Computer Vision and Image Understanding 136 (2015) 59–68

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Discriminative structured dictionary learning with hierarchical group sparsity ☆

Yong Xu *, Yuping Sun, Yuhui Quan, Bo Zheng

School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 24 April 2014
Accepted 20 January 2015

Keywords:
Discriminative dictionary learning
Structured sparse coding
Group sparsity
Image classification

ABSTRACT

Learning adaptive dictionaries for sparse coding has been the focus of latest research as it provides a promising way to maximize the efficiency of sparse representation. In particular, learning discriminative dictionaries rather than reconstructive ones has demonstrated significantly improved performance in pattern recognition. In this paper, a powerful method is proposed for discriminative dictionary learning. During the dictionary learning process, we enhance the discriminability of sparse codes by promoting hierarchical group sparsity and reducing linear prediction error on sparse codes. With the employment of joint within-class collaborative hierarchical sparsity, our method is able to learn adaptive dictionaries from labeled data for classification, which encourage coefficients to be sparse at both group level and singleton level and thus enforce the separability of sparse codes. Benefiting from joint dictionary and classifier learning, the discriminability of sparse codes is further strengthened. An efficient alternating iterative scheme is presented to solve the proposed model. We applied our method to face recognition, object recognition and scene classification. Experimental results have demonstrated the excellent performance of our method in comparison with existing discriminative dictionary learning approaches.

1. Introduction

In recent years, sparse representation has drawn much attention from the computer vision community and led to state-of-the-art results in many computer vision tasks, e.g. image classification [1–4] and image restoration [5–8]. The success of sparse representation based classification is attributed to the fact that high-dimensional image data from the same class lie on a low-dimensional manifold and thus can be coded using a few representative elements (the so-called *atoms*). The collection of such elements is often referred to as a *dictionary*.

The choice of dictionary is one of the fundamental considerations in employing sparse representation based models. While predefined dictionaries such as off-the-shelf bases like wavelets [5,7] have been successfully applied to sparse modeling in many signal processing applications, many reconstructive dictionary learning methods [9–12] have shown that noticeable performance improvement can be obtained by learning adaptive dictionaries from data themselves. The learned reconstructive dictionaries are adapted to the underlying structures of data and hence are able to improve the efficiency of sparse coding.

However, the reconstructive dictionaries often suffer from the insufficiency of discrimination in complex recognition tasks. In fact, a dictionary becomes useful for sparsity-based recognition when it not only enjoys excellent sparse-representational power but also has the ability to induce discriminative representation for samples from different categories. As a result, many discriminative dictionary learning methods [13–25] have been proposed to learn both reconstructive and discriminative dictionaries for sparse coding.

In this paper, a powerful discriminative dictionary learning method is proposed for sparse coding based classification. In the proposed method, we simultaneously learn a structured dictionary with hierarchical group sparsity and train a linear classifier for classification. By promoting joint within-class collaborative hierarchical sparsity in sparse codes, our method is able to learn dictionaries adapted to the underlying structures of data. The learned dictionaries encourage samples from different categories to exhibit distinct hierarchical group sparsity patterns, making

☆ Yong Xu would like to thank the supports by National Nature Science Foundations of China (61273255 and 61070091), Engineering & Technology Research Center of Guangdong Province for Big Data Analysis and Processing ([2013]1589-1-11), The Project of High Level Talents in Higher Institution of Guangdong Province (2013-2050205-47) and GuangDong Technological Innovation Project (2013KJCX0010). Yuping Sun would like to thank the support by China Scholarship Council Program. Yuhui Quan would like to thank the partial support by Singapore MOE Research Grant (R-146-000-178-112).

* Corresponding author.
E-mail addresses: yxu@scut.edu.cn (Y. Xu), sun.yp@mail.scut.edu.cn (Y. Sun), yuhui.quan@mail.scut.edu.cn (Y. Quan), bozheng@mail.scut.edu.cn (B. Zheng).

http://dx.doi.org/10.1016/j.cviu.2015.01.006
1077-3142/

the sparse codes more separable between classes. Meanwhile, benefiting from joint dictionary learning and classifier training, the learned dictionaries are both reconstructive and discriminative. As a result, the discriminability of sparse codes and the discrimination of active groups of dictionary atoms are further strengthened. Experimental results on face recognition, object recognition and scene classification have demonstrated the excellent performance of our method in comparison with many state-of-the-art discriminative dictionary learning methods.

1.1. Related work

1.1.1. Discriminative dictionary learning

The techniques used in discriminative dictionary learning methods can be roughly categorized into two types, which are detailed as follows:

Learning class-associated subdictionaries. In order to obtain discriminative dictionaries, many approaches (e.g. [15,19,20]) train class-associated subdictionaries using labeled data. The differences of these approaches lie in two aspects: the way of associating subdictionaries with class information and the regularization imposed on subdictionaries. For instance, Yang et al. [23] proposed to associate each dictionary atom with a class label and encourage each input signal to be well represented only by the dictionary atoms that share the same class label with the signal. In [20], the class-specific subdictionaries dictionaries are encouraged to be independent of each other.

Integrating supervised learning. To enforce discriminability in sparse codes, many approaches incorporate classifier training into the dictionary learning, i.e., the dictionary and classifier are jointly learned. The classification loss functions vary in these methods, e.g., the softmax discriminative cost [15,16], Fisher discrimination criterion [13,23,24,26], linear predictive classification error [17,22,25], hinge loss [21,27], and logistic loss [16,28].

1.1.2. Structured sparse coding

While the standard sparsity induced by ℓ_0 norm or its convex relaxation ℓ_1 norm has been widely-used for sparse coding, recent literature [29–43] seeks for some higher-level sparsity (often referred to as structured sparsity) to encode higher-order information about the patterns of non-zero coefficients in sparse codes. The concept of structured sparsity is first introduced in image restoration and the benefits of structured sparse coding have been confirmed both theoretically and via numerous applications [33]. There are many types of structured sparsity, e.g., sparsity defined on disjunct groups [38,39] or overlapping groups [31], tree sparsity [34,35,43], and graph sparsity [30].

The structured sparsity is an effective tool to reveal the underlying structures of data. Thus, a few methods [40–43] employ structured sparsity in dictionary learning for classification. In [43], Szlam et al. proposed a fast approximate sparse coding algorithm that uses a tree structure for inference and applied it to build an accurate real time object recognition system. For face recognition, Jenatton et al. [41] proposed the so-called structured sparse PCA, in which the sparsity patterns of all dictionary elements are structured and constrained in a pre-specified set of shapes. Although impressive results have been achieved by these methods, there is still much room for the improvement of discriminability in sparse codes.

1.2. Motivation and contributions

We illustrate our motivation by two examples. See Fig. 1 for an illustration of the benefits of using hierarchical group sparsity in document representation. Assume that we have multiple documents that are related to different topics, e.g. political news and fashion magazines. The key words of the documents serve naturally as a good dictionary, and thus each document can be represented by histogram of key words. Then we assign the dictionary atoms (i.e. key words) into several groups, e.g. city-state nouns, fashion-related words, etc. The histogram of key words of a document should be sparse because each only contains a tiny part of all the key words. This can be considered as the *singleton-level sparsity*. Besides, each document can be efficiently represented by the words from a few specific groups, which can be interpreted as *group-level sparsity*. Furthermore, documents from the same category would share the same sparsity pattern at the group-level but not necessarily at the singleton-level. This naturally yields to the *joint within-class collaborative hierarchical sparse representation*. Such a well-designed group sparsity is very effective and efficient for classifying documents.

Now turn to the case of image classification, in which we do not have a good dictionary containing well-clustered visual key words. Thus, it is necessary to learn a structured dictionary to obtain joint within-class collaborative hierarchical sparse representation for image classification. See Fig. 2 for the example. Suppose we have obtained some low-level features (e.g. bag-of-words based on SIFT descriptors [44]) of multiple object images from different cate-

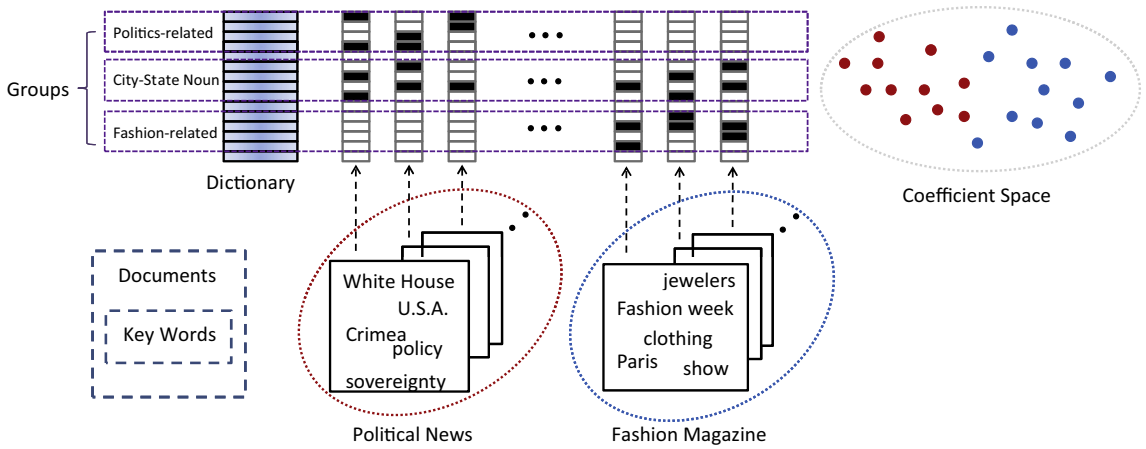


Fig. 1. Illustration of the benefits of using hierarchical group sparsity in document representation. Based on the key words of documents, a group-structured dictionary can be constructed to represent each document as a histogram of key words. Documents from the same category would share the same sparsity pattern at the group-level but not necessarily at the singleton-level, while different categories correspond to distinct hierarchical group sparsity patterns. Such a well-designed group sparsity is very effective and efficient for classification.

