# Dictionary Reduction: Automatic Compact Dictionary Learning for Classification

Yang Song(✉), Zhifei Zhang, Liu Liu, Alireza Rahimpour, and Hairong Qi

Department of Electrical Engineering and Computer Science,
University of Tennessee, Knoxville, TN 37996, USA
ysong18@vols.utk.edu

**Abstract.** A complete and discriminative dictionary can achieve superior performance. However, it also consumes extra processing time and memory, especially for large datasets. Most existing compact dictionary learning methods need to set the dictionary size manually, therefore an appropriate dictionary size is usually obtained in an exhaustive search manner. How to automatically learn a compact dictionary with high fidelity is still an open challenge. We propose an automatic compact dictionary learning (ACDL) method which can guarantee a more compact and discriminative dictionary while at the same time maintaining the state-of-the-art classification performance. We incorporate two innovative components in the formulation of the dictionary learning algorithm. First, an indicator function is introduced that automatically removes highly correlated dictionary atoms with weak discrimination capacity. Second, two additional constraints, namely, the sum-to-one and the non-negative constraints are imposed on the sparse coefficients. On one hand, this achieves the same functionality as the $L_2$-normalization on the raw data to maintain a stable sparsity threshold. On the other hand, this effectively preserves the geometric structure of the raw data which would be otherwise destroyed by the $L_2$-normalization. Extensive evaluations have shown that the preservation of geometric structure of the raw data plays an important role in achieving high classification performance with smallest dictionary size. Experimental results conducted on four recognition problems demonstrate the proposed ACDL can achieve competitive classification performance using a drastically reduced dictionary (https://github.com/susanqq/ACDL.git).

## 1 Introduction

Given a set of measurements $\{\mathbf{y}_i\}_{i=1}^n$, dictionary learning (DL) is designed to learn an overcomplete dictionary $\{\mathbf{d}_i\}_{i=1}^d$, which is utilized to represent the measurement in a sparse manner. Most DL algorithms fall into one of the two categories: unsupervised learning and supervised learning. Many unsupervised DL approaches [1–5] aim to minimize the reconstruction error between observations and their sparse representation, making them suitable to solve problems like image denosing, image decoding and inpainting.


S.-H. Lai et al. (Eds.): ACCV 2016, Part I, LNCS 10111, pp. 305–320, 2017.
DOI: 10.1007/978-3-319-54181-5_20

Driven by the classification task, supervised DL approaches [6–13] aim to learn a dictionary as discriminative as possible by exploiting the label information. In the big data era, such a large and overcomplete dictionary brings challenges to both processing time and storage. Obviously, a large dictionary requires large memory, and it is more time-consuming in solving sparse coefficient. To reduce the dictionary size, compact dictionary learning techniques have been studied to learn a dictionary with less redundancy and high distinguishability [8,9,14–19]. DLSI [6] learns each class-wise dictionary with less coherence to ensure compactness. FDDL [9] simultaneously learns the discriminative class-wise dictionary and sparse coefficient which satisfies the Fisher criterion. DL-COPAR [16] fixed this problem by separating the dictionary into the common part and class-specific part via a predefined threshold. However, it is difficult to find an appropriate threshold that balances these two parts. ITDL [19] enforces the incoherence of selected dictionary atoms by maximizing the mutual information measurement on the dictionary. All of above methods need the user to specify the size of dictionary. Therefore, given a dataset, an appropriate dictionary size could only be obtained by exhaustively experimenting with different dictionary sizes, which is a tedious procedure for different applications and is often not desirable. LCKSVD [8] represents state-of-the-art DL and has shown superior performance in various computer vision related applications, but it does not have a mechanism to automatically determine the dictionary size. SADL [20] adaptively learns a compact codebook by involving the row sparsity, but it is an unsupervised method which is not for classification purpose. LDL [18] is also an automatic compact DL algorithm, where a latent matrix is used to control the redundancy, but it is a class-wise learning algorithm which cannot guarantee the most discriminative and compact dictionary. Another drawback of existing works is that they require $L_2$-normalization on the raw data to facilitate the determination of the sparsity threshold. However, this normalization also destroys the geometric structure of the raw data that should be utilized as an important and discriminate feature.

In this paper, we propose an automatic compact dictionary learning (ACDL) method globally. Inspired by DKSVD [7], a classification error term is also introduced to learn a linear classifier jointly with the dictionary. The contribution of ACDL is two-fold. First, an indicator function is designed to automatically remove highly correlated dictionary atoms with weak discrimination capacity. By alternatively updating the dictionary and classifier, the indicator function automatically identifies those common and redundant atoms, and an appropriate dictionary is obtained until the function output is stable. Second, instead of using the common $L_2$-normalization to maintain a stable sparsity threshold which, on the other hand, unavoidably destroys the geometric structure of the raw data set, ACDL introduces two constraints, namely, the sum-to-one (S2O) and the non-negative (NN) constraints, on the sparse coefficients. These two constraints achieve the same effect as $L_2$-normalization in terms of maintaining a stable sparsity threshold. However, they also effectively preserve the structure of the raw data in the original space. We show through extensive experiments

as well as the toy example (Sect. 2) that the geometric structure of raw data is essential in achieving high classification performance.

The rest of the paper is organized as follows: Sect. 2 overviews representative works and compares them with the propose method using a toy example. Sect. 3 elaborates on the proposed automatic compact dictionary learning (ACDL). Experimental results are shown in Sect. 4. Section 5 concludes the paper.

