Image Denoising Using a Deep Encoder-Decoder Network with Skip Connections

Raphael Couturier, Gilles Perrot, Michel Salomon

Image Denoising using a Deep Encoder-Decoder Network with Skip Connections

Raphaël Couturier[1], Gilles Perrot[1], and Michel Salomon[1]

FEMTO-ST Institute, CNRS - Univ. Bourgogne Franche-Comté (UBFC), Belfort, France
{raphael.couturier,gilles.perrot,michel.salomon}@univ-fcomte.fr

Abstract. In many areas images can be corrupted by various types of noise and therefore image denoising is a prerequisite. For example, medical images like the 4D-CT or ultrasound ones, are prone to noise and artifacts that can affect diagnostic confidence. Remote sensing is another field for which image preprocessing is mandatory to improve the quality of source images. Synthetic Aperture Radar (SAR) images are typically corrupted by multiplicative speckle noise. In this paper, a deep neural network able to deal with both additive white Gaussian and multiplicative speckle noises is developed, showing also some blind denoising capacity. The experiments on noisy images show that the proposal, which consists in a encoder-decoder, is efficient and competitive in comparison with state-of-the-art methods.

Keywords: Image denoising, Additive and multiplicative noises, Deep learning, Encoder-decoder

1 Introduction

In today's digital world, an increasing amount of digital images is produced every day. Nevertheless, the visual quality of an image is not guaranteed, since different sources of noise can influence the pixel values. A main source is the acquisition process and particularly the presence of defaults in the capturing device: noise can be produced by the sensor, misaligned lenses, and so on, but noise can also be added during its edition, storage or transmission. As a result, different types of noise can appear in a digital image, such as Gaussian noise, Salt-and-pepper noise, etc., and at different levels. For an observer, the impact of noise can range from isolated speckles up to images that seem to show nothing but noise.

To recover as precisely as possible a clean image y from a noisy version x that is the outcome of an arbitrary stochastic corruption process n: $x = n(y)$, an efficient image denoising method is needed. Formally, the goal of image denoising is thus to find a function f that approximates as well as possible the inverse function of n:

$$f = \underset{f}{\operatorname{argmin}} \, \mathrm{E}_y \parallel f(x) - y \parallel_2^2 . \tag{1}$$

It should be noticed that additive white Gaussian noise is often targeted, in which case the corruption process can be rewritten as $x = y + \mathcal{N}(0, \sigma)$ where σ is the standard deviation.

To solve this problem, there are two main categories of methods: model-based optimization methods and discriminative learning methods. The objective of the former methods is to directly solve the optimization problem, but, as this problem is usually complex, they are time consuming. On the other hand, discriminative learning methods try to learn a set Θ of parameters defining a nonlinear function $\hat{f}$ that approximates n^{-1} by minimizing a loss function according to a data set that consists of clean-noisy images pairs. In that case, the previous problem can be expressed as follows:

$$\Theta = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \parallel \hat{f}(x_i) - y_i \parallel_2^2 \tag{2}$$

where x_i is the noisy version of y_i and N is the size of the data set. Compared to model-based methods, discriminative ones are less flexible since they are usually trained to deal for a specific underlying model of corruption.

Typical examples of model-based methods are BM3D [4] and WNNM [7], while neural networks are representatives of the discriminative family. Obviously, even if the MLP has been investigated [2], with the current rise of deep learning, deep neural networks are now the most actively studied discriminative methods. One of the first deep network proposal was made by Xie *et al.* [18] in 2012, it consisted in a stacking of auto-encoders where each auto-encoder was trained one after the other. In 2014, Long *et al.* [10] introduced the Fully Convolutional Networks (FCN) for semantic segmentation, an architecture that allows to produce segmentation maps whatever the image size and faster than with patch classification approaches. A work that has led to the widespread use of deep networks in which the fully connected part is dropped for dense predictions.

