TASK-DRIVEN DICTIONARY LEARNING FOR INPAINTING

Huiyi Hu[*]

University of California, Los Angeles
Department of Mathematics

Brendt Wohlberg, Rick Chartrand[†]

Los Alamos National Laboratory
Theoretical Division

ABSTRACT

Several approaches used for inpainting of images take advantage of sparse representations. Some of these seek to learn a dictionary that will adapt the sparse representation to the available data. A further refinement is to adapt the learning process to the task itself. In this paper, we formulate a task-driven approach to inpainting as an optimization problem, and derive an algorithm for solving it. We demonstrate via numerical experiments that a purely task-driven approach gives superior results to other dictionary-learning approaches.

***Index Terms*—** Sparse representations, dictionary learning, inpainting

1. INTRODUCTION

We consider the application of sparse representations and dictionary learning methods to imaging problems that can be posed in terms of filling in unknown information, such as image inpainting and demosaicing. Due to space constraints, we will focus on inpainting, where missing or corrupted portions of an image are to be estimated. The general framework for this type of approach (see, e.g., [1]) consists of (i) estimation of missing pixels for individual image patches, and (ii) a mechanism for propagating these estimates into the interior of a missing region that is too large to be estimated using a single patch. Since these two components are largely independent, we will develop methods and compare performance purely on the patch-level estimation process; a complete inpainting algorithm would combine these methods with a standard patch-propagation technique.

Given a dictionary D, an image patch $\mathbf{y}$, and projections P and $Q = I - P$ onto the known and missing regions of $\mathbf{y}$ respectively, the standard sparse-representation based reconstruction of the missing region of $\mathbf{y}$ is $QD\mathbf{x}$, where $\mathbf{x}$ is the minimizer of

$$\min_{\mathbf{x}} \lambda\|\mathbf{x}\|_1 + \tfrac{1}{2}\|P(D\mathbf{x} - \mathbf{y})\|_2^2. \quad (1)$$

[*]The work of all authors was supported by the UC Lab Fees Research grant 12-LR-236660. The work of Hu was also supported by the ONR grants N000141210040 and N000141210838.

[†]The work of Chartrand and Wohlberg was also supported by the U.S. Department of Energy through the LANL/LDRD Program.

A simple approach is to choose a different D for each target patch $\mathbf{y}$, by searching the known region of the image being inpainted for the patches $\mathbf{y}_k$ for which the distance between $P\mathbf{y}$ and $P\mathbf{y}_k$ is sufficiently small. These "nearest-neighbor dictionaries" have been shown to provide good performance [1, 2]. For computational and other reasons, however, it would be desirable to construct a single dictionary D for use across the entire set of target patches (e.g. as in [3]).

Learning of the dictionary can be formulated as follows. Let each column of a matrix Y_{train} contain one element of training dataset; most commonly, these are image patches that do not have any missing pixels. We then seek a dictionary D, typically overcomplete, that approximates the columns of Y_{train} in terms of sparse coefficient vectors that form the columns of a matrix X:

$$\min_{X,D} \lambda\|X\|_1 + \tfrac{\xi}{2}\|X\|_2^2 + \tfrac{1}{2}\|DX - Y_{\text{train}}\|_2^2, \quad \text{subject to each } \|\mathbf{d}^i\|_2 \leq 1, \quad (2)$$

where $\mathbf{d}^i$ is the i^{th} column of D. We have used an elastic-net model [4], though in many cases $\xi = 0$ is used. The constraint on the columns of D is used to prevent the degeneracy of obtaining a lower energy by replacing (X, D) with $(\alpha X, \frac{1}{\alpha} D)$ for $\alpha \ll 1$. The dictionary D is learned based on its ability to reconstruct the training data Y_{train}. We refer to this standard approach [5, 6, 7] as Reconstruction Dictionary Learning (RDL). Once the RDL dictionary D is learned, the missing data can be estimated as $Q(DX)$, where X is the minimizer of

$$\min_{X} \lambda\|X\|_1 + \tfrac{\xi}{2}\|X\|^2 + \tfrac{1}{2}\|P(DX - Y_{\text{test}})\|_2^2, \quad (3)$$

and P and Q are the projections onto the known and missing portions of Y_{test}.

