Multi-Level Discriminative Dictionary Learning With Application to Large Scale Image Classification

Li Shen, Gang Sun, Qingming Huang, *Senior Member, IEEE,* Shuhui Wang, *Member, IEEE,* Zhouchen Lin, *Senior Member, IEEE,* and Enhua Wu, *Member, IEEE*

***Abstract*— The sparse coding technique has shown flexibility and capability in image representation and analysis. It is a powerful tool in many visual applications. Some recent work has shown that incorporating the properties of task (such as discrimination for classification task) into dictionary learning is effective for improving the accuracy. However, the traditional supervised dictionary learning methods suffer from high computation complexity when dealing with large number of categories, making them less satisfactory in large scale applications. In this paper, we propose a novel multi-level discriminative dictionary learning method and apply it to large scale image classification. Our method takes advantage of hierarchical category correlation to encode multi-level discriminative information. Each internal node of the category hierarchy is associated with a discriminative dictionary and a classification model. The dictionaries at different layers are learnt to capture the information of different scales. Moreover, each node at lower layers also inherits the dictionary of its parent, so that the categories at lower layers can be described with multi-scale information. The learning of dictionaries and associated classification models is jointly conducted by minimizing an overall tree loss. The experimental results on challenging data sets demonstrate that our approach achieves excellent accuracy and competitive computation cost compared with other sparse coding methods for large scale image classification.**

***Index Terms*— Sparse coding, discriminative dictionary learning, hierarchical method, large scale classification.**

Manuscript received December 21, 2014; revised April 9, 2015; accepted May 13, 2015. Date of publication June 1, 2015; date of current version June 12, 2015. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2012CB316400, in part by the National Natural Science Foundation (NSF) of China under Grant 61332016, Grant 61025011, Grant 61303160, and Grant 61272326, and in part by Grant of University of Macau (MYRG202(Y1-L4)-FST11-WEH). The work of Z. Lin was supported in part by the 973 Program of China under Grant 2015CB352502, in part by the NSF of China under Grant 61272341 and Grant 61231002, and in part by the Microsoft Research Asia Collaborative Research Program. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jong Chul Ye. *(Corresponding author: Zhouchen Lin.)*

L. Shen is with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: li.shen@vipl.ict.ac.cn).

G. Sun is with the University of Chinese Academy of Sciences, Beijing 100049, China, also with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China. (e-mail: sung@ios.ac.cn).

Q. Huang is with the University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: qmhuang@ucas.ac.cn).

S. Wang is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wangshuhui@ict.ac.cn).

Z. Lin is with the Key Laboratory of Machine Perception, School of Electrical Engineering and Computer Science (Ministry of Education), Peking University, Beijing 100871, China, and also with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zlin@pku.edu.cn).

E. Wu is with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Macau, Macao 999078, China (e-mail: ehwu@umac.mo).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2438548

I. Introduction

IMAGE representation plays a critical role in image processing and analysis. Much work has been developed to generate representations by feature encoding schemes, such as sparse coding [1], [2], local coding [3], [4], super-vector coding [5] and Fisher vector [6]. In particular, the sparse coding technique has received much attention in recent years. It has shown flexibility and capability in many applications, such as image denoising [7], [8], image super-resolution [9] and face recognition [10]. In these tasks, the input signal (e.g., image or patch) is represented as a sparse linear combination of the bases in a dictionary. Some work also takes the viewpoint of analysis model to sparse representation [11]. Moreover, sparse coding has been successfully used in the Bag-of-Words model for general image classification [12]. Instead of vector quantization (VQ), image representation is computed based on the sparse codes of local descriptors (e.g., SIFT [13]).

To find an appropriate set of bases (i.e., dictionary), much effort is devoted to dictionary learning. Some methods are proposed to learn the dictionary in an unsupervised way [14], [15], where the dictionary is learnt by minimizing the reconstruction error of the input. Besides pure reconstruction, researchers also consider incorporating other properties of task, such as discrimination for classification task, into the learning of dictionary. Some work has shown that discriminative dictionary learning is effective for improving the performance [16]–[21]. A globally shared dictionary or multiple class-specific dictionaries are learnt to encode the underlying discriminative information into feature representations. However, when the number of categories is large, the complexity of these models grows dramatically. Thus, they usually suffer from considerable computing time in both training and testing, making them less attractive in large scale applications.

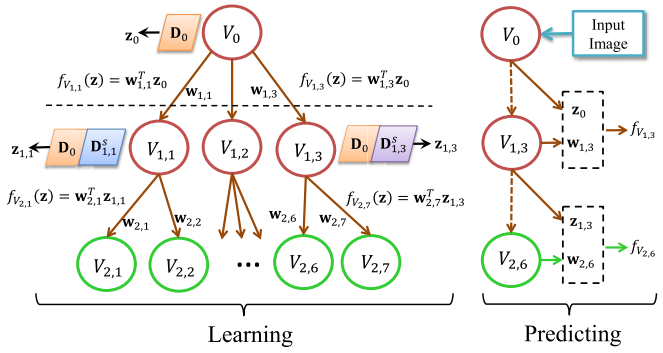


Fig. 1. Illustration of ML-DDL for image classification. In the learning stage, the dictionary $\mathbf{D}_0$, which is learnt at the root node V_0, is used to discriminate the child nodes $V_{1,1}$, $V_{1,2}$ and $V_{1,3}$. For the node $V_{1,1}$, its dictionary $\mathbf{D}_{1,1}$ consists of the inherited part $\mathbf{D}_0$ and the specific part $\mathbf{D}^s_{1,1}$. The corresponding representation $\mathbf{z}_{1,1}$ is fed into the classification model (i.e., classifiers with parameters $\mathbf{w}_{2,1}$ and $\mathbf{w}_{2,2}$). During the predicting stage, the test image goes through one path of the hierarchy by selecting the nodes with the maximal response. Only two (i.e., the depth of the hierarchy) feature representations have to be computed.

Much work has shown that exploiting the hierarchy to guide the model learning can bring improvement in efficiency [22]–[24] and accuracy [25]–[27]. The set of class labels is organized as a hierarchical structure (e.g., tree). The root node at the top layer contains all the classes. Each internal node (i.e., non-leaf node), as a hyper-category, is associated with a set of related classes. Each leaf node corresponds to a single class. The hierarchy can be built upon the semantic correlation or visual similarity among categories. Thus the structure reflects the hierarchical correlation among categories.

In this paper, we aim to take advantage of category hierarchy for discriminative dictionary learning. We can exploit some key observations on the benefit of hierarchy. First, category hierarchy displays diversified inter-correlation among the sibling nodes at different layers. The siblings at higher layers are less related, thus the discrimination among them is much easier [28], [29]. Second, the features of different granularity can be spotted from natural images. Simple features extracted from relatively small regions are generic and useful to classify less related concepts. Conversely, the features extracted from larger regions can capture more specific information (e.g., indicative of object parts), which can support the discrimination at lower layers [30]. Moreover, the nodes at lower layers are supposed to possess the general properties from their ancestors and additional class-specific details. In other words, the features chosen by different internal nodes, even at the same layer, are likely to be different.

Based on the above observations, we propose the Multi-Level Discriminative Dictionary Learning (ML-DDL) method, and exploit it for large scale image classification. The framework is shown in Fig. 1. Given the hierarchy, each internal node is associated with a discriminative dictionary and a classification model that discriminates its child nodes. Considering that the discrimination should rely on more complex and specific patterns at lower layers, we propose to learn the associated dictionaries to encode the descriptors at a larger scale (i.e., extracted from larger image regions). Moreover, each node at lower layers also inherits the dictionary of its parent, so that the categories at lower layers can be described with sufficient discriminative information at multiple scales. The ensemble of dictionaries and classification models are jointly learnt by optimizing an overall tree loss. Besides, ML-DDL can extend to multiple feature channels, i.e., features from different sources. Accordingly, complementary information can be explored from data to further improve the performance. We summarize the main contributions of our approach as follows:

- We propose a Multi-Level Discriminative Dictionary Learning (ML-DDL) method which incorporates the hierarchical category relation into dictionary learning. We learn hierarchical discriminative dictionaries to encode the descriptors at different scales. Moreover, by virtue of dictionary inheritance, the discriminative information of multiple scales can be leveraged to further improve the separability of low-layer nodes. Compared with other sparse coding methods [12], [17], [19], [31], ML-DDL can effectively capture multi-level discriminative information, and achieve improved accuracy for large scale classification.
- Both the training and testing computation cost can be significantly reduced when compared with other supervised dictionary learning methods [17], [19]. The hierarchy-based learning and predicting scheme makes ML-DDL computationally tractable when dealing with large number of categories.

