A NOTE ON THE SIZE OF DENOISING NEURAL NETWORKS*

YI-QING WANG†

Abstract. Patch based denoising algorithms seek to approximate the conditional expectation of clean patches given their related noisy observations. In this note, we give a probabilistic account of how various algorithms approach this problem and in particular, we argue that small neural networks can denoise small-scale texture patterns almost as well as their large counterparts. The analysis further indicates that self-similarity and Bayesian approaches such as neural networks are complementary paradigms for patch denoising, which we illustrate with an algorithm that effectively complements BM3D with small neural networks, thereby outperforming BM3D with minor additional cost.

Key words. small neural networks, small-scale texture denoising

AMS subject classifications.

1. Introduction.

1.1. Overview of image denoising. Image denoising aims to recover an image from its noisy observation affected by Gaussian noise of a known standard deviation. A typical image denoising algorithm involves identifying and modelling natural image regularities, whether they lie in the spatial or transform domains. Historically, spatial domain methods such as total variation [25], anisotropic diffusion [22] and bilateral filter [27] all grew out of a desire to preserve image continuity, edges in particular. On the other hand, transform domain methods were developed on the belief that natural images can be sparsely represented with certain linear transforms. Thanks to coefficient shrinkage, they also led to effective algorithms [12, 13, 26, 24].

Since the appearance of non-local means [5], development in this field has blurred the distinction between these two approaches. BM3D [9], for instance, uses spatial block matching to obtain similar patches, which are then jointly filtered in a fixed basis with coefficient shrinkage. In addition, how best to construct a basis has also been investigated, which becomes an umbrella subject known as dictionary learning and ultimately leads the research community to look beyond the noisy image and seek data driven priors. Various techniques including Gaussian mixture [33, 28] as well as unsupervised and supervised K-SVD [14, 21, 20] have been successfully employed.

Increasingly powerful and accessible computing facilities also make another line of research possible. Rather than try to summarize the analytically intractable patch space with dictionaries, one can now draw extensive samples from it using external image databases, which produced impressive results [19, 32]. Moreover, with training cost no longer as prohibitive as it once was, researchers have also started to harness the potential of neural networks [3]. Convolutional neural networks [16], stacked sparse auto-encoders [31] and plain multilayer neural networks [7] all have had success at image denoising. Several areas for improvements have been suggested as well, ranging from activation function [30], latent representation [8], to training schema and extension to other noise levels and types [29, 1].

1.2. Patch denoising and its probabilistic interpretation. Patch denoising, which lies at the heart of most denoising algorithms, is our focus in what follows.

†CMLA, Ecole Normale Supérieure de Cachan, 94230 Cachan, France (yqwang9@gmail.com).

It is concerned with estimating a noise-free patch X from a noisy observation $\tilde{Y}$

$$\tilde{Y} = Y + N$$

where Y is also noise-free and N zero-mean Gaussian noise with a known standard deviation σ. Y is related to X although we do not make any assumption on its nature until later. The ideal filter in the sense of mean squared error (MSE) is thus the conditional expectation $\mathbb{E}_{\mathbb{P}}[X|\tilde{Y}]$ because of the equality

$$\mathbb{E}_{\mathbb{P}}[X|\tilde{Y}] = \underset{f}{\operatorname{argmin}} \, \mathbb{E}_{\mathbb{P}}\|f(\tilde{Y}) - X\|_2^2, \quad \mathbb{P} \; a.e.$$

where the probability $\mathbb{P}$ is the triplet $(X, Y, \tilde{Y})$'s joint distribution and f an estimator of X from $\tilde{Y}$.

When X and $\tilde{Y}$ form a Gaussian family, this conditional expectation is the classic Wiener filter [17]. More generally, a version of this $\mathbb{P}$ almost surely defined function of $\tilde{Y}$ can be estimated with weakly correlated samples $(x_i, y_i)_{i \geq 1}$ of (X, Y) thanks to the equality

$$\mathbb{E}_{\mathbb{P}}[X|\tilde{Y} = \tilde{y}] = \lim_{n \to \infty} \frac{\sum_{i=1}^n p(\tilde{y}|y_i) x_i}{\sum_{i=1}^n p(\tilde{y}|y_i)}, \quad \mathbb{P} \; a.e.$$

with

$$p(\tilde{y}|y_i) \propto \exp(-\frac{\|\tilde{y} - y_i\|_2^2}{2\sigma^2}).$$

This point-wise estimator was used as an upper bound of the optimal patch denoising MSE [19]. Despite its theoretical appeal, this estimator suffers from the curse of dimensionality of a generic Monte-Carlo simulation for its inability of performing importance sampling, that is, prioritizing the samples whose y component is close to $\tilde{y}$. In addition, given the huge variety of natural patches, the mere existence of a single noise-free candidate for each noisy patch requires a data storage capacity too demanding for practical use.

