Multi-level Wavelet-CNN for Image Restoration (Supplementary Material)

Pengju Liu[1], Hongzhi Zhang[1], Kai Zhang[1], Liang Lin[2], Mengwang Zuo[1]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[2]School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China
lpj008@126.com, zhanghz0451@gmail.com, linliang@ieee.org, cskaizhang@gmail.com, cswmzuo@gmail.com

1. Contents

The following items are included in our supplementary material:

- More details on discussion about Haar wavelet connection to dilated filtering of MWCNN in Section 2.
- More details on discussion about training datasets in Section 3.
- Visual Comparisons of MWCNN variants to verify the effectiveness of embedded wavelet in Section 4.
- More visual comparisons of three tasks about MWCNN in Section 5.

2. More Discussion

As described in the paper, there exists a strong connection between dilated filtering and MWCNN. Here, we give the whole discussion about analyzing the connection between dilated filtering and MWCNN for $(\mathbf{x} \otimes_2 \mathbf{k})(2i-1, 2j-1)$, $(\mathbf{x} \otimes_2 \mathbf{k})(2i-1, 2j)$, $(\mathbf{x} \otimes_2 \mathbf{k})(2i, 2j-1)$, $(\mathbf{x} \otimes_2 \mathbf{k})(2i, 2j)$, where $\mathbf{x}$ is the input image with size of $m \times n$, and $\mathbf{k}$ denotes the 3×3 convolution kernel.

According to the theory of Haar transform [10], the (i, j)-th value of $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$ after 2D Haar transform can be written as

$$\begin{cases} \mathbf{x}_1(i,j) = \mathbf{x}(2i-1,2j-1) + \mathbf{x}(2i-1,2j) \\ \qquad\qquad +\mathbf{x}(2i,2j-1) + \mathbf{x}(2i,2j), \\ \mathbf{x}_2(i,j) = -\mathbf{x}(2i-1,2j-1) - \mathbf{x}(2i-1,2j) \\ \qquad\qquad +\mathbf{x}(2i,2j-1) + \mathbf{x}(2i,2j), \\ \mathbf{x}_3(i,j) = -\mathbf{x}(2i-1,2j-1) + \mathbf{x}(2i-1,2j) \\ \qquad\qquad -\mathbf{x}(2i,2j-1) + \mathbf{x}(2i,2j), \\ \mathbf{x}_4(i,j) = \mathbf{x}(2i-1,2j-1) - \mathbf{x}(2i-1,2j) \\ \qquad\qquad -\mathbf{x}(2i,2j-1) + \mathbf{x}(2i,2j), \end{cases} \tag{1}$$

It can be easily found that the coefficients of Eqn.(1) are the same with the weights of four pass-filters, *i.e.* $\mathbf{f}_{LL}$, $\mathbf{f}_{LH}$, $\mathbf{f}_{HL}$ and $\mathbf{f}_{HH}$. Due to the invertibility of 2D Haar transform, the weights of inverse Haar transform are utilized to reconstruct signals. Then we can get the following equations

$$\begin{cases} \mathbf{x}(2i\text{–}1, 2j\text{–}1) = (\mathbf{x}_1(i,j) - \mathbf{x}_2(i,j) - \mathbf{x}_3(i,j) + \mathbf{x}_4(i,j))/4, \\ \mathbf{x}(2i\text{–}1, 2j) \quad = (\mathbf{x}_1(i,j) - \mathbf{x}_2(i,j) + \mathbf{x}_3(i,j) - \mathbf{x}_4(i,j))/4, \\ \mathbf{x}(2i\text{–}1, 2j) \quad = (\mathbf{x}_1(i,j) + \mathbf{x}_2(i,j) - \mathbf{x}_3(i,j) - \mathbf{x}_4(i,j))/4, \\ \mathbf{x}(2i, 2j) \qquad = (\mathbf{x}_1(i,j) + \mathbf{x}_2(i,j) + \mathbf{x}_3(i,j) + \mathbf{x}_4(i,j))/4. \end{cases} \tag{2}$$

The dilated filtering with factor 2 on the position $(2i-1, 2j-1)$, $(2i-1, 2j)$, $(2i, 2j-1)$ and $(2i, 2j)$ of $\mathbf{x}$ can be written as

$$\begin{cases} (\mathbf{x} \otimes_2 \mathbf{k})(2i-1, 2j-1) = \sum_{\substack{p+2s=2i-1, \\ q+2t=2j-1}} \mathbf{x}(p,q)\mathbf{k}(s,t), \\ (\mathbf{x} \otimes_2 \mathbf{k})(2i-1, 2j) = \sum_{\substack{p+2s=2i-1, \\ q+2t=2j}} \mathbf{x}(p,q)\mathbf{k}(s,t), \\ (\mathbf{x} \otimes_2 \mathbf{k})(2i, 2j-1) = \sum_{\substack{p+2s=2i, \\ q+2t=2j-1}} \mathbf{x}(p,q)\mathbf{k}(s,t), \\ (\mathbf{x} \otimes_2 \mathbf{k})(2i, 2j) = \sum_{\substack{p+2s=2i, \\ q+2t=2j}} \mathbf{x}(p,q)\mathbf{k}(s,t). \end{cases} \tag{3}$$