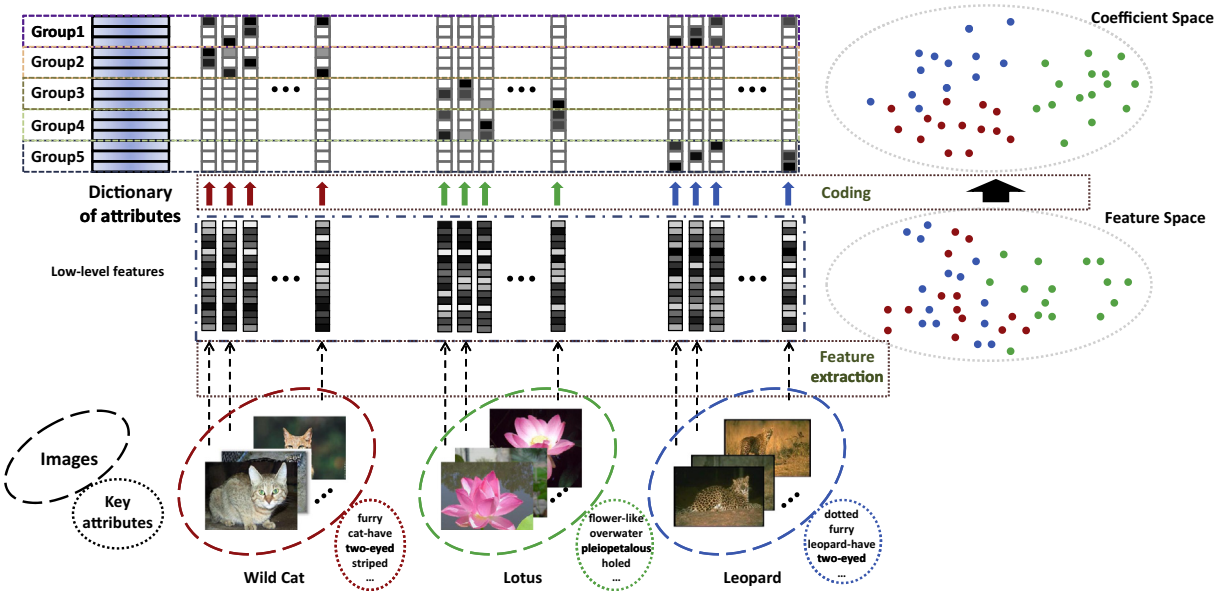


Fig. 2. Illustration of the benefits of using hierarchical group sparsity in image representation with a structured visual dictionary. Low-level features of object images from different classes often fail to be linearly separable in feature space. In order to construct high-level representations that are sufficiently discriminative for classification task, one need to learn a well-designed group-structured dictionary consisting of good visual attributes. Given such a dictionary, collaborative hierarchical group sparsity patterns can be obtained for each object category to improve image classification performance, just like the case in document representation.

gories. Such features on some objects (e.g. cats and leopards) are quite similar such that they are rarely linearly separable in feature space. Then we need to pursue an ideal group-structured dictionary consisting of good atoms (e.g. visual attributes), each of which could capture certain common properties across different categories, either semantic (e.g. furry) or discriminative (e.g. cats have it but leopards do not). Once we get such a dictionary, collaborative hierarchical group sparsity patterns can be obtained for each object category, just like the case in document representation. Such joint within-class collaborative hierarchical sparse representation is more separable for classification than the original feature. In Fig. 2, active groups of two distinct categories could be very different, e.g. group 1 and 2 for wild cat and group 3 and 4 for lotus. Two closely-related categories (e.g. cats and leopards) may have some overlap group (e.g. group 1) but can still be distinguished.

In real-world applications, often the learned group-structured dictionary is insufficiently discriminative for classification. In particular, each active group is not guaranteed to be shared by few classes. In the case where active groups are shared by a lot of classes, the discriminability of the structured sparse codes would dramatically decrease. Thus, more discriminative information like classification feedback should be incorporated into structured dictionary learning, which can improve the interclass discrimination of active atom groups and sparse codes.

The arguments above inspire us to propose a discriminative structured dictionary learning approach for sparse coding, which integrates hierarchical group sparsity promotion and classifier training into the dictionary learning process. Compared with the traditional purely reconstructive dictionary learning methods, the learned dictionaries are both reconstructive and discriminative. In comparison with the purely supervised dictionary learning methods like D-KSVD [22], the dictionaries learned by our method can reveal the underlying structures of data and encourage distinct hierarchical group sparsity patterns on sparse codes of samples from different categories. Compared to the purely structured dictionary learning methods, our method is able to support the discrimination of active groups shared by different classes and produce discriminative sparse codes for classification. Experimental results on face recognition, object recognition and scene classification have demonstrated the excellent performance of our method in comparison with many state-of-the-art discriminative dictionary learning methods.

The rest of this paper is organized as follows. Section 2 is devoted to the preliminaries and background knowledges on sparse representation and discriminative dictionary learning. Section 3 describes our method. Section 4 is for experimental evaluation and results. Section 5 concludes the paper and discusses future work.

2. Preliminaries

We first give an introduction to the definitions and notations used in this paper. Bold upper letters are used for matrices, bold lower letters for column vectors, regular lower letters for scalars, and calligraphic English alphabets for sets. For example, $\boldsymbol{y}_j$ denotes the j-th column of the matrix $\boldsymbol{Y}$, y_i denotes the i-th element of the vector $\boldsymbol{y}$, and $Y_{i,j}$ denotes the entry of $\boldsymbol{Y}$ at the i-th row and the j-th column. Besides, let $\boldsymbol{y}^{(t_i)}$ denote the t_i-th element of a sequence $\{\boldsymbol{y}^{(t)}\}_{t\in\mathbb{N}}$. Given a matrix $\boldsymbol{Y}\in\mathbb{R}^{n\times p}$, a set of row indices $\mathcal{G}\subseteq\{1,\ldots,n\}$ and a set of column indices $\mathcal{P}\subseteq\{1,\ldots,p\}$, let $\boldsymbol{Y}_{|\mathcal{G}|}$ denote a sub-matrix of $\boldsymbol{Y}$ formed by taking rows from $\boldsymbol{Y}$ with indices $\mathcal{G}$, and let $\boldsymbol{Y}_{(\mathcal{P})}$ denote the sub-matrix of $\boldsymbol{Y}$ formed by collecting columns from $\boldsymbol{Y}$ with indices $\mathcal{P}$. For a matrix $\boldsymbol{X}$, its Frobenious norm is defined as $\|\boldsymbol{X}\|_F=\sqrt{\sum_{i,j}|X_{i,j}|^2}$, its ℓ_0 norm is defined as the number of nonzero entries in $\boldsymbol{X}$. Given a vector $\boldsymbol{x}$, its ℓ_1 norm is defined as $\|\boldsymbol{x}\|_1=\sum_j|x_j|$. Besides, $\boldsymbol{I}_M$ denotes the $M\times M$ identity matrix.

2.1. Structured sparse coding

Let $\boldsymbol{Y}=[\boldsymbol{y}_1,\ldots,\boldsymbol{y}_p]\in\mathbb{R}^{n\times p}$ be a set of input signals and $\boldsymbol{D}=[\boldsymbol{d}_1,\ldots,\boldsymbol{d}_m]\in\mathbb{R}^{n\times m}$ denote a given dictionary. Sparse coding is to find the sparse code $\boldsymbol{c}_i$ of each signal $\boldsymbol{y}_i$ with the dictionary $\boldsymbol{D}$, which can be accomplished by solving the following problem:

$$\min_{\boldsymbol{C}\in\mathbb{R}^m}\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{C}\|_2^2+\lambda\psi(\boldsymbol{C}), \tag{1}$$

where $\boldsymbol{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_p]$ is the sparse codes to pursue, ψ is a sparsity-inducing functional, and λ is a parameter that balances the trade-off between sparsity and fidelity of the solution. A common choice of ψ is ℓ_0 norm or its convex relaxation ℓ_1 norm. Both of ℓ_0 norm and ℓ_1 norm primarily encourage sparse solutions, regardless of the potential structural relationships (e.g. spatial, temporal or hierarchical) existing among variables.

