Number of Useful Components in Gaussian Mixture Models for Patch-Based Image Denoising

Dai-Viet Tran[1,2(✉)], Sébastien Li-Thiao-Té[1], Marie Luong[2], Thuong Le-Tien[3], and Françoise Dibos[1]

[1] Université Paris 13, Sorbonne Paris Cité, LAGA, CNRS UMR 7539, Villetaneuse, France
tran@math.univ-paris13.fr

[2] Université Paris 13, Sorbonne Paris Cité, L2TI, EA 3043, Villetaneuse, France

[3] Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam

Abstract. When using Gaussian mixture models (GMMs) as a prior for image denoising under the Bayesian maximum a posteriori (MAP) perspective, only a single prominent Gaussian component is usually selected to recover a noisy image patch, which leads to computationally efficient implementations. We attempt to justify this on several image datasets by evaluating the number of Gaussian components required for recovering patches. We show that even patches without a prominent component in the prior can be recovered with little loss of performance. Comparisons between two dictionary choices and between small and large models suggest that large gains are attainable, but only one component is required for reconstruction.

Keywords: Gaussian mixture model · Image denoising · Image priors

1 Introduction

Image denoising aims to recover a latent clean image $\mathbf{X}$ from its degraded observation $\mathbf{Y}$, which is modeled as $\mathbf{Y} = \mathbf{X} + \mathbf{V}$ where $\mathbf{V} \sim \mathcal{N}(0, \sigma^2)$ denotes Gaussian noise with zero mean and standard deviation σ. It is an ill-posed inverse problem because noise prevents the full recovery of image details. Consequently, prior information is used to regularize the problem: internal reference in nonlocal self-similarity [1,2], arbitrary priors in sparse representation [3,4], or estimated priors in deep-learning [5,6]. In patch-based image denoising, each image is partitioned into a set of overlapping patches, and the degradation model is written on each patch i as $\mathbf{y}_i = \mathbf{x}_i + \mathbf{v}_i$. Without loss of generality, we can represent image patches in the vector space generated by a dictionary $\mathbf{D} \in R^{m \times M}$, i.e. a basis set of M basic vectors (also called atoms) of size m. For each image patch $\mathbf{y}_i$, our objective is to seek a vector $\boldsymbol{\alpha}_i \in R^M$ such that the clean latent patch $\mathbf{x}_i$ verifies $\mathbf{x}_i = \mathbf{D}\boldsymbol{\alpha}_i$. Using MAP we have:

$$\hat{\boldsymbol{\alpha}}_i = \arg\max_{\boldsymbol{\alpha}_i} \log(p(\boldsymbol{\alpha}_i|\mathbf{D}, \mathbf{y}_i)) \propto \arg\min_{\boldsymbol{\alpha}_i} \{||\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 - \lambda \log(p(\boldsymbol{\alpha}_i))\} \tag{1}$$

where $\lambda = \tau\sigma^2$ (with $\tau > 0$) controls the amount of regularization.

A. Mansouri et al. (Eds.): ICISP 2018, LNCS 10884, pp. 108–116, 2018.
https://doi.org/10.1007/978-3-319-94211-7_13

We consider a Gaussian Mixture Model (GMM) to model the distribution of image patches, i.e. $p(\boldsymbol{\alpha}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\alpha}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where π_k are the mixing weights, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ are the mean and covariance matrix of the k^{th} component. The model can be estimated from a set of image patches [7] or a collaborative group of patches [8,9] extracted from standard images. However, solving Eq. (1) with the complete mixture is time-consuming and, to our knowledge, existing methods only use one prominent component for each image patch and justification for this is lacking. In this contribution, we divide the patches in an input image into a set P_1 of simple patches with a prominent component and a set P_2 of the remaining patches. We focus on the set P_2 and conduct multiple experiments to show that only marginal gains can be obtained by considering the full GMM in denoising. We explore different choices of dictionary (identity matrix and K-SVD based) and two choices of GMM complexity on PSNR and reconstruction error and discuss the type of images that are difficult to reconstruct.

The rest of the paper is organized as follows. Section 2 gives a brief introduction to the datasets. The details of the Gaussian mixture model as Prior for Image Denoising (GPID) method are described in Sect. 3. Section 4 presents the experimental results and discussion.

2 Datasets

We explore the following datasets, with different image types and structures:

Cartoon [10] contains 590 images of popular cartoon characters. We choose 45 images to train a GMM and 80 images for evaluation.

