Deep K-SVD Denoising

Meyer Scetbon, Michael Elad, *Fellow, IEEE,* and Peyman Milanfar, *Fellow, IEEE*

Abstract—This work considers noise removal from images, focusing on the well known K-SVD denoising algorithm. This sparsity-based method was proposed in 2006, and for a short while it was considered as state-of-the-art. However, over the years it has been surpassed by other methods, including the recent deep-learning-based newcomers. The question we address in this paper is whether K-SVD was brought to its peak in its original conception, or whether it can be made competitive again. The approach we take in answering this question is to redesign the algorithm to operate in a supervised manner. More specifically, we propose an end-to-end deep architecture with the exact K-SVD computational path, and train it for optimized denoising. Our work shows how to overcome difficulties arising in turning the K-SVD scheme into a differentiable, and thus learnable, machine. With a small number of parameters to learn and while preserving the original K-SVD essence, the proposed architecture is shown to outperform the classical K-SVD algorithm substantially, and getting closer to recent state-of-the-art learning-based denoising methods. Adopting a broader context, this work touches on themes around the design of deep-learning solutions for image processing tasks, while paving a bridge between classic methods and novel deep-learning-based ones.

Index Terms—K-SVD Denoising algorithm, Network Unfolding, Iterative Shrinkage Algorithms.

1 Introduction

This paper addresses the classic image denoising problem: an ideal image $\mathbf{x}$ is measured in the presence of an additive zero-mean white and homogeneous Gaussian noise, $\mathbf{v}$, with standard deviation σ. The measured image $\mathbf{y}$ is thus $\mathbf{y} = \mathbf{x} + \mathbf{v}$, and our goal is the recovery of $\mathbf{x}$ from $\mathbf{y}$ with the knowledge of the parameter σ. This is quite a challenging task due to the need to preserve the fine details in $\mathbf{x}$ while rejecting as much noise as possible.

The importance of the image denoising problem cannot be overstated. First and foremost, noise corruption is inevitable in any image sensing process, often times heavily degrading the visual quality of the acquired image. Indeed, today's cell-phones all deploy a denoising algorithm of some sort in their camera pipelines [26]. Removing noise from an image is also an essential and popular pre-step in various image processing and computer vision tasks [14]. Last but not least, many image restoration problems can be addressed effectively by solving a series of denoising sub-problems, further broadening the applicability of image denoising algorithms [1], [28]. Due to its practical importance and the fact that it is the simplest inverse problem, image denoising has become the entry point for many new ideas brought over the years to the realm of image processing. Over a period of several decades, many image denoising algorithms have been proposed and tested, forming an evolution of methods with gradually improved performance.

A common and systematic approach for the design of novel denoising algorithms is the Bayesian point of view. This calls for image priors, used as regularizers within the Maximum a Posteriori (MAP) or the Minimum Mean Squared Error (MMSE) estimators. In this paper we concentrate on one specific regularization approach, as introduced in [10]: the use of sparse and redundant representation modeling of image patches – this is the K-SVD denoising algorithm, which stands at the center of this paper. The authors of [10] defined a global image prior that forces sparsity over patches in every location in the image. Their algorithm starts by breaking the image into small fully overlapping patches, solving their MAP estimate (i.e., finding their sparse representation), and ending with a tiling of the results back together by an averaging. As the MAP estimate relies on the availability of the dictionary, this work proposed two approaches, both harnessing the well known K-SVD dictionary learning algorithm [2]. The first option is to train off-line on an external large corpus of image patches, aiming for a universally good dictionary to serve all test images. The alternative, which was found to be more effective, suggests using the noisy patches themselves in order to learn the dictionary, this way adapting to the denoised image.

K-SVD has been widely used and extended, as evidenced by its many followup papers. For a short while, this algorithm was considered as state-of-the-art, standing at the top in denoising performance[1]. However, over the years it has been surpassed by other methods, such as BM3D [5], EPLL [39], WNNM [12], and many others. The recent newcomers to this game – supervised deep-learning based denoising methods – are currently at the lead [4], [15], [17], [37], [38].

Can K-SVD denoising make a comeback and compete favorably with the most recent and best performing denoising algorithms? In this paper we answer this question positively. We show that this algorithm can be brought to perform far better, provided that its parameters are tuned in a supervised manner. By following the exact K-SVD computational path, we preserve its global image prior. This includes (i) breaking the image into small fully overlapping patches, (ii) solving their MAP estimate as a pursuit that aims to get their sparse representation in a learned dictionary, and

M. Scetbon is with Ecole Normale Superieure - Paris-Saclay, `mscetbon@ens-paris-saclay.fr`*. M. Elad and P. Milanfar are with Google Research,* `[melad,milanfar]@google.com`*.*

[1] Ranking denoising algorithms is typically done by evaluating synthetic denoising performance on agreed-upon image databases (e.g. set12 or BSD68), measuring Peak-Signal-to-Noise (PSNR) and/or Structured Similarity Index Measure (SSIM) results.

then (ii) averaging the overlapping patches to restore the clean image. A special care is given to the redesign of all these steps into a differentiable and learnable computational scheme. We therefore end up with a deep architecture that reproduces the exact K-SVD operations, and can be trained by back-propagation for best denoising results. Our work shows that with small number of parameters to learn and while preserving the original K-SVD essence, the proposed machine outperforms the original K-SVD and other classical algorithms (e.g. BM3D and WNNM), and getting closer to state-of-the-art learning based denoising methods.

Our motivation in this paper goes beyond a simple improvement of the K-SVD denoising algorithm, aiming higher and broader. What are the lessons to be taken from the derived solution? How should we design novel and well-justified architectures for solving signal and image processing problems? What is the relation between classic (old fashioned) solutions and learning-based novel ones, in the context of such tasks? How can we further improve the proposed scheme in a principled way? All these and more are central questions, discussed towards the end of this paper. We urge the readers to go through the discussion towards the end of the paper carefully, as is gives the proper context to this work, and to its future prospects.

This paper is organized as followed. Section 2 recalls the K-SVD denoising algorithm, serving as the background for our derived alternative. In Section 3 we present the designed architecture with various modifications and adjustments that enable differentiabilty, local adaptivity, and more. Section 4 describes series of experiments that demonstrate the superiority of the proposed learned network over the classic K-SVD denoising algorithm, and show the tendency of our proposed network to have competitive performance with recent learned methods. We conclude this work in Section 5 with a wide discussion about this work and its contributions, and highlight potential future research directions.