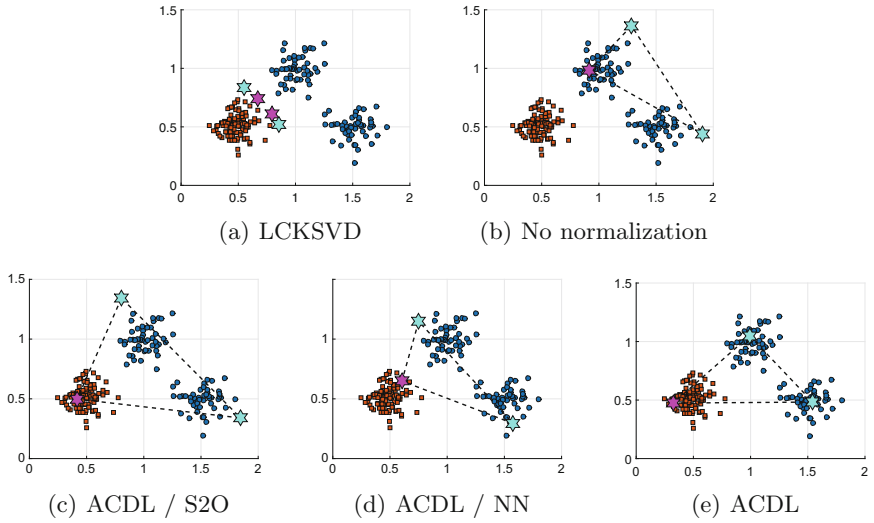
## 2 Motivation

Most state-of-the-art DL works designed for classification, e.g., SDL [17], DL-COPAR [16], FDDL [9] and LCKSVD [8], will first normalize (i.e., $L_2$ normalization) the dataset and then learn a dictionary of certain size that is manually set in ahead. However, normalization will destroy the geometric structure of the data and may mix the data together that leads misclassification. For example, two 2-D points of different classes locate at $(0, 0)$ and $(2, 0)$ will overlap at $(1, 0)$ after $L_2$-normalization. In addition, manual setting of the dictionary size is difficult to achieve the optimal because the latent optimal dictionary size will vary with applications. A common way to get an appropriate size is exhaustive searching that is time-consuming.

To preserve the geometric structure of the raw dataset, we propose the automatic compact dictionary learning (ACDL) method that does not adopt any normalization methods. Therefore, the data from different classes will not overlap by mistake. The non-negative and sum-to-one constraints are incorporated to force the learned dictionary items to represent the skeleton of geometric structure. A toy example is shown in Fig. 1 to illustrate the effect of non-negative and sum-to-one constraints.

Two classes of data samples are constructed based on the Gaussian distribution with the first class using a uni-modal distribution and the second class using a bi-modal distribution. The purpose of the experiment is to see the effect of normalization and the two constraints. Figure 1 shows the data set and learned dictionary, where red squares and blue circles indicate samples from two classes. The hexagrams in magenta and cyan, denote the learned dictionary atoms belonging to the red and blue class, respectively. It mainly demonstrates two advantages of the proposed ACDL: preservation of the geometric structure and automatic determination of dictionary size.

Figure 1(a) has illustrated the importance of geometric structure because the normalization used in most algorithms (e.g., LCKSVD) will overlap the two clusters on the diagonal direction of the coordinate system. Therefore, the dictionary learned by LCKSVD cannot well distinguish the samples located along the diagonal direction. Figure 1(b) shows the learned dictionary without normalization. The dictionary atoms tends to approach the clusters. However, the atoms cannot well represent the geometric structure, and the location of atoms are unstable in each learning process. Compared to Fig. 1(c) and (d), ACDL (Fig. 1(e)) well preserves the geometric structure through the non-negative and sum-to-one constraints and automatically learns a dictionary with three atoms.



(a) LCKSVD (b) No normalization

(c) ACDL / S2O (d) ACDL / NN (e) ACDL

**Fig. 1.** A toy example to illustrate the learning results of different conditions. Red squares and blue circles are samples from two classes. Hexagrams denote the learned dictionary atoms, and magenta and cyan have the same label with red and blue, respectively. The $L_2$-normalization is required and the dictionary size needs to be set manually in all methods except the ACDL. (a) The result of LCKSVD, and the dictionary size is set to be 3. (b) Illustration of LCKSVD without $L_2$-normalization. (c) and (d) The results of ACDL without sum-to-one (S2O) and non-negative (NN) constraints, respectively. (e) The result of the proposed ACDL method, which incorporates the S2O and NN, relaxing the requirement of normalization and preserving the geometric structure of the data and dictionary items. (Color figure online)

To summarize, the key advantage of ACDL over the other methods is that it can automatically determine the dictionary size according the dataset without human intervention. In addition, the geometric structure of the raw data set is preserved to potentially improve classification performance.

## 3 Approach

In this section, we elaborate on the proposed ACDL approach for the automatic determination of the size of a compact dictionary while at the same time maintaining competitive performance as compared to the state-of-the-art. Assume a set of observations $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n] \in \mathbb{R}^{m \times n}$ that can be represented by linear composition of a few atoms in a dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_d] \in \mathbb{R}^{m \times d}$, with the corresponding weight of each atom (or sparse coefficients) being $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$. Note that $n > d$, and each column of $\mathbf{X}$ is sparse. Equation 1 shows the sparse representation.

$$\mathbf{Y} \approx \mathbf{DX} \tag{1}$$

For classification purpose, each atom in the dictionary $\mathbf{D}$ is learned to distinguishably represent certain class. Therefore, the label of an observation can be decided based on the corresponding sparse coefficients. For simplicity, the dictionary atom with the largest sparse coefficient indicates that the observation belongs to the same class as the atom. Suppose the label of observation is $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_n] \in \mathbb{R}^{c \times n}$ (assume there are $c$ classes), and each $\mathbf{g}_i$ is a column vector whose elements are zero except for the one having the same row index as the class index. For example, if $c = 2$, then $\mathbf{g}_1 = [1, 0]^T$ means the label of the first observation is 1. By the same token, a linear classifier can be written as $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d] \in \mathbb{R}^{c \times d}$, where for a $\mathbf{w}_i = [w_1, w_2, \cdots, w_c]^T$, certain element $w_j$ indicates the probability that $\mathbf{d}_i$ belongs to the $j$th class. Therefore, class information of the observations could be represented sparsely as in Eq. 2.

$$\mathbf{G} \approx \mathbf{W}\mathbf{X} \tag{2}$$

Equations 1 and 2 consider reconstruction fidelity and classification error, receptively. By combining these two terms, a global dictionary learning algorithm can be formulated.