Image denoising is such a dense prediction task, whose objective is to recover for each pixel its original gray level value. Consequently, among the various deep networks that have recently been investigated to tackle the image denoising problem, almost all of them have adopted the FCN paradigm. However, even if these networks belong to the same family, differences among them can be observed. First, a FCN can be trained to recover directly the clean image or to predict a residual image that is subtracted from the noisy input one [19]. Second, a central problem when using a CNN for image denoising (or segmentation) is due to pooling layers. Indeed, a pooling layer usually performs a spatial downsampling and as the input and output images must have the same size, it means that an upsampling process is needed. On the one hand the pooling permits to enlarge the field of view, but on the other hand the aggregation throws away useful spatial information. To address this issue, several different architectures have emerged: architectures without pooling layers and the encoder-decoder architecture.

The proposal is presented thereafter throughout the following sections. Section 2 starts with a discussion on related existing discriminative denoising methods and more specifically deep learning ones. An overview of the proposed deep

neural network design with its main characteristics is given in the following section. Section 4 is dedicated to the experiments, showing the relevance of the proposed approach. Finally, some concluding remarks are given in Section 5.

2 Related Works

A first example of architecture for image denoising that only has convolutional layers is the deep network called DnCNN (Deep network CNN) proposed by Zhang *et al.* [19] considering a residual learning formulation. The CNN is composed of layers with three different convolutional blocks using a unique convolution kernel size of 3×3: Convolution+ReLU for the first layer, Convolution+BatchNorm+ReLU in the intermediate layers, and only Convolution in the final layer. It has a receptive field whose size depends on the network depth and which is correlated with the effective patch size of other denoising methods. In fact, most of the denoising methods such as BM3D, WNNM, MLP, and so on operate on patches. The authors have thus chosen to increase the receptive field through a large depth. The networks of 17 and 20 layers that they trained for additive Gaussian denoising, respectively for a specific noise level and for blind denoising, outperformed slightly both BM3D and WNNM on the BSD68 data set of grayscale images. A recently proposed alternative to an increased depth or to increasing filter sizes is the use of dilated kernels, also known as atrous convolution. Indeed, convolutional layers that only use 3×3 kernels but with multiple atrous rates perform an analysis of the image at multiple scales without needing a large depth. This approach has also been studied by Zhang *et al.* in another work [21], leading to similar denoising performances.

Zhang *et al.* have finally introduced another architecture [20] to handle a wide range of noise levels and spatially variant noise. This architecture, called FFDNet for fast and flexible denoising convolutional neural network, consists of a CNN similar to the one of DnCNN, but that does not predict the noise. The CNN receives as input four sub-images obtained from the initial input image using a reversible downsampling operator (factor is set to 2) and a tunable noise level map. As output it produces four denoised sub-images which are then upsampled to recover the final output image. For Additive White Gaussian Noise (AWGN) removal, the experiments show that DnCNN is better for low noise levels ($\sigma \leqslant 25$), whereas for larger values FFDNet becomes gradually slightly better with the increase of noise level. This result is all the more interesting as it is the version of DnCNN trained for a specific noise level that is considered, whereas FFDNet is trained in a blind context with noise level $\sigma \in [0; 75]$.

An encoder-decoder is quite different. The encoder consists of convolutional layers that successively downsample the input image into small abstraction maps from which the noise is removed as the process goes deeper. The decoder is then fed by the final abstraction map in order to reconstruct a clean image thanks to deconvolutional layers. The reconstruction by the decoder is clearly the most difficult part since image details might be lost during the features extraction performed by the encoder. To mitigate this problem, a common approach is to adopt

the skip connection method. In this work we have considered an encoder-decoder with such connections. A similar architecture, but considering a residual learning pattern, has been investigated by Gu *et al.* in [6] for SAR image despeckling. Compared to SAR-BM3D and DnCNN, this Residual Encoder-Decoder NETwork (RED-NET) has given improved denoising performances.