2 The K-SVD Denoising Algorithm

In [10] the authors address the image denoising problem by using local sparsity and redundancy as ingredients in the formation of a global Bayesian objective. In this section we describe this K-SVD denoising algorithm by discussing (i) their global prior; (ii) the objective function induced; (iii) its corresponding numerical solver; and (iv) the two approaches for training the corresponding dictionary.

2.1 From the Patch- to a Global Objective Function

We start by introducing the local prior as imposed on patches in [10]. Let $\mathbf{x}$ be a small image patch of size $\sqrt{p} \times \sqrt{p}$ pixels, ordered lexicographically as a column vector of length p. The sparse representation model assumes that $\mathbf{x}$ is built as a linear combination of $s \ll p$ columns (also referred to as atoms) taken from a pre-specified dictionary[2] $\mathbf{D} \in \mathbb{R}^{p \times m}$. Put formally, $\mathbf{x} = \mathbf{D}\alpha$, where $\alpha \in \mathbb{R}^m$ is a sparse vector with s non-zeros (this is denoted by $\|\alpha\|_0 = s$). Consider $\mathbf{y}$, a noisy version of $\mathbf{x}$, contaminated by an additive zero-mean white Gaussian noise with standard deviation σ. The MAP estimator for denoising this patch is obtained by solving

$$\hat{\alpha} = \arg\min_{\alpha} \ \|\alpha\|_0 \quad \text{s.t.} \quad \|\mathbf{D}\alpha - \mathbf{y}\|_2^2 \leq p\sigma^2, \tag{1}$$

aiming to recover the sparse representation vector of $\mathbf{x}$. This is followed by $\hat{\mathbf{x}} = \mathbf{D}\hat{\alpha}$, obtaining the denoised result [3], [8], [34]. Note that the above optimization can be changed to a Lagrangian form,

$$\hat{\alpha} = \arg\min_{\alpha} \ \lambda\|\alpha\|_0 + \frac{1}{2}\|\mathbf{D}\alpha - \mathbf{y}\|_2^2, \tag{2}$$

such that the constraint becomes a penalty. With a proper choice of λ, which is signal (the vector $\mathbf{y}$) dependent, the two problems can become equivalent.

Moving now to handle a complete and large image $\mathbf{X}$ of size $\sqrt{N} \times \sqrt{N}$ and its noisy version $\mathbf{Y}$ (both held as vectors of length N), the global image prior proposed in [10] imposes the above-described local prior on every patch in $\mathbf{X}$, considering their extractions with full overlaps. This leads to the following global MAP estimator for the denoising:

$$\min_{\{\alpha_k\}_k, \mathbf{X}} \ \frac{\mu}{2}\|\mathbf{X} - \mathbf{Y}\|_2^2 + \sum_k \left(\lambda_k \|\alpha_k\|_0 + \frac{1}{2}\|\mathbf{D}\alpha_k - \mathbf{R}_k\mathbf{X}\|_2^2 \right).$$

In this expression, the first term is the log-likelihood global force that demands a proximity between the measured image, $\mathbf{Y}$, and its denoised (and unknown) version $\mathbf{X}$. Put as a constraint, this penalty would have read $\|\mathbf{X} - \mathbf{Y}\|_2^2 \leq N\sigma^2$, which reflects the direct relationship between μ and σ.

The second term stands for the image prior that assures that in the constructed image, $\mathbf{X}$, every patch[3] $\mathbf{x}_k = \mathbf{R}_k\mathbf{X}$ of size $\sqrt{p} \times \sqrt{p}$ in every location (thus, the summation by k) has a sparse representation with bounded error. The matrix $\mathbf{R}_k \in \mathbb{R}^{p \times N}$ stands for an operator that extracts the k-th block from the image. As to the coefficients λ_k, those must be spatially dependent, so as to comply with a set of constraints of the form $\|\mathbf{D}\alpha_k - \mathbf{x}_k\|_2^2 \leq p\sigma^2$.

2.2 Numerical Solution

Assume for the moment that the underlying dictionary $\mathbf{D}$ is known. The objective function in Equation (**??**) has two kinds of unknowns: the sparse representations α_k per each location, and the output image $\mathbf{X}$. Instead of addressing both together, the authors of [10] propose a block-coordinate minimization algorithm that starts with an initialization $\mathbf{X} = \mathbf{Y}$, and then seeks the optimal $\hat{\alpha}_k$ for all locations k. This leads to a decoupling of the minimization task to many smaller pursuit problems of the form

$$\hat{\alpha}_k = \arg\min_{\alpha_k} \ \lambda_k\|\alpha_k\|_0 + \frac{1}{2}\|\mathbf{D}\alpha_k - \mathbf{x}_k\|_2^2, \tag{3}$$

each handling a separate patch. This is solved in [10] using the Orthonormal Matching Pursuit (OMP) [9], which gathers one atom at a time to the solution, and stops when the error $\|\mathbf{D}\alpha_k - \mathbf{x}_k\|_2^2$ goes below[4] $p\sigma^2$. This way, the choice of λ_k has been handled implicitly. Thus, this stage works as a sliding

[2] The option $m > n$ implies that the dictionary is redundant.

[3] For simplicity and without loss of generality, a single index is used to account for the spatial image location.

[4] In fact, the threshold used in [10] is $c \cdot p\sigma^2$, with $c = 1.15$, which was found empirically to perform best.

window sparse coding stage, operated on each patch of size $\sqrt{p} \times \sqrt{p}$ pixels at a time.

