Fast Single Image Inpainting via Recurrent Neural Networks and Learning Dictionaries with Orthogonality Constraint

Yang Jiao
Johns Hopkins University
yjiao8@jhu.edu

Minh Bui
Johns Hopkins University
minhbui@jhu.edu

Ruizhi Zuo
Johns Hopkins University
rzuo3@jhu.edu

Abstract

In sparse representation, a signal can be represented by a linear combination of different atoms in a dictionary. This process can be iteratively achieved by: 1) sparse coding (SC) for sparse coefficients and 2) dictionary learning (DL) for updating dictionary atoms. However, the bottleneck of these two steps are: 1) hard to optimization for $l0$ and $l1$ and 2) in-feasibility of pseudo-inverse when dictionary is large. However, the huge success of deep neural networks proves the high efficiency of back propagation algorithm for large scale optimization. Motivated by this, in this paper, we proposed an Long-Short Term Memory based Sparse Coding Netowrk (LSTM-SCN) for fast codes estimation, and extend the current K-SVD pipeline for neural netowrks. LSTM-SCN can be learned end-to-end, and the experiments conducted on image inpainting show that the LSTM-SCN achieves best performance among different baselines with much lower computational cost.

Besides this, we also investigate the dictionary learning step on how to update an orthogonal matrix to minimize Frobenius norm and keeps the orthogonality property intact.

1. Introduction

Sparse Representation (SR) [1]] try to solve the problem in which a signal Y can be represented by a linear combination of different atoms from a over-complete dictionary D, while keeping the coefficients of atoms sparse as Eq. (1) shows.

$$\hat{X} = \arg\min_{X} ||Y - DX||_F + ||X||_0 \tag{1}$$

in which the coefficients X is a column sparse matrix that has limited non-zero elements on each columns. However, for Eq. 1 to gives a unique solution, the dictionary D needs to satisfy several constraints, most prominently known are Spark, NUP, RIP [2]. These constraints are quite easily satisfied with random dictionaries. However, for interpretability, discriminative and generative power of the sparse coding coefficients, one may desire to study a better and meaningful dictionary by data. Particularly, we want to find dictionary D to optimize:

$$\hat{D} = \arg\min_{D,X} ||Y - DX||_F + ||X||_0 \tag{2}$$

This process can be achieved by two steps: 1) Sparse Coding for getting coefficients, and 2) Dictionary Learning for a compact dictionary. Following this principle, dictionary learning and sparse coding enables huge successful in different tasks, such as face classification [1], image inpainting [3], denoising [4] and super-resolution [5], matrix completion, dimensional reduction [6].

In sparse coding step, we need to find the sparse coding of the signal. Due to the high complexity of l_0 optimization (combination explosion, NP-Hard), traditional methods often involves greedy techniques like Orthogonal Matching Pursuit (OMP) and Iterative Soft-Thresholding Algorithm (ISTA). These methods include pseudo-inverse operations, which have very high computational cost and is hard to implemented for real-time tasks. However, the huge success of deep neural networks make it possible to solve large optimization problem by backward propagation (BP) algorithm. This motivated us to design specific neural networks to solve the l_0 or l_1 approximation optimization problem, while still reserve the consistency term in dictionary task.

In dictionary updating step, the trivial solution to optimize the Frobenius norm is to take pseudo inverse of matrix X, as in Method of Optimal Direction (MOD) [7]. However, for X large, it is computationally infeasible to compute the pseudo inverse, and X also does not fully available at the computational time. We try to look for a different way to update the dictionary, particularly in form of Singular Value Decomposition (SVD) and look for a way to ensure orthogonality everywhere we go.

Hence in this paper, we propose an Long-Short Term Memory (LSTM)[8] based Sparse Coding Network (LSTM-SCN) to estimate the coefficients X end-to-end learning, and extend the current pipeline in K-SVD [9] for neural network. We evaluate the proposed method on image inpainting task, and the experiments show that LSTM-SCN achieves not only the best results (PSNR=30.8), but also the highest computational efficiency (3.7 faster than KSVD). And the visualization of the learned dictionary also shows a wide range of image frequencies with the high interpretability. The contributions are summarized as follows:

1. We proposed a LSTM based Sparse Coding Network (LSTM-SCN) for fast approximation of coefficients estimation. LSTM-SCN can be trained end-to-end and achieve better performance with much lower running time.
2. We extend the current pipeline of K-SVD to neural netowrks. The extend pipeline is prove efficient in experiments and can be easily extend to any deep neural networks.
3. We demonstrate the idea to learn dictionary while maintaining orthogonality.

