (k) DnCNN (l) BM3D

(m) NLRN (n) TNRD (o) NLNet (p) LIDIA (ours)

Figure 8: Denoising example with $\sigma = 15$.



(a) Original (b) Noisy with $\sigma = 25$ (c) DnCNN [6] PSNR = 24.47dB (d) BM3D [3] PSNR = 23.81dB

(e) NLRN [5] PSNR = 24.58dB (f) TNRD [1] PSNR = 24.14dB (g) NLNet [4] PSNR = 24.12dB (h) LIDIA (ours) PSNR = 24.38dB

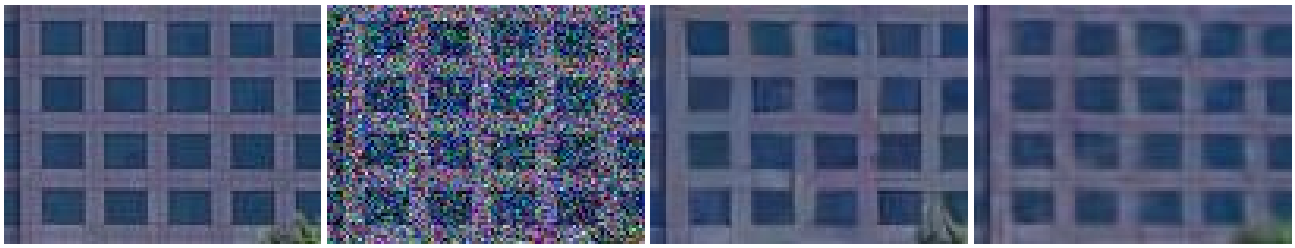
(i) Original (j) Noisy (k) DnCNN (l) BM3D

(m) NLRN (n) TNRD (o) NLNet (p) LIDIA (ours)

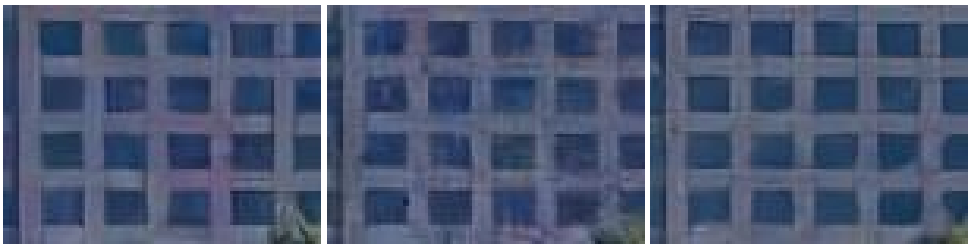
Figure 9: Denoising example with $\sigma = 25$.

(a) Original (b) Noisy with $\sigma = 50$ (c) CDnCNN [6] PSNR = 27.81dB (d) CBM3D [2] PSNR = 26.98dB

(e) CFFDNet [7] PSNR = 27.72dB (f) CNLNet [4] PSNR = 27.41dB (g) C-LIDIA (ours) PSNR = 27.79dB