Given all the sparse representations of the patches, $\{\hat{\alpha}_k\}_k$, we can now fix those and turn to update $\mathbf{X}$. Returning to the expression in Equation (**??**), we need to solve

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \ \frac{\mu}{2}\|\mathbf{X}-\mathbf{Y}\|_2^2 + \frac{1}{2}\sum_k \|\mathbf{D}\hat{\alpha}_k - \mathbf{R}_k\mathbf{X}\|_2^2. \quad (4)$$

This is a simple quadratic term that has a closed-form solution of the form

$$\hat{\mathbf{X}} = \left(\sum_k \mathbf{R}_k^T\mathbf{R}_k + \mu\mathbf{I}\right)^{-1}\left(\mu\mathbf{Y} + \sum_k \mathbf{R}_k^T\mathbf{D}\hat{\alpha}_k\right). \quad (5)$$

The matrix to invert in the above expression is a diagonal one, and thus the required computation is quite simple. In fact, all that this expression does is to put back the patches to their original locations, and average these with a weighted version of the noisy image itself.

All the above stands for a single update of $\{\alpha_k\}_k$ and then $\hat{\mathbf{X}}$. For an effective block-coordinate minimization of the cost function in Equation (**??**) we should repeat these pair of updates several times. However, a difficulty with such an approach is the fact that once $\hat{\mathbf{X}}$ has been modified, we no longer know the level of noise in each patch, and thus the stopping criteria for the OMP becomes more challenging. The original K-SVD denoising algorithm, as proposed in [10], chose to apply only the first round of updates. The work reported in [30] adopts an EPLL point of view [39], extending the iterative algorithm further for getting improved results.

2.3 Obtaining the Dictionary $\mathbf{D}$

The discussion so far has been based on the assumption that the dictionary $\mathbf{D}$ is known. This could be the case if we train it using the K-SVD algorithm over a corpus of clean image patches [9]. An interesting alternative is to embed the identification of $\mathbf{D}$ within the Bayesian formulation. Returning to the objective function in Eq. (**??**), the authors of [10] also considered the case where $\mathbf{D}$ is an unknown,

$$\min_{\{\alpha_k\}_k,\mathbf{X},\mathbf{D}} \ \frac{\mu}{2}\|\mathbf{X}-\mathbf{Y}\|_2^2 + \sum_k \left(\lambda_k\|\alpha_k\|_0 + \frac{1}{2}\|\mathbf{D}\alpha_k - \mathbf{R}_k\mathbf{X}\|_2^2\right).$$

In this case, $\mathbf{D}$ is learned using all the existing noisy patches taken from $\mathbf{Y}$ itself. Put more formally, a block-coordinate minimization is done: Initialize the dictionary $\mathbf{D}$ as the overcomplete DCT matrix and set $\mathbf{X} = \mathbf{Y}$. Then iterate between the OMP over all the patches and an update of $\mathbf{D}$ using the K-SVD strategy [2]. After $T = 10$ such rounds, the dictionary admits a content adapted to the image being treated, and the representations $\{\hat{\alpha}_k\}_k$ are ready for a final stage in which the output image is computed via Eq. (5).

3 Proposed Architecture

In this work our goal is to design a network that reproduces the K-SVD denoising algorithm, while having the capacity to learn its parameters. One of the main difficulties we encounter is the pursuit stage, in which we are supposed to replace the greedy OMP algorithm by an equivalent learnable alternative. This may seem as an easy task, as we can use the L_1-based Iterated Soft-Thresholding Algorithm (ISTA), unfolded appropriately for several iterations [6], [11]. However, the challenge is the fact that OMP easily adapts the treatment for each patch using a stopping criterion based on the noise level. The equivalence in the ISTA case requires an identification of the appropriate regularization parameter λ_k for each patch, which is a non-trivial task. Assuming that this issue has been resolved, our computational process includes a decomposition of the image into its overlapped patches, cleaning of each by an appropriate pursuit, and a reconstruction of the overall image by averaging the cleaned patches. We propose to learn the parameters of this network by training over pairs of corrupted and ground-truth images. Next, we describe in details this overall architecture.

3.1 Patch Denoising

Figure 1 illustrates our end-to-end architecture. We start by describing the three stages that perform the denoising of the individual patches.

Sparse Coding: Given a patch $\mathbf{y} \in \mathbb{R}^{\sqrt{p}\times\sqrt{p}}$ (held as a column vector of length p) corrupted by an additive zero-mean Gaussian noise with standard deviation σ, we aim to derive its sparse code according to a known dictionary $\mathbf{D} \in \mathbb{R}^{p\times m}$. This objective can be formulated as in Equation (1). An approximate solution to this problem can be obtained by replacing the ℓ_0-norm with an ℓ_1 [7], [8]:

$$\min_{\alpha\in\mathbb{R}^m} \|\alpha\|_1 \quad \text{s.t} \quad \|\mathbf{D}\alpha - \mathbf{y}\|_2^2 \leq p\sigma^2. \quad (6)$$

For a proper choice of λ, the above can be reformulated as

$$\hat{\alpha} = \arg\min_{\alpha} \frac{1}{2}\|\mathbf{D}\alpha - \mathbf{y}\|_2^2 + \lambda\|\alpha\|_1. \quad (7)$$

A popular and effective algorithm for solving the above problem is the Iterative Soft Thresholding Algorithm (ISTA) [6], which is guaranteed to converge to the global optimum

$$\hat{\alpha}_{t+1} = S_{\lambda/c}\left(\hat{\alpha}_t - \frac{1}{c}\mathbf{D}^T(\mathbf{D}\hat{\alpha}_t - \mathbf{y})\right) \quad ; \quad \hat{\alpha}_0 = 0, \quad (8)$$

where c is the square spectral norm of $\mathbf{D}$ and $S_{\lambda/c}$ is the component-wise soft-thresholding operator,

$$[S_\theta(\mathbf{v})]_i = \text{sign}(v_i)(|v_i| - \theta)_+. \quad (9)$$

The motivation to adopt a proximal gradient descent method, as done above, is the fact that it allows an unrolling of the sparse coding stage into a meaningful and learnable scheme, just as practiced in [11]. Indeed, replacing the ℓ_0-norm by the ℓ_1 supports this goal as it allows to differentiate through this scheme. Because of these reasons, in this work we consider a learnable version of ISTA by keeping exactly the same recursion with a fixed number of iterations T, and letting c and $\mathbf{D}$ become the learnable parameters.