The paper is organised as follows. Section 2 introduces the related works and Section 3 introduces the proposed LSTM-SCN methods. Experiments are conducted and discussed in Section 4. And finally, Section 5 gives the conclusion and future works.

2. Related Works

This section, we first introduce the sparse coding and dictionary learning in SubSec. 2.1 and SubSec. 2.2, and then given the background of Coordinate Descent in SubSec. 2.3 and learning orthogonal matrix in SubSec. 2.4.

2.1. Sparse Coding

In the general form of sparse coding, the goal is to find the optimal sparse code $X \in R^n$ to minimize the energy function that computes the $l2$ norm between the measurement Y and reconstructed signal DX.

$$\hat{X} = \arg\min_X \frac{1}{2}||Y - DX||_2^2 + \alpha||X||_0 \tag{3}$$

in which dictionary D with size $m \times n$ $(m > n)$ is given and fixed, and α is the trade-off to balance the fedelity and sparse regularization.

To solve Eq. 3, many pursuit methods are proposed, such as Matching Pursuit (MP), Orthogonal Matching Pursuit (OMP), Subspace Pursuit. In these algorithms, MP independently picks the atom which contributes most to the reconstruction error once a time, and OMP makes the atoms orthogonal to the error space for faster convergence. However, $l0$ optimization is usually intractable because of the combination explosion in big data applications. Hence, people use $l1$ to approximate the solution of $l0$, and propose Least Absolute Shrinkage and Selection Operator (LASSO) model as Eq. 4 shows.

$$\hat{X} = \arg\min_X \frac{1}{2}||Y - DX||_2^2 + \alpha||X||_1 \tag{4}$$

A popular method to solve this is the Iterative Shrinkage and Thresholding Algorithm (ISTA). Given an input measurement Y, ISTA iterates the following shrinkage function at time k until convergence:

$$Z_{k+1} = h_\theta(Z_k - L^{-1}D^T(DZ_k - Y)) \tag{5}$$

in which h_θ is soft thresholding function and $\theta = s/L$, s is sparsity and $L > eigen(D^TD)$ is Lipschitz constant. Though ISTA is easily implemented, but can only achieve computational complex by $O(mn)$ or $O(mk)$ [10]. To speed up ISTA, Fast ISTA (FISTA) is proposed by Beak et. al [11] by introducing the 'momentum' term in the dynamics. As discussed in [11], while FISTA is faster than most standards, but may still need several dozen iterations before convergence.

2.2. Dictionary Learning

Instead of using a given engineered dictionary, such as Fourier Transform (FT) basis or Discrete Consine Transform (DCT) basis, dictionary learning updates the atoms from the training data to better fit the model dynamically. In dictionary learning, the learned dictionary matrix is often optimized by minimizing the mean of fidelity term in Eq. 4 over a set of training samples. Different DL methods are proposed aiming to better represent the signal with more sparse code, in which K-SVD and Online Dictionary Learning (ODL) [12] are two baselines.

In K-SVD method, given the representation X, do the update on dictionary D. This can be done sequentially for every atom in D, by forming the error when remove the atom k: $E_k = Y - \sum_{j \neq k} d_j x_T^j$. The reconstruction error is now become $||E_k - d_k x_T^k||_F^2$. Now, for every k, choosing the signal that related to atom k: Y_k^R, sparse representation that use atom k: x_R^k by the indicating matrix Ω_k. By this selection (slicing) operation, the sparsity of X is kept.The problem now become minimizing $||E_k^R - d_k x_R^k||_2^F$. From SVD, we know E_k^R can be decomposed to $U\Delta V^T = \sum U_i \Delta_i V_i^T$, where U and T orthogonal and Δ_i decreasing sorted. Thus, the solution to the problem is by changing atom d_k and x_R^k to the vector U_1 and $\Delta_1 V_1^T$ respectively. Given the sparse coding algorithm behave well (high recovery rate), the K-SVD algorithm is guaranteed to converge, as the reconstruction error minimizing step by SVD is monotonically decreasing. Several tweaks in implementation can be use, like replace the unused atom or repeat (close) atoms with least represented signal element.

2.3. Coordinate Descent and Line search

Coordinate Descent is a popular technique to optimize an function toward its maxima or minima. Unlike steepest gradient descent, for each step of coordinate descent, the algorithm try to step following one of the coordinate direction, and try to minimizing the function by optimize one variable at a time.