Actually, it also be obtained by using the 3×3 convolution with the subband images,

$$\begin{cases} (\mathbf{x} \otimes_2 \mathbf{k})(2i\text{–}1, 2j\text{–}1) = ((\mathbf{x}_1 - \mathbf{x}_2 - \mathbf{x}_3 + \mathbf{x}_4) \otimes \mathbf{k})(i,j)/4, \\ (\mathbf{x} \otimes_2 \mathbf{k})(2i-1, 2j) = ((\mathbf{x}_1 - \mathbf{x}_2 + \mathbf{x}_3 - \mathbf{x}_4) \otimes \mathbf{k})(i,j)/4, \\ (\mathbf{x} \otimes_2 \mathbf{k})(2i-1, 2j) = ((\mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 - \mathbf{x}_4) \otimes \mathbf{k})(i,j)/4, \\ (\mathbf{x} \otimes_2 \mathbf{k})(2i, 2j) = ((\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 + \mathbf{x}_4) \otimes \mathbf{k})(i,j)/4. \end{cases} \tag{4}$$

Therefore, the 3×3 dilated convolution on $\mathbf{x}$ can be treated as a special case of $4 \times 3 \times 3$ convolution on the subband images.

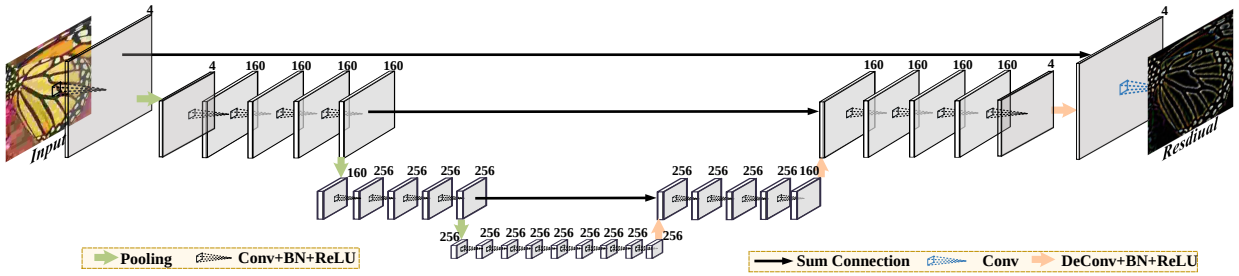

Figure 1: U-Net(S) architecture. It is designed as similar architecture with our MWCNN. The difference is that the input is first mapped to 4 feature maps and then max pooling is adopted.

Table 1: Average PSNR(dB)/SSIM results of the competing methods for image denoising with noise levels $\sigma = 15$, 25 and 50 on datasets Set14, BSD68 and Urban100. Red color indicates the best performance.

Dataset	σ	BM3D [4]	TNRD [3]	DnCNN [19]	IRCNN [20]	RED30 [11]	MemNet [15]	MWCNN(S)	MWCNN
Set12	15	32.37 / 0.8952	32.50 / 0.8962	32.86 / 0.9027	32.77 / 0.9008	-	-	32.98 / 0.9053	33.15 / 0.9088
	25	29.97 / 0.8505	30.05 / 0.8515	30.44 / 0.8618	30.38 / 0.8601	-	-	30.62 / 0.8665	30.79 / 0.8711
	50	26.72 / 0.7676	26.82 / 0.7677	27.18 / 0.7827	27.14 / 0.7804	27.34 / 0.7897	27.38 / 0.7931	27.56 / 0.7978	27.74 / 0.8056
BSD68	15	31.08 / 0.8722	31.42 / 0.8822	31.73 / 0.8906	31.63 / 0.8881	-	-	31.80 / 0.8933	31.86 / 0.8947
	25	28.57 / 0.8017	28.92 / 0.8148	29.23 / 0.8278	29.15 / 0.8249	-	-	29.34 / 0.8330	29.41 / 0.8360
	50	25.62 / 0.6869	25.97 / 0.7021	26.23 / 0.7189	26.19 / 0.7171	26.35 / 0.7245	26.35 / 0.7294	26.43 / 0.7302	26.53 / 0.7366
Urban100	15	32.34 / 0.9220	31.98 / 0.9187	32.67 / 0.9250	32.49 / 0.9244	-	-	32.73 / 0.9263	33.17 / 0.9357
	25	29.70 / 0.8777	29.29 / 0.8731	29.97 / 0.8792	29.82 / 0.8839	-	-	30.23 / 0.8940	30.66 / 0.9026
	50	25.94 / 0.7791	25.71 / 0.7756	26.28 / 0.7869	26.14 / 0.7927	26.48 / 0.7991	26.64 / 0.8024	26.87 / 0.8176	27.42 / 0.8371

Table 2: Average PSNR(dB) / SSIM results of the competing methods for SISR with scale factors $S = 2$, 3 and 4 on datasets Set5, Set14, BSD100 and Urban100. Red color indicates the best performance.