This paper is an extension of our previous work [32]. Compared with the earlier version, we enrich our methodology by extending ML-DDL to multiple feature channels. Moreover, we give a detailed and principled analysis and deduction. Besides, we also provide a richer experiment section with quantitative and qualitative improvement, which includes more experimental results and analysis, as well as the improved performance over the earlier version.

II. Related Work

In this section, we briefly review the related work on dictionary learning. Moreover, we also review a series of hierarchy-based methods for image classification.

A. Dictionary Learning

Current dictionary learning approaches can be categorized to two main types: unsupervised and supervised (discriminative) dictionary learning. In the field of unsupervised dictionary learning, dictionary is optimized by minimizing the reconstruction errors of signals [2], [14], [15], [33]. Yang *et al.* [12] propose to learn a unique unsupervised dictionary by sparse coding for classification and achieve impressive results. In the work [34], a two-layer sparse coding scheme is proposed by using the spatial neighborhood dependency among local patches. Besides the sparsity, other constraints (such as local geometry and dependency between dictionary elements) are imposed to incorporate more information [3], [4], [35]. Due to the learning criterion which is based on reconstruction

rather than discrimination, unsupervised dictionary may lack sufficient discriminative power.

In this regard, much research has focused on discriminative dictionary learning. A family of such methods is proposed to train a single discriminative dictionary which is shared by all the categories [16]–[18], [36], [37]. Another type of approaches learn multiple class-specific dictionaries for each class [19], [38], [39]. Moreover, the work [40] and [41] is proposed to train structured dictionaries in which the elements explicitly correspond to category labels. In these approaches, the learning of dictionaries integrates a discrimination criterion (e.g., classification loss or Fisher discrimination criterion) and the reconstruction error into an optimization objective. However, these supervised dictionary learning methods suffer from high computation complexity in both training and testing, making them less attractive in large scale applications.

Our method aims at learning discriminative dictionaries and exploit them for large scale image classification, which is connected to the work [31]. In [31], the authors learn one common dictionary and multiple category-specific dictionaries for each group of categories according to the Fisher discrimination criterion. Different from their two-stage procedure, ML-DDL learns the dictionaries and discriminative models simultaneously. Besides, categories are required to be clustered into groups in [31], which restricts the method from being applicable to a hierarchy with more than two layers. In contrast, ML-DDL can be developed with a flexible hierarchical structure.

B. Hierarchy-Based Classification Models

When dealing with a large number of categories, much of the effort has been devoted to hierarchy-based models, which take advantage of hierarchical structure to guide model learning. One line of work is imposing statistical constraints on the learning of classifiers in the hierarchy. In the work [27], [42], [43], similarity priors are imposed to encourage the sibling nodes to share model parameters. This constraint drives the classifiers of nearby nodes much closer. On the other hand, dissimilarity constraint is introduced in [44] and [45], which encourages the classifier at each node to be different from the ones at its ancestors. Another direction is exploiting hierarchical loss in model learning, which considers the weighted classification error at different nodes [28], [46], [47].

With respect to hierarchy, it is usually generated according to prior knowledge (e.g., semantic relation) or by other process. Some work is developed on learning the hierarchy [23], [24], [26], [48], [49]. Besides, in [50] hierarchical structure is exploited to capture contextual information, such as object co-occurrence and spatial layout.

III. Discriminative Dictionary Learning

In this section, we review traditional discriminative dictionary learning with a flat label structure. Then we describe ML-DDL algorithm and extend it to multi-feature channels in the following sections. For reading convenience, we list the frequently used notations and their nomenclature in Table I.

TABLE I
NOTATION AND NOMENCLATURE IN THIS PAPER

$\mathbf{a}$	Vector
$\mathbf{a}_i, \mathbf{a}_i^j$	Vector indexed for some purpose
$(\mathbf{a})_j$	The j-th entry of vector $\mathbf{a}$
$\|\mathbf{a}\|_p$	The ℓ_p norm of vector $\mathbf{a}$
$\mathbf{A}$	Matrix
$\mathbf{A}_i, \mathbf{A}_i^j$	Matrix indexed for some purpose
$(\mathbf{A})_{ij}$	The (i,j)-th entry of matrix $\mathbf{A}$
$\mathbf{A}_{\cdot j}$	The j-th column of matrix $\mathbf{A}$
$\mathbf{A}_{i\cdot}$	The i-th row of matrix $\mathbf{A}$
$\|\mathbf{A}\|_F$	The Frobenius norm of matrix $\mathbf{A}$
$\mathcal{A}$	Set
$\|\mathcal{A}\|$	The cardinality of set $\mathcal{A}$

We assume that the training set $(\mathcal{X}, \mathcal{Y})$ consists of N training samples. Each sample (i.e., image) $x \in \mathcal{X}$ is represented by a local descriptor set $\{\mathbf{x}_n\}_{n=1}^{N_p}$, $\mathbf{x}_n \in \mathbb{R}^m$, and $y \in \mathcal{Y}$ is the label of sample x. N_p is the number of descriptors belonging to the sample. Given a dictionary $\mathbf{D} \in \mathbb{R}^{m \times b}$, where b is the dictionary size and $b > m$, the sparse code $\hat{\boldsymbol{\alpha}}_n \in \mathbb{R}^b$ for descriptor $\mathbf{x}_n$ can be computed by

$$\hat{\boldsymbol{\alpha}}_n = \arg\min_{\boldsymbol{\alpha}_n} \frac{1}{2} \|\mathbf{x}_n - \mathbf{D}\boldsymbol{\alpha}_n\|_2^2 + \mu \|\boldsymbol{\alpha}_n\|_1, \tag{1}$$

where $\mu > 0$ is a sparsity parameter. The code set $\{\hat{\boldsymbol{\alpha}}_n\}$ of sample x, can be regarded as a matrix $\mathbf{A}$, which each column corresponds to the code of a descriptor. Max pooling can be applied on these codes to generate an image-level representation $\mathbf{z}$: $\mathbf{z} = p_{\max}(\mathbf{A})$. $p_{\max}$ operates on each row of the matrix, and returns a vector whose j-th element is

$$(\mathbf{z})_j = (p_{\max}(\mathbf{A}))_j = \max\{|(\hat{\boldsymbol{\alpha}}_1)_j|, \ldots, |(\hat{\boldsymbol{\alpha}}_{N_p})_j|\}. \tag{2}$$

The generated representation by max pooling is endowed with some good property (e.g., translation invariance), but the spatial layout of local descriptors has been discarded. Spatial pyramid pooling [12] further considers the spatial statistical information, where the max pooling operation performs on the spatial pyramid of sample x. The representation $\mathbf{z}$ is produced by a concatenation of the pooling results, which is denoted as $\mathbf{z} = p_{\text{sp}}(\mathbf{A})$. Representation $\mathbf{z}$ can be regarded as the transformation of x related to $\mathbf{D}$, and we define the process as $\mathbf{z} = \phi(x, \mathbf{D})$.

Each sample x is associated with a label $y \in \mathcal{Y}$, which we aim to predict from x. Regarding a traditional flat model, a classification model $f(\phi(x, \mathbf{D}), \mathbf{W})$ needs to be trained according to loss $\ell_0(y, f(\phi(x, \mathbf{D}), \mathbf{W}))$, where $\mathbf{W}$ denotes the classification model parameters. To obtain a dictionary in supervised setting [18], [19], the learning of $\mathbf{D}$ and $\mathbf{W}$ can be formulated as follows,

$$\min_{\mathbf{D},\mathbf{W}} \sum_{i=1}^{N} \ell_0(y_i, f(\phi(x_i, \mathbf{D}), \mathbf{W})) + \frac{\lambda}{2} \|\mathbf{W}\|_F^2 \tag{3}$$

where $\lambda > 0$ is a regularization parameter. To prevent the dictionary $\mathbf{D}$ from becoming arbitrarily large, $\mathbf{D}$ satisfies the constraint $\mathbf{D} \in \mathcal{B}_D$, where

$$\mathcal{B}_D \triangleq \left\{\mathbf{D} \in \mathbb{R}^{m \times b} \,|\, \|\mathbf{d}_j\|_2 \leq 1, \forall j \in \{1, \ldots, b\}\right\}. \tag{4}$$

Algorithm 1 Classify Input x Given Hierarchy $\mathcal{H}$

Initialize v to the root node
while $\mathcal{C}(v) \neq \emptyset$ **do**
  $v \leftarrow \arg\max_{c \in \mathcal{C}(v)} f_c(\phi(x, \mathbf{D}_v), \mathbf{w}_c)$
end
return v.

The joint learning of $\mathbf{D}$ and $\mathbf{W}$ endows the dictionary $\mathbf{D}$ with discriminative power as we minimize the classification loss.