Line search is often perform in each step of Coordinate Descent to roughly determine the step size.

2.4. Learning Orthogonal matrix by Coordinate Descent

In [13], the Riemannian directional derivative of f in the direction of a vector $U \in T_U O_d$ is defined as the derivative of a single variable function which involves looking at f along a single curve:

$$\nabla_\Omega f(U) = \frac{d}{d\theta} f(\gamma(\theta))|_{\theta=0} = \frac{d}{d\theta} f(UExp(\theta\Omega))|_{\theta=0}$$

Stepping along the coordinate of orthogonal basis for Skew(d), $H_{ij} = e_i e_j^T - e_j e_i^T$, updating U with the step size α equivalent to: update $U_{t+1} = U_t Exp(-\alpha H_{ij})$ where $Exp(-\alpha H_{ij})$ equal to a Givens rotation matrix that rotate i and j dimension with angle α (i.e. $G(i, j, \alpha)$).

Algorithm 1 Riemannian coordinate minimization on O_d

```
Input: Differentiable objective f, initial matrix $U_0 \in O_d$
Output: $U_{final}$
$t = 0$
while not converged do
    1. Sample $i, j \in [1, d]$
    2. $\theta_{t+1} = argmin_\theta f(U_t G(i, j, \theta))$
    3. $U_{t+1} = U_t G(i, j, \theta_{t+1})$
    4. $t = t + 1$
end while
```

3. Proposed Method

In this section, we propose the improved K-SVD method with LISTA technique and LSTM based fast approximation of sparse coding in Sec. 3.1 and Sec. 3.2 respectively. Then we give the idea of dictionary learning with orthogonal matrix constraint in section 3.3.

3.1. Learning Sparse Code in K-SVD

As discussed before, the bottle neck of current sparse representation algorithms is the expensive computational cost of sparse coding step. Instead of using traditional optimization methods, we propose to learn the sparse codes by introducing deep neural networks, and embed it into the current pipeline as Algorithm 3.1 describes.

Algorithm 2 Learning Sparse Code in K-SVD

```
input: $D$, $Y$
output: $\hat{D}$, $X$
step 1: sparse code network training
i=1
repeat
    $\mathrm{X}_i = net.forward(Y_i)$loss = l$(\mathrm{X}_i, Y_i)$
    net = net.backward(loss)
until i=max iteration
step 2: predict coefficients
net = net.forward(Y)
step 3: update dictionary
D, X = KSVD(D, Y)
repeat step 1 to 3 until converging.
```

The proposed pipeline follows the recursive pipeline of K-SVD, and each iteration includes three steps: 1) training the sparse coding network, 2) predicting the sparse codes X and 3) update the dictionary $\hat{D}$. It should be noticed that because of update process of dictionary, the distribution of new predicted sparse codes varies. Hence, we need to fine-tune the sparse coding network in each iteration. However, even though the network need to be trained for several times, the experiments show that the proposed pipeline can still achieve very high computation efficiency.

3.2. LSTM-SCN

To learn the sparse code in the proposed pipeline, we design a LSTM based sparse coding network (LSTM-SCN) which mimics the ISTA algorithm to estimate the coefficients. In the original ISTA algorithm, the recursive coefficient update step shown in Eq. 5 can be re-formulated as:

$$Z_{k+1} = h_\theta((1 - L^{-1} D^T D) Z_k + L^{-1} D^T Y) \qquad (6)$$

in which the two terms in $h(.)$ can be seen as the function of the measurement Y and the function of the coefficients Z_k at time step k. Hence, motivated by LISTA, we approximate Eq. 6 with Eq. 7 as follows.

$$Z_{k+1} = h_\theta(W Z_k + S Y) \qquad (7)$$

In Eq. 7, W and S are two metrics from fully-connected layers that can be learned by the back propagation in neural networks.

Different from the LISTA, we adopt two variables to maintain the approximation of coefficients learning, and design the learned ISTA based on LSTM framework shown as Fig. 1. In LSTM-ISTA, the input is raw measurement Y and dictioanry D, and the output is the learned sparse codes which is the hidden state of LSTM h_K at last step K. At the first step of LSTM-ISTA, the hidden state h_0 of LSTM is initialized by data Y, and the cell state C_0 is set to 0.