The RDL process does not incorporate any knowledge of the task to be performed. While it is possible to apply a dictionary learned via RDL to a task other than reconstruction (see e.g. [8, 9]), it is reasonable to expect that improved performance would be obtained by adapting the dictionary learning to the task. A natural and relatively easy way of achieving this is to augment the objective function of (2) with a term

that penalizes the error in estimating the missing data:

$$\min_{X,D_1,D_2} \lambda\|X\|_1 + \tfrac{\xi}{2}\|X\|^2 + \tfrac{1}{2}\|P(D_1X - Y_{\text{train}})\|_2^2 + \tfrac{\mu}{2}\|Q(D_2X - Y_{\text{train}})\|_2^2 + \tfrac{\gamma}{2}\|D_2\|_2^2, \quad \text{subject to each } \|(\mathbf{d}_j)^i\|_2 \le 1, \quad (4)$$

where we have added a second dictionary to allow the model greater freedom in learning to fill in missing information, and a term regularizing D_2 to mitigate over-fitting. We refer to this as Task-Augmented Dictionary Learning (TADL). Filling in missing information then proceeds as in RDL (3). A variety of methods based on this framework have been considered [10, 11, 12, 13, 14], primarily with $D_1 = D_2$.

However, TADL compromises the ability to fill in missing data by devoting some of the energy in (4) to reconstruction of the known portions of the training data. While this seems desirable, it is in fact not of direct importance: we seek the dictionary that fills in missing information as well as possible, regardless of how well the known portions are reconstructed. Instead, we propose to use Task-Driven Dictionary Learning (TDDL). The sparse representation of the known portions will still be used as in RDL and TADL. The difference is that we remove the quality of this reconstruction entirely from the objective function, giving the following optimization problem:

$$\min_{D_1,D_2} \tfrac{1}{2}\|Q(D_2X^* - Y_{\text{train}})\|_2^2 + \tfrac{\gamma}{2}\|D_2\|_2^2, \text{ subject to } X^* = \arg\min_X \lambda\|X\|_1 + \tfrac{\xi}{2}\|X\|_2^2 + \tfrac{1}{2}\|P(D_1X - Y_{\text{train}})\|_2^2. \quad (5)$$

With TDDL, the objective function only seeks to minimize the error in estimation of the missing data; the sparse representation of the known portions is defined via a constraint rather than an additional term in the objective function. While (5) no longer has the scale degeneracy in D_1 or D_2, we add a penalty on the size of D_2 to guard against overfitting of the training set. We also use an elastic-net model to provide greater regularity, which helps in solving the very challenging problem (5). Once D_1 and D_2 are learned, filling in the missing data proceeds similarly to RDL and TADL, except here X is obtained by solving (3) using $D = D_1$, and then the missing data is $Q(D_2X)$.

In Sec. 2 we present our algorithm for solving (5). In Sec. 3 we present numerical experiments that compare the inpainting performance of RDL, TADL, and TDDL.

1.1. Relation to Prior Work

The general TDDL framework [15, 16, 17] is not new, but has received far less attention in the literature than the simpler dictionary learning frameworks. Of these, the most closely related to our work is that of Mairal et al. [17], where a probabilistic version of (5) is solved using a stochastic gradient descent algorithm. Part of the contribution of this work is to show that a deterministic approach can be used, which results in an algorithm that is more readily parallelizable.

2. TASK-DRIVEN DICTIONARY LEARNING ALGORITHM

In this section we describe our algorithm for solving (5). Let $H = H(D_1, D_2)$ be the functional that is minimized in (5). We let $f = f(D_1)$ be the function defined implicitly in (5) (letting $Y = Y_{\text{train}}$ here and henceforth):

$$f(D_1) = \arg\min_X \lambda\|X\|_1 + \tfrac{\xi}{2}\|X\|_2^2 + \tfrac{1}{2}\|P(D_1X - Y)\|_2^2. \quad (6)$$