An alternative is to make direct use of noisy observations. Assuming that X and Y are two disjoint sets of pixels, we may use

$$\mathbb{E}_{\mathbb{P}}[X|\tilde{Y} = \tilde{y}] = \mathbb{E}_{\mathbb{P}}[\tilde{X}|\tilde{Y} = \tilde{y}] \approx \frac{\sum_{i=1}^n \tilde{x}_i \mathcal{W}(\|\tilde{y} - \tilde{y}_i\|_2)}{\sum_{i=1}^n \mathcal{W}(\|\tilde{y} - \tilde{y}_i\|_2)}$$

with the choice of weight function $\mathcal{W}(\cdot)$ reflecting our belief in the conditional expectation's smoothness. The proof of this estimator's asymptotic consistency is generally known as Stone's theorem [11]. Owing to self-similarity in natural images, this approximation scheme turns out to be very effective and was popularized by non-local means [5], leading eventually to BM3D [9].

Nonetheless these nonparametric estimators are constrained by the availability of data. A potential remedy is a parametric approach. However, unlike the Wiener filter which results from a parametric data modelling, we can look for a parametric approximation of the conditional expectation, which is possible with a large and well structured class of functions parameterized by a vector-valued θ:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \, \mathbb{E}_{\mathbb{P}}\|f_\theta(\tilde{Y}) - X\|_2^2 = \underset{\theta}{\operatorname{argmin}} \, \mathbb{E}_{\mathbb{P}}\|f_\theta(\tilde{Y}) - \mathbb{E}[X|\tilde{Y}]\|_2^2.$$

This is where multilayer feedforward neural networks come into play. Their fully-connected structure consists of a succession of non-linear hidden layers followed by a linear decoder

$$f_\theta(\cdot) = s \circ h_n \circ \cdots \circ h_1(\cdot), \;\; n \geq 1$$

with

$$\forall 1 \leq l \leq n, \;\; h_l(z) = \tanh(W_l z + b_l)$$

and

$$s(z) = W_{n+1} z + b_{n+1}$$

where the activation function $\tanh(\cdot)$ is applied element-wise and the parameter vector θ comprises all the connection weights and biases (W_l, b_l). Known to be universal approximators [2, 15], they are tuned with stochastic gradient descent, also called backpropagation [18, 4] due to their particular function structure. In spite of their non-convex training objective, neural networks enjoy widespread use thanks to their superior practical performances [3]. Several recently published image denoising neural networks [7], for instance, have outperformed BM3D [9], long held as the state-of-the-art. Unfortunately, with four hidden layers having over 2000 units each, these neural networks effectively require tens of millions of multiplication operations per pixel when denoising a grayscale image.

Since it is recognized [7] that neural networks do not strictly dominate BM3D, it was proposed [6] to train one more neural network of comparable size which takes in the denoised patches from both BM3D and neural networks in addition to the original noisy patch and outputs a combined result. This method improves both algorithms because, again in view of the conditional expectation, we have for any random triplet $(X, \tilde{Y}, \tilde{H})$

$$\mathbb{E}_\mathbb{P}\|X - \mathbb{E}_\mathbb{P}[X|\tilde{Y}]\|_2^2 - \mathbb{E}_\mathbb{P}\|X - \mathbb{E}_\mathbb{P}[X|\tilde{Y}, \tilde{H}]\|_2^2 = \mathbb{E}_\mathbb{P}\|\mathbb{E}_\mathbb{P}[X|\tilde{Y}, \tilde{H}] - \mathbb{E}_\mathbb{P}[X|\tilde{Y}]\|_2^2 \geq 0,$$

that is, more information helps lower the theoretical MSE bound. However, the proposed approach doubles the already heavy computational load.

1.3. Our contribution. We investigate whether it is possible to scale down the neural networks while preserving their performance. It is argued that as noise increases, small neural networks are a better alternative to self-similarity for small-scale texture pattern denoising and that large neural networks and their heavy computational cost may be unnecessary for this specific task. Thanks to a conditional expectation decomposition, we show that self-similarity and Bayesian methods such as neural networks are complementary approaches to patch denoising and propose an algorithm to combine BM3D and small neural networks geared towards granular texture denoising, which achieves better performance than either individually at minor additional cost.