Urban [11] contains urban scenes with high self-similarity and many repeated patterns. We use 25 images for training and 25 images for denoising.

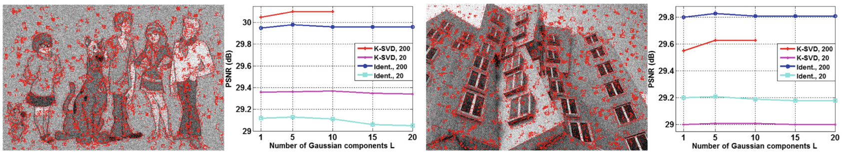
Nature We use 200 training images in [12] and 20 popular natural test images presented in [8].

Brodatz [13] contains 112 grayscale images of natural textures. We select 30 good quality and content-rich images and split each of them into 4 non-overlapping sub-images. 90 sub-images are used for training the GMM and 30 sub-images for denoising.

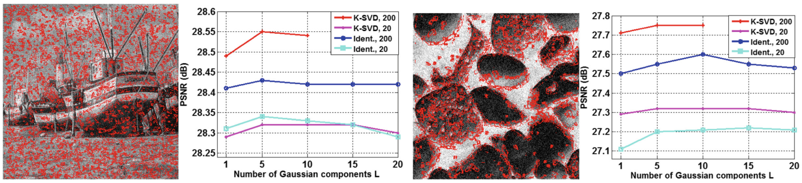
Dtd [14] contains textural images in the wild such as band, braid, spiral, grid, etc. We choose 55 images for training and 40 images for denoising.

CT of Thorax and CT of Lung We download 7 sequences of CT lung images and 12 sequences of CT thorax images from [15]. 40 thorax images are used for training the GMM and 40 images for testing. The numbers of images for training and testing of CT images of Lung are 40 and 60.

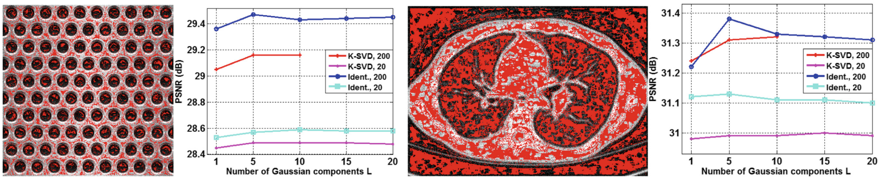
MRI Brain We download 16 sequences of MRI brain images from [16]. 80 images are selected from 7 sequences for training and 60 images are chosen in other sequences for denoising.



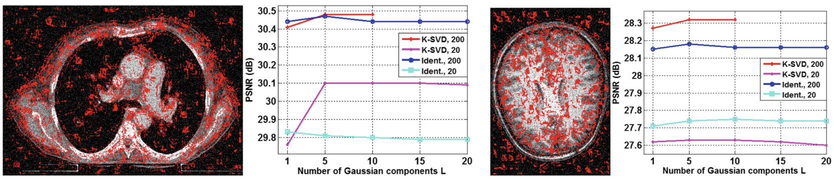
Sas34 (Cartoon), $p = 83.9$, $e = 1.2$ Img75 (Urban), $p = 86.6$, $e = 1.2$



Boat (Nature), $p = 77.1$, $e = 1.1$ D31 (Brodatz), $p = 79.6$, $e = 1.2$.



Perforated_0012 (DTD), $p = 79.5$, $e = 1.2$ CT of Lung, $p = 52.7$, $e = 1.2$



CT of Thorax, $p = 83$, $e = 1.1$ MRI of Brain, $p = 79.4$, $e = 1.2$

Fig. 1. Examples from the 8 datasets, test image with P_2 patches in red (left), PSNR as a function of L (right). Captions indicate image name, percentage of P_1 patches and average reconstruction error $||\hat{\mathbf{X}}_{L=1} - \hat{\mathbf{X}}_{L=5}||_{L_1}$. (Color figure online)

3 Image Denoising with a Gaussian Mixture Model

3.1 Training the GMM on a Patch Database

From the training set of good quality noise-free images, we randomly extract N patches $\mathbf{x}_j$ $(1 \leq j \leq N)$ of size $\sqrt{m} \times \sqrt{m}$. After mean subtraction, each patch $\mathbf{x}_j$ is encoded in the vector space generated by M atoms of dictionary $\mathbf{D}$, $\mathbf{x}_j = \mathbf{D}\boldsymbol{\alpha}_j$ by least squares fitting $\boldsymbol{\alpha}_j = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{x}_j$.