### 3.1 Automatic Compact Dictionary Learning

A preliminary objective function to globally learn a dictionary and the corresponding classifier can be written in Eq. 3,

$$\begin{aligned} \arg\min_{\mathbf{D},\mathbf{W}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \gamma\|\mathbf{G} - \mathbf{W}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_1 \\ s.t.\ \mathbf{X} \succeq 0,\ \sum \mathbf{x}_i = 1 \end{aligned} \tag{3}$$

where the first term denotes the reconstruction error, the second term represents the classification error, and the last term indicates the sparsity constraint. Different from existing compact dictionary learning methods, in ACDL, we constrain the sparse coefficients $\mathbf{X}$ under two additional conditions, i.e., the non-negative condition and the column-wise sum-to-one condition in order to preserve the geometric structure of the dataset. Specifically, these two constraints together force the learned dictionary to form a polyhedron upon the dataset, whose vertices (dictionary atoms) have the tendency to be placed in the middle of the data samples with dense distributions. Thus, the polyhedron more explicitly reflects the distribution of the dataset in its raw spatial scale.

Given a set of observations and their labels, as well as predefined dictionary size, minimizing Eq. 3 could obtain a dictionary with appropriate size along with the classifier. However, we aim to learn a dictionary with its size automatically determined. Ideally, the dictionary $\mathbf{D}$ is initialized to be the observation matrix $\mathbf{Y}$, and then $\mathbf{D}$ is updated and reduced automatically according to the dataset given. From this perspective, both the size of the dictionary and the atom need to be updated during the learning procedure. To achieve this goal, we rewrite

Eqs. 3 and 4 by adding an operator $\mathcal{F}(\mathbf{D}, \mathbf{W})$ which is a function of $\mathbf{D}$ and $\mathbf{W}$ to indicate the atoms that need to be removed from the dictionary.

$$\begin{aligned} \arg\min_{\mathbf{D},\mathbf{W}} \|\mathbf{Y} - \mathbf{D}\mathcal{F}(\mathbf{D}, \mathbf{W})\mathbf{X}\|_F^2 + \gamma\|\mathbf{G} - \mathbf{W}\mathcal{F}(\mathbf{D}, \mathbf{W})\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_1 \\ s.t.\ \mathbf{X} \succeq 0,\ \sum \mathbf{x}_i = 1 \end{aligned} \tag{4}$$

Specifically, $\mathcal{F}(\mathbf{D}, \mathbf{W})$ yields a diagonal matrix of dimension $d \times d$. The elements on the diagonal are either 0 or 1 with 0 indicating the corresponding atom in $\mathbf{D}$ should be removed. Assume $\mathcal{D}_i = \mathbf{D}\mathcal{F}(\mathbf{D}, \mathbf{W})diag(\mathcal{B}(\mathbf{W})_{i*})$, where $i$ is the class index, $\mathbf{W}_{i*}$ denotes the $i$th row of $\mathbf{W}$, $\mathcal{B}$ is a column-wised binary non maximum suppression operator, for example, $\mathbf{W} = [0.8, 0.2, 0.1; 0.7, 0.2, 0.1]^T$, $\mathcal{B}(\mathbf{W}) = [1, 0, 0; 1, 0, 0]^T$. $diag(\mathbf{W}_{i*})$ forms a diagonal matrix indicating the probability that each atom of $\mathbf{D}$ belongs to the $i$th class. That is, $\mathcal{D}_i$ is the same matrix as $\mathbf{D}$ but the atoms that are indicated as removed or do not belong to the $i$th class are set to zero. By the same token, $\mathcal{W}_i = \mathbf{W}\mathcal{F}(\mathbf{D}, \mathbf{W})diag(\mathbf{W}_{i*})$. Based on $\mathcal{D}_i$'s and $\mathcal{W}_i$'s, $\mathcal{F}(\mathbf{D}, \mathbf{W})$ is updated by Eq. 5.

$$\mathcal{F}(\mathbf{D}, \mathbf{W})_{jj} = \begin{cases} \mathcal{F}(\mathbf{D}, \mathbf{W})_{jj},\ \mathbf{d}_j = 0, \mathbf{d}_j \in \mathcal{D}_i \\ 0, \quad \begin{array}{l} \dfrac{\min\limits_k ||\mathbf{d}_j - \mathbf{d}_k||_2}{\max\limits_{p,q} ||\mathbf{d}_p - \mathbf{d}_q||_2} < \epsilon \\ \text{and } \mathbf{w}_j \ln \mathbf{w}_j < \mathbf{w}_k \ln \mathbf{w}_k \\ \mathbf{w}_j, \mathbf{w}_k \in \mathcal{W}_i \\ \mathbf{d}_* \in \mathcal{D}_i, \mathbf{d}_* \neq 0 \\ or\ \mathcal{F}(\mathbf{D}, \mathbf{W})_{jj} = 0 \end{array} \\ 1, \quad \text{otherwise} \end{cases} \tag{5}$$

where $\mathcal{F}(\mathbf{D}, \mathbf{W})_{jj}$ denotes the $j$th element on the diagonal of $\mathcal{F}(\mathbf{D}, \mathbf{W})$. $\epsilon$ is a threshold from 0 to 1. Generally speaking, if a dictionary atom is close to another from the same class as measured in Euclidian distance or other distance metrics (we use Euclidian distance in this paper) and less discriminative (the corresponding column in $\mathbf{W}$ has lower entropy), it will be removed (or set to zero) in the subsequent iterations. $\mathcal{F}(\mathbf{D}, \mathbf{W})$, $\mathbf{D}$ and $\mathbf{W}$ are updated alternatively. Usually, $\mathbf{D}$ and $\mathbf{W}$ are initialized to be $\mathbf{Y}$ and $\mathbf{G}$, respectively. The initial $\mathbf{X}$ and $\mathcal{F}(\mathbf{D}, \mathbf{W})$ are set to be an identity matrix. Then, iterating Eqs. 4 and 5 will result in a compact dictionary with its size (i.e., number of non-zero columns) automatically and optimally determined. Finally, the dictionary and classifier are generated using the non-zero columns of $\mathbf{D}\mathcal{F}(\mathbf{D}, \mathbf{W})$ and $\mathbf{W}\mathcal{F}(\mathbf{D}, \mathbf{W})$, respectively.