Inspired by the fact that often nonzero coefficients in sparse codes are not randomly distributed but have certain patterns, recent approaches [29–43] seek for new sparsity-inducing functionals which are capable of encoding higher-order information about the patterns of non-zero coefficients. The resulting sparsity on sparse codes is called *structured sparsity*. One simple but efficient method for structured sparse coding is the so-called *collaborative hierarchical group Lasso (CHiLasso)* method [36,39], which induces both standard sparsity and *group sparsity* by solving the following problem:

$$\min_{\boldsymbol{C}\in\mathbb{R}^{m\times p}} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 + \lambda_1 \sum_i \|\boldsymbol{c}_i\|_1 + \lambda_2 \psi_{\mathcal{S}}(\boldsymbol{C}), \tag{2}$$

where $\mathcal{S} = \{\mathcal{G}_l\}_{l=1}^{|\mathcal{S}|}$ is a set of groups generated by a partition of atom indices, $\mathcal{G}_l \subseteq \{1, \ldots, m\}$ is a subset of atoms indices, and ψ is the group sparsity regularizer defined as

$$\psi_{\mathcal{S}}(\boldsymbol{C}) = \sum_{\mathcal{G}\in\mathcal{S}} \|\boldsymbol{C}_{[\mathcal{G}]}\|_F, \tag{3}$$

where $\boldsymbol{C}_{[\mathcal{G}]}$ denotes the coefficients of $\boldsymbol{C}$ which correspond to the atoms with indices $\mathcal{G}$. It can be seen from (3) that, each sparse code $\boldsymbol{c}_i$ is partitioned into several groups according to $\mathcal{S}$, and the mixed $\ell 1/\ell 2$ norm is computed on the grouped sparse codes to induce structured sparsity. Note that the group sparsity regularizer $\psi_{\mathcal{S}}(\boldsymbol{C})$ is not separable w.r.t each signal, making the sparse coding procedure collaborative. When $|\mathcal{S}| = 1$ and $\mathcal{G}_1 = \{1, \ldots, m\}$, it is easy to verify that $\psi_{\mathcal{S}}(\boldsymbol{c}) = \|\boldsymbol{c}\|_1$. As a generalization of ℓ_1-norm, $\psi_{\mathcal{S}}(\boldsymbol{c})$ encourages sparsity at group level instead of singleton level. By jointly employing ℓ_1 norm and $\psi_{\mathcal{S}}(\boldsymbol{c})$, the CHiLasso model (2) encourages sparsity both at group and singleton level while allowing the input signals share the same sparsity patterns at the group level but not necessarily at singleton level. Thus, the resulting structured sparsity is referred to as *collaborative hierarchical group sparsity*.

Solving the CHiLasso Problem (2) is challenging. Two effective algorithms with guaranteed convergence to global minimum have been proposed by Sprechmann et al. [36,39].[1] For brevity, these algorithms are summarized in Appendix A.

2.2. Dictionary learning for classification

Learning an adaptive dictionary can improve the efficiency of sparse coding. The dictionary learning problem can be generally formulated as follows:

$$\min_{\boldsymbol{D},\boldsymbol{C}} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 + \lambda\psi(\boldsymbol{C}) + \alpha\phi(\boldsymbol{D}), \tag{4}$$

where α is a scalar controlling the contribution of $\phi(\boldsymbol{D})$, and $\phi(\cdot)$ is the constraint on dictionary, e.g., indicator function forcing each column of dictionary to be normalized with unit norm.[2] The obtained sparse codes $\boldsymbol{C}$ can be directly used as features for classification. But separating dictionary learning from classifier construction is not optimal as sparse codes are not enforced to be discriminative for classification. One alternative is to combine dictionary learning and classifier construction in a unified framework (e.g. [16,17,22,27]), which can be formulated as follows:

$$\min_{\boldsymbol{D},\boldsymbol{W},\boldsymbol{C}} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 + \lambda \sum_{i=1}^{p} \psi(\boldsymbol{c}_i) + \gamma \sum_{i=1}^{p} \chi(h_i, f(\boldsymbol{c}_i, \boldsymbol{W})) + \beta\omega(W) + \alpha\phi(\boldsymbol{D}), \tag{5}$$

where $\boldsymbol{W} \in \mathbb{R}^{N\times m}$ denotes the classifier parameters, χ is the loss functional of classification (see Section 1.1 for some examples), ω is the regularization functional on $\boldsymbol{W}$ that determines the type of the classifier, β is a scalar controlling the contribution of ω, N is the total number of categories of training samples, and $h_i \in \{1, \ldots, N\}$ is the label of training sample $\boldsymbol{y}_i$.

3. Our method

In this section, we propose a discriminative dictionary learning model based on joint within-class collaborative hierarchical sparse representation. Hierarchical group sparsity promotion and classifier training are integrated into the dictionary learning process. With the employment of joint within-class collaborative hierarchical sparse coding, our method is able to learn dictionaries which encourage signals from the same category to share the same hierarchical group sparsity pattern. Besides, benefiting from joint dictionary learning and classifier training, discriminability of sparse codes is further strengthened. Then an alternating iterative scheme is presented for solving the proposed model.

3.1. Problem formulation

As mentioned in Section 1, correlation of data from the same category can be efficiently encoded with collaborative hierarchical group sparsity, i.e., sparsity at both singleton-level and group-level. Thus we can train a useful dictionary by inducing collaborative hierarchical sparse representation within each category of signals during dictionary learning. But in this way the interclass discrimination cannot be obtained, as similar categories may still have the potential to share the same active groups. In order to further enhance discriminability of the sparse codes, the training of a multi-class linear classifier is integrated into the dictionary learning process. Finally, we construct an effective dictionary learning models follows. Let $\mathcal{L} = \{\mathcal{P}_1, \ldots, \mathcal{P}_N\}$ denote the group partition of signals with $\mathcal{P}_k \subseteq \{1, \ldots, p\}$ as a subset of signal indices corresponding to kth category, $\mathcal{S} = \{\mathcal{G}_1, \ldots, \mathcal{G}_{|\mathcal{S}|}\}$ denote the group partition of atom indices satisfying $\bigcup_{l=1}^{|\mathcal{S}|} \mathcal{G}_l = \{1, \ldots, m\}$ and $\forall l \neq j,\ \mathcal{G}_l \cap \mathcal{G}_j = \varnothing$, and $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_p] \in \mathbb{R}^{N\times p}$ denote the binary labels of training samples with $\boldsymbol{h}_i = [0, 0, \ldots, 1, \ldots, 0] \in \mathbb{R}^N$ as the binary label vector of sample $\boldsymbol{y}_i$. Then our model, called Collaborative HIerarchicaL Discriminative Dictionary Learning (CHILD-DL), is defined as

$$\operatorname*{argmin}_{\boldsymbol{D},\boldsymbol{W},\boldsymbol{C}} \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 + \lambda_1 \sum_{i=1}^{p} \|\boldsymbol{c}_i\|_1 + \lambda_2 \sum_{k=1}^{N} \psi_{\mathcal{S}}(\boldsymbol{C}_{(\mathcal{P}_k)}) + \frac{\gamma}{2}\|\boldsymbol{H} - \boldsymbol{W}\boldsymbol{C}\|_F^2, \tag{6}$$

where $\lambda_1, \lambda_2, \gamma$ are the scalars controlling the relative contribution of each term, $\boldsymbol{C}_{(\mathcal{P}_k)}$ is a submatrix of $\boldsymbol{C}$ that contains the sparse codes of signals from the k-th category, and $\boldsymbol{W}$ is a multi-class linear classifier to be learned.[3]

[1] In the experimental section, the performance of both the algorithms will be investigated.

[2] Such normalization constraint on the norm of each atom is often required in dictionary learning methods. For brevity, we omit this unit norm constraint in the following sections if unnecessary.

[3] Similar in spirit to the D-KSVD and LC-KSVD methods [22,25], instead of using explicit regularization in the model, we implicitly control the energy of $\boldsymbol{W}$ in the optimization procedure followed by a renormalization stage.

In the CHILD-DL model (6), the joint within-class collaborative hierarchical group sparsity induced by $\lambda_1\sum_{i=1}^{p}\|\boldsymbol{c}_i\|_1 + \lambda_2\sum_{k=1}^{N}\psi_{\mathcal{S}}(\boldsymbol{C}_{(\mathcal{P}_k)})$ adapts the learned dictionary to the underlying structures of input signals and enforces signals from the same category to share the same hierarchical sparsity patterns. This enhances the separability of the sparse codes associated with different signal categories and benefits the improvement of classification performance. On the other hand, reducing the classification error of sparse codes according to $\|\boldsymbol{H}-\boldsymbol{W}\boldsymbol{C}\|_F^2$ can not only help to enhance discriminability of sparse codes, but also encourage signals from different categories to use distinct active groups, making the collaborative hierarchical group sparse representation more useful in classification.