The probability distribution of the patch coordinates $\boldsymbol{\alpha}_j$ can be modeled by a GMM of K components, $p(\boldsymbol{\alpha}_j) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. We can get the parameters $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ by maximizing the likelihood function via Expectation-Maximization algorithm. In the E-step, the expected value of the conditional probability of $\boldsymbol{\alpha}_j$ given the parameters of GMM (also called the "membership probability") is computed as in (2).

$$\gamma_{jk} = \frac{\pi_k \mathcal{N}(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\boldsymbol{\alpha}_j|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} \tag{2}$$

In the M-step, the parameters of Gaussian components are updated using (3)

$$\boldsymbol{\mu}_k = \frac{\sum_{j=1}^{N} \gamma_{jk}\boldsymbol{\alpha}_j}{\sum_{j=1}^{N} \gamma_{jk}}; \boldsymbol{\Sigma}_k = \frac{\sum_{j=1}^{N} \gamma_{jk}(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_k)^T(\boldsymbol{\alpha}_j - \boldsymbol{\mu}_k)}{\sum_{j=1}^{N} \gamma_{jk}}; \pi_k = \frac{\sum_{j=1}^{N} \gamma_{jk}}{N} \tag{3}$$

3.2 Denoising Algorithm

After mean substraction, denoising a patch $\mathbf{y}_i$ is equivalent to finding the optimal clean latent patch $\hat{\mathbf{x}}_i = \mathbf{D}\hat{\boldsymbol{\alpha}}_i$. Solving problem (1) with the whole GMM of $p(\boldsymbol{\alpha}_j)$ is a very time-consuming process. To overcome this issue, existing studies [7–9] propose to assign the noisy patch $\mathbf{y}_i$ to a single Gaussian component according to the posterior probability:

$$\gamma_{il} = \frac{\pi_l \mathcal{N}(\mathbf{y}_i|\mathbf{D}\boldsymbol{\mu}_l, \mathbf{D}\boldsymbol{\Sigma}_l\mathbf{D}^T)}{\sum_{n=1}^{K} \pi_n \mathcal{N}(\mathbf{y}_i|\mathbf{D}\boldsymbol{\mu}_n, \mathbf{D}\boldsymbol{\Sigma}_n\mathbf{D}^T)} \tag{4}$$

where $0 \leq \gamma_{il} \leq 1$ and $\sum_{l=1}^{K} \gamma_{il} = 1$. For convenience, we assume that $\gamma_{i1} \geq \gamma_{i2} \geq \ldots \geq \gamma_{iK}$. Problem (1) has a closed form solution when using only γ_{i1}

$$\hat{\boldsymbol{\alpha}}_i = \left(\mathbf{D}^T\mathbf{D} + \lambda\boldsymbol{\Sigma}_1^{-1}\right)^{-1}\left(\mathbf{D}^T\mathbf{y}_i + \lambda\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1\right) \tag{5}$$

Typically, this approach is acceptable when the first component is dominant and the other components do not contribute much to the optimization. In practice, we define the set of dominant patches $P_1 = \{\mathbf{y}_i \, s.t. \, \gamma_{i1} \geq 0.9\}$ and we call P_2 the set of the remaining patches. P_1 patches are restored via (5) whereas the patches in P_2 are restored by considering the largest L components of the GMM. Consequently, in the following, we only solve the simplified problem (6). The denoising method is presented in Algorithm 1.

$$\boldsymbol{\alpha}_i = \underset{\boldsymbol{\alpha}_i}{\arg\min} \, ||\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 - \lambda \log \left\{ \sum_{l=1}^{L} \pi_l \mathcal{N}(\boldsymbol{\alpha}_i|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \right\} \tag{6}$$

3.3 Complexity Analysis

The denoising method GPID consists of two parts: off-line training and denoising. In the training phase, the overall complexity to learn the GMM of K components from N decomposition vectors $\boldsymbol{\alpha}_j \in R^M$ is $O(KM^3N)$. In the denoising process, P_1 patches are restored via (5) that needs $O(M^2P_1)$ operations, and P_2 patches ($P_2 \approx 10\%P - 20\%P$) are recovered using gradient descent with the complexity $O(LT_{gd}M^3P_2)$, where T_{gd} is the number of iterations of the gradient descent algorithm. The computation of the membership probabilities requires $O(m^3PK)$ operations. The denoising step is repeated T times and therefore totally takes $O(m^3PKT + M^2P_1T + LT_{gd}M^3P_2T)$ complexity.