Thus we can regard (5) as the following unconstrained optimization problem:

$$\min_{D_1,D_2} \tfrac{1}{2}\|Q(D_2f(D_1) - Y)\|_2^2 + \tfrac{\gamma}{2}\|D_2\|_2^2. \quad (7)$$

Since the part of the data that is known or missing can vary from column to column of Y, it is convenient to decompose our problem into pieces where the known portion of each column is the same. Suppose there are C distinct configurations, and let P_c, Q_c be the corresponding projections for each $c \in \{1, \dots, C\}$. (These can be expressed as diagonal matrices, while P and Q can only be so expressed if their arguments are vectorized.) Let Ω_c be the set of column indices of Y having the c^{th} configuration, and let $Y_c = Y|_{\Omega_c}$ denote the corresponding columns of Y. Similarly define $f_c(D) = (f(D))_c$. Then we have:

$$\|P(D_1X - Y)\|_2^2 = \sum_{c=1}^{C} \|P_cD_1X_c - P_cY_c\|_2^2, \quad (8)$$

$$\|Q(D_2f_c(D) - Y)\|_2^2 = \sum_{c=1}^{C} \|Q_cD_2f_c(D) - Q_cY_c\|_2^2. \quad (9)$$

The minimization process consists of alternating between updating D_1 and D_2 using gradient descent. During the iterations, $f(D_1)$ is computed using a version of the alternating directions, method of multipliers (ADMM) algorithm suitable for elastic net regularization [18].

Because H is a quadratic function of D_2, $\nabla_{D_2}H$ is straightforward to calculate. The step size b is computed as

$$\arg\min_b H(D_1, D_2 - b\nabla_{D_2}H(D_1, D_2)), \quad (10)$$

which can be written down explicitly (see Algorithm 2).

It is more complicated to compute $\nabla_{D_1}H$ because f is not differentiable in general. What is required is that the support and sign of $f_{ij}(D)$ is stable under perturbations. To this end, we adopt the same assumptions as in [17, Appendix], which amount to a restriction on the domain of f (and hence H). Under such assumptions,

$$\nabla_{D_1}H = \sum_{c=1}^{C} \sum_{i,j} \left[D_2^TQ_c^T\left(Q_cD_2f_c - Q_cY_c\right)\right]_{ij} \nabla[f_c]_{ij}, \quad (11)$$

noting that the gradient of the scalar $[f_c(D)]_{ij}$ with respect to D is a matrix. In order to compute ∇f, we use the first order optimality condition for $f_c(D_1)$:

$$0 \in \left[P_c D_1^T (P_c D_1 f_c - P_c Y_c)\right]_{ij} + \lambda \partial |\cdot|([f_c]_{ij}) + \xi [f_c]_{ij}. \tag{12}$$

The assumption that the support of $f(D_1)$ is stable means that $\nabla[f_c]_{ij} = 0$ if $[f_c]_{ij} = 0$. Then we can differentiate (12) and compute $\nabla[f_c]_{ij}$ on the support of f_c. The process of computing $\nabla_{D_1} H$ is described in Algorithm 1.

Algorithm 1 Computing $\nabla_{D_1} H$

Input: D_1, D_2, $f = f(D_1)$, $\{P_c\}$, $\{Q_c\}$, Y, and ξ.
for $j = 1, \ldots, N$, where Y has N columns **do**

- Let c be the configuration of f^j. Let $D_P = P_c D_1$, $D_Q = Q_c D_2$, $Y_P^j = P_c Y^j$ and $Y_Q^j = Q_c Y^j$.
- $\Lambda_j = \mathrm{supp}(f^j)$, $\hat{f}^j = f^j|_{\Lambda_j}$, $\hat{D}_P = D_P|_{\Lambda_j}$ (column-wise), $\hat{D}_Q = D_Q|_{\Lambda_j}$ (column-wise).
- $(\beta^j)^T = \left(\hat{D}_Q \hat{f}^j - Y_Q^j\right)^T \hat{D}_Q \left(\hat{D}_P^T \hat{D}_P + \xi I\right)^{-1}$.
- $\beta_*^j = \begin{cases} \beta^j \text{ on } \Lambda_j, \\ 0 \text{ otherwise}, \end{cases}$ a column vector.
- $\nabla_{D_1} H^j = (Y_P^j - D_P f^j)\beta_*^j - D_P(\beta_*^j)^T (f^j)^T$.

end for
$\nabla_{D_1} H = \sum_{j=1}^N \nabla_{D_1} H^j$.

