DISCRIMINATIVE AND COMPACT DICTIONARY DESIGN FOR HYPERSPECTRAL IMAGE CLASSIFICATION USING LEARNING VQ FRAMEWORK

Zhaowen Wang[†]*, Nasser Nasrabadi*[‡] *and Thomas Huang*[†]

[†]Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61820, USA
[‡]U.S. Army Research Laboratory, Adelphi, MD 20783, USA

ABSTRACT

Sparse representation provides an efficient description for high-dimensional Hyperspectral Imagery (HSI) and also encodes discriminative information useful for classification. However, due to the large size of typical HSI images, the naive way to construct a dictionary with all training pixels is neither efficient nor practical. In this paper, a novel approach is proposed to design compact dictionary for Sparse Representation-based Classification (SRC). Inspired by Learning Vector Quantization (LVQ) techniques, we use a hinge loss function directly related to classification task as our objective function, and optimize the dictionary by exploiting the differentiable parts of sparse codes. The resultant dictionary updating procedure adapts the "push" and "pull" actions in LVQ to SRC, which is therefore named as Learning Sparse Representation-based Classification (LSRC). Experiments on different HSI images demonstrate that our LSRC approach can achieve higher classification accuracy with substantially smaller dictionary size than using the whole training set, and also outperforms existing dictionary learning methods.

Index Terms— sparse representation, learning vector quantization, hyperspectral image classification

1. INTRODUCTION

Hyperspectral Imagery (HSI) is an important tool in remote sensing which can measure distinct spectral signatures for different ground materials, and it is widely applied in agriculture, military, mineralogy, etc. Different approaches have been used to classify HSI data; successful examples include Support Vector Machine (SVM) [1] and its variations [2, 3].

More recently, Sparse Representation-based Classification (SRC) [4] has also been applied to HSI classification, and achieves competitive results [5]. Sparse representation expresses a signal as the linear combination of very few atoms from an over-complete dictionary, and the resulting sparse code can reveal its class information if signals from different classes lie in different subspaces. The effectiveness of SRC has already been proven in face recognition [4], expression recognition [6], and speaker verification [7]. Good performance on HSI classification is also expected because the high correlation among different channels of HSI image intrinsically induces a low dimensional subspace in which samples can sparsely be represented.

A good dictionary characterizing the subspace structure of each class is the key for SRC to attain high classification accuracy. Conventionally, SRC dictionary is constructed by directly combining all the training samples [4, 5], which is neither efficient nor practical for HSI data with huge number of data samples. Random sampling or clustering methods can give compact dictionaries, but generative as well as discriminative capabilities are lost in such sub-optimal dictionaries. There has been a hot trend lately in computer vision and machine learning communities trying to learn condensed dictionaries well fitted to large scale training data. Generative approaches, such as Method of Optimal Direction (MOD) [8], K-SVD [9, 10], and the relaxed l_1 formulations [11, 12], have focused on minimizing signal reconstruction errors. For better performance on classification, discrimination costs have also been incorporated in a supervised manner [13, 14, 15], and classification models other than SRC have been used with sparse codes as inputs [16, 17, 18, 19, 20, 21]. However, the discrimination metrics used in existing methods are not geared to the mechanism of SRC, and the employment of an extra classification model leads to more parameters which increase the risk of over-fitting and break the unified framework of SRC.

In this paper, a new dictionary learning algorithm is proposed particularly for the purpose of classification with SRC. We optimize the dictionary by minimizing the hinge loss of residual difference between competing classes, which is inspired by the idea behind Learning Vector Quantization (LVQ) [22]. LVQ techniques were first applied to dictionary learning by Chen et al. [23] in an ad-hoc way; while here we adapt the philosophy of LVQ to SRC in a more principled manner (as formulated in Section 2), and hence name the algorithm as Learning Sparse Representation-based Classification (LSRC). Stochastic gradient descent is used in LSRC to circumvent the non-differentiable part of sparse code, and leads to updating rules (derived in Section 3) mimicking the "push" and "pull" actions of LVQ. Superior classification results are achieved using the proposed LSRC algorithm on

The work is supported by the U.S. Army Research Laboratory and U.S. Army Research Office under grant number W911NF-09-1-0383.

several HSI images (reported in the experiments in Section 4). We also discuss our contributions related to prior works (in Section 5) and draw concluding remarks (in Section 6).