Algorithm 1. GMM as Prior for Image Denoising (GPID)

1 Learn the GMM model $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ (once per image database)
2 Initialization: $\mathbf{X}^{(0)} = \mathbf{Y}$.
3 **for** $t = 1$ *to* T **do**
4   **for** *each patch* $\mathbf{y}_i \in \mathbf{Y}$ **do**
5   - Subtract its mean value (μ_y): $\mathbf{y}_i = \mathbf{y}_i - \mu_y$.
6   - Calculate γ_{il} $(1 \leq l \leq K)$ via (4) and arrange in descending order.
7   - **If** $\gamma_{i1} \geq 0.9$ **then** $\hat{\boldsymbol{\alpha}}_i = \left(\mathbf{D}^T\mathbf{D} + \lambda\boldsymbol{\Sigma}_1^{-1}\right)^{-1}\left(\mathbf{D}^T\mathbf{y}_i + \lambda\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1\right)$
8   - **Else** select L components with largest value of γ_{il}, **then** solve for $\hat{\boldsymbol{\alpha}}_i$ in (6) using gradient descent.
9   - Restore the noisy patch: $\hat{\mathbf{x}} = \mathbf{D}\hat{\boldsymbol{\alpha}}_i + \mu_y$.
10   **end**
11   Aggregate the denoised patches $\hat{\mathbf{x}}$ to recover the entire denoised image $\mathbf{X}^t$
12   Regularize the denoised image: $\mathbf{X}^{(t)} = (\eta\mathbf{Y}^{(t-1)} + \beta\mathbf{X}^t)/(\eta + \beta)$.
13 **end**

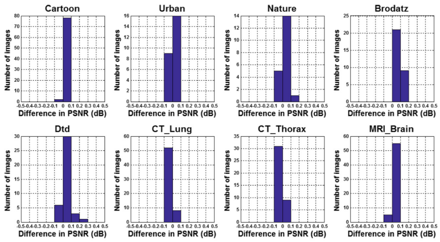
4 Experimental Results

To show the effect of the restriction to the dominant component, we examine the performance of the GPID method on P_2 patches with a varying number of components $L \in \{1, 5, 10, 15, 20\}$ in step 7 of the optimization algorithm[1]. We study the differences in peak signal-to-noise ratio (PSNR) and mean gray-level reconstruction error for the 8 datasets presented in Sect. 2, for the identity dictionary $\mathbf{D} = \mathbf{I}$ and a K-SVD dictionary, and for small ($K = 20$) and large ($K = 200$) numbers of components.

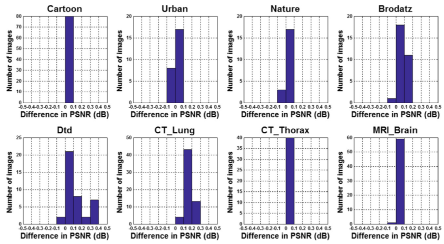
In all experiments, we degrade the images from the database with white Gaussian noise with standard deviation $\sigma = 30$. We train the two GMMs for each dataset on $N = 2.10^6$ randomly extracted patches of size $m = 8 \times 8$. In the GPID denoising method, we use $T = 5$ regularizing iterations with $\eta = m^2/\sigma^2$, $\beta = [1, 4, 8, 16, 32]/\sigma^2$. We set $\lambda = 0.9\sigma^2$ and 1000 maximum iterations in gradient descent optimization. All the experiments are implemented in the Matlab 2013a environment on a machine with Intel Core i7-4770K CPU of 3.5 GHz and 16 GB of RAM.

From the examples in Fig. 1, we notice that P_2 patches can usually be found close to the edges or contours. We also compute the PSNR values obtained for the GPID method as a function of L. On these examples, only modest gains can be obtained by considering several components in the reconstruction. These properties are explored further by computing the distributions of PSNR gains and reconstruction error.

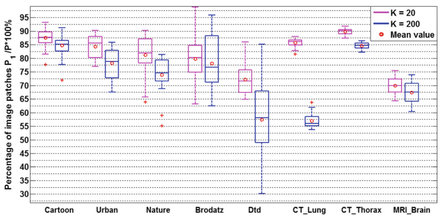
[1] We only present $L \in \{1, 5, 10\}$ for the K-SVD dictionary in the case $K = 200$ due to the high computational requirements.



