ANALYSIS DICTIONARY LEARNING FOR SCENE CLASSIFICATION

Wen Tang [*], *Ives Rey Otero* [*], *Hamid Krim* [*], *Liyi Dai* [†]

[*]Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC 27695, USA
{wtang6, ireyote, ahk}@ncsu.edu
[†]Army Research Office, RTP, NC 27703, USA
liyi.dai@us.army.mil

ABSTRACT

This paper proposes a new framework for scene classification based on an analysis dictionary learning approach. Despite their tremendous success in various image processing tasks, synthesis-based and analysis-based sparse models fall short in classification tasks. It was hypothesized that this is partly due to the linear dependence of the dictionary atoms. In this work, we aim at improving classification performances by compensating for such dependence. The proposed methodology consists in grouping the atoms of the dictionary using clustering methods. This allows to sparsely model images from various scene classes and use such a model for classification. Experimental evidence shows the benefit of such an approach. Finally, we propose a supervised way to train the baseline representation for each class-specific dictionary, and achieve multiple classification by finding the minimum distance between the learned baseline representation and the data's sub-dictionary representation. Experiments seem to indicate that such approach achieves scene-classification performances that are comparable to the state of the art.

Index Terms— Analysis Dictionary Learning, Sparse signal model, scene classification, optimization

1. INTRODUCTION

Understanding the content of an image remains one of the most challenging problems in vision. Over the years, high-level challenges like scene-classification have encouraged the development and use of countless techniques for image processing, computer vision and machine learning [1, 2, 3]. Another major advance of the last decade was the use of sparse and redundant signal representations [4]. The Synthesis Dictionary Learning (SDL) framework and the Analysis Dictionary Learning (ADL) framework are two examples of such sparse modelization tools. But despite their tremendous success for low-level image processing tasks [4, 5], the SDL and ADL framework have only demonstrated limited performance in classification tasks, especially in high-level ones.

The synthesis-based sparse model assumes that the signal of interest x can be approximated by a *sparse* linear combination $x \approx D\alpha$, meaning that it is approximated by a linear combination of *a few* atoms from a given dictionary D. SDL methods [6, 7] aim at simultaneously learning from example signals, a dictionary D and a sparse representation α which approximate them well while promoting sparsity. This is generally achieved by solving a minimization problem with a target function such as

$$\underset{D,\alpha}{\operatorname{argmin}} \|x - D\alpha\|_2^2 + \lambda\|\alpha\|_1, \tag{1}$$

where the second term favors sparse coefficients α.

An alternate viewpoint is the *analysis*-based sparse model in which the signal x is multiplied by an analysis dictionary Ω and the resulting vector Ωx is expected to be sparse. ADL methods [8, 9, 10] aim at learning the analysis dictionary by solving minimization problems such as:

$$\underset{\Omega,U}{\operatorname{argmin}} \|\Omega X - U\|_2^2 + \lambda\|U\|_0. \tag{2}$$

It has been observed that the linear dependence of the dictionary atoms (respectively the columns of D and the rows of Ω) result in poor performances in typical classification tasks of SDL and ADL-based methods [11]. This also reduces the ability of classification. Therefore, the present research aims at addressing such limitation for the analysis-based model. Since dictionary can characterize the features of the signal, and there are several general content features for each scene category image, such as tree and house for suburban, peak for mountain, we provide a practical way to reduce the linear dependence between the atoms of the analysis dictionary by using K-means to group atoms based on various visual feature transforms. The signal is then represented by using the normalized histogram of the clusters. By doing so we not only reduce the coherence in atoms, but also keep the general content features. Our methodology is then applied to the challenging task of scene-classification.

The remainder of this article is organized as follows: Section 2 discusses other attempts to improve distinctiveness of sparse model based classification algorithms by reducing the linear dependence of dictionary atoms. These attempts aim at reducing the linear dependence of dictionary atoms. They mainly aiming focused on synthesis-based sparse model. Section 3 describes the analysis-based sparse model proposed by Bian *et al.* [9] as well as its present use within the task of scene-classification. In Section 4, an experimental section shows classification results on a scene-classification dataset. This experimental section demonstrates that grouping the atoms improves the performance of analysis-based classification. The performances are comparable to the state-of-the-art in scene-classification. Section 5 provides a final discussion.