The paper is organized as follows: Section 2 argues and presents numerical evidence that large neural networks are not necessary for denoising small-scale texture patterns and that neural networks and self-similarity are complementary patch denoising paradigms. Section 3 presents a principled framework which effectively complements BM3D with small neural networks. Section 4 concludes.

2. Small-scale texture pattern denoising. Patterns in natural images tend to repeat themselves. Even the objects composing a random texture are of regular shape when viewed at a close distance relative to their dimensions. Non-local means [5] and BM3D [9] use this prior to find similar neighboring patches to denoise. An algorithm's ability to adapt patch size and shape to image content is also crucial [10] because self-similarity is present at different scales of observation. However, there is a limit to which this strategy can be meaningfully pursued because the smallest unit of observation in digital images is a pixel. As a consequence, self-similarity may not be reliably exploited on small-scale texture patterns affected by strong noise.

Neural networks thus offer a valuable solution. Moreover, if the pixel interactions in certain texture areas are of short-range nature [23], it may even be possible to replace large neural networks by smaller ones. For an experimental investigation, we set up three identical neural networks acting on a 7-by-7 noisy input to estimate its central 3-by-3 block. Comprising three hidden layers with 147 units each, their architecture can be summarized as 49-147-147-147-9. They were trained 2×10^8 rounds on the grayscale *Pascal VOC2012* dataset under Gaussian noise standard deviation set to 10, 25, 35 respectively in order to match the published ones [7], whose architectures are 441-2047-2047-2047-2047-81, 1521-3072-3072-2559-2047-289 and 1521-3071-3071-2559-2047-289.

Two images were selected for testing. The first, of dimension 746×264, was cropped from a standard *Kodak PhotoCD* benchmark image rendered to grayscale with *matlab*'s `rgb2gray` function. It was chosen because of its rich small-scale texture content. The second 1280-by-1280 image has artificially generated structured patterns. Neither image was used in the training or validation set. We compared the standard BM3D, BM3D$_4$ which is BM3D with reduced patch size (4 instead of 8) at the first round of block matching, BM3D-SAPCA [10] and small and large neural networks.

Fig. 1. *The two test images*

Table 1 indicates that small neural networks denoised small-scale texture patterns almost as well as their large counterparts (also see Figure 2). On the other hand, BM3D underperformed both neural networks where self-similarity was lacking on its own fixed scale. Thanks to a better matched patch size, BM3D$_4$ did better albeit at the price of much poorer showing on highly structured and large-scale patterns, a problem shared by small neural networks versus the large ones. BM3D's 39-by-39 search window, vis-a-vis the large neural networks' input size (21-by-21 for σ=10 and 39-by-39 for the rest) also shows that it is more effective under lower noise. With additional adaptivity from flexible patch size and shape, BM3D-SAPCA [10] does much better, although it also loses its edge as noise grows because it becomes

$\sigma = 10$	BM3D	BM3D$_4$	BM3D-SAPCA	small	large
structure	45.79	42.08	**46.84**	40.45	43.78
texture	29.95	30.00	30.20	30.09	**30.22**

$\sigma = 25$	BM3D	BM3D$_4$	BM3D-SAPCA	small	large
structure	39.10	34.65	39.90	35.11	**40.24**
texture	24.66	24.89	24.90	24.92	**25.11**

$\sigma = 35$	BM3D	BM3D$_4$	BM3D-SAPCA	small	large
structure	36.67	31.94	37.35	32.80	**38.42**
texture	23.17	23.30	23.28	23.37	**23.59**

increasingly difficult to determine similarity, especially on small-scale texture patterns. Computationally speaking, on a grayscale image, these small neural networks require roughly 5×10^4 operations per pixel, which is more than 500 times cheaper than their large counterparts and makes them on a par with BM3D because BM3D needs 4×10^4 operations per pixel if all of its transforms are implemented with a time complexity equal to $N \log_2 N$ [9].

3. Complementing self-similarity with small neural networks. Loosely speaking, a natural image consists of texture and structure, whose definition may depend on the scale of observation. Formally, we thus consider $\mathbb{P}$ the distribution which governs the patterns on a fixed patch size. Let it be the sum $\alpha\mathbb{T} + (1 - \alpha)\mathbb{S}$ for some $\alpha \in (0, 1)$ and $\mathbb{T}$, $\mathbb{S}$ two probabilities responsible for texture and structure generation. The next theorem helps understand the behaviour of an ideal filter.