IV. Proposed Algorithm

In the context of hierarchical model, the set of K categories $\mathcal{K} = \{1, \cdots, K\}$ is organized into a tree structure $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$. The depth of tree is denoted by L. The root node containing all classes is at layer 0. The leaf nodes are at layer L. Let $\mathcal{I}$ denote the set of internal nodes, which are non-leaf nodes. Let $\mathcal{D} = \{\mathbf{D}_v\}$ and $\mathcal{W} = \{\mathbf{W}_v\}$ denote the set of dictionaries and classification model parameters, respectively.

Each node $v \in \mathcal{I}$ is associated with a class label set $\mathcal{K}_v \subseteq \mathcal{K}$ and a classification model $f(\phi(x, \mathbf{D}_v), \mathbf{W}_v)$. $\mathcal{C}(v)$ denotes the set of its child nodes, and $f_c(\phi(x, \mathbf{D}_v), \mathbf{w}_c)$ represents the response of f on the child node $c \in \mathcal{C}(v)$, where $\mathbf{w}_c$ is a column of $\mathbf{W}_v$. With respect to each child node c, its associated label set is required to be a subset of its parent's set, i.e., $\mathcal{K}_v = \bigcup_{c \in \mathcal{C}(v)} \mathcal{K}_c$. Moreover, siblings share no common label, i.e., $\mathcal{K}_{c_1} \cap \mathcal{K}_{c_2} = \emptyset, \forall c_1, c_2 \in \mathcal{C}(v)$.

Given a hierarchy $\mathcal{H}$, when classifying a sample x we can make the prediction by applying Algorithm 1. The prediction starts at the root node and proceeds recursively until reaching a leaf node. When arriving at a node, the child with the largest response is selected.

A. Hierarchical Discriminative Dictionary Learning

To learn the dictionaries and classification models, we need to define the tree loss. For a sample x, $\hat{y}$ denotes the final prediction, which is made by finding the best path in the hierarchy. Let $\mathcal{A}(\hat{y})$ denote a node set, which corresponds to the path from the root to the leaf node containing label $\hat{y}$. The classification error occurs when the true label does not appear in the path, i.e., $y \notin \mathcal{K}_v, v \in \mathcal{A}(\hat{y})$. Assuming that selecting a node only depends on its adjacent node at the higher layer, we resort to maximum likelihood to solve the problem. The true path is denoted by $\mathcal{A}(y) = \{v_0, \cdots, v_L\}$. Thus, the probability of predicting the true label y given x can be expressed as

$$p(y|x) = p(v_0|x) \prod_{l=0}^{L-1} p(v_{l+1}|v_l, x) = \prod_{l=0}^{L-1} p(v_{l+1}|v_l, x). \tag{5}$$

$p(v_0|x) = 1$ as the root contains all classes. Here, when the nodes do not have a common parent, the competition among them does not appear in (5). It is consistent with the strategy of finding an optimal path in Algorithm 1.

According to the tree properties, i.e., siblings share no common label, we consider that the nodes under the same parent to be mutually exclusive. With this assumption, (5) can be rewritten by using the indicator function $\mathbb{I}(\cdot)$,

$$\begin{aligned} p(y|x) &= \prod_{l=0}^{L-1} \prod_{c \in \mathcal{C}(v_l)} p\left(c|v_l, x\right)^{\mathbb{I}(c=v_{l+1})} \\ &= \prod_{l=0}^{L-1} \prod_{c \in \mathcal{C}(v_l)} p\left(c|v_l, x\right)^{\mathbb{I}(c \in \mathcal{A}(y))}. \end{aligned} \tag{6}$$

Given node $v_l \in \mathcal{A}(y)$ and x (supposing that the children number of v_l is C), the indicator $\mathbb{I}\left(c = v_{l+1}\right)$ on child nodes $c \in \mathcal{C}(v_l)$ provides the one-of-C encoding of the true prediction $v_{l+1} \in \mathcal{A}(y)$, and the c-th bit turns on if node c belongs to the true path. That is to say, the competition only occurs among the sibling nodes along the true path of the label y. And for the other nodes, the bit of indicator turns off naturally. When considering the ensemble of the nodes in the hierarchy, with some algebra (multiplying by some constant terms whose value is equal to 1), (6) takes an equivalent form

$$p(y|x) = \prod_{v \in \mathcal{I}} \prod_{c \in \mathcal{C}(v)} p(c|v, x)^{\mathbb{I}(c \in \mathcal{A}(y))}. \tag{7}$$

Each internal node v corresponds to a multi-class classification problem for its child nodes. Multinomial logistic regression can be applied to model the problem, i.e., the probability takes the form

$$p(c|v, x) = \frac{\exp(f_c(x))}{\sum_{u \in \mathcal{C}(v)} \exp(f_u(x))}, \tag{8}$$

where the response $f_c(x)$ can be defined as

$$f_c(x) \triangleq f_c(\phi(x, \mathbf{D}_v), \mathbf{w}_c) = \mathbf{w}_c^T \phi(x, \mathbf{D}_v). \tag{9}$$

$\mathbf{D}_v$ denotes the associated dictionary of node v. Accordingly, the tree loss over sample x can be estimated as the negative log-likelihood,

$$\begin{aligned} &\ell(y, x, \mathcal{D}, \mathcal{W}) \\ &= -\sum_{v \in \mathcal{I}} \sum_{c \in \mathcal{C}(v)} \mathbb{I}(c \in \mathcal{A}(y)) \log p(c|v, x) \\ &= -\sum_{v \in \mathcal{I}} \sum_{c \in \mathcal{C}(v)} \mathbb{I}(c \in \mathcal{A}(y)) \log \frac{\exp\left(f_c(x)\right)}{\sum_{u \in \mathcal{C}(v)} \exp\left(f_u(x)\right)} \\ &= -\sum_{v \in \mathcal{I}} \left[\sum_{c \in \mathcal{C}(v)} \mathbb{I}(c \in \mathcal{A}(y)) f_c(x) - \log\left(\sum_{u \in \mathcal{C}(v)} \exp(f_u(x)) \right) \right]. \end{aligned} \tag{10}$$

It means that we need to maximize the response of the nodes with true label, which can attain small loss in the hierarchy. Meanwhile, the optimization would reduce the probability of selecting other nodes under the same parent when increasing the probability of the node with true label.

The decision at each node v is associated with the dictionary $\mathbf{D}_v$ as well as the classifier parameters $\mathbf{W}_v$. This is different from the traditional hierarchical models, which are built upon an identical feature space. By virtue of the dictionary $\mathbf{D}_v$, the original features are projected into the corresponding subspace, where the associated classifiers can effectively classify the samples of child nodes.

The classification at different layers of the hierarchy is likely to rely on different features. The node at higher layers can be regarded to describe a general concept. In contrast, the node at lower layers corresponds to more specific concept. Besides, different scale features can be extracted from samples, from simple features (such as edges) to more specific features (indicative of object parts). The simple features are generic and useful for classifying general classes. Conversely, specific features can be used to describe more specific concepts [30]. In order to exploit such properties, each sample x is represented by a set of multi-scale descriptors, $\{s^0, \cdots, s^M\}$, which are extracted with different patch scales. We learn the dictionaries of different layers to encode descriptors at different scales accordingly.

B. Dictionary Inheritance

The dictionaries learnt at higher layers can be regarded as the shared properties for the nodes at lower layers. Thus, we propose *dictionary inheritance*, which allows them to be inherited by the child nodes. With respect to the internal node v, if it is a non-root node, the dictionary $\mathbf{D}_v$ decomposes into two parts: the inherited part $\mathbf{D}_v^i$ and the specific part $\mathbf{D}_v^s$. For example, considering the node $V_{1,1}$ in Fig. 1, the corresponding dictionary $\mathbf{D}_{1,1}$ is expressed as $\mathbf{D}_{1,1} \triangleq \{\mathbf{D}_{1,1}^i, \mathbf{D}_{1,1}^s\} = \{\mathbf{D}_0, \mathbf{D}_{1,1}^s\}$. $\mathbf{D}_0$ is the inherited part, which denotes the dictionary learnt at V_0. $\mathbf{D}_{1,1}^s$ is the specific part learnt at $V_{1,1}$. Sample x is represented by a multi-scale descriptor set $\{s^0, s^1\}$, where s^i is the subset of descriptors at a certain scale. s^1 corresponds to the descriptors at a scale larger than s^0. Accordingly, the response of sample x on the child node $V_{2,1}$ in (9) can be rewritten as

$$\begin{aligned} f_{V_{2,1}}(x) &= \mathbf{w}_{2,1}^T \phi(x, \mathbf{D}_{1,1}) \\ &= \mathbf{w}_{2,1}^T \left[\phi(s^0, \mathbf{D}_0)^T, \phi(s^1, \mathbf{D}_{1,1}^s)^T\right]^T. \end{aligned} \tag{11}$$

The descriptors at different scales are encoded by respective dictionaries, i.e., we encode s^0 via $\mathbf{D}_0$ and s^1 via $\mathbf{D}_{1,1}^s$. The generated representations are concatenated to describe sample x at current node $V_{1,1}$. Thus, the image-level representation $\phi(x, \mathbf{D}_{1,1})$ integrates the discriminative information of multiple scales.