3 The Proposed Deep Network for Image Denoising

3.1 Network Architecture

Our network is similar to a generator architecture introduced by Isola *et al.* [8] in their investigation of conditional adversarial networks to solve image-to-image translation problems, an architecture which is itself an adaptation of one issued from [14]. The generator we consider is the U-Net [15] version corresponding to an encoder-decoder having skip connections between mirrored layers in both encoder and decoder stacks. The encoder extracts salient features preserving the detailed underlying structure of the image, while simultaneously removing the noise, whereas the decoder produces a clean version of the input image by recovering successive image details as it progresses through its layers in a bottom-up way from the bottleneck layer of the encoder. Each skip connection allows to directly shuttle the information from an encoder layer to its corresponding decoder one, and this is appealing since the input noisy image and output clean version share large parts of the low-level information like the location of prominent edges. In fact, skip connections allow to remember different levels of details that are useful to reconstruct the final output image.

Symmetric skip connections are very interesting because they facilitate the training and improve image recovery. On the one hand, skip connections allow to solve the vanishing gradient problem by backpropagating the signal directly and, on the other hand as both input and output images have the same content, the recovery of the clean version can benefit from the details appearing in the corrupted one. Thus, better results are usually obtained with skip connections.

From an architecture point of view, following the specification given in [8], the encoder is almost exclusively composed of Convolution-BatchNorm-ReLU layers, a typical choice in CNN, except the first layer that does not undergo the batch normalization. Let us notice that, as illustrated in the previous section, this kind of convolutional block is also the one used in many FCN that do not have an encoder-decoder architecture. For its part, the decoder consists of a mixing of this kind of layer and a variant of it integrating a dropout rate of 50% before the ReLU activation. Neither pooling nor unpooling operations are used, because aggregation induces some losses of details which, in the context of image denoising, can be awkward. Since convolutions and deconvolutions use 4×4 kernels with stride 2, each encoder and decoder layer will produce feature maps which are downsampled and upsampled, respectively, by a factor of 2. The last top decoder layer is mapped back to the output clean image with a deconvolution followed by a tanh activation. Our code is available on GitHub[1].