After $\nabla_{D_1} H$ is computed, D_1 is updated in the descent direction $-\nabla_{D_1} H$ with a pre-chosen step size a. The process of minimizing H is summarized in Algorithm 2.

Algorithm 2 Task-Driven Dictionary Learning (TDDL)

Input: D_0, $\{P_c\}$, $\{Q_c\}$, $\{Y_c\}$, λ, ξ, γ, n_{iter}, and a.
Initialization: $D_1 \leftarrow D_0$, $D_2 \leftarrow D_0$.
for $k = 1, \ldots, n_{\text{iter}}$ **do**

- Compute $f(D_1)$

for $c = 1, \ldots, C$ **do**
$f_c = \arg\min_X \frac{1}{2}\|P_c D_1 X - P_c Y_c\|_2^2 + \lambda \|X\|_1 + \frac{\xi}{2}\|X\|_2^2$
end for

- Compute $\nabla_{D_1} H \leftarrow$ Algorithm 1.
- Update $D_1 \leftarrow D_1 - a \nabla_{D_1} H$.
- Update $D_2 \leftarrow D_2 - b \nabla_{D_2} H$, where

$$\nabla_{D_2} H = \sum_{c=1}^C Q_c^T (Q_c D_2 f_c - Q_c Y_c) f_c^T + \gamma D_2,$$

$$b = \frac{\gamma \langle D_2, \nabla_{D_2} H\rangle + \sum_{c=1}^C \langle Q_c \nabla_{D_2} H f_c, Q_c D_2 f_c - Q_c Y_c\rangle}{\gamma \|D_2\|_2^2 + \sum_{c=1}^C \|Q_c \nabla_{D_2} H f_c\|_2^2}.$$

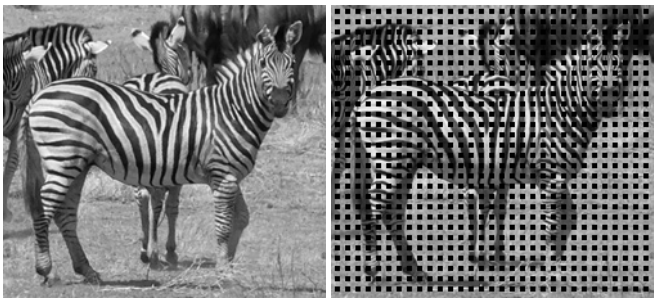
end for

The solution of (4), which we will use for comparison, is computed by alternately updating X, D_1 and D_2. An ADMM algorithm is used to solve for X with given D_1 and D_2, and the two dictionaries are updated using gradient descent and optimal step sizes. After each update, the columns of D_1 and D_2 are rescaled such that the norms do not exceed 1.

3. NUMERICAL EXPERIMENTS

3.1. Single configuration, patch-level inpainting

Here we implement TDDL on a 228×252 grayscale zebra image (Fig. 1(a)), for a patch-level inpainting task. (All zebra images are from the ImageNet database [19].) The testing data Y_{test} consists of four sets of tiled 8×8 patches with relative offsets of $(0,0)$, $(0,4)$, $(4,0)$, and $(4,4)$ pixels. The center 4×4 square of each patch is the missing region. The inpainting mask of one set of the tiled patches is shown in Fig. 1(b). The projections P and Q correspond to the known region of the patch and the center missing region respectively. The goal is to inpaint the center missing region of each patch.



(a) Zebra image (b) One set of corrupted patches

Fig. 1. Inpainting test where the missing region of all patches is in the same location within the patch.