Given the training set $(\mathcal{X}, \mathcal{Y}) = \{(x_i, y_i)\}_{i=1}^N$, the joint learning of dictionary set $\mathcal{D}$ and classification model parameters $\mathcal{W}$ can be formulated by minimizing the following regularized loss,

$$R = \sum_{i=1}^{N} \ell\left(y_i, x_i, \mathcal{D}, \mathcal{W}\right) + \frac{\lambda}{2} \sum_{v \in \mathcal{I}} \|\mathbf{W}_v\|_F^2 \tag{12}$$

where the loss function ℓ is given in (10).

The information propagates via multi-level dictionaries in a top-down fashion. Different from traditional sharing models [27], [42], our approach can leverage the shared information from the parent node to improve the discrimination among its children. As a matter of fact, dictionary inheritance encourages the model to exploit sufficient information at multiple scales.

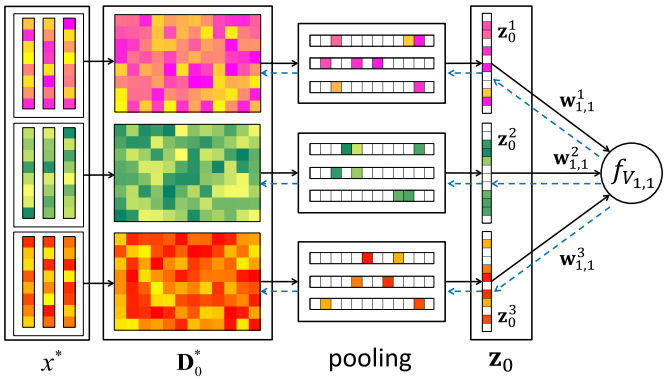


Fig. 2. ML-DDL with multi-feature channels. For the node V_0 in Fig. 1, $\mathbf{D}_0^*$ is composed of three subdictionaries corresponding to different sources of features. The image-level representation $\mathbf{z}_0$ is generated by combining the outputs $\{\mathbf{z}_0^1, \mathbf{z}_0^2, \mathbf{z}_0^3\}$ via respective subdictionaries. Then $\{\mathbf{z}_0^1, \mathbf{z}_0^2, \mathbf{z}_0^3\}$ are weighted by the parameter vector $\mathbf{w}_{1,1}^* = \{\mathbf{w}_{1,1}^1, \mathbf{w}_{1,1}^2, \mathbf{w}_{1,1}^3\}$, and the sum of the responses at three channels is regarded as the final response at the child node $V_{1,1}$. The classification error propagates backwards to update multiple subdictionaries simultaneously.

C. Extending ML-DDL to Multi-Feature Channels

Besides enhancing the discriminative power of a single feature, integrating multiple sources of features (color, shape, texture, etc.) is also an effective way to improve the descriptive ability of representation [51]–[53]. ML-DDL is flexible in extending the discriminative dictionary learning to multi-feature channels.

Considering that each sample x is represented by multiple sets of local descriptors corresponding to different features, we use $x^* = \{x^1, \cdots, x^J\}$ instead to denote the descriptor set, where x^j corresponds to the subset of the j-th feature and J is the number of features. For the node v, we exploit a generalized dictionary $\mathbf{D}_v^*$ to fuse the information conveyed by multiple features, i.e., $\mathbf{D}_v^j$ denotes the subdictionary used to encode the j-th feature. Consequently, the response $f_c(\cdot)$ at the child c in (9) can be redefined as

$$f_c(x^*) = \sum_{j=1}^{J} e_j (\mathbf{w}_c^j)^T \phi(x^j, \mathbf{D}_v^j) \triangleq (\mathbf{w}_c^*)^T \phi(x^*, \mathbf{D}_v^*), \tag{13}$$

where $(\mathbf{w}_c^*)^T = \left[e_1(\mathbf{w}_c^1)^T, e_2(\mathbf{w}_c^2)^T, \cdots, e_J(\mathbf{w}_c^J)^T\right]$, and $\phi(x^*, \mathbf{D}_v^*) = \left[\phi(x^1, \mathbf{D}_v^1)^T, \phi(x^2, \mathbf{D}_v^2)^T, \cdots, \phi(x^J, \mathbf{D}_v^J)^T\right]^T$. e_j is the weight for the j-th feature, and $\mathbf{w}_c^j$ denotes the corresponding parameters of classification model. In (13), we apply a linear combination of multiple feature responses. Considering the joint learning of classification model and representations, we do not explicitly choose the value of weight e_j, which is integrated in $\mathbf{w}_c^*$.

$\mathbf{w}_c^*$ and $\mathbf{D}_v^*$ are regarded as the classification model parameters and the dictionary with multi-feature channels. They can be learnt by minimizing the regularized loss R in (12), and the response f_c uses (13) instead. The classification error propagates backwards to update multiple subdictionaries simultaneously, as shown in Fig. 2.

V. Optimizing Dictionaries and Model Parameters

For notational convenience, the following optimization is based on single feature. It is straightforward to extend to the multi-feature version.

Although optimizing the entire objective function in (12) is complicated, the problem can be decomposed into a set of sub-problems. The learning proceeds sequentially in a top-down fashion over layers, and the nodes at the same layer can be tackled independently. For an internal node υ, the problem can be solved by performing the following two steps iteratively:

1) *Coding*: Fixing the dictionary $\mathbf{D}_\upsilon$, we compute the corresponding code coefficients $\mathbf{A}_\upsilon$ for each sample x, and generate the image-level representation $\mathbf{z}_\upsilon$.
2) *Dictionary and model parameters updating*: Based on the representation computed by the previous dictionary, we update model parameters $\mathbf{W}_\upsilon$ and dictionary $\mathbf{D}_\upsilon$ according to the objective function (12). Note that only the specific part $\mathbf{D}_\upsilon^s$ is updated. The inherited part $\mathbf{D}_\upsilon^i$ is unchanged as it has been optimized at the higher layer.

A. Optimization Over Dictionary

As the regularized loss R in (12) is differentiable with respect to the dictionary and the model parameters [19], we can deduce the gradient of R with respect to the specific dictionary $\mathbf{D}_\upsilon^s$: $\nabla_{\mathbf{D}_\upsilon^s} R = \sum_{i=1}^N \partial \ell(y_i, x_i, \mathcal{D}, \mathcal{W})/\partial \mathbf{D}_\upsilon^s$ by using the chain rule,

$$\frac{\partial \ell}{\partial \mathbf{D}_\upsilon^s} = \frac{\partial \ell}{\partial \mathbf{z}_\upsilon}\frac{\partial \mathbf{z}_\upsilon}{\partial \mathbf{D}_\upsilon^s} = \frac{\partial \ell}{\partial \mathbf{z}_\upsilon}\frac{\partial \mathbf{z}_\upsilon}{\partial \mathbf{A}_\upsilon}\frac{\partial \mathbf{A}_\upsilon}{\partial \mathbf{D}_\upsilon^s}. \tag{14}$$

The gradient of ℓ with respect to $\mathbf{z}_\upsilon$ can be computed as

$$\frac{\partial \ell}{\partial \mathbf{z}_\upsilon} = -\sum_{c\in\mathcal{C}(\upsilon)} \mathbb{I}\left(\upsilon \in \mathcal{A}(y)\right)\mathbf{w}_\upsilon + \frac{\sum\limits_{u\in\mathcal{C}(\upsilon)} \exp(f_u(x))\mathbf{w}_u}{\sum\limits_{u\in\mathcal{C}(\upsilon)} \exp(f_u(x))}. \tag{15}$$

The main difficulty of the optimization comes from obtaining the derivative of coefficients $\mathbf{A}_\upsilon$ with respect to dictionary $\mathbf{D}_\upsilon^s$, because they are implicitly connected and the ℓ_1 regularization on $\mathbf{A}_\upsilon$ is non-smooth. To prevent notation clutter, we drop the subscript υ and the superscript of dictionary in the following deductions.

