Image Denoising via Sequential Ensemble Learning

Xuhui Yang, Yong Xu, Yuhui Quan*, Hui Ji

***Abstract*—Image denoising is about removing measurement noise from input image for better signal-to-noise ratio. In recent years, there has been great progress on the development of data-driven approaches for image denoising, which introduce various techniques and paradigms from machine learning in the design of image denoisers. This paper aims at investigating the application of ensemble learning in image denoising, which combines a set of simple base denoisers to form a more effective image denoiser. Based on different types of image priors, two types of base denoisers in the form of transform-shrinkage are proposed for constructing the ensemble. Then, with an effective re-sampling scheme, several ensemble-learning-based image denoisers are constructed using different sequential combinations of multiple proposed base denoisers. The experiments showed that sequential ensemble learning can effectively boost the performance of image denoising.**

***Index Terms*—Image denoising, Ensemble learning, Image recovery, Ensemble denoiser**

I. Introduction

IMAGE denoising is about removing noise from image. Despite the great advance in sensing instruments and technologies, the signal-to-noise ratio of image remains unsatisfactory in many scenarios. For example, taking pictures using low-cost cameras at high sensitivities with low light conditions or high ISO settings. The problem of image denoising is still of great practical value to many low-level vision tasks, and its importance keeps growing with the prevalence of webcams and mobile phones. In addition, image denoising plays an important role in many image recovery tasks, as it serves as one fundamental module in many image recovery methods; see *e.g.* [1], [2].

A noisy image, denoted by $\boldsymbol{g}$, is usually modeled by

$$\boldsymbol{g} = \boldsymbol{f} + \boldsymbol{n},$$

where $\boldsymbol{f}$ denotes the noise-free image, and $\boldsymbol{n}$ denotes the measurement noise. To estimate $\boldsymbol{f}$ from its noisy observation $\boldsymbol{g}$, it is necessary to impose certain priors on both the image and the noise. In most cases, the noise is modeled as the realization of a random vector following some probability distribution. For instance, many existing methods assume measurement noise of each pixel is independent and identically distributed (*i.i.d.*) with zero mean. Then, the focus of designing an effective denoiser is about how to define an accurate prior of noise-free images so as to separate the image and the noise.

In the past, various image priors have been proposed for denoising, *e.g.*, the sparsity-based prior (or hyper-Laplacian prior) of local variations of image intensities (*e.g.* [3], [4], [5], [6]), and the recurrence of local image patches (*e.g.* [7], [8], [9], [10], [11]). The advantages of such approaches lie in their simplicity and fair generality, but there is a great room for improvement considering the great variation of image content in practice. In recent years, the data-driven (or learning-based) approach has become more appealing as it allows the prior to be adaptive to image content. The learning-based image denoisers showed noticeable improvement over the image denoisers that use manually-crafted image priors.

In the past, many machine learning techniques have been introduced in the design of image denoisers, including

- Sparsity-based dictionary learning for denoising [12], [13], [14], [15];
- Probabilistic generative models of pixel values or transform coefficients of images, such as Gaussian scale mixture model on wavelet coefficients [16], [17], Markov random field in transform domain [18], and mixture of Gaussians on image patch groups [19], [20], [21];
- Neural network based denoisers, such as convolutional neural network (CNN) [22], [23], auto-encoder [24], [25], [23], and multi-layer perception (MLP) [26];
- Trainable iterative denoising methods with strong motivations from variational models [27], [28], [29].

All these learning-based denoisers have their merits and shortcomings.

Ensemble learning is an effective paradigm in machine learning that combines multiple simple learning models to produce a more accurate solution than a single model does. Motivated by the success of ensemble learning in many applications, this paper aims at exploiting the potential of ensemble learning in image denoising. In this paper, we proposed an ensemble framework for removing noise from images, which learns a set of transform-shrinkage-based simple denoisers, and sequentially combines them to achieve good performance on image denoising. The experiments showed performance gain of the proposed method over several well-established denoising methods. The study presented in this paper clearly indicates the potential of sequential ensemble learning in image denoising. In other words, we have an alternative promising learning approach for boosting the performance of image denoising.

Xuhui Yang, Yong Xu and Yuhui Quan are with School of Computer Science and Engineering at South China University of Technology, China. Hui Ji is with Department of Mathematics, National University of Singapore. Yong Xu is also with Peng Cheng Laboratory, Shenzhen, China. Yuhui Quan is also with Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, China. Email: csyoe@mail.scut.edu.cn (Xuhui Yang), yxu@scut.edu.cn (Yong Xu), csyhquan@scut.edu.cn (Yuhui Quan), matjh@nus.edu.sg (Hui Ji).

This work is supported by National Natural Science Foundation of China (61872151, 61672241, 61602184, U1611461), Natural Science Foundation of Guangdong Province (2017A030313376, 2016A030308013), Science and Technology Program of Guangdong Province (2019A050510010, 20140904-160), Science and Technology Program of Guangzhou (201802010055), Fundamental Research Funds for Central Universities of China (x2js-D2181690), and Singapore MOE AcRF (R146000229114, MOE2017-T2-2-156).

Asterisk indicates the corresponding author.

A. Motivation and rationale

An image, expressed as an array, can be viewed as a point in a high dimensional linear space. It is widely accepted that the set of noise-free natural images are concentrated on a nonlinear low-dimensional manifold in such a high dimensional linear space [30], [31]. The image prior imposed on noise-free images, either manually-crafted or data-driven, is about the characterization of the geometry of such an underlying nonlinear manifold. Therefore, the denoiser derived from an image prior can be viewed as the projection of an noisy image onto a low-dimensional manifold characterized by the image prior used in the denoiser. Considering the great variation of image content in practice, the geometry of such a low-dimensional manifold certainly is very complex. Thus, it is very natural to consider using some machine learning techniques to characterize the geometry of such a manifold.

Recall that the same phenomenon also happens in classification, where a weak classifier cannot accurately model decision boundaries that have complex geometrical structure. For example, a linear classifier can not effectively model the decision boundary with large curvature. One approach of addressing such an issue is the so-called *ensemble learning* [32]. Ensemble learning provides a simple yet effective way to increase the modeling capability of an classifier, which is done by combining multiple weak classifiers to form a strong one that can effectively represent the decision boundary that has complex geometrical shape.

Motivated by the effectiveness of ensemble learning on representing the decision boundary of an classifier with complex geometrical shape, we propose an ensemble-learning-based framework for more accurately modeling the geometry of the low-dimensional manifold of noise-free images, which is done by combining multiple simple denoisers to form a powerful image denoiser. Indeed, many existing iterative image denoising methods (*e.g.* [30], [27]) can be interpreted as sequential denoising ensemble, in which each iteration of the method is a simple denoiser. Take the iterative wavelet thresholding method for example. An iterative wavelet thresholding method is done by iteratively thresholding the wavelet coefficients of an image, which can be interpreted as a sequential concatenation of many simple denoisers (wavelet thresholding). See Fig. 1 for the denoising performance of such an iterative wavelet shrinkage method, where the ensemble size is equivalent to the number of iterations. It can be seen that the performance gain in terms of PSNR supports the argument that ensemble can help boosting the performance of denoising.

However, existing iterative denoising methods are not optimized for fully exploiting the potential of ensemble learning. In existing ones, the simple denoiser in each iteration is more or less the same, which is against the diversity paradigm of ensemble learning that makes the ensemble powerful, *i.e.*, the variations among simple classifiers (denoisers) to make them complementary to each other for boosting the performance. In addition, the resampling technique which is a powerful tool in ensemble learning, cannot be applied to these methods. Thus, there is certainly the need to design a class of image denoisers that are specifically optimized for utilizing ensemble learning.

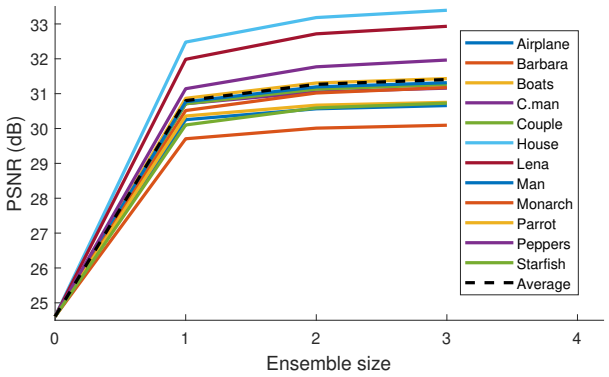


Fig. 1: Performance of the ensemble of wavelet thresholding operator versus the ensemble size on 12 natural images shown in Fig. 8, in terms of PSNR value. The noise level is $\sigma = 15$. It can be seen that the the PSNR value of the denoising results increases as the ensemble size becomes larger.

B. Contributions and significance

In recent years, the concepts and techniques in machine learning, supervised or un-supervised, have been one main driving force in the development of image denoisers. As one important technique in machine learning, ensemble learning is a simple yet efficient approach for performance boost in many learning-based applications. This paper is the first one that introduces ensemble learning to solve the problem of image denoising. The proposed framework of constructing ensemble-learning-based image denoiser, together with its concrete implementation, not only shows the potential of ensemble learning in image denoising, but also provides an effective solution with solid performance.

