A DEEP DICTIONARY MODEL FOR IMAGE SUPER-RESOLUTION

Jun-Jie Huang and Pier Luigi Dragotti

Communications and Signal Processing Group (CSP), Imperial College London, UK

ABSTRACT

Inspired by the recent success of deep neural network architectures and the recent effort to develop multi-layer sparse models, we propose a novel deep dictionary learning architecture which is optimized to address a specific regression task known as single image super-resolution. Contrary to other multi-layer dictionaries, our architecture contains $L-1$ analysis dictionaries to extract high-level features and one synthesis dictionary which is designed to optimize the regression task. We propose a variation of an existing method to learn the analysis dictionaries and we update them without the need to use a back-propagation approach. Results on image super-resolution are satisfactory.

***Index Terms*—** Sparse representation, dictionary learning, sparse dictionary, deep learning

1. INTRODUCTION

Deep neural networks (DNNs) have achieved remarkable performance in a wide range of computer vision and image processing applications. DNN is a cascade of multiple layers each characterized by a linear step followed by a point-like non-linearity.

Some recent works have tried to provide insights into the working of DNNs. Bruna and Mallat [1] proposed a scattering convolutional network by replacing the learned filters with wavelet-like transforms. The scattering convolutional network provides features which are translation and rotation invariant. Zeiler and Fergus [2] proposed a deconvolution technique to visualize the intermediate feature layers of a convolutional neural network (CNN) trained for image classification. The filters in the first layer are Gabor like, and the deeper layer feature maps tend to be active only for certain objects. By extending the theory of sparsity, Papyan *et al.* [3] proposed to analyze CNN using multiple layers of convolutional sparse coding (ML-CSC). Theoretical guarantees were also presented.

Some of the sparse representation literature shows connections between dictionary learning and DNNs. The ML-CSC model [3, 4] makes an effort to understand and find a new way of learning DNNs from a dictionary learning point of view. Rubinstein and Elad [5] proposed an analysis-synthesis thresholding framework for image deblurring which consists of an analysis dictionary, point-wise hard thresholding functions and a synthesis dictionary. A synthesis dictionary $\boldsymbol{D}$ is usually a "fat" matrix. The input signal $\boldsymbol{x}$ is assumed to be a linear combination of a few columns of $\boldsymbol{D}$, i.e. $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\varphi}$ where $\|\boldsymbol{\varphi}\|_0 \leq T$. An analysis dictionary $\boldsymbol{\Omega}$ is a "tall" matrix and its rows represent atoms. The analyzed signal refers to $\boldsymbol{\Omega x}$ and $\|\boldsymbol{\Omega x}\|_0 \leq T$. It assumes that $\boldsymbol{x}$ lies on a low dimensional subspace which is the orthogonal complement to the subspace defined by the row atoms corresponding to the zero coefficients of $\boldsymbol{\Omega x}$. The analysis-synthesis thresholding model restores an input signal $\boldsymbol{x}$ through $\boldsymbol{D}\mathcal{S}_\lambda(\boldsymbol{\Omega x})$ where $\mathcal{S}_\lambda(\cdot)$ is a point-wise non-linear function. This can be interpreted as a two-layer DNN.

The double-sparsity model [6, 7] proposes to learn a sparse dictionary $\boldsymbol{A}$ with sparse dictionary atoms over a base dictionary $\boldsymbol{\Phi}$. The effective dictionary $\boldsymbol{\Phi A}$ is more efficient and adaptive and enables learning large dictionary from high-dimensional data. The ML-CSC model has L layers of synthesis dictionaries where the first dictionary $\boldsymbol{D}_1$ is a non-sparse matrix while the following dictionaries $\boldsymbol{D}_2$, ..., $\boldsymbol{D}_L$ are sparse. An input signal $\boldsymbol{x}$ can be expressed as $\boldsymbol{x} = \boldsymbol{D}_1\boldsymbol{D}_2...\boldsymbol{D}_L\boldsymbol{\gamma}$ where $\boldsymbol{\gamma}$ is the sparse vector. This can be considered as an extension of the double-sparsity model.