Dataset	S	VDSR [6]	DnCNN [19]	RED30 [11]	LapSRN [7]	DRRN [14]	MemNet [15]	WaveResNet [2]	MWCNN(S)	MWCNN
Set5	×2	37.53 / 0.9587	37.58 / 0.9593	37.66 / 0.9599	37.52 / 0.9590	37.74 / 0.9591	37.78 / 0.9597	37.57 / 0.9586	37.71 / 0.9608	37.91 / 0.9600
	×3	33.66 / 0.9213	33.75 / 0.9222	33.82 / 0.9230	-	34.03 / 0.9244	34.09 / 0.9248	33.86 / 0.9228	33.91 / 0.9258	34.18 / 0.9272
	×4	31.35 / 0.8838	31.40 / 0.8845	31.51 / 0.8869	31.54 / 0.8850	31.68 / 0.8888	31.74 / 0.8893	31.52 / 0.8864	31.76 / 0.8619	32.12 / 0.8941
Set14	×2	33.03 / 0.9124	33.04 / 0.9118	32.94 / 0.9144	33.08 / 0.9130	33.23 / 0.9136	33.28 / 0.9142	33.09 / 0.9129	33.21 / 0.9163	33.70 / 0.9182
	×3	29.77 / 0.8314	29.76 / 0.8349	29.61 / 0.8341	-	29.96 / 0.8349	30.00 / 0.8350	29.88 / 0.8331	29.85 / 0.8385	30.16 / 0.8414
	×4	28.01 / 0.7674	28.02 / 0.7670	27.86 / 0.7718	28.19 / 0.7720	28.21 / 0.7720	28.26 / 0.7723	28.11 / 0.7699	28.01 / 0.7778	28.41 / 0.7816
BSD100	×2	31.90 / 0.8960	31.85 / 0.8942	31.98 / 0.8974	31.80 / 0.8950	32.05 / 0.8973	32.08 / 0.8978	31.92 / 0.8965	32.15 / 0.8995	32.23 / 0.8999
	×3	28.82 / 0.7976	28.80 / 0.7963	28.92 / 0.7993	-	28.95 / 0.8004	28.96 / 0.8001	28.86 / 0.7987	29.10 / 0.8091	29.12 / 0.8060
	×4	27.29 / 0.7251	27.23 / 0.7233	27.39 / 0.7286	27.32 / 0.7280	27.38 / 0.7284	27.40 / 0.7281	27.32 / 0.7266	27.45 / 0.7312	27.62 / 0.7355
Urban100	×2	30.76 / 0.9140	30.75 / 0.9133	30.91 / 0.9159	30.41 / 0.9100	31.23 / 0.9188	31.31 / 0.9195	30.96 / 0.9169	31.62 / 0.9238	32.30 / 0.9296
	×3	27.14 / 0.8279	27.15 / 0.8276	27.31 / 0.8303	-	27.53 / 0.8378	27.56 / 0.8376	27.28 / 0.8334	27.74 / 0.8438	28.13 / 0.8514
	×4	25.18 / 0.7524	25.20 / 0.7521	25.35 / 0.7587	25.21 / 0.7560	25.44 / 0.7638	25.50 / 0.7630	25.36 / 0.7614	25.85 / 0.7797	26.27 / 0.7890

Table 3: Average PSNR(dB) / SSIM results of the competing methods for JPEG image artifacts removal with quality factors $Q = 10$, 20, 30 and 40 on datasets Classic5 and LIVE1. Red color indicates the best performance.

Dataset	Q	JPEG	ARCNN [5]	TNRD [3]	DnCNN [19]	MemNet [15]	MWCNN(S)	MWCNN
Classic5	10	27.82 / 0.7595	29.03 / 0.7929	29.28 / 0.7992	29.40 / 0.8026	29.69 / 0.8107	29.84 / 0.8155	30.01 / 0.8195
	20	30.12 / 0.8344	31.15 / 0.8517	31.47 / 0.8576	31.63 / 0.8610	31.90 / 0.8658	32.00 / 0.8678	32.16 / 0.8701
	30	31.48 / 0.8744	32.51 / 0.8806	32.78 / 0.8837	32.91 / 0.8861	-	33.21 / 0.8910	33.43 / 0.8930
	40	32.43 / 0.8911	33.34 / 0.8953	-	33.77 / 0.9003	-	34.11 / 0.9044	34.27 / 0.9061
LIVE1	10	27.77 / 0.7730	28.96 / 0.8076	29.15 / 0.8111	29.19 / 0.8123	29.45 / 0.8193	29.53 / 0.8219	29.69 / 0.8254
	20	30.07 / 0.8512	31.29 / 0.8733	31.46 / 0.8769	31.59 / 0.8802	31.83 / 0.8846	31.92 / 0.8864	32.04 / 0.8885
	30	31.41 / 0.9000	32.67 / 0.9043	32.84 / 0.9059	32.98 / 0.9090	-	33.21 / 0.9132	33.45 / 0.9153
	40	32.35 / 0.9173	33.63 / 0.9198	-	33.96 / 0.9247	-	34.29 / 0.9285	34.45 / 0.9301