2. PROBLEM FORMULATION

2.1. Sparse Representation-based Classification

Suppose we have a data set containing N labeled HSI pixels of m channels coming from C classes: $\{\mathbf{x}_i, y_i\}_{i=1...N}$, $\mathbf{x}_i \in \mathbb{R}^m$, $y_i \in \{1, ..., C\}$. A dictionary $\mathbf{D} \in \mathbb{R}^{m \times n}$ of size n used in SRC [4] is composed of C class-wise sub-dictionaries $\mathbf{D}^c \in \mathbb{R}^{m \times \frac{n}{C}}$ such that $\mathbf{D} = \left[\mathbf{D}^1, ..., \mathbf{D}^C\right]$. The sparse code $\boldsymbol{\alpha}_i \in \mathbb{R}^n$ for pixel $\mathbf{x}_i$ can be recovered by solving the following l_1 regularized problem as in compressive sensing [24]:

$$\boldsymbol{\alpha}_i = \arg\min_{\mathbf{z}} ||\mathbf{D}\mathbf{z} - \mathbf{x}_i||_2^2 + \lambda|\mathbf{z}|_1, \quad \text{with } \lambda > 0. \tag{1}$$

The sparse code can be decomposed into C sub-codes in a similar way: $\boldsymbol{\alpha}_i = \left[\boldsymbol{\alpha}_i^1; ...; \boldsymbol{\alpha}_i^C\right]$. SRC makes classification decision based on the residual of signal approximated by sub-code of each class: $r_i^c = ||\mathbf{e}_i^c||^2$, where $\mathbf{e}_i^c = \mathbf{x}_i - \mathbf{D}^c\boldsymbol{\alpha}_i^c$ is the class-wise reconstruction error. The predicted class label is obtained as

$$\hat{y}_i = \arg\min_c r_i^c. \tag{2}$$

Generally, our goal is to find an optimal dictionary $\mathbf{D}^*$ that achieves the best classification on the data set:

$$\mathbf{D}^* = \arg\min_{\mathbf{D} \in \mathcal{D}} \frac{1}{N} \sum_i I(\hat{y}_i \neq y_i), \tag{3}$$

where $I(\cdot)$ is the indicator function, and $\mathcal{D}$ is the matrix space with unit-length columns.

2.2. Objective Function with Insight from LVQ

Although closely related to our task of classification, Eq. (3) cannot be solved directly. A recent work in [23] applied the LVQ technique to learn the dictionary for SRC, which motivated us to design a more appropriative objective function based on the insight from LVQ.

LVQ [22] is a supervised learning algorithm which generates a codebook optimized for a prototype-based classifier. In testing, LVQ classifies a sample with the same label as the closest prototype in the codebook to it, which is essentially the same as the nearest neighbor classification. During training, LVQ (in its simplest version) iteratively goes through each training sample $\mathbf{x}_i$ and moves its nearest prototype $\mathbf{m}_{n(i)}$ towards or away from $\mathbf{x}_i$ based on whether $\mathbf{m}_{n(i)}$ belongs to the same class as $\mathbf{x}_i$:

$$\mathbf{m}_{n(i)} \leftarrow \begin{cases} \mathbf{m}_{n(i)} + \rho(\mathbf{x}_i - \mathbf{m}_{n(i)}), & \text{if } \mathbf{m}_{n(i)} \text{ has label } y_i \\ \mathbf{m}_{n(i)} - \rho(\mathbf{x}_i - \mathbf{m}_{n(i)}), & \text{otherwise} \end{cases} \tag{4}$$

where $0 < \rho < 1$ is a monotonically decreasing step size.