To establish the connection between $\mathbf{A}$ and $\mathbf{D}$, we consider the relationship between a sparse code $\hat{\boldsymbol{\alpha}}_n$ (i.e., a column of $\mathbf{A}$) and dictionary $\mathbf{D}$. The derivative of (1) with respect to $\boldsymbol{\alpha}_n$ at its minimum $\hat{\boldsymbol{\alpha}}_n$ can be expressed as,

$$\left.\frac{\partial \|\mathbf{x}_n - \mathbf{D}\boldsymbol{\alpha}_n\|_2^2}{\partial \boldsymbol{\alpha}_n}\right|_{\boldsymbol{\alpha}_n=\hat{\boldsymbol{\alpha}}_n} = -2\mu \left.\frac{\partial \|\boldsymbol{\alpha}_n\|_1}{\partial \boldsymbol{\alpha}_n}\right|_{\boldsymbol{\alpha}_n=\hat{\boldsymbol{\alpha}}_n}. \tag{16}$$

According to the subdifferential $\partial\|\hat{\boldsymbol{\alpha}}_n\|_1 = \mathbf{q}$ [54], where

$$q_j = \begin{cases} \operatorname{sgn}(\hat{\boldsymbol{\alpha}}_n)_j, & \text{if } (\hat{\boldsymbol{\alpha}}_n)_j \neq 0, \\ |q_j| \leq 1, & \text{otherwise,} \end{cases} \tag{17}$$

we can get the following equation from (16),

$$\mathbf{D}_\Lambda^T(\mathbf{x}_n - \mathbf{D}_\Lambda\hat{\boldsymbol{\alpha}}_\Lambda) = \mu \operatorname{sgn}(\hat{\boldsymbol{\alpha}}_\Lambda), \tag{18}$$

where Λ denotes the active set, comprised of the indices of nonzero coefficients in $\hat{\boldsymbol{\alpha}}_n$. $\mathbf{D}_\Lambda$ denotes the matrix comprised of the corresponding columns (atoms) of $\mathbf{D}$ whose column indices are in Λ.

When the perturbation of dictionary is small, a stable active set can be obtained. We only compute the gradient of the active coefficients $\hat{\boldsymbol{\alpha}}_\Lambda$ with respect to the active atoms $\mathbf{D}_\Lambda$, and set other entries of the gradient to zeros [19]. Here we apply implicit differentiation on (18),

$$\frac{\partial\left(\mathbf{D}_\Lambda^T(\mathbf{x}_n - \mathbf{D}_\Lambda\hat{\boldsymbol{\alpha}}_\Lambda)\right)}{\partial(\mathbf{D}_\Lambda)_{ij}} = \frac{\partial\left(\mu \operatorname{sgn}(\hat{\boldsymbol{\alpha}}_\Lambda)\right)}{\partial(\mathbf{D}_\Lambda)_{ij}}. \tag{19}$$

As the sign in $\hat{\boldsymbol{\alpha}}_\Lambda$ should not change for small perturbation of the dictionary, we have $\partial \operatorname{sgn}(\hat{\boldsymbol{\alpha}}_\Lambda)/\partial\mathbf{D}_\Lambda = 0$. Thus, we obtain the derivative according to (19),

$$\begin{aligned}\frac{\partial \hat{\boldsymbol{\alpha}}_\Lambda}{\partial(\mathbf{D}_\Lambda)_{ij}} &= \left(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda\right)^{-1}\left[\frac{\partial \mathbf{D}_\Lambda^T}{\partial(\mathbf{D}_\Lambda)_{ij}}\mathbf{x}_n - \frac{\partial(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)}{\partial(\mathbf{D}_\Lambda)_{ij}}\hat{\boldsymbol{\alpha}}_\Lambda\right] \\ &= \left(\mathbf{x}_n - \mathbf{D}\hat{\boldsymbol{\alpha}}\right)_i\mathbf{Q}_{\cdot j} - (\hat{\boldsymbol{\alpha}}_\Lambda)_j\left(\mathbf{Q}\mathbf{D}_\Lambda{}^T\right)_{\cdot i},\end{aligned} \tag{20}$$

where $\mathbf{Q} = \left(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda\right)^{-1}$. Accordingly we can compute the gradient of $\mathbf{z}$ based on (2),[1]

$$\frac{\partial(\mathbf{z})_k}{\partial \mathbf{D}} = \begin{cases} \operatorname{sgn}\left((\hat{\boldsymbol{\alpha}}_n)_k\right)\frac{\partial(\hat{\boldsymbol{\alpha}}_n)_k}{\partial \mathbf{D}}, & \text{if } (\mathbf{z})_k = |(\hat{\boldsymbol{\alpha}}_n)_k| \wedge (\mathbf{z})_k \neq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

That is to say, the gradient of multivariate max operation is a union of the ones of the active entries in the coefficient matrix $\mathbf{A}$ (i.e., the entries with the largest absolute values in each row of $\mathbf{A}$). Consequently, we get the gradient of ℓ with respect to the dictionary $\mathbf{D}$ in (14) by integrating (15) and (21).

B. Overall Optimization Procedure

We employ stochastic gradient descent algorithm and mini-batch strategy for optimization. The overall optimization procedure is given in Algorithm 2. To initialize the dictionaries, we first learn a set of unsupervised dictionaries $\mathcal{D}_{\text{un}} = \{\mathbf{D}_l\}_{l=0}^{L-1}$. Each dictionary $\mathbf{D}_l$ corresponds to the descriptors at a certain scale.

For an internal node υ at layer l, its model parameters $\mathbf{W}_\upsilon$ are initialized to zeros. The dictionary $\mathbf{D}_\upsilon$ is comprised of the inherited part $\mathbf{D}_\upsilon^i$ from its parent node and the specific part $\mathbf{D}_\upsilon^s$. (In particular, the dictionary at the root node only has the specific part, i.e., $\mathbf{D}_0^i = \emptyset$.) The specific dictionary $\mathbf{D}_\upsilon^s$ is initialized with $\mathbf{D}_l$. Given the training image set $\{(x, y)\}_\upsilon$, a number of iterations (the number is denoted by $nIter$) are performed for updating $\mathbf{D}_\upsilon^s$ and $\mathbf{W}_\upsilon$. At each iteration, we randomly choose a batch of samples from data (denoted by $\mathcal{S}_k$), and compute the representation $\mathbf{z}_\upsilon = \phi(x, \mathbf{D}_\upsilon)$ for each sample x belonging to the batch. Considering the costly computation in recomputing the coding at each iteration, we approximate the representation $\mathbf{z}$ by coding parts of the descriptors in each sample (typically ten percent). Then we apply a two-layer spatial pooling (p_{sp} with the pyramid 1×1 and 3×3)

[1]The computation with spatial pyramid pooling can be done by accumulating the gradients over the pyramid.

Algorithm 2 ML-DDL

Input: Data $(\mathcal{X}, \mathcal{Y})$, category hierarchy $\mathcal{H}$, initial dictionary set $\mathcal{D}_{\text{un}} = \{\mathbf{D}_l\}_{l=0}^{L-1}$, parameters L, μ, λ, $nBatch$, $nIter$, η_0, ρ

Output: Dictionaries $\mathcal{D} = \{\mathbf{D}_v\}_{v\in\mathcal{I}}$, model parameters $\mathcal{W} = \{\mathbf{W}_v\}_{v\in\mathcal{I}}$

for $l = 0$ **to** $L-1$ **do**
  foreach v *at layer* l **do**
    Initialize: $\{(x,y)\}_v \leftarrow \{x \in \mathcal{X}, s.t. v \in \mathcal{A}(y)\}$, $\mathbf{W}_v \leftarrow \mathbf{0},\ \mathbf{D}_v^s \leftarrow \mathbf{D}_l,\ \mathbf{D}_v \leftarrow [\mathbf{D}_v^i, \mathbf{D}_v^s]$
    for $k = 1$ **to** $nIter$ **do**
      1. Choose a data batch $\mathcal{S}_k \subset \{(x,y)\}_v$, where $|\mathcal{S}_k| = nBatch$
      2. Compute representation $\mathbf{z}_v = \phi(x, \mathbf{D}_v)$ for $x \in \mathcal{S}_k$
      3. Update dictionary and model parameters $\eta_k = \eta_0 \cdot \rho/(\rho + k)$
      $\mathbf{W}_v \leftarrow \Pi_{\mathcal{B}_W}\big(\mathbf{W}_v - \eta_k \nabla_{\mathbf{W}_v} R_k\big)$
      $\mathbf{D}_v^s \leftarrow \Pi_{\mathcal{B}_D}\big(\mathbf{D}_v^s - \eta_k \nabla_{\mathbf{D}_v^s} R_k\big)$
    end
  end
end