3. Training Small Dataset *vs*. Big Dataset

Our MWCNN is first training on small dataset following DnCNN *et al*. [19], *e.g*., it contains 91 images from Yang *et al*. [17] and 200 images from Berkeley Segmentation Dataset (BSD) [12] for SISR task. Then, a large training set is constructed by using images from three dataset, *i.e*. BSD, Waterloo Exploration Database (WED) [9], and DIV2K [1]. Table 1,2,3 compares MWCNN with SOTAs using the same training set in DnCNN [19], which is denoted as MWCNN(S). MWCNN(S) achieves favorable results over most competing methods with SOTAs and is much efficient than the start-of-art methods, such as DRRN, MemNet, indicating the effectiveness of embedded wavelet. The results also indicate that enlarging training set likewise ef-

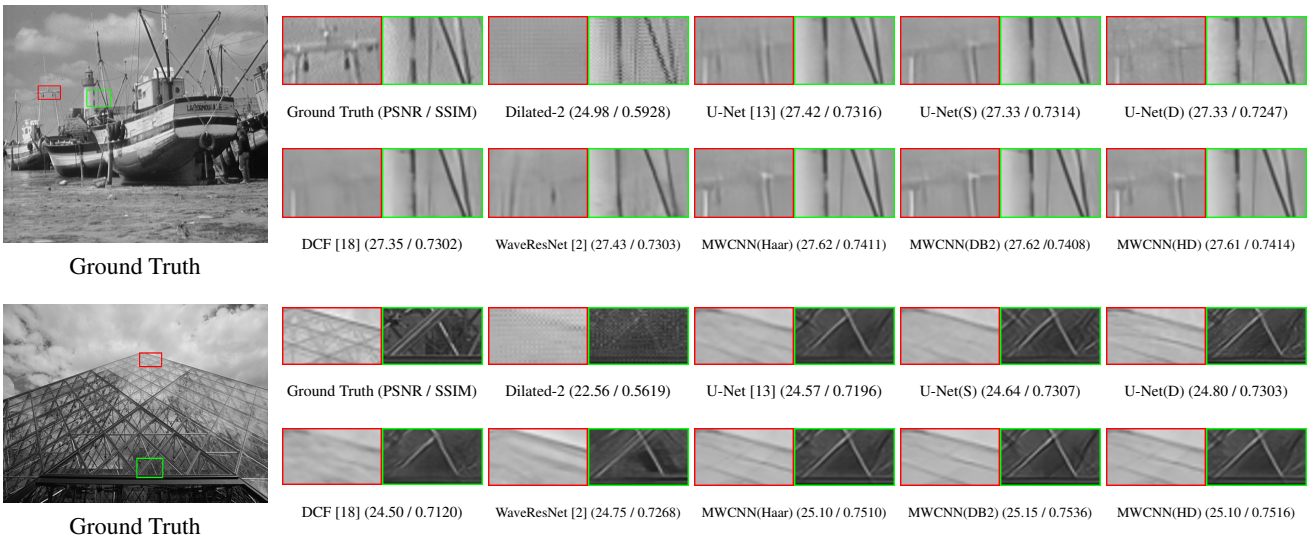

Figure 2: *Image denoising with* $\sigma = 50$ visual results of MWCNN variants, including "10" (Set12) and "*Test044*" (BSD68).

