This is the Pre-Published Version.

The following publication H. Zheng, H. Yong and L. Zhang, "Deep Convolutional Dictionary Learning for Image Denoising," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 630-641 is available at https://doi.org/10.1109/CVPR46437.2021.00069.

Deep Convolutional Dictionary Learning for Image Denoising

Hongyi Zheng[a,b,*] Hongwei Yong[a,b,*] Lei Zhang[a,b,†]
[a]The Hong Kong Polytechnic University [b]DAMO Academy, Alibaba Group
{cshzheng,cshyong,cslzhang}@comp.polyu.edu.hk

Abstract

Inspired by the great success of deep neural networks (DNNs), many unfolding methods have been proposed to integrate traditional image modeling techniques, such as dictionary learning (DicL) and sparse coding, into DNNs for image restoration. However, the performance of such methods remains limited for several reasons. First, the unfolded architectures do not strictly follow the image representation model of DicL and lose the desired physical meaning. Second, handcrafted priors are still used in most unfolding methods without effectively utilizing the learning capability of DNNs. Third, a universal dictionary is learned to represent all images, reducing the model representation flexibility. We propose a novel framework of deep convolutional dictionary learning (DCDicL), which follows the representation model of DicL strictly, learns the priors for both representation coefficients and the dictionaries, and can adaptively adjust the dictionary for each input image based on its content. The effectiveness of our DCDicL method is validated on the image denoising problem. DCDicL demonstrates leading denoising performance in terms of both quantitative metrics (e.g., PSNR, SSIM) and visual quality. In particular, it can reproduce the subtle image structures and textures, which are hard to recover by many existing denoising DNNs. The code is available at: https://github.com/natezhenghy/DCDicL_denoising.

1. Introduction

How to represent an image signal plays a key role in traditional image processing applications [9, 41, 42, 15, 14]. One popular approach is to represent an image patch vector $\mathbf{y}\in\mathbb{R}^m$ as a linear combination of atomic bases, *i.e.*, $\mathbf{y}=\mathbf{D}\mathbf{x}$, where $\mathbf{D}\in\mathbb{R}^{m\times d}$ is the dictionary of atoms, and $\mathbf{x}\in\mathbb{R}^d$ is the representation coefficient vector. In the early stage, cosine functions [4], wavelets [5] and contourlets [11] are commonly used as the dictionary atoms. However, such dictionaries are manually designed under some mathematical constraints and are not flexible enough to represent the complex natural image structures. Later on, researchers turned to learn the dictionary directly from image data, and many dictionary learning (DicL) methods have been developed [30, 56, 71, 64].

The DicL model can be formulated as follows:

$$\min_{\mathbf{D},\mathbf{X}} \tfrac{1}{2}\|\mathbf{D}\mathbf{X}-\mathbf{Y}\|_2^2+\lambda_{\mathbf{X}}\psi(\mathbf{X})+\lambda_{\mathbf{D}}\phi(\mathbf{D}) \quad (1)$$

where $\mathbf{Y}\in\mathbb{R}^{m\times N}$ is a set of N training samples and each column of it is a stretched image patch vector; $\mathbf{X}\in\mathbb{R}^{d\times N}$ is the representation coefficient matrix of $\mathbf{Y}$ over dictionary $\mathbf{D}$; $\psi(\cdot)$ denotes the prior on coefficient $\mathbf{X}$ and $\phi(\cdot)$ denotes the regularization term on $\mathbf{D}$ (*e.g.*, $\|\mathbf{D}\|_2^2$); $\lambda_{\mathbf{X}}$ and $\lambda_{\mathbf{D}}$ are the regularization parameters for $\mathbf{X}$ and $\mathbf{D}$, respectively. The most widely used priors of $\psi(\cdot)$ are sparsity priors, such as $\|\mathbf{X}\|_0$ and $\|\mathbf{X}\|_1$, and the corresponding DicL models are often called Sparse DicL. K-SVD [3, 71] is the most representative Sparse DicL method. It alternatively performs two steps to learn the dictionary: fix $\mathbf{D}$ and perform sparse coding (SC) to compute $\mathbf{X}$, and update $\mathbf{D}$ through singular value decomposition (SVD).

Inspired by K-SVD, many DicL methods have been proposed [36, 65, 14, 71, 27, 46, 45] and successfully used in various image restoration applications, such as denoising [16, 9] and super-resolution [63, 62, 61]. One problem of the patch-based DicL model in Eq. (1) is its lack of shift-invariant property, and convolutional dictionary learning (CDicL) [19] was proposed to address this issue by using the convolution operation to replace the matrix multiplication in signal representation. Specifically, the objective function of CDicL can be written as:

$$\min_{\mathbf{D},\{\mathbf{X}_i\}} \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\tfrac{1}{2}\|\mathbf{D}\circledast\mathbf{X}_i-\mathbf{Y}_i\|_2^2+\lambda_{\mathbf{X}}\psi(\mathbf{X}_i)+\lambda_{\mathbf{D}}\phi(\mathbf{D}) \quad (2)$$

where $\mathbf{D}\circledast\mathbf{X}_i=\sum_{c=1}^{C}\mathbf{D}_c*\mathbf{X}_{i,c}$, $*$ is the 2D convolution operator, and C is the number of channels; $\mathbf{D}=\{\mathbf{D}_c\}_{c=1}^{C}$ is the convolutional dictionary and $\mathbf{D}_c\in\mathbb{R}^{k\times k}$ is the c-th 2D dictionary atom (*i.e.*, filter); $\mathbf{X}_i=\{\mathbf{X}_{i,c}\}_{c=1}^{C}$ is the representation coefficient (also called feature map) of image $\mathbf{Y}_i\in\mathbb{R}^{h\times w}$ and $\mathbf{X}_{i,c}\in\mathbb{R}^{h\times w}$ is the c-th channel of $\mathbf{X}_i$.

*The first two authors contribute equally to this work.
†Corresponding author.
This work is supported by the Hong Kong RGC RIF grant (R5001-18).

In CDicL, the sparse prior is commonly used for the feature map $\mathbf{X}_i$ (*e.g.*, $\|\mathbf{X}_i\|_1$) and convolutional sparse coding (CSC) [8, 58] is used to solve the feature map. CDicL has demonstrated its advantages over patch-based DicL in several image processing tasks [34, 19, 21, 32].

With the rapid development of deep learning (DL) techniques in recent years, many deep neural network (DNN) based image restoration methods have been proposed [67, 69, 22, 13, 12]. Driven by a large amount of training data and the strong learning capacity of DNN, these methods have surpassed traditional image restoration methods, including those DicL based ones, by a large margin. Nonetheless, due to the black-box nature of DNN, there lacks a clear interpretation for its success in image restoration, while DicL has good interpretability. Therefore, researchers have attempted to integrate DicL, SC and DL for both good performance and clear physical meaning. These methods, often called deep unfolding methods, unfold the traditional SC and DicL models through certain algorithms, and parameterize the model by DNN in an end-to-end learning manner. Representative methods include DKSVD [47], Learned-CSC [52], CSCNet [50], DCSC [18], etc.