after coding. The total loss is approximated by accumulating the loss on the batch $\mathcal{S}_k$, where $R_k = \sum_{(x,y)\in\mathcal{S}_k} \ell(y, x, \mathbf{D}_v, \mathbf{W}_v) + \frac{\lambda}{2}\|\mathbf{W}_v\|_F^2$. Thus, the gradient with respect to model parameters is expressed as $\nabla_{\mathbf{w}_c} R_k = \sum_{(x,y)\in\mathcal{S}_k} \partial\ell(y, x, \mathbf{D}_v, \mathbf{W}_v)/\partial\mathbf{w}_c + \lambda\mathbf{w}_c$, where

$$\frac{\partial \ell}{\partial \mathbf{w}_c} = -\mathbb{I}\left(c \in \mathcal{A}(y)\right)\mathbf{z}_v + \frac{\exp\left(f_c(x)\right)\mathbf{z}_v}{\sum\limits_{u\in\mathcal{C}(v)} \exp\left(f_u(x)\right)}, \tag{22}$$

and $\mathbf{w}_c$ is a column of $\mathbf{W}_v$. The gradient $\nabla_{\mathbf{D}_v^s} R_k$ can be computed according to (14). Then the parameters are updated with a projected gradient step, by projecting the model parameters $\mathbf{W}_v$ and dictionary $\mathbf{D}_v^s$ onto the convex set $\mathcal{B}_W = \{\mathbf{W} : \|\mathbf{W}\|_F \le \sqrt{N/\lambda}\}$ (N is the sample number) and $\mathcal{B}_D$ (4), respectively. $\Pi_{\mathcal{B}}(\mathbf{u})$ defines the closest point in the set $\mathcal{B}$ to the target $\mathbf{u}$. η_0 and ρ are the parameters for learning rate, which are respectively set to 1.0 and 100.

The overall optimization follows a top-down fashion, i.e., the learning performs sequentially from top to bottom. The nodes at the same layers can be tackled independently. Moreover, due to the fact that the inherited dictionary is not updated at the descendant nodes, the corresponding representations can be directly used at lower layers.

VI. Experiments

In this section, we evaluate the performance of ML-DDL on three datasets: SUN397 [55], which is a large database for scene recognition, ImageNet200, which is a subset of ImageNet [56] with imbalanced sample distribution, and ImageNet1K, which is a large scale dataset for object classification and is used in ILSVRC2010 [57]. First, we verify the effectiveness of ML-DDL in accuracy and efficiency by comparing with other sparse coding methods on SUN397 and ImageNet200. Then we validate the scalability and flexibility of ML-DDL on large scale data ImageNet1K. Furthermore, we show the necessity of hierarchical discriminative dictionary learning and dictionary inheritance, and investigate the effect of ML-DDL with multi-feature channels. Finally, we examine the effects of pooling strategy for training dictionaries, as well as training sample size and dictionary size on the method.

TABLE II
Configurations of Different Methods

Method	Label Struct.	Dict. Property	# of Dict.
F-UDL	Flat	Unsupervised	1
F-SDL	Flat	Supervised	1
F-MDL	Flat	Supervised	K
H-UDL	Hierarchical	Unsupervised	1
ML-DDL	Hierarchical	Supervised	$\lvert\mathcal{I}\rvert$

Label Struct.: the structure of label space; Dict. Property: the scheme of dictionary learning; # of Dict.: the number of trained dictionaries, where $|\mathcal{I}| \ll K$.

A. Basic Setup

1) Baselines: ML-DDL has two components: hierarchical structure and supervised dictionary learning. We compare it with the following baselines in different component combinations:

1. Flat structure + Unsupervised dictionary (F-UDL). The model is learnt based on a single unsupervised dictionary and the one-vs-all strategy [12].

2. Flat structure + One supervised dictionary (F-SDL). The model is learnt based on a common supervised dictionary and the one-vs-all strategy [17].

3. Flat structure + Multiple supervised dictionaries (F-MDL). Multiple supervised dictionaries are trained, and each class is associated with one dictionary [19]. The model is learnt based on the one-vs-all strategy.

4. Hierarchical structure + Unsupervised dictionary (H-UDL). The model is learnt based on a single unsupervised dictionary. We use a similar hierarchical discrimination criterion as that in [44], and train the model with the package provided by [58].

To facilitate reading, we list the methods according to their configurations in Table II. Supposing that K categories have been organized as a hierarchy $\mathcal{H}$, in which the set of internal nodes is denoted by $\mathcal{I}$.

2) Configurations: The prior hierarchy is built according to [23] by recursively clustering class labels with the standard spectral clustering method [59]. Each image is resized such that the longer side is no more than 300 pixels and no less than 100 pixels. For basic feature, we use 128-dimensional SIFT [13] from patches as input descriptors. Moreover, we employ another two types of local descriptors, 100-dim block color histogram [55] and 30-dim self-similarity [60] for the evaluation of ML-DDL with multi-feature channels in Section VI-F. The size of patch side is set to $16+8l$ pixels,

TABLE III
TRAINING COST (C_{train}, IN HOURS), TESTING COST (C_{test}, IN HOURS) AND ACCURACY (Acc%) ON SUN397 AND IMAGENET200

Method	SUN397			ImageNet200		
	C_{train}	C_{test}	Acc%	C_{train}	C_{test}	Acc%
F-UDL	16.0	5.5	23.6%	30.5	41.5	27.5%
F-SDL	>800	5.5	23.8%	>1500	41.5	27.9%
F-MDL	>300	>300	24.2%	>600	>1000	28.2%
H-UDL	12.5	5.5	22.1%	28.0	41.5	29.0%
ML-DDL	27.0	3.0	24.8%	52.5	23.0	33.9%

where l denotes the layer index (the root node corresponds to $l = 0$). The stride of descriptor is set to be half of the patch size. For a fair comparison, the descriptors extracted with multiple patch scales are used as input for all methods. The sparsity parameter μ in coding is set to 0.12, and the regularization parameter λ is set to 0.1.

3) Evaluation Criteria: We describe the evaluation criteria in terms of accuracy and efficiency. The classification accuracy is obtained by averaging the per-class accuracy. Moreover, for hierarchical models we also evaluate the accuracy at different layers, i.e., Acc_l in Table VI. Acc_l can be computed by averaging the per-node accuracy when predicting at current layer l. For efficiency evaluation, the timing is based on a single core of an 8-core Intel Xeon 3.20 GHz server. We calculate the time cost in hours.

B. Evaluation on SUN397

In this section, we evaluate ML-DDL on the SUN397 dataset. We use all 397 categories and follow the protocol in [55]. The data has been partitioned. In each partition there are 50 training images and 50 testing images per class. We obtain the results by averaging the performance on all the partitions. A three-layer (excluding the root node) tree is established as the prior hierarchy. With respect to dictionary size, we choose 512 for the specific dictionaries at different layers in ML-DDL. The size in F-UDL, F-SDL and H-UDL is set to 1024. As each class is associated with one dictionary in F-MDL, we adopt a smaller size for each dictionary, which is set to 256.

1) Efficiency Evaluation: The time cost of a method on training and testing is a critical issue when dealing with many classes. We evaluate the computation cost of these methods on both training and testing. The result is shown in Table III.

The main time cost on training consists of three parts: dictionary learning, image-level representation generation and classification model learning. For unsupervised dictionary learning methods, the three steps perform sequentially. The dictionary is learnt based on a certain set of descriptors sampled from data, and the training time on dictionary is much less than representation generation and classification model learning. Different discrimination criteria lead to the different training costs of F-UDL and H-UDL. On the other hand, with respect to supervised dictionary learning, a joint learning process of dictionary and classification model is adopted. Meanwhile, image-level representations are generated during the process. The methods usually suffer from high time complexity on training due to recomputing the representations and dictionary updating. The learning of each dictionary in F-MDL utilizes all data, and the number of dictionaries equals to the class number, which is much larger than the number of dictionaries in ML-DDL. Moreover, learning a single larger supervised dictionary in F-SDL costs more time than learning a smaller dictionary per class in F-MDL, which is consistent with the statement in [19]. Compared with F-MDL and F-SDL, ML-DDL has much less training cost. By virtue of hierarchy, the learning of ML-DDL is decomposed to multiple sub-problems, one for each internal node. Each dictionary and the associated classification model is optimized in one sub-problem, using a subset of the whole data.

Considering testing time, the computation of representation dominates the cost. Multi-scale descriptors are encoded by one dictionary for each test sample in F-UDL, H-UDL and F-SDL. Although ML-DDL seems to have more dictionaries compared with these methods, it is worth noting that ML-DDL achieves the best efficiency among all the methods. As a matter of fact, each test sample traverses L (the depth of tree) dictionaries from the root to a leaf node in ML-DDL, and each dictionary is employed to encode the descriptors at a certain scale, with a smaller dictionary size. With respect to F-MDL, each sample should be computed through all the dictionaries, resulting in drastic increase in the time cost.