LVQ shares a common spirit with the SRC in several ways. Both of them represent data samples with a subset of elements in codebook or dictionary, and classify the samples based on the energy distribution in the selected prototypes or atoms. This justifies the attempt in [23] to use updating rules similar to Eq. (4) in learning dictionary for SRC. However, the underlying principles of sparse coding and vector quantization are quite different, which makes the performance of the ad-hoc approach in [23] unguaranteed.

A deeper insight into LVQ has been developed in [25] which regards the learning procedure as a scholastic gradient descent algorithm with a loss function defined on any *misclassified* sample $\mathbf{x}_i$:

$$\mathcal{L}_{LVQ}(\mathbf{x}_i, y_i) \propto ||\mathbf{x}_i - \mathbf{m}_{n(i)}^+||^2 - ||\mathbf{x}_i - \mathbf{m}_{n(i)}^-||^2, \tag{5}$$

where $\mathbf{m}_{n(i)}^+$ and $\mathbf{m}_{n(i)}^-$ are the nearest prototypes to $\mathbf{x}_i$ with label y_i and other than y_i, respectively. We adopt an objective function with a similar form as in Eq. (5) with the hope that the merits of LVQ can be exploited in building an SRC dictionary. Specifically, a hinge loss function is enforced on each data point:

$$\mathcal{L}_{LSRC}(\mathbf{x}_i, y_i; \mathbf{D}) = \max(0, r_i^{y_i} - r_i^{\hat{c}_i} + b), \tag{6}$$

where

$$\hat{c}_i = \arg\min_{c \in \{1,...,C\} \setminus y_i} r_i^c \tag{7}$$

is the most competitive class in reconstructing the signal excluding the true class y_i. b is a non-negative parameter controlling the "margin" between the classes. The loss function in Eq. (6) is zero when the residual of true class is smaller than any other class by at least an amount of b. Otherwise, it gives a penalty proportional to the residual difference between the true class and the most competitive "imposter" class. Intuitively, this loss function is also related to the misclassification rate of SRC. Thus, we can formulate the problem of LSRC as:

$$\mathbf{D}^* = \arg\min_{\mathbf{D} \in \mathcal{D}} \frac{1}{N} \sum_i \mathcal{L}_{LSRC}(\mathbf{x}_i, y_i; \mathbf{D}). \tag{8}$$

3. DICTIONARY OPTIMIZATION

Since the sample size N is usually large, stochastic gradient descent methods are favored to optimize a dictionary online when the objective function is an expectation over all the training samples [12]. The dictionary is first initialized with a reasonable guess $\mathbf{D}^0$ (through K-means or an unsupervised training for each class), and then it is updated iteratively by going through the whole data set multiple epochs until convergence. In the t-th iteration, a single sample $(\mathbf{x}_i, y_i)$[1] is drawn from the data set randomly and the dictionary is updated in the gradient direction of its cost term:

$$\mathbf{D}^t = \mathbf{D}^{t-1} - \rho^t \nabla_{\mathbf{D}} \mathcal{L}_{LSRC}(\mathbf{x}, y; \mathbf{D}^{t-1}), \tag{9}$$

[1] For simplicity, we drop all the data indices i hereafter.

where $\rho^t = \frac{\rho^0}{\sqrt{(t-1)/N+1}}$ is the step size at iteration t with initial value ρ^0. The gradient of hinge loss is

$$\nabla_{\mathbf{D}}\mathcal{L}(\mathbf{x}, y; \mathbf{D}) = \nabla_{\mathbf{D}} r^y - \nabla_{\mathbf{D}} r^{\hat{c}}, \quad \text{if } r^y - r^{\hat{c}} + b > 0, \quad (10)$$

and zero or undefined otherwise. We can ignore the case of undefined gradient, because it occurs with very low probability in practice (only when $r^y - r^{\hat{c}} + b = 0$) and thus will not affect the convergence of stochastic gradient descent as long as a suitable step size is chosen [25].