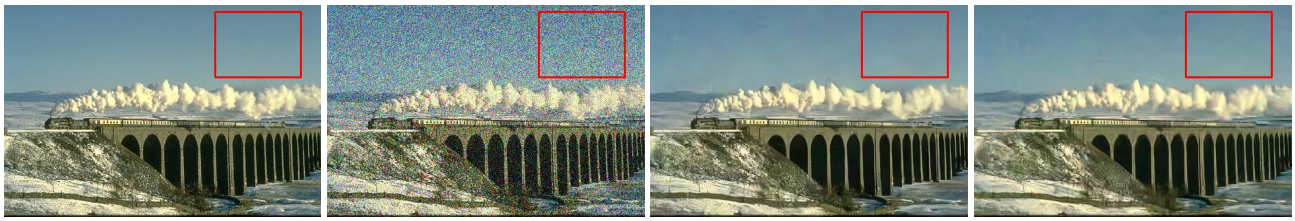


(h) Original (i) Noisy (j) CDnCNN (k) CBM3D



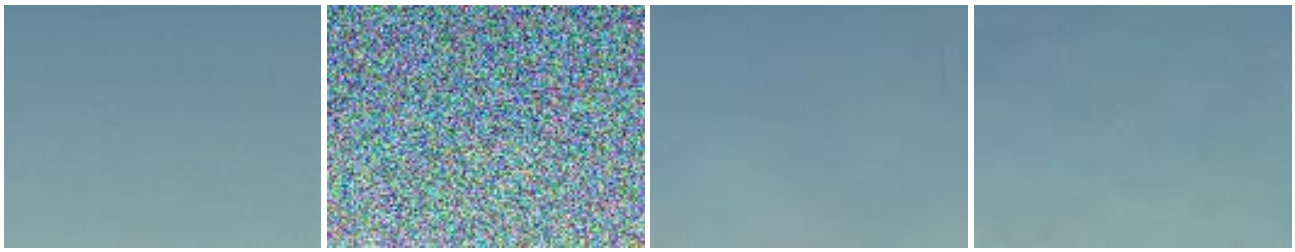
(l) CFFDNet (m) CNLNet (n) C-LIDIA (ours)

Figure 10: Color image denoising example with $\sigma = 50$.



(a) Original (b) Noisy with $\sigma = 50$ (c) CDnCNN [6] PSNR = 26.85dB (d) CBM3D [2] PSNR = 26.23dB

(e) CFFDNet [7] PSNR = 26.78dB (f) CNLNet [4] PSNR = 26.62dB (g) C-LIDIA (ours) PSNR = 26.90dB



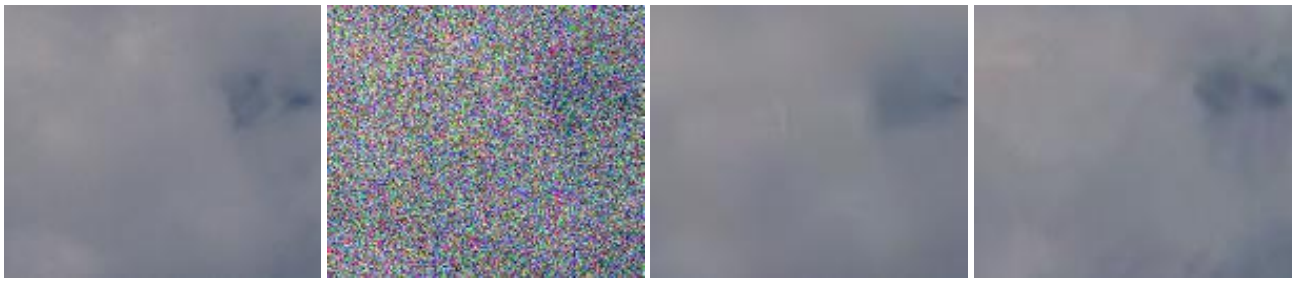
(h) Original (i) Noisy (j) CDnCNN (k) CBM3D

(l) CFFDNet (m) CNLNet (n) C-LIDIA (ours)

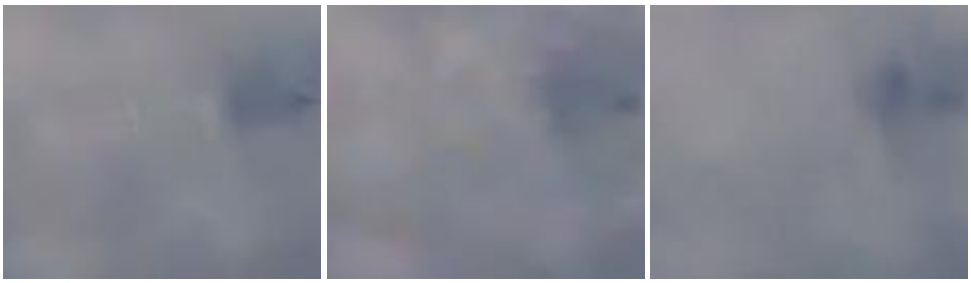
Figure 11: Color image denoising example with $\sigma = 50$.