We will discuss more on solving Eq. 4. Since $\mathcal{F}(\mathbf{D}, \mathbf{W})$ is updated after $\mathbf{D}$ and $\mathbf{W}$, it can be considered as a constant when $\mathbf{D}$ and $\mathbf{W}$ are being updated. Because $\mathcal{F}(\mathbf{D}, \mathbf{W})$ forces certain atoms in $\mathbf{D}$ and $\mathbf{W}$ to be zero, the exact dictionary and classifier that need to be updated are $\widehat{\mathbf{D}} = \{\mathbf{D}\mathcal{F}(\mathbf{D}, \mathbf{W})_{*j} | \mathcal{F}(\mathbf{D}, \mathbf{W})_{*j} \neq \mathbf{0}\}$ and $\widehat{\mathbf{W}} = \{\mathbf{W}\mathcal{F}(\mathbf{D}, \mathbf{W})_{*j} | \mathcal{F}(\mathbf{D}, \mathbf{W})_{*j} \neq \mathbf{0}\}$, respectively, where $\mathcal{F}(\mathbf{D}, \mathbf{W})_{*j}$ denotes

the $j$th column of $\mathcal{F}(\mathbf{D}, \mathbf{W})$. Correspondingly, the equivalent sparse coefficients $\widehat{\mathbf{X}} = \{\mathcal{F}(\mathbf{D}, \mathbf{W})_{j*}\mathbf{X} | \mathcal{F}(\mathbf{D}, \mathbf{W})_{j*} \neq \mathbf{0}\}$. Thus, Eq. 4 can be rewritten as Eq. 6.

$$\begin{aligned} \arg\min_{\widehat{\mathbf{D}}, \widehat{\mathbf{W}}} \; & \|\mathbf{Y} - \widehat{\mathbf{D}}\widehat{\mathbf{X}}\|_F^2 + \gamma\|\mathbf{G} - \widehat{\mathbf{W}}\widehat{\mathbf{X}}\|_F^2 + \lambda\|\widehat{\mathbf{X}}\|_1 \\ s.t. \; & \widehat{\mathbf{X}} \succeq 0, \; \sum \widehat{\mathbf{x}}_i = 1 \end{aligned} \tag{6}$$

Combining the first two terms and augmenting the sum-to-one constraint, Eq. 6 is simplified as Eq. 7.

$$\begin{aligned} \arg\min_{\widehat{\mathbf{D}}, \widehat{\mathbf{W}}} \; & \left\| \begin{pmatrix} \mathbf{Y} \\ \sqrt{\gamma}\mathbf{G} \\ \delta\mathbf{1}_{1\times n} \end{pmatrix} - \begin{pmatrix} \widehat{\mathbf{D}} \\ \sqrt{\gamma}\widehat{\mathbf{W}} \\ \delta\mathbf{1}_{1\times d} \end{pmatrix} \widehat{\mathbf{X}} \right\|_F^2 + \lambda\|\widehat{\mathbf{X}}\|_1 \\ s.t. \; & \widehat{\mathbf{X}} \succeq 0 \end{aligned} \tag{7}$$

Assume $\widetilde{\mathbf{Y}} = \begin{pmatrix} \mathbf{Y} \\ \sqrt{\gamma}\mathbf{G} \\ \delta\mathbf{1}_{1\times n} \end{pmatrix}$ and $\widetilde{\mathbf{D}} = \begin{pmatrix} \widehat{\mathbf{D}} \\ \sqrt{\gamma}\widehat{\mathbf{W}} \\ \delta\mathbf{1}_{1\times d} \end{pmatrix}$, where $\mathbf{1}$ denotes the matrix of all 1's and $\delta$ balances the effect of the sum-to-one constraint. Equation 7 is further simplified to Eq. 8.

$$\begin{aligned} \arg\min_{\widetilde{\mathbf{D}}, \widehat{\mathbf{X}}} \; & \|\widetilde{\mathbf{Y}} - \widetilde{\mathbf{D}}\widehat{\mathbf{X}}\|_F^2 + \lambda\|\widehat{\mathbf{X}}\|_1 \\ s.t. \; & \widehat{\mathbf{X}} \succeq 0 \end{aligned} \tag{8}$$

Given $\widetilde{\mathbf{Y}}$ and applying the proximal gradient descent [21], $\widetilde{\mathbf{D}}$ and $\widehat{\mathbf{X}}$ are updated alternatively. Then, new $\widehat{\mathbf{D}}$ and $\widehat{\mathbf{W}}$ can be obtained. Let $\mathbf{D} = \widehat{\mathbf{D}}$, $\mathbf{W} = \widehat{\mathbf{W}}$ and $\mathbf{X} = \widehat{\mathbf{X}}$, then $\mathcal{F}(\mathbf{D}, \mathbf{W})$ is updated through Eq. 5. Note that the size of $\mathbf{D}$ and $\mathbf{W}$ may be smaller than their original size now. Iterating the above procedure, a compacted dictionary, as well as the corresponding classifier, is learned in an automatic and joint manner. Algorithm 1 provides the pseudo-code for the propose ACDL method.

### 3.2 Proximal Gradient Descent

The proximal mapping or proximal operator [21] of a convex function $g(x)$ is defined in Eq. 9.

$$\mathbf{prox}_g(x) = \arg\min_u \left( g(u) + \frac{1}{2}||u - x||_2^2 \right) \tag{9}$$

If $g(x) = \lambda||x||_1$, then $\mathbf{prox}_g(x)$ is the shrinkage or soft threshold operation as shown in Eq. 10.

$$\mathbf{prox}_{\lambda,g}(x)_i = \begin{cases} x_i - \lambda, & x_i \geq \lambda \\ x_i + \lambda, & x_i \leq -\lambda \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