THEOREM 3.1. *Let $\mathbb{P}, \mathbb{T}, \mathbb{S}$ be three probabilities defined on a common measurable space satisfying $\mathbb{P} = \alpha\mathbb{T} + (1 - \alpha)\mathbb{S}$ for some $\alpha \in (0, 1)$. Let U, V be two random vectors with $\mathbb{E}_{\mathbb{P}}\|U\|_1 < +\infty$ defined on the same space. Then we have*

$$(3.1) \qquad \mathbb{E}_{\mathbb{P}}[U|V] = \alpha\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{T}}{d\mathbb{P}}|V]\mathbb{E}_{\mathbb{T}}[U|V] + (1-\alpha)\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{S}}{d\mathbb{P}}|V]\mathbb{E}_{\mathbb{S}}[U|V], \quad \mathbb{P} \ a.e.$$

where $\frac{d\mathbb{T}}{d\mathbb{P}}$ and $\frac{d\mathbb{S}}{d\mathbb{P}}$ are the Radon-Nikodym derivatives of $\mathbb{T}$ and $\mathbb{S}$ with respect to $\mathbb{P}$.
Proof: by the definition of a conditional expectation, we have for any bounded Borel function $\phi(\cdot)$

$$
\begin{aligned}
\mathbb{E}_{\mathbb{P}}[\mathbb{E}_{\mathbb{P}}[U|V]\phi(V)] =& \mathbb{E}_{\mathbb{P}}[U\phi(V)] \\
=& \alpha\mathbb{E}_{\mathbb{T}}[U\phi(V)] + (1-\alpha)\mathbb{E}_{\mathbb{S}}[U\phi(V)] \\
=& \alpha\mathbb{E}_{\mathbb{T}}[\mathbb{E}_{\mathbb{T}}[U|V]\phi(V)] + (1-\alpha)\mathbb{E}_{\mathbb{S}}[\mathbb{E}_{\mathbb{S}}[U|V]\phi(V)] \\
(3.2) \qquad =& \alpha\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{T}}{d\mathbb{P}}\mathbb{E}_{\mathbb{T}}[U|V]\phi(V)] + (1-\alpha)\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{S}}{d\mathbb{P}}\mathbb{E}_{\mathbb{S}}[U|V]\phi(V)] \\
=& \mathbb{E}_{\mathbb{P}}[\Big(\alpha\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{T}}{d\mathbb{P}}|V]\mathbb{E}_{\mathbb{T}}[U|V] + (1-\alpha)\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{S}}{d\mathbb{P}}|V]\mathbb{E}_{\mathbb{S}}[U|V]\Big)\phi(V)].
\end{aligned}
$$

The equation (3.2) holds because both $\mathbb{T}$ and $\mathbb{S}$ are absolutely continuous with respect to $\mathbb{P}$ by construction. As a result, if we define $D(V)$ to be the first coordinate of the random vector

$$(3.3) \qquad \mathcal{V} := \alpha\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{T}}{d\mathbb{P}}|V]\mathbb{E}_{\mathbb{T}}[U|V] + (1-\alpha)\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{S}}{d\mathbb{P}}|V]\mathbb{E}_{\mathbb{S}}[U|V] - \mathbb{E}_{\mathbb{P}}[U|V],$$

(a)