Compared to other learning-based image denoising approaches, *e.g.* the probability generative models [16], [17] or the variational-model-motivated trainable iterative methods [27], [29], the proposed ensemble-learning-based approach allows the integration of different types of optimization models corresponding to different types of image priors, while those approaches are derived from certain variational models with specific image priors. The flexibility and the simplicity of integrating different types of image priors certainly make the proposed ensemble-learning-based approach more appealing. Compared to the neural-network-based approaches (*e.g.* [22], [23], [24], [25], [23], [26]), our ensemble-learning-based framework allows much easier incorporation of image priors derived from specific domain knowledge. Such a property can be very attractive in the case that there is no sufficient amount of training image data. It also allows the integration of some well-established powerful denoisers, *e.g.* the well-known BM3D method, which can not be efficiently implemented in the framework of neural network. Furthermore, the proposed ensemble denoising framework has better interpretability than neural networks, since each base denoiser is interpretable.

In addition, based on simple transform-shrinkage-based denoisers, a practical denoiser ensemble is implemented in this

paper. Different from [27], [29], the simple transform-based denoiser is based on the shrinkage function represented by the linear combination of Gabor functions. Two types of base denoisers are proposed for constructing ensemble denoisers, which exploit different characteristics of images. An effective resampling scheme is developed for performance boost in image denoising. Comprehensive experiments are conducted to evaluate the performance of the proposed approach, and the results showed that the proposed ensemble denoiser achieved solid performance on several widely-used test datasets.

C. Notations and organization

Through this paper, bold upper letters are used for matrices (*e.g.* $\boldsymbol{A}, \boldsymbol{W}$), bold lower letters for column vectors (*e.g.* $\boldsymbol{a}, \boldsymbol{w}$), light lower letters for scalars (*e.g.* a, w), and hollow letters for sets (*e.g.* $\mathbb{R}, \mathbb{Z}$). Given a sequence $\{\boldsymbol{y}^{(i)}\}_{t\in\mathbb{Z}}$, $\boldsymbol{y}^{(i)}$ denotes the i-th element in the sequence. For a vector $\boldsymbol{x} \in \mathbb{R}^N$, let $\boldsymbol{x}[i]$ denote the i-th element in $\boldsymbol{x}$, and define $\|\boldsymbol{x}\|_2 = \sqrt{\sum \boldsymbol{x}(i)^2}$.

The rest of this paper is organized as follows. Section II gives a brief review on the related work. Section III is devoted to the discussion of the proposed ensemble denoiser, and Section IV is on the experimental evaluation. Lastly, the paper is concluded in Section V.

II. Related Work

In the past, many denoising methods have been proposed with different motivations. They can roughly be classified into knowledge-driven (*i.e.* image-prior-based) approaches and data-driven (*i.e.* learning-based) approaches. In this section, we only give a brief review on the image-prior-based approaches and focus more on the learning-based approaches.

Using the image prior that the gradients of a noise-free image are smooth, early work employed spatial smoothing filtering or diffusion for suppressing noise; see *e.g.* [7] for more details. In recent years, the sparsity of noise-free images in certain transform domain and the patch recurrence within an image are two dominant priors used in image recovery. The sparsity prior assumes that images can be sparsified under certain transforms, *i.e.*, most transform coefficients are zero or close to zero. Representative transforms for sparsifying natural images include discrete cosine transform (DCT) [33] and wavelet transform [34]. In addition to these transforms, the data-driven transforms for sparsifying images could yield better results, *e.g.* dictionary learning using K-SVD [12], orthogonal dictionary learning [35], data-driven tight frame [36], and multi-resolution dictionary learning [37], to name a few. For the patch recurrence prior, the BM3D method [38] is arguably the most prevalent one with state-of-the-art performance. The idea of BM3D is to first group similar image patches into different stacks and then apply collaborative filtering on these stacks. There are many variants of such an approach, *e.g.* local PCA [39], low rank approximation [40], [41], Bayesian inference [42], smooth patch ordering on patch graph [43], patch-graph Laplacian regularization [44], patch clustering [45], and multi-scale patch-recurrence image denoising [10]. The sparsity prior and patch recurrence prior are combined together in [30], [46], [47], [48], [49] for better performance.

Meanwhile, many approaches also have been proposed to learn image priors from a set of training images. In [16], [17], natural images are characterized by their multi-level wavelet transform coefficients which are modeled by Gaussian scale mixtures. The FoE method [18] learns a high-order Markov random field for modeling natural images. In [19], image patches are modeled by the mixture of Guassians, and this method is extended to the multi-scale setting in [20]. By grouping similar image patches together, the prior on such groups is learned in [21].

Instead of learning image priors for separating noise-free image and noise, an alternative is to directly learn the mapping between noisy images and noise-free images. There has been rapid progress along this line in the context of deep learning. Many different architectures of neural networks have been proposed to learn such a map. Convolutional neural networks (CNNs) are implemented in [22], [23] and auto-encoders are implemented in [24], [25] for image denoising. In [26], a multi-layer perceptron (MLP) is trained for denoising image patches. Based on Gaussian conditional random field, a deep neural network is proposed in [50] for improving the applicability of neural network to different noise levels. In [23], a deep CNN with residual learning and batch normalization is proposed for blind denoising.

An alternative approach of learning the mapping that maps noisy images to noise-free ones is to convert the iterative solver of some existing regularization methods into a data-driven denoiser, which is done by making the parameters involved in the iterative solver trainable. In [18], [27], [28], the Markov Random Field (MRF) based image denoisers are expanded to construct a random field-based architecture that combines the image model and the optimization algorithm in a single unit. In [29], the PDE diffusion process is converted to a data-driven diffusion denoiser by making those pre-defined parameters trainable.

The proposed method in this paper also learns the mapping that maps the noisy images to their noise-free counterparts. Different from the aforementioned learning-based methods, our approach is based on ensemble learning. The proposed ensemble-learning-based image denoisers have better interpretability than those neural-network-based methods. Also, they have more flexibility of utilizing image priors than those iterative-solver-based methods.

III. Ensemble Denoiser

A. Ensemble learning framework for denoising

Let $\boldsymbol{f} \in \mathbb{R}^N$ denote a noise-free image, and $\boldsymbol{g} \in \mathbb{R}^N$ denote its noisy observation generated by the following process:

$$\boldsymbol{g} = \boldsymbol{f} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{n} \in \mathcal{N}(0, \sigma^2\boldsymbol{I})$ denotes additive white noise. Given the noisy image $\boldsymbol{g}$, image denoising is about recovering $\boldsymbol{f}$ from $\boldsymbol{g}$. In this section, we first introduce a sequential ensemble learning based framework for image denoising, and then propose an implementation of such a framework using transform-thresholding-based denoisers. In the following, such

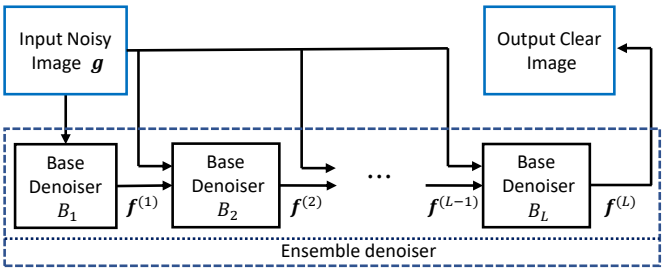


Fig. 2: Framework of proposed ensemble denoiser.

an ensemble denoiser, denoted by $E : \mathbb{R}^N \to \mathbb{R}^N$, is built from L base denoisers, denoted by $\{B_\ell\}_{\ell=1}^L$.

The proposed architecture of ensemble learning for denoising is illustrated in Fig. 2. The architecture is similar to a cascade classifier. In ensemble learning, the ensemble classifier is constructed by the concatenation of several base classifiers. Therefore, in the proposed framework of ensemble denoiser, the base denoisers $\{B_\ell\}_{\ell=1}^L$ are also sequentially concatenated, and they cooperate in a cascade form such that $E(\boldsymbol{f}) = B_L(B_{L-1}(\cdots(B_1(\boldsymbol{f}))))$. Recall that in a cascade classifier, each base classifier accepts the results of the previous classifier as well as the supervised information as input during training. Analogously, each base denoiser $B_\ell : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ takes two inputs: the output $\boldsymbol{f}^{(\ell-1)}$ from the previous base denoiser $B_{\ell-1}$ and the noisy measurement $\boldsymbol{g}$, and outputs an estimate $\boldsymbol{f}^{(\ell)}$ of $\boldsymbol{f}$:

$$B_\ell : (\boldsymbol{f}^{(\ell-1)}, \boldsymbol{g}) \longrightarrow \boldsymbol{f}^{(\ell)}.$$

The first base denoiser B_1, $\boldsymbol{f}^{(0)}$ is simply set as $\boldsymbol{g}$. In the proposed ensemble denoiser, the training of each base denoiser is based on the denoised results of the previous base denoiser and the ground-truth noise-free images.

In such a framework, each base denoiser B_ℓ aims at improving the denoised result generated from the previous estimator $B_{\ell-1}$ by simultaneously suppressing the residual error (remaining noise) and recovering the information of $\boldsymbol{f}$ lost in $\boldsymbol{f}^{(\ell-1)}$. The B_ℓ is learned in the way that it is optimized for minimizing the approximation error of the estimation $\boldsymbol{f}^{(\ell)}$ to the truth $\boldsymbol{f}$. Indeed, the reason why inputing the original noisy measurement $\boldsymbol{g}$ to the denoiser B_ℓ is to provide the source such that the denoiser B_ℓ is able to recover the information of $\boldsymbol{f}$ lost in the previous estimate $\boldsymbol{f}^{(\ell-1)}$.