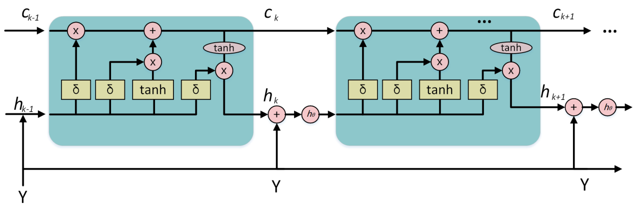


Figure 1. The structure of proposed LSTM-ISTA framework for sparse coding approximation.

Then in each following iteration k, the new hidden state h_k is added by raw input Y followed by a fully-connected layer S to model $WZ_K + SY$ term in Eq. 7. After this, a shrinkage function h_θ with $\theta = s/L$ is applied at the end of LSTM cell.

LSTM-ISTA can be trained end-to-end with standard LASSO loss function, including the reconstruction loss $loss_r$ and sparsity loss $loss_s$ over each batch. The two loss are balanced by the scale factor α and β, and the final loss function of LSTM-ISTA is shown in Eq. 8.

$$loss = \alpha loss_r + \beta loss_s = \alpha||Y - Dh_K||_2^2 + \beta||h_K||_1 \quad (8)$$

3.3. Dictionary Learning with orthogonal constraint

In dictionary learning algorithm, the dictionary updating steps try to update so it minimize the consistency term:

$$D = argmin_D||Y - DX||_F^2 \quad (9)$$

For simplicity, we assume D is square, and the resulting matrix from optimization is orthogonal, i.e. $DD^T = I$. Thus, we can apply algorithm 2.4 to find D that minimizing the objective function $f = ||Y - DX||_F^2$. Assume we update D_t with $G(i, j, \alpha)$, that means:

$$D_{t+1} = D_t G(i, j, \alpha)$$

, step 2 of the algorithm 2.4 becomes

$$\alpha = argmin_\alpha||Y - D_t G(i, j, \theta)X||_F^2$$

. After update D_t, D_{t+1} is still orthogonal because $D_{t+1}D_{t+1}^T = D_t G(i, j, \alpha) G^T(i, j, \alpha) D_t^T = D_t D_t^T = I$ One can find close form of α that maximize the above equation by taking derivative of f with respect to α and set to 0. Applying chain rule and taking derivative of Frobenius norm with a matrix, one can find a close form solution for optimal α (we have not worked it out).

Instead of finding optimal α, we borrow idea of line search to perform gradient descent on α for several steps to find α that reduce the objective function:

$$\alpha = \alpha - rate * \frac{\delta f}{\delta \alpha}$$

For a generalized size of D, one can do Singular Value Decomposition $D_t = USV$, where U,V is orthogonal matrix, and S is diagonal. We can use Method of alternating direction to optimize iteratively, and pay attention when update S to keep it diagonal.

4. Experiments

We evaluate the proposed method with different sparse coding techniques on image inpainting task. In this section, we first give the implementation details in SubSec. 4.1, and then evaluate the effectiveness of the proposed LSTM-SCN with 3 different methods, which are ISTA, LISTA, and K-SVD. In this section, we first introduce 4 models we used to achieve the image inpainting, with their experiment setup. Then we compare their inpainting results, which is quantified as Peak signal-to-noise ratio(PSNR) and time cost.

4.1. Implementation Details

The goal of single image inpainting is to reconstruct the raw image I_{raw} from the degenerated image I_d with missing pixels. In our experiments, we use different gray scale images to evaluate the performances of the proposed method. All the images are resized to 256x256, and the ratio of missing pixels is set to 0.5. For all the experiments, the single input image is cropped by 8x8 overlapped patches with stride 3, totally 7056 image patches. The number of atoms in dictionary is set to 256, and the sparsity is set to 0.2 (keep 50 non-zeros entries). As a result, the length of input signal is 64, the size of dictionary is 64x256, and the sparse codes have the size 256x1.

In the training the proposed LSTM-SCN, the length of hidden state and cell state in LSTM is set to 256, and the network is optimized by SGD optimizer with momentum 0.9, learning rate 0.01 and batch size 128 with 300 epochs. α and β for loss are set to 1. in All the experiments are conducted on a PC with CPU Intel Core i7-8750H @2.2GHz, and GPU Nvidia GeForce 1060 Max-Q with 6GB video memory.

4.2. Image Inpainting Comparisons

We compare the proposed LSTM-SCN method with traditional sparse coding method ISTA, K-SVD, and the network based LISTA. In K-SVD method, the sparse coding is achieved by ISTA. The quantitative results and the reconstruction images are listed in Tab. 1 and Figure. 5.