3.2. Optimization

Solving the minimization problem of (6) is nontrivial. Intuitively, we need to alternately estimate the unknown variables one at a time. To make the optimization procedure more efficient, we propose to update the dictionary $\boldsymbol{D}$ and classifier $\boldsymbol{W}$ simultaneously, which is similar to schemes used in [22,25]. For this purpose, we rewrite (6) as

$$\begin{aligned}&\underset{\boldsymbol{D},\boldsymbol{W},\boldsymbol{C}}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{C}\|_F^2+\lambda_1\sum_{i=1}^{p}\|\boldsymbol{c}_i\|_1+\lambda_2\sum_{k=1}^{N}\psi_{\mathcal{S}}(\boldsymbol{C}_{(\mathcal{P}_k)}),\\ &\text{s.t.}\quad \forall j,\ \|\boldsymbol{u}_j\|_2=1,\end{aligned}\tag{7}$$

where $\boldsymbol{X}=\begin{bmatrix}\boldsymbol{Y}\\ \sqrt{\gamma}\boldsymbol{H}\end{bmatrix}$ and $\boldsymbol{U}=[\boldsymbol{u}_1,\ldots,\boldsymbol{u}_m]=\begin{bmatrix}\boldsymbol{D}\\ \sqrt{\gamma}\boldsymbol{W}\end{bmatrix}$.

Then we find the optimal solutions for $\boldsymbol{U}$ and $\boldsymbol{C}$ in (7) with an alternating iteration scheme. Such a scheme divides the minimization problem into several simpler ones in each iteration. The iteration stops until either of the following stopping criteria is satisfied: (1) the change of objective functional are small enough; (2) the maximum iteration number has been reached.

3.2.1. Class-wise collaborative hierarchical group sparse approximation

At the beginning of the $(\ell+1)$ iteration, with dictionary $\boldsymbol{U}^{(\ell)}$ fixed, the update of the structured sparse codes $\boldsymbol{C}$ is as follows:

$$\boldsymbol{C}^{(\ell+1)}=\underset{\boldsymbol{C}}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{U}^{(\ell)}\boldsymbol{C}\|_F^2+\lambda_1\sum_{i=1}^{p}\|\boldsymbol{c}_i\|_1+\lambda_2\sum_{k=1}^{N}\psi_{\mathcal{S}}(\boldsymbol{C}_{(\mathcal{P}_k)}),\tag{8}$$

which is group separable and can be decomposed into N independent subproblems:

$$\boldsymbol{C}_{(\mathcal{P}_k)}^{(\ell+1)}=\underset{\boldsymbol{C}\in\mathbb{R}^{m\times|\mathcal{P}_k|}}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{X}_{(\mathcal{P}_k)}-\boldsymbol{U}^{(\ell)}\boldsymbol{C}\|_F^2+\lambda_1\sum_{i\in\mathcal{P}_k}\|\boldsymbol{c}_i\|_1+\lambda_2\psi_{\mathcal{S}}(\boldsymbol{C}).\tag{9}$$

This is actually the CHiLasso problem (2) and can be solved by the algorithms proposed in [36,39]. Interested readers can refer to Appendix A[4] or the original literature for more details.

3.2.2. Dictionary refinement via projected gradient descent

With fixed $\boldsymbol{C}^{(\ell+1)}$ from the previous step, the dictionary $\boldsymbol{U}$ is updated as follows:

$$\boldsymbol{U}^{(\ell+1)}=\underset{\boldsymbol{U}}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{C}^{(\ell+1)}\|_F^2\quad\text{s.t. }\forall j,\ \|\boldsymbol{u}_j\|_2=1.\tag{10}$$

By applying the projected gradient descent method, we update $\boldsymbol{U}^{(\ell+1)}=[\boldsymbol{u}_1^{(\ell+1)},\ldots,\boldsymbol{u}_m^{(\ell+1)}]$ column by column as follows:

$$\boldsymbol{u}_j^{(\ell+1)}\in\underset{\|\boldsymbol{u}_j\|_2=1}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{u}_j-\boldsymbol{s}_j^{(\ell)}\|_2,\quad j=1,\ldots,m,\tag{11}$$

where

$$\boldsymbol{s}_j^{(\ell)}=\boldsymbol{u}_j^{(\ell)}-\frac{1}{\mu_j^{\ell}}\nabla_{\boldsymbol{u}_j}Q(\boldsymbol{C}^{(\ell+1)},\widetilde{\boldsymbol{U}}_j^{(\ell)}),\tag{12}$$

where μ_j^{ℓ} is the step size, $Q(\boldsymbol{C},\boldsymbol{U})=\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{C}\|_F^2$ and $\widetilde{\boldsymbol{U}}_j^{(\ell)}=[\boldsymbol{u}_1^{(\ell+1)},\ldots,\boldsymbol{u}_{j-1}^{(\ell+1)},\boldsymbol{u}_j^{(\ell)},\boldsymbol{u}_{j+1}^{(\ell)},\ldots,\boldsymbol{u}_m^{(\ell)}]$. By direct calculation, the problem of (11) has a closed-form solution

$$\boldsymbol{u}_j^{(\ell+1)}=\boldsymbol{s}_j^{(\ell)}/\|\boldsymbol{s}_j^{(\ell)}\|_2.\tag{13}$$

3.3. Classification strategy

By the above algorithm we can obtain the solution $\bar{\boldsymbol{U}}$ in (7), which is column-wise normalized and can be directly decomposed into $\bar{\boldsymbol{D}}$ and $\bar{\boldsymbol{W}}$. Then the final learned dictionary $\boldsymbol{D}$ and classifier $\boldsymbol{W}$ are computed by renormalization:

$$\begin{aligned}\boldsymbol{D}&=\{\boldsymbol{d}_1,\boldsymbol{d}_2,\ldots,\boldsymbol{d}_m\}=\left\{\frac{\bar{\boldsymbol{d}}_1}{\|\bar{\boldsymbol{d}}_1\|_2},\frac{\bar{\boldsymbol{d}}_2}{\|\bar{\boldsymbol{d}}_2\|_2},\ldots,\frac{\bar{\boldsymbol{d}}_m}{\|\bar{\boldsymbol{d}}_m\|_2}\right\};\\ \boldsymbol{W}&=\{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m\}=\left\{\frac{\bar{\boldsymbol{w}}_1}{\|\bar{\boldsymbol{d}}_1\|_2},\frac{\bar{\boldsymbol{w}}_2}{\|\bar{\boldsymbol{d}}_2\|_2},\ldots,\frac{\bar{\boldsymbol{w}}_m}{\|\bar{\boldsymbol{d}}_m\|_2}\right\}.\end{aligned}\tag{14}$$

Given a test sample $\boldsymbol{y}_{test}$, we compute the sparse code $\boldsymbol{c}_{test}$ as follows:

$$\boldsymbol{c}_{test}=\underset{\boldsymbol{c}\in\mathbb{R}^m}{\operatorname{argmin}}\|\boldsymbol{y}_{test}-\boldsymbol{D}\boldsymbol{c}\|_2^2+\lambda_1\|\boldsymbol{c}\|_1+\lambda_2\psi_{\mathcal{S}}(\boldsymbol{c}),\tag{15}$$

which is a special case of the CHiLasso model (2). Once $\boldsymbol{c}_{test}$ is computed, the learned linear classifier $\boldsymbol{W}$ is applied to $\boldsymbol{c}_{test}$ to generate a label prediction vector $\boldsymbol{l}_{test}$ on $\boldsymbol{y}_{test}$ by

$$\boldsymbol{l}_{test}=\boldsymbol{W}\boldsymbol{c}_{test}.\tag{16}$$

For the values of $\boldsymbol{l}_{test}$ are unnecessarily binary, the final label of $\boldsymbol{y}_{test}$ is set to be the index which corresponding to the largest element of $\boldsymbol{l}_{test}$.

3.4. Initialization and configuration

In our implementation, the initial dictionary $\boldsymbol{D}^{(0)}$ is generated by random sampling of training data. More precisely, a certain number of samples from each category are randomly picked up and combined as one group of the dictionary. The sparse codes $\boldsymbol{C}^{(0)}$ is initialized by solving

$$\underset{\boldsymbol{C}}{\operatorname{argmin}}\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{C}\|_F^2,\quad\text{s.t. }\forall i,\|\boldsymbol{c}_i\|_0\leqslant T,\tag{17}$$

which is solved by the OMP algorithm[5] [45]. Then the multi-class linear classifier $\boldsymbol{W}$ is initialized by

$$\boldsymbol{W}^{(0)}=\underset{\boldsymbol{W}}{\operatorname{argmin}}\frac{\gamma}{2}\|\boldsymbol{H}-\boldsymbol{W}\boldsymbol{C}^{(0)}\|_F^2+\frac{\beta}{2}\|\boldsymbol{W}\|_F^2,\tag{18}$$

which is the ridge regression with explicit solution

$$\boldsymbol{W}^{(0)}=\boldsymbol{H}\boldsymbol{C}^{(0)\top}\left(\boldsymbol{C}^{(0)}\boldsymbol{C}^{(0)\top}+\frac{\beta}{\gamma}\boldsymbol{I}_m\right)^{-1}.\tag{19}$$

[4] Two algorithms are mentioned in Appendix A. In Section 4, the performance of both the algorithms will be tested and compared.

[5] In our experiments we found that using pseudo-inverse solution of $\operatorname{argmin}\|\boldsymbol{Y}-\boldsymbol{D}^{(0)}\boldsymbol{C}\|_F^2$ instead for the initialization has almost no influence on the classification results.

