Revisiting Convolutional Sparse Coding for Image Denoising: From a Multi-Scale Perspective

Jingyi Xu, Xin Deng, *Member, IEEE*, and Mai Xu, *Senior Member, IEEE*

***Abstract*—Recently, convolutional sparse coding (CSC) has shown great success in many image processing tasks, such as image super-resolution and image separation. However, it performs poorly in image denoising task. In this letter, we provide a new insight for CSC denoising by revisiting the CSC from a multi-scale perspective. We propose a multi-scale CSC model for image denoising. By unrolling the multi-scale solution into a learnable network, we obtain an interpretable lightweight multi-scale network, namely MCSCNet. Experimental results show that the proposed MCSCNet significantly advances the denoising performance, with an average PSNR improvement of 0.32 dB over the state-of-the-art (SOTA) CSC based method. In addition, our MCSCNet is on par with many SOTA deep learning based methods, with less network parameters and lower FLOPs. The ablation study also validates the effectiveness of the multi-scale CSC mechanism.**

***Index Terms*—Convolutional sparse coding, image denoising.**

I. Introduction

SPARSE coding (SC) has long been explored in solving inverse image restoration problems, such as image denoising, image super-resolution, etc. Considering a signal $\boldsymbol{x} \in \mathbb{R}^N$ and an overcomplete dictionary $\boldsymbol{D} \in \mathbb{R}^{N\times M}(N \ll M)$, the SC model aims to approximately represent $\boldsymbol{x}$ using the least amount of atoms in dictionary $\boldsymbol{D}$, i.e., $\boldsymbol{x} \simeq \boldsymbol{D}\boldsymbol{\alpha}$ where $\boldsymbol{\alpha} \in \mathbb{R}^M$ should be sparse. The process to find the sparse $\boldsymbol{\alpha}$ can be formulated as follows:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0, \quad \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2 \leq \epsilon. \tag{1}$$

Here, $\|\cdot\|_0$ is the ℓ_0 pseudo-norm, which indicates the number of nonzero elements in vector $\boldsymbol{\alpha}$. The above optimization problem is NP-hard for unstructured dictionaries. The classical solving methods either use the greedy methods such as orthogonal matching pursuit (OMP) [1], or relax the ℓ_0 norm to ℓ_1 norm which can be solved by basis pursuit [2]. In addition to $\boldsymbol{\alpha}$, another important element in the SC model is dictionary $\boldsymbol{D}$ which contains the basic representation atoms. Many methods have been proposed to learn the dictionary, such as K-SVD [3], MOD [4], and online dictionary learning [5].

To assure sparse representations, the dictionary $\boldsymbol{D}$ should be sufficiently overcomplete. This makes the SC model only appropriate for signals with low dimensions, but may face the curse of dimensionality for image data with high dimensions. To solve this problem, many methods [6], [7] choose to divide the image into overlapping patches and impose the SC model on each patch. The reconstructed patches are then merged together to form the final restored image. However, this treatment ignores the dependencies across patches, which leads to sub-optimal solutions. To overcome this drawback, the convolutional sparse coding (CSC) model [8], [9] has been proposed, which utilizes the shift-invariant convolutional filters to replace the conventional dictionary atoms. Instead of operating on image patches, the CSC model can be directly applied on the whole image. Given an image $\boldsymbol{y}$, the CSC model calculates a set of convolutional sparse maps $\boldsymbol{\alpha}_p$ based on the filters $\boldsymbol{d}_p$ as follows:

$$\underset{\{\boldsymbol{\alpha}_p\}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y} - \sum_{p=1}^{P} \boldsymbol{d}_p * \boldsymbol{\alpha}_p\|_2 + \lambda \sum_p \|\boldsymbol{\alpha}_p\|_1. \tag{2}$$

