

Discriminative Tensor Sparse Coding for Image Classification

Yangmuzi Zhang¹
ymzhang@umiacs.umd.edu

Zhuolin Jiang²
zhuolin.jiang@huawei.com

Larry S. Davis¹
lsd@umiacs.umd.edu

¹ University of Maryland
College Park, USA

² Noah's Ark Lab
Huawei Technologies

Abstract

A novel approach to learn a discriminative dictionary over a tensor sparse model is presented. A structural incoherence constraint between dictionary atoms from different classes is introduced to promote discriminating information into the dictionary. The incoherence term encourages dictionary atoms to be as independent as possible. In addition, we incorporate classification error into the objective function of dictionary learning. The dictionary is learned in a supervised setting to make it useful for classification. A linear multi-class classifier and the dictionary are learned simultaneously during the training phase. Our approach is evaluated on three types of public databases, including texture, digit, and face databases. Experimental results demonstrate the effectiveness of our approach.

1 Introduction

Sparse models have been successfully applied to many problems in image processing, computer vision, and machine learning. Many algorithms [9, 25] have been proposed to learn an over-complete and compact dictionary based on such models. In general, the input feature representations to these approaches are based on traditional vector descriptors. As pointed out in recent work [8, 17], vectorizing the original data structure, however, may destroy some inherent ordering information in the data. One example are the $n \times n$ symmetric positive semi-definite matrices. The kernel matrix in many popular kernelized machine learning algorithms [14] is of this type. Another example is the diffusion tensor (a 3×3 positive definite matrix) which is used to represent voxels in medical imaging. In computer vision, the region covariance feature, introduced in [19], is an image descriptor that captures natural correlations amongst multiple features. Hence, there has been growing interest in the development of sparse coding for positive definite descriptors. In [20], the problem of sparse coding within the space of symmetric positive definite matrices is tackled by embedding Riemannian manifolds into kernel Hilbert spaces. [17] proposed tensor sparse coding on positive definite matrices, which keeps descriptors in their original space and uses a set of randomly selected training samples as the dictionary. It successfully extended sparse coding

techniques to the space of positive definite matrices. However, little research has been done to learn a discriminative and compact dictionary over such spaces.

We present a discriminative dictionary learning method for tensor sparse coding. Rather than simply using a subset of region covariance descriptors for training images as the dictionary [17], we learn a discriminative dictionary from the training set. A structural incoherence term is introduced into the dictionary learning process to regularize the incoherence between different sub-dictionaries, which increases the discriminativeness of the learned dictionary. We further incorporate classification error into the objective function to make the learned dictionary effective for classification tasks. Instead of learning multiple classifiers for each pair of classes [11, 12, 22], a linear multi-class classifier can be easily obtained during the training process. Unlike [18], which focuses on the reconstructive capability of a dictionary, the dictionary learned by our approach has both good reconstruction and discrimination capabilities. Based on this learned high-quality dictionary, we are able to obtain discriminative tensor sparse representations. Classification can be efficiently performed on these representations using the learned multi-classifier as it only involves matrix multiplication. The main contributions of this paper are:

- A supervised learning approach to construct a discriminative and reconstructive dictionary, which is used to obtain discriminative tensor sparse representations for classification.
- A method to effectively learn a compact discriminative dictionary and a universal multi-class linear classifier (for all categories) simultaneously.

The paper is organized as follow. Section 2 gives a brief review of tensor sparse coding and dictionary learning. Section 3 describes our approach to learn a discriminative and reconstructive dictionary. Experimental results are presented in Section 4. Section 5 concludes the paper.