4. Experiments

We evaluate the both the effectiveness and efficiency of our CHILD-DL method by applying it to face recognition on the Extended YaleB dataset [46] and the AR face dataset [47], object recognition on the Caltech-101 dataset [48], and scene classification on the Scene-15 dataset [49]. The evaluation protocols are consistent with [25], including the experimental setup and the employed image features,[6] which is detailed in the following subsections.

4.1. Experimental configuration

4.1.1. Methods for comparison

The methods for comparison mainly include K-SVD [10], D-KSVD [22], LC-KSVD [25], SRC [1], FDDL [23], and LLC [3].[7] The SRC method is implemented with two different dictionary sizes: (1) the original version denoted by SRC that stacks all the training samples as a dictionary; (2) the reduced version denoted by SRC* whose dictionary size is the same as ours. As there are two algorithms for solving the subproblem defined in (9), we derive two CHILD-DL methods, which is denoted by CHILD-DL-A and CHILD-DL-B respectively. In the experiments, we compared these two methods to distinguish which one is better.

Besides, to demonstrate the necessity to jointly learn dictionary and classifier in our method, we implemented a baseline method denoted by *Baseline* for comparison. The baseline method separates the dictionary learning and classifier training as follows: (1) Setting $\gamma = 0$ in (7), we learn a dictionary $\boldsymbol{D}$ without training $\boldsymbol{W}$ and also the sparse codes $\boldsymbol{C}$ on all training samples; (2) The sparse codes $\boldsymbol{C}$ and the label matrix $\boldsymbol{H}$ are used to train a multi-class linear classifier via ridge regression; (3) The test stage is done via (15) and (16).

Considering the randomness in partitioning training and testing sets, we run all experiments 10 times and report the averages as well as the standard deviations of the prediction accuracies. For other compared methods, we only report the standard deviations where available.

4.1.2. Parameter setting

The parameters of CHILD-DL include the regularization parameters, the dictionary size, and the configuration of groups. The setting of these parameters on each dataset is summarized in Table 1. Note that the CHILD-DL-A, CHILD-DL-B and Baseline methods are using the same parameter setting.

Regularization parameters. The choice of regularization parameters in our model (7), like λ_1, λ_2 and γ, depends on the application and data. In all experiments, if no specific instructions mentioned, we use fivefold cross validation to find the parameters of the proposed model that give the best results while avoiding over-fitting. We also tested the effects of regularization parameter selection, which is detailed in Section 4.2.

Dictionary size. As shown in [23,50], the larger the dictionary size is, the better the performance of the dictionary learning methods can be achieved. To have a fair comparison in all the experiments, the dictionary size of all the compared methods mentioned above except SRC are set to be the same as [25].

Group configuration. The configuration of groups $\mathcal{S}$ is another key factor in our method. Although the configuration of groups are independent of the signal categories, it is natural to expect that signals from different categories could use distinct groups in the context of classification. For simplicity, we set $|\mathcal{S}|$ to be equal to the number of categories N. The set of groups $\mathcal{S}$ is set to satisfy the following three conditions: (1) $\bigcup_{i=1}^{N} \mathcal{G}_i = \{1, \ldots, m\}$; (2) $\forall i \neq j,\ \mathcal{G}_i \cap \mathcal{G}_j = \varnothing$; (3) $\forall k,\ |\mathcal{G}_k| = r = m/N$. Although we do not employ group overlap here,[8] different categories are still allowed to share atoms during the dictionary learning process. The number of active groups shared by signals from the same category is implicitly determined by the scalars λ_1, λ_2 and γ.

Table 1
Detailed parameter setting of CHILD-DL on each dataset.

Dataset	λ_1	λ_2	γ	No. of training samples per class	m	r
Extended YaleB	0.02	0.02	5	32	570	15
AR face	0.02	0.02	0.05	20	500	5
Caltech-101	0.008	0.006	0.01	5	510	5
				10	1020	10
				15	1530	15
				20	2040	20
				25	2550	25
				30	3060	30
Scene-15	0.02	0.02	2	100	450	30

4.2. Recognition on the extended YaleB dataset

The Extended YaleB dataset [46] contains 2,414 images of 38 human frontal faces under about 64 illumination conditions and expressions, as shown in Fig. 3(a). There are about 64 images for each person. The original images were cropped to 192×168 pixels. Each face image is projected into a 504-dimensional feature vector using a random matrix of zero-mean normal distribution [22]. We randomly select half of the images for training and the other half for testing in each class.

The experimental results are summarized in Table 2. We can see that CHILD-DL achieved very competitive results among all the compared methods. The CHILD-DL-B method performed better than the baseline method and the CHILD-DL-A method. The CHILD-DL-B method outperformed many state-of-the-art approaches except FDDL and SRC. But note that the performance of the SRC method degrades dramatically when using dictionary of the same size as ours (see SRC*). Although our method performed worse than the FDDL method, it can be seen in Section 4.6 that our method is more efficient compared with the FDDL method.

The sparsity regularization parameters λ_1 and λ_2 as well as the discrimination regularization parameter γ are determined by fivefold cross validation. To analyze the parameter sensitivity of our method, we conduct a test on the Extended YaleB dataset by adjusting the regularization parameters λ_1 and γ while fixing ratio between λ_2 and λ_1.[9] The effects of parameter selection are shown in Fig. 4. We can observe that good performance is achieved at $\lambda_1 = 0.02$ and $\gamma = 5$.

4.3. Recognition on the AR face dataset

The AR face dataset [47] consists of over 4000 frontal images from 126 individuals. For each individual, 26 pictures were taken in tow separate sessions. The main characteristic of the AR dataset is that it includes frontal views of faces with different facial expressions, lighting and occlusion conditions, as shown in Fig. 3(b). Same as the standard evaluation procedure from [22], we use a

[6] All the data for the experiments are provided by Jiang et al. and available on the website: http://www.umiacs.umd.edu/zhuolin/projectlcksvd.html.

[7] In [25], two versions of LC-KSVD are presented. Here we select the improved version with better performance reported. The parameter setting and results of all these compared methods are consistent with [25].

[8] It is worth mentioning that our model is not restricted to such a simple group assignment scheme.

[9] We fix the ratio between λ_2 and λ_1 as 1 and test the CHILD-DL-B algorithm.

(a) Extended YaleB (b) AR face

Fig. 3. Some samples of two face datasets for evaluation. (a) The extended YaleB dataset; (b) the AR face dataset.

Table 2
The performance of the compared methods (recognition accuracy in %) on the Extended YaleB dataset.

Method	Accuracy	Method	Accuracy
K-SVD [10]	93.10	SRC* [1]	80.50
D-KSVD [22]	94.10	SRC [1]	97.20
LLC [3]	90.70	LC-KSVD [25]	95.00
FDDL [23]	**98.07** ± 0.40	Baseline	94.62 ± 0.92
CHILD-DL-A	95.53 ± 0.60	CHILD-DL-B	97.17 ± 0.66

Table 3
The performance of the compared methods (recognition accuracy in %) on the AR face dataset.

Method	Accuracy	Method	Accuracy
K-SVD [10]	86.50	SRC* [1]	66.50
D-KSVD [22]	88.80	SRC [1]	**97.50**
LLC [3]	88.70	LC-KSVD [25]	93.70
FDDL [23]	97.45 ± 0.65	Baseline	88.23 ± 1.27
CHILD-DL-A	95.40 ± 0.62	CHILD-DL-B	95.31 ± 0.73

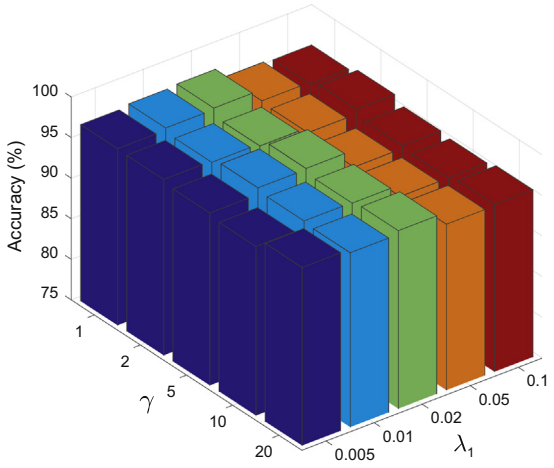


Fig. 4. Effects of parameter selection of λ_1 and γ on the recognition accuracy (%) on the extended YaleB dataset.

subset of the dataset consisting of 2,600 images from 50 male subjects and 50 female subjects. For each person, twenty images are randomly picked up for training and the rest for testing. Each face image is cropped to 165×120 and then projected onto a 540-dimensional feature vector.

The experimental results are summarized in Table 3. Similar to what happened on the YaleB dataset, CHILD-DL outperformed the baseline method and several state-of-the-art approaches including K-SVD, D-KSVD, LLC and LC-KSVD, but did not perform as well as the FDDL and SRC methods. The CHILD-DL-A method is slightly better than the CHILD-DL-B method in this case.