2) Classification Accuracy Evaluation: We summarize the accuracy of different methods in Table III. Compared with the result of SIFT-based algorithm using K-means quantization (21.5% reported in [55]), these methods take advantage of sparse coding to achieve better results. The accuracy of ML-DDL is better than other supervised dictionary learning methods (F-MDL and F-SDL). Moreover, the superiority of ML-DDL over H-UDL is clear. In contrast, supervised dictionary learning methods with flat structure, i.e., F-SDL and F-MDL, have limited increase compared with the unsupervised dictionary learning method F-UDL.

C. Evaluation on ImageNet200

ImageNet is a large scale dataset, where the classes are organized based on WordNet [61]. Different from SUN397, the sample distribution is imbalanced, i.e., some classes contain lots of training images, and a number of classes contain few data. We evaluate ML-DDL with such a more realistic setting. We use a subset, ImageNet200, for evaluation.

TABLE IV
SOME CLASS EXAMPLES IN IMAGENET200

Class	 lion	 steeplechaser	 flagship	 iceberg	 skeleton	 cup	 endoscope	 paper
# of Sample	1795	312	48	1050	76	1307	285	1352
$\text{Acc}_f(\%)$	23.2	20.7	6.3	35.0	3.9	19.8	51.6	20.1
$\text{Acc}_h(\%)$	40.0	29.3	12.5	47.4	11.8	30.3	39.5	11.4
Incre(%)	16.8	8.6	6.2	12.4	7.9	10.5	-12.1	-8.7

# of Sample denotes the total number of samples in the class; $\text{Acc}_f(\%)$ and $\text{Acc}_h(\%)$ denotes the accuracy of F-MDL and ML-DDL on the class, respectively. Incre(%) denotes the increment of ML-DDL over F-MDL.

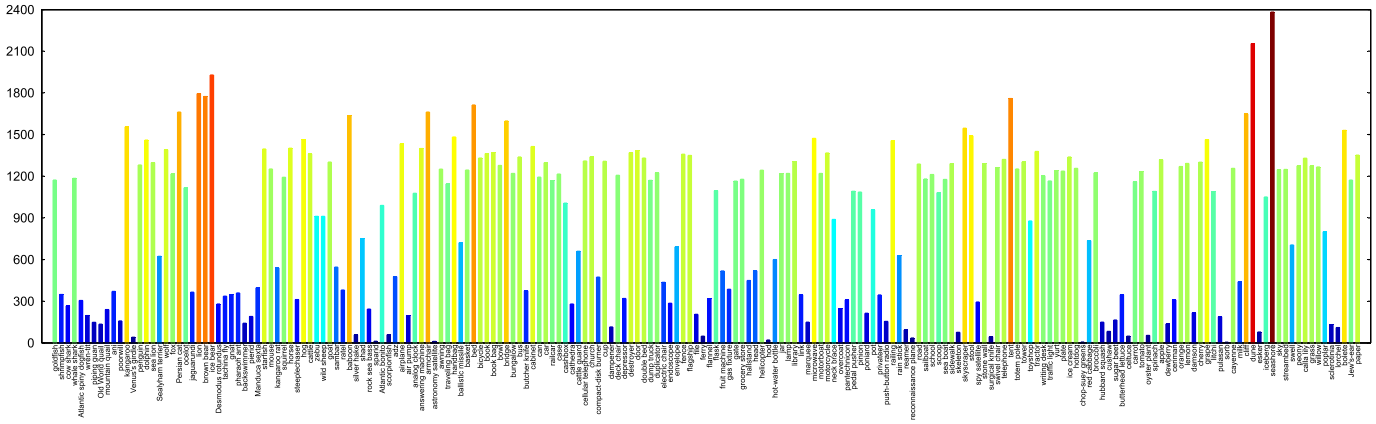


Fig. 3. Numbers of images in the 200 classes selected from ImageNet.

ImageNet200 contains randomly chosen 200 categories covering a wide range of semantic domains, such as *animal*, *tools*, *vehicle* and *construction*, as shown in Fig. 3. The number of images in each class is quite different, varying from several to thousands. The average number of samples in each class is about 900. We split the samples of each class into two sets: one-third data are used for training and the others are for testing. We generate a two-layer (excluding the root node) prior structure for evaluation. The size is set to 512 for each specific dictionary in ML-DDL. The dictionary size is set to 1024 for F-UDL, F-SDL and H-UDL, and 256 for each dictionary in F-MDL.

1) Efficiency Evaluation: We evaluate the training and testing cost of different methods. The results in Table III illustrate the significant advantage of ML-DDL compared with other supervised dictionary learning methods. Although the training time of ML-DDL is more than those of F-UDL and H-UDL, the testing time of ML-DDL is much less than theirs. We have to emphasize that in practice the efficiency in testing is relatively more important than that in training. So we can conclude that the overall time cost of ML-DDL is advantageous over other methods in comparison.

2) Classification Accuracy Evaluation: We also evaluate the classification accuracy of these methods, as shown in Table III. The low accuracy of flat structure methods indicates that the imbalanced distribution of samples incurs difficulty when classifying a large number of classes with a flat label space. By virtue of category relation, the classes with few training samples borrow the strength of related classes (i.e., the siblings) at higher layers. Accordingly, hierarchical models consistently outperforms flat structure methods on ImageNet200. Moreover, ML-DDL effectively explores and transfers information via supervised dictionary learning, thus achieves better accuracy than H-UDL does.

Compared with the flat structure method F-MDL, which achieves better result than other flat structure methods, the accuracy of many classes has increased in ML-DDL. We take some classes as examples, which are shown in Table IV. The results demonstrate that the classification can be improved regardless of the number of samples, such as “lion”, “steeplechaser”, “flagship” and “skeleton”. These classes take advantage of hierarchical structure (i.e., groups) to narrow the choices of predictions at each layer. In particular, the classes with few samples (e.g., “flagship” and “skeleton”) leverage the knowledge from siblings (i.e., {“ferry”,“sea boat”} and {“fence”,“gate”}) which have many samples, to improve their accuracy. Besides, for many classes ML-DDL is helpful to resolve the confusion on appearance via the hierarchical structure. For example, varied patterns (e.g., logo, text or figures) can be observed in “cup” images. However, most of these patterns may be ineffective, or even unfavorable, to represent class “cup”, and hence lead to low accuracy even if having many training samples. ML-DDL leverages its related classes (e.g., “bowl”,“jar” and “pot”) to improve its accuracy (+10.5%). On the other hand, hierarchy may also lead to negative transfer, i.e., hurt the accuracy on

TABLE V
COMPARISON OF DIFFERENT METHODS ON THE IMAGENET1K DATASET

Method	Features	Coding	Feat. Dim.	Acc%
NEC	LBP, HOG	LCC, Super-vector	262,144	52.9
Fisher Vector	SIFT	Fisher Vector	131,072	45.7
MC	GIST, HoG, SSIM, SIFT	MC feature	15,458	36.4
JDL	SIFT	Sparse Coding	40,000	38.9
ML-DDL	SIFT	Sparse Coding	40,960	40.3

some classes. For example, distinctive appearance and large intra-class similarity lead to high distinguishability of class "endoscope", but group structure introduces extra noise from its siblings.

D. Evaluation on ImageNet1K

In order to verify the scalability of ML-DDL on large scale data, we further evaluate it on the ImageNet1K dataset used in ILSVRC2010 [57]. The dataset contains 1.2M images from 1000 categories for training, 50K images for validation and 150K images for testing. First, we compare ML-DDL with some state-of-the-art methods. Second, we assess the performance of ML-DDL with different prior hierarchies.

1) Comparison With State-of-the-Art Results: We compare ML-DDL with some state-of-the-art methods on the ILSVRC2010 dataset, including JDL [31], Fisher Vector [6], the method of NEC [62] and the Meta-Class feature (MC) [63]. The comparison among different methods is shown in Table V. The configuration of the baselines follows that in [31]. As the state-of-the-art sparse coding method, JDL [31] is only applicable to a two-layer hierarchy. For a fair comparison, we also exploit a two-layer hierarchy T_2, and use single feature, SIFT, as the local descriptors for ML-DDL. Each specific dictionary is set to have 2048 atoms in ML-DDL. We train ML-DDL in about five days on the 8-core server.

JDL is a discriminative dictionary learning method, which is more connected with ML-DDL. Table V shows that ML-DDL achieves better accuracy than JDL and MC do. But it does not perform as well as the method of NEC and Fisher Vector. This is because the method of NEC and Fisher Vector apply other coding strategies to encode local descriptors. By taking advantage of higher dimensional features, they obtain better results.

