2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops

Multi-level Wavelet-CNN for Image Restoration

Pengju Liu[1], Hongzhi Zhang *[1], Kai Zhang[1], Liang Lin[2], and Wangmeng Zuo[1]

[1]School of Computer Science and Technology, Harbin Institute of Technology, 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

Abstract

The tradeoff between receptive field size and efficiency is a crucial issue in low level vision. Plain convolutional networks (CNNs) generally enlarge the receptive field at the expense of computational cost. Recently, dilated filtering has been adopted to address this issue. But it suffers from gridding effect, and the resulting receptive field is only a sparse sampling of input image with checkerboard patterns. In this paper, we present a novel multi-level wavelet CNN (MWCNN) model for better tradeoff between receptive field size and computational efficiency. With the modified U-Net architecture, wavelet transform is introduced to reduce the size of feature maps in the contracting subnetwork. Furthermore, another convolutional layer is further used to decrease the channels of feature maps. In the expanding subnetwork, inverse wavelet transform is then deployed to reconstruct the high resolution feature maps. Our MWCNN can also be explained as the generalization of dilated filtering and subsampling, and can be applied to many image restoration tasks. The experimental results clearly show the effectiveness of MWCNN for image denoising, single image super-resolution, and JPEG image artifacts removal.

1. Introduction

Image restoration, which aims to recover the latent clean image $\mathbf{x}$ from its degraded observation $\mathbf{y}$, is a fundamental and long-standing problem in low level vision. For decades, varieties of methods have been proposed for image restoration from both prior modeling and discriminative learning perspectives [6, 27, 10, 11, 17, 44, 52]. Recently, convolutional neural networks (CNNs) have also been extensively studied and achieved state-of-the-art performance in several representative image restoration tasks, such as single image super-resolution (SISR) [16, 29, 32], image denoising [57], image deblurring [58], and compressed imaging [34]. The popularity of CNN in image restoration can be explained from two aspects. On the one hand, existing CNN-based solutions have outperformed the other methods with a large margin for several simple tasks such as image denoising and SISR [16, 29, 32, 57]. On the other hand, recent studies have revealed that one can plug CNN-based denoisers into model-based optimization methods for solving more complex image restoration tasks [40, 58], which also promotes the widespread use of CNNs.

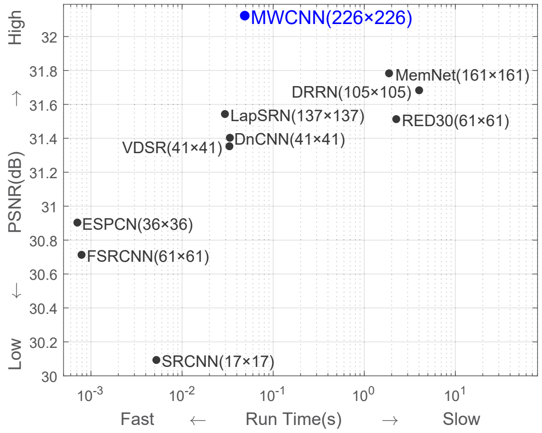


Figure 1. The run time vs. PSNR value of representative CNN models, including SRCNN [16], FSRCNN [14], ESPCN [45], VDSR [29], DnCNN [57], RED30 [37], LapSRN [31], DRRN [47], MemNet [47] and our MWCNN. The receptive field of each model are also provided. The PSNR and time are evaluated on Set5 with the scale factor ×4 running on a GTX1080 GPU.

For image restoration, CNN actually represents a mapping from degraded observation to latent clean image. Due to the input and output images usually should be of the same size, one representative strategy is to use the fully convolutional network (FCN) by removing the pooling layers. In general, larger receptive field is helpful to restoration performance by taking more spatial context into account. However, for FCN without pooling, the receptive field size can be enlarged by either increasing the network depth or using filters with larger size, which unexceptionally results in higher computational cost. In [58], dilated filtering [55]

*Corresponding author.

2160-7516/18 $31.00 © 2018 IEEE
DOI 10.1109/CVPRW.2018.00121

is adopted to enlarge receptive field without the sacrifice of computational cost. Dilated filtering, however, inherently suffers from gridding effect [50], where the receptive field only considers a sparse sampling of input image with checkerboard patterns. Thus, one should be careful to enlarge receptive field while avoiding the increase of computational burden and the potential sacrifice of performance improvement. Taking SISR as an example, Figure 1 illustrates the receptive field, run times, and PSNR values of several representative CNN models. It can be seen that FSRCNN [14] has relatively larger receptive field but achieves lower PSNR value than VDSR [29] and DnCNN [57].

In this paper, we present a multi-level wavelet CNN (MWCNN) model to enlarge receptive field for better tradeoff between performance and efficiency. Our MWCNN is based on the U-Net [41] architecture consisting of a contracting subnetwork and an expanding subnetwork. In the contracting subnetwork, discrete wavelet transform (DWT) is introduced to replace each pooling operation. Since DWT is invertible, it is guaranteed that all the information can be kept by such downsampling scheme. Moreover, DWT can capture both frequency and location information of feature maps [12, 13], which may be helpful in preserving detailed texture. In the expanding subnetwork, inverse wavelet transform (IWT) is utilized for upsampling low resolution feature maps to high resolution ones. To enrich feature representation and reduce computational burden, elementwise summation is adopted for combining the feature maps from the contracting and expanding subnetworks. Moreover, dilated filtering can also be explained as a special case of MWCNN, and ours is more general and effective in enlarging receptive field. Experiments on image denoising, SISR, and JPEG image artifacts removal validate the effectiveness and efficiency of our MWCNN. As shown in Figure 1, MWCNN is moderately slower than LapSRN [31], DnCNN [57] and VDSR [29] in terms of run time, but can have a much larger receptive field and higher PSNR value. To sum up, the contributions of this work include:

- A novel MWCNN model to enlarge receptive field with better tradeoff between efficiency and restoration performance.
- Promising detail preserving ability due to the good time-frequency localization of DWT.
- State-of-the-art performance on image denoising, SISR, and JPEG image deblocking.

2. Related work

In this section, we present a brief review on the development of CNNs for image denoising, SISR, JPEG image artifacts removal, and other image restoration tasks. Specifically, more discussions are given to the relevant works on enlarging receptive field and incorporating DWT in CNNs.