λ Evaluation: Referring to the pursuit formulation in Equation (7), an important issue is the need to set the parameter λ. This regularization coefficient depends not only on σ but also on the patch $\mathbf{y}$ itself. Following the computational path of the K-SVD denoising algorithm in [10], we should set λ_k for each patch $\mathbf{y}_k$ so as to yield sparse representation with a controlled level of error, $\|\mathbf{D}\hat{\alpha}_k - \mathbf{y}_k\|_2^2 \leq p\sigma^2$. As there is no closed-form solution to this evaluation of λ-s, we propose to learn a regression function from the patches $\mathbf{y}_k$ to their corresponding regularization parameters λ_k. A Multi-Layer

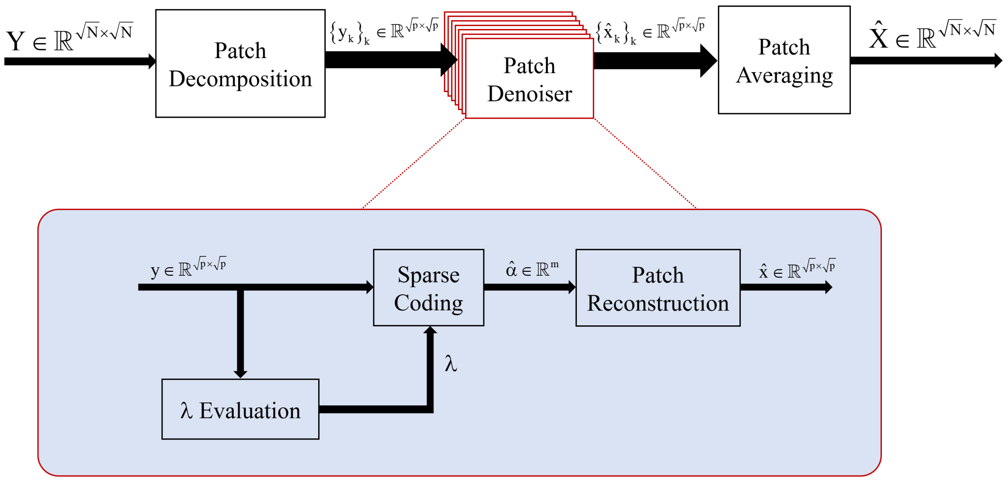
$Y \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$
Patch Decomposition
$\{y_k\}_k \in \mathbb{R}^{\sqrt{p}\times\sqrt{p}}$
Patch Denoiser
$\{\hat{x}_k\}_k \in \mathbb{R}^{\sqrt{p}\times\sqrt{p}}$
Patch Averaging
$\hat{X} \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$
$y \in \mathbb{R}^{\sqrt{p}\times\sqrt{p}}$
Sparse Coding
$\hat{\alpha} \in \mathbb{R}^m$
Patch Reconstruction
$\hat{x} \in \mathbb{R}^{\sqrt{p}\times\sqrt{p}}$
λ
λ Evaluation

Figure 1: Architecture of the proposed method.

Perceptron (MLP) network is used to represent this function, $\lambda = f_\theta(\mathbf{y})$, where θ is the vector of the parameters of the MLP. Our MLP consists of three hidden layers, each composed of a fully connected linear mapping followed by a ReLU (apart from the last layer). The input layer has p nodes, which is the dimension of the vectorized patch, and the output layer consists of a single node, being the regularization parameter. The overall structure of the network is given by the following expression, in which $[a \times b]$ symbolizes a multiplication by a matrix of that size: MLP: $\mathbf{y} \rightarrow [p \times 2p] \rightarrow$ ReLU $\rightarrow [2p \times p] \rightarrow$ ReLU $\rightarrow [p/2 \times 1] \rightarrow \lambda$. Thus, an overall of nearly $4p^2$ parameters are needed for this regression network.

Patch Reconstruction: This stage reconstructs the cleaned version $\hat{\mathbf{x}}$ of the patch $\mathbf{y}$ using $\mathbf{D}$ and the sparse code $\hat{\alpha}$. This is given by $\hat{\mathbf{x}} = \mathbf{D}\hat{\alpha}$. Note that in our learned network, the dictionary stands for a set of parameters that are shared in all locations where we multiply by either $\mathbf{D}$ or $\mathbf{D}^T$.

3.2 End-to-end Architecture

We can now discuss the complete architecture. We start by breaking the input image into fully overlapping patches, then treat each corrupted patch via the above-described patch denoising stage, and conclude by rebuilding the image by averaging the cleaned version of these patches. In the last stage we slightly deviate from the original K-SVD, by allowing a learned weighted combination of the patches. Denoting by $\mathbf{w} \in \mathbb{R}^{\sqrt{p}\times\sqrt{p}}$ this patch of weights, the reconstructed image is obtained by

$$\hat{\mathbf{X}} = \frac{\sum_k \mathbf{R}_k^T(\mathbf{w} \odot \hat{\mathbf{x}}_k)}{\sum_k \mathbf{R}_k^T \mathbf{w}}, \tag{10}$$

where $\odot$ is the Schur product, and the division is done element-wise. This weighted averaging aligns with Guleryuz' approach as advocated in [13].

To conclude, the proposed network F is a parametrized function of θ (the parameters of the MLP network computing λ), c (the step-size in the ISTA algorithm), $\mathbf{D}$ (the dictionary) and $\mathbf{w}$ (the weights for the patch-averaging). The overall number of parameters stands on $p(4p + m + 3/2) + 1$; for example, for $p = 64$ and $m = 256$, this number is $32,865$.

Given a corrupted image $\mathbf{Y}$, the computation $\hat{\mathbf{X}} = F(\mathbf{Y})$ returns a cleaned version of it. Training F is done by minimizing the loss function $\mathcal{L} = \sum_i \|\mathbf{X}_i - F(\mathbf{Y}_i)||_2^2$, with respect to all the above parameters. In the above objective, the set $\{\mathbf{X}_i\}_i$ stands for our training images, and $\{\mathbf{Y}_i\}_i$ are their synthetically noisy versions, obtained by $\mathbf{Y}_i = \mathbf{X}_i + \mathbf{V}_i$, where $\mathbf{V}_i$ is a zero mean and white Gaussian iid noise vector.