[1] https://github.com/rcouturier/ImageDenoisingwithDeepEncoderDecoder

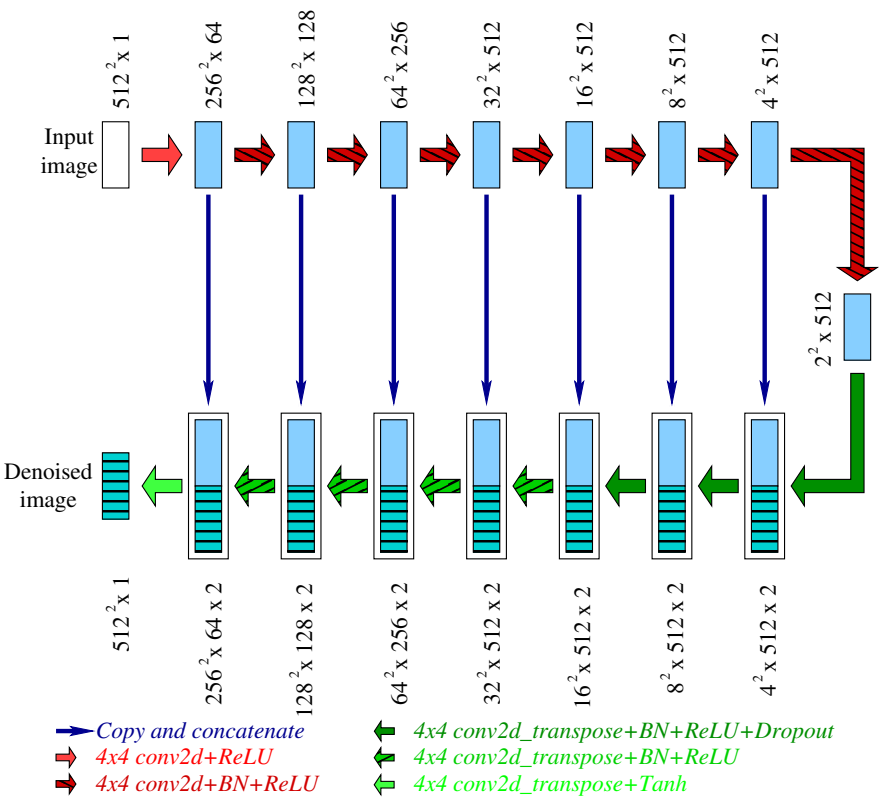


Fig. 1. Schematic diagram of the proposed deep encoder-decoder network architecture.

Figure 1 shows the detailed structure of the proposed encoder-decoder. The symmetric u-shape of the network can be in particular noticed, with the contracting path which consists of the encoding layers (left-right arrows) on the top part, while the expansive path on the bottom part is made of the same number of decoding layers (right-left arrows). Each light blue box represents a set of feature maps issued from an encoding layer and each greenish-blue box with horizontal line pattern one from a decoding layer. Obviously, the size of the input image defines an upper bound on the number of layers in the encoder and decoder. The x-y size of the feature maps, as well as their number, is provided on the top (encoding part) or bottom (decoding part) of each box. For example, in the case of the decoder, $4^2 \times 512 \times 2$ means that there are 1024 maps of size 4^2, where 512 maps are the result of the decoding of the bottleneck layer and the other 512 ones the higher resolution features map copied from encoder. The different arrows denote the different operations. On the one hand convolution operations are represented by left-right arrows and on the other hand deconvolution ones by right-left arrows. The respective TensorFlow module implementing each operation, namely *conv2d* and *conv2d_transpose*, are used in the labels.

3.2 Loss Function

A factor that has a major impact on the obtained neural network is obviously the loss function used to drive the training process. Despite its importance, the choice of this function is hardly ever discussed in most research works. Usually the choice simply consists in deciding whether to use the $L1-$norm or the $L2-$norm, the latter being the most popular option. However, even if properties of the $L2-$norm explain why it is the default choice, in the case of image restoration tasks and particularly for image denoising it is disputable. First, the key objective of image denoising is to improve the visual quality from a human observer's point of view and the $L2-$norm is clearly not correlated with this desirable objective. Second, it is known that the Euclidean metric is optimal when white Gaussian noise is encountered, but for other noise schemes alternative metrics should be considered [5]. Therefore, a loss function that is based on a metric reflecting the visual quality should be investigated.

Such an investigation can be found in [22], a paper in which the authors compared several losses considering two state-of-the-art metrics for image quality: the Structural SIMilarity (SSIM) index [16] and the MultiScale Structural SIMilarity (MS-SSIM) index [17]. They compared both norms, SSIM, MS-SSIM, and their own loss function that is a combination of MS-SSIM and $L1-$norm on different image restoration tasks, among which joint denoising and demosaicking of color image patches (31×31 pixels) using a FCN of three layers with PReLU activation in the first two ones. We independently came up with the same idea to investigate a loss function that combines the losses $\mathcal{L}^{L1}$ and $\mathcal{L}^{SSIM}$ denoted by $\mathcal{L}^{L1+SSIM}$ in the following. It should be noted that the work presented in [22] focuses on the analysis of loss functions and not on the design of a FCN for image denoising. Formally, $\mathcal{L}^{L1}$ and $\mathcal{L}^{SSIM}$ losses are defined by:

$$\mathcal{L}^{L1}(x,y) = \frac{1}{|x|}\sum_{p\in x}|x-y|, \mathcal{L}^{SSIM}(x,y) = 1-\frac{1}{|x|}\sum_{p\in x}\frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1}\cdot\frac{2\sigma_{xy}+C_2}{\sigma_x^2+\sigma_y^2+C_2} \tag{3}$$