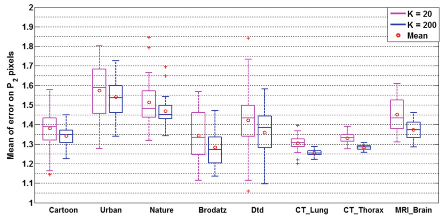
Maximum PSNR improvement for $K = 20$



Maximum PSNR improvement for $K = 200$



Percentage of P_1 pixels



Average reconstruction error $||\hat{\mathbf{X}}_{L=1} - \hat{\mathbf{X}}_{L=5}||_{L_1}$

Fig. 2. Denoising performance for the identity dictionary

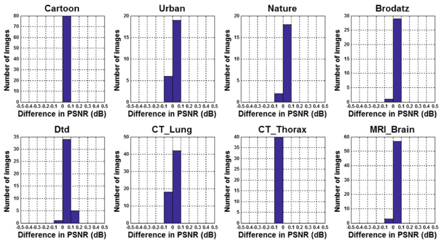
4.1 Denoising Performance

When using the identity matrix as a dictionary, image patches are denoised without transformation. Note that the GPID method coincides with the method EPLL proposed by Zoran and Weiss in [7] when $L = 1$. We first observe that most image patches correspond to a single dominant component from Fig. 2. As expected, more complex images such as textures (Brodatz and Dtd datasets) require more components and have more P_2 patches.

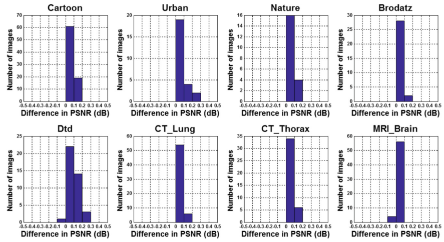
We compute the PSNR for P_2 patches for $L \in \{1, 5, 10, 15, 20\}$ and study the distribution of the maximum improvement $(\max_L PSNR) - PSNR_{L=1}$.

As shown on Fig. 2, for five datasets (Cartoon, Urban, Nature, CT Thorax and MRI), the maximum improvement is negligible, less than 0.1 dB for all test images except one from Nature. For complex images such as textures (Brodatz and DTD datasets) and the CT Lung images, some images can be modestly improved, up to 0.2 dB for $K = 20$. PSNR gains larger than 0.2 dB are only observed for complex images (Dtd and CT Lung), with $K = 200$, i.e. with enough components in the GMM to model distribution details and only for a small fraction of images (around 20%). These gains require around 200 s for a 256×256 image, whereas only 10 s are needed for one Gaussian component.

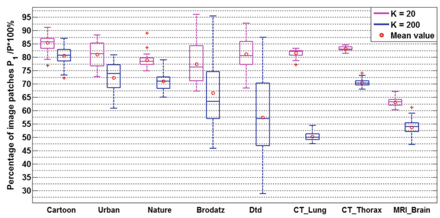
We also analyze the reconstruction error on the central pixel of P_2 patches $||\hat{\mathbf{X}}_{L=1} - \hat{\mathbf{X}}_{L=5}||_{L_1}$ on Fig. 2. For all images, this difference is less than 2 gray levels per pixel, and cannot be seen by eye.



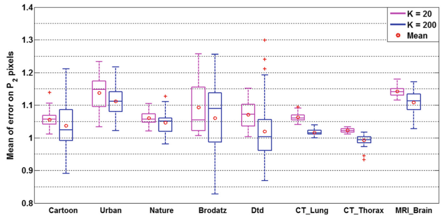
Maximum PSNR improvement for $K = 20$



Maximum PSNR improvement for $K = 200$



Percentage of P_1 pixels



Average reconstruction error $||\hat{\mathbf{X}}_{L=1} - \hat{\mathbf{X}}_{L=5}||_{L_1}$

Fig. 3. Denoising performance for the K-SVD dictionary with 256 atoms

For each dataset, we learn an over-complete dictionary $\mathbf{D}$ with 256 atoms as in [3]. Figure 3 shows a similar situation as Fig. 2. Most patches in the test images belong to P_1, and most images can be reconstructed with only one component with a penalty less than 0.1 dB. PSNR gains larger than 0.2 dB can only be observed for a few complex images in the Dtd and Urban datasets. Gray-level differences are lower than for the identity dictionary, around 1.1 gray-levels per pixel.