To evaluate the gradient of r^c for a particular class c, we first find its derivative with respect to the (i, j)-th element of $\mathbf{D}$ as

$$\begin{aligned} \frac{\partial r^c}{\partial d_{ij}} &= -2\mathbf{e}^{cT}\frac{\partial \mathbf{D}\mathbf{P}_c\boldsymbol{\alpha}}{\partial d_{ij}} \\ &= -2\mathbf{e}^{cT}\left[\mathbf{P}_c(j,j)\alpha_j\mathbf{u}_i + \mathbf{D}\mathbf{P}_c\frac{\partial \boldsymbol{\alpha}}{\partial d_{ij}}\right], \quad (11) \end{aligned}$$

where $\mathbf{P}_c$ is a $n \times n$ diagonal matrix with 1 at positions corresponding to class c and 0 otherwise, and $\mathbf{u}_i$ is a $m \times 1$ unit column vector with the i-th element equal to 1.

The sparse code $\boldsymbol{\alpha}$ is an implicit function of $\mathbf{D}$, and it has been shown differentiable [20, 26] with respect to any dictionary atom $\mathbf{d}_j$ with index j in the active set $\Lambda = \{j|\alpha_j \neq 0\}$. For the other atoms, the gradient is zero with overwhelming probability and thus can be ignored for the same reason mentioned above. Directly using the result given in [26], we can find the sparse code derivative as:

$$\frac{\partial \boldsymbol{\alpha}_\Lambda}{\partial \mathbf{D}_\Lambda} = -\mathbf{A}^{-1}\frac{\partial[\mathbf{D}_\Lambda^T(\mathbf{D}_\Lambda\boldsymbol{\alpha}_\Lambda - \mathbf{x})]}{\partial \mathbf{D}_\Lambda}, \quad (12)$$

where $\boldsymbol{\alpha}_\Lambda$ and $\mathbf{D}_\Lambda$ denote the sparse coefficients and dictionary columns corresponding to the active set Λ. $\mathbf{A} = \mathbf{D}_\Lambda^T\mathbf{D}_\Lambda$, and in practice we set $\mathbf{A} = \mathbf{D}_\Lambda^T\mathbf{D}_\Lambda + \epsilon \cdot \mathbf{I}$ to ensure the stability of the inverse of $\mathbf{A}$, where ϵ is a small positive constant. It is then easy to obtain for any $j \in \Lambda$:

$$\frac{\partial \boldsymbol{\alpha}}{\partial d_{ij}} = \mathbf{P}_\Lambda[\mathbf{A}^{-1}(:, \Lambda^{-1}(j)) \cdot e_i - \mathbf{A}^{-1}\mathbf{D}_\Lambda^T(i,:) \cdot \alpha_j], \quad (13)$$

where $\mathbf{P}_\Lambda \in \mathbb{R}^{n \times |\Lambda|}$, $\mathbf{P}_\Lambda(j,k) = I(j = \Lambda(k))$, $\Lambda(k)$ denotes the k-th element of Λ sorted in ascending order, $\Lambda^{-1}(\cdot)$ is the inverse function of $\Lambda(\cdot)$, and $\mathbf{e} = \mathbf{x} - \mathbf{D}\boldsymbol{\alpha} = \mathbf{x} - \mathbf{D}_\Lambda\boldsymbol{\alpha}_\Lambda$.