1.1 Related Work

The region covariance descriptor was first proposed in [19] to encode an image region. The descriptor is the covariance matrix of the d -dimensional feature vectors at each pixel within a region. Given an image I , let Φ define a function that extracts a d -dimensional feature vector z_i from each pixel $i \in I$, i.e. $\Phi(I, x_i, y_i) = z_i$, where $z_i \in R^d$, and (x_i, y_i) is the location of the i^{th} pixel. Φ can be any mapping such as intensity, gradient, filter responses, etc. F is the $W \times H \times d$ dimensional features extracted from I , i.e. $F(x, y) = \Phi(I, x, y)$. For a given rectangular region $R \subset F$, $\{z_k\}_{k=1,2,\dots,N}$ is the set of d -dimensional features of all N points inside the region R . Then the region covariance descriptor $C_R \in R^{d \times d}$ is computed by:

$$C_R = \frac{1}{N-1} \sum_{k=1}^N (z_k - \mu)(z_k - \mu)^T \quad (1)$$

where μ is the mean of all points. The region covariance descriptor fuses multiple features which might be naturally correlated to describe a region or cuboid in images or videos. The average filter during covariance computation also helps to filter out noise that corrupts individual samples. It has become a popular descriptor for human detection [20], tracking [20], object detection [6, 18], action recognition [23], and pedestrian detection [21].

The tensor sparse model introduced in [17] learns a sparse representation over positive definite matrices. In [6], the Stein kernel is introduced to embed the space of symmetric positive definite matrices into a kernel Hilbert space. These algorithms, however, take the entire

training set as the dictionary. Tensor sparse coding with a large dictionary is computationally expensive when the number of training samples is large. Hence learning compact dictionaries for tensor sparse coding is desirable. In [18], a dictionary learning method is developed based on the K-SVD algorithm [1]. However, the dictionary atoms are updated independently, and the updating aims to reduce reconstruction errors. So the learned dictionary may not perform well for classification tasks.

Compared to previous work, our approach learns a discriminative and reconstructive dictionary effectively. With respect to this dictionary, discriminative sparse representations can be obtained by solving a determinant maximization (MAXDET) problem. We simultaneously train a linear classifier along with dictionary learning, resulting in a learned dictionary good for classification.

1.2 Notation

\mathbb{S}_d^+ denotes the space of $d \times d$ symmetric positive semi-definite matrices, while \mathbb{S}_d^{++} refers to the space of strictly positive definite matrices. $A \succ 0$ ($A \succeq 0$) means A is positive (semi)definite. $A \succ B$ ($A \succeq B$) indicate that $(A - B)$ is positive (semi)definite. Let $S = \{S_l\}_{l=1}^N$ denote the data set, $S_l \in \mathbb{S}_d^{++}$. K is the number of categories. Then the dictionary is $A = [A^1, A^2, \dots, A^K]$. $A^i = [\mathbf{a}_1^i, \mathbf{a}_2^i, \dots, \mathbf{a}_{K_i}^i]$ denotes the sub-dictionary for class i . K_i is the number of atoms within that sub-dictionary, and each dictionary atom $\mathbf{a}_t^i \in \mathbb{S}_d^{++}$. $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}^{K \times N}$ represents the sparse representation for S , with \mathbf{x}_l for S_l . Then the reconstructed data \hat{S} is:

$$\hat{S} = X \otimes A \quad (2)$$

$$\mathbf{x}_l \otimes A = \sum_{i=1}^K x_l^i \otimes A^i = \sum_{i=1}^K \sum_{t=1}^{K_i} x_{lt}^i \mathbf{a}_t^i \quad (3)$$

with x_l^i denoting the representation coefficients for S_l corresponding to sub-dictionary A^i , and x_{lt}^i is the coefficient corresponding to dictionary atom \mathbf{a}_t^i .

The LogDet divergence $D_{ld} : \mathbb{S}_d^+ \times \mathbb{S}_d^{++} \rightarrow \mathbb{R}_+$ is defined as:

$$D_{ld}(X, Y) = \text{tr}(XY^{-1}) - \log \det(XY^{-1}) - n \quad (4)$$

This measures the distance between two positive definite matrices [3, 17].