2. RELATED WORK

Sparse models are notoriously unadapted to classification tasks [5, 12]. Linear dependence betweens atoms is thought to be the reason for such a low performance. For synthesis-based sparse modeling, various modifications of the minimization problem have been proposed either to penalize coherent atoms in the dictionary D or to make the sparse representation α more discriminative.

978-1-4673-7802-4/16/$31.00 ©2016 IEEE

Ramirez *et al.* [11] add a constraint in the minimization problem that penalizes coherent dictionary atoms. Such simple modification allows them to achieve good performance in classic classification tasks on various datasets (speech, digit, texture). Mairal *et al.* jointly learn the dictionary and a classifier. This was again achieved by simply embedding the classification logistical regression function in the minimization problem during the learning process. This led to improved classification performance. Similarly, Zhang *et al.* [13] proposed to learn a linear classifier jointly with the dictionary. Such an approach led to more discriminative sparse coefficients that were used for face recognition. Kong and Wang [14] simultaneously learned a set of dictionaries specific to each class and by adding a constraint on the scalar product of pairs of atoms, were able to identify common dictionary atoms shared by various classes, and separate them for improved classification performance, which also worked well for scene classification and object recognition.

Inspired by these methods, our approach is an attempt in the ADL framework to avoid atoms linear dependence that are detrimental to scene classification.

3. OUR APPROACH

We first provide a brief overview of our image scene classification method. Training the classifier consists in performing the following steps for *each* scene class:

Step 1: The algorithm proposed by Bian *et al.* [9] is used to train one analysis dictionary for each class.
Step 2: The atoms of each class-specific dictionary are clustered using the K-means clustering method with various visual feature transforms. We calculate the group indicator of each atom (i.e., the index of the cluster the atom is in). (see Fig. 2).
Step 3: The group indicator of each atom is used to calculate the normalized histograms of groups to represent an image, which is also a mid-level feature for a scene image (see Fig. 2). Here we call it sub-dictionary representation (See Fig. 3).
Step 4: Learn the mid-level (sub-dictionary) representation from the labeled image and set this as a baseline for this class.

Once those training steps are performed for each class, classifying an unlabeled image consists in computing its mid-level representation in each class and comparing them to the class baseline. The output of the classifier is the class for which the distance to the baseline is the smallest one. In what follows, we detail the different steps of the approach.

3.1. Class-specific Analysis Dictionary

Let $X_c = [x_1, \dots, x_n]$ denote scene-specific data matrix, *i.e.,* the matrix formed of patches extracted from images in the scene class c, each x_i is a column vector, $x_i \in \mathbb{R}^d$. A class-specific analysis dictionary $\Omega_c = [\omega_1^T, \dots, \omega_n^T]^T$, with $m = d$ and where ω_i denotes the i-th atom is computed for each class along with the corresponding sparse coefficients $U_c = [u_1, \dots, u_n] \in \mathbb{R}^{m \times n}$. This consists in minimizing the following target function:

$$\{U_c, \Omega_c\} = \underset{U_c, \Omega_c}{\operatorname{argmin}} \quad \|\Omega_c X_c - U_c\|_2^2 + \lambda \|U_c\|_1. \tag{3}$$

Such minimization is performed in practice using the the analysis dictionary learning algorithm proposed by Bian *et al.* [9] that we describe in what follows.

Analysis dictionary learning algorithm

In the following description, let $X \in \mathbb{R}^{d \times l}$ denote the data matrix, Ω denote the analysis dictionary and U denote the sparse coefficients of X relative to the analysis dictionary Ω, *i.e.,* $\Omega X = U$.

The algorithm proposed in [9] departs from methods based on the co-sparse analysis model [15, 16, 8] and relates instead with the sparse null space problem [17]. It consists of the three following steps: **Step 1:** Build a matrix A such that $XA^T = 0$. **Step 2:** Build the matrix $U \in \mathbb{R}^{m \times n}$ of sparse representations so that its rows are sparse and form a basis of null(A). **Step 3:** Estimate the dictionary $\Omega \in \mathbb{R}^{m \times d}$ from X and the sparse representation U.

In practice, the matrix A is constructed so that its rows are the right-singular vectors of X relative to null singular values. This results in a $(n - r) \times n$ full-rank matrix A, where r denotes the rank of the data matrix X.