(b) PSNR=20.17

(c) PSNR=25.99

(d) PSNR=26.09
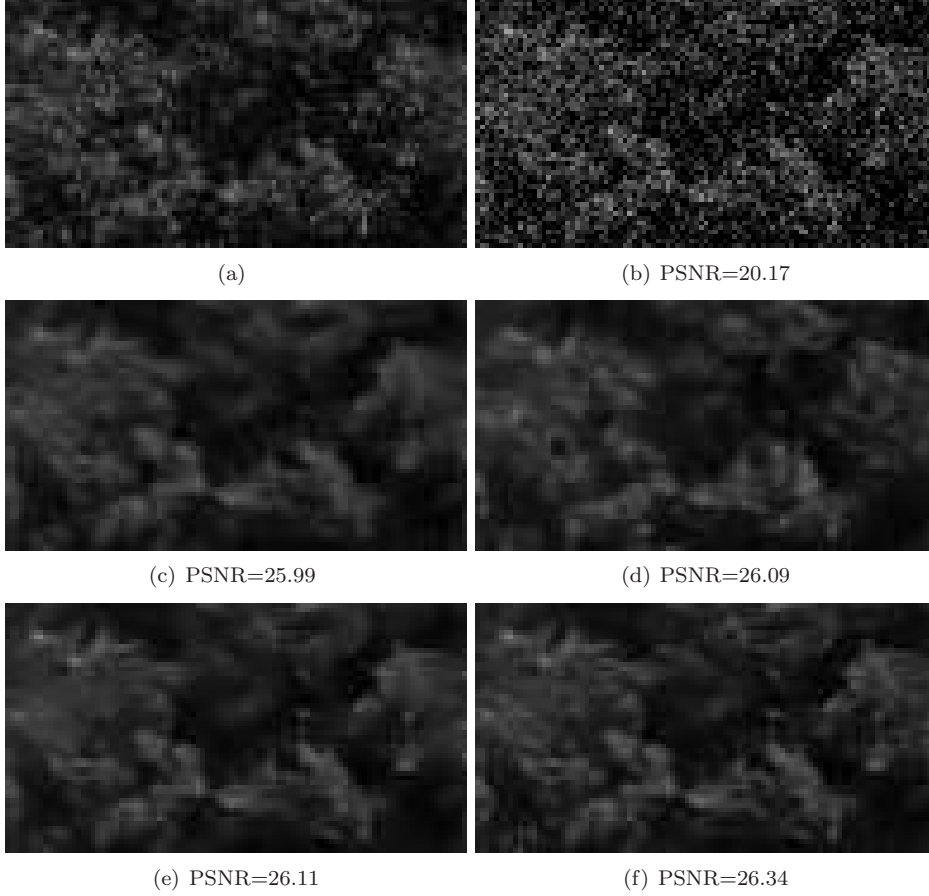
(e) PSNR=26.11

(f) PSNR=26.34

Fig. 2. *(a) original (b) noisy (c) BM3D (d) BM3D-SAPCA (e) small NN (f) large NN*

then for all $n \in \mathbb{N}^*$

$$\mathbb{E}_{\mathbb{P}}[D(V)1_{D(V)<-n^{-1}}] = \mathbb{E}_{\mathbb{P}}[D(V)1_{D(V)>n^{-1}}] = 0$$

because both $1_{D(\cdot)>n^{-1}}$ and $1_{D(\cdot)<-n^{-1}}$ are bounded. So we deduce

$$n^{-1}\mathbb{P}(|D(V)| > n^{-1}) \leq \mathbb{E}_{\mathbb{P}}[|D(V)|1_{|D(V)|>n^{-1}}] = \mathbb{E}_{\mathbb{P}}[D(V)1_{D(V)>n^{-1}}] - \mathbb{E}_{\mathbb{P}}[D(V)1_{D(V)<-n^{-1}}] = 0$$

which implies

$$\mathbb{P}(|D(V)| > 0) = \lim_{n\to\infty} \mathbb{P}(|D(V)| > n^{-1}) = 0 \Leftrightarrow \mathbb{P}(D(V) = 0) = 1.$$

The same reasoning applies to the random vector $\mathcal{V}$'s other coordinates. Hence the probability for it to vanish is 1. $\square$

The classic likelihood ratios [11] found in the conditional expectation decomposition satisfy

$$\alpha\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{T}}{d\mathbb{P}}|V] + (1-\alpha)\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{S}}{d\mathbb{P}}|V] = \mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{P}}{d\mathbb{P}}|V] = 1.$$

If we interpret $\mathbb{E}_{\mathbb{T}}[U|V]$ and $\mathbb{E}_{\mathbb{S}}[U|V]$ as the patches restored from a noisy observation V by two filters dedicated to denoising texture and structure, and the weights $\alpha\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{T}}{d\mathbb{P}}|V]$ and $(1-\alpha)\mathbb{E}_{\mathbb{P}}[\frac{d\mathbb{S}}{d\mathbb{P}}|V]$ as a measure of how certain we are of V being part of texture as oppposed to structure, the previous decomposition shows that an ideal filter essentially combines the outputs of two specialised filters on the basis of a noisy input's perceived content type. Therefore a good filter mechanically involves pattern classification and estimation. Intricately built, a well trained large neural network has filters dedicated to structure and texture as well as a classification device entangled in a single black box. In light of the previous section, it is tempting to replace them by less costly alternatives, namely, BM3D and small neural networks. As to the classification, we may resort to a zero-one rule so that unlike the weighting scheme in the conditional expectation decomposition, a noisy patch is processed by one and only one filter.