Method	PSNR	Time(s)
ISTA	28.47	1510
LISTA	29.52	231
K-SVD	30.40	299
LSTM-SCN (ours)	**30.80**	**81**

Table 1. Result of image inpainting of different methods

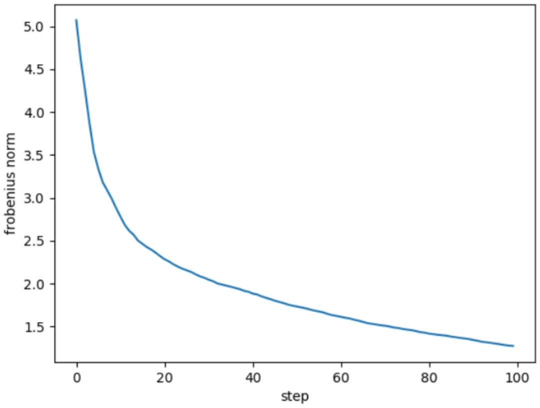


Figure 2. Consistency term after iterations of algorithm 2.4

From Table. 1, it can be seen that the proposed LSTM-SCN achieves best performance with PSNR=30.80 compared with KSVD (30.04) and LISTA (29.52) which is also the network based method. Besides the reconstruction error, the experiments prove that the proposed method also have the very high computational efficiency comparing with all the competitors. LSTM-SCN only takes 81 seconds to convergence, which is 19, 3.7 and 2.8 faster than ISTA, K-SVD and LISTA respectively.

To compare the learning process of the proposed LSTM-SCN, we also plot the loss curve in Fig. 4 and the learned dictionary in Fig. 3. Fig. 4 shows that the proposed LSTM-SCN converges much faster than LISTA at the beginning compred with LISTA. The ripple is caused by the update of the dictionary. Once the dictionary updates, the network has to be trained again to fit the new dictionary.

From the illustration of learned dicionary in Fig. 3, it is surprising that LSTM-SCN learns highly meaningful atoms. The top left shows the learned very low frequency atoms, while the bottom right atoms show the capability of high frequency extraction. And between the top left and bottom right, the learned atoms in the middle region have very clear edges, performing like different edge detection filters. All of these prove that the proposed LSTM-SCN achieves very good results with low computational cost, and can learn highly interpretable dictionary.

4.3. Orthogonal dictionary learning

We tested algorithm 2.4 with consistent term loss function 9. We perform 2.4 for 100 iteration, with number of gradient descent step to choose α is 4 steps. The result is depicted in Figure. 2. We randomly sample X and orthogonal D, then we compute Y, discard D and try to find D to minimize consistency term. All matrices are in size 8x8.