**Algorithm 1.** Automatic Compact Dictionary Learning (ACDL)

**Input:** $\mathbf{Y}_{m\times n}$, $\mathbf{G}_{c\times n}$, $\gamma$, $\lambda$, $\delta$, $\epsilon$
**Output:** $\mathbf{D}$ and $\mathbf{W}$
**Initialization:** $\mathbf{D}=\mathbf{Y}$, $\mathbf{W}=\mathbf{G}$, $\mathbf{X}=\mathbf{I}_{n\times n}$, $\mathcal{F}(\mathbf{D},\mathbf{W})=\mathbf{I}_{n\times n}$, $stop=$ false
**while** not $stop$ **do**
  $\widehat{\mathbf{D}}=\{\mathbf{D}\mathcal{F}(\mathbf{D},\mathbf{W})_{*j}|\mathcal{F}(\mathbf{D},\mathbf{W})_{*j}\neq\mathbf{0}\}$
  $\widehat{\mathbf{W}}=\{\mathbf{W}\mathcal{F}(\mathbf{D},\mathbf{W})_{*j}|\mathcal{F}(\mathbf{D},\mathbf{W})_{*j}\neq\mathbf{0}\}$
  $\widehat{\mathbf{X}}=\{\mathcal{F}(\mathbf{D},\mathbf{W})_{j*}\mathbf{X}|\mathcal{F}(\mathbf{D},\mathbf{W})_{j*}\neq\mathbf{0}\}$
  $\widetilde{\mathbf{Y}}=\begin{pmatrix}\mathbf{Y}\\ \sqrt{\gamma}\mathbf{G}\end{pmatrix}$, $\widetilde{\mathbf{D}}=\begin{pmatrix}\widehat{\mathbf{D}}\\ \sqrt{\gamma}\widehat{\mathbf{W}}\end{pmatrix}$
  Solve Eq. 8 using proximal gradient descent (Eqs. 12 and 13)
  Get updated $\widehat{\mathbf{D}}$, $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{X}}$
  **if** $size(\mathbf{D})=size(\widehat{\mathbf{D}})$ **then**
    $stop=$ true
  **end if**
  $\mathbf{D}=\widehat{\mathbf{D}}$, $\mathbf{W}=\widehat{\mathbf{W}}$, $\mathbf{X}=\widehat{\mathbf{X}}$
  Compute $\mathcal{F}(\mathbf{D},\mathbf{W})$ through Eq. 5
**end while**

$\widetilde{\mathbf{D}}$ and $\widehat{\mathbf{X}}$ are updated alternatively by minimizing one while keeping the other fixed. The update of $\widetilde{\mathbf{D}}$ can adopt the basic gradient descent method. Since the function with respect to $\widehat{\mathbf{X}}$ is not continuously differentiable because of the $L_1$-norm, the proximal mapping is employed to update $\widehat{\mathbf{X}}$. As illustrated in Eq. 11, the two terms in Eq. 8 correspond to the two functions $f(\widehat{\mathbf{X}},\widetilde{\mathbf{D}})$ and $g(\widehat{\mathbf{X}})$.

$$\arg\min_{\widetilde{\mathbf{D}},\widehat{\mathbf{X}}} \underbrace{\|\widetilde{\mathbf{Y}}-\widetilde{\mathbf{D}}\widehat{\mathbf{X}}\|_F^2}_{f(\widehat{\mathbf{X}},\widetilde{\mathbf{D}})}+\lambda\underbrace{\|\widehat{\mathbf{X}}\|_1}_{g(\widehat{\mathbf{X}})} \tag{11}$$

Then, updating $\widetilde{\mathbf{D}}$ and $\widehat{\mathbf{X}}$ can be expressed in Eq. 12.

$$\widehat{\mathbf{X}}^{k+1}:=\max\left\{\mathbf{prox}_{\lambda,\eta_1^k,g}\left(\widehat{\mathbf{X}}^k-\eta_1^k\nabla f(\widehat{\mathbf{X}}^k,\widetilde{\mathbf{D}})\right),\ 0\right\} \tag{12}$$

$$\widetilde{\mathbf{D}}^{k+1}:=\widetilde{\mathbf{D}}^k-\eta_2^k\nabla f(\widehat{\mathbf{X}},\widetilde{\mathbf{D}}^k) \tag{13}$$

where

$$\nabla_{\widehat{\mathbf{X}}}f(\widehat{\mathbf{X}},\widetilde{\mathbf{D}})=\widetilde{\mathbf{D}}^T(\widetilde{\mathbf{D}}\widehat{\mathbf{X}}-\widetilde{\mathbf{Y}}) \tag{14}$$

$$\nabla_{\widetilde{\mathbf{D}}}f(\widehat{\mathbf{X}},\widetilde{\mathbf{D}})=(\widetilde{\mathbf{D}}\widehat{\mathbf{X}}-\widetilde{\mathbf{Y}})\widehat{\mathbf{X}}^T \tag{15}$$

Iterating Eqs. 12 and 13 until certain criterion is satisfied, the optimal $\widetilde{\mathbf{D}}$ and $\widehat{\mathbf{X}}$ can be obtained. Specifically, the optimal step size are calculated through ($\beta=1$ [22]) as shown in Eqs. 16 and 17, where $\|\cdot\|_2$ denotes the spectral norm.

$$\eta_1^k=\frac{1}{\beta\|\widetilde{\mathbf{D}}^{k^T}\widetilde{\mathbf{D}}^k\|_2} \tag{16}$$

$$\eta_2^k=\frac{1}{\beta\|\widehat{\mathbf{X}}^k\widehat{\mathbf{X}}^{k^T}\|_2} \tag{17}$$

### 3.3 Classification

Given the learned dictionary $\mathbf{D}$ and classifier $\mathbf{W}$, new observations $\mathbf{Y}$ can be sparsely represented by $\mathbf{X}$ via Eq. 18, which can be solved by combining the sum-to-one constraint into the objective function like Eq. 7 and applying the proximal gradient descent.

$$\begin{aligned} &\arg\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_1 \\ &s.t.\ \mathbf{X} \succeq 0,\ \sum \mathbf{x}_i = 1 \end{aligned} \tag{18}$$