3.3 Extension to Multiple Update

As already mentioned in the previous section, an EPLL version of the K-SVD can be envisioned, in which the process of cleaning the patches is repeated several times. This implies that once the above architecture obtains its output $\hat{\mathbf{X}}$, the whole scheme could be applied again (and again). This diffusion process of repeated denoisings has been shown in [30] to improve the K-SVD denoising performance. However, the difficulty is in setting the noise level to target in each patch after the first denoising, as it is no longer $p\sigma^2$. In our case, we adopt a crude version of the EPLL scheme, in which we disregard the noise level problem altogether, and simply assume that the λ evaluation stage takes care of this challenge, adjusting the MLP in each round to best predict the λ values to be used. Thus, our iterated scheme shares the dictionary across all denoising stages, while allowing a different λ evaluation network for each stage.

4 Experimental Results

We turn to present experiments with the proposed Learned K-SVD (LKSVD). Our goals are to show that LKSVD is

- Much better than the original KSVD in its two forms – the image adaptive algorithm (KSVD$_1$), and the one using a universal dictionary (KSVD$_2$);
- Better than other classic denoising algorithms; and
- Competitive with recent deep-learning based denoisers.

4.1 Training

Dataset: In order to train our model we generate the training data using the Berkeley segmentation dataset (BSDS) [22], which consists of 500 images. We split these images into a training set of 432 images and the validation/test set that consists of the remaining 68 images. We note that these 68 images are exactly the ones used in the standard evaluation dataset of [29]. In addition, following [17], [37], we test our proposed method on the benchmark Set12 – a collection of widely-used testing images. The training and the two test sets are strictly disjoint and all the images are converted to gray-scale in each experiment setup. This allows a fair and comprehensive comparison with recent deep learning based methods, as we train and test on the same datasets and benchmarks used in [4], [15], [16], [17], [21], [37].

Training Settings: During training we randomly sample cropped images of size 128×128 from the training set. We add i.i.d. Gaussian noise with zero mean and a specified level of noise σ to each cropped image as the noisy input during training. We train a different model for each noise level, considering $\sigma = 15, 25, 50$.

We use SGD optimizer to minimize the loss function. We set the learning rate as $1e-4$ and consider one cropped image as the minibatch size during training. We use the same initialization as in the K-SVD algorithm to initialize the dictionary $\mathbf{D}$, i.e the overcomplete DCT matrix. We also initialize the normalization paramater c of the sparse coding stage using the squared spectral norm of the DCT matrix. The other parameters of the network are randomly initialized using Kaiming Uniform method. Training a model takes about 2 days with a Titan Xp GPU.

Test Settings: Our network does not depend on the input size of the image. Thus, in order to test our architecture's performance, we simply add white Gaussian noise with a specified power to the original image, and feed it to the learned scheme. The metric used to determine the quality is the standard Peak-Signal-to-Noise (PSNR).

4.2 Results

In Tables 1, 2 and 3 we compare[5] LKSVD with the two original K-SVD versions (KSVD$_1$ and KSVD$_2$) and two leading classic denoising algorithms, BM3D [5] and WNNM [12]. Tables 1 and 2 refer to the BSD68 test-set (one showing PSNR and the other SSIM quality measures) and Table 3 shows the Set12 results (PSNR only). In this comparison, LKSVD is set to use the same patch and dictionary sizes as in KSVD$_1$ and KSVD$_2$ from [10], namely $p = 64$ and $m = 256$. Also, LKSVD applies $T = 7$ unfolded iterations of ISTA, and $K = 3$ EPLL-like denoising rounds.

Dataset	Noise	BM3D	WNNM	KSVD$_1$	KSVD$_2$	LKSVD
BSD 68	15	31.07	31.37	30.91	30.87	**31.48**
	25	28.57	28.83	28.32	28.28	**28.96**
	50	25.62	25.87	25.03	25.01	**25.97**

Table 1: LKSVD vs. classic methods (BSD68): Denoising performance (PSNR [dB]) for various noise levels.

Dataset	Noise	BM3D	WNNM	KSVD$_1$	KSVD$_2$	LKSVD
BSD 68	15	0.8717	0.8766	0.8692	0.8685	**0.8835**
	25	0.8013	0.8087	0.7876	0.7894	**0.8171**
	50	0.6864	0.6982	0.6322	0.6462	**0.7035**

Table 2: LKSVD versus classic methods (BSD68): Denoising performance (SSIM) for various noise levels.

A clear conclusion from the above tables is the fact that LKSVD is much better performing compared to the classic K-SVD, be it the universal dictionary approach or the image adaptive one. Indeed, the PSNR BSD68 results suggest that LKSVD is better than BM3D (by ~ 0.5dB) and WNNM (by ~ 0.1dB) as well. Table 3 displays a slightly different story, where LKSVD is still better performing compared to BM3D, while being slightly weaker than the WNNM. Recall that BM3D and WNNM both leverage non-local self-similarity, which gives them an edge over K-SVD. In addition, these two methods have been tuned for best results for this test set (see in particular their exceptional results for Lena and Barbara). As a final note we add that Table 2 shows that the ordering of the methods remains the same as we move from PSNR to the SSIM quality measure, which explains our choice to use PSNR for the rest of the experiments.

We proceed by exploring the effect of p (patch size), m (dictionary size) and K (number of denoising steps) on the LKSVD performance. We denote by LKSVD$_{K,p,m}$ the result for the proposed architecture with these specified parameters. Table 4 presents the obtained results for the two benchmarks (BSD68 and Set12) and a noise level of $\sigma = 25$. As can be seen, even with $[p, m, K] = [64, 256, 1]$, LKSVD is markedly better than the classic K-SVD. As K grows, the performance improves by ~ 0.1dB per each additional denoising round. A boost in performance is also obtained when growing the patch-size to 16×16 while preserving the redundancy factor of the dictionary. This also shows that the proposed scheme has the capacity to yield results that go beyond the ones reported in Tables 1 and 3.

We conclude by comparing the LKSVD$_{2,16,1024}$ with recent learning-based denoising competitors: TNRD [4], NLNet [15], DnCNN [37] and NLRNet [17]. The results are shown in Table 5, referring to the two benchmarks. As can be seen, our scheme surpasses TNRD [4] and even the non-local deeply-learned denoiser by Lefkimmiatis [15], [16]. Still, there is a gap between LKSVD and the best performing denoisers DnCNN [37] and NLRNet [17]. Table 6 sheds more light on these results by presenting the model complexities involved in this experiment. As can be seen, our network (LKSVD$_{3,8,256}$) uses about 10% of the overall number of parameters compared to the better performing methods.