Here, $\{\boldsymbol{d}_p\}$ is a set of P convolutional filters, $\{\boldsymbol{\alpha}_p\}$ is the corresponding convolutional sparse maps, and λ is a regularization parameter. The size of $\boldsymbol{\alpha}_p$ is the same as that of $\boldsymbol{y}$. Many efficient algorithms have been proposed [10]–[12] to solve the above optimization problem and apply it in image processing tasks. However, despite that CSC model has achieved great success in many tasks like image super-resolution and image fusion, it struggles with image denoising or the tasks with noisy images. The first successful application of CSC model on image denoising is from Sreter *et al.* [13], which unfolds the CSC solving process into a learnable recurrent network. However, it only achieves denoising performance comparable to the patch based K-SVD algorithm. Later, Simon *et al.* [14] exposed the limitations of the CSC model in representing noisy natural images and proposed a CSC-Net through stride convolutions from Bayesian point. The performance of [14] is much better than [13], but still far from the state-of-the-art (SOTA) deep learning based methods. The question is whether the potential of CSC has been fully explored, and whether it can compete with SOTA deep learning based denoising methods? In this letter, we give a positive answer.

The main contribution of this letter is that we propose a multi-scale variation of the CSC model, which provides a new deployment of CSC in image denoising. Based on the multi-scale CSC model, we derive a mathematical multi-scale solution to calculate the convolutional sparse maps at different scales. We then unroll the solution in an order from large to small scale to form a learnable multi-scale CSC network, namely MCSCNet.

Manuscript received March 11, 2022; revised May 1, 2022; accepted May 3, 2022. Date of publication May 13, 2022; date of current version May 26, 2022. This work was supported in part by the NSFC under Grant 62001016 and Grant 62050175 and in part by the Beijing Natural Science Foundation under Grant JQ20020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Panagiotis Petrantonakis. *(Corresponding author: Xin Deng.)*

Xin Deng is with the School of Cyber Science and Technology, Beihang University, Beijing 100190, China (e-mail: cindydeng@buaa.edu.cn).

Jingyi Xu and Mai Xu are with the Department of Electrical Information Engineering, Beihang University, Beijing 100190, China (e-mail: jingyixu@buaa.edu.cn; maixu@buaa.edu.cn).

The software code is available at https://github.com/JingyiXu404/MCSCNet

Digital Object Identifier 10.1109/LSP.2022.3175096

We apply it on image denoising task, and the experimental results show that our MCSCNet not only performs much better than other CSC based methods, but also is comparable to many SOTA deep learning based approaches.

II. Proposed Approach

In this section, we first introduce the multi-scale CSC model for image denoising, and then provide the mathematical solution to it. By unfolding the solution, we propose an interpretable multi-scale CSC network namely MCSCNet to solve image denoising task.

A. Multi-Scale CSC Model

In image denoising task, given a low-quality (LQ) noisy image $\boldsymbol{x}_l \in \mathbb{R}^{m\times n}$, we aim to restore a high-quality (HQ) clean image $\boldsymbol{x}_h \in \mathbb{R}^{m\times n}$ with the least distortion. In this letter, inspired by the multi-scale perception in human visual system (HVS) [15], [16], we build a set of multi-scale convolutional filters to mimic the multi-scale perception in HVS. Specifically, $\boldsymbol{x}_l$ can be decomposed into several scales, and each scale has its own dictionary filters, as follows:

$$\boldsymbol{x}_l = \sum_{s=1}^{S} \boldsymbol{x}_s = \sum_{s=1}^{S}\sum_{t=1}^{T} \boldsymbol{d}_{s,t} * \boldsymbol{z}_{s,t}. \tag{3}$$

Here, $\boldsymbol{x}_s$ represents the image decomposition at scale s, and S denotes the total number of scales. For scale s, we have $\boldsymbol{d}_{s,t} \in \mathbb{R}^{k_s\times k_s}$ which denotes the convolutional dictionary filter with size $k_s \times k_s$, and the number of dictionary filters is T. Note that to mimic the multi-scale perception, the size of dictionary filter is different across scales. Each filter $\boldsymbol{d}_{s,t}$ is convolved with the sparse map $\boldsymbol{z}_{s,t} \in \mathbb{R}^{m\times n}$ to reconstruct the image $\boldsymbol{x}_l$. The symbol * denotes the convolutional operation. Assuming that all dictionary filters are known, the multi-scale CSC problem aims to calculate the sparse maps $\boldsymbol{z}_{s,t}$ through solving the following optimization problem:

$$\underset{\{\boldsymbol{z}_{s,t}\}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{x}_l - \sum_{s=1}^{S}\sum_{t=1}^{T} \boldsymbol{d}_{s,t} * \boldsymbol{z}_{s,t}\|_2 + \lambda \sum_{s=1}^{S}\sum_{t=1}^{T} \|\boldsymbol{z}_{s,t}\|_1. \tag{4}$$