where x is the noisy version of the clean image y and μ, σ, are means and standard deviations that depend on pixel p. Both are computed using a Gaussian filter with standard deviation σ_G. $C_1 = (K_1L)^2$ and $C_2 = (K_2L)^2$ are two constants, where L is the dynamic range of the pixel values (1 in our case due to normalization in $[0;1]$), $K_1 \ll 1$ and $K_2 \ll 1$. In fact, the SSIM index is a similarity measure that combines three comparison functions measuring different kinds of changes between images: luminance, contrast, and structure, but thanks to a simplification it can be expressed as a product of two terms.

4 Experimental Results

4.1 Data Set and Network Training

For image denoising, images from the Berkeley Segmentation Data Set (BSD or BSDS)[2] [13] are widely used for training and testing. For example, in [12]

[2] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

they used the 300 images from BSDS300 to generate patches for training and 200 images for testing (the 200 fresh images from BSDS500). In [19], Zhang *et al.* followed [3] and hence trained their image denoising model DnCNN using 400 BSD images considering three different noise levels. More precisely, for each noise level they cropped $128 \times 1,600$ patches of size 40×40. In [20], they used a similar approach to train FFDNet for AWGN denoising in grayscale images.

However, it is admitted that the training of deep networks can benefit from a large data set and therefore the question of extending the routinely used small BSD training set arises. Hence, in their most recent works [20, 21], Zhang *et al.* not only considered 400 images from BSD, but also selected 400 images from the validation set of ImageNet database and $4,744$ images of Waterloo Exploration Database [11]. According to their experimental study in [21], the training with an enlarged data set does not improve the denoising performance. In a first evaluation we do not consider the BSD data set, neither for training, nor for testing. We used as data set a subset of the $10,000$ gray images of 512×512 pixels provided by the BOSS database [1]: the first $3,000$ images of the database are used, with $2,800$ images for training and the 200 remaining ones for testing.

A network is trained during 50 epochs for a specific type of noise, using the Adam optimizer [9]. The traditional SGD is replaced due to the observation of Mao *et al.* [12] that Adam provides a faster training convergence of their encoder-decoder networks. The computations have been completed on a NVIDIA Tesla Titan X GPU, with a training time for a given noise level of about 5 hours.

4.2 Denoising Performance

A measure used to assess the denoising performance of an approach is the Peak Signal-to-Noise Ratio (PSNR), even if it is known to be a poor quality metric when the purpose is to compare the images as perceived by the human visual system. Indeed, a high PSNR value and good visual quality do not necessarily go together [16]. It is rather the simultaneous taking into account of PSNR and mean SSIM index which is a good indicator of the visual quality: when both metrics have high values, the quality is regarded as high.

Quantitative Results Table 1 shows the quantitative results gained for AWGN, including noise levels $\sigma \in \{10, 30, 50, 70, 80\}$, and speckle reduction on the test set of 200 images. The speckle noise is modeled as a multiplicative noise that follows a Gamma distribution $\Gamma(L, 1)$ of unit mean and variance $\frac{1}{L}$, where $L = 1$. In each case the average PSNR and SSIM values of the noisy input images are given, as well as the corresponding outcomes produced by BM3D, and those issued by the proposed encoder-decoder and DnCNN. The results of DnCNN have been obtained by using directly network models provided by the authors in the `GitHub`[3] of their Matlab implementation. For the speckle case, the BM3D values have not been computed since the corresponding SAR version should have been used, while DnCNN is dropped due its focus on Gaussian denoising.

[3] https://github.com/cszn/DnCNN

Table 1. Average PSNR (dB) / SSIM obtained for AWGN and speckle.