Most works for analyzing DNNs focus on classification tasks, while there are less papers which focus on regression problems. In this paper, image super-resolution is selected as a sample application to analyze DNNs for regression tasks. Motivated by the analysis-synthesis thresholding framework [5] and the double-sparsity model [6], we propose a deep dictionary model for image super-resolution with L layers of dictionaries. Both the synthesis model and the analysis model are important building blocks for our deep dictionary model. The first $L-1$ dictionaries are analysis dictionaries and the L^{th} layer dictionary is a synthesis one. Point-wise soft-thresholding is performed after each analysis block. The forward pass of this model shares a high similarity with that of DNNs. Therefore our aim is to gain insights into the working of DNNs for regression tasks through this deep dictionary model.

The rest of the paper is organized as follows: Section 2 gives an introduction about image super-resolution. Section 3 introduces our deep dictionary model. Section 4 presents simulation results and Section 5 draws conclusions.

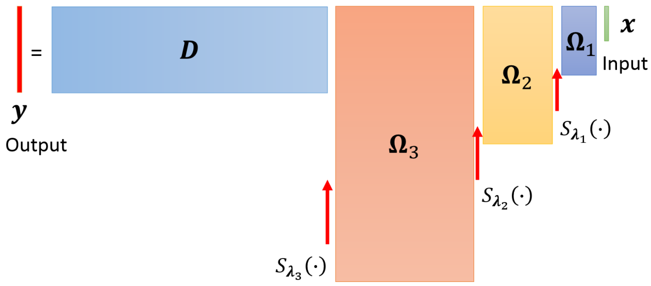


Fig. 1: A 4-layer deep dictionary model for image super-resolution. There are 3 analysis dictionaries $\{\boldsymbol{\Omega}_i\}_{i=1}^{3}$, 1 synthesis dictionary $\boldsymbol{D}$, and 3 point-wise soft-thresholding operators $\{\mathcal{S}_{\boldsymbol{\lambda}_i}(\cdot)\}_{i=1}^{3}$.

2. IMAGE SUPER-RESOLUTION

Image super-resolution (SR) is a classic problem in signal processing. Given a low-resolution (LR) image, a SR algorithm aims to restore a high-resolution (HR) image with sharp edges and rich textures. This is an ill-posed problem. Most of recent SR algorithms are learning-based and they learn correspondences between LR images and HR images from an external training dataset.

Various models have been proposed for image super-resolution. Sparse coding based algorithms [8, 9] assume that there is a common sparse code shared between a LR patch and its corresponding HR patch over a coupled synthesis dictionary pair. For an input LR patch, sparse pursuit is performed over the LR dictionary to find a sparse representation. The HR prediction is achieved by multiplying this sparse code with the HR dictionary. The anchored neighbor regression (ANR) method [10, 11] performs K-means clustering to the LR patches and assigns each LR cluster with a linear regression model to map the input LR patch to its HR version. This combination of classification and regression strategy is effective and fast. Random forest and decision tree based approaches [12, 13] further improve the SR performance and run time efficiency. Deep learning based methods [14, 15, 16] have also been applied for resolution enhancement. Instead of breaking the image into small image patches, these methods perform SR over the whole image using convolutional neural network. The advantage of CNN for SR is that a wider range of information can be incorporated for prediction and there is no explicit patch averaging during reconstruction.

3. PROPOSED METHOD

Training LR-HR patch pairs $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ with $\boldsymbol{x}_i \in \mathbb{R}^{m^2}$ and $\boldsymbol{y}_i \in \mathbb{R}^{(s\times m)^2}$ are extracted from LR images and their corresponding HR images, where the size of the LR patch is $m \times m$ and the HR patch size is $(s \times m) \times (s \times m)$ with s being the up-scaling factor. To gain illumination invariance property, the mean of each patch has been removed. By grouping the training vectors into matrix, we have $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N] \in \mathbb{R}^{m^2\times N}$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N] \in \mathbb{R}^{(s\times m)^2\times N}$.