2.1. Image denoising

Since 2009, CNNs have been applied for image denoising [25]. These early methods generally cannot achieve state-of-the-art denoising performance [2, 25, 53]. Recently, multi-layer perception (MLP) has been adopted to learn the mapping from noise patch to clean pixel, and achieve comparable performance with BM3D [8]. By incorporating residual learning with batch normalization [24], the DnCNN model by Zhang *et al*. [57] can outperform traditional non-CNN based methods. Mao *et al*. [37] suggest to add symmetric skip connections to FCN for improving denoising performance. For better tradeoff between speed and performance, Zhang *et al*. [58] present a 7-layer FCN with dilated filtering. Santhanam *et al*. [43] introduce a recursively branched deconvolutional network (RBDN), where pooling/unpooling is adopted to obtain and aggregate multicontext representation.

2.2. Single image super-resolution

The application of CNN in SISR begins with SRCNN [16], which adopts a 3-layer FCN without pooling and has a small receptive field. Subsequently, very deep network [29], residual units [32], Laplacian pyramid [31], and recursive architecture [28, 47] have also been suggested to enlarge receptive field. These methods, however, enlarge the receptive field at the cost of either increasing computational cost or loss of information. Due to the speciality of SISR, one effective approach is to take the low-resolution (LR) image as input to CNN [14, 45] for better tradeoff between receptive field size and efficiency. In addition, generative adversarial networks (GANs) have also been introduced to improve the visual quality of SISR [26, 32, 42].

2.3. JPEG image artifacts removal

Due to high compression rate, JPEG image usually suffers from blocking effect and results in unpleasant visual quality. In [15], Dong *et al*. adopt a 4-layer ARCNN for JPEG image deblocking. By taking the degradation model of JPEG compression into account [10, 51], Guo *et al*. [18] suggest a dual-domain convolutional network to combine the priors in both DCT and pixel domains. GAN has also been introduced to generate more realistic result [19].

2.4. Other restoration tasks

Due to the similarity of image denoising, SISR, and JPEG artifacts removal, the model suggested for one task may be easily extended to the other tasks simply by retraining. For example, both DnCNN [57] and MemNet [48] have been evaluated on all the three tasks. Moreover, CNN denoisers can also serve as a kind of plug-and-play prior. By incorporating with unrolled inference, any restoration tasks can be tackled by sequentially applying the CNN denoisers [58]. Romano *et al*. [40] further propose a regularization by denoising framework, and provide an explicit functional for defining the regularization induced by denoisers. These methods not only promote the application of CNN in low level vision, but also present many solutions to exploit CNN denoisers for other image restoration tasks.

Several studies have also been given to incorporate wavelet transform with CNN. Bae *et al*. [5] find that learning CNN on wavelet subbands benefits CNN learning, and suggest a wavelet residual network (WavResNet) for image denoising and SISR. Similarly, Guo *et al*. [20] propose a deep wavelet super-resolution (DWSR) method to recover missing details on subbands. Subsequently, deep convolutional framelets [21, 54] have been developed to extend convolutional framelets for low-dose CT. However, both of WavResNet and DWSR only consider one level wavelet decomposition. Deep convolutional framelets independently processes each subband from decomposition perspective, which ignores the dependency between these subbands. In contrast, multi-level wavelet transform is considered by our MWCNN to enlarge receptive field without information loss. Taking all the subbands as inputs after each transform, our MWCNN can embed DWT to any CNNs with pooling, and owns more power to model both spatial context and inter-subband dependency.

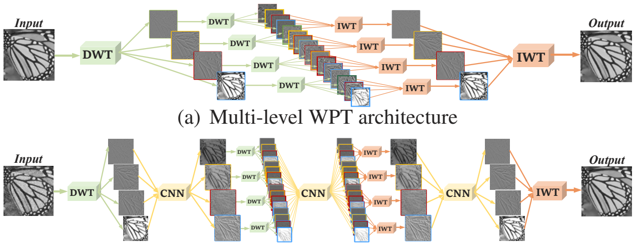
3. Method

In this section, we first introduce the multi-level wavelet packet transform (WPT). Then we present our MWCNN motivated by multi-level WPT, and describe its network architecture. Finally, discussion is given to analyze the connection of MWCNN with dilated filtering and subsampling.

3.1. From multi-level WPT to MWCNN

In 2D discrete wavelet transform (DWT), four filters, i.e. $\mathbf{f}_{LL}$, $\mathbf{f}_{LH}$, $\mathbf{f}_{HL}$, and $\mathbf{f}_{HH}$, are used to convolve with an image $\mathbf{x}$ [36]. The convolution results are then downsampled to obtain the four subband images $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$. For example, $\mathbf{x}_1$ is defined as $(\mathbf{f}_{LL} \otimes \mathbf{x}) \downarrow_2$. Even though the downsampling operation is deployed, due to the biorthogonal property of DWT, the original image $\mathbf{x}$ can be accurately reconstructed by the inverse wavelet transform (IWT), *i.e.*, $\mathbf{x} = IWT(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$.

In multi-level wavelet packet transform (WPT) [4, 13], the subband images $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$ are further processed with DWT to produce the decomposition results. For two-level WPT, each subband image $\mathbf{x}_i$ (i = 1, 2, 3, or 4) is decomposed into four subband images $\mathbf{x}_{i,1}$, $\mathbf{x}_{i,2}$, $\mathbf{x}_{i,3}$, and $\mathbf{x}_{i,4}$. Recursively, the results of three or higher levels WPT can be attained. Figure 2(a) illustrates the decomposition and reconstruction of an image with WPT. Actually, WPT is a special case of FCN without the nonlinearity layers. In the decomposition stage, four pre-defined filters are deployed to each (subband) image, and downsampling is then adopted as the pooling operator. In the reconstruction stage, the four subband images are first upsampled and then convolved with the corresponding filters to produce the reconstruction result at the current level. Finally, the original image $\mathbf{x}$ can be accurately reconstructed by inverse WPT.



(a) Multi-level WPT architecture

(b) Multi-level wavelet-CNN architecture

Figure 2. From WPT to MWCNN. Intuitively, WPT can be seen as a special case of our MWCNN without CNN blocks.