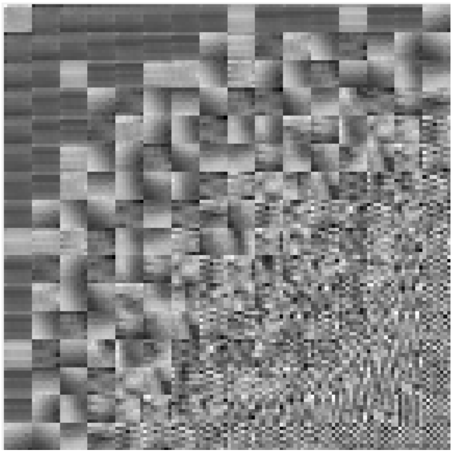


Figure 3. Visualization of learned dictionary from the proposed LSTM-SCN

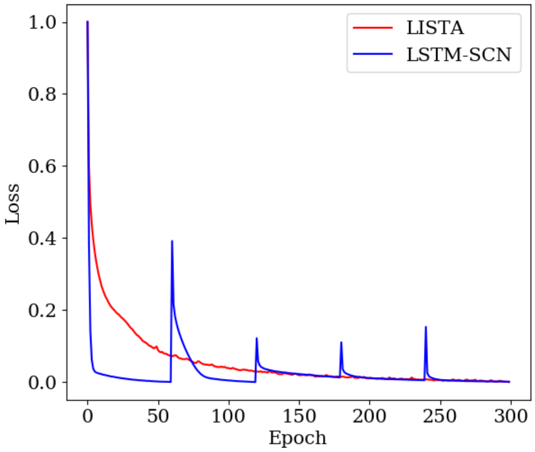


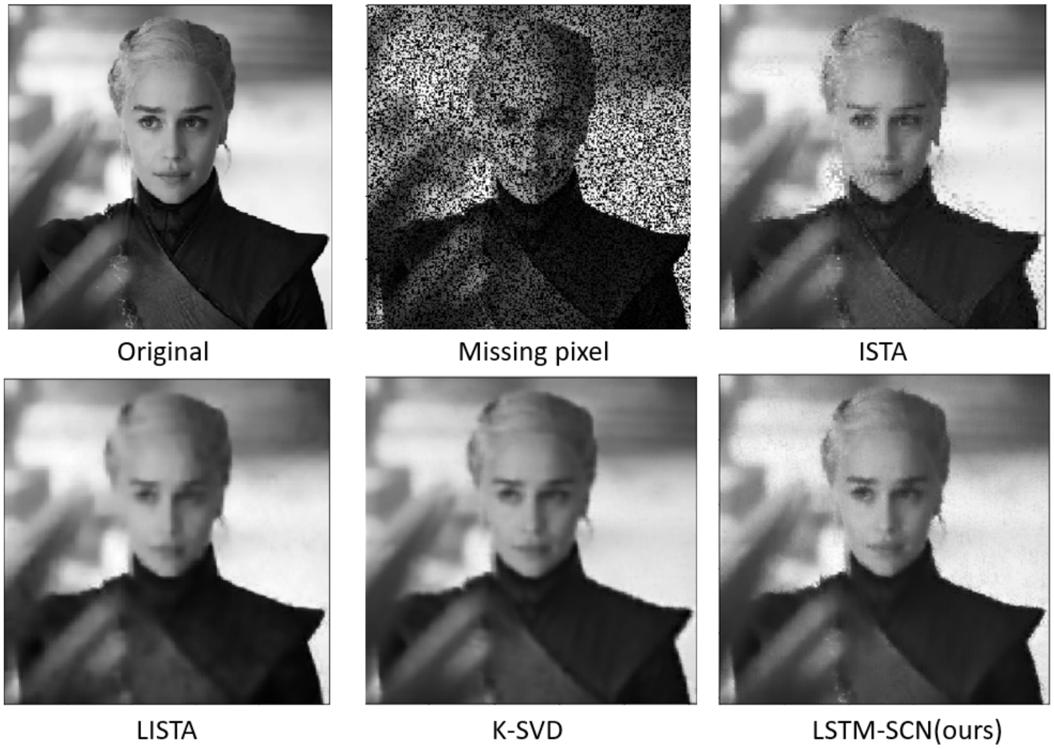
Figure 4. Loss comparison between LISTA and LSTM-SCN. It should be noticed that the ripple is caused by the dictionary update.

5. Conclusion and Discussion

In this paper, we proposed an LSTM based Sparse Coding Network (LSTM-SCN), and combine it into the current K-SVD pipeline. The results show that LSTM-SCN achieves the best performance among different baselines with much lower computational cost.

We study different type of sparse coding algorithm and dictionary learning algorithm. Toward dictionary update step, we try to perform coordinate descent update on $O(d)$ sphere to keep the orthogonality of the learned dictionary. In future work, we can look at how to perform dictionary learning on a fat matrix, as well as, try to update the dictionary by block.

.



Original

Missing pixel

ISTA

LISTA

K-SVD

LSTM-SCN(ours)

Figure 5. Image inpainting results of different methods.

References

[1] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2008. 1

[2] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005. 1

[3] Qiang Li, Yahong Han, and Jianwu Dang. Image decomposing for inpainting using compressed sensing in dct domain. *Frontiers of Computer Science*, 8(6):905–915, 2014. 1

[4] Amin Tavakoli and Ali Pourmohammad. Image denoising based on compressed sensing. *International Journal of Computer Theory and Engineering*, 4(2):266, 2012. 1

[5] Lipeng Ning, Kawin Setsompop, Oleg Michailovich, Nikos Makris, Martha E Shenton, Carl-Fredrik Westin, and Yogesh Rathi. A joint compressed-sensing and super-resolution approach for very high-resolution diffusion imaging. *NeuroImage*, 125:386–400, 2016. 1

[6] Iain M Johnstone and Arthur Yu Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682–693, 2009. 1

[7] Kjersti Engan, Sven Ole Aase, and J Hakon Husoy. Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 5, pages 2443–2446. IEEE, 1999. 1

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[9] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. 1

[10] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 399–406, 2010. 2

[11] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 693–696. IEEE, 2009. 2

[12] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010. 2

[13] Uri Shalit and Gal Chechik. Coordinate-descent for learning orthogonal matrices through givens rotations. *31st International Conference on Machine Learning, ICML 2014*, 1, 12 2013. 3