4.2 Dictionary Choice and Model Complexity

Using the denoising results from the 8 datasets with $L = 1$, we compare the two dictionaries and GMM sizes in Fig. 4 (see also the examples in Fig. 1). We observe that increasing the GMM model complexity is nearly always beneficial, sometimes up to 2 dB PSNR gains, and that the K-SVD dictionary tends to benefit more from $K = 200$. The K-SVD dictionary yields slightly better PSNR especially for large GMM models overall, but the results are variable, which implies that dictionary choice is largely image-specific.

In this manuscript, we consider only two sizes for the dictionary $\mathbf{D}$. When $\mathbf{D}$ is the identity matrix, its size corresponds to the patch size m. With the K-SVD method, we obtain an overcomplete dictionary $\mathbf{D}$ with 256 vectors. In both cases the dictionary determines the basis set for representing image patches and is sufficiently rich to represent all image features. Dictionaries of the same size

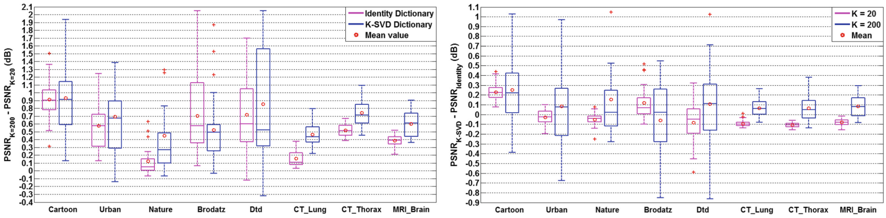


Fig. 4. Effect of model complexity (left) and dictionary choice (right) for $L = 1$

would yield the same results due to basis rotation. Our experiments suggest that complex GMM models can take advantage of the additional degrees of freedom of an overcomplete dictionary to model image details.

5 Conclusion

This paper studies the number of useful components in the GMM for patch-based image denoising on 8 image datasets. We first remark that most of the patches in an input image are well represented by a single prominent component. By exploring denoising with increasing number of components $L \in \{1, 5, 10, 15, 20\}$, we show that only modest gains can be obtained in terms of PSNR and L1 reconstruction error (gray-level differences) in all datasets when using more than one component. This justifies current practice and drastically reduces computational cost. Much larger improvements can be obtained with a suitable dictionary and GMM model, but reconstruction only requires a single component.

References

1. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: CVPR, pp. 60–65 (2005)
2. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. 2080–2095 (2007)
3. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Sig. Process. 4311–4322 (2006)
4. Trinh, D.H., Luong, M., Dibos, F., Rocchisani, J.M., Pham, C.D., Nguyen, T.Q.: Novel example-based method for super-resolution and denoising of medical images. IEEE Trans. Image Process. 1882–1895 (2014)
5. Jain, V., Seung, S.: Natural image denoising with convolutional networks. Adv. Neural Inf. Process. Syst. **21**, 769–776 (2009)
6. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans. Image Process. 3142–3155 (2017)

7. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: IEEE International Conference on Computer Vision, pp. 479–486 (2011)
8. Xu, J., Zhang, L., Zuo, W., Zhang, D., Feng, X.: Patch group based nonlocal self-similarity prior learning for image denoising. In: IEEE ICCV, pp. 244–252 (2015)
9. Niknejad, M., Rabbani, H., Babaie-Zadeh, M.: Image restoration using Gaussian mixture models with spatially constrained patch clustering. IEEE TIP 3624–3636 (2015)
10. Khan, F.S., Anwer, R.M., van de Weijer, J., Bagdanov, A.D., Vanrell, M., Lopez, A.M.: Color attributes for object detection. In: CVPR, pp. 3306–3313 (2012)
11. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR, pp. 5197–5206 (2015)
12. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 898–916 (2011)
13. Brodatz, P.: Textures: A Photographic Album for Artists and Designers. Dover, New York (1966)
14. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: CVPR (2014)
15. Clark, K., et al.: The cancer imaging archive (TCIA): maintaining and operating a public information repository. J. Digit. Imaging 1045–1057 (2013)
16. Martino, A.D., et al.: The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. Mol. Psychiatry 659–667 (2013)