The $\|\cdot\|_1$ is the ℓ_1 norm, and λ is a regularization parameter.

In image denoising task, we typically assume that the HQ and LQ images share the same sparse maps. Thus, the HQ clean image $\boldsymbol{x}_h$ can be obtained as follows:

$$\boldsymbol{x}_h = \sum_{s=1}^{S}\sum_{t=1}^{T} \boldsymbol{g}_{s,t} * \boldsymbol{z}_{s,t}, \tag{5}$$

where $\boldsymbol{g}_{s,t}$ represents the t-th reconstruction filter at scale s for the HQ image.

B. Network Architecture of MCSCNet

In (4), we need to calculate the $\boldsymbol{z}_{s,t}$ of all S scales. We assume that the higher scale refers to finer image details. Since the first scale contains most of the image content, we first fix the $\boldsymbol{z}_{s,t}$ of other scales to update $\boldsymbol{z}_{1,t}$ of the first scale, which can be formulated as follows:

$$\underset{\{\boldsymbol{z}_{1,t}\}}{\operatorname{argmin}} \frac{1}{2}\|\hat{\boldsymbol{x}}_1 - \sum_{t=1}^{T} \boldsymbol{d}_{1,t} * \boldsymbol{z}_{1,t}\|_2 + \lambda \sum_{t=1}^{T} \|\boldsymbol{z}_{1,t}\|_1, \tag{6}$$

where we have $\hat{\boldsymbol{x}}_1 = \boldsymbol{x}_l$ with the expectation that the first scale should be able to reconstruct the original image well. To solve (6), we use the learned convolutional sparse coding (LCSC) algorithm in [13], which leads to the following iterative solution:

$$\boldsymbol{Z}_1^{(j+1)} = S_\lambda\left(\boldsymbol{Z}_1^{(j)} - \boldsymbol{E}_1 * \boldsymbol{D}_1 * \boldsymbol{Z}_1^{(j)} + \boldsymbol{E}_1 * \hat{\boldsymbol{x}}_1\right). \tag{7}$$

Here, $\boldsymbol{Z}_1 \in \mathbb{R}^{m\times n\times T}$ is the stack of the sparse maps $\{\boldsymbol{z}_{1,t}\}_{t=1}^{T}$, $\boldsymbol{D}_1 \in \mathbb{R}^{k_1\times k_1\times T}$ and $\boldsymbol{E}_1 \in \mathbb{R}^{k_1\times k_1\times T}$ are the learnable filters related with $\{\boldsymbol{d}_{1,t}\}_{t=1}^{T}$. S_λ indicates the soft-thresholding operator. Each iteration in (7) forms a LCSC block, and we have in total J iterations to form the sparse map prediction (SMP) module as shown in Fig. 1.

The above equations finish the calculation of the $\boldsymbol{z}_{1,t}$ for the first scale. However, it is not possible for the first scale to reconstruct the image without any missing details. The remained finer details can be restored by the second scale, as follows,

$$\underset{\{\boldsymbol{z}_{2,t}\}}{\operatorname{argmin}} \frac{1}{2}\|\hat{\boldsymbol{x}}_2 - \sum_{t=1}^{T} \boldsymbol{d}_{2,t} * \boldsymbol{z}_{2,t}\|_2 + \lambda \sum_{t=1}^{T} \|\boldsymbol{z}_{2,t}\|_1, \tag{8}$$