Then the rows of U are computed sequentially. Once the $(i-1)$ first rows $(u_1, u_2, \dots, u_{i-1})$ have been computed, computing u_i consists in picking the sparsest vector among the solutions of the following n convex optimization problems: For each index $j = 1 \dots n$

$$\begin{aligned} \text{minimize} \quad & \|u\|_1 \\ \text{subject to} \quad & \left(P_{\text{span}\{u_1, \dots, u_{i-1}\}^\perp}(u)\right)_j > 0 \\ & Au = 0, \end{aligned} \tag{4}$$

where $(.)_j$ denotes the j-th component of a vector. In each optimization problem, the ℓ_1 norm promotes sparsity while the two constrains assure that the rows of U are linearly independent and in the null space of A. As a direct consequence, each of the n convex optimization problems is feasible as long as the number of atoms m is not larger than the dimension d [1] .

Finally, the dictionary matrix $\Omega \in \mathbb{R}^{m \times d}$ is estimated directly from U and X, namely, as the product of U with the Moore-Penrose pseudo-inverse of X [2] . The interested reader is referred to [9] for more details.

3.2. Sub-dictionary Space

Visually, the atoms linear dependence manifests itself with similar shapes and layouts. This implies that applying feature transforms to similar atoms will lead to almost identical representations. Thus, we map each atom into a feature space, where similar atoms will form a cluster. We use a feature transform F that associates to each atom $\omega_i \in \mathbb{R}^d$ a feature vector $F(\omega_i) \in \mathbb{R}^p$.

The resulting feature vectors form clusters that are then easily separated into k groups $G = [g_1|g_2|\dots|g_K]$ using the K-means clustering algorithm. Formally

$$\omega_i \in g_k, \quad \text{with } k = \underset{k'}{\operatorname{argmin}}\, d(F(\omega_i), C(g_{k'})), \tag{5}$$

with $i = 1, \dots, m, \quad k = 1, \dots, K$ and where $C(g_k)$ denotes the center of the cluster g_k (*i.e.*, the average position of all features within that cluster) and $d(\cdot, \cdot)$ denotes the Euclidean distance. The results of the clustering is a set of K group indicator vectors $b^k \in \{0, 1\}^m$ for $k = 1 \dots K$.

$$b_i^k = \mathbf{1}_{\omega_i \in g_k} \tag{6}$$

[1] Elad *et al.* [15] point out that, in the determined case ($m = n$ and X invertible) the ADL problem has an equivalent SDL formulation.

[2] Note that in general, the product of the resulting Ω with the data matrix X is an approximation of the sparse matrix U.

with $i = 1, \ldots, m$ and where $\mathbf{1}_{\omega_i \in g_k}$ denotes an indicator function that equals 1 if the feature vector of atom ω_i falls inside the cluster g_k and 0 otherwise.

3.3. Sub-dictionary Representation

Using the trained class-specific dictionaries and atoms grouping during the previous steps, we are able to associate to any new data signal with one sub-dictionary representation (namely a vector in $\mathbb{R}^K$) specific to each class.

Given a set of n data vector in $\mathbb{R}^m$, we compute for each scene class the sparse coefficients $U_c = [u_1, \ldots, u_n] \in \mathbb{R}^{m \times n}$. In the ADL framework, this is achieved directly by multiplying the class-specific dictionary Ω_c by the data matrix. For the sparse representation u_i (*i.e.*, corresponding to the i-th data vector), the j-th coefficient $(u_i)_j = \omega_j x_i$ is the response of atom ω_j to data x_i. And after clustering, the binary vector b^k, group indicator , stands for whether ω_i is in group k. Thus, for each patch x_j, we can get the group frequency vector v_j for its sparse coefficient u_j as follows:

$$v_j = [\|b^1 \circ u_j\|_0, \|b^2 \circ u_j\|_0, \ldots, \|b^K \circ u_j\|_0]^T \tag{7}$$

where $j = 1, \ldots, n$, and the $\circ$ is the Schur product (element wise product) and where $\|.\|_0$ returns the number of non-zero elements of a vector.

For a specific class, the sub-dictionary representation vector of the data x is therefore

$$W = \frac{1}{Z} \sum_{j=1}^{n} v_j; \quad \|W\|_1 = 1, \tag{8}$$

where Z is a normalization term to get a normalized histogram of the groups.