B. Base denoisers

Similar to many ensemble classifiers in which each base classifier is a weak one such as the linear classifier, the base denoisers in the proposed ensemble based denoiser are also the ones that are of simple forms and have efficient implementation. There are many such candidates. In this paper, we consider the well-established shrinkage (attenuation) based denoising technique. The basic procedure is as follows. Given a noisy input $\boldsymbol{y} \in \mathbb{R}^N$, it is converted by the transform $\boldsymbol{W}$ into another representation $\boldsymbol{c} = \boldsymbol{W}\boldsymbol{y}$, followed by being applied with a shrinkage operator Γ. Then, the denoised result is obtained by converting $\Gamma(\boldsymbol{c})$ back to an image using some operator $\boldsymbol{W}^+$. Such a process can be formulated as:

$$\boldsymbol{y}^* = \boldsymbol{W}^+\Gamma(\boldsymbol{W}\boldsymbol{y}; \boldsymbol{\beta}), \tag{2}$$

where $\Gamma(\cdot; \boldsymbol{\beta})$ denotes the shrinkage operator with the parameter vector $\boldsymbol{\beta}$.

Indeed, many well-established denoising methods fall into the category of shrinkage-based denoising methods. For example, the Wiener estimator uses the discrete Fourier transform (DFT) as $\boldsymbol{W}$ and its inverse as $\boldsymbol{W}^+$, and uses the attenuation function as the shrinkage operator whose parameters are determined by the signal-to-noise ratio. The wavelet shrinkage methods use the discrete wavelet transform as $\boldsymbol{W}$ and its inverse as $\boldsymbol{W}^+$. The shrinkage operator used in wavelet shrinkage methods is either hard thresholding operator or soft thresholding operator. Recall that a hard thresholding operator sets small entries to zero, and a soft thresholding operator not only sets small entries to zero but also attenuates large entries.

In the proposed scheme, we consider an element-wise shrinkage operator:

$$\Gamma(\boldsymbol{y}; \boldsymbol{\beta})[i] = h(\boldsymbol{y}[i]; \boldsymbol{\beta}),$$

where the parameter vector $\boldsymbol{\beta} = [\boldsymbol{\beta}^1; \boldsymbol{\beta}^2]$ with

$$\begin{aligned} \boldsymbol{\beta}^1 &= \left[\beta^1_{1,1}, \ldots, \beta^1_{1,Q}, \beta^1_{2,1}, \ldots, \beta^1_{P,Q}\right]^\top, \\ \boldsymbol{\beta}^2 &= \left[\beta^2_{1,1}, \ldots, \beta^2_{1,Q}, \beta^2_{2,1}, \ldots, \beta^2_{P,Q}\right]^\top. \end{aligned} \tag{3}$$

In our approach, the attenuation function $h(\cdot, \boldsymbol{\beta})$ is represented by the linear combination of a set of Gabor atoms:

$$h(z; \boldsymbol{\beta}) = \sum_{p=1}^{P}\sum_{q=1}^{Q} \beta^1_{p,q} d^1_{p,q}(z) + \sum_{p=1}^{P}\sum_{q=1}^{Q} \beta^2_{p,q} d^2_{p,q}(z), \tag{4}$$

where

$$\begin{aligned} d^1_{p,q}(z) &= \exp\left(\frac{(z-\mu_p)^2}{2\rho^2}\right)\cos(\omega_q x), \\ d^2_{p,q}(z) &= \exp\left(\frac{(z-\mu_p)^2}{2\rho^2}\right)\sin(\omega_q x). \end{aligned} \tag{5}$$

In our implementation, the grid points $\{\mu_p\}_p$ are equi-spaced in the interval $[-310, 310]$ with step size 10. Recall that the range of image pixel value is $[0, 255]$, and thus the range of the output with respect to a normalized high-pass filter is $[-255, 255]$. As the intermediate results in the ensemble denoiser might exceed the range $[-255, 255]$, a larger range $[-310, 310]$ is then used such that most entries fall into such an interval in practice. As the distance between two adjacent grid points is 10, the standard deviation ρ of Gaussian function used in (5) is then also set to 10, so that the resulting atoms have sufficient polynomial reproducing capacity while they are not too redundant. The parameters in $\boldsymbol{\beta}$ are randomly initialized and learned from training data.

Note that in many existing works (*e.g.* [27], [51]), Gaussian functions are used for representing the shrinkage operator. In contrast, our approach uses Gabor functions for the representation of the shrinkage operator. The main motivation comes from the powerful capability of Gabor functions on local time-frequency analysis that allows more fine-grained operations on

image details. To verify the benefit of using Gabor functions over Gaussian function, two base denoisers are learned on 68 natural images from [27] with the same configuration, except that one uses Gabor functions ($P = 63, Q = 2$) and the other uses Gaussian functions ($P = 63, Q = 0$)[1]. The results using these two denoisers on the classic natural images given in Fig. 8 show that the one using Gabor functions has the average 0.06dB PSNR improvement over the one using Gaussian functions. Such an improvement on a base denoiser will be further magnified in the resulting ensemble denoiser.

Recall that a base denoiser takes two inputs: the original noisy measurement $\boldsymbol{f}$ and an estimate of noise-free image $\boldsymbol{f}^{(\ell-1)}$ from the base denoiser in the previous stage. Based on how the previous estimation is used in the current denoising, there are two possible types of base denoisers that can be used:

- **Shrinkage after merging**. The denoised image $\boldsymbol{f}^{(\ell-1)}$ from the previous base denoiser is first merged with the noisy image $\boldsymbol{g}$ to take back the details, and then the shrinkage is applied to removing the residual noise. In this case, the base denoiser is formulated by

$$B_\ell(\boldsymbol{f}^{(\ell-1)}, \boldsymbol{g}) = \boldsymbol{W}^+\Gamma(\boldsymbol{W}((1-\lambda)\boldsymbol{f}^{(\ell-1)} + \lambda\boldsymbol{g}); \boldsymbol{\beta}). \tag{6}$$

- **Shrinkage before merging**. Shrinkage is first applied to the denoised image $\boldsymbol{f}^{(\ell-1)}$ from the previous base denoiser for removing residual noise, and then the denoised result is merged with the noisy image $\boldsymbol{g}$ for fetching the details. In this case, the base denoiser is formulated by

$$B_\ell(\boldsymbol{f}^{(\ell-1)}, \boldsymbol{g}) = (1-\lambda)\boldsymbol{W}^+\Gamma(\boldsymbol{W}\boldsymbol{f}^{(\ell-1)}; \boldsymbol{\beta}) + \lambda\boldsymbol{g}, \tag{7}$$

  or equivalently

$$B_\ell(\boldsymbol{f}^{(\ell-1)}, \boldsymbol{g}) = \boldsymbol{W}^+\Gamma(\boldsymbol{W}\boldsymbol{f}^{(\ell-1)}; \boldsymbol{\beta}) + \lambda(\boldsymbol{g} - \boldsymbol{f}^{\ell-1}).$$

It is empirically observed that shrinkage after merging yields better performance. Thus, we use shrinkage after merging in our implementation. In Section IV, some results using shrinkage before merging are given for comparison.

C. Local base denoiser

The base denoiser discussed above relies on the design of the transform $\boldsymbol{W}$. Motivated by the success of wavelet transforms and convolution neural networks in image recovery, we also use the filter-bank-based computational scheme for constructing $\boldsymbol{W}$. Consider a filter bank $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_K\}$ which can be either pre-defined or learned. The corresponding transform, denoted by $\boldsymbol{W} : \mathbb{R}^N \to \mathbb{R}^{NK}$, is defined as follows.

$$\boldsymbol{W} : \boldsymbol{y} \longrightarrow [\boldsymbol{a}_1 \otimes \boldsymbol{y}; \boldsymbol{a}_2 \otimes \boldsymbol{y}; \ldots; \boldsymbol{a}_K \otimes \boldsymbol{y}], \tag{8}$$

where $\otimes$ denotes the discrete convolution operator. The reconstruction operator, denoted by $\boldsymbol{W}^+ : \mathbb{R}^{NK} \to \mathbb{R}^N$, is the adjoint operator of $\boldsymbol{W}$ of the form:

$$\boldsymbol{W}^+ : [\boldsymbol{c}_1; \boldsymbol{c}_2; \ldots; \boldsymbol{c}_K] \longrightarrow \sum_{k=1}^{K} \widetilde{\boldsymbol{a}}_k \otimes \boldsymbol{c}_k, \tag{9}$$

[1]It is noted that Gaussian functions with $P = 53$ and with $P = 63$ are used in [27], [29] respectively

where $\{\widetilde{\boldsymbol{a}}_1, \widetilde{\boldsymbol{a}}_2, \ldots, \widetilde{\boldsymbol{a}}_K\}$ denotes another filter bank for reconstruction. In our implementation, $\widetilde{\boldsymbol{a}}_k$ is the reverse order of $\boldsymbol{a}_k$, denoted by $\boldsymbol{a}_k(-\cdot)$. Such a transform is closely related to un-decimal single-level wavelet transform [36].