where $\hat{\boldsymbol{x}}_2$ represents the remained details after the reconstruction of the first scale, i.e., $\hat{\boldsymbol{x}}_2 = \boldsymbol{x}_l - \sum_{t=1}^{T} \boldsymbol{d}'_{1,t} * \boldsymbol{z}_{1,t}$, with $\boldsymbol{d}'_{1,t}$ denoting the t-th reconstruction filter of the first scale. Equation (8) can be solved by the same method as (6). Finally, for the S scale, the $\boldsymbol{z}_{S,t}$ can be calculated as follows:

$$\underset{\{\boldsymbol{z}_{S,t}\}}{\operatorname{argmin}} \frac{1}{2}\|\hat{\boldsymbol{x}}_S - \sum_{t=1}^{T} \boldsymbol{d}_{S,t} * \boldsymbol{z}_{S,t}\|_2 + \lambda \sum_{t=1}^{T} \|\boldsymbol{z}_{S,t}\|_1, \tag{9}$$

where $\hat{\boldsymbol{x}}_S$ represents the remained details after the reconstruction of all previous scales, i.e., $\hat{\boldsymbol{x}}_S = \boldsymbol{x}_l - \sum_{s=1}^{S-1}\sum_{t=1}^{T} \boldsymbol{d}'_{s,t} * \boldsymbol{z}_{s,t}$, and $\boldsymbol{d}'_{s,t}$ is the t-th reconstruction filter of scale s.

In the actual network deployment, the above solving process is repeated twice for the following reasons. Recall that in (6), we assume the sparse maps of other scales as zeros when calculating the sparse map of the first scale. This is actually contradictory to our multi-scale assumption. Thus, in the second round, with the predicted sparse maps from the first round, the $\hat{\boldsymbol{x}}_1$ in (6) can be updated by $\hat{\boldsymbol{x}}_1 = \boldsymbol{x}_l - \sum_{s=2}^{S}\sum_{t=1}^{T} \boldsymbol{d}'_{s,t} * \boldsymbol{z}_{s,t}$. In this way, the error occurred in the first round can be compensated in the second round, as shown in Fig. 1. Finally, after we obtain all the sparse maps of different scales $\{\boldsymbol{z}_{s,t}\}_{s=1}^{S}$ through the above steps, we can reconstruct the clean image by (5).

III. Experimental Results

A. Experimental Settings

1) Implementation Details: The training images include 432 color images from BSD500 dataset [25]. The testing datasets include BSD68 [25], Urban100 [26], Kodak24 [27], and McMaster [28]. The number of scales is $S = 3$ with the filter size as $k_1 = 7$, $k_2 = 5$ and $k_3 = 3$, respectively. The training patch size is 64×64 and the filter number $T = 64$. Inspired by DCSC [29], the mean absolute error (MAE) loss is used to train our network which can preserve the structures and edges well. The network is initialized via uniform distribution. The Adam optimizer is adopted with default parameters (ρ_1=0.9, ρ_2=0.999, δ=1×10^{-8}), and the initial learning rate as 1×10^{-4}. The soft-thresholding S_λ is used as the activation function.