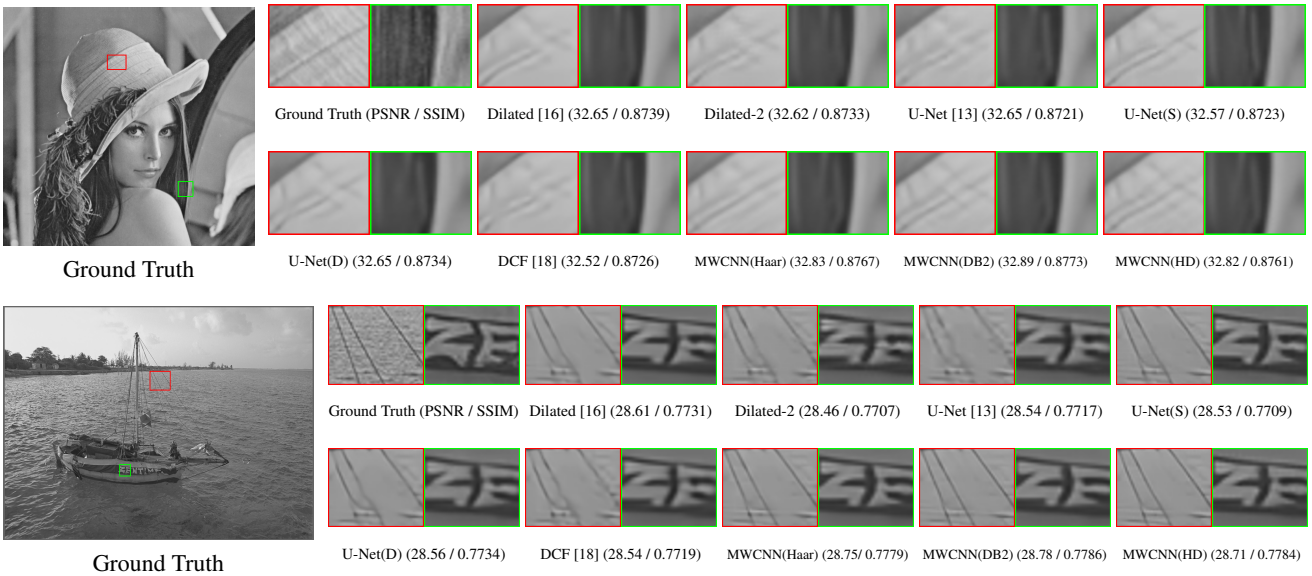

Figure 3: *JPEG image artifacts removal* visual results of MWCNN variants with $PC = 10$, including "*lena*" (Classic5) and "*salting1*" (LIVE1).

fectively improves performance.

4. Visual Comparisons of MWCNN variants

As shown in Figure 1, U-Net(S) is designed as similar architecture with our MWCNN for fair comparison. Due to decimation of pooling operator, the input is first mapped to 4 feature maps and then max pooling is adopted. Analogously, U-Net and U-Net(D) are designed and trained for comparison. To verify the effectiveness of embedded wavelet, we provide visual results of MWCNN variants. Figure 2 shows image denoising results with noise level 50, including *10* on Set12 and *Test044* on BSD68. And Figure 3 shows JPEG image artifacts removal results with quality 10. In terms of visual comparison with U-Net and Dilated CNN, our MWCNN exhibits more detailed textures, indicating that embedded wavelet benefits the separation of signal/noise and recovers details from degraded image. Moreover, the visual results of Dilated-2 on image denoising task are still blurring in Fifure 2, indicating that the gridding effect with the sparse sampling and inconsistence of local information authentically has adverse influence on restoration performance.

5. More Visual Comparisons of Three Tasks

In this section, more visual results on the three tasks are posted. For image denoising, we compare with six compet-

ing denoising methods, *i.e*., BM3D [4], TNRD [3], DnCNN [19], IRCNN [20], RED30 [11], and MemNet [15]. As shown in Figure 4, image denoising results with noise level 50, including *02*, *04* and *12* on Set12, *Test039* on BSD68, *img011* and *img044* on Urban100, are provided for comparison. In SISR experiments, our MWCNN is compared with seven CNN-based SISR methods, including VDSR [6], DnCNN [19], RED30 [11], SRResNet [8], LapSRN [7], DRRN [14], and MemNet [15]. In Figure 5, we show the visual comparisons on Set5, Set14 and BSD100 with upscaling factor of $4\times$. For JPEG image artifacts removal visual, our MWCNN is compared with four competing methods, *i.e*., ARCNN [5], TNRD [3], DnCNN [19], and MemNet [15]. Figure 6 and Figure 7 show JPEG image artifacts removal visual results with two qualities, *e.g*. 10 and 20. Intuitively, our MWCNN has strong ability to recover more detailed textures and sharp edges.

References

[1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1122–1131. IEEE, 2017.

[2] W. Bae, J. Yoo, and J. C. Ye. Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1141–1149. IEEE, 2017.

[3] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2015.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[5] C. Dong, Y. Deng, C. Change Loy, and X. Tang. Compression artifacts reduction by a deep convolutional network. In *IEEE Conference on International Conference on Computer Vision*, pages 576–584, 2015.

[6] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016.

[7] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep Laplacian pyramid networks for fast and accurate super-resolution. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[8] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[9] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2017.

[10] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[11] X. Mao, C. Shen, and Y. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems*, pages 2802–2810, 2016.

[12] 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 *IEEE Conference on International Conference Computer Vision*, volume 2, pages 416–423. IEEE, 2001.

[13] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[14] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[15] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *IEEE Conference on International Conference on Computer Vision*, 2017.

[16] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. *arXiv preprint arXiv:1702.08502*, 2017.

[17] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[18] J. C. Ye and Y. S. Han. Deep convolutional framelets: A general deep learning for inverse problems. *Society for Industrial and Applied Mathematics*, 2018.

[19] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, PP(99):1–1, 2016.

[20] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3929–3938, 2017.