We conclude by presenting visual results of the various methods compared. Figure 2 shows the denoising results of BM3D, WNNM, KSVD$_1$, DnCNN, and LKSVD. The figure refers to a noise level of $\sigma = 25$ and the images used are taken from the BSD68 test set.

5 Discussion and Conclusions

5.1 Why bother improving K-SVD denoising?

The rationale behind this work goes beyond a simple improvement of the K-SVD denoising algorithm. Indeed, our motivation is drawn from the hope to propose systematic ways of designing deep-learning architectures and connecting novel solutions to classical algorithms.

[5] The results in these tables corresponding to BM3D and WNNM have been taken from [17] and [38], respectively.

Images	C.man	House	Peppers	Starfish	Monarch	Airplane	Parrot	Lena	Barbara	Boat	Man	Couple	Average
Noise level						$\sigma = 15$							
BM3D	31.91	34.93	32.69	31.14	31.85	31.07	31.37	34.26	33.10	32.13	31.92	32.10	32.37
WNNM	32.17	35.13	32.99	31.82	32.71	31.39	31.62	34.27	33.60	32.27	32.11	32.17	32.70
KSVD$_1$	31.43	32.21	34.23	30.80	31.59	30.99	31.64	31.45	30.95	31.83	32.44	33.78	31.95
KSVD$_2$	31.39	32.16	33.85	30.96	31.66	30.96	31.62	31.71	30.99	31.63	30.58	33.49	31.75
DKSVD	32.16	32.92	34.59	31.54	32.11	31.66	32.22	32.78	31.78	32.18	32.22	34.24	32.53
Noise level						$\sigma = 25$							
BM3D	29.45	32.85	30.16	28.56	29.25	28.42	28.93	32.07	30.71	29.90	29.61	29.71	29.97
WNNM	29.64	33.22	30.42	29.03	29.84	28.69	29.15	32.24	31.24	30.03	29.76	29.82	30.26
KSVD$_1$	28.75	29.64	31.86	28.21	29.10	28.42	29.26	28.83	28.27	29.44	29.77	31.37	29.41
KSVD$_2$	28.78	29.74	31.46	28.39	29.04	28.57	29.16	28.85	28.24	29.18	27.61	31.04	29.17
DKSVD	29.70	30.35	32.53	28.92	29.71	29.13	29.85	30.15	28.99	30.07	30.06	31.99	30.12
Noise level						$\sigma = 50$							
BM3D	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.22	26.78	26.81	26.46	26.72
WNNM	26.45	30.33	26.95	25.44	26.32	25.42	26.14	29.25	27.79	26.97	26.94	26.64	27.05
KSVD$_1$	25.12	25.93	27.82	24.86	25.56	24.80	26.16	25.11	24.45	25.98	25.78	27.71	25.78
KSVD$_2$	25.29	26.02	27.71	24.85	25.44	25.15	25.98	24.82	24.32	25.93	24.04	27.32	25.58
DKSVD	26.67	26.97	29.37	25.61	26.55	26.00	26.95	26.54	25.38	26.98	27.03	28.86	26.91

Table 3: LKSVD versus classic methods (Set12): Denoising performance (PSNR [dB]) for various noise levels. The two best results are marked in red and blue.

Dataset	Noise	KSVD$_1$	KSVD$_2$	LKSVD$_{1,8,256}$	LKSVD$_{3,8,256}$	LKSVD$_{1,16,1024}$	LKSVD$_{2,16,1024}$
BSD 68	25	28.32	28.28	28.76	28.96	28.95	29.07
Set 12		29.41	29.17	29.76	30.12	30.09	30.22

Table 4: LKSVD parameter effect: Denoising performance (PSNR [dB]) for $\sigma = 25$ on BSD68 and Set12 while varying p, m, K.