In image denoising and compression, some operations, *e.g*., soft-thresholding and quantization, usually are required to process the decomposition result [9, 33]. These operations can be treated as some kind of nonlinearity tailored to specific task. In this work, we further extend WPT to multi-level wavelet-CNN (MWCNN) by adding a CNN block between any two levels of DWTs, as illustrated in Figure 2(b). After each level of transform, all the subband images are taken as the inputs to a CNN block to learn a compact representation as the inputs to the subsequent level of transform. It is obvious that MWCNN is a generalization of multi-level WPT, and degrades to WPT when each CNN block becomes the identity mapping. Due to the biorthogonal property of WPT, our MWCNN can use subsampling operations safely without information loss. Moreover, compared with conventional CNN, the frequency and location characteristics of DWT is also expected to benefit the preservation of detailed texture.

3.2. Network architecture

The key of our MWCNN architecture is to design the CNN block after each level of DWT. As shown in Figure 3, each CNN block is a 4-layer FCN without pooling, and takes all the subband images as inputs. In contrast, different CNNs are deployed to low-frequency and high-frequency bands in deep convolutional framelets [21, 54]. We note that the subband images after DWT are still dependent, and the ignorance of their dependence may be harmful to the restoration performance. Each layer of the CNN block is composed of convolution with 3 × 3 filters (Conv), batch normalization (BN), and rectified linear unit (ReLU) operations. As to the last layer of the last CNN block, Conv without BN and ReLU is adopted to predict residual image.

Figure 3 shows the overall architecture of MWCNN which consists of a contracting subnetwork and an expanding subnetwork. Generally, MWCNN modifies U-Net from three aspects. (i) For downsampling and upsampling, max-

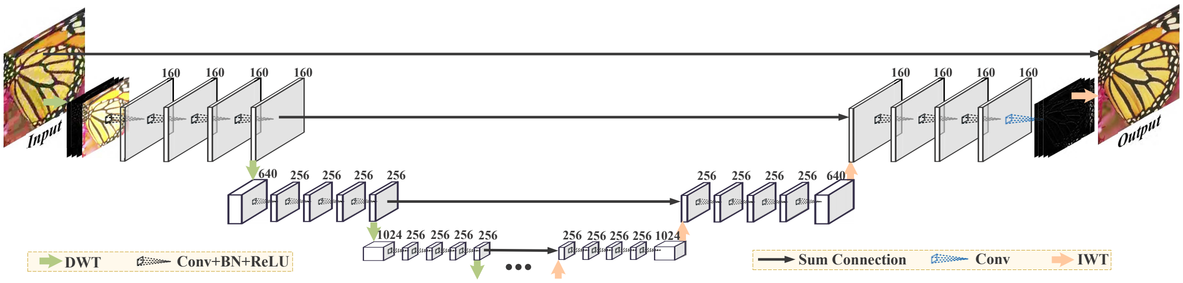


Figure 3. Multi-level wavelet-CNN architecture. It consists two parts: the contracting and expanding subnetworks. Each solid box corresponds to a multi-channel feature map. And the number of channels is annotated on the top of the box. The network depth is 24. Moreover, our MWCNN can be further extended to higher level (e.g., ≥ 4) by duplicating the configuration of the 3rd level subnetwork.

pooling and up-convolution are used in conventional U-Net[41], while DWT and IWT are utilized in MWCNN. (ii) For MWCNN, the downsampling results in the increase of feature map channels. Except the first one, the other CNN blocks are deployed to reduce the feature map channels for compact representation. In contrast, for conventional U-Net, the downsampling has no effect on feature map channels, and the subsequent convolution layers are used to increase feature map channels. (iii) In MWCNN, element-wise summation is used to combine the feature maps from the contracting and expanding subnetworks. While in conventional U-Net concatenation is adopted. Then our final network contains 24 layers. For more details on the setting of MWCNN, please refer to Figure 3. In our implementation, Haar wavelet is adopted as the default in MWCNN. Other wavelets, *e.g.*, Daubechies 2 (DB2), are also considered in our experiments.

Denote by Θ the network parameters of MWCNN, and $F(\mathbf{y};\Theta)$ be the network output. Let $\{(\mathbf{y}_i, \mathbf{x}_i)\}_{i=1}^{N}$ be a training set, where $\mathbf{y}_i$ is the i-th input image, $\mathbf{x}_i$ is the corresponding ground-truth image. The objective function for learning MWCNN is then given by

$$\mathcal{L}(\Theta) = \frac{1}{2N}\sum_{i=1}^{N} \|F(\mathbf{y}_i;\Theta) - \mathbf{x}_i\|_F^2. \tag{1}$$

The ADAM algorithm [30] is adopted to train MWCNN by minimizing the objective function. Different from VDSR [29] and DnCNN [57], we do not adopt the residual learning formulation for the reason that it can be naturally embedded in MWCNN.

3.3. Discussion

The DWT in MWCNN is closely related with the pooling operation and dilated filtering. By using the Haar wavelet as an example, we explain the connection between DWT and sum-pooling. In 2D Haar wavelet, the low-pass filter $\mathbf{f}_{LL}$ is defined as,

$$\mathbf{f}_{LL} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{2}$$

One can see that $(\mathbf{f}_{LL} \otimes \mathbf{x}) \downarrow_2$ actually is the sum-pooling operation. When only the low-frequency subband is considered, DWT and IWT will play the roles of pooling and up-convolution in MWCNN, respectively. When all the subbands are taken into account, MWCNN can avoid the information loss caused by conventional subsampling, and may benefit restoration result.