In this paper, we propose to learn a deep dictionary model for image super-resolution. There are L layers of dictionaries and $L-1$ non-linear operations. The dictionaries between layer 1 and layer $L-1$ are treated as analysis dictionaries $\{\boldsymbol{\Omega}_i\}_{i=1}^{L-1}$ [17] and are used to extract complex features from the data while the dictionary at layer L is treated as a synthesis dictionary $\boldsymbol{D}$ and is designed to optimize the regression task at hand. Let us define the size of $\boldsymbol{\Omega}_1$ as $d_1 \times m^2$, the size of $\boldsymbol{\Omega}_i$ as $d_i \times d_{i-1}$ for $2 \le i < L$ and the size of $\boldsymbol{D}$ as $(s\times m)^2 \times d_{L-1}$ where $m^2 \le d_1 \le d_2 \le ... \le d_{L-1}$ and $(s \times m)^2 \le d_{L-1}$. The non-linear operation is point-wise soft-thresholding $\{\mathcal{S}_{\boldsymbol{\lambda}_i}(\cdot)\}_{i=1}^{L-1}$ where $\boldsymbol{\lambda}_i$ is the threshold. Figure 1 shows an example of the deep dictionary model with $L = 4$. The analysis dictionaries are learned using analysis dictionary learning algorithm in an unsupervised manner from the LR training patches, while the synthesis dictionary is learned in a supervised manner. After all the dictionaries have been obtained, they are updated in a backward fashion.

During testing, overlapped LR patches are extracted from the input LR image and vectorized into column vectors. The mean value of each LR patch is removed. The HR prediction of a LR patch $\boldsymbol{x}$ is expressed as:

$$\boldsymbol{y} = \boldsymbol{D}\mathcal{S}_{\boldsymbol{\lambda}_{L-1}}\left(\boldsymbol{\Omega}_{L-1}\mathcal{S}_{\boldsymbol{\lambda}_{L-2}}\left(...\boldsymbol{\Omega}_2\mathcal{S}_{\boldsymbol{\lambda}_1}\left(\boldsymbol{\Omega}_1\boldsymbol{x}\right)...\right)\right). \quad (1)$$

The mean value of $\boldsymbol{x}$ is then added back to $\boldsymbol{y}$. With the estimated HR patches, the HR image is reconstructed by patch overlapping.

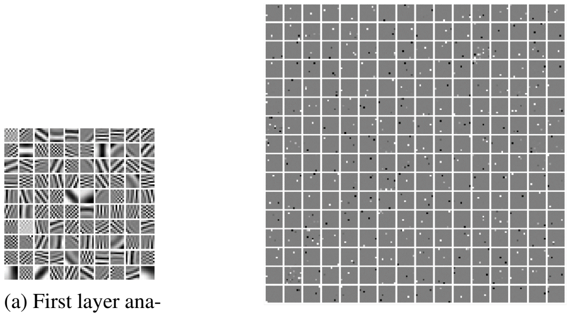
3.1. ANALYSIS DICTIONARY LEARNING

Analysis operator learning [17, 18, 19] aims to learn an analysis dictionary $\boldsymbol{\Omega}$ which is able to sparsify the analyzed signal $\boldsymbol{\Omega x}$. The geometric analysis operator learning (GOAL) method [18] is selected as the analysis learning algorithm for our deep dictionary model due to its fast learning speed and good performance in image reconstruction. The objective function to be minimized in GOAL is expressed as:

$$\boldsymbol{\Omega} = \arg \min_{\boldsymbol{\mathcal{O}}^T \in \mathrm{OB}(n,k)} f(\boldsymbol{\mathcal{O}}, \boldsymbol{S}), \quad (2)$$

where $f(\boldsymbol{\mathcal{O}}, \boldsymbol{S}) = J(\boldsymbol{\mathcal{O}S}) + \kappa h(\boldsymbol{\mathcal{O}}) + \upsilon r(\boldsymbol{\mathcal{O}})$, $\boldsymbol{S}$ is the matrix containing all the training data as column vectors, $\mathrm{OB}(n, k)$ is the set of matrices with size $n \times k$ in oblique manifold [20], $J(\cdot)$ is a sparsity promoting term (for example, l_p norm) to find an $\boldsymbol{\Omega}$ which sparsifies $\boldsymbol{\Omega s}$, $h(\cdot)$ is a full rank constraint term, $r(\cdot)$ penalizes linear dependent row atoms, and κ and υ are the corresponding weighting parameters for $h(\cdot)$ and $r(\cdot)$, respectively.