Plugging (8) and (9) into (2) leads to the following base denoiser:

$$\boldsymbol{y}^*_{\text{loc}} = \sum_{k=1}^{K} \boldsymbol{a}_k(-\cdot) \otimes \Gamma(\boldsymbol{a}_k \otimes \boldsymbol{y}; \boldsymbol{\beta}_k), \tag{10}$$

where the same shrinkage operator Γ but with different parameters is used for different filters. It is noted that the pair of transforms $(\boldsymbol{W}, \boldsymbol{W}^+)$ defined by (8) and (9) are based on the convolutions using the filters of small support. Thus, the transform $\boldsymbol{W}$ only measures local variations of the input over different locations. Thus, we call the base denoiser defined by (10) as the *local base denoiser*.

When a local base denoiser is learned from data, the first filter in both filter banks is a pre-defined low-pass filter $\boldsymbol{a}_1^{(\ell)} = \frac{1}{\#\boldsymbol{a}^{(\ell)}}\mathbf{1}$, where $\#\boldsymbol{a}^{(\ell)}$ is the length of $\boldsymbol{a}_1^{(\ell)}$. Moreover, each filter in $\{\boldsymbol{a}_k\}_{k=2}^K$ is expressed as

$$\boldsymbol{a}_k = \boldsymbol{D}\boldsymbol{\gamma}_k, k = 2, \cdots, K,$$

where the columns of $\boldsymbol{D}$ are the filters in DCT except the low-pass one, and $\boldsymbol{\gamma}_k$ is the coefficient vector to be learned. Note that the learned filters can be normalized by rescaling.

D. Nonlocal base denoiser

The local base denoiser exploits local variations of its input for denoising, *e.g.*, the wavelet shrinkage method takes advantage of the sparsity prior of local tensity variations of image. To utilize another powerful image prior, nonlocal patch recurrence prior of images, we define another type of base denoisers called *nonlocal base denoiser*. Given an image $\boldsymbol{y} \in \mathbb{R}^N$, let $\boldsymbol{V}_{\boldsymbol{y}} \in \mathbb{R}^{N\times N}$ denote the nonlocal similarity matrix of $\boldsymbol{y}$, whose element $\boldsymbol{V}_{\boldsymbol{y}}(i,j)$ is defined by

$$\boldsymbol{V}_{\boldsymbol{y}}(i,j) = \exp(\frac{\|\boldsymbol{R}_i\boldsymbol{y} - \boldsymbol{R}_j\boldsymbol{y}\|_2^2}{Z^2\sigma^2}), \tag{11}$$

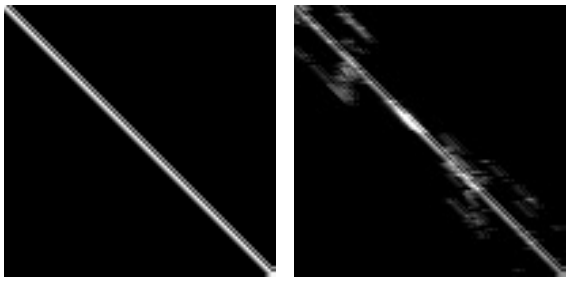
where $\boldsymbol{R}_i : \mathbb{R}^N \to \mathbb{R}^{Z^2}$ denotes the operator that extracts the i-th patch (*i.e.* the patch centered at the i-th location) of size $Z \times Z$ from $\boldsymbol{y}$, and σ is the noise level of image $\boldsymbol{y}$. The transforms $\boldsymbol{W} \in \mathbb{R}^{NK\times N}, \boldsymbol{W}^+ \in \mathbb{R}^{N\times NK}$ in our nonlocal base denoiser are defined by

$$\boldsymbol{W} : \boldsymbol{y} \longrightarrow [(\boldsymbol{a}_1 \otimes \boldsymbol{y}); \ldots; (\boldsymbol{a}_K \otimes \boldsymbol{y})], \tag{12}$$

$$\boldsymbol{W}^+ : [\boldsymbol{c}_1; \ldots; \boldsymbol{c}_k] \longrightarrow \sum_{k=1}^{K} \widetilde{\boldsymbol{a}}_k \otimes (\boldsymbol{V}_{\boldsymbol{y}}\boldsymbol{c}_k), \tag{13}$$

where $\{\boldsymbol{a}_k, \widetilde{\boldsymbol{a}}_k\}_{k=1}^K$ is the pair of filter banks defined in Section III-C. Due to the existence of $\boldsymbol{V}_{\boldsymbol{y}}$, the pair of transforms, $\boldsymbol{W}$ and $\boldsymbol{W}^+$, relates the pixels far away so that the mapping is no longer local. See Fig. 3 for an illustration.

The construction of an optimal $\boldsymbol{V}_{\boldsymbol{y}}$ is computationally expensive. Instead, we adopt a simple patch matching scheme that is often used in the nonlocal denoising methods (*e.g.* [38], [21]). For each patch $\boldsymbol{R}_i\boldsymbol{y}$, we find its top-$T$ ($T = 8$ in practice) similar patches within an $R \times R$ neighborhood and



(a) Local $\boldsymbol{W}$ (b) Nonlocal $\boldsymbol{W}$

Fig. 3: Examples of local / nonlocal version of $\boldsymbol{W}$ on image 'Lena'. For constructing $\boldsymbol{W}$, we set $T = 8$ and $R = 33$. For clarity we only show a 100×100 submatrix of $\boldsymbol{W}$.

then calculate the pair-wise distances between $\boldsymbol{R}_i\boldsymbol{y}$ and each similar patch. The distance between two unmatched patches is set to infinity such that the similarity coefficient is 0. With a GPU-acceleration implementation, such a patch matching scheme only takes around 2.55 seconds for a 256×256 image using a PC with i7-6700K CPU and Nvidia Titan V GPU. It is noted that the resulting matrix $\boldsymbol{V}_{\boldsymbol{y}}$ is sparse which is efficient in both memory usage and related numerical computations.

By substituting (12) and (9) into (2), we have another base denoiser:

$$\boldsymbol{y}^*_{\text{nloc}} = \sum_{k=1}^{K} \boldsymbol{a}_k(-\cdot) \otimes (\boldsymbol{V}_{\boldsymbol{y}}\Gamma((\boldsymbol{a}_k \otimes \boldsymbol{y}); \boldsymbol{\beta}_k)). \tag{14}$$

In comparison to local base denoiser, the nonlocal one encodes the nonlocal patch recurrence similarity structure of image with the additional nonlocal operation $\boldsymbol{V}_{\boldsymbol{y}}$. It is noted that $\boldsymbol{V}_{\boldsymbol{y}}$ is not a shift-invariant operator, and it varies for different images. For computational efficiency, instead of running patch matching for intermediate results, all nonlocal base denoisers use the same $\boldsymbol{V}_{\boldsymbol{y}}$ to construct nonlocal $\boldsymbol{W}$ in one pass.

E. Training

The proposed ensemble denoiser $E(\cdot;\Theta)$ is defined on the set of base denoisers, $B_\ell(\cdot,\cdot;\boldsymbol{\Theta}^{(\ell)})$, $\ell = 1,..,L$, where $\boldsymbol{\Theta}^{(\ell)}$ contains the parameters of B_ℓ. The full set of parameters $\boldsymbol{\Theta} = \bigcup_{\ell=1}^{L} \boldsymbol{\Theta}^{(\ell)}$. Define $\boldsymbol{A}^{(\ell)} = (\boldsymbol{\gamma}_2^{(\ell)}, \cdots, \boldsymbol{\gamma}_K^{(\ell)})$. Then, regardless the type of base denoiser, the parameters to be learned in each base denoiser $B^{(\ell)}$ are

$$\boldsymbol{\Theta}^{(\ell)} = \{\boldsymbol{A}^{(\ell)}, \boldsymbol{\beta}^{(\ell)}, \lambda^{(\ell)}\}.$$

To efficiently train ensemble denoiser, we use a sequential scheme which is widely used in training cascade ensemble classifiers [52]. That is, sequentially training each base classifier from the beginning to the end.

Given a set of training image pairs $\{\boldsymbol{f}_j, \boldsymbol{g}_j\}_{j=1}^{J}$ where $\boldsymbol{f}_j$ is a noise-free image, and $\boldsymbol{g}_j$ is its noisy measurement. Let $\boldsymbol{f}_j^{(0)} = \boldsymbol{g}_j$ for all possible j and $\boldsymbol{f}_j^{(\ell)}$ be the output of the ℓ-th base denoiser B_ℓ that takes $\boldsymbol{g}_j$ as input. The loss function for learning the ℓ-th base denoiser is then defined as

$$\min_{\boldsymbol{\Theta}^{(\ell)}} F_\ell(\boldsymbol{\Theta}^{(\ell)}) := \frac{1}{2}\sum_{j=1}^{J} \|\boldsymbol{f}_j - B_\ell(\boldsymbol{f}_j^{(\ell-1)}, \boldsymbol{g}_j; \boldsymbol{\Theta}^{(\ell)})\|_2^2. \tag{15}$$

An alternative gradient descend scheme [53] is called to solve (15). The iteration is done as follows. Given $\boldsymbol{\Theta}_i^{(\ell)}$ at the i-th iteration, update the estimation, denoted by $\boldsymbol{\Theta}_{i+1}^{(\ell)}$, by

$$\begin{cases} \lambda_{i+1}^{(\ell)} = \lambda_i^{(\ell)} - s_i^{\lambda}\nabla_{\lambda^{(\ell)}} F_\ell(\lambda_i^{(\ell)}, \boldsymbol{\beta}_i^{(\ell)}, \boldsymbol{A}_i^{(\ell)}), \\ \boldsymbol{\beta}_{i+1}^{(\ell)} = \boldsymbol{\beta}_i^{(\ell)} - s_i^{\beta}\nabla_{\boldsymbol{\beta}^{(\ell)}} F_\ell(\lambda_i^{(\ell)}, \beta_i^{(\ell)}, \boldsymbol{A}_i^{(\ell)}), \\ \boldsymbol{A}_{i+1}^{(\ell)} = \boldsymbol{A}_i^{(\ell)} - s_i^{\boldsymbol{A}}\nabla_{\boldsymbol{A}^{(\ell)}} F_\ell(\lambda_i^{(\ell)}, \boldsymbol{\beta}_i^{(\ell)}, \boldsymbol{A}_i^{(\ell)}) \end{cases} \tag{16}$$

for $i = 0, 1, \cdots$, where $s_i^{\lambda}, s_i^{\boldsymbol{\beta}}, s_i^{\boldsymbol{A}} > 0$ are the step sizes. The iteration stops when there is no noticeable gain in PSNR.