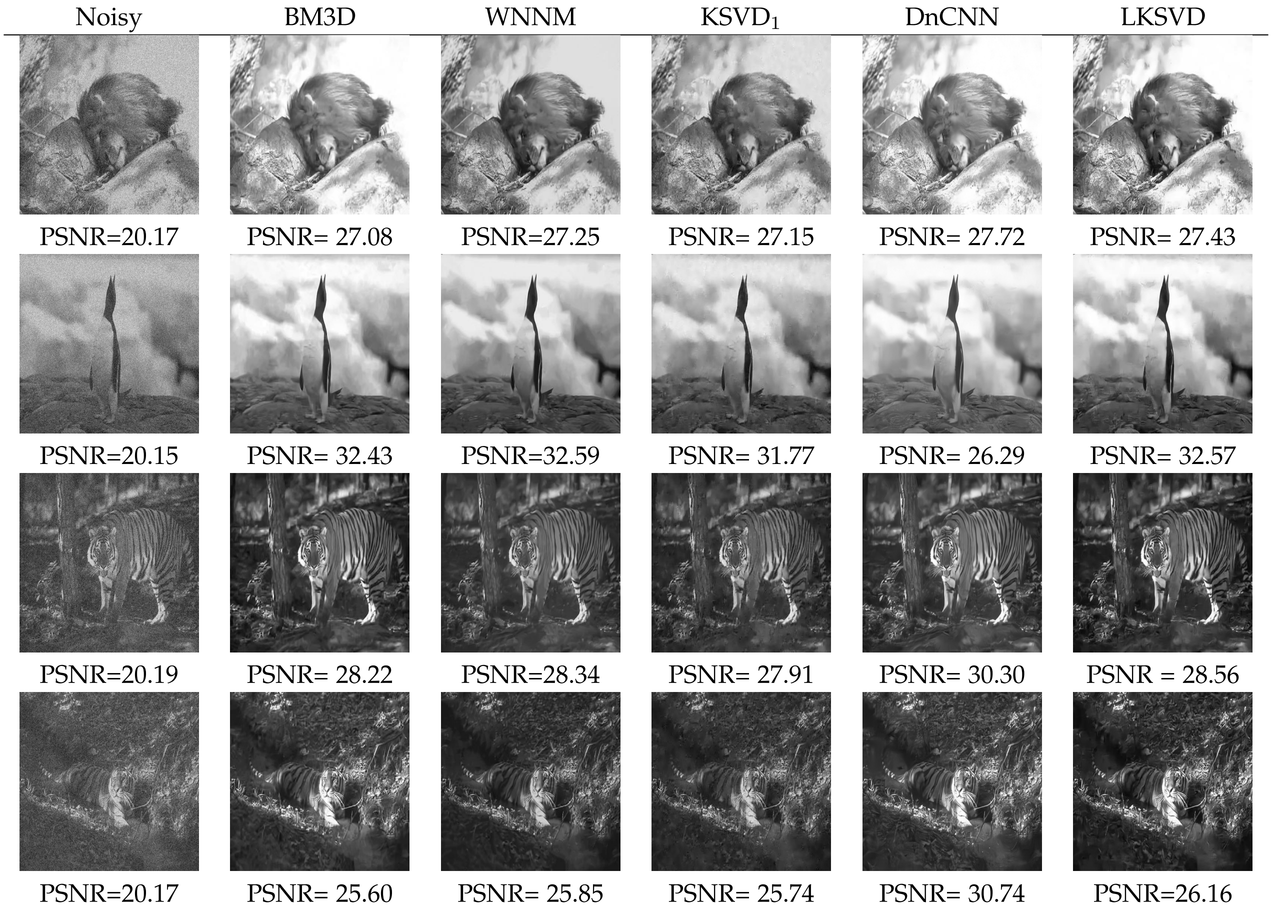

Figure 2: Denoising results for noise level $\sigma = 25$.

A fundamental question nowadays in computational imaging is whether old/classic methods should be discarded and replaced by their deep-learning alternatives. In the context of image denoising, classical methods focused on data modeling and optimization, and searched for ways to identify and exploit the redundancies existing in the visual data. The recent deep networks, which lead the denoising performance charts today, take an entirely different route, targeting the inference stage directly, and learning their parameters for optimized end-to-end performance. Now that these methods are getting close to touch their ceiling, our work comes to argue that the classic methods are still very much relevant, and could become key in breaking such barriers. We believe that classic image processing algorithms will have a comeback for this exact reason.

Adopting a different point of view, this work offers a

Dataset	Noise	TNRD	NLNet	DnCNN	NLRNet	LKSVD$_{2,16,1024}$
BSD 68	15	31.42	31.52	**31.73**	**31.88**	31.54
	25	28.92	29.03	**29.23**	**29.41**	29.07
	50	25.97	26.07	**26.23**	**26.47**	26.13
Set 12	15	32.50	-	**32.86**	**33.16**	32.61
	25	30.06	-	**30.44**	**30.80**	30.22
	50	26.81	-	**27.18**	**27.64**	27.04

Table 5: LKSVD versus learned methods: Denoising performance (PSNR [dB]) for various noise levels on BSD68 and Set12. Results exceeding LKSVD are marked in bold.

	DnCNN	NLRNet	LKSVD
Max effective depth	17	38	21
Parameter sharing	No	Yes	Yes
Parameter no.	554k	330k	45k

Table 6: Model complexities comparison of our proposed model with state-of-the-art network models.

migration from intuitively chosen architectures, as many recent papers have offered, towards well-justified ones based on domain knowledge of the problem we are trying to solve. That is, the structure of the denoising problem is embedded into the deep learning architecture, making the overall algorithm enjoy both the flexibility of the deep methods, and the structure brought by the more classical approaches. The option of piling convolutions, ReLU's, batch-normalization steps, skip connections, strides and pooling operations, dilated filtering, and many other tricks, and seeking for best performing architectures by trial an error, has been the dominating approach so far. It is time to return to the theoretical foundations of signal and image processing in order to go beyond this point. Relying on sparse representation modeling, the K-SVD network we introduce in this work has a clear objective, a concise structure, and yet it works quite well. In fact, we believe that the results shown here stand as yet another testimony for the central role that sparse modeling plays in broad data processing.

And related to the above, here is an interesting question: What is the simplest possible network, in terms of the number of free parameters to learn and the number of computations to apply, for getting state-of-the-art image denoising? In single-image super-resolution it has become common practice in the literature to compare different solutions by considering their complexity as well (e.g., [33]). This is done by showing points in a 2D graph of PSNR versus computational cost. Doing the same in image denoising may reveal interesting patterns. The general deep-learning based methods, while showing the best PSNR, tend to be quite heavy and cumbersome. Could much lighter networks perform nearly as well (and perhaps even better)? In this work we offer one such avenue to explore, and we are certain that many others will follow.

5.2 Going Beyond Sparsity?

Why has it been so easy to outperform the original K-SVD denoising algorithm in the first place? A possible answer could be that this algorithm builds its cleaning abilities on two prime forces: (i) the spatial redundancy that exists in image patches, exposed by the sparse modeling; and (ii) the patch-averaging effect, which has an MMSE flavor to it [24]. Many of the better performing competitors strengthen their performance by considering several additional ideas:

- **Non-Locality:** Non-local self-similarity can be practiced as an additional prior, as done by BM3D [5] and low-rank modeling [12], [36]. Indeed, the paper by Mairal et. al [18] extended the K-SVD denoising by incorporating joint sparsity on groups of patches, this way introducing non-locality. Broadly speaking, non-local methods are known to be effective in capturing the correlation between far-apart patches, leading to improved restoration.
- **Patch Consensus:** Patch based methods must address the disagreement found between overlapping patches. The original K-SVD scheme we embark from in this paper proposed an averaging[6] of these patches. However, the EPLL approach [39] suggests a far better strategy, by imposing the prior on patches taken from the resulting image, rather than ones extracted from the measured one. In the context of sparse modeling, this idea boils down to an iterated K-SVD algorithm, as was shown in [30]. In such a scheme the cleaned image is aggregated and broken to patches again for subsequent pursuit. We have deployed this very idea in an elementary way by replicating the filtering process. Closely related alternatives to this strategy are the SOS boosting method [27] and the deployment of the CSC model [25].
- **Multi-Scale:** Multi-scale analysis of visual data seems to be a natural strategy to follow, and various papers have shown the benefit of this for image denoising [24]. More specifically, a multi-scale extension of the K-SVD denoising algorithm has been considered in various practical ways [19], [20], [23], [31].