The objective function is iteratively minimized using a geometric conjugate gradient method. The obtained analysis



(a) First layer analysis dictionary.

(b) Second layer analysis dictionary.

Fig. 2: An example of the learned analysis dictionaries.

dictionary is with unit norm row atoms and has full column rank. For the detailed description of GOAL we refer to [18].

By replacing the training data $\boldsymbol{S}$ in Eqn. (2) with the LR training data $\boldsymbol{X}$, we obtain the analysis dictionary of layer one. Figure 2 (a) shows an example of the learned first layer analysis dictionary. We can see that the dictionary atoms in the learned dictionary $\boldsymbol{\Omega}_1$ are Gabor like filters representing edges in different directions and with different scales. There is a high similarity between the atoms in $\boldsymbol{\Omega}_1$ and the filters learned in the first layer of a DNN, for example, AlexNet [2].

Let us define $\boldsymbol{Z}_1 = \boldsymbol{\Omega}_1 \boldsymbol{X}$ as the data after applying $\boldsymbol{\Omega}_1$ to $\boldsymbol{X}$ and $\boldsymbol{A}_1 = \mathcal{S}_{\boldsymbol{\lambda}_1}(\boldsymbol{Z}_1)$. The histogram of the i^{th} row of $\boldsymbol{Z}_1$ has a shape which can be modeled as a combination of a Laplacian distribution and a spike near zero. Let us denote the mean and standard deviation of the Laplacian distribution of $\boldsymbol{Z}_1(i,:)$ as μ_i and $\sigma_i = \frac{\sum_{j=1}^{N} |\boldsymbol{Z}_1(i,j) - \mu_i|}{N}$. If $p \in [0,1]$ is defined as the proportion of data belonging to the Laplacian distribution that will be zeroed after the soft-thresholding, the threshold for the i^{th} row atom can then be expressed as:

$$\lambda_1(i) = -\sigma_i \log(1-p) + \mu_i. \tag{3}$$

Under the unit norm constraint on $\boldsymbol{\Omega}_1$, the magnitude of the responses from different atoms differ greatly. In order to have an equal mean absolute response for every atom, the analysis dictionary is weighted as follows:

$$\boldsymbol{\Omega}_1 = \mathrm{diag}(1/\boldsymbol{\sigma})\boldsymbol{\Omega}_1, \tag{4}$$

where $\boldsymbol{\sigma} = [\sigma_1, ..., \sigma_{s_1}]$.

The threshold $\lambda_1(i)$ should also be weighted by $1/\sigma_i$, and so $\lambda_1(i) = -\log(1-p) + \mu_i/\sigma_i$.

Given $\boldsymbol{\Omega}_1$ and $\mathcal{S}_{\boldsymbol{\lambda}_1}(\cdot)$, a new analysis dictionary $\boldsymbol{\Omega}_2$ is learned to encode higher level structures. As $\boldsymbol{A}_1$ is a sparse matrix with many zeros, we would like $\boldsymbol{\Omega}_2$ to be a sparse dictionary which could capture structures in $\boldsymbol{A}_1$. The objective function in Eqn. (2) however does not promote sparse dictionary atoms. Therefore the original GOAL algorithm would generate $\boldsymbol{\Omega}_2$ with sparse atoms as well as noisy atoms if applied with $\boldsymbol{S} = \boldsymbol{A}_1$. In order to have sparse dictionary atoms, a sparse constraint over the dictionary is imposed:

$$\boldsymbol{\Omega} = \arg \min_{\boldsymbol{\mathcal{O}}^T \in \mathrm{OB}(n,k)} f(\boldsymbol{\mathcal{O}}, \boldsymbol{S}) + \tau J(\boldsymbol{\mathcal{O}}), \tag{5}$$

where τ is the weighting parameter and $J(\boldsymbol{\mathcal{O}})$ is the sparse constraint on $\boldsymbol{\mathcal{O}}$.