However, the existing deep unfolding methods usually fail to compete with DL methods for several reasons. First, the unfolded architectures do not strictly follow the original DicL models, which impairs the physical meaning and sacrifices the advantages of DicL. Second, most of them [52, 50, 18] still use the handcrafted priors, *e.g.*, L_1 (sparsity) prior, instead of learning the priors from data, wasting the learning capacity of DNN architectures. Third, they usually learn a universal dictionary for all images, reducing the model's representation capability. In this work, we propose a new unfolding framework, called deep convolutional dictionary learning (DCDicL), which resolves the above issues of previous unfolding methods. The contributions of this paper are summarized as follows:

- DCDicL learns the priors for both dictionary and representation coefficients from the training data, overcoming the disadvantages of handcrafted priors.
- DCDicL learns a specific dictionary for each image, which is adaptive to the image content. This endows DCDicL with more powerful capability for recovering image subtle structures.
- To testify the effectiveness of our framework, we apply DCDicL on the image denoising task. It achieves leading denoising performance over not only previous unfolding methods but also DL methods.

2. Related Works

2.1. Dictionary learning

Dictionary learning (DicL) is an important image modeling and representation learning approach and it has been widely studied in image restoration [63, 16, 19, 21, 32]. DicL aims to optimize a dictionary of atoms for representing the signal with handcrafted priors such as the sparsity prior on representation coefficients. In the seminal work of K-SVD [3, 71], the dictionary is optimized alternatively in two steps. The SC step employs the greedy orthogonal matching pursuit method to estimate the coefficients with L_0 constraint, while the singular value decomposition is used in the second step to update the dictionary. Many methods have been proposed to improve K-SVD [36, 65, 14, 71, 27, 46, 45]. For example, Mairal *et al*. [36] extended K-SVD to color image restoration. Zhang *et al*. [65] used group sparsity to make the learned dictionary more structured. Dong *et al*. [14] introduced the nonlocal self-similarity prior into DicL for image restoration.

DicL is a patch-based image modeling method and it lacks the shift-invariant property. Convolution dictionary learning (CDicL) [19] was proposed to address this issue. It replaces the dictionary atoms with a set of filters and reconstructs the original image by convolutional operation instead of matrix multiplication. The sparsity priors are imposed on the convolution feature maps, which can be solved by CSC [8, 58]. CDicL takes advantage of shift-invariant property and exploits better the image global information, exhibiting better performance than patch-based DicL in various image restoration applications [34, 19, 48, 21, 32].

2.2. Deep learning

The great success of deep learning (DL) in image recognition [31, 51, 23] facilitates its application to image restoration and enhancement tasks. Mao *et al*. [37] proposed a residual encoder-decoder network for image restoration. Dong *et al*. [12] proposed to use a 3-layer convolutional neural network (CNN), called SRCNN, for single image super-resolution (SISR). With the rapid development of deep neural network (DNN) training techniques, in [28] a 20-layer CNN, namely VDSR, was trained, which outperforms significantly traditional SISR methods. Zhang *et al*. [67] proposed the DnCNN model, which is a milestone of image denoising. The FFDNet [69] was developed for fast and flexible image denoising with multiple noise levels. Tai *et al*. [53] proposed the persistent memory network (MemNet) for image restoration. Jia *et al*. [26] proposed FOCNet, which solves a fractional optimal control problem for image denoising. N3Net [40], NLRN [33], RNAN [72] adopt the non-local modules to exploit the nonlocal image prior for noise removal.

Generally speaking, the above DNN models act as an implicit regularizer and learn the image priors from training data, surpassing the handcrafted priors used in traditional methods by a large margin. Nonetheless, most of the DNN based image restoration methods lack good interpretability.

2.3. Deep unfolding

Deep unfolding methods attempt to integrate the merits of model-based SC and DicL methods (*e.g.*, good interpretability) and the merits of DL (*e.g.*, strong learning capability). They unfold certain optimization algorithms, such as iterative shrinkage-threshold [6, 50, 18], alternating direction method of multipliers [7], half-quadratic splitting [2, 66] and primal-dual [1], parameterize the unfolded model, and update the learnable parameters by DNN. For the deep unfolding of DicL, the iterative shrinkage-threshold algorithm (ISTA) is usually used in the unfolding process. The DKSVD [47] model first replaces the L_0 prior in K-SVD by L_1 prior, then unfolds the SC process by ISTA iterations. Finally, it parameterizes the dictionaries by multi-layer perceptron (MLP) modules. The Learned-CSC [52] unfolds the CSC process by ISTA, which can be integrated with DNN. Simon *et al*. [50] improved the Learned-CSC through strided convolution. Fu *et al*. [18] introduced a multi-scale feature extraction module before ISTA unfolding, and applied it to JPEG artifacts removal.

Though improving the interpretability of DL, current deep unfolding methods have some problems and their performances still lag behind DL based methods. For example, the unfolded structures mismatch the original DicL model, fixed priors are adopted without fully utilizing the learning capability of DNN, and the learned dictionary is universal but not image adaptive. We will discuss these issues in Section 3 and present a new framework of deep unfolding.

3. Methodology

3.1. Problems of current deep unfolding methods

Most of the deep unfolding methods for DicL [47] and CDicL [52, 50, 18] assume L_1 prior on coefficient $\mathbf{X}$. Without loss of generality, we focus on the unfolding of CDicL in the following discussion. The objective function is a special case of Eq. (2), which can be written as:

$$\min_{\mathbf{D},\{\mathbf{X}_i\}} \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\|\mathbf{D}\circledast\mathbf{X}_i-\mathbf{Y}_i\|_2^2+\lambda_{\mathbf{X}}\|\mathbf{X}_i\|_1 \tag{3}$$

With a set of N training sample pairs $\{\mathbf{Y}_i,\mathbf{Y}_i^{gt}\}$, where $\mathbf{Y}_i^{gt}$ is the clean ground truth of noisy image $\mathbf{Y}_i$. Eq. (3) can be formulated as a bi-level optimization problem:

$$\min_{\mathbf{D}} \frac{1}{N}\sum_{i=1}^{N}L\big(\mathbf{D}\circledast\mathbf{X}_i,\mathbf{Y}_i^{gt}\big) \tag{4a}$$

$$\text{s.t. } \mathbf{X}_i=\operatorname{argmin}_{\mathbf{X}} \frac{1}{2\sigma_i^2}\|\mathbf{D}\circledast\mathbf{X}-\mathbf{Y}_i\|_2^2+\lambda_{\mathbf{X}}\|\mathbf{X}\|_1 \tag{4b}$$

where $L(\cdot,\cdot)$ measures the loss and σ_i is the noise level of image $\mathbf{Y}_i$. Eq. (4b) can be solved by ISTA [10] iteratively:

$$\mathbf{X}=\mathcal{S}_{\lambda,\eta}\Big(\mathbf{X}-\frac{1}{\eta}\text{rot180}(\mathbf{D})\circledast(\mathbf{D}\circledast\mathbf{X}-\mathbf{Y}_i)\Big) \tag{5}$$

where $\mathcal{S}$ is the shrinkage function, λ and η are hyper-parameters, and $\text{rot180}(\mathbf{D})$ rotates $\mathbf{D}$ by $180°$. Existing deep unfolding methods [47, 52, 50, 18] parameterize $\text{rot180}(\mathbf{D})$ and $\mathbf{D}$ by two DNN modules to build the architecture. In the forward pass, Eq. (4b) is solved to estimate $\mathbf{X}$. In the backward pass, $L(\cdot,\cdot)$ is calculated to update dictionary $\mathbf{D}$ as weight matrices of convolutional layers.