4.4. Recognition on the Caltech-101 dataset

The Caltech-101 dataset [48] is a large dataset, which contains 8677 images in 101 object categories with different shapes and appearances and 467 images selected from an additional background category. The number of images per category varies greatly from 31 to 800. Some sample images selected from the caltech101 dataset are illustrated in Fig. 5(a). As recommended in [25], the spatial pyramid features [49] based on SIFT descriptors are extracted and the dimension of each feature is further reduced to be 3000 via PCA.

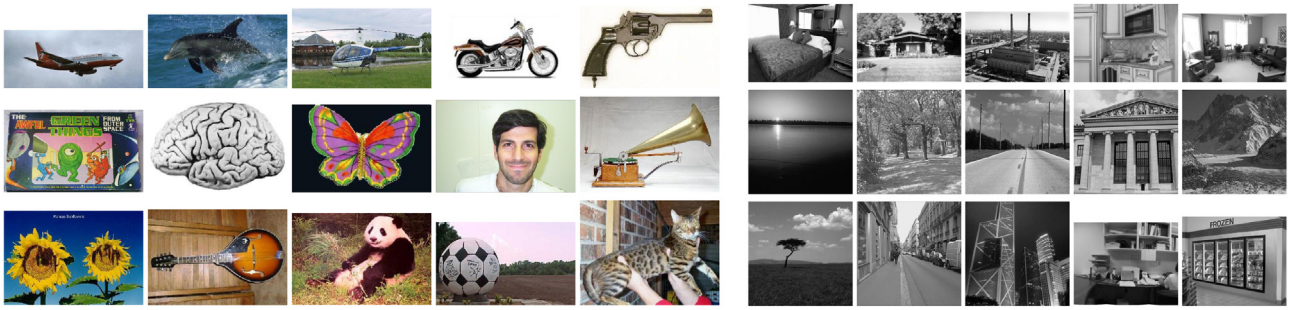
Following the experimental settings in [25], we randomly pick up 5, 10, 15, 20, 25 and 30 samples per category for training the dictionary as well as the classifier, and test on the remaining samples. The dictionary size of our method is set proportional to the size of training set per category, as shown in Table 1. Besides the compared methods used in the above face recognition, the reported results of other state-of-the-art approaches [2,17,49,51–55] are also involved for comparison.

See Table 4 for the performance comparison. We can see that both the CHILD-DL-A and CHILD-DL-B methods performed better than the baseline method and each has its own advantages in cases with certain number of training samples per category. Besides, CHILD-DL outperformed many competitive supervised dictionary learning methods and some other state-of-the-art approaches, while achieving accuracy comparable to FDDL and LC-KSVD.

4.5. Recognition on the Scene-15 dataset

The Scene-15 dataset introduced in [49] contains a wide range of outdoor and indoor scenes. The scene categories include bedroom, suburb, industrial, kitchen, living room, coast, forest, highway, inside cite, mountain, open country, street, tall building, office and store. See Fig. 5(b) for the sample images per category. The number of images per category varies from 210 to 410, and the resolution of each image is about 250×300. Similar to the feature extraction process on the Caltech-101 dataset, the spatial pyramid features [49] based on SIFT descriptors are extracted and the dimension of each feature is further reduced to be 3000 via PCA.

Following the experimental settings recommended in [25], we randomly select 100 images per category as training data and use the rest as test data. Besides the methods used in the above face recognition experiments, several state-of-the-art scene classi-



(a) Caltech-101 (b) Scene-15

Fig. 5. Some samples from (a) the Caltech-101 dataset; (b) the Scene-15 dataset.

Table 4
Recognition results (classification accuracy in %) using spatial pyramid features on the Caltech-101 Dataset.

Method	Number of training samples					
	5	10	15	20	25	30
Malik et al. [51]	45.60	54.80	59.05 ± 0.56	62.00	–	66.23 ± 0.48
Lazebnik et al. [49]	–	–	56.40	–	–	64.60 ± 0.80
Griffin [52]	44.20	54.50	59.00	63.30	65.80	67.60
Irani [53]	–	–	65.00 ± 1.14	–	–	70.40
Grauman [54]	–	–	61.00	–	–	69.60
Pham [17]	–	–	42.00 ± 1.00	–	–	–
Xu et al. [56]	53.60	64.01	69.15	72.40	74.52	76.22
Yang et al. [2]	–	–	67.00 ± 0.45	–	–	73.20 ± 0.54
LLC [3]	51.15	59.77	65.43	67.74	70.16	73.44
SRC [1]	48.80	60.10	64.90	67.70	69.20	70.70
K-SVD [10]	49.80	59.80	65.20	68.70	71.00	73.20
D-KSVD [22]	49.60	59.50	65.10	68.60	71.10	73.00
FDDL [23]	53.87 ± 0.57	63.27 ± 0.59	67.61 ± 0.51	70.65 ± 0.48	71.94 ± 0.49	73.21 ± 0.46
LC-KSVD [25]	54.00	63.10	67.70	70.50	72.30	73.60
Baseline	52.78 ± 1.12	62.95 ± 0.92	67.53 ± 0.85	70.33 ± 0.76	72.13 ± 0.81	73.41 ± 0.73
CHILD-DL-A	54.14 ± 0.62	63.35 ± 0.63	**68.09** ± 0.65	**70.82** ± 0.58	72.61 ± 0.56	**73.69** ± 0.55
CHILD-DL-B	**54.22** ± 0.73	**63.39** ± 0.69	67.94 ± 0.68	70.73 ± 0.46	**72.74** ± 0.56	73.58 ± 0.63

Table 5
The performance of the compared methods (recognition accuracy in %) on the Scene-15 dataset.

Method	Accuracy	Method	Accuracy
Lazebnik et al. [49]	81.40 ± 0.50	LLC [3]	89.20
Yang et al. [2]	80.28 ± 0.93	SRC [1]	91.80
Boureau et al. [57]	84.30 ± 0.50	LC-KSVD [25]	92.90
Gao et al. [58]	89.75 ± 0.50	FDDL [23]	98.35 ± 0.21
Lian et al. [21]	86.43 ± 0.41	Baseline	95.82 ± 0.50
K-SVD [10]	86.70	CHILD-DL-A	96.42 ± 0.51
D-KSVD [22]	89.10	CHILD-DL-B	**98.72** ± 0.24

Table 6
Training time (seconds per iteration) and test time (milliseconds) of several dictionary learning methods on the Extended YaleB dataset.

Running time	D-KSVD [22]	LC-KSVD [25]	SRC [1]	FDDL [23]	CHILD-DL-A	CHILD-DL-B
Training (s)	0.86	1.68	–	471.92	355.58	10.37
Test (ms)	0.35	0.37	48.41	2523.72	286.94	226.02

fication approaches [2,21,49,57,58] with available results are included for comparison.

The experimental results on Scene-15 dataset is are listed in Table 5. It can be seen that the CHILD-DL-B method is slightly better than the FDDL method and achieved the best average recognition accuracy among all the compared methods.

4.6. Efficiency

In order to evaluate the complexity of our method, we compare the computational efficiency of several tested methods above, in terms of the average running time on the Extended YaleB dataset during the training phase and the testing stage. More specifically, for each tested method, both the average training time per iteration during dictionary learning and the average test time for an test image during classification are reported. To have a fair comparison, all the tested methods are implemented under the same computational environment. The software environment is the MATLAB 2014a platform run on the Windows 8 operating system, and the hardware platform is a laptop computer with Intel Dual-Core i7-2640 M 2.8 GHz CPU and 8 GB memory.

From Table 6, it can be seen that CHILD-DL is much better than the FDDL method in terms of computational time, but worse than D-KSVD, LC-KSVD and SRC methods. It can be also seen that CHILD-DL-B is much faster than CHILD-DL-A, especially in terms of training time.

5. Conclusions

Learning discriminative sparse representations from labeled data has drawn much interest in computer vision community. In order to obtain efficient representations with structured sparsity and strong discriminability for image classification, we introduced the concept of collaborative hierarchical group sparsity and the integration of classification feedback into discriminative dictionary learning. Our

dictionary learning model is constructed by combining reconstruction error, joint within-class hierarchical group sparsity and linear prediction error into a unified minimization framework. Benefiting from using the joint within-class collaborative hierarchical sparsity, our method encourages signals from the same category to share the same hierarchical sparsity patterns, which promotes the separability of the sparse codes associated with different signal categories. The discriminability of sparse codes is further strengthened by joint dictionary construction and classifier learning. An efficient alternating iterative scheme is developed to solve the proposed model. Our method is applied to image classification by simultaneously learning dictionary, sparse representation and classifier from image features. We applied our method to face recognition, object classification and scene recognition. The experimental results have demonstrated the excellent performance of our method. In future, we would like to investigate higher-level structured sparsity for discriminative dictionary learning.

Appendix A. Numerical solvers for CHiLasso

Two Effective solvers for the ChiLasso Problem (2) have been proposed by Sprechman et al. in [36,39]. We denote these two solvers as ChiLasso-Solver-A and ChiLasso-Solver-B respectively.