2 Tensor Sparse Coding and Dictionary Learning

In this section, we give a brief review of tensor sparse coding and algorithms for learning an over-complete dictionary. Given a dictionary A and a data set S , the tensor sparse coding problem in [17] is formulated as:

$$\begin{aligned} \min_{\mathbf{x} \geq 0} \quad & D_{ld}(X \otimes \mathbf{A}, S) + \lambda \|\mathbf{x}\|_1 \\ \text{s.t.} \quad & 0 \leq X \otimes \mathbf{A} \leq S, \end{aligned} \quad (5)$$

where D_{ld} is the LogDet divergence defined in (4), and λ is the regularization parameter inducing sparsity on X . The problem can be reduced to a MAXDET problem [17] and solved by utilizing CVX [5].

In [17], the dictionary A was constructed by simply selecting a subset of the training set for the classification setting. In [18], a dictionary from the training data is learned via minimizing a reconstruction error. Each dictionary atom is updated based on a gradient descent or an alternating formulation method. Minimizing the reconstruction error in problem (5), however, may not be optimal for classification tasks. We will show that by introducing structural incoherence into the objective function of dictionary learning, the discriminability of the learned dictionary can be greatly improved. Meanwhile, by incorporating classification error into the dictionary learning process, we can obtain a linear multi-class classifier jointly, which will improve efficiency of classification performance and reduce computation time.

3 Discriminative Tensor Sparse Coding

To enhance the discriminativeness of tensor sparse codes, we want to learn a reconstructive and discriminative dictionary. Each sub-dictionary corresponds to one class. The dictionary will be more discriminative if each sub-dictionary is much more representative and specific to a particular class of images. Hence we explicitly encourage independence between dictionary atoms from different sub-dictionaries. We subsequently leverage the supervised label information of input signals into the optimization problem.

3.1 Structural Dictionary Learning 1 (SDL1)

The quality of the dictionary influences the discriminativeness of the tensor sparse representations. Updating each dictionary atoms separately does not result in sufficient discriminative information in the sub-dictionaries. Following [2, 15], we introduce structural incoherence into sub-dictionary atoms. Incoherence will promote dictionary atoms from different classes to be independent from each other; thus it leads to sparse and discriminating representations for images.

Based on the above analysis, we add a structural incoherence regularization term into the objective function. Given a training data set $S = \{S_l\}_{l=1}^N$, we will learn a dictionary $A = \{A^i\}_{i=1}^K$, with sub-dictionary $A^i = [\mathbf{a}_1^i, \mathbf{a}_2^i, \dots, \mathbf{a}_{K_i}^i]$ for class i . The problem is formulated as:

$$\begin{aligned} \min_{A, X} \quad & \sum_{l=1}^N D_{ld}(\mathbf{x}_l \otimes A, S_l) + \lambda \|\mathbf{x}\|_1 + \eta \sum_{i \neq j, s, t} \|(\mathbf{a}_s^i)^T \mathbf{a}_t^j\|_F^2 \\ \text{s.t.} \quad & \mathbf{x}_l \geq \mathbf{0} \quad \forall l \\ & \mathbf{a}_t^i, \mathbf{a}_s^j \geq \mathbf{0} \quad \forall i, t, j, s \\ & \mathbf{0} \preceq \mathbf{x}_l \otimes A \preceq S_l \quad \forall l \end{aligned} \quad (6)$$

The first two terms are the reconstruction error and the sparsity regularization. The last term sums up the Frobenius norms between every two dictionary atoms $\mathbf{a}_s^j, \mathbf{a}_t^i$ which belong to different sub-dictionaries A^j and A^i . λ, η are penalty parameters balancing reconstruction error, sparsity, and dictionary structural incoherence.

3.2 Structural Dictionary Learning 2 (SDL2)

As pointed out in [9], the learned dictionary can be more adaptive to classification tasks when minimizing the classification error in the objective function of dictionary learning. A linear multi-classifier $f(x; W) = Wx$ is used for classification. W denotes the linear classifier's

parameters. Hence, the classification error can be explicitly included in the objective function during the dictionary learning. The classifier will be learned through the training process, as well. The objective function is formulated as below:

$$\begin{aligned} \min_{A, X, W} \quad & \sum_{l=1}^N D_{ld}(\mathbf{x}_l \otimes A, S_l) + \lambda \|\mathbf{x}\|_1 + \eta \sum_{i \neq j, s, t} \|(\mathbf{a}_s^i)^T \mathbf{a}_t^i\|_F^2 + \gamma \|H - WX\|_2^2 \quad (7) \\ \text{s.t.} \quad & \mathbf{x}_l \geq \mathbf{0} \quad \forall l \\ & \mathbf{a}_t^i, \mathbf{a}_s^j \succeq \mathbf{0} \quad \forall i, t, j, s \\ & \mathbf{0} \preceq \mathbf{x}_l \otimes A \preceq S_l \quad \forall l \end{aligned}$$

where the term $\|H - WX\|_2^2$ represents the classification error. $H = [h_1, h_2, \dots, h_N] \in \mathbb{R}^{K \times N}$ denotes the label matrix. The column vector $h_i = [0, 0, \dots, 1 \dots 0, 0]^T \in \mathbb{R}^K$ is a label vector for sample i . The position of 1 indicates its class index. γ controls the contribution of the classification error regularization term in the optimization process.

3.3 Optimization

In this section, we only describe the optimization procedures for SDL2 here. To solve SDL1, we utilize a similar procedure excluding the calculation of classifier W . The classifier will be calculated directly through Equation (16) using the final result X for SDL1.

The dictionary learning problem is convex in any one of the elements in the triple (A, X, W) when the others are fixed. Hence, the optimization can be divided into three subproblems: (1) updating dictionary atoms with fixed X and W ; (2) solving the max determinant problem with fixed A and W ; (3) computing a linear classifier with fixed X and A . If we set $\gamma = 0$ in subproblem (2), this is exactly the optimization procedure for problem (6). The complete optimization is summarized in Algorithm 1.

3.3.1 Dictionary Update with fixed W and X

Following [18], we use a steepest descent approach to update each dictionary atom \mathbf{a}_t^i . With fixed W and X , the objective function (7) can be rewritten as a function of \mathbf{a}_t^i as below:

$$f(\mathbf{a}_t^i) = \sum_l D_{ld}(\mathbf{x}_l \otimes A, S_l) + \lambda \|\mathbf{x}\|_1 + \eta \sum_{j \neq i} \sum_s \|(\mathbf{a}_s^j)^T \mathbf{a}_t^i\|_F^2 + \gamma \|H - WX\|_2^2 \quad (8)$$

$$= \sum_l \text{tr}(x_{tl}^i \mathbf{a}_t^i S_l^{-1}) - \log \det \hat{S}_j + \eta \sum_{j \neq i} \sum_s \text{tr}((\mathbf{a}_t^i)^T \mathbf{a}_s^j (\mathbf{a}_s^j)^T) \mathbf{a}_t^i + C, \quad (9)$$

where C includes all those terms independent of \mathbf{a}_t^i . When updating one dictionary atom, other atoms remain fixed. The gradient descent direction $-\nabla f(\mathbf{a}_t^i)$ is:

$$-\nabla f(\mathbf{a}_t^i) = \sum_l x_{tl}^i (\hat{S}_l^{-1} - S_l^{-1}) - 2\eta \sum_{j \neq i} \sum_s (\mathbf{a}_s^j)^T \mathbf{a}_s^j \mathbf{a}_t^i \quad (10)$$

Since $A_i \in \mathbb{S}_d^{++}$, we need to ensure that $d\mathbf{a}_t^i \succeq \mathbf{0}$. Meanwhile, from $\hat{S}_j \preceq S_j$, we know that $\hat{S}_j^{-1} \succeq S_j^{-1}$, yielding the first term in Equation (10) positive semidefinite. Thus the gradient direction $d\mathbf{a}_t^i$ is given by:

$$d\mathbf{a}_t^i = \begin{cases} \sum_l x_{tl}^i (\hat{S}_l^{-1} - S_l^{-1}) - 2\eta \sum_{j \neq i} \sum_s (\mathbf{a}_s^j)^T \mathbf{a}_s^j \mathbf{a}_t^i, & \nabla f(\mathbf{a}_t^i) \leq \mathbf{0} \\ \sum_l x_{tl}^i (\hat{S}_l^{-1} - S_l^{-1}), & \text{otherwise} \end{cases} \quad (11)$$