F. Resampling

To further boost the performance, we propose to introduce the resampling technique in ensemble learning for training. In the training of the base denoiser $B^{(\ell)}$ where $\ell > 1$, we randomly discard a small percentage of the input images $\{\boldsymbol{f}_j^{(\ell-1)}\}_j$ which are the outputs of the previous base denoiser $B^{(\ell-1)}$. Then, new noisy images are created by re-adding noise to the ground truths of those discarded images. These new noisy images are denoised by passing them to the previous trained base denoisers, and the previous denoisers are not retrained. Then the new denoised images from $B^{(\ell-1)}$ are added to the set of images for training of $B^{(\ell)}$. See Fig. 4 for an illustration of our resampling scheme.

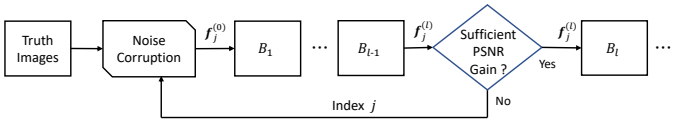


Fig. 4: Illustration of proposed resampling scheme.

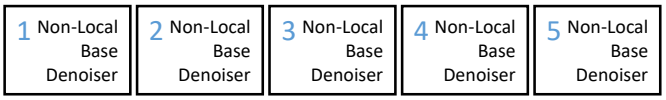
G. Ensemble structure

By stacking multiple local base denoisers or multiple nonlocal base denoisers, we can get two different ensemble denoisers shown in Fig. 5 (a)-(b). In practice, the nonlocal ensemble denoiser performs better than the local one, since it can exploit additional image prior (*i.e.* patch recurrence). However, it is also much more computationally expensive as it requires a patch matching process for constructing nonlocal matrix. In other words, these two ensemble denoisers have their own merits and suitable applications.

With these two types of base denoisers in hand, we have the flexibility of using different combinations of local and nonlocal base denoisers to construct an ensemble denoiser that fits the need of target application in terms of denoising effectiveness and computational efficiency. In Fig. 5 (c)-(d), we illustrate two different structures of ensemble denoiser: *(i)* a half-and-half structure in which the first half is concatenated using local base denoisers and the second half is concatenated using nonlocal ones, and *(ii)* an alternative structure in which local base denoisers and nonlocal ones are alternatively concatenated. Furthermore, an adaptive construction of ensemble denoising can be employed as follows. At each stage of the training, we train a local base denoiser and a nonlocal base denoiser separately, and select the one with higher PSNR value on the

training data to be the base denoiser of the current stage. The drawback of such an adaptive construction is its very high computational cost. We will investigate such an adaptive construction in future.

(a) fully local structure



(b) fully nonlocal structure

(c) half-and-half structure

(d) alternative structure

Fig. 5: Candidates of ensemble structure.

The overall computational cost of an ensemble denoiser is determined by those of local base denoisers and nonlocal base denoisers. For local base denoisers, the complexity of $\boldsymbol{W}$ defined in (8) is $\mathcal{O}(KM^2N)$, as it only involves K convolutions with filter size $M \times M$. The same is true for $\boldsymbol{W}^+$ defined in (9). The complexity of the shrinkage operator Γ is $\mathcal{O}(PQKN)$. Then, the total computational complexity of a local base denoiser is $\mathcal{O}((M^2+PQ)KN)$. For nonlocal base denoisers, in addition to the two processes above, there are two more operations: the patch matching for constructing $\boldsymbol{V}$ and the matrix multiplication with $\boldsymbol{V}$. The complexity of patch matching is $\mathcal{O}(Z^2R^2N)$ where $Z \times Z$ is the patch size, and the complexity of nonlocal matrix multiplication is $\mathcal{O}(TKN)$ since $\boldsymbol{V}$ is a sparse matrix. Then, the total computational complexity of a nonlocal base denoiser is $\mathcal{O}((M^2K+PQK+Z^2R^2+TK)N)$. Suppose an ensemble denoiser contains L_1 local base denoisers and L_2 nonlocal base denoisers. Then, the overall computational complexity of the ensemble denoiser is

$$\mathcal{O}\big((L_1+L_2)(M^2+PQ)KN + L_2N(Z^2R^2+TK)\big).$$

IV. Experiments

A. Datasets and settings

To evaluate the performance of the proposed method, we use the same training/test datasets as [29]. The training dataset contains 400 cropped images of size 180×180 from the Berkeley segmentation dataset [54], and the test dataset (called BSD68) contains 68 natural images of various sizes. See Fig. 6 for the illustration of some samples of the training/test images. In addition, we use 12 widely used images in existing literature for the test, as shown in Fig. 8. To train the proposed ensemble denoiser, we generated the noisy images by adding *i.i.d.* Gaussian white noise with standard derivation (the so-called noise level) σ, to noise-free images. Five noise levels, $\sigma = 10, 15, 20, 25, 30$, are tested in the experiments.

(a) Samples of images in training

(b) Samples of images in test

Fig. 6: Samples of images from training and test datasets.

We use $\text{NLED}^L_{M\times M}$ (*NonLocal Ensemble Denoiser*) to denote the proposed ensemble denoiser that employs the fully nonlocal ensemble structure with the configuration parameters (M, L), where L is the number of base denoiser and $M \times M$ corresponds to the filter size. Accordingly, the proposed ensemble denoiser with the fully local structure, half-and-half structure and alternative structure are denoted by $\text{LED}^L_{M\times M}$, $\text{HLED}^L_{M\times M}$, and $\text{ALED}^L_{M\times M}$ respectively. In our implementation, considering the balance between running time and denoising performance, we set the number of base denoisers to 6. The filters used in $\boldsymbol{W}$ are initialized by DCT and the ones in $\boldsymbol{W}^+$ are initialized as the inverse DCT, both with the normalization factor M^2. As a result, the number of filters is $M \times M$.

B. Performance evaluation

Recall that in Section III-B, we have two possible orders of shrinkage and merging when constructing transform-shrinkage-based denoisers. An experimental evaluation is done to see which yields better performance. It is shown in Fig. 7 that the one using shrinkage after merging has better performance. Thus, as described in Section. III-B, we selected the strategy of shrinkage after merging in our implementation.

Next, we run an evaluation on which ensemble structure of ensemble denoising performs best. In terms of PSNR value, the comparison of the results on the BSD68 dataset from $\text{NLED}^5_{5\times5}$, $\text{LED}^5_{5\times5}$, $\text{HLED}^5_{5\times5}$, $\text{ALED}^5_{5\times5}$ is summarized in Table I. It can be seen that NLED performs the best. This is not surprising. By exploiting nonlocal patch recurrence prior,

nonlocal base denoisers are generally more effective than those local ones, and thus the ensemble denoiser with the fully nonlocal structure is the best performer. The second best is ALED, which is slightly better than HLED, and the last one is LED.

These results showed that nonlocal base denoisers are more powerful than local ones. When the percentage of nonlocal based denoisers increases, the performance of the resulting ensemble denoiser is improved. Nevertheless, despite its relatively-weak performance, LED still has its value in practice, especially in those real-time applications, as its computational cost is the lowest among all. The proposed framework allows a flexible combination of local and nonlocal base denoiser to balance denoising performance and computational efficiency to fit the needs of real applications.

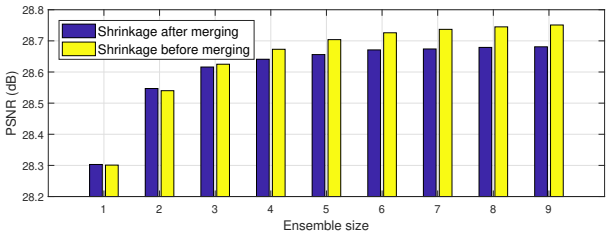


Fig. 7: Comparison of shrinkage before merging and shrinkage after merging.

TABLE I: PSNR (dB) of the denoised results by the proposed ensemble denoiser with different ensemble structures.