Then the predicted labels $\mathbf{L}$ of the observations $\mathbf{Y}$ can be obtained through Eq. 19,

$$\mathbf{L} = [\mathbf{l}_1, \mathbf{l}_2, \cdots, \mathbf{l}_n] = \mathbf{W}\mathbf{X} \tag{19}$$

where $\mathbf{L}$ has the same number of columns as $\mathbf{Y}$ (assume $n$), and each column of $\mathbf{L}$ is the label vector of the corresponding observation in $\mathbf{Y}$. Note that $\mathbf{W}$ is $L_2$-normalized as the following, assuming the dictionary size is $d$.

$$\mathbf{W} = \left[\frac{\mathbf{w}_1}{\|\mathbf{w}_1\|_2}, \frac{\mathbf{w}_2}{\|\mathbf{w}_2\|_2}, \cdots, \frac{\mathbf{w}_d}{\|\mathbf{w}_d\|_2}\right] \tag{20}$$

Finally, the label of certain observation is the index corresponding to the largest element in its label vector as expressed in Eq. 21.

$$\begin{aligned} &label(\mathbf{y}_i) = \arg\max_j \{l_j\} \\ &s.t.\ l_j \in \mathbf{l}_i \end{aligned} \tag{21}$$

where $\mathbf{y}_i$ denotes the $i$th column of $\mathbf{Y}$, and $\mathbf{l}_i = [l_1, l_2, \cdots]^T$ is the $i$th column of $\mathbf{L}$. $label(\cdot)$ yields an integer that indicates the class label.

## 4 Experimental Evaluation

We evaluate the proposed ACDL algorithm on four classification tasks, including multi-view classification using the Berkeley multiview wireless (BMW) dataset [23] (Sect. 4.1), scene recognition using the 15 scene categories dataset [24] (Sect. 4.2), object recognition using Caltech101 [25] (Sect. 4.3), and handwritten digit recognition using MNIST (Sect. 4.4). We compare ACDL with SDL [17], FDDL [9], DL-COPAR [16], KSVD [3], DKSVD [7], LCKSVD [8], and SRC [12] in terms of classification accuracy and dictionary size. In addition, some state-of-the-art approaches, such as spatial pyramid matching [24], sparse PCA [26], CNN [27,28], and LeNet [29], are cited to compare to our approach in terms of classification accuracy. Finally, Sect. 4.5 discusses the parameter setting.

### 4.1 The Berkeley Multiview Wireless Dataset

The Berkeley multiview wireless (BMW) [23] is a dataset of 20 landmark buildings on the Berkeley campus under multiple views. For each building, wide-baseline images were captured from 16 different vantage points. At each vantage point, 5 short-baseline images were taken by five cameras simultaneously. And thereby there are 80 images per category. All images are $640 \times 480$ RGB color images. We follow the common setting, the training dataset are images captured at the even vantage by camera #2 (320 images), and the rest are the testing dataset. The SURF features are postprocessed by Sparse PCA [26] in this experiment to filter noisy and background points.

In this experiment, the parameters are set as $\lambda = 0.1$, $\gamma = 20$, $\delta = 2$, $\epsilon = 0.6$ (see details in Sect. 4.5). The learned dictionary size is 109, and the average accuracy is 90.2%. The confusion matrix is shown in Fig. 2(b). Similarly, we compare our results with FDDL [9], SDL [17], DL-COPAR [16], LCKSVD [8] and a baseline method Naikal [26] as listed in Table 1. To illustrate the effectiveness of dictionary reduction, the dictionary-based algorithms except for ACDL are performed with two dictionary sizes, respectively. One is around 109 that is learned automatically from ACDL, and the other is 200. Our approach achieves a competitive performance while using a much smaller dictionary. This, again, demonstrates that ACDL can automatically learn a compact dictionary with appropriate size without any prior knowledge. In addition, the learned dictionary is discriminative enough to yield state-of-the-art classification performance. At the end of the table, the results of ACDL without sum-to-one (S2O) and non-negative (NN) constraints are reported to illustrate the significance of S2O and NN constraints.

**Table 1.** Classification results of different algorithms on the BMW dataset (the results marked by * are directly cited from corresponding papers)

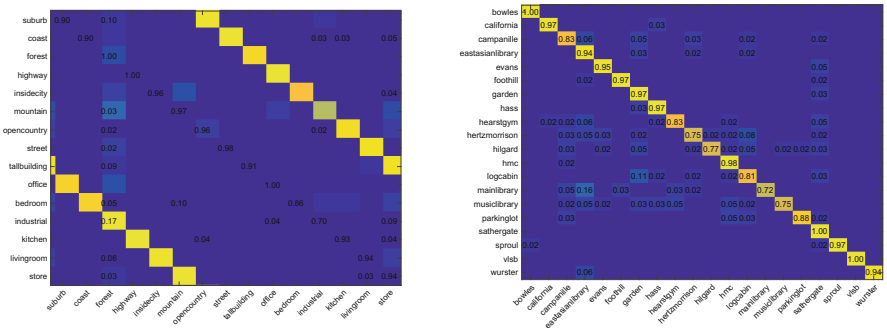
| Method | Accuracy(%) | Dictionary size |
|---|---|---|
| Naikal [26] | 84.5* | – |
| FDDL [9] | 74.0 | 320 |
| SDL [17] | **90.8** | 200 |
| SDL [17] | 90.1 | 109 |
| DL-COPAR [16] | 69.5 | 200 |
| DL-COPAR [16] | 73.8 | 100 |
| LCKSVD1 [8] | 89.9 | 200 |
| LCKSVD1 [8] | 89.8 | 109 |
| LCKSVD2 [8] | 90.4 | 200 |
| LCKSVD2 [8] | 89.7 | 109 |
| **ACDL** | **90.2** | **109** |
| ACDL/S2O | 81 | 127 |
| ACDL/NN | 76 | 80 |

### 4.2 The Fifteen Scene Categories Dataset

This dataset [24] contains 15 outdoor and indoor scene categories. The average image size is about 250×300 pixels, and each category has 200 to 400 images. The spatial pyramid features algorithm [24] was the first applied on this dataset and achieved an impressive classification accuracy of 81.4%. Following the same experimental setup in the spatial pyramid features [24] and LCKSVD [8], we randomly select 100 images per category as training data and the rest are testing data.