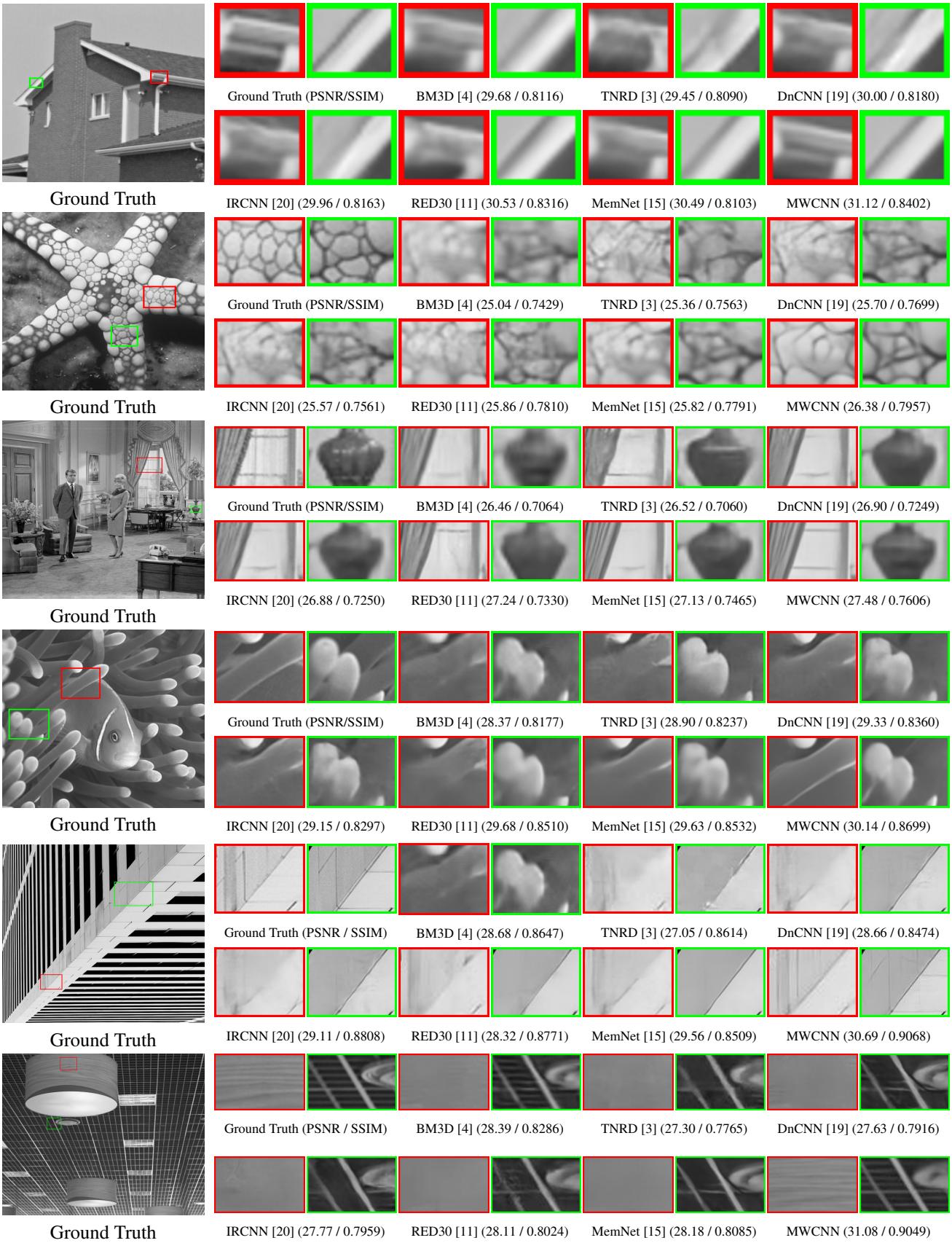

Figure 4: *Image denoising* visual results of "*02*" (Set12), "*04*" (Set12), "*12*" (Set12), "$Test039$" (BSD68), "$img011$" (Urban100) and "$img044$" (Urban100) with noise level 50.