Combining all the equations above and after some manipulations, we get the gradient of r^c with respect to the j-th dictionary atom for any $j \in \Lambda$:

$$\nabla_{\mathbf{d}_j} r^c = -2\alpha_j I(cls(j) = c) \cdot \mathbf{e}^c - 2\beta^c_{\Lambda^{-1}(j)} \cdot \mathbf{e} + 2\alpha_j \mathbf{D}_\Lambda \boldsymbol{\beta}^c, \quad (14)$$

where $cls(j)$ is the class label for j-th dictionary atom, and $\boldsymbol{\beta}^c = \mathbf{A}^{-1}\mathbf{P}_\Lambda^T\mathbf{P}_c\mathbf{D}^T\mathbf{e}^c$. Thus, the update for each atom in the active set Λ is:

$$\begin{aligned} \Delta\mathbf{d}_j^t &= \mathbf{d}_j^t - \mathbf{d}_j^{t-1} \\ &= 2\rho^t\Big[\alpha_j I(cls(j){=}y)\cdot\mathbf{e}^y - \alpha_j I(cls(j){=}\hat{c})\cdot\mathbf{e}^{\hat{c}} \\ &\quad + (\beta^y_{\Lambda^{-1}(j)} - \beta^{\hat{c}}_{\Lambda^{-1}(j)})\cdot\mathbf{e} - \alpha_j\mathbf{D}_\Lambda(\boldsymbol{\beta}^y - \boldsymbol{\beta}^{\hat{c}})\Big]. \quad (15) \end{aligned}$$

Algorithm 1 Dictionary learning with LSRC

Require: labeled data set $\mathcal{S} = \{\mathbf{x}_i, y_i\}$, sparse regularization coefficient λ, margin b
Ensure: dictionary $\mathbf{D}$

```
initialize D
set t = 1
while not converge do
    randomly permute data set S
    for each (x, y) ∈ S do
        find sparse code α with Eq. (1)
        find r^c = ||x − D^c α^c||^2 for any c = 1...C
        find ĉ with Eq. (7)
        if r^y − r^ĉ + b > 0 then
            d_j ← d_j + Δd_j for any j ∈ Λ by Eq. (15)
            d_j ← d_j / ||d_j|| for any j ∈ Λ
        end if
        t ← t + 1
    end for
end while
return D
```

The resultant dictionary atoms are projected to unit length to ensure $\mathbf{D} \in \mathcal{D}$. The overall method of LSRC is summarized in Algorithm 1. The first two terms in Eq. (15) have the effects of "pulling" the active dictionary atoms of correct class towards the signal, and "pushing" the active dictionary atoms of the most competitive wrong class away from the signal, which is similar to what has been done in [23] to mimic the procedure used in the LVQ. The third and fourth terms in Eq. (15) are unique in our LSRC method. They bring the overall reconstruction error and every active atom as ingredients for dictionary updating, which makes sense as the sparse code is jointly determined by all the atoms in the active set.

4. EXPERIMENTAL RESULTS

We test the proposed method on three benchmark HSI images: the Indian Pines [27], the University of Pavia, and the Center of Pavia [28]. The experiments setup and classification accuracies are listed in Table 1. We compare the performance of SRC with dictionaries obtained from the full training set ("Full") [5], the K-means clustering ("K-means"), the unsupervised training ("Unsup ") [12] [2], the ad-hoc LVQ approach ("LVQ") [23], and our method ("LSRC"). Accuracies are also reported for the SVM classifiers with a linear kernel ("SVM") and an RBF-kernel ("KSVM"), the later of which is known to give the state-of-the-art results on high dimensional HSI data [2]. We follow the same way as in [5] in pre-processing the multi-band features. Since our focus is dictionary learning, all the results shown are based on pixel-wise classification. Our learned dictionaries have a small size of only 5 atoms per

[2] Our dictionary is not as good as the one learned in [12] in terms of sparse reconstruction, but it gives more discriminative sparse codes for SRC.

Table 1. Experiment settings and classification accuracies (%) on three HSI images.