We randomly choose 16,000 8×8 patches from 40 zebra images as the training data Y_{train} for RDL and TDDL with $C = 1$, and learn dictionaries of size 64×256. The parameters for TDDL are chosen as $\lambda = 0.01$, $\xi = 0.1$, $\gamma = 3$, $n_{\text{iter}} = 300$ and $a = 2 \times 10^{-5}$, while for RDL we use $\lambda = 10^{-4}$ and $\xi = 0$. The initial dictionary is one previously learned from 10^7 patches from separate zebra images. With a single configuration, the dictionaries D_1 and D_2 can be considered the complementary portions of a single dictionary D.

After learning the dictionary D, we compute the sparse coefficients using the known region of the patches:

$$\hat{X} = \arg\min_X \tfrac{1}{2}\|D_P X - P Y_{\text{test}}\|_2^2 + \lambda \|X\|_1 + \tfrac{\xi}{2}\|X\|_2^2. \tag{13}$$

Then $D_Q \hat{X}$ gives us the reconstruction of the missing region. For TDDL, λ and ξ used in (13) are the same as in the learning process, while the best performance for RDL was obtained with $\lambda = 0.01$ and $\xi = 0$. The SNR and SSIM [20] of the

missing region are compared with the ground truth as a measure of inpainting performance (see Table 1). We find TDDL gives substantially better performance. Moreover, simply using the initial dictionary in (13) gives roughly the same performance as RDL, so unlike TDDL, RDL is not providing any improvement in quality. (We do not include TADL for this test because in the single configuration case, TADL reduces to an RDL model with parts of the dictionary and data rescaled.)

We also compute the performance for reconstruction (with $\lambda = 0.1$) using the Nearest Neighbor Dictionaries (NND) discussed in the Introduction. This method performs relatively poorly in this particular test, but this is not surprising, since in this case the nearest-neighbor search is performed on a set of training images rather than the image to be inpainted itself, and thus the set of exemplars included in the dictionary is expected to be of reduced efficacy [21].

	TDDL	RDL	NND
SNR (dB)	8.65	6.59	1.44
SSIM	0.843	0.816	0.581

Table 1. Single configuration, patch-level inpainting. SNR, and SSIM [20] are computed over the inpainting region only.

3.2. Multi-configuration inpainting

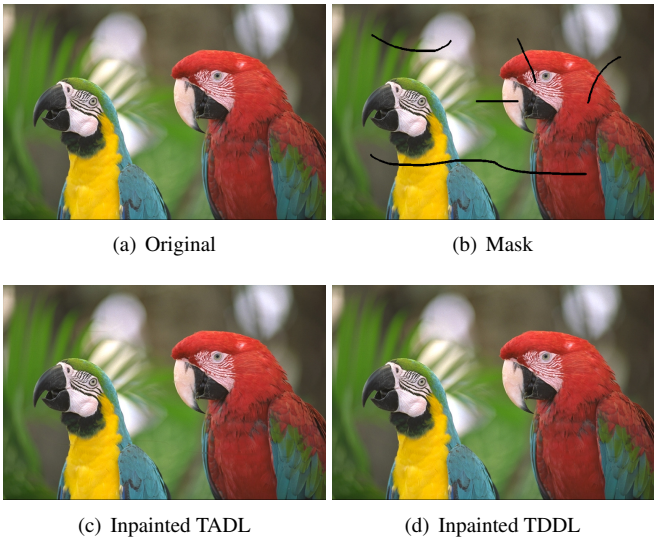
We test a more general inpainting task to demonstrate TDDL with many configurations. The 512×768 color test image ("kodim23" from [22]) and the corresponding inpainting masks are displayed in Figs. 2(a) and 2(b) respectively.

We take every 8×8 patch from the original image, and let the ones containing any missing pixels (about 1.5×10^4) be the testing data Y_{test}, and the others (about 3.6×10^5) be the training data Y_{train}. Since the color image consists of three channels, each patch corresponds to a 192-dimensional vector. Among all the testing data, there are in total 3,022 distinctive configurations of the missing region on patch (i.e., $C = 3022$). Let $\{P_c, Q_c\}_{c=1}^{3022}$ denote these configurations. We randomly assign the patches in Y_{train} to the configurations, such that the frequency of each configuration in Y_{train} is the same as in Y_{test}.