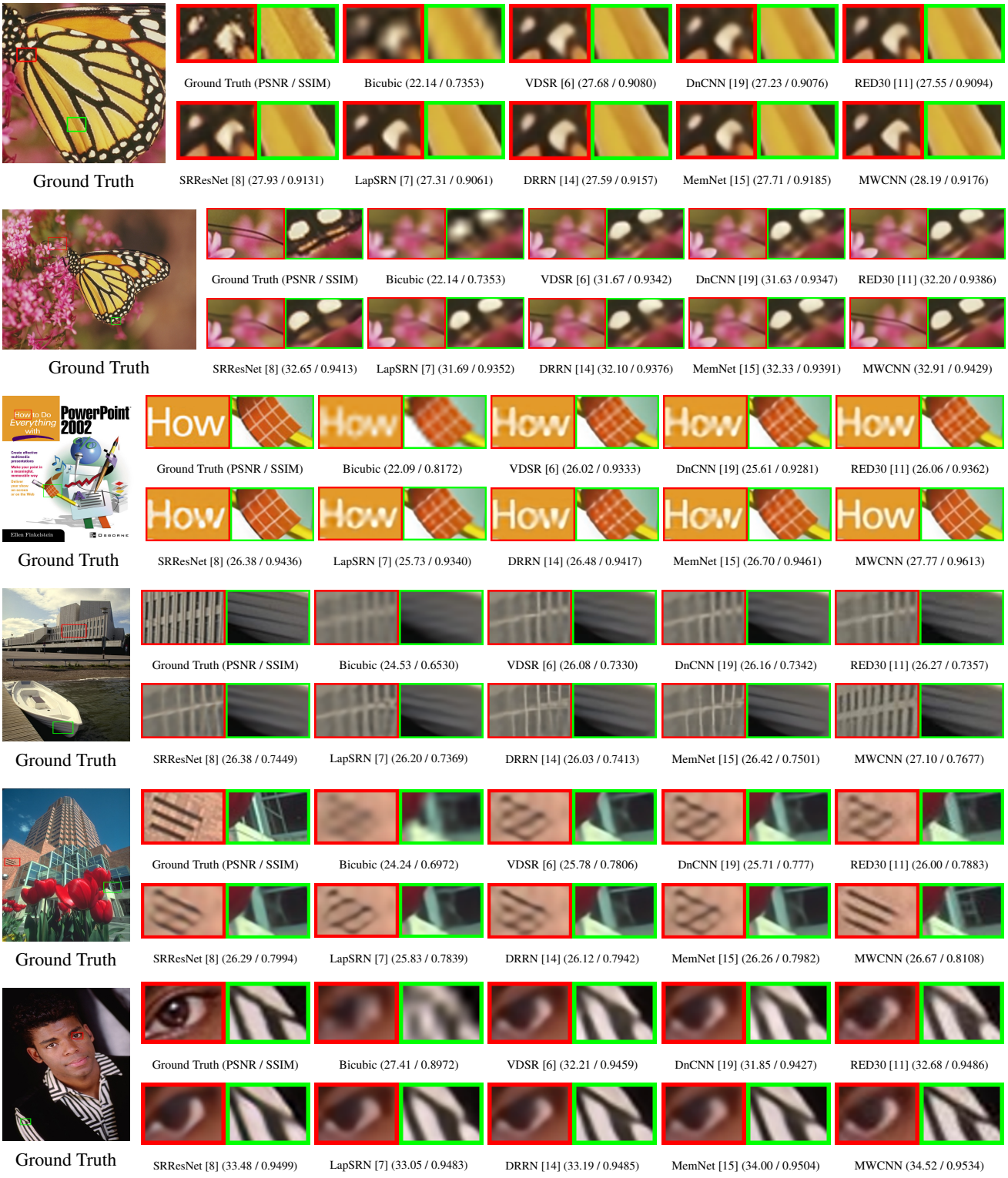

Figure 5: *Single image super-resolution* visual results of "*butterfly_GT*" (Set5), "*monarch*" (Set14), "*ppt3*" (Set14), "*78004*" (BSD100), "*86000*" (BSD100) and "*302008*" (BSD100) with upscaling factor $\times 4$.