AWGN		$\mathcal{L}^{L1+SSIM}$		
σ	Noisy input images	BM3D	Encoder decoder	DnCNN
10	28.37 / 0.5798	36.97 / 0.9282	36.07 / 0.9273	37.23 / 0.9304
30	19.17 / 0.2002	31.02 / 0.8284	32.06 / 0.8626	31.11 / 0.8334
50	15.10 / 0.1052	27.56 / 0.7591	29.97 / 0.8181	27.45 / 0.7613
70	12.64 / 0.0664	24.97 / 0.7091	28.48 / 0.7865	24.55 / 0.7041
80	11.69 / 0.0545	23.76 / 0.6882	27.99 / 0.7743	
Speckle $L = 1$	10.24 / 0.1441		27.86 / 0.7852	

Table 2. Average PSNR (dB) obtained by Zhang *et al.* [20] for AWGN on BSD68.

AWGN σ	BM3D	WNNM	MLP	DnCNN	FFDNet
15	31.07	31.37	-	31.72	31.63
25	28.57	28.83	28.96	29.23	29.19
35	27.08	27.30	27.50	27.69	27.73
50	25.62	25.87	26.03	26.23	26.29
75	24.21	24.40	24.59	24.64	24.79

As can be seen, the encoder-decoder can achieve satisfactory denoising results and outperforms almost systematically BM3D and DnCNN. Indeed, except for AWGN with $\sigma = 10$, in which case the encoder-decoder gives high values but slightly lower than those of BM3D and DnCNN, better PSNR and SSIM results are obtained. Moreover, the noisier the images, the more advantageous it is to use the encoder-decoder to recover clean images. For the speckle case, a comparison with the RED-NET results [6] shows that the proposal achieves a nearly similar performance for $L = 1$. Finally, we can notice that the encoder-decoder is able to deal with AWGN and speckle noise, once trained on the targeted noise, an important feature which is looked for in the perspective of blind denoising. Overall, the residual learning strategy adopted by DnCNN seems interesting for low noise levels, but as the noise increases the reconstruction of a clean image as performed by the proposed network is clearly more appropriate.

To further highlight the suitability of the encoder-decoder, it might be interesting to have an idea of the denoising performance given by other methods. Therefore, in Table 2 are shown the behaviors observed by Zhang *et al.* on BSD68 set [20] for AWGN removal with BM3D [4], WNNM [7], MLP [2], DnCNN, FFDNet. It can be seen that DnCNN and FFDNet outperform other methods. Even if these results are obtained on a different data set, considering the performances of BM3D and DnCNN in both tables as reference, the proposed encoder-decoder appears as a valuable competitor for state-of-the-art approaches.

We have also completed a preliminary evaluation of the encoder-decoder blind denoising ability for AWGN. To train the network, the first thousand image from