To illustrate the connection between MWCNN and dilated filtering with factor 2, we first give the definition of $\mathbf{f}_{LH}$, $\mathbf{f}_{HL}$, and $\mathbf{f}_{HH}$,

$$\mathbf{f}_{LH} = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}, \mathbf{f}_{HL} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, \mathbf{f}_{HH} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{3}$$

Given an image $\mathbf{x}$ with size of $m \times n$, the (i,j)-th value of $\mathbf{x}_1$ after 2D Haar transform can be written as $\mathbf{x}_1(i,j) = \mathbf{x}(2i-1, 2j-1) + \mathbf{x}(2i-1, 2j) + \mathbf{x}(2i, 2j-1) + \mathbf{x}(2i, 2j)$. And $\mathbf{x}_2(i,j)$, $\mathbf{x}_3(i,j)$, and $\mathbf{x}_4(i,j)$ can be defined analogously. We also have $\mathbf{x}(2i-1, 2j-1) = (\mathbf{x}_1(i,j) - \mathbf{x}_2(i,j) - \mathbf{x}_3(i,j) + \mathbf{x}_4(i,j))/4$. The dilated filtering with factor 2 on the position $(2i-1, 2j-1)$ of $\mathbf{x}$ can be written as

$$(\mathbf{x} \otimes_2 \mathbf{k})(2i-1, 2j-1) = \sum_{\substack{p+2s=2i-1,\\ q+2t=2j-1}} \mathbf{x}(p,q)\mathbf{k}(s,t), \tag{4}$$

where $\mathbf{k}$ is the 3×3 convolution kernel. Actually, it also can be obtained by using the 3×3 convolution with the subband images,

$$(\mathbf{x} \otimes_2 \mathbf{k})(2i-1, 2j-1) = ((\mathbf{x}_1 - \mathbf{x}_2 - \mathbf{x}_3 + \mathbf{x}_4) \otimes \mathbf{k})(i,j)/4. \tag{5}$$

Analogously, we can analyze the connection between dilated filtering and MWCNN for $(\mathbf{x} \otimes_2 \mathbf{k})(2i-1, 2j)$, $(\mathbf{x} \otimes_2 \mathbf{k})(2i, 2j-1)$, $(\mathbf{x} \otimes_2 \mathbf{k})(2i, 2j)$. 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.

Compared with dilated filtering, MWCNN can also avoid the gridding effect. After several layers of dilated filtering, it only considers a sparse sampling of locations with the checkerboard pattern, resulting in large portion of information loss (see Figure 4(a)). Another problem with dilated filtering is that the two neighbored pixels may be based on information from totally non-overlapped locations

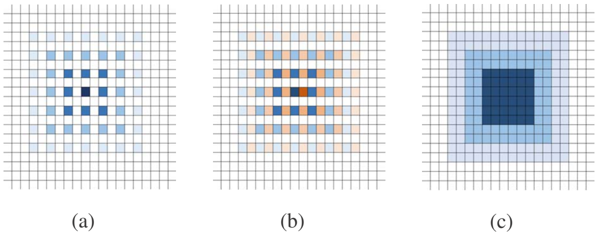


Figure 4. Illustration of the gridding effect. Taken 3-layer CNNs as an example: (a) the dilated filtering with factor 2 surfers large portion of information loss, (b) and the two neighbored pixels are based on information from totally non-overlapped locations, (c) while our MWCNN can perfectly avoid underlying drawbacks.

(see Figure 4(b)), and may cause the inconsistence of local information. In contrast, Figure 4(c) illustrates the receptive field of MWCNN. One can see that MWCNN is able to well address the sparse sampling and inconsistence of local information, and is expected to benefit restoration performance quantitatively and qualitatively.

4. Experiments

Experiments are conducted for performance evaluation on three tasks, *i.e.*, image denoising, SISR, and compression artifacts removal. Comparison of several MWCNN variants is also given to analyze the contribution of each component. The code and pre-trained models will be given at `https://github.com/lpj0/MWCNN`.

4.1. Experimental setting

4.1.1 Training set

To train our MWCNN, a large training set is constructed by using images from three dataset, *i.e.* Berkeley Segmentation Dataset (BSD) [38], DIV2K [3] and Waterloo Exploration Database (WED) [35]. Concretely, we collect 200 images from BSD, 800 images from DIV2K, and $4,744$ images from WED. Due to the receptive field of MWCNN is not less than 226×226, in the training stage $N = 24 \times 6,000$ patches with the size of 240×240 are cropped from the training images.

For image denoising, Gaussian noise with specific noise level is added to clean patch, and MWCNN is trained to learn a mapping from noisy image to denoising result. Following [57], we consider three noise levels, *i.e.*, σ = 15, 25 and 50. For SISR, we take the result by bicubic upsampling as the input to MWCNN, and three specific scale factors, *i.e.*, $\times 2$, $\times 3$ and $\times 4$, are considered in our experiments. For JPEG image artifacts removal, we follow [15] by considering four compression quality settings Q = 10, 20, 30 and 40 for the JPEG encoder. Both JPEG encoder and JPEG image artifacts removal are only applied on the Y channel [15].

4.1.2 Network training

A MWCNN model is learned for each degradation setting. The network parameters are initialized based on the method described in [22]. We use the ADAM algorithm [30] with $\alpha = 0.01$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ for optimizing and a mini-batch size of 24. As to the other hyper-parameters of ADAM, the default setting is adopted. The learning rate is decayed exponentially from 0.001 to 0.0001 in the 40 epochs. Rotation or/and flip based data augmentation is used during mini-batch learning. We use the MatConvNet package [49] with cuDNN 6.0 to train our MWCNN. All the experiments are conducted in the Matlab (R2016b) environment running on a PC with Intel(R) Core(TM) i7-5820K CPU 3.30GHz and an Nvidia GTX1080 GPU. The learning algorithm converges very fast and it takes about two days to train a MWCNN model.

4.2. Quantitative and qualitative evaluation

In this subsection, all the MWCNN models use the same network setting described in Sec. 3.2, and 2D Haar wavelet is adopted.

4.2.1 Image denoising

Except CBM3D [11] and CDnCNN [57], most denoising methods are only tested on gray images. Thus, we train our MWCNN by using the gray images, and compare with six competing denoising methods, *i.e.*, BM3D [11], TNRD [10], DnCNN [57], IRCNN [58], RED30 [37], and MemNet [48]. We evaluate the denoising methods on three test datasets, *i.e.*, Set12 [57], BSD68 [38], and Urban100 [23]. Table 1 lists the average PSNR/SSIM results of the competing methods on these three datasets. We note that our MWCNN only slightly outperforms DnCNN by about $0.1 \sim 0.3$dB in terms of PSNR on BSD68. As to other datasets, our MWCNN generally achieves favorable performance when compared with the competing methods. When the noise level is high (*e.g.*, $\sigma = 50$), the average PSNR by our MWCNN can be 0.5dB higher than that by DnCNN on Set12, and 1.2dB higher on Urban100. Figure 5 shows the denoising results of the images *Test011* from Set68 with the noise level $\sigma = 50$. One can see that our MWCNN is promising in recovering image details and structures, and can obtain visually more pleasant result than the competing methods. Please refer to the supplementary materials for more results on Set12 and Urban100.