(a) Original (b) Noisy with $\sigma = 50$ (c) CDnCNN [6] PSNR = 36.68dB (d) CBM3D [2] PSNR = 35.51dB

(e) CFFDNet [7] PSNR = 36.62dB (f) CNLNet [4] PSNR = 35.98dB (g) C-LIDIA (ours) PSNR = 36.56dB

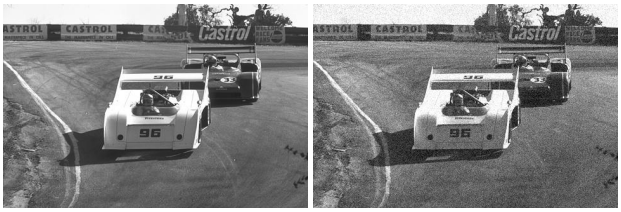


(h) Original (i) Noisy (j) CDnCNN (k) CBM3D



(l) CFFDNet (m) CNLNet (n) C-LIDIA (ours)

Figure 12: Color Image denoising example with $\sigma = 50$.



(a) Original (b) Noisy with $\sigma = 15$

(c) LIDIA
PSNR = 29.38dB

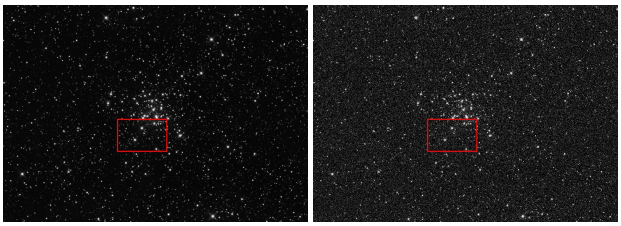
(d) LIDIA-S
PSNR = 29.34dB

(e) Original (f) Noisy with $\sigma = 25$

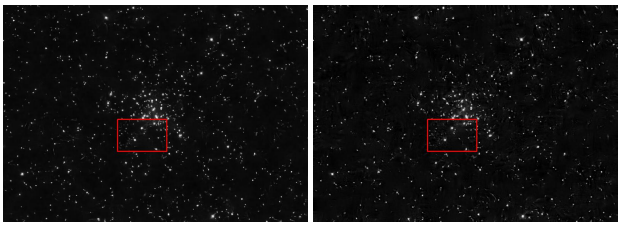
(g) LIDIA
PSNR = 28.96dB

(h) LIDIA-S
PSNR = 28.89dB

Figure 13: Comparison between full and small versions of the LIDIA network.

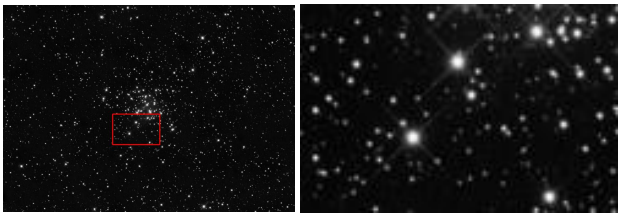


(a) Clean astronomical (800×570) (b) Noisy with $\sigma = 50$



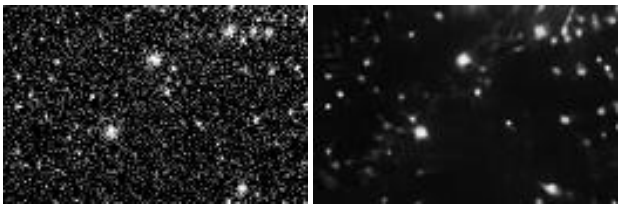
(c) CDnCNN [6]
PSNR = 27.05dB

(d) LIDIA (before adaptation)
PSNR = 26.44dB

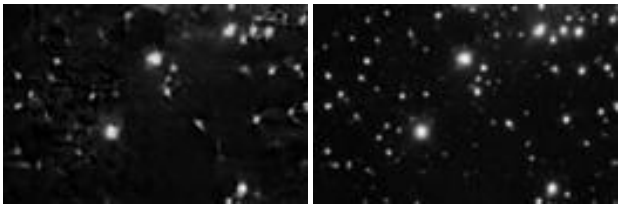


(e) LIDIA (after adaptation)
PSNR = 28.04dB

(f) Clean



(g) Noisy (h) DnCNN



(i) LIDIA (ours)
(before adaptation)

(j) LIDIA (ours)
(after adaptation)

Figure 14: An example of external adaptation for an astronomical image.