3.4. Multiple Supervised Classification

We have proposed a method for representing an image into a sub-dictionary representation. Our strategy to deal with multiclass classification is to train the class-specific dictionary for each class by labeled data first. So, we can get a set of dictionaries $\{\Omega_1, \ldots, \Omega_C\}$, C is the number of classes. For each Ω_c, we can form its sub-dictionary space and obtain its sub-dictionary representation for data. Hence, for each data X, its new sub-dictionary representations are $\{W_1(X), \ldots, W_C(X)\}$. Then, the second step is to train the class-specific sub-dictionary representation for each dictionary by the training data. Here we name baseline the trained class-specific sub-dictionary representation and denote it by $\{W_1^b, \ldots, W_C^b\}$. Finally, we compare the distance (as defined in [18, 9]) of each class-specific baseline to the sub-dictionary representation of the signal Y to be classified. The data Y is then assigned to the class with the minimal distance. Formally

$$\text{class}(Y) = \underset{c' \leq C}{\text{argmin}}\, d(W_{c'}^b, W_{c'}(Y)),$$

where $d(\cdot, \cdot)$ denotes the Manhattan distance in $\mathbb{R}^p$.

4. EXPERIMENTS

Dataset Our method is evaluated on the 15 scene dataset provided by Lazebnik [2]. This dataset contains a total of 4485 images divided into 15 scene categories (see Fig. 1). For each category, there are between 200 and 400 images of size 300×250 pixels. Here, for each category, we randomly sampled 200 images as the dataset. 80 images are used to train a class-specific analysis dictionary. Another 40 images are used to learn the baseline of the sub-dictionary representation for each category. The rest of the images are used as test images.

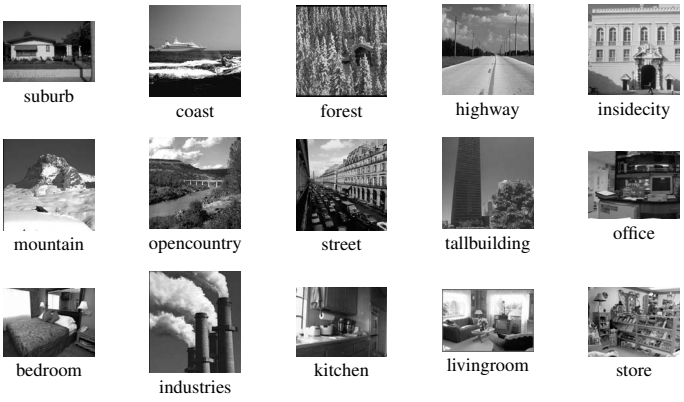

Fig. 1. 15 categories in the dataset.

Experimental Setup: For each image, we extract patches of size 100×100. Patches are extracted every 70 pixels, resulting in overlapping patches. An analysis dictionary of 60 atoms is learned for each category. Each of the 60 atoms (vectors in $\mathbb{R}^{10000}$ is reshaped into a 100×100 images so as to be able to extract local features). Large patches like these allow to capture middle level image features that are useful for classification. Note that it departs from the more conventional approach of using over-complete dictionaries ($m > d$) and results from the theoretical and computational limitations of the algorithm algorithm proposed by Bian *et al* [9]. Furthermore, the present task (scene classification) doesn't necessarily require over-complete dictionary, because clustering is ultimately to perform a low dimensional representation. For each reshaped atom, we densely compute `SIFT` (Scale-invariant Feature Transform) [1] features. This consists in computing a `SIFT` feature vector from a 16×16 window every 8 pixels. For each reshaped atom, we also compute a set of `LBP` (Local Binary Pattern) [19] features with 8 neighbors of radius 1. For the `HoG` (Histogram of Oriented Gradients) [20] features, we set the cell size as 8×8 and the block size as 2×2. An implementation of `GIST` (Spectral Envelope) [21] descriptors is also considered. Finally, we consider the concatenation of the four features. Such descriptor is denoted `All`.

Fig. 2 shows the example of our sub-dictionary space, after using K-means clustering on the features. The atoms in different color box are different groups. Note the coherence between the atoms. For each group, it contains the semantic meanings to understand the content of scene image. For example, the red group represents the tree leaves, while the green one represents the branch of the tree. The blue group stands for the house and its roof. The yellow group is the fence part of a house. And the purple one is the house windows.