The above unfolding scheme, however, has some inherent problems as listed below:

- First, $\text{rot180}(\mathbf{D})$ and $\mathbf{D}$ are parameterized as two independent Conv modules, violating the mathematical constraint (*i.e.*, $\text{rot180}(\mathbf{D})$ and $\mathbf{D}$ are the rotation of each other) and losing the physical meaning of CDicL.
- Second, the learning capability of DNN is misused. In Eq. (3), given the coefficient $\mathbf{X}$ and the signal $\mathbf{Y}_i$, the dictionary $\mathbf{D}$ can be solved explicitly and there is no need to update them using DNN. On the other hand, the DNN should be used to learn the complex priors of $\mathbf{X}$, whereas the handcrafted sparsity priors (*e.g.*, L_1-sparsity) are used, which is far less effective.
- Third, a universal dictionary $\mathbf{D}$ is parameterized to represent all images in the existing unfolding scheme, which impairs the flexibility of image representation.

The above three problems of deep CDicL unfolding limits its performance in image restoration, lagging behind those standard DNN based methods.

3.2. Deep convolutional dictionary learning

To solve the limitations of current deep CDicL unfolding methods, we propose a new deep unfolding framework, namely DCDicL, which can effectively integrate the advantages of CDicL and DL. Instead of unfolding the objective function in Eq. (3), which enforces handcrafted priors on $\mathbf{X}$ and employs a universal $\mathbf{D}$, we unfold the general objective function in Eq. (2), employ adaptive $\mathbf{D}$ on each image, and learn the deep priors of both $\mathbf{X}$ and $\mathbf{D}$ from data. The learning model of DCDicL can be written as:

$$\min_{\{\mathbf{D}_i,\mathbf{X}_i\}} \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\|\mathbf{D}_i\circledast\mathbf{X}_i-\mathbf{Y}_i\|_2^2+\lambda_{\mathbf{X}}\psi(\mathbf{X}_i)+\lambda_{\mathbf{D}}\phi(\mathbf{D}_i) \tag{6}$$

We rewrite Eq. (6) as the bi-level optimization problem:

$$\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}L\big(\mathbf{D}_i\circledast\mathbf{X}_i,\mathbf{Y}_i^{gt}\big) \tag{7a}$$

$$\text{s.t. } \{\mathbf{D}_i,\mathbf{X}_i\}=\operatorname{argmin}_{\mathbf{D},\mathbf{X}} \frac{1}{2\sigma_i^2}\|\mathbf{D}\circledast\mathbf{X}-\mathbf{Y}_i\|_2^2 \tag{7b}$$

$$+\lambda_{\mathbf{X}}\psi(\mathbf{X})+\lambda_{\mathbf{D}}\phi(\mathbf{D})$$

where θ is the learnable parameters.

The architecture of DCDicL can be derived by unfolding the inner objective in Eq. (7b). For the convenience of expression, we omit the subscript "i" in the following development. To separate the data term and prior term, we introduce two auxiliary variables $\mathbf{X}'$ and $\mathbf{D}'$, and solve the following two objective functions:

$$\min_{\mathbf{X},\mathbf{X}'} \frac{1}{2\sigma^2}\|\mathbf{D}\circledast\mathbf{X}'-\mathbf{Y}\|_2^2+\lambda_{\mathbf{X}}\psi(\mathbf{X}),\ \text{s.t. } \mathbf{X}=\mathbf{X}' \tag{8a}$$

$$\min_{\mathbf{D},\mathbf{D}'} \frac{1}{2\sigma^2}\|\mathbf{D}'\circledast\mathbf{X}-\mathbf{Y}\|_2^2+\lambda_{\mathbf{D}}\phi(\mathbf{D}),\ \text{s.t. } \mathbf{D}=\mathbf{D}' \tag{8b}$$

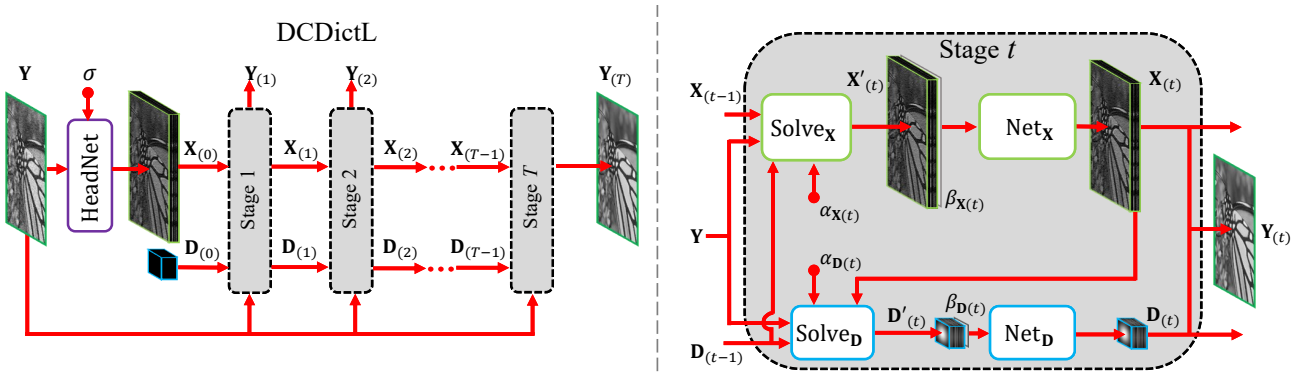


Figure 1: Left: the overall architecture of DCDicL. Right: the architecture of one stage in DCDicL.

According to the Half Quadratic Splitting (HQS) algorithm [24], solving Eq. (8) is equivalent to minimizing the following objective functions:

$$\min_{\mathbf{X},\mathbf{X}'} \frac{1}{2\sigma^2}\|\mathbf{D}\circledast\mathbf{X}'-\mathbf{Y}\|_2^2+\lambda_{\mathbf{X}}\psi(\mathbf{X})+\frac{\mu_{\mathbf{X}}}{2}\|\mathbf{X}-\mathbf{X}'\|_2^2 \quad (9a)$$

$$\min_{\mathbf{D},\mathbf{D}'} \frac{1}{2\sigma^2}\|\mathbf{D}'\circledast\mathbf{X}-\mathbf{Y}\|_2^2+\lambda_{\mathbf{D}}\phi(\mathbf{D})+\frac{\mu_{\mathbf{D}}}{2}\|\mathbf{D}-\mathbf{D}'\|_2^2 \quad (9b)$$

where $\mu_{\mathbf{D}}$ and $\mu_{\mathbf{X}}$ are the penalty parameters for $\mathbf{D}$ and $\mathbf{X}$, respectively. When $\mu_{\mathbf{D}}$ and $\mu_{\mathbf{X}}$ are large enough, the constraint in Eq. (8) can be met.