Combining all dictionary atoms together, the dictionary is updated as below. The updating step size $\alpha \geq 0$ is determined by a line search technique.

$$dA = [dA^1, dA^2, \dots, dA^K] = [d\mathbf{a}_1^i, d\mathbf{a}_2^i, \dots, d\mathbf{a}_{K_i}^i] \quad (12)$$

$$A \leftarrow A + \alpha dA \quad (13)$$

3.3.2 Solving X with fixed A and W

With fixed A and W , the subproblem to solve X can be formulated as:

$$\begin{aligned} \min_X \quad & \sum_{l=1}^N D_{ld}(\mathbf{x}_l \otimes A, S_l) + \lambda \|\mathbf{x}\|_1 + \gamma \|H - WX\|_2^2 \\ \text{s.t.} \quad & \mathbf{x}_l \geq \mathbf{0} \forall l, \quad \mathbf{a}_l^i \succeq \mathbf{0} \forall i, t, \quad \mathbf{0} \preceq \mathbf{x}_l \otimes A \preceq S_l \forall l \end{aligned} \quad (14)$$

This is a problem of sparse decomposition over positive definite matrices. As shown in [17], this problem is convex and well-behaved. It falls into the general class of optimization problems known as MAXDET problems. CVX [5] is used to solve this problem.

3.3.3 Calculating W with fixed A and X

We use the multivariate ridge regression model [4, 24] to obtain the linear classifier W :

$$\hat{W} = \arg \min_W \|H - WX\|_2^2 + \lambda_w \|W\|_2^2 \quad (15)$$

where H is the class label matrix of X . Fixing X and A , the multi-class classifier can be easily derived as:

$$W = HX^T (XX^T + \lambda_w I)^{-1} \quad (16)$$

Algorithm 1 Structural Dictionary Learning

Input: Data S , and Parameters $\alpha, \hat{\lambda}, \eta, \gamma$

Output: A, W

Initialize: Initial Dictionary A , Classifier W , $\varepsilon_A = 10^{-3}$

while not converged, $i \leq \text{maxIter}A$ **do**

 fix A and W , solve X for MAXDET problem (14)

 fix A and X , calculate W according to (16)

 fix W and X , calculate dA according to (11)

 update A with (12) and (13)

 check the convergence condition:

$$\|S - X \otimes A\|_\infty < \varepsilon_A$$

end while

3.3.4 Initialization

To initialize the dictionary A , we randomly sample A from the training data. The initialization of subdictionary A_i is a subset of training data belongs to class i . To initialize W , we first solve problem (14) with $\gamma = 0$ using the initialized dictionary. Then W is calculated according to Equation (16).

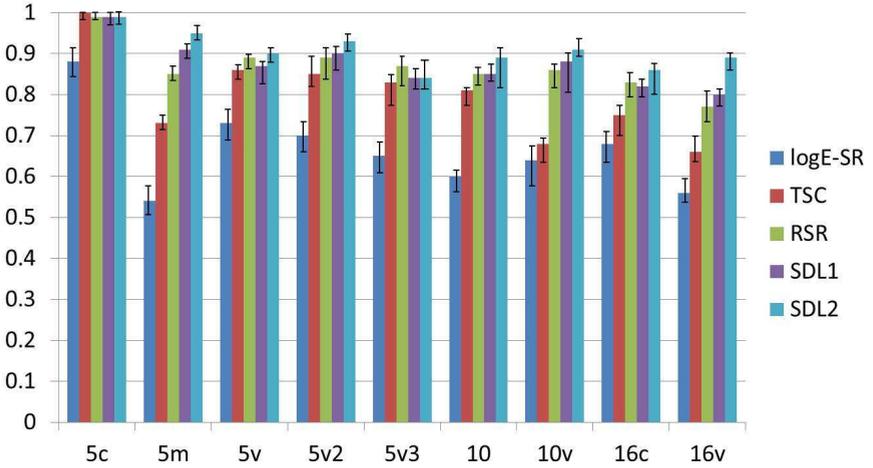


Figure 1: Texture classification results on the Brodatz dataset. Each group (five bars) indicates the recognition accuracy for one test scenario. Each bar in a group corresponds to one method.