σ	$\text{NLED}^5_{5\times5}$	$\text{LED}^5_{5\times5}$	$\text{HLED}^5_{5\times5}$	$\text{ALED}^5_{5\times5}$
15	31.27	31.13	31.20	31.21
25	28.79	28.67	28.68	28.71

Lastly, the results from our best ensemble denoiser $\text{NLED}^6_{7\times7}$ is compared to that from several popular denoising methods, including BM3D [38], WNNM [41], EPLL [19], MLP [26], CSF [27], and TNRD [29]. See Table II for the comparison of the average PSNR on the BSD68 dataset. It can be seen that $\text{NLED}^6_{7\times7}$ is the best performer[2]. In Table III, the comparison of the PSNR values on the 12 test images shown in Fig. 8 is listed for two noise levels. It can be seen that our method yielded very competitive results, with top performance on at least half of the images. See Fig. 9 for visual inspection of some examples, and it can be seen that that our results show better edge preservation and sharper image details than other methods do.

C. Performance in different configurations

1) Performance vs. ensemble size: The performance of ensemble denoiser is evaluated with different numbers of base denoisers. The results by $\text{NLED}^6_{M\times M}$ ($M = 3, 5, 7$) on the BSD68 dataset are shown in Fig. 10. It can be seen that the increase of PSNR is fast at the beginning and becomes slower as the number of base denoisers increases. This is because at the latter stages, the input images is less noisy, and thus the improvement on SNR is less too.

2) Performance vs. filter size: The performance of the proposed ensemble denoiser $\text{NLED}^6_{M\times M}$ is evaluated with different filter size M. Some of the learned filters are shown in Fig. 11. It can be seen that larger filter encodes richer local image patterns. The PSNR results on BSD68 using different filter sizes with different noise levels are listed in Table IV, where better performance is obtained by using larger filters. It

[2]The results from MLP are missing for several noise levels. The reason is that the trained models of MLP are published online only for 2 noise levels, $\sigma = 10, 25$, among all tested noise levels in Table III.

Fig. 8: Twelve widely-used images.

TABLE II: Average PSNR (dB) of denoised results by different methods on BSD68 dataset.

σ	BM3D	WNNM	EPLL	MLP	CSF	TNRD	$\text{NLED}^6_{7\times7}$
10	33.32	33.56	33.37	33.49	33.25	**33.62**	**33.62**
15	31.07	31.37	31.21	-	31.24	31.42	**31.43**
20	29.62	29.84	30.25	-	29.55	29.97	**29.98**
25	28.57	28.83	28.68	28.96	28.74	28.92	**28.93**
30	27.75	27.97	27.84	-	27.50	28.07	**28.10**

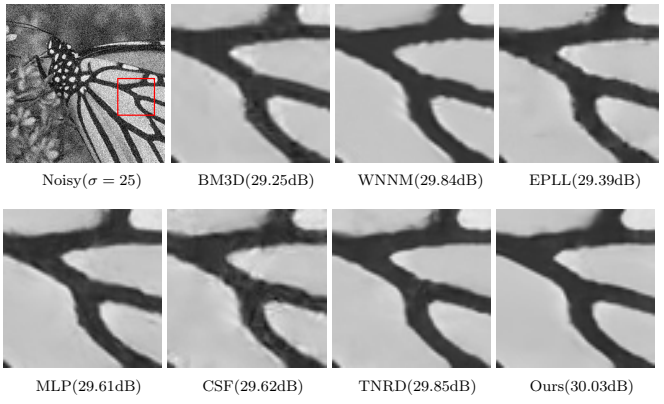
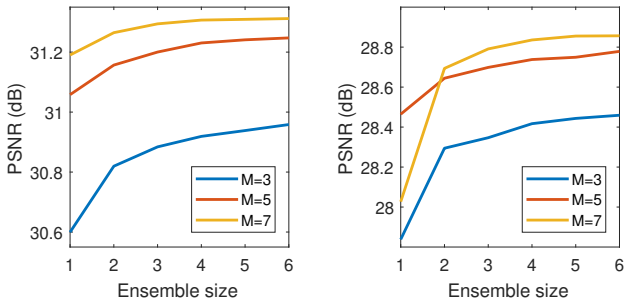


Fig. 9: Visual comparison of denoised results.

TABLE III: Performance comparison of different methods on individual images in terms of PSNR (dB).

$\sigma = 15$							
Images	BM3D	WNNM	EPLL	MLP	CSF	TNRD	Ours
Airplane	31.07	31.39	31.19	-	31.33	31.46	**31.59**
Barbara	33.11	**33.60**	31.38	-	31.92	32.13	32.53
Boats	32.14	**32.27**	31.93	-	32.01	32.14	32.16
C.man	31.91	32.17	31.85	-	31.95	32.19	**32.28**
Couple	32.11	**32.17**	31.93	-	31.98	32.11	32.13
House	34.94	**35.13**	34.17	-	34.39	34.53	34.76
Lena	34.27	34.27	33.92	-	34.06	34.24	**34.35**
Man	31.93	32.11	32.00	-	32.08	**32.23**	32.22
Monarch	31.85	**32.71**	32.10	-	32.33	32.56	**32.71**
Parrot	31.37	31.62	31.42	-	31.37	31.63	**31.70**
Peppers	32.70	32.99	32.64	-	32.85	33.04	**33.10**
Starfish	31.14	**31.82**	31.13	-	31.55	31.75	31.75
$\sigma = 25$							
Images	BM3D	WNNM	EPLL	MLP	CSF	TNRD	Ours
Airplane	28.42	28.69	28.61	28.82	28.72	28.88	**28.99**
Barbara	30.71	**31.24**	28.61	29.54	29.03	29.41	30.11
Boats	29.90	**30.03**	29.74	29.97	29.76	29.91	29.90
C.man	29.45	29.64	29.26	29.61	29.48	29.72	**29.75**
Couple	29.71	**29.82**	29.53	29.73	29.53	29.71	29.74
House	32.85	**33.22**	32.17	32.56	32.39	32.53	32.81
Lena	32.07	**32.24**	31.73	32.25	31.79	32.00	32.18
Man	29.61	29.76	29.66	**29.88**	29.71	29.87	29.86
Monarch	29.25	29.84	29.39	29.61	29.62	29.85	**30.03**
Parrot	28.93	29.15	28.95	29.25	28.90	29.18	**29.29**
Peppers	30.16	30.42	30.17	30.30	30.32	30.57	**30.66**
Starfish	28.56	29.03	28.51	28.82	28.80	29.02	**29.09**

is noted that larger filter size increases the computational cost in learning.



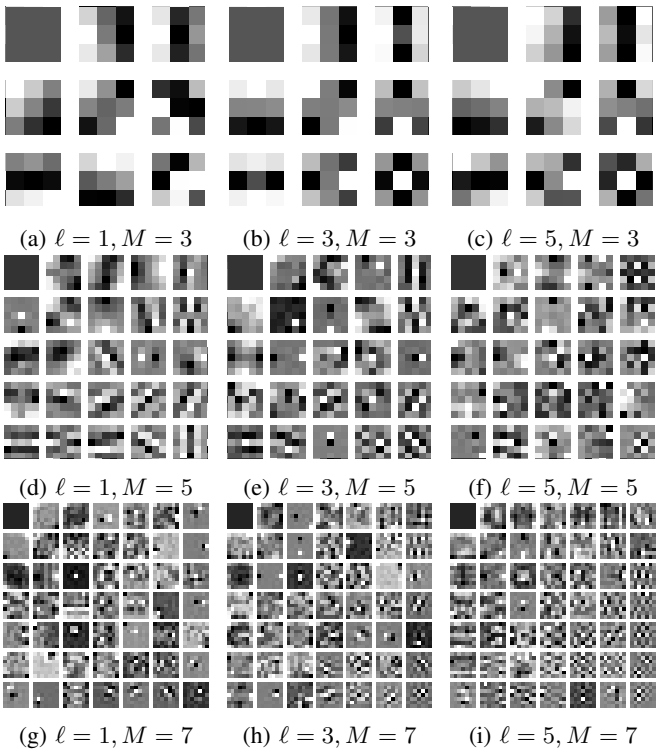
(a) $\sigma = 15$ (b) $\sigma = 25$

Fig. 10: Performance of proposed ensemble denoiser versus ensemble size. We trained $\text{NLED}^6_{M\times M}$ ($M = 3, 5, 7$) for noise level $\sigma = 15, 25$.

TABLE IV: PSNR (dB) of the denoised results by the proposed ensemble denoiser with different filter sizes.

σ	$\text{NLED}^6_{3\times3}$	$\text{NLED}^6_{5\times5}$	$\text{NLED}^6_{7\times7}$
15	31.10	31.36	31.43
25	28.56	28.88	28.93

3) Performance vs. number of training samples: It is natural to ask whether more training samples can improve the performance of ensemble denoiser. We conducted an experiment to uses 20, 50, 100, 200 and 400 training images to train an ensemble denoiser respectively. In the experiment, the noise level is set to 25 and the BSD68 dataset is used for test. See Table. V for the summary of the results. It can be seen that using a larger dataset gives little improvement on performance. Indeed, similar phenomena were also observed on other compared methods. One hypothesis for such phenomena is that the dimension of the manifold of noise-free small image patches is not very high. As image denoising essentially can be viewed as the mapping of small image patches to small image patches, the tens of image patches contained in the training images with sufficient diversity are sufficient for avoiding over-fitting.



(a) $\ell = 1, M = 3$ (b) $\ell = 3, M = 3$ (c) $\ell = 5, M = 3$
(d) $\ell = 1, M = 5$ (e) $\ell = 3, M = 5$ (f) $\ell = 5, M = 5$
(g) $\ell = 1, M = 7$ (h) $\ell = 3, M = 7$ (i) $\ell = 5, M = 7$

Fig. 11: Learned filters in the ℓth base denoiser of the ensemble denoiser $\text{NLED}^6_{M\times M}$ trained with the noise level $\sigma = 25$. The results by different ℓ and M are shown. Note that we do not learn the first low frequency pass filter in each base denoiser.