4.2.2 Single image super-resolution

Following [29], SISR is only applied to the luminance channel, *i.e.* Y in YCbCr color space. We test MWCNN on four datasets, *i.e.*, Set5 [7], Set14 [56], BSD100 [38], and

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 [11]	TNRD [10]	DnCNN [57]	IRCNN [58]	RED30 [37]	MemNet [48]	MWCNN
Set12	15	32.37 / 0.8952	32.50 / 0.8962	32.86 / 0.9027	32.77 / 0.9008	-	-	33.15 / 0.9088
	25	29.97 / 0.8505	30.05 / 0.8515	30.44 / 0.8618	30.38 / 0.8601	-	-	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	27.74 / 0.8056
BSD68	15	31.08 / 0.8722	31.42 / 0.8822	31.73 / 0.8906	31.63 / 0.8881	-	-	31.86 / 0.8947
	25	28.57 / 0.8017	28.92 / 0.8148	29.23 / 0.8278	29.15 / 0.8249	-	-	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	26.53 / 0.7366
Urban100	15	32.34 / 0.9220	31.98 / 0.9187	32.67 / 0.9250	32.49 / 0.9244	-	-	33.17 / 0.9357
	25	29.70 / 0.8777	29.29 / 0.8731	29.97 / 0.8792	29.82 / 0.8839	-	-	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	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	RCN [46]	VDSR [29]	DnCNN [57]	RED30 [37]	SRResNet [32]	LapSRN [31]	DRRN [47]	MemNet [48]	WaveResNet [5]	MWCNN
Set5	×2	37.17 / 0.9583	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	37.91 / 0.9600
	×3	33.45 / 0.9175	33.66 / 0.9213	33.75 / 0.9222	33.82 / 0.9230	-	-	34.03 / 0.9244	34.09 / 0.9248	33.86 / 0.9228	34.18 / 0.9272
	×4	31.11 / 0.8736	31.35 / 0.8838	31.40 / 0.8845	31.51 / 0.8869	32.05 / 0.8902	31.54 / 0.8850	31.68 / 0.8888	31.74 / 0.8893	31.52 / 0.8864	32.12 / 0.8941
Set14	×2	32.77 / 0.9109	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	33.70 / 0.9182
	×3	29.63 / 0.8269	29.77 / 0.8314	29.76 / 0.8349	29.61 / 0.8341	-	-	29.96 / 0.8349	30.00 / 0.8350	29.88 / 0.8331	30.16 / 0.8414
	×4	27.79 / 0.7594	28.01 / 0.7674	28.02 / 0.7670	27.86 / 0.7718	28.49 / 0.7783	28.19 / 0.7720	28.21 / 0.7720	28.26 / 0.7723	28.11 / 0.7699	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	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	29.12 / 0.8060
	×4	-	27.29 / 0.7251	27.23 / 0.7233	27.39 / 0.7286	27.56 / 0.7354	27.32 / 0.7280	27.38 / 0.7284	27.40 / 0.7281	27.32 / 0.7266	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	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	28.13 / 0.8514
	×4	-	25.18 / 0.7524	25.20 / 0.7521	25.35 / 0.7587	26.07 / 0.7839	25.21 / 0.7560	25.44 / 0.7638	25.50 / 0.7630	25.36 / 0.7614	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 [15]	TNRD [10]	DnCNN [57]	MemNet [48]	MWCNN
Classic5	10	27.82 / 0.7595	29.03 / 0.7929	29.28 / 0.7992	29.40 / 0.8026	29.69 / 0.8107	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	32.16 / 0.8701
	30	31.48 / 0.8744	32.51 / 0.8806	32.78 / 0.8837	32.91 / 0.8861	-	33.43 / 0.8930
	40	32.43 / 0.8911	33.34 / 0.8953	-	33.77 / 0.9003	-	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	29.69 / 0.8254
	20	30.07 / 0.8512	31.29 / 0.8733	31.46 / 0.8769	31.59 / 0.8802	31.83 / 0.8846	32.04 / 0.8885
	30	31.41 / 0.9000	32.67 / 0.9043	32.84 / 0.9059	32.98 / 0.9090	-	33.45 / 0.9153
	40	32.35 / 0.9173	33.63 / 0.9198	-	33.96 / 0.9247	-	34.45 / 0.9301

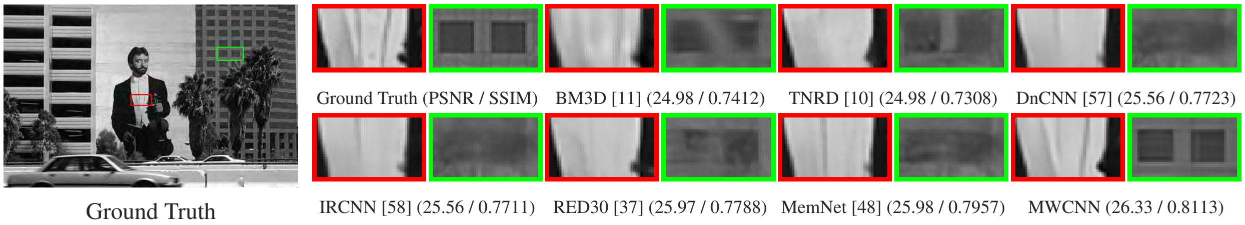
Urban100 [23], because they are widely adopted to evaluate SISR performance. Our MWCNN is compared with eight CNN-based SISR methods, including RCN [46], VDSR [29], DnCNN [57], RED30 [37], SRResNet [32], LapSRN [31], DRRN [47], and MemNet [48]. Due to the source code of SRResNet is not released, its results are from [32] and are incomplete.