Then, we order these "object" groups. Here, for instance, red group is group 1, blue group is group 2, yellow group is group 3, green group is group 4 and purple group is group 5. According to the group indicators, we could construct our sub-dictionary representation, *i.e.* mid-level representation, which is shown in Fig. 3. For a suburb image, we can understand it as high probability in tree-leaves, tree-branches and house pulsing a few house-windows and house-fences by our sub-dictionary representation in Fig. 3. The

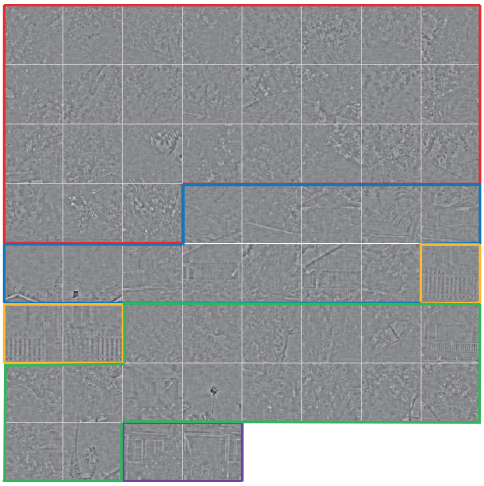

Fig. 2. Example of class-specific dictionary analysis dictionary. The atoms are clustered into sub-dictionaries by using K-means algorithm based on the visual features. Each cluster (or group) is here assigned a different color.

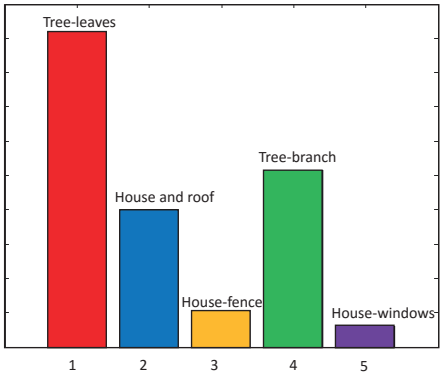

Fig. 3. Example of Sub-dictionary Representation.

roof, fence and windows of a house never appear in the nature and indoor scene images. Thus, these sub-dictionary representation could keep the features to characterize the content of an image.

Method	Accuracy (%)
KSPM [2]	81.10
ScSPM [22]	80.28
Object Bank [23]	80.90
DL-COPAR [14]	**85.37**
Bian *et al.* [9]	37.78
Our approach (SIFT)	71.67 ± 8.39
Our approach (LBP)	70.00 ± 3.85
Our approach (HoG)	68.33 ± 12.62
Our approach (GIST)	66.67 ± 14.78
Our approach (All)	80.00 ± 10.88

Table 1. Classification results on 15 Scene dataset.

We experimentally compared our approach with the state-of-the-art. These methods mainly focused on finding an appropriate codebook to represent the image. The comparison results are shown in Table 1. Lazebnik *et.al.* proposed KSPM [2] that use the K-means clustering algorithm is used to learn a codebook and introduced in spatial pyramid matching to achieve an accuracy of 81.10% on the 15 scene classification dataset. SsSPM [22] uses sparse coding, max pooling and linear SVM classifier based on the SPM for image classification to achieve a 80.28% classification accuracy. Object bank [23] proposed to obtain a scale-invariant response map by training generic object detectors, and then collected them together as a codebook reaching an accuracy of 80.9%. DL-COPAR simultaneously learned a set of synthesis dictionaries specific to each class with common dictionary atoms shared by various classes. By separating the common atoms to others, the method improves classification performance and is the current state of the art for scene classification with 85.9% accuracy.

We tested our approach with a variety of image features. We tested separately SIFT, LBP, HoG and GIST with our approach. The accuracy achieved using these features are respectively 71.67 ± 8.39, 70.00 ± 3.85, 68.33 ± 12.62, 66.67 ± 14.78 (see Table 1). Note that using LBP and HoG results in an increase classification accuracy and a lower variance than when using SIFT and GIST. Concatenating the four features (feature All) results in an accuracy of 80% with a variance of $^{+}/_{-}10\%$. Such high variance is the result of K-means clustering algorithm and its random initialization which limits the robustness and stability of the proposed approach.