By setting $\boldsymbol{A}_1$ as the training data for the second layer analysis dictionary learning, the geometric conjugate gradient method is applied to iteratively minimize the objective function in Eqn. (5). With this additional sparse constraint on the analysis dictionary, the resultant second layer dictionary $\boldsymbol{\Omega}_2$ is sparse with sparse row atoms as shown in Figure 2 (b). The row atoms in the effective dictionary $\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1$ can be considered as a weighted combination of the row atoms in $\boldsymbol{\Omega}_1$. This leads to more complex patterns than those in $\boldsymbol{\Omega}_1$.

Similar to the first layer, we define $\boldsymbol{Z}_l = \boldsymbol{\Omega}_l \boldsymbol{A}_{l-1}$ as the analyzed signal of layer l and $\boldsymbol{A}_l = \mathcal{S}_{\boldsymbol{\lambda}_l}(\boldsymbol{Z}_l)$ is the sparsified signal for $2 \leq l < L$. The learned analysis dictionary is reweighted by the inverse of the standard deviations and the threshold values are determined as in Eqn. (3). The output of the previous layer $\boldsymbol{A}_{l-1}$ is the input for the next layer analysis dictionary learning. The learned dictionary $\boldsymbol{\Omega}_l$ is also a sparse dictionary. The overall effective dictionary can be represented as $\boldsymbol{\Omega}_{\text{eff}} = \boldsymbol{\Omega}_{L-1}\boldsymbol{\Omega}_{L-2}...\boldsymbol{\Omega}_1$. The row atoms of $\boldsymbol{\Omega}_{\text{eff}}$ can still be considered as linear combinations of the row atoms in $\boldsymbol{\Omega}_1$where the weights are determined by the sparse dictionaries $\boldsymbol{\Omega}_{L-1}$, ..., $\boldsymbol{\Omega}_2$. More complex filters emerged in the effective dictionary through a deeper level of dictionaries.

3.2. SYNTHESIS DICTIONARY LEARNING

With the learned analysis dictionaries, the synthesis dictionary $\boldsymbol{D}$ which is to map $\boldsymbol{A}_{L-1}$ to the HR patches can be obtained in a supervised manner. The synthesis dictionary is learned using least squares:

$$\boldsymbol{D} = \boldsymbol{Y}\boldsymbol{A}_{L-1}^T \left(\boldsymbol{A}_{L-1}\boldsymbol{A}_{L-1}^T\right)^{-1}. \tag{6}$$

3.3. DICTIONARY UPDATE

As the analysis dictionaries are learned in an unsupervised way, the learned deep dictionary may not be appropriate for image super-resolution and there could be a high prediction error. We need a way to propagate the prediction error backwards so that the dictionaries can be updated accordingly. Taylor *et al.* [21] proposed an ADMM approach to train neural networks which does not involve gradients computation. We adopt a similar approach to update the learned dictionaries. The dictionary update is performed backwards: $\boldsymbol{A}_{L-1}$, $\boldsymbol{Z}_{L-1}$, $\boldsymbol{\Omega}_{L-1}$, $\boldsymbol{A}_{L-2}$, $\boldsymbol{Z}_{L-2}$, $\boldsymbol{\Omega}_{L-2}$..., $\boldsymbol{\Omega}_1$ are updated sequentially.

Let us define $\boldsymbol{Z}_L = \boldsymbol{Y}$, $\boldsymbol{\Omega}_L = \boldsymbol{D}$ and $\boldsymbol{A}_0 = \boldsymbol{X}$. The output $\boldsymbol{A}_l$ for $1 \leq l \leq L-1$ is updated by minimizing the objective function below:

$$\boldsymbol{A}_l = \arg\min_{\boldsymbol{A}} \gamma_1 ||\boldsymbol{Z}_{l+1} - \boldsymbol{\Omega}_{l+1}\boldsymbol{A}||_F^2 + \gamma_2 ||\boldsymbol{A} - \mathcal{S}_{\boldsymbol{\lambda}_l}(\boldsymbol{Z}_l)||_F^2, \tag{7}$$

Algorithm 1 Dictionary Update

Input: Training LR-HR data pair $\boldsymbol{X}$ and $\boldsymbol{Y}$, initial dictionaries $\{\boldsymbol{\Omega}_i\}_{i=1}^{L-1}$ and $\boldsymbol{D}$, and a validation set.