Eq. (9) can be solved iteratively. In the t-th iteration (stage), $\mathbf{X}'_{(t)}$, $\mathbf{X}_{(t)}$, $\mathbf{D}'_{(t)}$, $\mathbf{D}_{(t)}$ are solved as follows:

$$\begin{aligned}\mathbf{X}'_{(t)} &= \text{Solve}_{\mathbf{X}}\left(\mathbf{Y},\mathbf{D}_{(t\text{-}1)},\mathbf{X}_{(t\text{-}1)},\alpha_{\mathbf{X}}\right) \quad (10a)\\ &= \text{argmin}_{\mathbf{X}^\star}\ \tfrac{1}{2}\left\|\mathbf{D}_{(t\text{-}1)}\circledast\mathbf{X}^\star-\mathbf{Y}\right\|_2^2+\tfrac{\alpha_{\mathbf{X}}}{2}\left\|\mathbf{X}^\star-\mathbf{X}_{(t\text{-}1)}\right\|_2^2\\ \mathbf{X}_{(t)} &= \text{Net}_{\mathbf{X}}\left(\mathbf{X}'_{(t)},\beta_{\mathbf{X}}\right) \quad (10b)\\ &= \text{argmin}_{\mathbf{X}^\star}\ \psi(\mathbf{X}^\star)+\tfrac{\beta_{\mathbf{X}}}{2}\left\|\mathbf{X}'_{(t)}-\mathbf{X}^\star\right\|_2^2\\ \mathbf{D}'_{(t)} &= \text{Solve}_{\mathbf{D}}\left(\mathbf{Y},\mathbf{D}_{(t\text{-}1)},\mathbf{X}_{(t)},\alpha_{\mathbf{D}}\right) \quad (10c)\\ &= \text{argmin}_{\mathbf{D}^\star}\ \tfrac{1}{2}\left\|\mathbf{D}^\star\circledast\mathbf{X}_{(t)}-\mathbf{Y}\right\|_2^2+\tfrac{\alpha_{\mathbf{D}}}{2}\left\|\mathbf{D}^\star-\mathbf{D}_{(t\text{-}1)}\right\|_2^2\\ \mathbf{D}_{(t)} &= \text{Net}_{\mathbf{D}}\left(\mathbf{D}'_{(t)},\beta_{\mathbf{D}}\right) \quad (10d)\\ &= \text{argmin}_{\mathbf{D}^\star}\ \phi(\mathbf{D}^\star)+\tfrac{\beta_{\mathbf{D}}}{2}\left\|\mathbf{D}'_{(t)}-\mathbf{D}^\star\right\|_2^2\end{aligned}$$

where $\{\alpha_{\mathbf{X}},\alpha_{\mathbf{D}},\beta_{\mathbf{X}},\beta_{\mathbf{D}}\}=\left\{\mu_{\mathbf{X}}\sigma^2,\mu_{\mathbf{D}}\sigma^2,\frac{\mu_{\mathbf{X}}}{\lambda_{\mathbf{X}}},\frac{\mu_{\mathbf{D}}}{\lambda_{\mathbf{D}}}\right\}$. For data term subproblems in Eqs. (10a) and (10c), we have closed-form fast solutions, which will be presented in Sections 3.3 and 3.4. For prior term subproblems in Eqs. (10b) and (10d), we learn two DNNs to solve them. For hyper-parameters $\{\alpha_{\mathbf{X}},\alpha_{\mathbf{D}},\beta_{\mathbf{X}},\beta_{\mathbf{D}}\}$, we use a simple network (HypaNet) with input σ to predict them for each stage.

The unfolding process of DCDicL is depicted in Algorithm 1. We use a simple network (HeadNet) to get an initial estimation for $\mathbf{X}_0$ and simply use the zero initialization for $\mathbf{D}_0$. The overall architecture of DCDicL is illustrated in Fig. 1. During the forward pass, DCDicL solves Eq. (7b)

Algorithm 1: Unfolding process of DCDicL

Input : Noisy image $\mathbf{Y}$, number of stages T
Output: Predicted image $\mathbf{Y}^{pred}$
$\mathbf{X}_0=\text{HeadNet}(\mathbf{Y},\sigma)$, $\mathbf{D}_0=\mathbf{0}$;
for $t=1,...,T$ **do**
  $\{\alpha_{\mathbf{X}(t)},\alpha_{\mathbf{D}(t)},\beta_{\mathbf{X}(t)},\beta_{\mathbf{D}(t)}\}=\text{HyperNet}_{(t)}(\sigma)$;
  $\mathbf{X}'_{(t)}=\text{Solve}_{\mathbf{X}}\left(\mathbf{Y},\mathbf{D}_{(t\text{-}1)},\mathbf{X}_{(t\text{-}1)},\alpha_{\mathbf{X}(t)}\right)$;
  $\mathbf{X}_{(t)}=\text{Net}_{\mathbf{X}}\left(\mathbf{X}'_{(t)},\beta_{\mathbf{X}(t)}\right)$;
  $\mathbf{D}'_{(t)}=\text{Solve}_{\mathbf{D}}\left(\mathbf{Y},\mathbf{D}_{(t\text{-}1)},\mathbf{X}_{(t)},\alpha_{\mathbf{D}}(t)\right)$;
  $\mathbf{D}_{(t)}=\text{Net}_{\mathbf{D}}\left(\mathbf{D}'_{(t)},\beta_{\mathbf{D}(t)}\right)$;
$\mathbf{Y}^{pred}=\mathbf{D}_{(T)}\circledast\mathbf{X}_{(T)}$

by addressing the four subproblems in Eq. (10) iteratively, and obtains the final estimations of $\mathbf{X}$ and $\mathbf{D}$ for each image adaptively. While during the backward pass, DCDicL calculates $L(\mathbf{D}\circledast\mathbf{X},\mathbf{Y}^{gt})$ and solves Eq. (7a), via which the image priors are updated from training data. The details of the network design will be presented in Section 3.5.

3.3. Solving X

For the convenience of expression, we omit the subscript "(t)" in the following development. The closed-form solution of $\mathbf{X}'$ can be derived by solving Eq. (10a). According to [8], Eq. (10a) can be efficiently solved using Fast Fourier Transform (FFT). Denote by $\mathcal{F}(\cdot)$ the 2D FFT, and let $\boldsymbol{\mathcal{D}}=\mathcal{F}(\mathbf{D})$, $\boldsymbol{\mathcal{X}}^\star=\mathcal{F}(\mathbf{X}^\star)$, $\boldsymbol{\mathcal{Y}}=\mathcal{F}(\mathbf{Y})$ and $\boldsymbol{\mathcal{X}}=\mathcal{F}(\mathbf{X})$. Taking the derivative of Eq. (10a) w.r.t. $\mathbf{X}^\star$, letting the derivative be zero and using the Sherman-Morrison formula [49], we have the following closed-form solution:

$$\mathbf{X}'=\frac{1}{\alpha_{\mathbf{X}}}\mathcal{F}^{-1}\left\{\boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{D}}\circ\left(\frac{(\bar{\boldsymbol{\mathcal{D}}}\odot\boldsymbol{\mathcal{Z}})}{\alpha_{\mathbf{X}}+(\bar{\boldsymbol{\mathcal{D}}}\odot\boldsymbol{\mathcal{D}})}\uparrow_C\right)\right\} \quad (11)$$

where $\boldsymbol{\mathcal{Z}}=\boldsymbol{\mathcal{D}}\circ(\boldsymbol{\mathcal{Y}}\uparrow_C)+\alpha_{\mathbf{X}}\boldsymbol{\mathcal{X}}$, $\mathcal{F}^{-1}(\cdot)$ denotes the inverse FFT, $\bar{\boldsymbol{\mathcal{D}}}$ denotes complex conjugate of $\boldsymbol{\mathcal{D}}$, $\circ$ is the Hadamard product, $\mathbf{A}\odot\mathbf{B}=\sum_{c=1}^{C}\mathbf{A}_c\circ\mathbf{B}_c$, $\mathbf{A}\uparrow_C$ expands the channel dimension of $\mathbf{A}$ to C, and $\div$ is the Hadamard

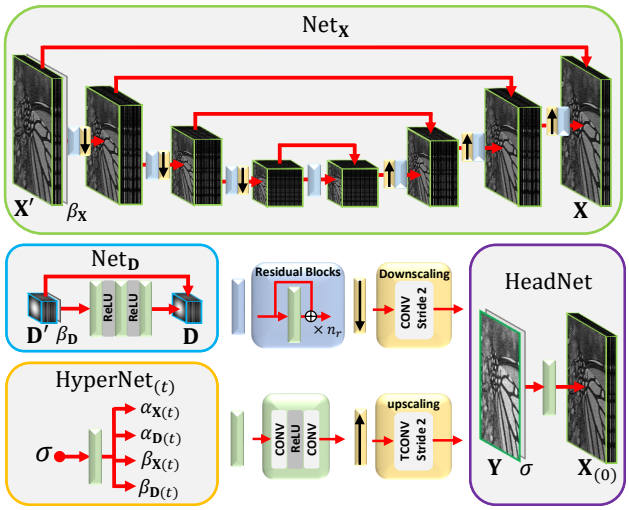


Figure 2: Sub-network architectures of DCDicL.

division. The detailed derivation can be found in the **supplementary file**.