Image	#class	#train/test	Parameters	Metric	Full	K-means	Unsup	LVQ	LSRC	SVM	KSVM
Indian Pines (200 bands)	16	1043 / 9323	120 iterations, $\rho^0 = 0.01$, $\lambda = 0.1, b = 0.2$	OA	82.96	69.87	66.41	75.39	83.84	74.44	**84.52**
				AA	76.66	72.11	67.59	73.97	77.69	65.49	**79.24**
				κ	0.805	0.662	0.624	0.723	0.816	0.708	**0.823**
University of Pavia (103 bands)	9	3921 / 40002	10 iterations, $\rho^0 = 0.001$, $\lambda = 0.05, b = 0.3$	OA	78.31	68.01	64.57	73.24	**81.08**	67.28	79.15
				AA	86.78	77.05	71.66	82.91	85.26	79.66	**87.66**
				κ	0.726	0.596	0.549	0.666	**0.754**	0.599	0.737
Center of Pavia (102 bands)	9	5536 / 97940	20 iterations, $\rho^0 = 0.001$, $\lambda = 0.1, b = 0.3$	OA	97.45	95.86	95.91	96.85	**97.93**	95.68	96.13
				AA	95.41	91.35	91.95	93.93	**96.11**	93.77	85.29
				κ	0.954	0.925	0.926	0.943	**0.962**	0.923	0.928

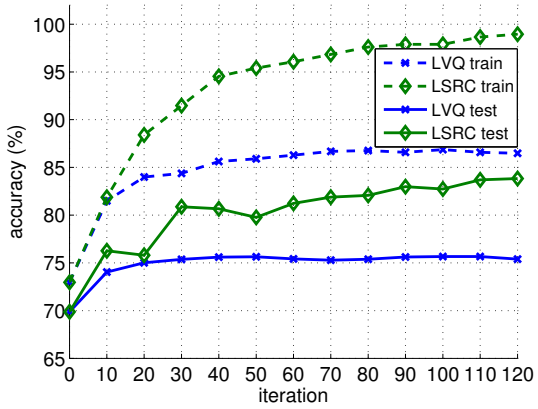
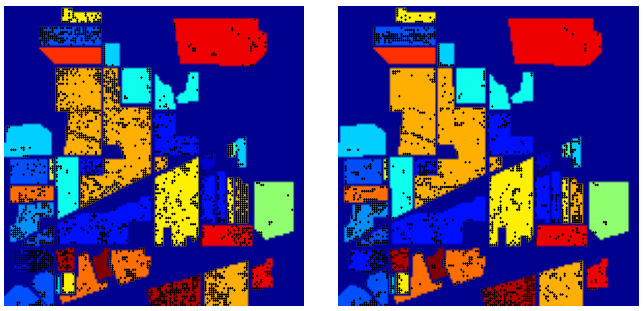


Fig. 1. Accuracies change during training iterations for both "LVQ" and "LSRC" on the Indian Pines image.



(a) "LVQ" (b) "LSRC"

Fig. 2. Classification results on the Indian Pines image. Color encodes true labels, and black dots denote misclassification.

Table 2. Effect of parameter b on the Indian Pines image.

b	0.0	0.1	0.2	0.3	0.4
Train Acc. (%)	99.81	99.14	98.85	98.27	97.60
Test Acc. (%)	81.30	83.51	**83.84**	83.19	82.88

class – a great reduction compared with the full training set used in the models of "Full" and "KSVM", yet the overall accuracy (OA), class-averaged accuracy (AA) and κ coefficient [29] achieved by LSRC are higher than using the "Full" set and other dictionary learning methods. Although built on linear input space, our method attains better performance than the nonlinear "KSVM" except for the small Indian Pines data set, on which SVM shows a better generalization capability.

Fig. 1 demonstrates that our learning algorithm effectively reduces both training and test errors during training, and converges to much higher accuracies than "LVQ". The labels of the Indian Pines image predicted using the "LVQ" and "LSRC" methods are also given in Fig. 2 for comparison.

The effect of tuning margin parameter b is examined in Table 2. A too small value of b leads to over-fitting to training set, while a too large value leads to bias of classification objective. A proper value of b is determined using part of training data as a validation set.