In ACDL, we initialize the dictionary with the whole training set (1500 atoms), and the parameters are set as $\lambda = 0.1$, $\gamma = 5$, $\delta = 2$, $\epsilon = 0.6$ (see details in Sect. 4.5). The learned dictionary size is drastically reduced to 122, with an average classification accuracy as 93.6%. The confusion matrix is shown in Fig. 2(a). We also compare our results with FDDL [9], DL-COPAR [16], LCKSVD [8], DKSVD [7], SDL [17] and some common baseline works as listed in Table 2. The results marked with * are the best results directly cited from corresponding papers. Among the baseline works, Zhou [30] obtained the highest accuracy of 91.6%.

For comparison purpose, the dictionary size of SDL and DKSVD are set to be 450 that is the same as used in LCKSVD. Under this setting, LCKSVD performs better than SDL, but SDL improves its performance to a great extent when we set the dictionary size according to the automatically learned size from ACDL. However, if there is no prior knowledge, it will be difficult to manually set an appropriate dictionary size. This experiment has demonstrated that ACDL can automatically learn a much more compact dictionary with appropriate size, while at the same time achieving higher recognition accuracy than state-of-the-art works.




(a) 15 scene (b) BMW

**Fig. 2.** (a) Confusion matrix using the dictionary (122 atoms) learned from ACDL on the 15 scene categories. (b) Confusion matrix using the dictionary (109 atoms) learned from ACDL on the BMW dataset.

**Table 2.** Classification results of different algorithms on 15 scene categories (the results marked by * are directly cited from corresponding papers)

| Method | Accuracy(%) | Dictionary size |
|---|---|---|
| Lazebnik [24] | 81.4* | – |
| Gao [31] | 89.7* | – |
| Zhou [30] | **91.6*** | – |
| FDDL [9] | 73.5 | 317 |
| FDDL [9] | 54.1 | 170 |
| DL-COPAR [16] | 85.4* | 1024* |
| LCKSVD1 [8] | 90.4* | 450* |
| LCKSVD2 [8] | 92.9* | 450* |
| DKSVD [7] | 89.1 | 450 |
| SDL [17] | 84.7 | 450 |
| SDL [17] | **93.0** | 122 |
| **ACDL** | **93.6** | **122** |

### 4.3 The Caltech101 Dataset

The Caltech101 dataset [25] consists of 9144 images from 101 objects such as animals, cars, planes, etc. Each category has 31 to 800 images with significant shape variability. Following the common setting, the spatial pyramid features are used as the input and the feature dimension is 3000. We randomly choose 30% (2743 samples) to build the training dataset and the rest forms the testing dataset.

In this experiment, the parameters are set as $\lambda = 0.1$, $\gamma = 10$, $\delta = 2$, $\epsilon = 0.6$ (see details in Sect. 4.5). The dictionary size learned from ACDL is 161 and the average accuracy is 76.3%. We compare our algorithm with SRC [12], SDL [17], KSVD [3], DKSVD [7], LCKSVD [8], DL-COPAR [16] and some deep learning approaches as shown in Table 3. DeCAF [27] using convolutional neural network achieves the highest accuracy, and DL-COPAR performs the best among all dictionary-based approaches, but with a large dictionary size. Most dictionary-based methods achieve an accuracy around 70% or higher with a very large dictionary size, while the proposed ACDL can reach competitive performance with an extremely small dictionary size (more than 100 times smaller).

### 4.4 The MNIST Dataset

The MNIST dataset contains 70,000 handwritten digit images from 0 to 9. The size of each image is $28 \times 28$ pixels. LeNet [29] is employed to extract features from each image. The LeNet we used consists of two convolution-pooling layers and two fully connected networks, and we use the output of the last fully connected layer as the feature, which is a 10 dimensional vector. We randomly choose 4% (2400 samples) from the original training dataset which has 60000 samples, and the testing dataset has 10000 samples.

**Table 3.** Classification results of different algorithms on the Caltech dataset (the results marked by * are directly cited from corresponding papers)

| Method | Accuracy(%) | Dictionary size |
|---|---|---|
| Lazebnik [24] | 64.6* | – |
| Zeiler [28] | 86.5* | – |
| DeCAF [27] | **86.9*** | – |
| SRC [12] | 70.7* | 3060* |
| SDL [17] | 75.3* | 3060* |
| KSVD [3] | 73.0* | 3060* |
| DKSVD [7] | 73.0* | 3060* |
| LCKSVD1 [8] | 73.4* | 3060* |
| LCKSVD2 [8] | 73.6* | 3060* |
| DL-COPAR [16] | **83.3*** | 2048* |
| **ACDL** | **76.3** | **161** |

In this experiment, the parameters are set as $\lambda = 0.1$, $\gamma = 20$, $\delta = 2$, $\epsilon = 0.4$ (see details in Sect. 4.5). The learned dictionary size from ACDL is 21, and the average classification accuracy is 98.5%. We compare our approach with FDDL [9], SDL [17], LCKSVD [8], DL-COPAR [16] and some deep learning methods as listed in Table 4. Similar to the experiment on the BMW dataset, two dictionary sizes are set for each algorithm, one is similar to the size automatically learned from ACDL and the other is a larger size. Among the dictionary-based methods, ACDL performs best and achieves a compact and discriminative dictionary size with consistent performance as observed from previous experiments.

**Table 4.** Classification results of different algorithms on the MNIST dataset (the results marked by * are directly cited from corresponding papers)

| Method | Accuracy(%) | Dictionary size |
|---|---|---|
| LeNet-4 [29] | 98.9* | – |
| Jarrett [32] | 99.5* | – |
| Ciresan [33] | **99.7*** | – |
| FDDL [9] | 96.2 | 77 |
| SDL [17] | 98.5 | 100 |
| SDL [17] | 94.7 | 20 |
| LCKSVD [8] | 98.2 | 100 |
| LCKSVD [8] | 98.4 | 20 |
| DL-COPAR [16] | **98.5** | 100 |
| DL-COPAR [16] | 98.3 | 20 |
| **ACDL** | **98.5** | **21** |

### 4.5 Parameter Setting

From the above experiments conducted on four different datasets, the parameters need to be set manually are listed in Table 5. The parameter $\delta$ that balances the sum-to-one constraint and $\lambda$ that dominates sparse penalty remain as constant for different datasets. The parameters for reduction threshold $\epsilon$ do not vary a lot. Only the parameter $\gamma$ that controls the classification error has significant change when using different datasets. In practice, we adjust $\gamma$ with step size of 5 within a narrow range, i.e. from 5 to 30. Usually, an appropriate dictionary would be constructed within several trials.