This leads to our algorithm SSaNN, for self-similarity and neural network, which relies on a texture detector to switch between BM3D and small neural networks. Given a noisy image and its noise level, it first produces two denoised versions by the small neural network trained under the same noise condition and BM3D. Then it extends the noisy image in order to form a one-to-one mapping between the 7-by-7 patches in the extended image and the pixels in the original noisy image by associating a patch with its central pixel. Next, depending on the number of obtained ℓ_2-similar neighbors, the patches are classified as either texture or structure, a label then transfered to their associated pixel. Finally, the algorithm combines the two denoised image versions accordingly. For a faster execution, one could classify the pixels first and then choose to denoise them with the neural network or BM3D. Though here we use BM3D as a leading example of self-similarity based algorithms, there is no difficulty of substituting it by another more advanced variant. Their results are reported in Table 2 with the parameters of Algorithm 1 set to h=7, κ=12, n=3 and

$$\beta = \left\{ \begin{array}{ll} 1.7, & \sigma = 10 \\ 1.2, & \sigma = 25 \\ 1.1, & \sigma = 35 \end{array} \right.$$

which reflects a decreasing signal to noise ratio. Our test set (Figure 3) has the five large natural images acquired under good lighting conditions and downsampled specifically to ensure their quality and pattern diversity.

(a)　　　　　(b)　　　　　(c)　　　　　(d)　　　　　(e)

FIG. 3. *Test images (a) computer (b) garden (c) girl (d) traffic (e) valldemossa. All images are of dimension* 704×469 *except for valldemossa (*769×338*).*

Algorithm 1 Texture detector

1: **Input:** a noisy image and a reference patch.
2: **Parameter:** Gaussian noise standard deviation σ, patch size h, search window size κ, similarity threshold β, required number of similar neighbors n.
3: Compute the ℓ_2-norm of the difference between the reference patch and all its surrounding patches in the search window and declare a neighboring patch similar if the distance is smaller than $\sqrt{2}\beta\sigma h$.
4: Label the reference patch as structure if more than n similar patches have been found.

Algorithm 2 SSaNN

1: **Input:** noisy image $\tilde{I}$.
2: **Output:** denoised image.
3: **Parameter:** Gaussian noise standard deviation σ.
4: Denoise $\tilde{I}$ with BM3D and the neural network to get I_B and I_N.
5: Extend $\tilde{I}$ to $\bar{I}$ so that $\bar{I}$'s 7-by-7 patches form a one-to-one mapping with $\tilde{I}$'s pixels.
6: Use Algorithm 1 to classify $\bar{I}$'s patches (hence $\tilde{I}$'s pixels) into either structure or texture. Form a matrix T_S of the same size as $\tilde{I}$ with its entry at position (i,j) set to 1 if $\tilde{I}(i,j)$ is labeled structure and 0 otherwise.
7: Return $I_N \circ (1 - T_S) + I_B \circ T_S$ where $\circ$ denotes the element-wise product.

TABLE 2
Algorithm comparison in PSNR

$\sigma = 10$	small	BM3D	SSaNN$_1$	BM3D-SAPCA	SSaNN$_2$	large
computer	34.38	34.73	34.86	**35.26**	35.19	34.54
girl	38.74	40.58	40.60	**40.90**	**40.90**	40.33
garden	30.79	30.61	30.76	**30.88**	30.83	30.82
traffic	33.08	33.08	33.19	**33.47**	33.44	33.24
valldemossa	31.77	31.69	31.82	**31.94**	31.91	31.83
average	33.75	34.14	34.25	**34.49**	34.45	34.15

$\sigma = 25$	small	BM3D	SSaNN$_1$	BM3D-SAPCA	SSaNN$_2$	large
computer	29.21	29.90	29.97	**30.29**	30.26	29.84
girl	33.91	36.69	36.69	36.91	36.91	**37.21**
garden	25.14	25.00	25.16	25.13	25.21	**25.41**
traffic	27.97	28.13	28.24	28.37	28.39	**28.47**
valldemossa	26.34	26.29	26.44	26.43	26.52	**26.59**
average	28.51	29.20	29.30	29.43	29.46	**29.50**

$\sigma = 35$	small	BM3D	SSaNN$_1$	BM3D-SAPCA	SSaNN$_2$	large
computer	27.43	28.21	28.23	**28.56**	28.51	28.22
garden	23.46	23.39	23.52	23.44	23.54	**23.74**
girl	31.86	35.07	35.07	35.21	35.21	**36.00**
traffic	26.34	26.63	26.69	26.80	26.81	**26.96**
valldemossa	24.68	24.71	24.83	24.80	24.88	**25.00**
average	26.75	27.60	27.67	27.76	27.79	**27.98**