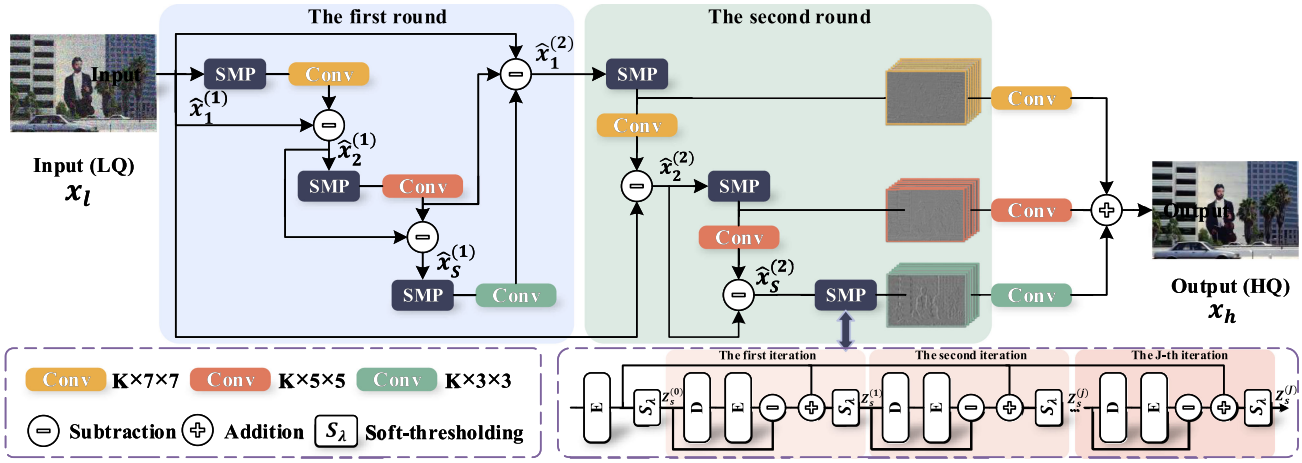

Fig. 1. The network architecture of the proposed MCSCNet. The sparse map prediction (SMP) module is used to predict the sparse map of each scale.

TABLE I
MCSCNET VERSUS OTHER CSC BASED METHODS FOR IMAGE DENOISING ON FOUR DIFFERENT DATASETS

Method	BSD68 dataset								Urban100 dataset							
	$\sigma=15$		$\sigma=25$		$\sigma=50$		$\sigma=75$		$\sigma=15$		$\sigma=25$		$\sigma=50$		$\sigma=75$	
	PSNR↑	SSIM↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM↑	PSNR↑	SSIM ↑	PSNR ↑	SSIM↑	PSNR↑	SSIM↑
CSC [9]	29.13	0.7825	23.87	0.5259	16.89	0.2412	13.48	0.1424	28.63	0.7960	23.72	0.5712	16.99	0.3085	13.59	0.2013
LCSC [13]	33.71	0.9254	30.98	0.8754	27.69	0.7776	25.99	0.7108	32.99	0.9325	30.53	0.8986	27.23	0.8273	25.20	0.7649
CSCNet [14]	33.75	0.9257	31.08	0.8770	27.80	0.7783	26.14	0.7126	33.26	0.9340	30.79	0.9000	27.44	0.8264	25.55	0.7688
MCSCNet	**34.07**	**0.9317**	**31.35**	**0.8859**	**28.09**	**0.7972**	**26.41**	**0.7340**	**33.50**	**0.9385**	**31.14**	**0.9083**	**27.82**	**0.8486**	**26.07**	**0.7977**
Method	Kodak24 dataset								McMaster dataset							
	$\sigma=15$		$\sigma=25$		$\sigma=50$		$\sigma=75$		$\sigma=15$		$\sigma=25$		$\sigma=50$		$\sigma=75$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CSC [9]	29.66	0.7588	23.97	0.4744	16.79	0.2010	13.38	0.1161	30.24	0.7784	24.54	0.4996	17.34	0.2109	13.80	0.1203
LCSC [13]	34.31	0.9162	31.77	0.8697	28.63	0.7840	26.93	0.7274	33.44	0.9049	31.17	0.8633	28.22	0.7868	26.46	0.7321
CSCNet [14]	34.38	0.9163	31.89	0.8706	28.74	0.7824	27.10	0.7256	33.92	0.9089	31.64	0.8696	28.52	0.7898	26.82	0.7367
MCSCNet	**34.75**	**0.9232**	**32.22**	**0.8807**	**29.12**	**0.8036**	**27.43**	**0.7484**	**33.99**	**0.9132**	**31.87**	**0.8783**	**28.90**	**0.8113**	**27.16**	**0.7628**

With the best results shown in bold, and the second bests that are underlined.

2) Comparison Methods: To verify the effectiveness of the proposed MCSCNet, we compare it with three CSC based denoising methods, including CSC [9], LCSC [13] and CSCNet [14]. In addition, we compare with three conventional denoising methods including BM3D [6], WNNM [17] and TNRD [18], and several SOTA deep learning based methods including DnCNN [19], FFDNET [20], LIDIA [21], ADNet [22], FLNet [23] and GroupSC [24]. The commonly used Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) metrics are used to measure the denoising performance.