3.4. Solving D

As in practice the dimension of $\mathbf{D}'$ is much lower than that of $\mathbf{X}$, the system is overdetermined and can be solved by least squares method. To utilize the modern least squares solvers, we unfold Eq. (10c) from the form of convolution into the form of matrix multiplication. The details can be found in the **supplementary file**. Here, we use $\mathfrak{X}$ and $\mathfrak{Y}$ to denote the unfolded results of original matrices $\mathbf{X}$ and $\mathbf{Y}$. The objective function in Eq. (10c) can be re-written as:

$$\text{argmin}_{\mathbf{d}^\star} \ \tfrac{1}{2}\|\mathfrak{X}\mathbf{d}-\mathfrak{Y}\|_2^2+\tfrac{\alpha_{\mathbf{D}}}{2}\|\mathbf{d}^\star-\mathbf{d}\|_2^2 \qquad (12)$$

where $\mathbf{d}^\star=\text{vec}(\mathbf{D}^\star)$, $\mathbf{d}=\text{vec}(\mathbf{D})$, and $\text{vec}(\cdot)$ is the vectorization operator.

By taking the derivative of the above objective function w.r.t. $\mathbf{d}^\star$ and letting the derivative be zero, we can obtain the closed-form solution of $\mathbf{D}'$:

$$\mathbf{D}'=\text{vec}^{-1}\left\{\left(\mathfrak{X}^T\mathfrak{X}+\alpha_{\mathbf{D}}\mathbf{I}\right)^{-1}\left(\mathfrak{X}^T\mathfrak{Y}+\alpha_{\mathbf{D}}\mathbf{d}\right)\right\} \qquad (13)$$

where $\text{vec}^{-1}(\cdot)$ reverses the vectorization. Eq. (13) can be efficiently solved by modern least square solvers such as LU solver provided by PyTorch. Notice that unfolding $\mathbf{X}$ to $\mathfrak{X}$ would increase the memory overhead by k^2, which is not desirable in practice. Fortunately, we can efficiently compute $\mathfrak{X}^T\mathfrak{X}$ and $\mathfrak{X}^T\mathfrak{Y}$ from $\mathbf{X}$ and $\mathbf{Y}$ without explicitly storing $\mathfrak{X}$. The details are in the **supplementary file**.

3.5. Network design

As shown in Fig. 1, our proposed DCDicL framework has four sub-networks, including HeadNet, NetX, NetD and HypaNet, whose architectures are illustrated in Fig. 2.

HeadNet takes the noisy image $\mathbf{Y}$ and noise level σ as input to initialize coefficient $\mathbf{X}_0$. It consists of 2 Conv layers (64 channels each layer) with ReLU activation.

NetX learns the prior on coefficients $\mathbf{X}$. It acts as an implicit regularizer as those end-to-end networks in DL based denoising methods [67, 69, 26, 40]. We adopt the U-Net [43] architecture for its effectiveness in image restoration tasks [66, 73, 57]. Specifically, our NetX consists of 7 blocks. The first 3 blocks down-sample the feature maps through strided convolution, and the last 3 blocks up-sample the feature maps by transposed convolution. Each block consists of several residual units, while each residual unit consists of 2 Conv layers with ReLU activation and a skip connection. The Conv layers in the first 4 blocks have 64, 128, 256 and 512 channels, respectively. The selection of the number of residual units is discussed in section 4.2.

NetD learns the prior on dictionary $\mathbf{D}$. As $\mathbf{D}$ has a much smaller spatial size than $\mathbf{X}$, a shallower network is enough to provide sufficient receptive field and learning capability. Our NetD consists of 6 Conv layers with ReLU activation, and there is no ReLU after the last Conv layer. Each Conv layer has 16 channels, and there is a skip connection between the first and the last units.

HypaNet takes noise level σ as inputs and predicts the hyperparameters for each stage. It consists of 2 Conv layers (kernel size 1) and a SoftPlus layer, ensuring all hyperparameters are positive.

4. Experiments

4.1. Training details

We follow [39] to use the combination of WED [35], DIV2K training set [55] and BSD400 [38, 67], for training. The noisy image $\mathbf{Y}$ is obtained by adding additive white Gaussian noise of standard deviantion σ on the ground truth image $\mathbf{Y}^{gt}$. Patches of size 128×128 are randomly cropped from $\mathbf{Y}^{gt}$ and $\mathbf{Y}$ image pairs for training.

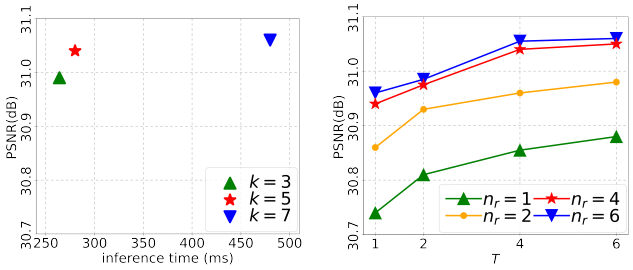
L_1 loss is used as the loss function $L(\cdot,\cdot)$ on the output of each stage. As in previous multi-stage learning work [60], we set the weight of the loss imposed on the last stage as 1, and set the weight as $\frac{1}{T-1}$ for all the other $T-1$ stages. The Adam optimizer [29] is used for updating the learnable parameters. The batch size is 32 and we train it for $1e6$ iterations. The learning rate starts from $1e$-4 and decays by a factor of 0.5 for every $2e5$ iterations. In order to speed up and stabilize the training, we first train a 1-stage model and reload its weights into the T-stage model for fine-tuning. (The selection of T is discussed in Section 4.2.) All the T stages share the same parameters and we train a shared model for all noise levels, which are set to $\{15,25,50\}$ as in [67, 26, 69]. The number of atoms in $\mathbf{D}$ is determined by the number of feature maps in NetX (*i.e.*, 64), while the spatial size of each atom is discussed in Section 4.2.

Table 1: Grayscale image denoising results in PSNR(dB)/SSIM(%). "-" means that the result is not available.