TABLE V: PSNR (dB) of the denoised results by the proposed ensemble denoiser with different numbers of training images.

Number of images	20	50	100	200	400
Average PSNR (dB)	28.44	28.63	28.68	28.71	28.72

D. Evaluation of computational cost

We implemented the proposed method using Matlab 2017a and evaluated its running time on a PC with i7-6700K CPU and Nvidia Titan V GPU. The GPU-based implementation was used to accelerate the patch matching and represent $\boldsymbol{V}$ optimized for sparse matrix. Regarding the shrinkage operation, we used a look-up table for further acceleration. See Fig. 12

for the list of running time of using our ensemble method to process an image, including: (1) the running time of two types of base denoisers (*i.e.* LED^1 and NLED^1); (2) the running time of our ensemble denoiser *vs.* filter size; (3) the running time of our ensemble denoiser *vs.* different ensemble structures. It can be seen that ALED^6 and HLED^6 have nearly the same running time, as both have the same number of local/nonlocal base denoisers. We also calculated the elapsed time for processing one image of size 256×256 using Matlab code of all compared methods codes under the same environment. The results are listed in Table VI. Regarding training time, it takes around 8.5 hours to train a local base denoiser and 9.5 hours for a nonlocal base denoiser, when filter sizes are both set to 7. The entire training process of $\text{NLED}^6_{7\times7}$ takes about 2.5 days.

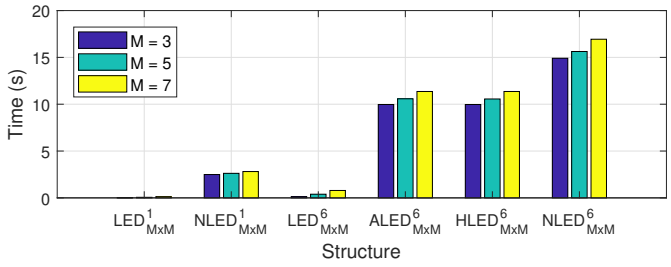


Fig. 12: Running time (seconds) vs. different filter sizes and different structures for processing a 256×256 image. Each result is the average over 10 runs on different images.

TABLE VI: Average running time (seconds) for processing an input image of size 256×256.

Method	BM3D	WNNM	EPLL	MLP	CSF	TNRD	$\text{LED}^6_{7\times7}$	$\text{NLED}^6_{7\times7}$
Time (s)	0.28	108.80	25.52	2.56	2.20	0.94	0.79	16.94

E. Observations

In ensemble learning, the diversity among base learners is crucial to the performance. In this subsection, we use $\text{NLED}^L_{M\times M}$ to study the behavior of individual base denoisers.

1) Learned shrinkage functions: Some examples of the shrinkage functions learned on the base denoisers of $\text{NLED}^6_{3\times3}$ are shown in Fig. 13, in which some diversities among the base denoisers can be found. In the early base denoisers, the range of coefficients that are set to zero is larger than that of the subsequent ones. In the subsequent denoisers, the shrinkage functions tend to only fine-tune the input, since most of noise has already been removed. In other words, for the last few denoisers, the image patches are just to be "rectified" so that they look more similar to the patterns of filters. This partially explains why the ensemble denoiser using larger filter size yields better performance, *i.e.*, more interesting patterns encoded by filters are involved for fine tuning the result.

2) Learned filters: In this study, the difference of two filter banks from two base denoisers B_i, B_j, $\{\boldsymbol{f}_k^{(i)}\}_{k=1}^K$, $\{\boldsymbol{f}_k^{(j)}\}_{k=1}^K$,

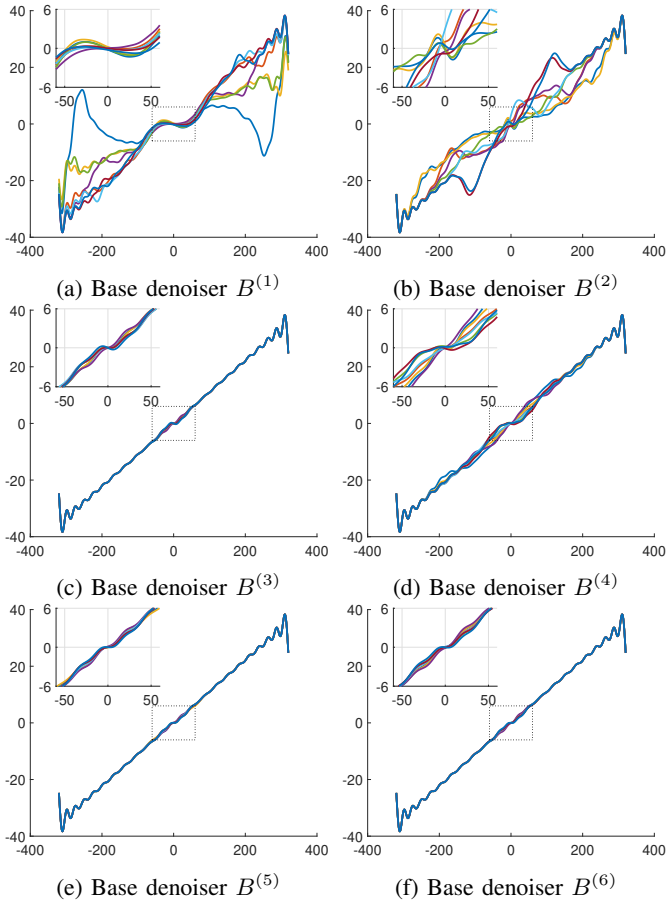


Fig. 13: Learned shrinkage functions of each base denoiser in $\text{NLED}^6_{3\times3}$ for the noise level $\sigma = 25$.

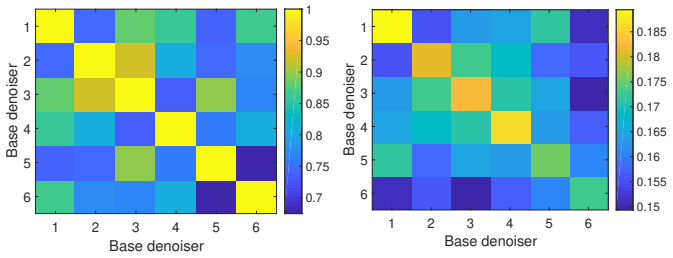
is quantified by both the maximal correlation and the average correlation of all possible pairs:

$$\max_{p,q} \frac{< \boldsymbol{f}_p^{(i)}, \boldsymbol{f}_q^{(j)} >}{\|\boldsymbol{f}_p^{(i)}\|_2 \|\boldsymbol{f}_q^{(i)}\|_2}, \quad \text{and} \quad \frac{1}{K^2} \sum_{p,q} \frac{< \boldsymbol{f}_p^{(i)}, \boldsymbol{f}_q^{(j)} >}{\|\boldsymbol{f}_p^{(i)}\|_2 \|\boldsymbol{f}_q^{(i)}\|_2}.$$

Lower correlation of two filter banks implies higher diversity between two base denoisers. See Fig. 14 for the results. It can be seen that overall, the correlations of filter banks among base denoisers are not high. Thus, although the base denoisers are initialized with the same filters, the diversity of filter banks of the base denoisers can be improved during learning. Also, both maximal and mean correlation coefficients between adjacent base denoisers tend to decrease as ℓ becomes larger, after ignoring the results related to B_1.

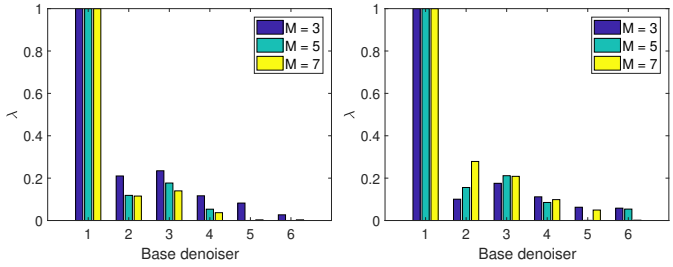
3) Tendency of the parameter λ: Recall that the parameter λ determines the percentage of residuals that should be merged into the last estimate of clear image. The values of λ at different base denoisers, namely $\lambda^{(\ell)}$, is listed in Fig. 15 in different configurations. It can seen that $\lambda^{(\ell)}$ tends to become smaller with the increase of ℓ. The reason is that as the estimates on noise-free images becomes more accurate, the estimates should be trusted more.

4) Effectiveness of resampling: To verify the effectiveness of the proposed resampling strategy in training ensemble denoisers, we constructed two baseline methods by (i) training



(a) Maximal correlation (b) Mean correlation

Fig. 14: Correlation matrices of filters of base denoisers in $\text{NLED}^6_{5\times5}$. The ensemble denoiser $\text{NLED}^6_{5\times5}$ is trained on the noise level $\sigma = 25$.



(a) $\sigma = 15$ (b) $\sigma = 25$

Fig. 15: Learned λ_k for all k in $\text{NLED}^6_{M\times M}$ with different M.

ensemble denoisers without resampling; and (ii) identifying the images that yield large prediction loss in the previous base denoiser and then increasing the weights of these images in the loss function of current base denoiser. The second scheme is an ad-hoc way to do resampling that makes current base denoiser focus more on hard samples. The comparison of the results in PSNR using different resampling schemes is shown in Fig. 16. It is interesting to see that using the ad-hoc reweighing based resampling scheme is harmful to the performance of ensemble denoisers, which demonstrates that resampling cannot be casually designed for performance boost. In contrast, our proposed resampling scheme does lead to performance gain in denoising.