3.4 Classification

After obtaining the dictionary, a tensor sparse representation X_{test} for test data S_{test} is calculated by solving Equation (14) with $\gamma = 0$. The representation \mathbf{x}_l for test sample l is the l -th column in X_{test} . Using the classifier W , the label for test sample l is given by $k = \arg \max_k (\mathbf{s} = W\mathbf{y}_l)$ which corresponds to the index of the largest element in the class label vector \mathbf{s} .

4 Experiments

We evaluate our approach on three datasets: Brodatz texture dataset [16], USPS digital dataset [7], and AR face dataset [13]. Our approach is compared with several state-of-the-art algorithms including tensor sparse coding (TSC) [17], tensor sparse coding with dictionary learning (TSCwD) [18], logE-SR [23] and Riemannian sparse representation (RSR) [6].

4.1 Texture Dataset

We follow the protocol in [6] and create mosaics under nine test scenarios with various number of classes, including 5-textures ('5c', '5m', '5v', '5v2', '5v3'), 10-textures ('10', '10v'), and 16-textures ('16c', '16v'). Spatial derivatives have been shown to be useful for texture characterization in [17, 19]. The feature vector $F(x, y)$ for any pixel with gray scale intensity $I(x, y)$ is $[I(x, y), |\frac{\partial I}{\partial x}|, |\frac{\partial I}{\partial y}|, |\frac{\partial^2 I}{\partial x^2}|, |\frac{\partial^2 I}{\partial y^2}|]$. Each image is 256×256 , and 32×32 blocks are cut out, yielding 64 data samples per image; a 5×5 region covariance descriptor is computed for each sample. For each scenario, we randomly select 5 data samples as training and the rest for testing. The random selection is repeated 10 times.

Figure 1 illustrates the classification results under different testing scenarios. We compare SDL1 and SDL2 against logE-SR [23], TSC [17, 18], and RSR [6]. The average accuracy of SDL2 achieves the highest score on all test scenarios except for '5v3' and '5c'. We use $\alpha = 0.0001, \lambda = 0.001, \eta = 0.0001$ in our experiments. However, our maximum classification results over 10 runs are comparable to the best scores. Figure 2 shows an example

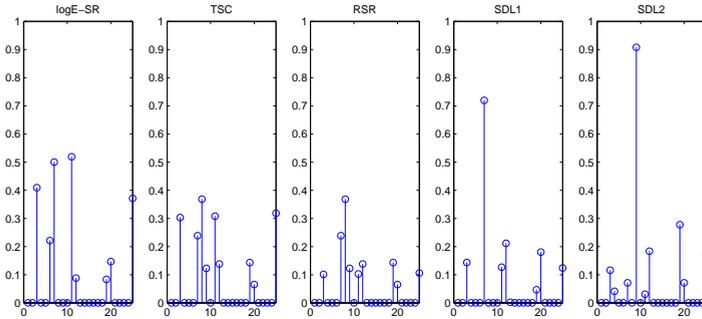


Figure 2: An example of tensor sparse codes using different approaches. X axis indicates the dimension of sparse codes, Y axis indicates the average tensor sparse codes for testing samples (first 10 blocks) from the 2nd class in '5v2'.

of tensor sparse codes on '5v2'. The indices 6~10 on the X-axis corresponds to the sub-dictionary for the 2nd class. The coefficients highly peak within the 2nd class in our method. We can see that SDL2 provides the most discriminative sparse codes among all methods.