In Table 2, SSaNN$_1$ (resp. SSaNN$_2$) denotes the combination of BM3D (resp. BM3D-SAPCA) with the small neural networks. It demonstrates that SSaNN$_1$ effec-

tively represents an improvement over both BM3D and small neural networks operating individually on images with small-scale texture content (see Figure 4). For images with little detectable small-scale texture, SSaNN$_1$ is reduced to BM3D. Coherent with Table 1, BM3D-SAPCA, with less scale mismatch but a considerably higher run-time, did better alone under lower noise.
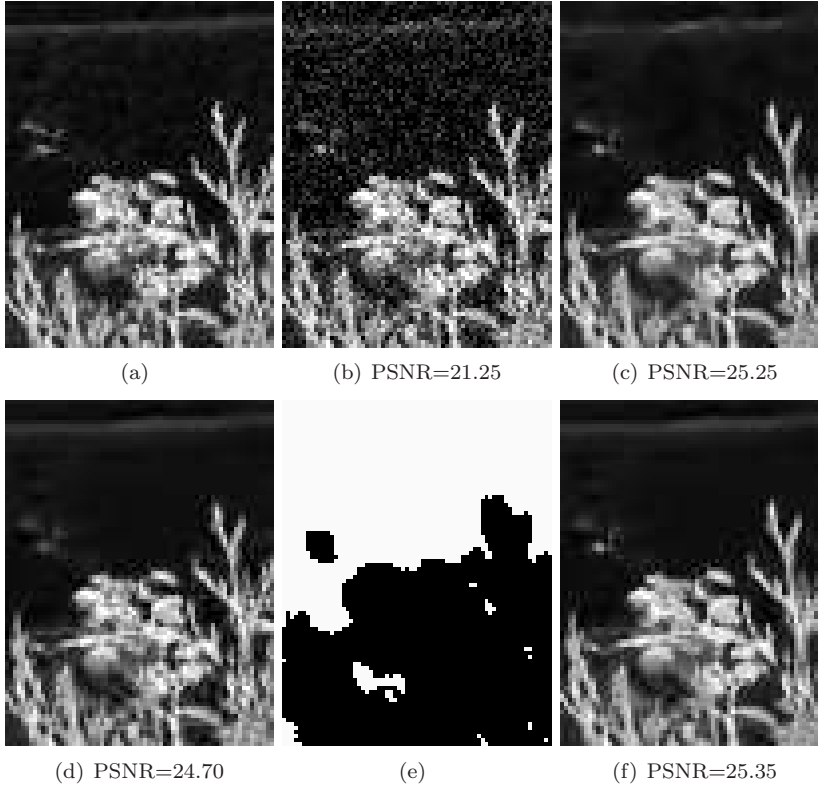

| (a) | (b) PSNR=21.25 | (c) PSNR=25.25 |
| (d) PSNR=24.70 | (e) | (f) PSNR=25.35 |

FIG. 4. *Experimental results: (a) a clean zoomed in example from Figure 3(b). (b) its noisy version with $\sigma = 25$. (c) restored image by the small neural network. (d) restored image by BM3D. (e) detected texture and structure zones (marked in black and white respectively). (f) combined result. Due to its small observation window, the neural network is unable to restore large scale regular patterns (the horizontal edge in the example) correctly. But as indicated by the PSNRs, it does better on small-scale texture.*

4. Discussion and conclusion. Self-similarity is one of nature's fundamental properties, whose usefulness in image processing however is constrained by the finite resolution of digital images and the presence of noise. As a result, it is crucial for a self-similarity guided denoising algorithm to adapt the patch size to the scale of image content and to the noise level. If a patch is too big with respect to its context, similar neighbors might not exist. On the other hand, if a patch is too small relative to the noise level, it is hard to correctly identify similar patches among its neighboring candidates via block matching. Restoring small-scale texture patterns under relatively strong noise is therefore a challenge.

Neural networks are a possible solution because they are made scale and noise conscious by explicitly seeking to approximate a conditional expectation in their training. However, for them to correctly handle large-scale and typically highly structured

patterns on their own, they need an even larger observation window, which unfortunately entails high computational cost as a result of their broad hidden layers and deep architectures.

To scale down large neural networks thus translates into making use of the complementary nature of self-similarity and Bayesian methods such as neural networks. Concretely, by having a self-similarity based algorithm work on larger-scale patterns and small neural networks focus on smaller-scale patterns, one may hope to get the best of both worlds without having to worry about the technical detail of patch size. Still, contrary to the soft weighting scheme of the ideal conditional expectation decomposition, a filter built upon a hard scale classification is more prone to errors because the accuracy of SSaNN's texture detector decreases with noise.

To conclude, in this work, we highlighted the conditional expectation because it not only helps understand most patch based denoising algorithms, but also inherently easier to analyze and approximate than its underlying law. In addition, we showed through experiments that the advantage of small neural networks is two-fold: they withstand relatively stronger noise better than BM3D and its variants on small-scale texture patterns and the notion of scale is transparent because of their fixed input size. Moreover, following the spirit of a conditional expectation decomposition, we presented a novel light-weight algorithm to effectively combine small neural networks with BM3D.