B. Experimental Results

1) Quantitative Results: Table I presents the color image denoising results of different CSC based methods in four different datasets with noise level as σ=15, 25, 50 and 75, respectively. As shown in Table I, the proposed method consistently outperforms the other CSC based algorithms on all the four datasets in different noise variances. Specifically, the proposed MCSCNet achieves 0.54 dB PSNR higher than LCSC, and 0.32 dB PSNR higher than CSCNet, which are averaged on all the four datasets with different noise levels. These results verify that the multi-scale CSC model significantly advances the development of CSC in image denoisng task. In addition to these CSC based methods, we also compare the proposed MCSCNet with the non-CSC based methods to find its location in the image denoising community. As shown in Table II, despite the weak advantage, our MCSCNet outperforms the SOTA deep learning methods including DnCNN [19], FFDNET [20], LIDIA [21], ADNet [22], FLNet [23], in terms of both PSNR and SSIM. For GroupSC [24], we are on par with it with lower noise levels σ =15 and 25, but better than it with higher noise levels σ =50 and 75.

TABLE II
MCSCNET VERSUS NON-CSC BASED IMAGE DENOISING METHODS ON BSD68

Method	$\sigma=15$		$\sigma=25$		$\sigma=50$		$\sigma=75$	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Noisy image	24.83	0.5728	20.53	0.3945	15.00	0.1994	12.11	0.1128
BM3D [6]	33.25	0.9186	30.18	0.8588	27.38	0.7627	25.74	0.6959
WNNM [17]	30.69	0.8486	28.28	0.7853	24.85	0.6638	22.60	0.5839
TNRD [18]	31.27	0.8757	28.77	0.8055	25.84	0.6888	24.34	0.6184
DnCNN [19]	33.99	0.9297	31.27	0.8834	27.90	0.7891	26.13	0.7196
FFDNet [20]	33.87	0.9290	31.21	0.8821	27.96	0.7887	26.24	0.7224
LIDIA [21]	34.03	0.9298	31.31	0.8838	27.99	0.7887	-	-
ADNet [22]	33.99	0.9298	31.31	0.8843	28.04	0.7936	26.32	0.7290
FLNet [23]	-	-	31.25	-	28.02	-	26.30	-
GroupSC [24]	**34.11**	**0.9322**	**31.44**	**0.8867**	28.05	0.7963	26.12	0.7196
MCSCNet	34.07	0.9317	31.35	0.8859	**28.09**	**0.7972**	**26.41**	**0.7340**

With best results in bold and second bests underlined.

2) Qualitative Results: Fig. 2 visualizes the denoised images of different methods with σ = 50 on the four datasets. As can be seen from this figure, the denoised image by our method