ChiLasso-Solver-A [36]. The basic idea of ChiLasso-Solver-A is to use ADMOM [59] iterations to divide the overall sparse coding problem into two subproblems: (1) breaking the multi-signal case into p single-signal ℓ_1 regressions; (2) treating the multi-signal case as a single group Lasso-like problem. For this purpose, we rewrite the ChiLasso Problem (2) as a constrained optimization model

$$\underset{\boldsymbol{C}}{\operatorname{argmin}}\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{C}\|_F^2+\lambda_1\sum_i\|\boldsymbol{c}_i\|_1+\lambda_2\psi_{\mathcal{S}}(\boldsymbol{B})\quad\text{s.t.}\quad\boldsymbol{C}=\boldsymbol{B}. \tag{A.1}$$

Then the ADMOM iterations for solving (A.1) are given as follows:

$$\boldsymbol{C}^{(t+1)}=\arg\min_{\boldsymbol{C}}\frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{C}\|_F^2+\lambda_1\sum_i\|\boldsymbol{c}_i\|_1+\operatorname{Tr}(\boldsymbol{C}^\top\boldsymbol{P}^{(t)})+\frac{\theta}{2}\|\boldsymbol{B}^{(t)}-\boldsymbol{C}\|_F^2; \tag{A.2}$$

$$\boldsymbol{B}^{(t+1)}=\arg\min_{\boldsymbol{B}}\frac{\theta}{2}\|\boldsymbol{B}-\boldsymbol{C}^{(t+1)}\|_F^2+\operatorname{Tr}(\boldsymbol{B}^\top\boldsymbol{P}^{(t)})+\lambda_2\psi_{\mathcal{S}}(\boldsymbol{B}); \tag{A.3}$$

$$\boldsymbol{P}^{(t+1)}=\boldsymbol{P}^{(t)}+\theta(\boldsymbol{C}^{(t+1)}-\boldsymbol{B}^{(t+1)}). \tag{A.4}$$

The problem of (A.2) is signal-signal separable and thus can be solved by updating $\boldsymbol{C}$ column by column, i.e.

$$\boldsymbol{c}_i^{(t+1)}=\arg\min_{\boldsymbol{c}}\frac{1}{2}\|\boldsymbol{y}_i-\boldsymbol{D}\boldsymbol{c}\|_2^2+\lambda_1\|\boldsymbol{c}_i\|_1+\boldsymbol{p}_i^\top\boldsymbol{c}+\frac{\theta}{2}\|\boldsymbol{b}_i^{(t)}-\boldsymbol{c}\|_2^2, \tag{A.5}$$

which can be solved by applying SpaRSA [60]. The problem of (A.3) is group separable and thus can be separated into $|\mathcal{S}|$ optimization problems in vectorial form as follows:

$$\arg\min_{\boldsymbol{f}}\lambda_2\|\boldsymbol{f}\|_2-\boldsymbol{q}^\top\boldsymbol{f}+\frac{\theta}{2}\|\boldsymbol{z}-\boldsymbol{f}\|_2^2, \tag{A.6}$$

where $\boldsymbol{f},\boldsymbol{z}$ and $\boldsymbol{q}$ are column vectors by concatenating the columns of $\boldsymbol{B}_{(\mathcal{P}_k)},\boldsymbol{C}_{(\mathcal{P}_k)}^{(t+1)}$ and $\boldsymbol{P}_{(\mathcal{P}_k)}^{(t)}$ respectively. This minimization problem can be solved by simple vectorial thresholding, i.e.,

$$\boldsymbol{f}=\begin{cases}\frac{\max\{0,\|\boldsymbol{z}+\theta\boldsymbol{q}\|_2-\lambda_2\}}{\theta\|\boldsymbol{z}+\theta\boldsymbol{q}\|_2}(\boldsymbol{z}+\theta\boldsymbol{q}) & \text{if } \|\boldsymbol{z}+\theta\boldsymbol{q}\|_2>0\\ \boldsymbol{0} & \text{if } \|\boldsymbol{z}+\theta\boldsymbol{q}\|_2=0\end{cases}. \tag{A.7}$$

ChiLasso-Solver-B [39]. In [39], the SpaRSA framework is employed to generate a sequence of iterates $\{\boldsymbol{C}^{(t)}\}_{t\in\mathbb{N}}$, which converges to the solution of (2) under certain conditions. At each iteration, $\boldsymbol{C}^{(t+1)}$ is obtained by solving:

$$\min_{\boldsymbol{Z}}\frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{V}^{(t)}\|_2^2+\frac{\lambda_1}{\alpha^{(t)}}\sum_i\|\boldsymbol{z}_i\|_1+\frac{\lambda_2}{\alpha^{(t)}}\psi_{\mathcal{G}}(\boldsymbol{Z}), \tag{A.8}$$

where $\boldsymbol{V}^{(t)}=[\boldsymbol{v}_1^{(t)},\ldots,\boldsymbol{v}_p^{(t)}]$ is defined a matrix with its ith column given by $\boldsymbol{v}_i^{(t)}=\boldsymbol{c}_i^{(t)}-\frac{1}{\alpha^{(t)}}\boldsymbol{D}^T(\boldsymbol{D}\boldsymbol{c}_i^{(t)}-\boldsymbol{y}_i)$, and $\{\alpha^{(t)}\}_{t\in\mathbb{N}}$ is some sequence of parameters with $\alpha^{(t)}\in\mathbb{R}^+$ which determine the convergence conditions about the algorithm. In the aforementioned formulation, all terms in the cost function can be group separable. Thus, the problem of (A.8) can be solved independently for each group, that is

$$\boldsymbol{C}_{(\mathcal{P}_k)}^{(t+1)}=\arg\min_{\boldsymbol{Z}}\frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{V}_{(\mathcal{P}_k)}^{(t)}\|_F^2+\frac{\lambda_1}{\alpha^{(t)}}\sum_i\|\boldsymbol{z}_i\|_1+\frac{\lambda_2}{\alpha^{(t)}}\|\boldsymbol{Z}\|_F. \tag{A.9}$$

The sub-gradient of (A.9) for the case where the optimum $\boldsymbol{Z}^*\neq\boldsymbol{0}$ is inspected as $\boldsymbol{V}_{(\mathcal{P}_k)}^{(t)}-(1+\frac{\lambda_2}{\alpha^{(t)}\|\boldsymbol{Z}^*\|_F})\boldsymbol{Z}^*\in\frac{\lambda_1}{\alpha^{(t)}}\partial\|\boldsymbol{Z}^*\|_1$. It can be observed that each element of $(1+\frac{\lambda_2}{\alpha^{(t)}\|\boldsymbol{Z}^*\|_F})\boldsymbol{Z}^*$ is the solution of the well known scalar soft thresholding operator. We can set $\boldsymbol{G}=T_{\frac{\lambda_1}{\alpha^{(t)}}}(\boldsymbol{V}_{(\mathcal{P}_k)}^{(t)})$, where $T_\lambda(\boldsymbol{X})$ denotes the matrix obtained when applying the soft-thresholding operator with parameter λ to each element of $\boldsymbol{X}$. From the information above, it can be easily inferred that $\|\boldsymbol{Z}^*\|_F^2=\frac{\|\boldsymbol{Z}^*\|_F^2}{(\|\boldsymbol{Z}^*\|_F+\frac{\lambda_2}{\alpha^{(t)}})^2}\|\boldsymbol{G}\|_F^2$ and then obtain $\|\boldsymbol{Z}^*\|_F=\|\boldsymbol{G}\|_F-\frac{\lambda_2}{\alpha^{(t)}}$. Since all terms are positive, this can only hold as $\|\boldsymbol{G}\|_F>\frac{\lambda_2}{\alpha^{(t)}}$, which shows a vectorial thresholding condition on the solution $\boldsymbol{Z}^*$ in terms of $\|\boldsymbol{G}\|_F$. It's easy to show that $\|\boldsymbol{G}\|_F<\frac{\lambda_2}{\alpha^{(t)}}$ is a sufficient condition for $\boldsymbol{Z}^*=\boldsymbol{0}$.

Therefore, the corresponding closed-form solution for each subproblem (A.9) is given by

$$\boldsymbol{C}_{(\mathcal{P}_k)}^{(t+1)}=\begin{cases}\frac{\max\{0,\|\boldsymbol{G}\|_F-\frac{\lambda_2}{\alpha^{(t)}}\}}{\|\boldsymbol{G}\|_F}\boldsymbol{G} & \text{if } \|\boldsymbol{G}\|_F>0\\ \boldsymbol{0} & \text{if } \|\boldsymbol{G}\|_F=0\end{cases}. \tag{A.10}$$

This ChiLasso-Solver-B provides such solution in closed-form, requiring just two thresholding, both linear in the dimension of $\boldsymbol{Y}$.

References

[1] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, Yi Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Machine Intell. 31 (2) (2009) 210–227.

[2] Jianchao Yang, Kai Yu, Yihong Gong, Thomas Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1794–1801.

[3] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, Yihong Gong, Locality-constrained linear coding for image classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3360–3367.

[4] Meng Yang, D. Zhang, Jian Yang, Robust sparse coding for face recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 625–632.

[5] Stéphane Mallat, A Wavelet Tour of Signal Processing, Academic Press, 1999.

[6] Michael Elad, Michal Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15 (12) (2006) 3736–3745.