**Table 5.** Parameter settings on different datasets

| Dataset | $\lambda$ | $\gamma$ | $\delta$ | $\epsilon$ |
|---|---|---|---|---|
| 15 scenes | 0.1 | 5 | 2 | 0.6 |
| BMW | 0.1 | 20 | 2 | 0.6 |
| Caltech101 | 0.1 | 10 | 2 | 0.6 |
| MNIST | 0.1 | 20 | 2 | 0.4 |

Specifically, $\gamma$ and $\epsilon$ affect the discrimination capability and the size, respectively, of the final dictionary. Therefore, $\gamma$ reflects the distinguishability of the dataset, and $\epsilon$, in a sense, reflects the geometric structure. For example, if the samples from the same class fall into two clusters that locate far from each other, then $\epsilon$ should be smaller. If the samples from different classes have large overlap, then $\gamma$ needs to pick a larger value.

## 5 Conclusion

This paper proposed an automatic compact dictionary learning (ACDL) method, which learns a more compact and discriminative dictionary as compared with existing works, while maintaining a competitive classification performance. While most literature requires manual intervention to construct a dictionary with appropriate size, ACDL automatically removes common and redundant atoms by introducing the class error and indicator function into the objective function. The former ensures global learning that facilitates the removal of common atoms between classes, and the latter guarantees automatic dictionary reduction. In addition, the geometric structure of the dataset is preserved and utilized during the learning procedure by the non-negative and sum-to-one constraints enforced on the sparse coefficient, which may increase the classification performance. Experimental results demonstrated the effectiveness of the propose ACDL method.

## References


1. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: a strategy employed by V1? Vis. Res. **37**, 3311–3325 (1997)
2. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process. **15**, 3736–3745 (2006)
3. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Sig. Process. **54**, 4311 (2006)
4. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 689–696. ACM (2009)
5. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8. IEEE (2008)
6. Ramirez, I., Sprechmann, P., Sapiro, G.: Classification and clustering via dictionary learning with structured incoherence and shared features. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3501–3508. IEEE (2010)
7. Zhang, Q., Li, B.: Discriminative K-SVD for dictionary learning in face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2691–2698. IEEE (2010)
8. Jiang, Z., Lin, Z., Davis, L.S.: Label consistent K-SVD: learning a discriminative dictionary for recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 2651–2664 (2013)
9. Yang, M., Zhang, L., Feng, X., Zhang, D.: Fisher discrimination dictionary learning for sparse representation. In: IEEE International Conference on Computer Vision (ICCV), pp. 543–550. IEEE (2011)
10. Mairal, J., Ponce, J., Sapiro, G., Zisserman, A., Bach, F.R.: Supervised dictionary learning. In: Advances in Neural Information Processing Systems, pp. 1033–1040 (2009)
11. Mairal, J., Bach, F., Ponce, J.: Task-driven dictionary learning. IEEE Trans. Pattern Anal. Mach. Intell. **34**, 791–804 (2012)
12. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 210–227 (2009)
13. Rahimpour, A., Taalimi, A., Luo, J., Qi, H.: Distributed object recognition in smart camera networks. In: IEEE International Conference on Image Processing, Phoenix, Arizona, USA. IEEE (2016)
14. Lazebnik, S., Raginsky, M.: Supervised learning of quantizer codebooks by information loss minimization. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 1294–1309 (2009)
15. Liu, J., Shah, M.: Learning human actions via information maximization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8. IEEE (2008)
16. Kong, S., Wang, D.: A dictionary learning approach for classification: separating the particularity and the commonality. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7572, pp. 186–199. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33718-5_14
17. Jiang, Z., Zhang, G., Davis, L.S.: Submodular dictionary learning for sparse coding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3418–3425. IEEE (2012)

18. Yang, M., Dai, D., Shen, L., Van Gool, L.: Latent dictionary learning for sparse representation based classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4138–4145. IEEE (2014)
19. Qiu, Q., Patel, V.M., Chellappa, R.: Information-theoretic dictionary learning for image classification. IEEE Trans. Pattern Anal. Mach. Intell. **36**, 2173–2184 (2014)
20. Lu, C., Shi, J., Jia, J.: Scale adaptive dictionary learning. IEEE Trans. Image Process. **23**, 837–847 (2014)
21. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends Optim. **1**, 123–231 (2013)
22. Rakotomamonjy, A.: Direct optimization of the dictionary learning problem. IEEE Trans. Sig. Process. **61**, 5495–5506 (2013)
23. Naikal, N., Yang, A.Y., Sastry, S.S.: Towards an efficient distributed object recognition system in wireless smart camera networks. In: 13th Conference on Information Fusion (FUSION), pp. 1–8. IEEE (2010)
24. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 2169–2178. IEEE (2006)
25. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In: CVPR Workshop on Generative-Mode Based Vision. IEEE (2004)
26. Naikal, N., Yang, A.Y., Sastry, S.S.: Informative feature selection for object recognition via sparse PCA. In: IEEE International Conference on Computer Vision (ICCV), pp. 818–825. IEEE (2011)
27. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: DECAF: a deep convolutional activation feature for generic visual recognition. In: Proceedings of the 31st International Conference on Machine Learning (ICML-2014), pp. 647–655 (2014)
28. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10590-1_53
29. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**, 2278–2324 (1998)
30. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in Neural Information Processing Systems, pp. 487–495 (2014)
31. Gao, S., Tsang, I.W.H., Chia, L.T.: Laplacian sparse coding, hypergraph laplacian sparse coding, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 92–104 (2013)
32. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: IEEE International Conference on Computer Vision (ICCV), pp. 2146–2153. IEEE (2009)
33. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)