Output: Updated dictionaries $\{\boldsymbol{\Omega}_i\}_{i=1}^{L-1}$ and $\boldsymbol{D}$.

```
do
for i = L − 1, L − 2, ..., 1 do
    Update A_i.
    Update Z_i .
    Update Ω_i.
end for
Update D.
Test on validation set.
while the validation error converged.
```

where γ_1 and γ_2 are weighting parameters.

The objective function balances fidelity between the previous and the current layer. In this way we update $\boldsymbol{A}_l$ without departing excessively from its original shape. There is a close form solution for Eqn. (7):

$$\boldsymbol{A}_l = \left(\gamma_1 \boldsymbol{\Omega}_{l+1}^T \boldsymbol{\Omega}_{l+1} + \gamma_2 \boldsymbol{I}\right)^{-1} \left(\gamma_1 \boldsymbol{\Omega}_{l+1}^T \boldsymbol{Z}_{l+1} + \gamma_2 \mathcal{S}_{\boldsymbol{\lambda}_l}(\boldsymbol{Z}_l)\right). \tag{8}$$

The analyzed signal at layer l can be updated as:

$$\boldsymbol{Z}_l = \arg\min_{\boldsymbol{Z}} \gamma_1 ||\boldsymbol{Z} - \boldsymbol{\Omega}_l \boldsymbol{A}_{l-1}||_F^2 + \gamma_2 ||\boldsymbol{A}_l - \mathcal{S}_{\boldsymbol{\lambda}_l}(\boldsymbol{Z})||_F^2. \tag{9}$$

Taking the derivative with respect to $\boldsymbol{Z}$ and setting it to zero yields:

$$\gamma_1(\boldsymbol{Z} - \boldsymbol{\Omega}_l \boldsymbol{A}_{l-1}) + \gamma_2(\mathcal{S}_{\boldsymbol{\lambda}_l}(\boldsymbol{Z}) - \boldsymbol{A}_l)\frac{\partial \mathcal{S}_{\boldsymbol{\lambda}_l}(\boldsymbol{Z})}{\partial \boldsymbol{Z}} = 0. \tag{10}$$

Since the non-linearity is applied point-wise, the optimization over elements of $\boldsymbol{Z}$ can be separated. Since the soft-thresholding is piece-wise linear, Eqn. (9) can be optimized over the 3 linear pieces of the soft-threshold and close form solutions can be obtained.

With the updated $\boldsymbol{Z}_l$ and the original $\boldsymbol{A}_{l-1}$, the analysis dictionary $\boldsymbol{\Omega}_l$ can be updated using least squares:

$$\boldsymbol{\Omega}_l = \boldsymbol{Z}_l \boldsymbol{A}_{l-1}^T \left(\boldsymbol{A}_{l-1} \boldsymbol{A}_{l-1}^T\right)^{-1}. \tag{11}$$

With the updated analysis dictionaries, the synthesis dictionary is re-estimated as in Eqn. (6). The dictionary update scheme is repeated until the prediction error on the validation set has converged. The dictionary update algorithm is presented in **Algorithm 1**.

4. SIMULATION RESULTS

In this section, simulation results are presented. The up-scaling factor s is set to 2. All the LR images are obtained by down-sampling the original HR image by a factor of 2. The size of the LR patch and HR patch is set to be 8×8, and 16×16, respectively. The standard 91 training images [9] are applied for dictionary learning and updating. *Set 5* [9] is used to evaluate the SR quality.

	Bi-cubic	Zeyde's [8]	DD w/o Update	DD w/ Update
Baby	37.02	37.72	37.43	38.21
Bird	36.84	38.96	38.27	39.12
Butterfly	27.43	28.86	29.14	29.17
Head	34.82	35.25	35.05	35.53
Woman	32.19	33.52	33.57	33.85
Average	33.66	34.86	34.69	35.18

Table 1: PSNR (dB) by different image SR methods evaluated on *Set 5* [9].