Since the algorithm proposed by Bian *et al.* achieved good performance for texture classification, it was incorporated in this comparison. As expected, it achieves poor performance for scene classification with an accuracy of only 37.78%.

Category	k	Category	k
CALsuburb	12	MITcoast	3
MITforest	4	MIThighway	9
MITinsidecity	7	MITmountain	13
MITopencountry	2	MITstreet	29
MITtallbuilding	8	PARoffice	4
bedroom	10	industries	21
kitchen	8	livingroom	3
store	17	Average	10

Table 2. The maximum dimension of discriminative sub-dictionary representation.

In our approach, the dimension of the final representation of an image is very low. From the Table 2, we can find the average dimension for each category to be around 10. Thus, our approach not only gives a meaningful representation, but also reduces the dimension for high dimensional data.

5. CONCLUSION

In this paper, we use the K-means clustering and feature transforms to find the subspace of atoms. We subsequently use the these sub-dictionary groups to represent the original data. Finally, we achieve the multiple classification by finding the minimum distance between the learned baseline representation and the data's sub-dictionary representation. Our experiments on the scene classification demonstrates that the method proposed by Bian *et al.* achieves poor performances for the task of scene classification. The experiments also demonstrates that grouping the atoms improves performances. As future work, we will work at grouping the linearly dependent atoms within an ADL optimization formulation directly to improve discriminativity.

6. ACKNOWLEDGMENTS

Thanks are due to the NC Biotech Center (Grant 2014-CFG-8002) and Scynexis of NC, and to the U.S. Army Research Office, Grant W911NF-04-D-0003-0022, as well as the Department of Energy Grant DE-NA0002576, and the NCSU College of Engineering for partial funding of this research.

7. REFERENCES

[1] Li Fei-Fei and Pietro Perona, "A bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 2, pp. 524–531.

[2] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, 2006, vol. 2, pp. 2169–2178.

[3] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[4] Michael Elad and Michal Aharon, "Image denoising via learned dictionaries and sparse representation," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, 2006, vol. 1, pp. 895–900.

[5] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R Bach, "Supervised dictionary learning," in *Advances in neural information processing systems*, 2009, pp. 1033–1040.

[6] Bruno A Olshausen et al., "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[7] Alfred M Bruckstein, David L Donoho, and Michael Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.

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

[9] Xiao Bian, Hamid Krim, Alex Bronstein, and Liyi Dai, "Sparse null space basis pursuit and analysis dictionary learning for high-dimensional data analysis.," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. ICASSP, 2015.

[10] Gabriel Peyré and Jalal M Fadili, "Learning analysis sparsity priors," in *Sampta'11*, 2011.

[11] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3501–3508.

[12] Julien Mairal, Francis Bach, and Jean Ponce, "Task-driven dictionary learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 4, pp. 791–804, 2012.

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

[14] Shu Kong and Donghui Wang, "A dictionary learning approach for classification: separating the particularity and the commonality," in *Computer Vision–ECCV 2012*, pp. 186–199. Springer, 2012.

[15] Michael Elad, Peyman Milanfar, and Ron Rubinstein, "Analysis versus synthesis in signal priors," *Inverse problems*, vol. 23, no. 3, pp. 947, 2007.

[16] Sangnam Nam, Mike E Davies, Michael Elad, and Rémi Gribonval, "The cosparse analysis model and algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, pp. 30–56, 2013.

[17] Thomas F Coleman and Alex Pothen, "The null space problem i. complexity," *SIAM Journal on Algebraic Discrete Methods*, vol. 7, no. 4, pp. 527–537, 1986.

[18] David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer, *Markov chains and mixing times*, American Mathematical Soc., 2009.

[19] David Harwood, Timo Ojala, Matti Pietikäinen, Shalom Kelman, and Larry Davis, "Texture classification by center-symmetric auto-correlation, using kullback discrimination of distributions," *Pattern Recognition Letters*, vol. 16, no. 1, pp. 1–10, 1995.

[20] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 886–893.

[21] Aude Oliva and Antonio Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[22] Jianchao Yang, Kai Yu, Yihong Gong, and Tingwen Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1794–1801.

[23] Li jia Li, Hao Su, Li Fei-fei, and Eric P. Xing, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Advances in Neural Information Processing Systems 23*, J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, Eds., pp. 1378–1386. Curran Associates, Inc., 2010.