Then we implement the TDDL algorithm with $\lambda = 10^{-3}$, $\xi = 10^{-4}$, $\gamma = 4$, $n_{\text{iter}} = 220$ and $a = 5 \times 10^{-7}$. The initial D_0 (192×256) was set to the dictionary learned on 2×10^7 natural image patches provided online by the authors of [23].

For TADL, we use the same D_0 and Y_{train}, with $\lambda = \mu = 10^{-6}$, $\xi = 0.1$, $\gamma = 4$ and $n_{\text{iter}} = 800$, for its best performance. RDL is learned with $\lambda = 0.01$ and the corresponding reconstruction is performed with $\lambda = 7 \times 10^{-4}$ and $\xi = 0$.

The reconstructions using both TDDL and TADL are computed using the same λ and ξ as in their learning processes. The reconstructed images are shown in Figs. 2(c) and 2(d), and inpainting performance, including reconstruction using NND, is summarized in Table 2. Note that the TDDL performance, using a single dictionary for all patches, is competitive with that of NND, which is able to adaptively choose the dictionary *for each patch*.



(a) Original (b) Mask

(c) Inpainted TADL (d) Inpainted TDDL

Fig. 2. Multi-configuration inpainting results. The missing region of each is simply its intersection with the mask. Traces of the mask are still visible in the TADL result, but are far less evident in the TDDL result. (For a proper comparison, images should be zoomed in the electronic version of this document.)

	TDDL	TADL	RDL	NND
SNR (dB)	18.47	17.01	16.00	18.82
SSIM	0.920	0.880	0.886	0.927
Color SSIM	0.950	0.937	0.939	0.949

Table 2. Multi-configuration inpainting results. SNR, SSIM [20], and color SSIM [24] are computed over the inpainting region only.

4. CONCLUSION

We have developed a method for addressing problems such as image inpainting via dictionary learning based on an optimization problem that directly represents the actual problem of interest. This problem is solved via a deterministic algorithm that is more readily parallelizable than a prior stochastic gradient descent algorithm for the same problem class. It is demonstrated that the proposed framework can be used to learn a single dictionary pair that can effectively estimate different configurations of known and missing elements of the problem data. This is vital for a practical application to inpainting problems with a general inpainting region mask. Inpainting performance is superior to dictionary learning methods that are not purely task-driven, and competitive with nearest-neighbor dictionaries that determine a separate dictionary for each patch.

5. REFERENCES

[1] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1153–1165, 2010.

[2] B. Wohlberg, "Inpainting by joint optimization of linear combinations of exemplars," *IEEE Signal Process. Lett.*, vol. 18, no. 1, pp. 75–78, Jan. 2011.

[3] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, 2008.

[4] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. R. Statist. Soc. B*, vol. 67, pp. 301–320, 2005.

[5] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[6] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, 1999, pp. 2443–2446.

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

[8] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *International Conference on Machine Learning (ICML)*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 759–766.

[9] J. Yang, K. Yu, Y. Gong, and T. S. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1794–1801.

[10] D.-S. Pham and S. Venkatesh, "Joint learning and dictionary construction for pattern recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[11] F. Rodriguez and G. Sapiro, "Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries," Institute for Mathematics and its Applications, IMA Preprint Series 2213, 2008.

[12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Neural Information Processing Systems Foundation, 2008, pp. 1033–1040.

[13] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2691–2698.

[14] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1697–1704.

[15] Y. Boureau, F. Bach, Y. A. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 2559–2566.

[16] J. Yang, K. Yu, and T. S. Huang, "Supervised translation-invariant sparse coding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3517–3524.

[17] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, April 2012.

[18] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," Rice University CAAM, Tech. Rep. 12-14, 2012.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[20] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[21] M. Zontak and M. Irani, "Internal statistics of a single natural image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 977–984.

[22] "Kodak lossless true color image suite." [Online]. Available: http://r0k.us/graphics/kodak/

[23] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 2272–2279.

[24] A. Toet and M. P. Lucassen, "A new universal colour image fidelity metric," *Displays*, vol. 24, no. 4-5, pp. 197–207, 2003.