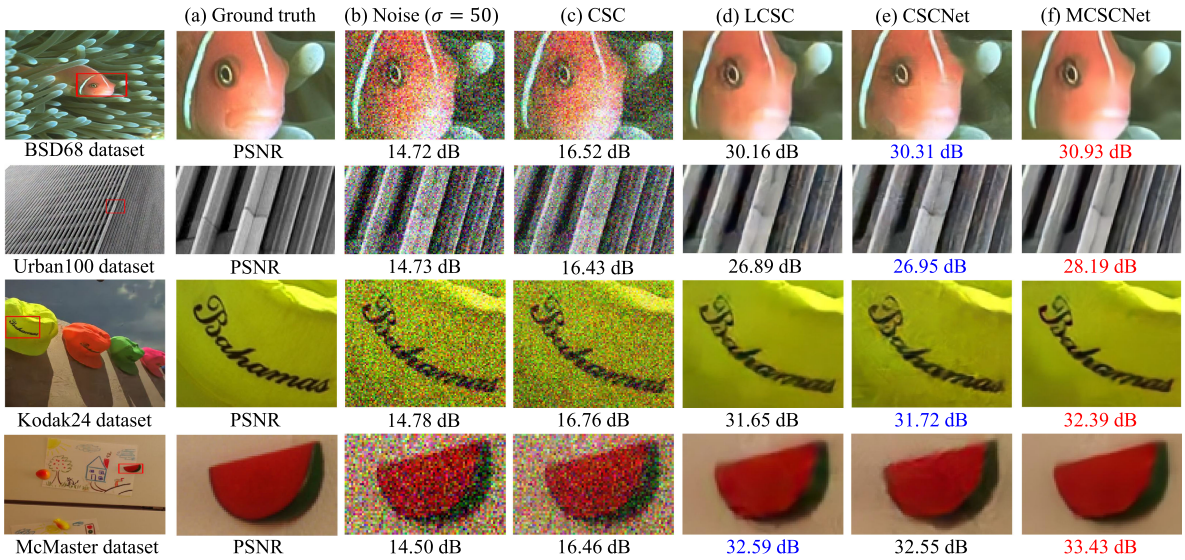

Fig. 2. Visual comparisons of image denoising results on BSD68, Urban100, Kodak24 and McMaster datasets with $\sigma = 50$. The best results are marked in red and the second bests are in blue. (a) the ground truth (b) noisy input image, (c) CSC [9], (d) LCSC [13], (e) CSCNet [14], (f) our MCSCNet.

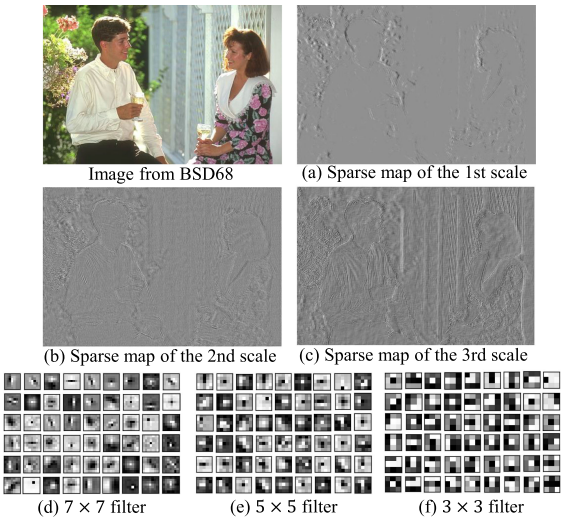

Fig. 3. The sparse maps and reconstruction filters of different scales.

is quite close to the ground-truth, and we provide more clear image edges and texture details than other methods. Fig. 3 visualizes the typical sparse maps and reconstruction filters of each scale. As shown in Fig. 3(a), (b) and (c), the sparse map of the first scale extracts the basic structure and coarse textures, while the sparse maps of the second and third scales extract more delicate features and details. Fig. 3(d), (e) and (f) visualize the corresponding 7×7, 5×5 and 3×3 reconstruction filters of the three scales. As we can see, most of the reconstruction filters have physical meanings, either directional or localized, which verifies the effectiveness of our MCSCNet.

3) Network Parameters and FLOPs: Table III compares the number of network parameters and floating point operations (FLOPs) of our and other deep learning based methods. The FLOPs of all networks are tested with $3 \times 112 \times 112$ image. As we can see from this table, our network has the lowest FLOPs, i.e., 4.18 G, among all comparison methods. Compared to CSCNet, although our network parameters is around 2 times than that of CSCNet, our network FLOPs are more than 190 times lower than it.

TABLE III
NETWORK PARAMETERS AND FLOPS COMPARISON

Methods	DnCNN [19]	CSCNet [14]	ADNet [22]	MCSCNet
Parameters	558.34K	190.75K	521.49K	334.85K
FLOPs	7.02G	841.40G	6.55G	4.18G

TABLE IV
THE EFFECT OF SCALE NUMBER ON BSD68 DATASET

Noise	$\sigma = 15$			$\sigma = 25$			$\sigma = 50$		
Scale	S=1	S=2	S=3	S=1	S=2	S=3	S=1	S=2	S=3
PSNR/dB	33.84	33.97	34.07	31.13	31.29	31.35	27.89	28.02	28.09