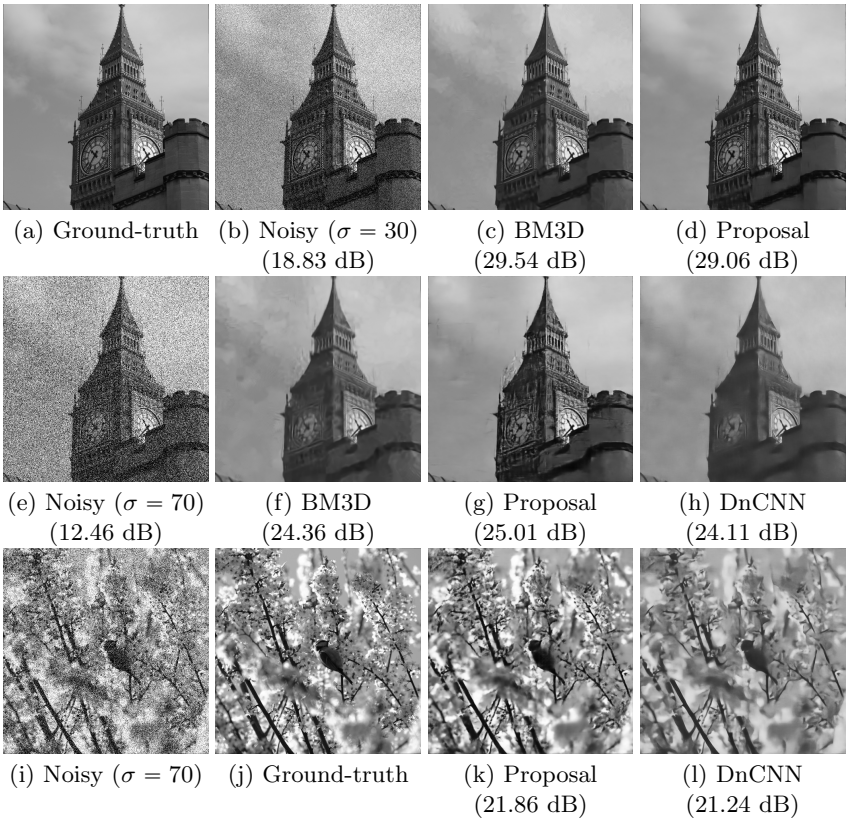
data set are used, where for each image its corresponding noisy versions with $\sigma = 10, 30, 50$, and 70 are computed. The training set size is thus increased by 43% (4,000 images), as is the computation time (7.2 hours). Once trained, this blind denoiser yields the following results for $\sigma = 80$: a PSNR of 27.50 dB and 0.7564 for SSIM value. Obviously, these values are inferior to the ones obtained with the network trained specifically for $\sigma = 80$ shown in Table 1. But they are slightly better than those given by the network trained only for $\sigma = 70$: a PSNR of 27.33 dB and 0.7542 for SSIM value. These results are encouraging but a deeper investigation is needed to confirm that the proposed network can be suitably trained to deal simultaneously with different unknown noise levels.

Visual Results Images (a) to (g) of Figure 2 illustrate the visual results of BM3D and the proposed deep network, considering a same image, for AWGN with $\sigma = 30$ and 70. It can be seen that the encoder-decoder preserves sharp edges and finer details as the noise level increases. This point is clearly highlighted through the comparison of images (f) and (g), since the clock on the left shaded part of Big Ben appears far blurrier with BM3D. Furthermore, even if for $\sigma = 30$ the PSNR result is better for BM3D, the neural network yields an image with a better visual quality: a look on the cloudy upper left part in the images (c) and (d) is convincing in our opinion. This observation is further supported by the SSIM value which is equal to 0.9021 for the clean image recovered by the encoder-decoder, whereas the one for BM3D is equal to 0.8929.

Figure 2 also shows the denoising results on two different images with noise level 70 given by DnCNN. In both cases the proposed network recovers a clean image with far more details and a better visual quality. This is again confirmed by the higher values of PSNR and SSIM: for Big Ben the values obtained from the image recovered by the encoder-decoder are, respectively, 25.01 dB and 0.8204 *versus* 24.11 dB and 0.7875 for DnCNN, while for the image with the bird they are 21.86 dB and 0.7162 *versus* 21.24 dB and 0.6547. In the case of the speckle noise presented in Figure 3, despite the huge corruption interesting details are brought out, especially in the shaded part of the building.

5 Conclusion

In this paper, a fully convolutional network that consists in an encoder-decoder with skip connections has been proposed for image denoising. The great lines of the network have been presented and the choice of the loss function used to carry out the training discussed. The results obtained on grayscale images show that the network can remove AWGN and multiplicative speckle noise, provided that it is suitably trained for the targeted noise. Moreover, compared to some competing approaches for image denoising, the network appears to be able to produce state-of-the-art denoising results. Finally, a preliminary evaluation of its ability to address blind Gaussian denoising has yielded favorable performance.