Datasets	σ	BM3D	WNNM	DnCNN	N3Net	NLRN	RNAN	FOCNet	IRCNN	FFDNet	DKSVD	CSCNet	DCDicL
Set12	15	32.37/89.52	32.70/89.82	32.86/90.24	-	33.16/90.70	-	33.07/-	32.76/90.06	32.75/90.24	32.61/-	31.46/89.96	**33.34/91.15**
	25	29.97/85.04	30.28/85.57	30.44/86.17	30.55/86.68	30.80/86.89	-	30.73/-	30.37/85.98	30.43/86.31	30.22/-	28.88/85.79	**31.03/87.48**
	50	26.72/76.76	27.05/77.75	27.18/78.28	27.43/79.31	27.64/79.80	27.70/80.09	27.68/-	27.12/78.04	27.32/78.99	26.13/-	25.56/78.31	**28.00/81.22**
BSD68	15	31.07/87.17	31.37/87.66	31.73/89.07	-	31.88/89.32	-	31.83/-	31.63/88.82	31.63/89.02	31.48/88.35	31.57/88.79	**31.95/89.57**
	25	28.57/80.13	28.83/80.87	29.23/82.79	29.30/83.77	29.41/83.31	-	29.38/-	29.15/82.48	29.19/82.88	28.96/81.71	29.11/82.57	**29.52/83.79**
	50	25.60/68.64	25.87/69.82	26.23/71.89	26.39/73.21	26.47/72.98	26.48/73.03	26.50/-	26.19/71.69	26.29/72.39	25.97/70.35	26.24/72.02	**26.63/73.95**
Urban100	15	32.35/92.20	32.97/92.71	32.64/92.46	-	33.45/93.54	-	33.15/-	32.46/92.36	32.40/92.65	-	32.31/92.34	**33.59/93.88**
	25	29.70/87.77	30.39/88.85	29.95/87.81	30.19/89.26	30.94/90.18	-	30.64/-	29.80/88.31	29.90/89.79	-	29.76/88.21	**31.30/91.08**
	50	25.95/77.91	26.83/80.47	26.26/78.56	26.82/81.48	27.49/82.79	27.65/83.32	27.40/-	26.22/79.18	26.50/80.47	-	26.23/79.69	**28.24/85.49**

Table 2: Color image denoising results in PSNR(dB)/SSIM(%). "-" means that the result is not available.

Datasets	σ	CBM3D	RNAN	RPCNN	BRDNet	DSNet	DnCNN	IRCNN	FFDNet	CSCNet	DCDicL
CBSD68	15	33.50/92.15	-	-	34.10/92.91	33.91/-	33.89/92.90	33.87/92.85	33.87/92.90	33.83/-	**34.36/93.48**
	25	30.69/86.72	-	31.24/88.80	31.43/88.47	31.28/-	31.23/88.30	31.18/88.24	31.21/88.21	31.18/-	**31.75/89.30**
	50	27.36/76.26	28.27/80.18	28.06/79.90	28.16/79.42	28.05/-	27.92/78.96	27.88/78.98	27.96/78.87	28.00/-	**28.57/81.07**
Kodak24	15	34.26/91.47	-	-	34.88/92.49	34.63/-	34.48/92.09	34.69/92.09	34.63/92.24	-	**35.38/93.00**
	25	31.67/86.70	-	32.34/88.40	32.41/88.56	32.16/-	32.03/87.75	32.15/87.79	32.13/87.91	-	**32.97/89.28**
	50	28.44/77.60	29.58/81.18	29.25/80.50	29.22/80.40	29.05/-	28.85/79.17	28.94/79.43	28.98/79.52	-	**29.96/82.19**
McMaster	15	34.03/91.14	-	-	35.08/92.69	34.67/-	33.44/90.35	34.58/91,95	34.66/92.16	-	**35.50/93.35**
	25	31.63/86.99	-	32.33/89.00	32.75/89.43	32.40/-	31.51/86.94	32.18/88.18	32.35/88.61	-	**33.26/90.48**
	50	28.48/79.11	29.72/83.29	29.35/82.60	29.52/82.65	29.28/-	28.61/79.86	28.93/80.69	29.18/81.49	-	**30.22/84.94**
Urban100	15	33.93/94.08	-	-	34.42/94.62	-	32.98/93.14	33.78/93.14	33.83/94.18	-	**34.90/95.11**
	25	31.36/90.92	-	31.81/91.90	31.99/91.94	-	30.81/90.15	31.20/90.88	31.40/91.20	-	**32.77/93.00**
	50	27.93/84.04	29.08/87.03	28.62/86.20	28.56/85.77	-	27.59/83.31	27.70/83.96	28.05/84.76	-	**29.88/88.84**



(a) (b)

Figure 3: Ablation studies on (a) k; (b) n_r and T.

4.2. Ablation studies

In DCDicL, the hyperparameters can be end-to-end learned by HypaNet. In this section, we perform ablation studies on the selection of spatial size k of $\mathbf{D}$, the no. of stages T, and the no. of residual units, denoted by n_r. We perform the ablation study on the Set12 dataset with noise level σ=25. The results are illustrated in Fig. 3.

Selection of k. We select k among $\{3,5,7\}$. It can be seen that the PSNR index rises with the increase of k; however, the improvement becomes minor when k=7. On the other hand, the inference time boosts when k=7. To balance the performance and efficiency, we set k=5 in the experiment.

Selection of T and n_r. The depth of our DCDicL model can be adjusted by 2 key factors. The first factor is the no. of unrolling stages T. The second factor is the size of sub-networks. Since NetD, HeadNet and HypaNet are much smaller than NetX, we can simply control the model size by adjusting the no. of residual units n_r, which consists of 2 Conv layers, in each block of NetX.

It can be seen that the PSNR index rises with the increases of both n_r and T. However, the improvement becomes minor when n_r=6 and T=6. To guarantee the efficiency, we set n_r=4 and T=4 in the experiment.

4.3. Comparison with state of the arts

In this section, we compare DCDicL with state of the art image denoising methods. Since some competing methods only provide the codes or results for grayscale images or color images, we compare DCDicL with different methods on different datasets. Since the sizes of testing images are often different from that of training patches (*i.e.*, 128×128), the regularization strength on $\mathbf{D}$ in Eq. (10c) should be different. Hence, in testing we scale $\alpha_{\mathbf{D}}$ by $\frac{h^{test}\times w^{test}}{128\times 128}$ to normalize the regularization strength, where h^{test} and w^{test} are the spatial sizes of testing image.

For grayscale image denoising, we adopt 3 widely used testing datasets, including Set12, BSD68 [44] and Urban100 [25], in the experiments. We compare the proposed DCDicL with representative model-based methods (*i.e.*, BM3D [9], WNNM [20]), DL based methods (*i.e.*, DnCNN [67], N3Net [40], NLRN [33], RNAN [72], FOCNet [26], IRCNN [68], FFDNet [69]), and deep unfolding methods (DKSVD [47], CSCNet [50]). The experimental results are shown in Table 1. For color image de-

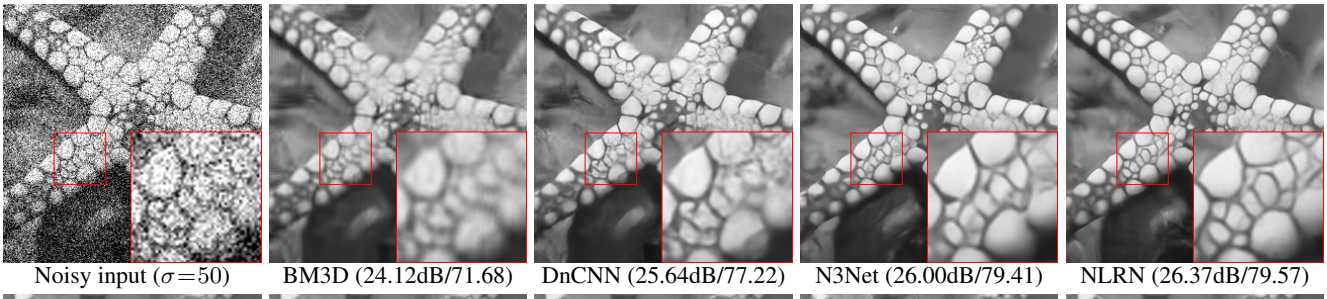


Figure 4: Denoising results on image 04 in Set12.