C. Ablation Study

Since the proposed network is derived from a multi-scale CSC model, it is necessary to investigate the effect of scale number on the denoising performance. Table IV presents the results for different number of scales with noise variance σ =15, 25 and 50. For $S = 1$, the filter size is $k_1 = 7$. For $S = 2$, the filter sizes are $k_1 = 7$ and $k_2 = 5$. Note that for fair comparison, the network is re-trained for $S = 1$ and $S = 2$ scales. As we can see from Table IV, when the number of scales increases, the denoising performance also improves for different noise levels, which validates the effectiveness of the multi-scale design in our network.

IV. CONCLUSION

In this letter, for the poor performance of convolution sparse coding (CSC) in image denoising, we claim that the CSC is far from reaching its potential limit yet. Thus, we propose a multi-scale variation of CSC model for image denoising, and unfold the multi-scale solution into a learnable multi-scale network namely MCSCNet. The network architecture is lightweight and has good interpretiblity. Experimental results show that the proposed MCSCNet not only significantly advances the performance of CSC based image denoising, but also competes advantageously with SOTA deep learning based methods. This provides a new insight for CSC denoising.

REFERENCES

[1] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[2] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *Soc. Ind. Appl. Math. Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[3] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[4] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1999, pp. 2443–2446.

[5] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 689–696.

[6] 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, Aug. 2007.

[7] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 479–486.

[8] H. Bristow, A. Eriksson, and S. Lucey, "Fast convolutional sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 391–398.

[9] B. Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 301–315, Jan. 2016.

[10] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang, "Convolutional sparse coding for image super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1823–1831.

[11] Y. Liu, X. Chen, R. K. Ward, and Z. J. Wang, "Image fusion with convolutional sparse representation," *IEEE Signal Process. Lett.*, vol. 23, no. 12, pp. 1882–1886, Dec. 2016.

[12] V. Papyan, Y. Romano, and M. Elad, "Convolutional neural networks analyzed via convolutional sparse coding," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 2887–2938, 2017.

[13] H. Sreter and R. Giryes, "Learned convolutional sparse coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 2191–2195.

[14] D. Simon and M. Elad, "Rethinking the CSC model for natural images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, Art. no. 204.

[15] P. Guo, D. Zeng, Y. Tian, S. Liu, H. Liu, and D. Li, "Multi-scale enhancement fusion for underwater sea cucumber images based on human visual system modelling," *Comput. Electron. Agriculture*, vol. 175, 2020, Art. no. 105608.

[16] Shahan C. Nercessian, Karen A. Panetta, and S. S. Agaian, "Non-linear direct multi-scale image enhancement based on the luminance and contrast masking characteristics of the human visual system," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3549–3561, Sep. 2013.

[17] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2862–2869.

[18] 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, Jun. 2017.

[19] 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.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[20] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[21] G. Vaksman, M. Elad, and P. Milanfar, "LIDIA: Lightweight learned image denoising with instance adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 524–525.

[22] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, "Attention-guided CNN for image denoising," *Neural Netw.*, vol. 124, pp. 117–129, 2020.

[23] Y. Guo, A. Davy, G. Facciolo, J.-M. Morel, and Q. Jin, "Fast, nonlocal and neural: A lightweight high quality solution to image denoising," *IEEE Signal Process. Lett.*, vol. 28, pp. 1515–1519, 2021.

[24] B. Lecouat, J. Ponce, and J. Mairal, "Fully trainable and interpretable non-local sparse models for image restoration," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 238–254.

[25] 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 8th Int. Conf. Comput. Vis.*, 2001, pp. 416–423.

[26] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5197–5206.

[27] R. Franzen, "Kodak lossless true color image suite," 1999. [Online]. Available: http://r0k.us/graphics/kodak

[28] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *J. Electron. Imag.*, vol. 20, no. 2, 2011, Art. no. 023016.

[29] X. Fu, Z.-J. Zha, F. Wu, X. Ding, and J. Paisley, "JPEG artifacts reduction via deep convolutional sparse coding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2501–2510.