Table 2 lists the average PSNR/SSIM results of the competing methods on the four datasets. Our MWCNN performs favorably in terms of both PSNR and SSIM indexes. Compared with VDSR, our MWCNN achieves a notable gain of about 0.4dB by PSNR on Set5 and Set14. On Urban100, our MWCNN outperforms VDSR by about 0.9∼1.4dB. Obviously, WaveResNet *et al.* [5] sightly outperform VDSR, and also is still inferior to MWCNN. We note that the network depth of SRResNet is 34, while that of MWCNN is 24. Moreover, SRResNet is trained with a much larger training set than MWCNN. Even so, when the scale factor is 4, MWCNN achieve slightly higher PSNR values on Set5 and BSD100, and is comparable to SRResNet on Set14. Figure 6 shows the visual comparisons of the competing methods on the images *Barbara* from Set14. Thanks to the frequency and location characteristics of DWT, our MWCNN can correctly recover the fine and detailed textures, and produce sharp edges. Furthermore, for Track 1 of NTIRE 2018 SR challenge (×8 SR) [1], our improved MWCNN is lower than the Top-1 method by 0.37dB.

4.2.3 JPEG image artifacts removal

In JPEG compression, an image is divided into non-overlapped 8 × 8 blocks. Discrete cosine transform (DCT) and quantization are then applied to each block, thus introducing the blocking artifact. The quantization is determined by a quality factor Q to control the compression rate. Following [15], we consider four settings on quality factor, *e.g.*, Q = 10, 20, 30 and 40, for the JPEG encoder. Both JPEG encoder and JPEG image artifacts removal are only applied to the Y channel. In our experiments, MWCNN is compared with four competing methods, *i.e.*, ARCNN [15], TNRD [10], DnCNN [57], and MemNet [48] on the two datasets, *i.e.*, Classic5 and LIVE1 [39]. We do not consider [18, 19] due to their source codes are unavailable.

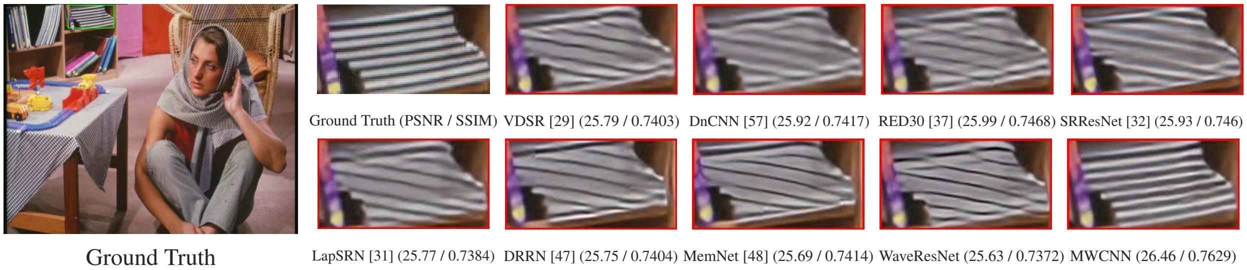
Table 3 lists the average PSNR/SSIM results of the competing methods on Classic5 and LIVE1. For any of the four quality factors, our MWCNN performs favorably in terms of quantitative metrics on the two datasets. On Classic5 and LIVE1, the PSNR values of MWCNN can be 0.2∼0.3dB higher than those of the second best method (*i.e.*, Mem-



Ground Truth (PSNR / SSIM) BM3D [11] (24.98 / 0.7412) TNRD [10] (24.98 / 0.7308) DnCNN [57] (25.56 / 0.7723)

Ground Truth IRCNN [58] (25.56 / 0.7711) RED30 [37] (25.97 / 0.7788) MemNet [48] (25.98 / 0.7957) MWCNN (26.33 / 0.8113)

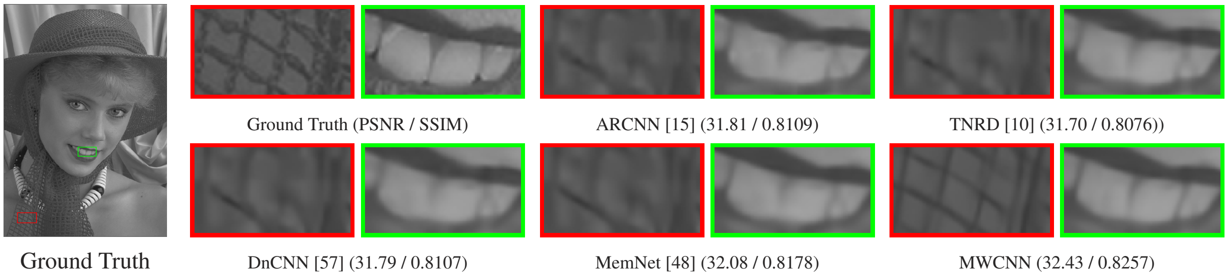
Figure 5. *Image denoising* results of "$Test011$" (BSD68) with noise level 50.



Ground Truth (PSNR / SSIM) VDSR [29] (25.79 / 0.7403) DnCNN [57] (25.92 / 0.7417) RED30 [37] (25.99 / 0.7468) SRResNet [32] (25.93 / 0.746)

Ground Truth LapSRN [31] (25.77 / 0.7384) DRRN [47] (25.75 / 0.7404) MemNet [48] (25.69 / 0.7414) WaveResNet (25.63 / 0.7372) MWCNN (26.46 / 0.7629)

Figure 6. *Single image super-resolution* results of "$barbara$" (Set14) with upscaling factor ×4.



Ground Truth (PSNR / SSIM) ARCNN [15] (31.81 / 0.8109) TNRD [10] (31.70 / 0.8076))

Ground Truth DnCNN [57] (31.79 / 0.8107) MemNet [48] (32.08 / 0.8178) MWCNN (32.43 / 0.8257)

Figure 7. *JPEG image artifacts removal* results of "$womanhat$" (LIVE1) with quality factor 10.

Net [48]) for the quality factor of 10 and 20. Figure 7 shows the results on the image *womanhat* from LIVE1 with the quality factor 10. One can see that MWCNN is effective in restoring detailed textures and sharp salient edges.