(a) Ground-truth (b) Noisy ($\sigma = 30$) (18.83 dB) (c) BM3D (29.54 dB) (d) Proposal (29.06 dB)

(e) Noisy ($\sigma = 70$) (12.46 dB) (f) BM3D (24.36 dB) (g) Proposal (25.01 dB) (h) DnCNN (24.11 dB)

(i) Noisy ($\sigma = 70$) (j) Ground-truth (k) Proposal (21.86 dB) (l) DnCNN (21.24 dB)

Fig. 2. AWGN denoising results (PSNR) of an image with noise level $\sigma = 30$: (a)-(d) and two images with noise level $\sigma = 70$: (e)-(h) for Big Ben and (i)-(l) for the bird.

(a) Ground-truth (b) Noisy ($L = 1$) (7.18 dB) (c) Proposal (24.54 dB)

Fig. 3. Speckle denoising results (PSNR) of one image with $L = 1$.

6 Acknowledgement

This work has been supported by the EIPHI Graduate School (contract "ANR-17-EURE-0002")

References

1. Bas, P., Filler, T., Pevný, T.: Break Our Steganographic System: The Ins and Outs of Organizing BOSS. In: 13th International Conference on Information Hiding. pp. 59–70. Springer, Heidelberg (2011)
2. Burger, H.C., Schuler, C.J., Harmeling, S.: Image Denoising: Can Plain Neural Networks Compete with BM3D? In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2392–2399. IEEE (2012)
3. Chen, Y., Pock, T.: Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6), 1256–1272 (2017)
4. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image Denoising by Sparse 3-D Transform-domain Collaborative Filtering. IEEE Transactions on image processing 16(8), 2080–2095 (2007)
5. François, D., Wertz, V., Verleysen, M., et al.: Non-Euclidean Metrics for Similarity Search in Noisy Datasets. In: 13th European Symposium on Artificial Neural Networks (ESANN). pp. 339–344 (2005)
6. Gu, F., Zhang, H., Wang, C., Zhang, B.: Residual Encoder-decoder Network Introduced for Multisource SAR image Despeckling. In: 2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA). pp. 1–5. IEEE (2017)
7. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted Nuclear Norm Minimization with Application to Image Denoising. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2862–2869 (2014)
8. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-Image Translation with Conditional Adversarial Networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5967–5976. IEEE (2017)
9. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: 3rd International Conference for Learning Representations (ICLR) (2015)
10. Long, J., Shelhamer, E., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3431–3440. IEEE (2015)
11. Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., Zhang, L.: Waterloo Exploration Database: New Challenges for Image Quality Assessment Models. IEEE Transactions on Image Processing 26(2), 1004–1016 (2017)
12. Mao, X., Shen, C., Yang, Y.: Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29 (NIPS 2016). pp. 2802–2810. Curran Associates, Inc. (2016)
13. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In: 8th IEEE International Conference on Computer Vision (ICCV). vol. 2, pp. 416–423. IEEE (2001)

14. Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In: 4th International Conference for Learning Representations (ICLR) (2016)
15. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional Networks for Biomedical Image Segmentation. In: 2015 International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). vol. 9351, pp. 234–241. Springer, Heidelberg (2015)
16. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: from Error Visibility to Structural Similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)
17. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale Structural Similarity for Image Quality Assessment. In: 37th Asilomar Conference on Signals, Systems Computers. vol. 2, pp. 1398–1402. IEEE (2003)
18. Xie, J., Xu, L., Chen, E.: Image Denoising and Inpainting with Deep Neural Networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25 (NIPS 2012). pp. 341–349. Curran Associates, Inc. (2012)
19. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. IEEE Transactions on Image Processing 26(7), 3142–3155 (2017)
20. Zhang, K., Zuo, W., Zhang, L.: FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising. IEEE Transactions on Image Processing 27(9), 4608–4622 (2018)
21. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning Deep CNN Denoiser Prior for Image Restoration. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2808–2817. IEEE (2017)
22. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss Functions for Image Restoration With Neural Networks. IEEE Transactions on Computational Imaging 3(1), 47–57 (2017)