The deep dictionary model is set to have $L = 4$ layers. The dictionary size for $\boldsymbol{\Omega}_1$, ..., $\boldsymbol{\Omega}_3$ and $\boldsymbol{D}$ is set to 100×64, 256×100, 1024×256, and 256×1024, respectively. The parameters of GOAL for learning $\boldsymbol{\Omega}_1$ is $\kappa = 40$ and $\upsilon = 6 \times 10^{-3}$ which is the default setting in [18]. For the analysis dictionary at layer $2 \leq l < L$, the parameter setting in Eqn. (5) is $\kappa = 4 \times 10^4$, $\upsilon = 6 \times 10^2$ and $\tau = 1 \times 10^{-3}$. The maximum number of iterations for GOAL is set to 1000. The weighting parameters in dictionary update algorithm is set to $\gamma_1 = 1$ and $\gamma_1 = 10$. The proportion of data belonging to the Laplacian distribution that will be zeroed after soft-thresholding is set to 5%, 10%, and 10% for layer 1, 2 and 3, respectively.

Table I shows the PSNR (dB) of Bi-cubic interpolation method, Zeyde's method [8], our proposed deep dictionary (DD) model without and with dictionary update evaluated on the 5 images from *Set 5*. For the Zeyde's method, the size of the LR patch is also set to 8×8, the number of atoms is set to 1024, and the maximal sparsity is tuned to 5 in order to achieve the best performance. The average PSNR of our initial deep dictionary is about 1 dB higher than that of Bi-cubic interpolation, however, lower than the PSNR of Zeyde's method. With dictionary update, the average PSNR of deep dictionary model is around 0.3 dB and 0.5 dB higher than that of Zeyde's method and our deep dictionary model without dictionary update, respectively.

5. CONCLUSIONS

In this paper, we proposed a deep dictionary model for image super-resolution. Our proposed model is with L layers of dictionaries where the first $L - 1$ dictionaries are analysis dictionaries and the last one is a synthesis dictionary. The learned analysis dictionary of layer 1 has Gabor like row atoms, while the remaining analysis dictionaries are designed to be sparse. The sparse dictionaries impose structure in the information retained. Moreover, our dictionary update algorithm iteratively improves the dictionaries in a backward fashion. Numerical results on image SR indicate the potential of the proposed architecture.

6. REFERENCES

[1] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[2] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[3] V. Papyan, Y. Romano, and M. Elad, "Convolutional neural networks analyzed via convolutional sparse coding," *arXiv preprint arXiv:1607.08194*, 2016.

[4] J. Sulam, V. Papyan, Y. Romano, and M. Elad, "Multi-layer convolutional sparse modeling: Pursuit and dictionary learning," *arXiv preprint arXiv:1708.08705*, 2017.

[5] R. Rubinstein and M. Elad, "Dictionary learning for analysis-synthesis thresholding," *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5962–5972, 2014.

[6] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1553–1564, 2010.

[7] J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad, "Trainlets: Dictionary learning in high dimensions," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3180–3193.

[8] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.

[9] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[10] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1920–1927.

[11] ——, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 111–126.

[12] J.-J. Huang, W.-C. Siu, and T.-R. Liu, "Fast image interpolation via random forests," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3232–3245, 2015.

[13] J.-J. Huang and W.-C. Siu, "Learning hierarchical decision trees for single image super-resolution," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[14] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*. Springer, 2014, pp. 184–199.

[15] ——, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[16] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[17] R. Rubinstein, T. Peleg, and M. Elad, "Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 661–677, 2013.

[18] S. Hawe, M. Kleinsteuber, and K. Diepold, "Analysis operator learning and its application to image reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2138–2150, 2013.

[19] J. Dong, W. Wang, W. Dai, M. D. Plumbley, Z.-F. Han, and J. Chambers, "Analysis simco algorithms for sparse analysis model based dictionary learning," *IEEE Transactions on Signal Processing*, vol. 64, no. 2, pp. 417–431, 2016.

[20] N. T. Trendafilov, "A continuous-time approach to the oblique procrustes problem," *Behaviormetrika*, vol. 26, no. 2, pp. 167–181, 1999.

[21] G. Taylor, R. Burmeister, Z. Xu, B. Singh, A. Patel, and T. Goldstein, "Training neural networks without gradients: A scalable admm approach," in *International Conference on Machine Learning*, 2016, pp. 2722–2731.