The above suggests that K-SVD denoising in its original form carries a built-in weakness in it. Yet, the results in this paper suggest otherwise. Consider the more recent and better performing deep-learning based solutions. These alternatives seem to disregard these extra forces (at least explicitly), concentrating instead on capturing image intrinsic properties by a direct supervised learning of the inference process. Recent such convolutional neural networks (CNNs) for image restoration [21], [37] achieve impressive performance over classical approaches. Do these methods exploit self-similarity? anything reminiscent of patch-consensus? a multi-scale architecture? One may argue that the answer is, at most, only partially positive, hidden by the wide receptive field and the global treatment that these networks entertain. Note that there are deep learning methods that explicitly use self-similarity in their processing [15], [17], however those do not necessarily improve over the simpler alternatives.

The conclusion we draw from the above is that there is room for introducing non-locality, patch-consensus and a multi-scale structure into the proposed K-SVD scheme, thereby driving the revised architecture towards even better results. Indeed, nothing is sacred in the K-SVD computational path, and the same treatment as done in this work could be given to well-performing classical denoising algorithms, such as BM3D [5], kernel-based methods [32] and WNNM [12]. We leave these ideas for future work.

[6] While the original K-SVD denoising algorithm has used a plain averaging, we deploy a slightly improved weighted option, due to its simplicity in the context of a learned machine.

5.3 Could we suggest an unsupervised version of this architecture?

This is perhaps a good time to recall that the denoising work in [10] offered two strategies for getting the dictionary – a globally universal approach that trains the dictionary off-line, and an image-adaptive alternative that trains on the noisy image patches themselves. Interestingly, despite the fact that the later (image-adaptive) approach was found to be better performing, the solution we put forward in this paper aligns solely with the first approach. Why? because the supervised strategy we adopt naturally leads to a single architecture that serves all images via the same set of parameters. Could we offer an unsupervised alternative, more in line with the image adaptive path? The answer, while tricky, could be positive. A related approach of great relevance is [35], in which a chosen network architecture is trained on each image all over again. A similar concept could be envisioned, where our own K-SVD architecture is used for synthesizing the clean image. However, this raises some difficulties and challenges, which is why we leave this activity for future work.

5.4 Conclusions

This work shows that the good old K-SVD denoising algorithm [10] can have a comeback and become much better performing, getting closer to leading deep-learning based denoisers. This is achieved very simply by setting its parameters in a supervised fashion, while preserving its exact original form. Our work have shown how to turn the K-SVD denoiser into a learnable architecture that enables back-propagation, and demonstrated the achieved boost in denoising performance. As the discussion above reveals, our story goes beyond the K-SVD denoising and its improvement, towards more fundamental questions related to the role of deep-learning in contemporary image processing.

References

[1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Trans. on Image Processing*, 19(9):2345–2356, 2010.

[2] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54(11):4311, 2006.

[3] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[4] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. on Pattern Anal. and Machine Intel.*, 39(6):1256–1272, 2016.

[5] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. In *2007 15th European Signal Processing Conference*, pages 145–149. IEEE, 2007.

[6] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

[7] D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[8] D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. on Information Theory*, 52(1):6–18, 2005.

[9] M. Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer, 2010.

[10] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. on Image Processing*, 15(12):3736–3745, 2006.

[11] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *ICML*, pages 399–406. Omnipress, 2010.

[12] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, pages 2862–2869, 2014.

[13] O. O. Guleryuz. Weighted averaging for denoising with overcomplete dictionaries. *IEEE Trans. on Image Processing*, 16(12):3020–3034, 2007.

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

[15] S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *CVPR*, pages 3587–3596, 2017.

[16] S. Lefkimmiatis. Universal denoising networks: a novel cnn architecture for image denoising. In *CVPR*, pages 3204–3213, 2018.

[17] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems*, pages 1673–1682, 2018.

[18] J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, volume 29, pages 54–62. Citeseer, 2009.

[19] J. Mairal, G. Sapiro, and M. Elad. Multiscale sparse image representationwith learned dictionaries. In *ICIP*, volume 3, pages III–105. IEEE, 2007.

[20] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *Multiscale Modeling & Simulation*, 7(1):214–241, 2008.

[21] X. Mao, C. Shen, and Y.-B. 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.

[22] 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 *ICCV*, volume 2, pages 416–423, July 2001.

[23] B. Ophir, M. Lustig, and M. Elad. Multi-scale dictionary learning using wavelets. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):1014–1024, 2011.

[24] V. Papyan and M. Elad. Multi-scale patch-based image restoration. *IEEE Trans. on Image Processing*, 25(1):249–261, 2015.

[25] V. Papyan, Y. Romano, and M. Elad. Convolutional neural networks analyzed via convolutional sparse coding. *The Journal of Machine Learning Research*, 18(1):2887–2938, 2017.

[26] T. Plotz and S. Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, pages 1586–1595, 2017.

[27] Y. Romano and M. Elad. Boosting of image denoising algorithms. *SIAM Journal on Imaging Sciences*, 8(2):1187–1219, 2015.

[28] Y. Romano, M. Elad, and P. Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

[29] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205, 2009.

[30] J. Sulam and M. Elad. Expected patch log likelihood with a sparse prior. In *Int. Workshop on Energy Minimization Methods*, pages 99–111. Springer, 2015.

[31] J. Sulam, B. Ophir, and M. Elad. Image denoising through multi-scale learnt dictionaries. In *ICIP*, pages 808–812. IEEE, 2014.

[32] H. Takeda, S. Farsiu, P. Milanfar, et al. *Kernel regression for image processing and reconstruction*. PhD thesis, Citeseer, 2006.

[33] R. Timofte, R. Rothe, and L. Van Gool. Seven ways to improve example-based single image super resolution. In *CVPR*, June 2016.

[34] J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Trans. on Information Theory*, 52(3):1030–1051, 2006.

[35] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *CVPR*, pages 9446–9454, 2018.

[36] N. Yair and T. Michaeli. Multi-scale weighted nuclear norm image restoration. In *CVPR*, June 2018.

[37] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. on Image Processing*, 26(7):3142–3155, 2017.

[38] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Trans. on Image Processing*, 27(9):4608–4622, 2018.

[39] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486. IEEE, 2011.