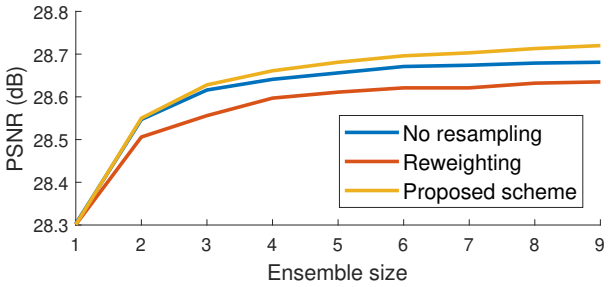


Fig. 16: Comparison of different resampling schemes.

V. Conclusion

This paper introduced sequential ensemble learning to the design of image denoiser. An ensemble framework for constructing effective denoiser from a set of base denoisers is presented, including various types of base denoisers and different ensemble structures. A resampling scheme is also proposed for further performance improvement. Comprehensive experiments were conducted to study most aspects of proposed ensemble framework, as well as to evaluate the performance of the proposed ensemble denoiser. The experimental results showed the effectiveness of the proposed method, which indicates the potential of ensemble learning in image recovery. In future, we would like to investigate the extension of the proposed ensemble framework to other image recovery problems. In addition, we would like to investigate more image priors for designing diverse base denoisers, as well as develop adaptive construction schemes for constructing ensemble denoisers with better performance.

References

[1] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM J. Imaging Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.
[2] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play admm for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comput. Imaging*, vol. 3, no. 1, pp. 84–98, 2017.
[3] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
[4] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, 2009.
[5] E. P. Simoncelli and E. H. Adelson, "Noise removal via bayesian wavelet coring," in *Proc. Int. Conf. Image Proc.*, vol. 1. IEEE, 1996, pp. 379–382.
[6] Y. Yu, P. Guo, Y. Chen, P. Chen, and K. Guo, "Graph laplacian and dictionary learning, lagrangian method for image denoising," in *Proc. IEEE Int. Conf. Signal & Image Process.* IEEE, 2016, pp. 236–240.
[7] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
[8] A. Danielyan, V. Katkovnik, and K. Egiazarian, "Bm3d frames and variational image deblurring," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1715–1728, 2012.
[9] M. Zontak and M. Irani, "Internal statistics of a single natural image," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*. IEEE, 2011, pp. 977–984.
[10] M. Zontak, I. Mosseri, and M. Irani, "Separating signal from noise using patch recurrence across scales," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2013, pp. 1195–1202.
[11] H. Talebi and P. Milanfar, "Global image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 755–768, 2014.
[12] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.
[13] C. Bao, H. Ji, Y. Quan, and Z. Shen, "Dictionary learning for sparse coding: Algorithms and convergence analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1356–1369, 2016.
[14] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1553–1564, 2010.
[15] Y. Romano and M. Elad, "Boosting of image denoising algorithms," *SIAM J. Imaging Sci.*, vol. 8, no. 2, pp. 1187–1219, 2015.
[16] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, 2003.
[17] S. Lyu and E. P. Simoncelli, "Modeling multiscale subbands of photographic images with fields of gaussian scale mixtures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 693–706, 2009.
[18] S. Roth and M. J. Black, "Fields of experts," *Int. J. Comput. Vision*, vol. 82, no. 2, pp. 205–229, 2009.
[19] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. IEEE Int. Conf. Comput. Vision*. IEEE, 2011, pp. 479–486.
[20] V. Papyan and M. Elad, "Multi-scale patch-based image restoration," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 249–261, 2016.

[21] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, "Patch group based nonlocal self-similarity prior learning for image denoising," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 244–252.
[22] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Inform. Process. Syst.*, 2009, pp. 769–776.
[23] 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. Image Process.*, 2017.
[24] F. Agostinelli, M. R. Anderson, and H. Lee, "Adaptive multi-column deep neural networks with application to robust image denoising," in *Advances in Neural Inform. Process. Syst.*, 2013, pp. 1493–1501.
[25] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in Neural Inform. Process. Syst.*, 2012, pp. 341–349.
[26] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?" in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*. IEEE, 2012, pp. 2392–2399.
[27] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2014, pp. 2774–2781.
[28] U. Schmidt, J. Jancsary, S. Nowozin, S. Roth, and C. Rother, "Cascades of regression tree fields for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 677–689, 2016.
[29] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, 2017.
[30] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*. IEEE, 2011, pp. 457–464.
[31] G. Peyré, "Manifold models for signals and images," *Comput. Vision & Image Understanding*, vol. 113, no. 2, pp. 249–260, 2009.
[32] T. G. Dietterich *et al.*, "Ensemble methods in machine learning," *Multiple Classifier Syst.*, vol. 1857, pp. 1–15, 2000.
[33] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1395–1411, 2007.
[34] S. Mallat, *A wavelet tour of signal processing*. Academic press, 1999.
[35] C. Bao, J.-F. Cai, and H. Ji, "Fast sparsity-based orthogonal dictionary learning for image restoration," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 3384–3391.
[36] J.-F. Cai, H. Ji, Z. Shen, and G.-B. Ye, "Data-driven tight frame construction and image denoising," *Appli. Comput. Harmonic Anal.*, vol. 37, no. 1, pp. 89–105, 2014.
[37] R. Yan, L. Shao, and Y. Liu, "Nonlocal hierarchical dictionary learning using wavelets for image denoising," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4689–4698, 2013.
[38] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.
[39] P. Chatterjee and P. Milanfar, "Clustering-based denoising with locally learned dictionaries," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1438–1451, 2009.
[40] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: A low-rank approach," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 700–711, 2013.
[41] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2014, pp. 2862–2869.
[42] M. Lebrun, A. Buades, and J.-M. Morel, "A nonlocal bayesian image denoising algorithm," *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1665–1688, 2013.
[43] I. Ram, M. Elad, and I. Cohen, "Image processing using smooth ordering of its patches," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2764–2774, 2013.
[44] J. Pang and G. Cheung, "Graph laplacian regularization for image denoising: Analysis in the continuous domain," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1770–1785, 2017.
[45] P. Chatterjee and P. Milanfar, "Patch-based near-optimal image denoising," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1635–1649, 2012.
[46] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE Int. Conf. Comput. Vision*. IEEE, 2009, pp. 2272–2279.
[47] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, 2013.
[48] W. Dong, G. Shi, Y. Ma, and X. Li, "Image restoration via simultaneous sparse coding: Where structured sparsity meets gaussian scale mixture," *Proc. Int. J. Comput. Vision*, vol. 114, no. 2-3, pp. 217–232, 2015.
[49] H. Liu, R. Xiong, J. Zhang, and W. Gao, "Image denoising via adaptive soft-thresholding based on non-local samples," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2015, pp. 484–492.
[50] R. Vemulapalli, O. Tuzel, and M.-Y. Liu, "Deep gaussian conditional random field network: A model-based deep network for discriminative denoising," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2016, pp. 4801–4809.
[51] G. Chen and B. Kégl, "Image denoising with complex ridgelets," *Pattern Recognition*, vol. 40, no. 2, pp. 578–585, 2007.
[52] P. Zhang, T. D. Bui, and C. Y. Suen, "A novel cascade ensemble classifier system with a high recognition performance on handwritten digits," *Pattern Recognition*, vol. 40, no. 12, pp. 3415–3429, 2007.
[53] J. Domke, "Generic methods for optimization-based modeling," in *Artificial Intell. & Stat.*, 2012, pp. 318–326.
[54] 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 *Proc. IEEE Int. Conf. Comput. Vision*, vol. 2, July 2001, pp. 416–423.

Xuhui Yang received the B.Eng degree in Computer Science from South China University of Technology in 2015. He continues his research life as a PH.D candidate. His research interests include image processing, machine learning and sparse representation.

Yong Xu received the B.S., M.S. and Ph.D. degrees in mathematics from Nanjing University, Nanjing, China, in 1993, 1996, and 1999, respectively. He was a Post-Doctoral Research Fellow of computer science with South China University of Technology, Guangzhou, China, from 1999 to 2001, where he became a Faculty Member and where he is currently a Professor with the School of Computer Science and Engineering. His current research interests include image analysis, video recognition, and image quality assessment. Dr. Xu is a member of the IEEE Computer Society and the ACM.

Yuhui Quan received the Ph.D. degree in Computer Science from South China University of Technology in 2013. He worked as the postdoc research fellow in Mathematics at National University of Singapore from 2013 to 2016. He is currently the associate professor at School of Computer Science and Engineering in South China University of Technology. His research interests include computer vision, image processing and sparse representation.

Hui Ji received the B.Sc. degree in Mathematics from Nanjing University in China, the M.Sc. degree in Mathematics from National University of Singapore and the Ph.D. degree in Computer Science from the University of Maryland, College Park. In 2006, the joined National University of Singapore as an assistant professor in Mathematics. Currently, he is an associate professor in mathematics at National University of Singapore. His research interests include computational harmonic analysis, optimization, computational vision, image processing and biological imaging.