REFERENCES

[1] F. Agostinelli, M. R. Anderson, and H. Lee. Adaptive Multi-Column Deep Neural Networks with Application to Robust Image Denoising. In *Advances in Neural Information Processing Systems*, pages 1493–1501, 2013.

[2] A. R. Barron. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14(1):115–133, 1994. `http://dx.doi.org/10.1007/BF00993164`.

[3] Y. Bengio. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009. `http://dx.doi.org/10.1561/2200000006`.

[4] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the 19th International Symposium on Computational Statistics*, pages 177–186. Springer, 2010. `http://dx.doi.org/10.1007/978-3-7908-2604-3_16`.

[5] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005. `http://dx.doi.org/10.1145/344779.344972`.

[6] H. C. Burger, C. Schuler, and S. Harmeling. Learning how to combine internal and external denoising methods. In *Pattern Recognition*, pages 121–130. Springer, 2013.

[7] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *Computer Vision and Pattern Recognition*, pages 2392–2399. IEEE, 2012. `http://dx.doi.org/10.1109/CVPR.2012.6247952`.

[8] K. Cho. Simple Sparsification Improves Sparse Denoising Autoencoders in Denoising Highly Corrupted Images. In *Proceedings of the 30th International Conference on Machine Learning*, pages 432–440, 2013.

[9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. `http://dx.doi.org/10.1109/TIP.2007.901238`.

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Bm3d image denoising with shape-adaptive principal component analysis. In *Signal Processing with Adaptive Sparse Structured Representations*, 2009.

[11] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*, volume 31. Springer, 1996. `http://dx.doi.org/10.1007/978-1-4612-0711-5`.

[12] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994. `http://dx.doi.org/10.2307/2337118`.

[13] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.

http://dx.doi.org/10.2307/2291512.

[14] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. http://dx.doi.org/10.1109/TIP.2006.881969.

[15] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. http://dx.doi.org/10.1016/0893-6080(89)90020-8.

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

[17] S. M. Kay. Fundamentals of statistical signal processing: estimation theory. 1993.

[18] Y. LeCun, L. Bottou, G. B. Orr, and K. R. Müller. Efficient backprop. In *Neural Networks: Tricks of the Trade*, pages 9–50. Springer, 1998. http://dx.doi.org/10.1007/3-540-49430-8_2.

[19] A. Levin and B. Nadler. Natural image denoising: Optimality and inherent bounds. In *Computer Vision and Pattern Recognition*, pages 2833–2840. IEEE, 2011. http://dx.doi.org/10.1109/CVPR.2011.5995309.

[20] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, 2012. http://dx.doi.org/10.1109/TPAMI.2011.156.

[21] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *International Conference on Computer Vision*, pages 2272–2279. IEEE, 2009. http://dx.doi.org/10.1109/ICCV.2009.5459452.

[22] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990. http://dx.doi.org/10.1109/34.56205.

[23] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000. http://dx.doi.org/10.1023/A:1026553619983.

[24] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003. http://dx.doi.org/10.1109/TIP.2003.818640.

[25] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992. http://dx.doi.org/10.1016/0167-2789(92)90242-F.

[26] J.L. Starck, E. J Candès, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684, 2002. http://dx.doi.org/10.1109/TIP.2002.1014998.

[27] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *International Conference on Computer Vision*, pages 839–846. IEEE, 1998. http://dx.doi.org/10.1109/ICCV.1998.710815.

[28] Y. Q. Wang and J. M. Morel. SURE Guided Gaussian Mixture Image Denoising. *SIAM Journal on Imaging Sciences*, 6(2):999–1034, 2013.

[29] Y. Q. Wang and J. M. Morel. Can a Single Image Denoising Neural Network Handle All Levels of Gaussian Noise? *IEEE Signal Processing Letters*, 21(9):1150–1153, 2014.

[30] Y. Wu, H. Zhao, and L. Zhang. Image Denoising with Rectified Linear Units. In *Neural Information Processing*, pages 142–149. Springer, 2014. http://dx.doi.org/10.1007/978-3-319-12643-2_18.

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

[32] H. Yue, X. Sun, J. Yang, and F. Wu. CID: Combined Image Denoising in Spatial and Frequency Domains Using Web Images. In *Computer Vision and Pattern Recognition*, pages 2933–2940. IEEE, 2014. http://dx.doi.org/10.1109/CVPR.2014.375.

[33] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *International Conference on Computer Vision*, pages 479–486. IEEE, 2011. http://dx.doi.org/10.1109/ICCV.2011.6126278.