4.2 Digit Dataset

The USPS Dataset is a handwritten digit database containing 9298 16×16 handwritten digit images. We follow the protocol in [26], using the images of digits 1,2,3,4 and randomly selecting 200 images for each category. The percentage of training samples ranges from 10% to 60%. For tensor sparse coding methods, a 9×9 covariance descriptor is used to describe a digit image, using the feature below:

$$F(x,y) = [I(x,y), |G_{0,0}(x,y)|, \dots, |G_{0,3}(x,y)|, |G_{1,0}(x,y)|, \dots, |G_{1,3}(x,y)|] \quad (17)$$

where $I(x,y)$ is the intensity value at position (x,y) and $G_{u,v}(x,y)$ is the response of a 2D Gabor wavelet [10] centered at (x,y) with orientation u and scale v :

$$G_{u,v} = \frac{k_v^2}{4\pi^2} \sum_{t,s} e^{-\frac{k_v^2}{8\pi^2}((x-s)^2+(y-t)^2)} (e^{ik_v((x-t)\cos(\theta_u)+(y-s)\sin(\theta_u))} - e^{-2\pi^2}) \quad (18)$$

with $k_v = \frac{1}{\sqrt{2^{v-1}}}$ and $\theta_u = \frac{\pi u}{4}$.

Table 1 summarizes classification performances using different approaches. The results of kNN0, kNN1, NNLRS-graph [26] are copied from [26]. It can be seen that our discriminative tensor sparse coding method is comparable to other methods and outperforms the previously proposed dictionary learning method for tensor sparse coding [18].

Table 1: Classification error rates (%) using different approaches with different sampling percentages

Database	kNN0	kNN1	NNLRS [26]	TSCwD [18]	SDL1	SDL2
USPS (10%)	3.13	3.21	2.80	3.03	2.92	2.80
USPS (20%)	2.22	2.10	1.62	1.98	2.02	1.65
USPS (30%)	1.55	1.53	1.13	1.20	1.56	1.02
USPS (40%)	1.20	1.18	0.88	1.01	0.94	0.82
USPS (50%)	0.82	0.86	0.59	2.80	0.58	0.49

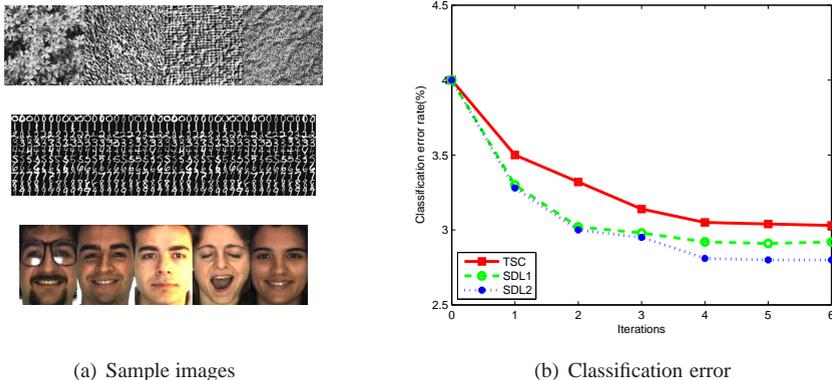


Figure 3: (a) The first row shows sample images from the Brodatz texture dataset; the second row shows sample images from the USPS dataset; the third row shows sample images from the AR dataset. (b) Classification error for different dictionary learning algorithms. For this experiment, we use 20 training images per class.

Table 2: Recognition rates on the AR face database

number of train samp.	TSC [17]	TSCwD [18]	RSR [6]	SDL1	SDL2
15 per person	70.2	78.6	81.4	80.0	82.3
18 per person	73.5	79.9	84.1	82.0	85.2
21 per person	75.8	80.8	85.7	83.2	86.1

In Figure 3(b), we illustrate how classification errors decrease with the number of iterations among dictionary learning methods. As expected, the error rate of SDL2 decreases the most rapidly compared to the parallel-sum method introduced in [18] and SDL1.

4.3 Face Dataset

The AR face database includes over 4,000 color face images of 126 individuals, 26 images for each person in two sessions. The images are cropped to 27×20 and converted into gray scale. The images from 10 subjects are used in our experiment. The images are convolved with Gabor filters using Equation (18) with 8 orientations $\theta_u = \frac{\pi u}{8}$, $u = 0, 1, \dots, 7$, and up to 5 scales $v = 0, 1, 2, 3, 4$. We use the same features as [6] for face recognition. Each face image is described with a 43×43 covariance descriptor using the following features:

$$F(x, y) = [I(x, y), x, y, |G_{0,0}(x, y)|, \dots, |G_{0,7}(x, y)|, |G_{1,0}(x, y)|, \dots, |G_{4,7}(x, y)|] \quad (19)$$

where $I(x, y)$ is the intensity at (x, y) and $G_{u,v}$ is the response of a 2D Gabor wavelet.