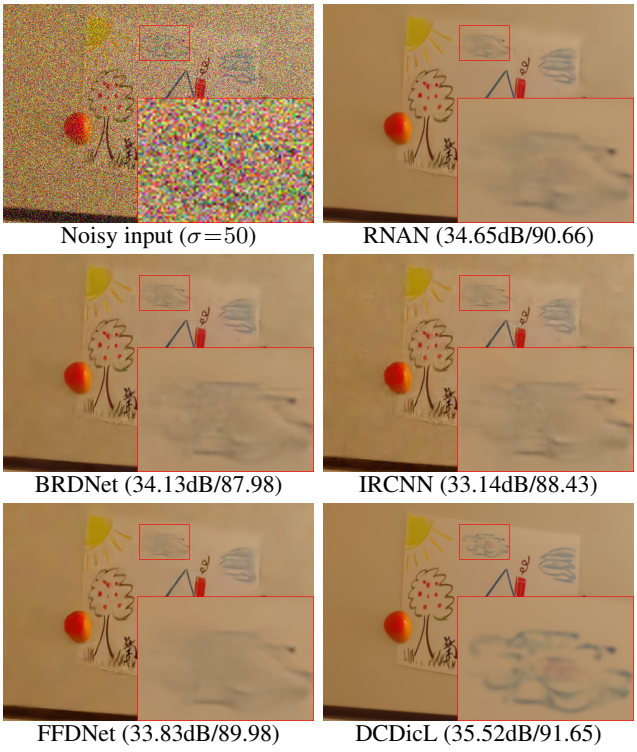


Figure 5: Denoising results on image 13 in McMaster.

noising, four widely used color image datasets, including CBSD68 [44], Kodak24 [17], McMaster [70] and Urban100 are used here. We compare DCDicL with model-based method (*i.e.*, CBM3D), DL based methods (*i.e.*, DnCNN, IRCNN, FFDNet, RNAN, RPCNN [59], BRDNet [54], DSNet [39]), and deep unfolding method (*i.e.*, CSCNet). The experimental results are shown in Table 2.

It can be seen that DCDicL achieves the best PSNR and

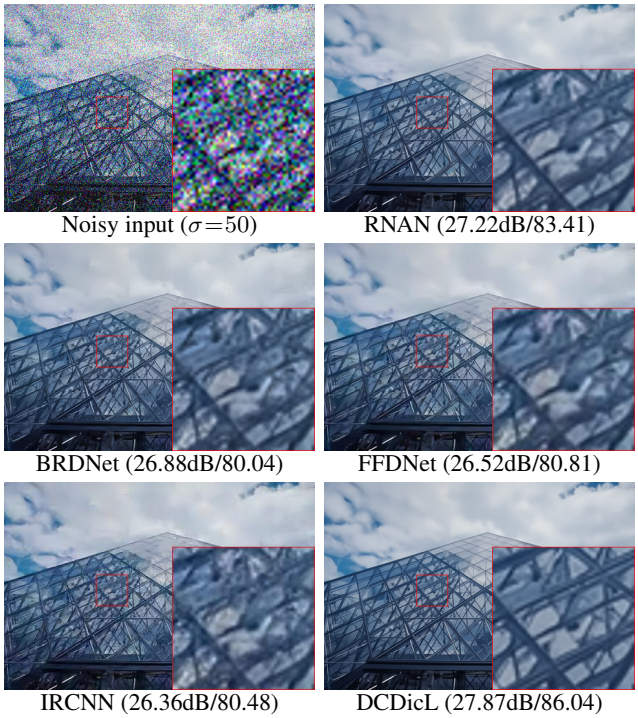


Figure 6: Denoising results on image 223061 in CBSD68.

SSIM performance under all experiment settings. Specifically, DCDicL outperforms previous deep unfolding methods DKSVD and CSCNet from 0.38dB to 2.44dB for different settings. Although we trained a shared DCDicL model for all noise levels, it still outperforms DL based methods DnCNN, N3Net, NLRN, RNAN, FOCNet, PRCNN, BRDNet and DSNet, which train a separated model for each noise level. RNAN, N3Net and NLRN exploit non-local similarity and achieve better performance than other DL

based methods, but fail to compete with DCDicL.

DCDicL performs extraordinary well on the Urban100 dataset, whose images contain lots of fine-scale repetitive structures and textures. It surpasses the second best method by 0.59dB in grayscale image denoising and 0.8dB in color image denoising even when $\sigma=50$. This is because, by solving the dictionary $\mathbf{D}$ in Eq. (10c), DCDicL perceives the global information from input image $\mathbf{Y}$ and exploits the image self-similarity adaptively and effectively. Figs. 4, 5 and 6 show the denoising results on images from Set12, McMaster and CSBD68. It can be seen that DCDicL restores many subtle edges and textures that cannot be restored by other competing methods. More visualizations can be found in the **supplementary file**.

We further compare the inference time of DCDicL and competing methods. All experiments are done on Set12 ($\sigma=50$) with a GTX 1080Ti GPU. As can be seen from Fig. 7, DCDicL is slower than DnCNN, FFDNet, IRCNN and DKSVD, but achieves much higher denoising performance. RNAN, N3Net and NLRN adopt complex non-local modules for improving the performance, but still fail to compete with DCDicL in terms of both denoising performance and inference speed. Overall, we can conclude that DCDicL provides a good solution in terms of both effectiveness and efficiency.

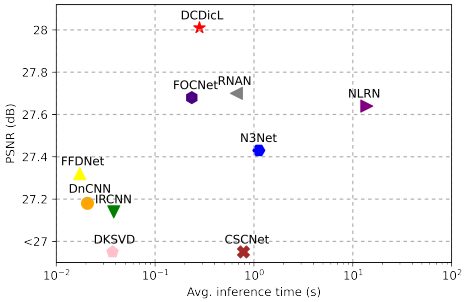


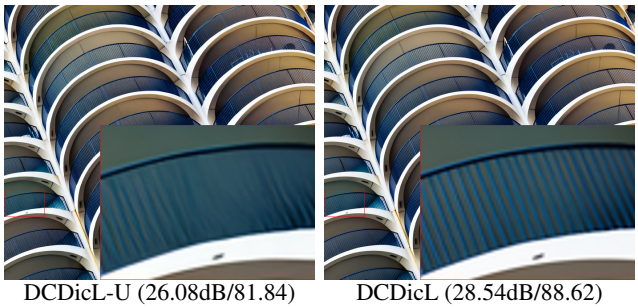
Figure 7: Inference time *vs*. PSNR for different methods.

4.4. Analysis on adaptive dictionaries

To demonstrate the effectiveness of the adaptive dictionaries learned by DCDicL, we replace the NetD in DCDicL with a universal dictionary (*i.e.*, weight matrix in a Conv layer) to train a new (vanilla U-Net) model, called DCDicL-U. We compare DCDicL and DCDicL-U on color image denoising. The denoising results (PSNR/SSIM) are shown in Table 3, and Fig. 8 shows an example. One can see that the adaptive dictionaries learned by NetD improve the denoising performance significantly. DCDicL recovers many image fine textures, which are lost by the universal dictionary. Fig. 9 visualizes the dictionaries learned on two images. It can be seen that DCDicL can adaptively adjust the dictionary according to the content of input image. More visualizations on the dictionaries and denoising results can be found in the **supplementary file**.

Table 3: Results (PSNR/SSIM) of DCDicL-U and DCDicL.