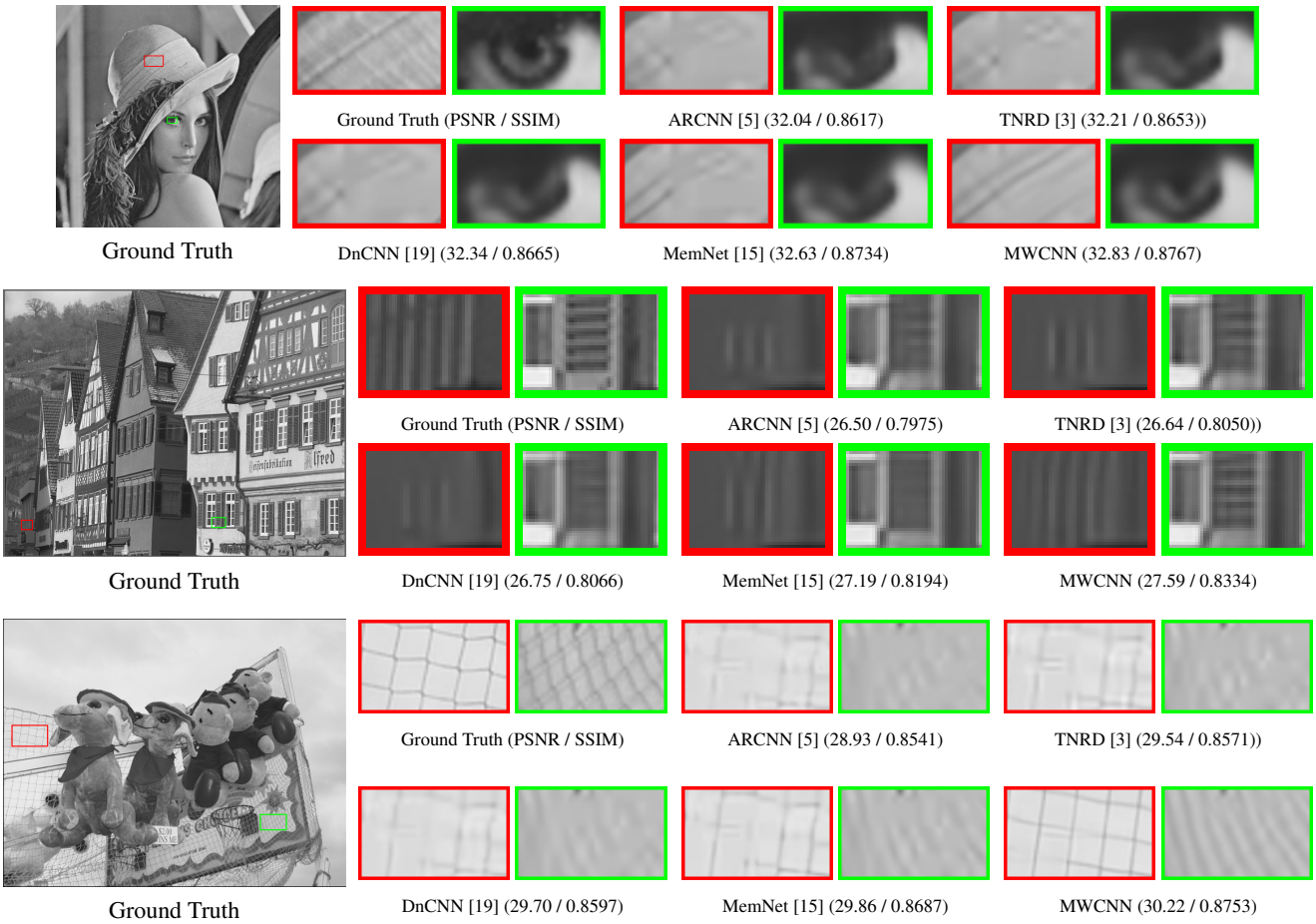

Figure 6: *JPEG image artifacts removal* visual results of "$Lena$" (classic5), "$buildings$" (LIVE1) and "$carnivaldolls$" (LIVE1) with quality factor 10.

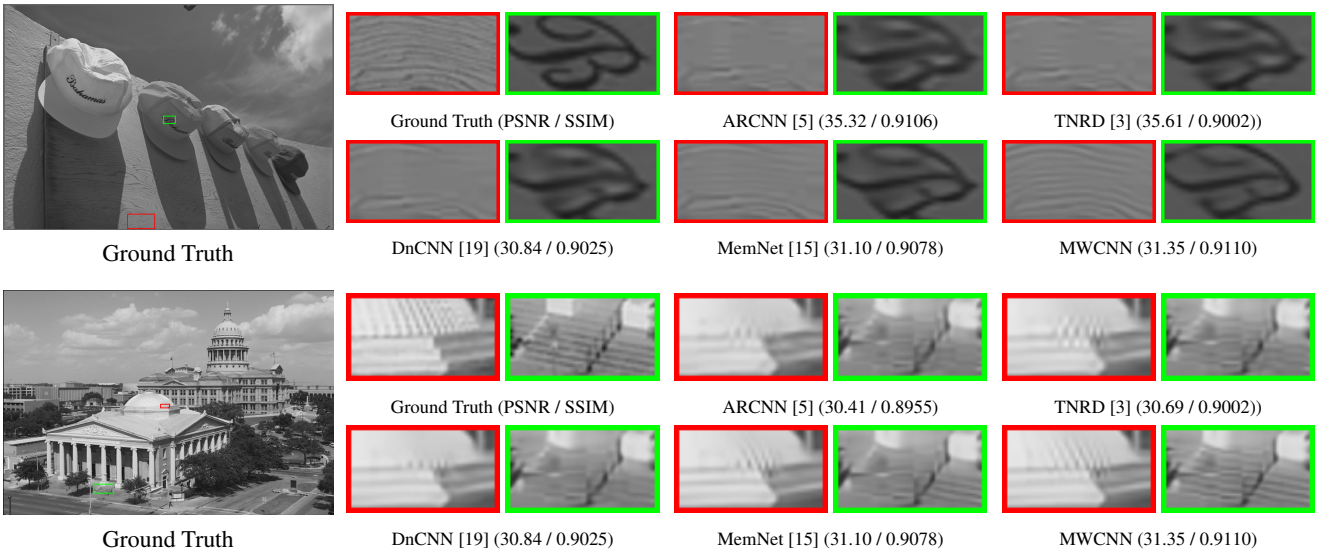

Figure 7: *JPEG image artifacts removal* visual results of "$caps$" and "$churchandcapito$" on LIVE1 with quality factor 20.