[7] Jian-Feng Cai, Hui Ji, Chaoqiang Liu, Zuowei Shen, Blind motion deblurring from a single image using sparse approximation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 104–111.

[8] Jian-Feng Cai, Hui Ji, Zuowei Shen, Gui-Bo Ye, Data-driven tight frame construction and image denoising, Appl. Comput. Harmonic Anal. (2013).

[9] Kjersti Engan, Sven Ole Aase, J.H. Husoy, Frame based signal compression using method of optimal directions (mod), Proceedings of IEEE Conference on International Symposium on Circuits and Systems, vol. 4, IEEE, 1999, pp. 1–4.

[10] Michal Aharon, Michael Elad, Alfred Bruckstein, K-svd: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322.

[11] Julien Mairal, Francis Bach, Jean Ponce, Task-driven dictionary learning, IEEE Trans. Pattern Anal. Machine Intell. 34 (4) (2012) 791–804.

[12] Chenglong Bao, Hui Ji, Yuhui Quan, Zuowei Shen, l0 norm based dictionary learning by proximal methods with global convergence, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3858–3865.
[13] Ke Huang, Selin Aviyente, Sparse representation for signal classification, in: Advances in Neural Information Processing Systems, 2006, pp. 609–616.
[14] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, Andrew Y Ng, Self-taught learning: transfer learning from unlabeled data, in: Proceedings of International Conference on Machine Learning, ACM, 2007, pp. 759–766.
[15] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, Discriminative learned dictionaries for local image analysis, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[16] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, et al., Supervised dictionary learning, in: Advances in Neural Information Processing Systems, vol. 21, 2008, pp. 1033–1040.
[17] Duc-Son Pham, Svetha Venkatesh, Joint learning and dictionary construction for pattern recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[18] Koray Kavukcuoglu, M. Ranzato, Rob Fergus, Yann LeCun, Learning invariant features through topographic filter maps, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1605–1612.
[19] Wei Zhang, Akshat Surve, Xiaoli Fern, Thomas Dietterich, Learning non-redundant codebooks for classifying complex objects, in: Proceedings of International Conference on Machine Learning, ACM, 2009, pp. 1241–1248.
[20] Ignacio Ramirez, Pablo Sprechmann, Guillermo Sapiro, Classification and clustering via dictionary learning with structured incoherence and shared features, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3501–3508.
[21] Xiao-Chen Lian, Zhiwei Li, Bao-Liang Lu, Lei Zhang, Max-margin dictionary learning for multiclass image categorization, in: Proceedings of European Conference on Computer Vision, Springer, 2010, pp. 157–170.
[22] Qiang Zhang, Baoxin Li, Discriminative julien mairal k-svd for dictionary learning in face recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2691–2698.
[23] Meng Yang, David Zhang, Xiangchu Feng, Fisher discrimination dictionary learning for sparse representation, in: Proceedings of IEEE International Conference on Computer Vision, IEEE, 2011, pp. 543–550.
[24] Ning Zhou, Yi Shen, Jinye Peng, Jianping Fan, Learning inter-related visual dictionary for object recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3490–3497.
[25] Zhuolin Jiang, Zhe Lin, L. Davis, Label consistent k-svd: learning a discriminative dictionary for recognition, IEEE Trans. Pattern Anal. Machine Intell. 35 (11) (2013) 2651–2664.
[26] Meng Yang, Dengxin Dai, Lilin Shen, Luc Van Gool, Latent dictionary learning for sparse representation based classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 4138–4145.
[27] Jianchao Yang, Kai Yu, Thomas Huang, Supervised translation-invariant sparse coding, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3517–3524.
[28] Xiao-Chen Lian, Zhiwei Li, Changhu Wang, Bao-Liang Lu, Lei Zhang, Probabilistic models for supervised dictionary learning, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2305–2312.
[29] Ming Yuan, Yi Lin, Model selection and estimation in regression with grouped variables, J. Roy. Stat. Soc.: Ser. B (Stat. Methodol.) 68 (1) (2006) 49–67.
[30] Francis R. Bach, Exploring large feature spaces with hierarchical multiple kernel learning, in: Advances in Neural Information Processing Systems, 2009, pp. 105–112.
[31] Laurent Jacob, Guillaume Obozinski, Jean-Philippe Vert, Group lasso with overlap and graph lasso, in: Proceedings of International Conference on Machine Learning, ACM, 2009, pp. 433–440.
[32] Karim Lounici, Massimiliano Pontil, Alexandre B. Tsybakov, Sara Van De Geer, Taking advantage of sparsity in multi-task learning, in: Proceedings of Computational Learning Theory Conference, 2009.
[33] Junzhou Huang, Tong Zhang, et al., The benefit of group sparsity, Ann. Stat. 38 (4) (2010) 1978–2004.
[34] Seyoung Kim, Eric P. Xing, Tree-guided group lasso for multi-task regression with structured sparsity, in: Proceedings of International Conference on Machine Learning, 2010, pp. 543–550.
[35] Jun Liu, Jieping Ye, Moreau-yosida regularization for grouped tree structure learning, in: Advances in Neural Information Processing Systems, vol. 23, 2010, pp. 1459–1467.
[36] Pablo Sprechmann, Ignacio Ramirez, Guillermo Sapiro, Yonina Eldar, Collaborative hierarchical sparse modeling, in: Proceedings of Conference on Information Sciences and Systems, IEEE, 2010, pp. 1–6.
[37] Junzhou Huang, Tong Zhang, Dimitris Metaxas, Learning with structured sparsity, J. Machine Learn. Res. 12 (2011) 3371–3412.
[38] Rodolphe Jenatton, Jean-Yves Audibert, Francis Bach, Structured variable selection with sparsity-inducing norms, J. Machine Learn. Res. 12 (2011) 2777–2824.
[39] Pablo Sprechmann, Ignacio Ramirez, Guillermo Sapiro, Yonina C Eldar, C-hilasso: a collaborative hierarchical sparse modeling framework, IEEE Trans. Signal Process. 59 (9) (2011) 4183–4198.
[40] Angshul Majumdar, Rabab K Ward, Classification via group sparsity promoting regularization, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2009, pp. 861–864.
[41] Rodolphe Jenatton, Guillaume Obozinski, Francis Bach, Structured sparse principal component analysis, arXiv preprint arXiv:0909.1440, 2009.
[42] Kevin Rosenblum, Lihi Zelnik-Manor, Yonina Eldar, Dictionary optimization for block-sparse representations, in: AAAI Fall 2010 Symposium on Manifold Learning, 2010, pp. 50–58.
[43] Arthur Szlam, Karol Gregor, Yann LeCun, Fast approximations to structured sparse coding and applications to object classification, in: Proceedings of European Conference on Computer Vision, Springer, 2012, pp. 200–213.
[44] David G Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[45] Yagyensh Chandra Pati, Ramin Rezaiifar, P.S. Krishnaprasad, Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition, in: Proceedings of Asilomar Conference on Signals, Systems and Computers, IEEE, 1993, pp. 40–44.
[46] Athinodoros S. Georghiades, Peter N. Belhumeur, David Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Machine Intell. 23 (6) (2001) 643–660.
[47] Aleix M. Martinez, The ar face database, CVC Technical Report, 24, 1998.
[48] Li Fei-Fei, Rob Fergus, Pietro Perona, Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories, 2004.
[49] Svetlana Lazebnik, Cordelia Schmid, Jean Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, 2006, pp. 2169–2178.
[50] Zhuolin Jiang, Zhe Lin, Larry S. Davis, Learning a discriminative dictionary for sparse coding via label consistent k-svd, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 1697–1704.
[51] Hao Zhang, Alexander C Berg, Michael Maire, Jitendra Malik, Svm-knn: discriminative nearest neighbor classification for visual category recognition, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, 2006, pp. 2126–2136.
[52] Gregory Griffin, Alex Holub, Pietro Perona, Caltech-256 object category dataset, 2007.
[53] Oren Boiman, Eli Shechtman, Michal Irani, Defense of nearest-neighbor based image classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[54] Prateek Jain, Brian Kulis, Kristen Grauman, Fast image search for learned metrics, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[55] Jan C van Gemert, Jan-Mark Geusebroek, Cor J Veenman, Arnold WM Smeulders, Kernel codebooks for scene categorization, in: Proceedings of European Conference on Computer Vision, Springer, 2008, pp. 696–709.
[56] Yong Xu, Yuhui Quan, Zhuming Zhang, Hui Ji, C. Fermuller, Morimichi Nishigaki, Daniel Dementhon, Contour-based recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3402–3409.
[57] Y-L Boureau, Francis Bach, Yann LeCun, Jean Ponce, Learning mid-level features for recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2559–2566.
[58] Shenghua Gao, Ivor Waihung Tsang, Liang-Tien Chia, Peilin Zhao, Local features are not lonely–laplacian sparse coding for image classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3555–3561.
[59] D. Bertsekas, J. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, 1989.
[60] Stephen J Wright, Robert D Nowak, Mário AT Figueiredo, Sparse reconstruction by separable approximation, IEEE Trans. Signal Process. 57 (7) (2009) 2479–2493.