4.2.4 Run time

Table 4 lists the GPU run time of the competing methods for the three tasks. The Nvidia cuDNN-v6.0 deep learning library is adopted to accelerate the GPU computation under Ubuntu 16.04 system. Specifically, only the CNN-based methods with source codes are considered in the comparison. For three tasks, the run time of MWCNN is far less than several state-of-the-art methods, including RED30 [37], MemNet [47] and DRRN [47]. Note that the three methods also perform poorer than MWCNN in terms of PSNR/SSIM metrics. In comparison to the other methods, MWCNN is moderately slower by speed but can achieve higher PSNR/SSIM indexes. The result indicates that, instead of the increase of network depth/width, the effectiveness of MWCNN should be attributed to the incorporation of CNN and DWT.

Table 4. Run time (in seconds) of the competing methods for the three tasks on images of size 256×256, 512×512 and 1024×1024: image denosing is tested on noise level 50, SISR is tested on scale ×2, and JPEG image deblocking is tested on quality factor 10.

Image Denoising

Size	TNRD [10]	DnCNN [57]	RED30 [37]	MemNet [47]	MWCNN
256×256	0.010	0.0143	1.362	0.8775	0.0586
512×512	0.032	0.0487	4.702	3.606	0.0907
1024×1024	0.116	0.1688	15.77	14.69	0.3575

Single Image Super-Resolution

Size	VDSR [29]	LapSRN [31]	DRRN [47]	MemNet [37]	MWCNN
256×256	0.0172	0.0229	3.063	0.8774	0.0424
512×512	0.0575	0.0357	8.050	3.605	0.0780
1024×1024	0.2126	0.1411	25.23	14.69	0.3167

JPEG Image Artifacts Removal

Size	ARCNN [15]	TNRD [10]	DnCNN [57]	MemNet [37]	MWCNN
256×256	0.0277	0.009	0.0157	0.8775	0.0531
512×512	0.0532	0.028	0.0568	3.607	0.0811
1024×1024	0.1613	0.095	0.2012	14.69	0.2931

4.3. Comparison of MWCNN variants

Using image denoising and JPEG image artifacts as examples, we compare the PSNR results by three MWCNN variants, including: (i) MWCNN (Haar): the default MWCNN with Haar wavelet, (ii) MWCNN (DB2): MWCNN with *Daubechies-2* wavelet, and (iii) MWCNN (HD): MWCNN with Haar in contracting subnetwork and *Daubechies-2* in expanding subnetwork. Then, ablation experiments are provided for verifying the effectiveness of additionally embedded wavelet: (i) the default U-Net with same architecture to MWCNN, (ii) U-Net+S: using sum connection instead of concatenation, and (iii) U-Net+D: adopting learnable conventional downsamping fil-

Table 5. Performance comparison in terms of average PSNR (dB) and run time (in seconds): image denosing is tested on noise level 50 and JPEG image deblocking is tested on quality factor 10.

Dataset	Dilated [55]	Dilated-2	U-Net [41]	U-Net+S	U-Net+D	DCF [21]	WaveResNet [5]	MWCNN (Haar)	MWCNN (DB2)	MWCNN (HD)
Image Denoising ($\sigma = 50$)										
Set12	27.45 / 0.181	24.81 / 0.185	27.42 / 0.079	27.41 / **0.074**	27.46 / 0.080	27.38 / 0.081	27.49 / 0.179	27.74 / 0.078	**27.77** / 0.134	27.73 / 0.101
BSD68	26.35 / 0.142	24.32 / 0.174	26.30 / 0.076	26.29 / **0.071**	26.21 / 0.075	26.30 / 0.075	26.38 / 0.143	26.53 / 0.072	**26.54** / 0.122	26.52 / 0.088
Urban100	26.56 / 0.764	24.18 / 0.960	26.68 / 0.357	26.72 / **0.341**	26.99 / 0.355	26.65 / 0.354	- / -	27.42 / 0.343	**27.48** / 0.634	27.35 / 0.447
JPEG Image Artifacts Removal (PC=10)										
Classic5	29.72 / 0.287	29.49 / 0.302	29.61 / 0.093	29.60 / **0.082**	29.68 / 0.097	29.57 / 0.104	- / -	30.01 / 0.088	**30.04** / 0.195	29.97 / 0.136
LIVE1	29.49 / 0.354	29.26 / 0.376	29.36 / 0.112	29.36 / **0.109**	29.43 / 0.120	29.38 / 0.155	- / -	29.69 / 0.112	**29.70** / 0.265	29.66 / 0.187

ters instead of Max pooling. Two 24-layer dilated CNNs are also considered: (i) Dilated: the hybrid dilated convolution [50] to suppress the gridding effect, and (ii) Dilated-2: the dilate factor of all layers is set to 2. The WaveResNet method in [5] is provided to be compared. Moreover, due to its code is unavailable, a self-implementation of deep convolutional framelets (DCF) [54] is also considered in the experiments.

Table 4 lists the PSNR and run time results of these methods. And we have the following observations. (i) The gridding effect with the sparse sampling and inconsistence of local information authentically has adverse influence on restoration performance. (ii) The ablation experiments indicate that using sum connection instead of concatenation can improve efficiency without decreasing PNSR. Due to the special group of filters with the biorthogonal and time-frequency localization property in wavelet, our embedded wavelet own more puissant ability for image restoration than pooling operation and learnable downsamping filters. The worse performance of DCF also indicates that independent processing of subbands harms final result. (iii) Compared to MWCNN (DB2) and MWCNN (HD), using Haar wavelet for downsampling and upsampling in network is the best choice in terms of quantitative and qualitative evaluation. MWCNN (Haar) has similar run time with dilated CNN and U-Net but achieves higher PSNR results, which demonstrates the effectiveness of MWCNN for tradeoff between performance and efficiency.

Note that our MWCNN is quite different with DCF [54]: DCF incorporates CNN with DWT in the view of decomposition, where different CNNs are deployed to each subband. However, the results in Table 5 indicates that independent processing of subbands is not suitable for image restoration. On the contrary, MWCNN combines DWT to CNN from perspective of enlarging receptive field without information loss, allowing to embed DWT with any CNNs with pooling. Moreover, our embedded DWT can be treated as predefined parameters to ease network learning, and the dynamic range of subbands can be jointly adjusted by the CNN blocks. Taking all subbands as input, MWCNN is more powerful in modeling inter-band dependency.