Datasets	CBSD68			Kodak24			McMaster			Urban100		
σ	15	25	50	15	25	50	15	25	50	15	25	50
DCDicL-U	34.23	31.63	28.45	35.20	32.80	29.78	35.24	33.01	29.99	34.58	32.35	29.34
	93.41	90.22	80.98	92.87	89.10	81.94	93.08	90.11	84.35	94.81	92.58	87.87
DCDicL	34.35	31.74	28.57	35.35	32.95	29.95	35.46	33.22	30.21	34.61	32.67	29.82
	93.48	89.29	81.09	92.96	89.25	82.19	93.29	90.44	84.92	94.88	92.90	88.73



DCDicL-U (26.08dB/81.84) DCDicL (28.54dB/88.62)

Figure 8: Denoising results by DCDicL-U and DCDicL.

Figure 9: Adaptive dictionaries learned on two images.

5. Conclusion

We proposed a new deep unfolding method, namely deep convolutional dictionary learning (DCDicL), and validated its effectiveness on image denoising. Following the mathematical modeling of dictionary learning strictly, DCDicL learns the priors on $\mathbf{X}$ and $\mathbf{D}$ from training data, and learns a specific dictionary for each image. The dictionary is not only adaptive to image content but also perceives image global information, endowing DCDicL with strong capability to recover subtle image structures even under severe noise. Extensive experiments on benchmark datasets demonstrated that DCDicL surpasses previous deep unfolding and deep learning based methods in terms of both quantitative metrics and visual quality. DCDicL provides new insight on deep image modeling, and can be extended to more image restoration tasks.

References

[1] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

[2] Manya V Afonso, José M Bioucas-Dias, and Mário AT Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE transactions on image processing*, 19(9):2345–2356, 2010.

[3] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[4] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

[5] Marc Antonini, Michel Barlaud, Pierre Mathieu, and Ingrid Daubechies. Image coding using wavelet transform. *IEEE Transactions on image processing*, 1(2):205–220, 1992.

[6] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[7] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[8] Hilton Bristow, Anders Eriksson, and Simon Lucey. Fast convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 391–398, 2013.

[9] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

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

[11] Minh N Do and Martin Vetterli. Contourlets. In *Studies in computational mathematics*, volume 10, pages 83–105. Elsevier, 2003.

[12] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014.

[13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[14] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE transactions on Image Processing*, 22(4):1620–1630, 2012.

[15] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on image processing*, 20(7):1838–1857, 2011.

[16] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[17] Rich Franzen. Kodak lossless true color image suite. *source: http://r0k. us/graphics/kodak*, 4(2), 1999.

[18] Xueyang Fu, Zheng-Jun Zha, Feng Wu, Xinghao Ding, and John Paisley. Jpeg artifacts reduction via deep convolutional sparse coding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2501–2510, 2019.

[19] Cristina Garcia-Cardona and Brendt Wohlberg. Convolutional dictionary learning: A comparative review and new algorithms. *IEEE Transactions on Computational Imaging*, 4(3):366–381, 2018.

[20] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014.

[21] Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xiangchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1823–1831, 2015.

[22] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1712–1722, 2019.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] Ran He, Wei-Shi Zheng, Tieniu Tan, and Zhenan Sun. Half-quadratic-based iterative minimization for robust sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):261–275, 2013.

[25] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015.

[26] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Lei Zhang. Focnet: A fractional optimal control network for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6054–6063, 2019.

[27] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2651–2664, 2013.

[28] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[30] Kenneth Kreutz-Delgado, Joseph F Murray, Bhaskar D Rao, Kjersti Engan, Te-Won Lee, and Terrence J Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2):349–396, 2003.

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[32] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. Video rain streak removal by multiscale convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6644–6653, 2018.

[33] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems*, pages 1673–1682, 2018.

[34] Yu Liu, Xun Chen, Rabab K Ward, and Z Jane Wang. Image fusion with convolutional sparse representation. *IEEE signal processing letters*, 23(12):1882–1886, 2016.

[35] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2016.

[36] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2007.

[37] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*, pages 2802–2810, 2016.

[38] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001.

[39] Yali Peng, Lu Zhang, Shigang Liu, Xiaojun Wu, Yu Zhang, and Xili Wang. Dilated residual networks with symmetric skip connection for image denoising. *Neurocomputing*, 345:67–76, 2019.

[40] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *Advances in Neural Information Processing Systems*, pages 1087–1098, 2018.

[41] Mark D Plumbley, Thomas Blumensath, Laurent Daudet, Rémi Gribonval, and Mike E Davies. Sparse representations in audio and music: from coding to source separation. *Proceedings of the IEEE*, 98(6):995–1005, 2009.

[42] Saiprasad Ravishankar and Yoram Bresler. Mr image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE transactions on medical imaging*, 30(5):1028–1041, 2010.

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[44] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 860–867. IEEE, 2005.

[45] Ron Rubinstein, Tomer Peleg, and Michael Elad. Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model. *IEEE Transactions on Signal Processing*, 61(3):661–677, 2012.

[46] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. Technical report, Computer Science Department, Technion, 2008.

[47] Meyer Scetbon, Michael Elad, and Peyman Milanfar. Deep k-svd denoising. *arXiv preprint arXiv:1909.13164*, 2019.

[48] Ana Serrano, Felix Heide, Diego Gutierrez, Gordon Wetzstein, and Belen Masia. Convolutional sparse coding for high dynamic range imaging. In *Computer Graphics Forum*, volume 35, pages 153–163. Wiley Online Library, 2016.

[49] Jack Sherman and Winifred J Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950.

[50] Dror Simon and Michael Elad. Rethinking the csc model for natural images. In *Advances in Neural Information Processing Systems*, pages 2274–2284, 2019.

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[52] Hillel Sreter and Raja Giryes. Learned convolutional sparse coding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2191–2195. IEEE, 2018.

[53] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017.

[54] Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 121:461–473, 2020.

[55] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[56] Ivana Tosic and Pascal Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, 2011.

[57] GM Venkatesh, YG Naresh, Suzanne Little, and Noel E O'Connor. A deep residual architecture for skin lesion segmentation. In *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, pages 277–284. Springer, 2018.

[58] Brendt Wohlberg. Efficient convolutional sparse coding. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7173–7177. IEEE, 2014.

[59] Zhihao Xia and Ayan Chakrabarti. Identifying recurring patterns with deep neural networks for natural image denoising. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2426–2434, 2020.

[60] Qi Xie, Minghao Zhou, Qian Zhao, Deyu Meng, Wangmeng Zuo, and Zongben Xu. Multispectral and hyperspectral image fusion by ms/hs fusion net. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1585–1594, 2019.

[61] Jianchao Yang, Zhaowen Wang, Zhe Lin, Scott Cohen, and Thomas Huang. Coupled dictionary training for image super-resolution. *IEEE transactions on image processing*, 21(8):3467–3478, 2012.

[62] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[63] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[64] Meng Yang, Lei Zhang, Xiangchu Feng, and David Zhang. Fisher discrimination dictionary learning for sparse representation. In *2011 International Conference on Computer Vision*, pages 543–550. IEEE, 2011.

[65] Jian Zhang, Debin Zhao, and Wen Gao. Group-based sparse representation for image restoration. *IEEE Transactions on Image Processing*, 23(8):3336–3351, 2014.

[66] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3217–3226, 2020.

[67] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[68] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.

[69] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.

[70] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic imaging*, 20(2):023016, 2011.

[71] Qiang Zhang and Baoxin Li. Discriminative k-svd for dictionary learning in face recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2691–2698. IEEE, 2010.

[72] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082*, 2019.

[73] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.