We compare our methods with other covariance descriptor based methods including TSC[17], TSC with dictionary update [18], and RSR [6]. The learned dictionary has 7 dictionary atoms per person. For each person, we randomly select 15, 18, 21 images for training and the remainder for testing. Table 2 summarizes the experimental results. SDL2 obtains the best performance in this experiment.

5 Conclusion

We introduced a discriminative dictionary learning approach for tensor sparse coding. The introduction of structural incoherence between dictionary atoms from different sub-dictionaries

encourages disparity among sub-dictionaries, thus enhancing discriminating ability of the sparse representation. We further incorporate label information into the optimization problem so that the learned dictionary is more useful for classification. The SDL1, SDL2 problems can be formulated as MAXDET problems and the dictionary atoms can be updated through gradient descent. Experimental results demonstrate that our approach is robust and effective.

6 Acknowledgement

This research was partially supported by ONR MURI grant N000141010934 and by a grant from Siemens Corporate Research in Princeton, NJ.

References

- [1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, (1):185–184, 1999.
- [2] C. Chen, C. Wei, and Y. Wang. Low-rank matrix recovery with structural incoherence for robust face recognition, 2012. *CVPR*.
- [3] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning, 2007. *ICML*.
- [4] G. Golub, P. Hansen, and D. O’leary. Tikhonov regularization and total least squares. *SIM J. Matrix Anal. Appl.*, pages 21(1):185–184, 1999.
- [5] M. Grant and S. Boyd. Cvx: Matlab software for disciplined convex programming, 2010. <http://cvxr.com/cvx>.
- [6] M. Harandi, C. Sanderson, R. Hartley, and B. Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach, 2006. *ECCV*.
- [7] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, ISBN: 0387952845, 2003.
- [8] T. Hazan, S. Polak, and A. Shashua. Sparse image coding using a 3d non-negative tensor factorization, 2005. *ICCV*.
- [9] Z. Jiang, Z. Lin, and L. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd, 2011. *CVPR*.
- [10] T. Lee. Image representation using 2d gabor wavelets. *PAMI*, pages 959–971, 1996.
- [11] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis, 2008. *CVPR*.
- [12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning, 2009. *NIPS*.
- [13] A. Martinez and R. Benavente. The ar face database. *CVC Technical Report 24*, 1998.

-
- [14] K. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, (2):181–201, 2001.
 - [15] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structural incoherence and shared features, 2010. *CVPR*.
 - [16] T. Randen and J. Husøy. Filtering for texture classification: a comparative study. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 291–310, 1999.
 - [17] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos. Tensor sparse coding for region covariances, 2010. *ECCV*.
 - [18] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos. Positive definite dictionary learning for region covariances, 2011. *ICCV*.
 - [19] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification, 2006. *ECCV*.
 - [20] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds, 2007. *CVPR*.
 - [21] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *PAMI*, 30(10):1713–1727, 2008.
 - [22] J. Yang, K. Yu, and T. Huang. Supervised translation-invariant sparse coding, 2010. *CVPR*.
 - [23] C. Yuan, W. Hu, X. Li, S. Maybank, and G. Luo. Human action recognition under log-euclidean riemannian metric, 2009. *ACCV*.
 - [24] G. Zhang, Z. Jiang, and L. Davis. Online semi-supervised discriminative dictionary learning for sparse representation, 2012. *ACCV*.
 - [25] Y. Zhang, Z. Jiang, and L. Davis. Learning structured low-rank representations for image classification, 2013. *CVPR*.
 - [26] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, and N. Yu. Non-negative low rank and sparse graph for semi-supervised learning, 2012. *CVPR*.