As described in Sec. 3.2, our MWCNN can be extended to higher level of wavelet decomposition. Nevertheless, higher level inevitably results in deeper network and heavier computational burden. Thus, a suitable level is required to balance efficiency and performance. Table 6 reports the PSNR and run time results of MWCNNs with the levels of 1 to 4 (*i.e.*, MWCNN-1 $\sim$ MWCNN-4). It can be observed that MWCNN-3 with 24-layer architecture performs much better than MWCNN-1 and MWCNN-2, while MWCNN-4 only performs negligibly better than MWCNN-3 in terms of the PSNR metric. Moreover, the speed of MWCNN-3 is also moderate compared with other levels. Taking both efficiency and performance gain into account, we choose MWCNN-3 as the default setting.

Table 6. Average PSNR (dB) and run time (in seconds) of MWCNNs with different levels on Gaussian denoising with the noise level of 50.

Dataset	MWCNN-1	MWCNN-2	MWCNN-3	MWCNN-4
Set12	27.14 / 0.047	27.62 / 0.068	27.74 / 0.082	27.74 / 0.091
BSD68	26.16 / 0.044	26.45 / 0.063	26.53 / 0.074	26.54 / 0.084
Urban100	26.08 / 0.212	27.10 / 0.303	27.42 / 0.338	27.44 / 0.348

5. Conclusion

This paper presents a multi-level wavelet-CNN (MWCNN) architecture for image restoration, which consists of a contracting subnetwork and a expanding subnetwork. The contracting subnetwork is composed of multiple levels of DWT and CNN blocks, while the expanding subnetwork is composed of multiple levels of IWT and CNN blocks. Due to the invertibility, frequency and location property of DWT, MWCNN is safe to perform subsampling without information loss, and is effective in recovering detailed textures and sharp structures from degraded observation. As a result, MWCNN can enlarge receptive field with better tradeoff between efficiency and performance. Extensive experiments demonstrate the effectiveness and efficiency of MWCNN on three restoration tasks, *i.e.*, image denoising, SISR, and JPEG compression artifact removal.

In future work, we will extend MWCNN for more general restoration tasks such as image deblurring and blind deconvolution. Moreover, our MWCNN can also be used to substitute the pooling operation in the CNN architectures for high-level vision tasks such as image classification.

Acknowledgement

This work is partially supported by the National Science Foundation of China (NSFC) (grant No.s 61671182 and 61471146).

References

[1] Ntire 2018 super resolution challenge. http://vision.ee.ethz.ch/ntire18. Accessed Mar, 2018.

[2] F. Agostinelli, M. R. Anderson, and H. Lee. Robust image denoising with multi-column deep neural networks. In *Advances in Neural Information Processing Systems*, pages 1493–1501, 2013.

[3] 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.

[4] A. N. Akansu and R. A. Haddad. *Multiresolution signal decomposition: transforms, subbands, and wavelets*. Academic Press, 2001.

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

[6] M. R. Banham and A. K. Katsaggelos. Digital image restoration. *IEEE Signal Processing Magazine*, 14(2):24–41, 1997.

[7] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.

[8] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2392–2399, 2012.

[9] S. G. Chang, B. Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, 2000.

[10] 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.

[11] 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.

[12] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, 1990.

[13] I. Daubechies. *Ten lectures on wavelets*. SIAM, 1992.

[14] C. Dong, C. L. Chen, and X. Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*, pages 391–407, 2016.

[15] 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.

[16] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016.

[17] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, 2014.

[18] J. Guo and H. Chao. Building dual-domain representations for compression artifacts reduction. In *European Conference on Computer Vision*, pages 628–644, 2016.

[19] J. Guo and H. Chao. One-to-many network for visually pleasing compression artifacts reduction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[20] T. Guo, H. S. Mousavi, T. H. Vu, and V. Monga. Deep wavelet prediction for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[21] Y. Han and J. C. Ye. Framing U-Net via deep convolutional framelets: Application to sparse-view CT. *arXiv preprint arXiv:1708.08333*, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[23] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015.

[24] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

[25] V. Jain and S. Seung. Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems*, pages 769–776, 2009.

[26] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711, 2016.

[27] A. K. Katsaggelos. *Digital image restoration*. Springer Publishing Company, Incorporated, 2012.

[28] J. Kim, J. Kwon Lee, and K. Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1637–1645, 2016.

[29] 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.

[30] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015.

[31] 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.

[32] 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.

[33] A. S. Lewis and G. Knowles. Image compression using the 2-d wavelet transform. *IEEE Transactions on Image Processing*, 1(2):244–250, 1992.

[34] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang. Learning convolutional networks for content-weighted image compression. *arXiv preprint arXiv:1703.10553*, 2017.

[35] 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.

[36] 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.

[37] 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.

[38] 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, 2001.

[39] A. K. Moorthy and A. C. Bovik. Visual importance pooling for image quality assessment. *IEEE journal of selected topics in signal processing*, 3(2):193–201, 2009.

[40] Y. Romano, M. Elad, and P. Milanfar. The little engine that could: Regularization by denoising (red). *arXiv preprint arXiv:1611.02862*, 2016.

[41] 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.

[42] M. S. M. Sajjadi, B. Scholkopf, and M. Hirsch. EnhanceNet: Single image super-resolution through automated texture synthesis. In *IEEE International Conference on Computer Vision*, pages 4501–4510, 2017.

[43] V. Santhanam, V. I. Morariu, and L. S. Davis. Generalized deep image to image regression. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5609–5619, 2017.

[44] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2774–2781, 2014.

[45] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[46] Y. Shi, K. Wang, C. Chen, L. Xu, and L. Lin. Structure-preserving image super-resolution via contextualized multitask learning. *IEEE Transactions on Multimedia*, PP(99):1–1, 2017.

[47] 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.

[48] 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.

[49] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692, 2015.

[50] 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.

[51] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang. D3: Deep dual-domain based fast restoration of jpeg-compressed images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2764–2772, 2016.

[52] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[53] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *International Conference on Neural Information Processing Systems*, pages 341–349, 2012.

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

[55] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[56] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730, 2010.

[57] 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.

[58] 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.