5. RELATION TO PRIOR WORK

The work presented here follows the classical framework of SRC proposed by Wright et al [4], and focuses on the less investigated problem of learning a dictionary well suited for SRC on HSI data. We take advantage of the underlying principle of Kohonen's LVQ [22] algorithm and adapt it to the dictionary design for SRC, leading to a novel LSRC algorithm which is more sound theoretically and more effective experimentally than the ad-hoc combination done previously by Chen et al [23].

6. CONCLUSION

A new dictionary design method for HSI classification is proposed by optimizing a hinge loss function sharing the same spirit with LVQ. Our stochastic gradient decent-based algorithm mimics the updating rule of LVQ, but performs substantially better than the ad-hoc adaptation of LVQ as well as other existing dictionary learning approaches. Classification results achieved with the obtained compact dictionaries on three HSI images are comparable to or better than the kernel SVM-based classifier. In future work, we will incorporate spatial information into the current classification framework and apply our method to other image modalities.

7. REFERENCES

[1] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, pp. 1778–1790, 2004.

[2] G. Camps-valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, pp. 1351–1362, 2005.

[3] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 93–97, 2006.

[4] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[5] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3973–3985, 2011.

[6] S. F. Cotter, "Sparse representation for accurate classification of corrupted and occluded facial expressions," in *ICASSP*, 2010, pp. 838–841.

[7] B. C. Haris and R. Sinha, "Sparse representation over learned and discriminatively learned dictionaries for speaker verification," in *ICASSP*, 2012, pp. 4785–4788.

[8] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *ICASSP*, 1999, pp. 2443–2446.

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

[10] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Kernel dictionary learning," in *ICASSP*, 2012, pp. 2021–2024.

[11] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *NIPS*, 2007, pp. 801–808.

[12] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *ICML*, 2009, pp. 689–696.

[13] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *CVPR*, 2010, pp. 2691–2698.

[14] F. Rodriguez and G. Sapiro, "Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries," DTIC Document, Tech. Rep., 2008.

[15] J. Yang, J. Wang, and T. S. Huang, "Learning the sparse representation for classification," in *ICME*, 2011, pp. 1–6.

[16] J. Yang, K. Yu, and T. S. Huang, "Supervised translation-invariant sparse coding," in *CVPR*, 2010, pp. 3517–3524.

[17] D. Bradley and J. A. D. Bagnell, "Differentiable sparse coding," in *NIPS*, December 2008.

[18] D.-S. Pham and S. Venkatesh, "Joint learning and dictionary construction for pattern recognition," in *CVPR*, 2008.

[19] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *NIPS*, 2008, pp. 1033–1040.

[20] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 791–804, 2012.

[21] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *CVPR*, 2011, pp. 1697–1704.

[22] T. Kohonen, "Improved versions of learning vector quantization," in *International Joint Conference on Neural Networks*, vol. 1, 1990, pp. 545–550.

[23] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Discriminative dictionary design using LVQ for hyperspectral image classification," in *IEEE 4th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, 2012.

[24] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[25] L. Bottou, "Stochastic learning," in *Advanced Lectures on Machine Learning*, ser. Lecture Notes in Artificial Intelligence, LNAI 3176, O. Bousquet and U. von Luxburg, Eds. Springer Verlag, 2004, pp. 146–168.

[26] J. Yang, Z. Wang, Z. Lin, X. Shu, and T. Huang, "Bilevel sparse coding for coupled feature spaces," in *CVPR*, 2012.

[27] D. Landgrebe, "AVIRIS NW Indiana's Indian Pines 1992 data set," 1992, https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html.

[28] A. Plaza, J. Benediktsson, J. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri *et al.*, "Recent advances in techniques for hyperspectral image processing," *Remote Sensing of Environment*, vol. 113, pp. S110–S122, 2009.

[29] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis: An Introduction*, 4th ed. Springer-Verlag, 2006.