Besides, compared with JDL, ML-DDL is more flexible and efficient in dealing with a large number of classes. Although JDL has not reported time cost in [31], the large number of learnt dictionaries (i.e., 1083, more than class number) and the two-stage learning of dictionary and classifiers lead to heavy computation cost in training and testing. In contrast, the number of learnt dictionaries in ML-DDL is determined by the number of internal nodes (here is 43), which is much smaller than class number. Moreover, the prediction of a sample is accompanied with only two dictionaries in ML-DDL.

2) Evaluation With Different Prior Hierarchies: ML-DDL aims at capturing discriminative information residing in the category hierarchy, which plays an important role in information exploiting and transfer among classes. To analyze the effectiveness of ML-DDL with different priors, we apply another two hierarchies: T_3 and T_4, where the subscript denotes the depth of hierarchy. Due to the high computation cost of F-SDL and F-MDL (shown in Table III), these two methods are intractable for ImageNet1K. Therefore, we only compare ML-DDL with hierarchical method H-UDL and flat structure method F-UDL. The dictionary size in these baselines is increased to 4096 in order to enhance the strength of representation. Considering that each internal node deals with a relatively small sub-problem when the tree is deep, we apply different configurations for ML-DDL according to different prior hierarchies. We set the specific dictionary sizes to (2048, 1024, 512) in T_3, and (1024, 1024, 512, 512) in T_4, from top to bottom layer.

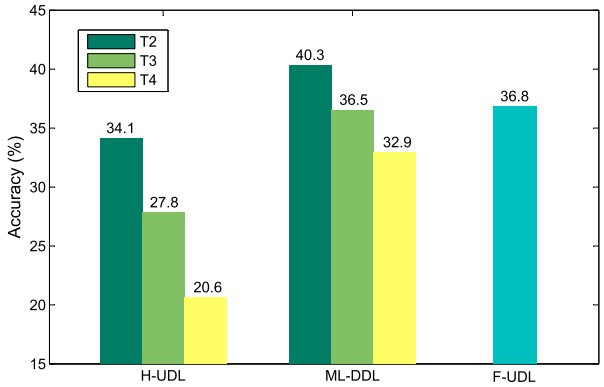


Fig. 4. Accuracy of the methods based on different hierarchies on ImageNet1K. T_L denotes the hierarchical structure with depth L.

As shown in Fig. 4, ML-DDL consistently outperforms H-UDL based on different prior hierarchies, and achieves competitive or better result than the flat structure model F-UDL. With respect to H-UDL, there is drastic change in accuracy when dealing with different hierarchies. In contrast, ML-DDL still achieves excellent results on different hierarchies. More importantly, when the hierarchy becomes deep, the superiority of ML-DDL over H-UDL becomes more salient. This further shows that ML-DDL is capable of appropriately capturing discriminative information by exploiting prior knowledge.

E. Effect of Hierarchical Discriminative Information and Dictionary Inheritance

In this section, we justify the effectiveness of two factors in ML-DDL, hierarchical discriminative dictionary learning and dictionary inheritance. We utilize a degraded version of

TABLE VI
ACCURACY (Acc_l%) AT DIFFERENT LAYERS OF HIERARCHY ON SUN397 AND IMAGENET200

Method	SUN397			ImageNet200	
	Acc_1	Acc_2	Acc_3	Acc_1	Acc_2
H-UDL	87.6	51.2	22.1	76.0	29.0
ML-DDL^0	89.3	52.1	23.0	78.3	31.6
ML-DDL	89.3	53.5	24.8	78.3	33.9

TABLE VII
THE COMPARISON ON ACCURACY (%) BETWEEN DICTIONARY LEARNING WITH SINGLE-SCALE DESCRIPTORS AND MULTI-SCALE DESCRIPTORS IN F-UDL AND ML-DDL

Dataset	F-UDL^s	F-UDL	ML-DDL^s	ML-DDL
SUN397	23.1	23.6	22.7	24.8
ImageNet200	27.0	27.5	31.5	33.9

ML-DDL named ML-DDL^0 as another baseline, which does not involve dictionary inheritance. In ML-DDL^0 the nodes at lower layers do not inherit the dictionaries from parents. We assess the accuracy of H-UDL, ML-DDL^0 and ML-DDL at different layers. The results are shown in Table VI.

With respect to hierarchical models, the accuracy decreases at lower layers, especially when reaching leaf nodes. Besides the misclassification at the current layer, the error at higher layers is also accumulated. In this regard, ML-DDL^0 consistently outperforms H-UDL, which demonstrates that the above problems can be alleviated with the aid of exploring hierarchical discriminative information.

The improvement of ML-DDL over ML-DDL^0 verifies the necessity of dictionary inheritance. The visual information captured from ancestor nodes is helpful for children. In the traditional sharing models [27], [42], the siblings inherit the common statistical information, which has little effect on the distinguishability among them. In contrast, ML-DDL makes better use of the inherited properties, which can be further weighted to improve the discrimination among siblings.

On the other hand, benefiting from dictionary inheritance, the image representations of low-layer nodes integrate multi-scale visual information. To investigate the effect of multi-scale information, we use another variation, ML-DDL^s, where the dictionaries are learnt to encode the descriptors at the same scale. The size is also set to 512 for each dictionary, and a 16×16 patch size is applied. For the standard unsupervised dictionary method F-UDL, we also test it with single-scale input, named F-UDL^s, and the results are shown in Table VII. Compared with single-scale input, F-UDL and ML-DDL both obtain improvements by using multi-scale descriptors as input. However, the improvement is marginal in F-UDL (no more than 0.5%), which is consistent with the analysis in [12]. The comparison between the improvements of ML-DDL and F-UDL shows that ML-DDL can make better use of multi-scale information.

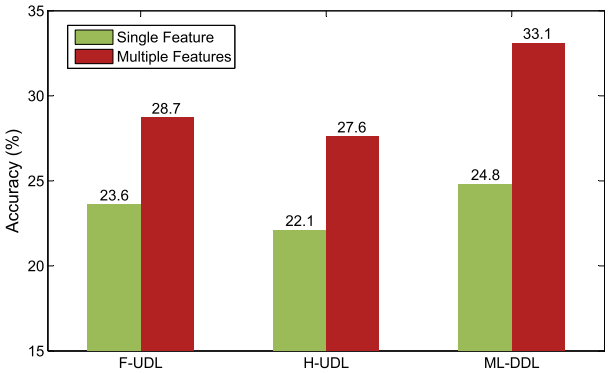


Fig. 5. Accuracy (%) Compared With Single-Feature and Multi-Feature Version of Three Methods on SUN397.

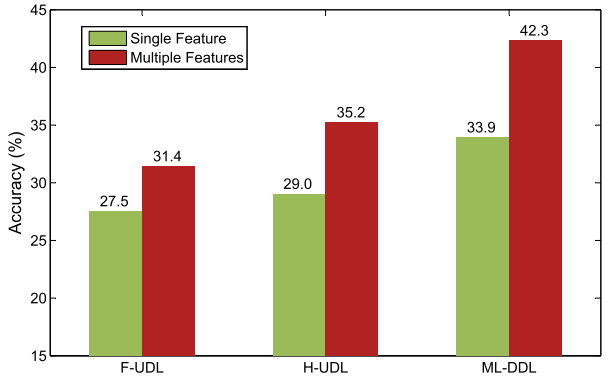


Fig. 6. Accuracy (%) compared with single-feature and multi-feature version of three methods on ImageNet200.

F. Effect of ML-DDL With Multi-Feature Channels

We have shown that ML-DDL is flexible in extending to multi-feature channels in Section IV-C. In this section, we investigate the effect of multi-feature dictionary learning.

Besides 128-dim SIFT, we employ another two types of local descriptors as input: 100-dim block color histogram [55] and 30-dim self-similarity [60]. With respect to multi-feature version of ML-DDL (MF-ML-DDL for short), each dictionary extends to multiple feature channels, and it can be regarded as a dictionary ensemble composed of three subdictionaries. The sizes of subdictionary are set to 512, 256 and 256, respectively. We compare MF-ML-DDL with multi-feature version of baselines (with prefix MF-): MF-F-UDL and MF-H-UDL. For MF-F-UDL and MF-H-UDL, multiple features are generated via respective unsupervised subdictionaries and then concatenated as the final representation. The dictionary sizes are 1024, 512 and 512, respectively.

The accuracy of the above three methods (including single-feature and multi-feature version) on SUN397 and ImageNet200 are shown in Fig. 5 and Fig. 6. Although the result on SUN397 is not as good as the one reported in [55], 38.0%, which is achieved by integrating more than ten types of features, MF-ML-DDL can obtain promising result (33.1%) with only three types of features.

Compared with the single-feature version (i.e., F-UDL, H-UDL and ML-DDL), all the methods with multi-feature channels obtain better accuracy, demonstrating that fusing the descriptive power of multiple complementary features is effective to enhance the performance of a model.

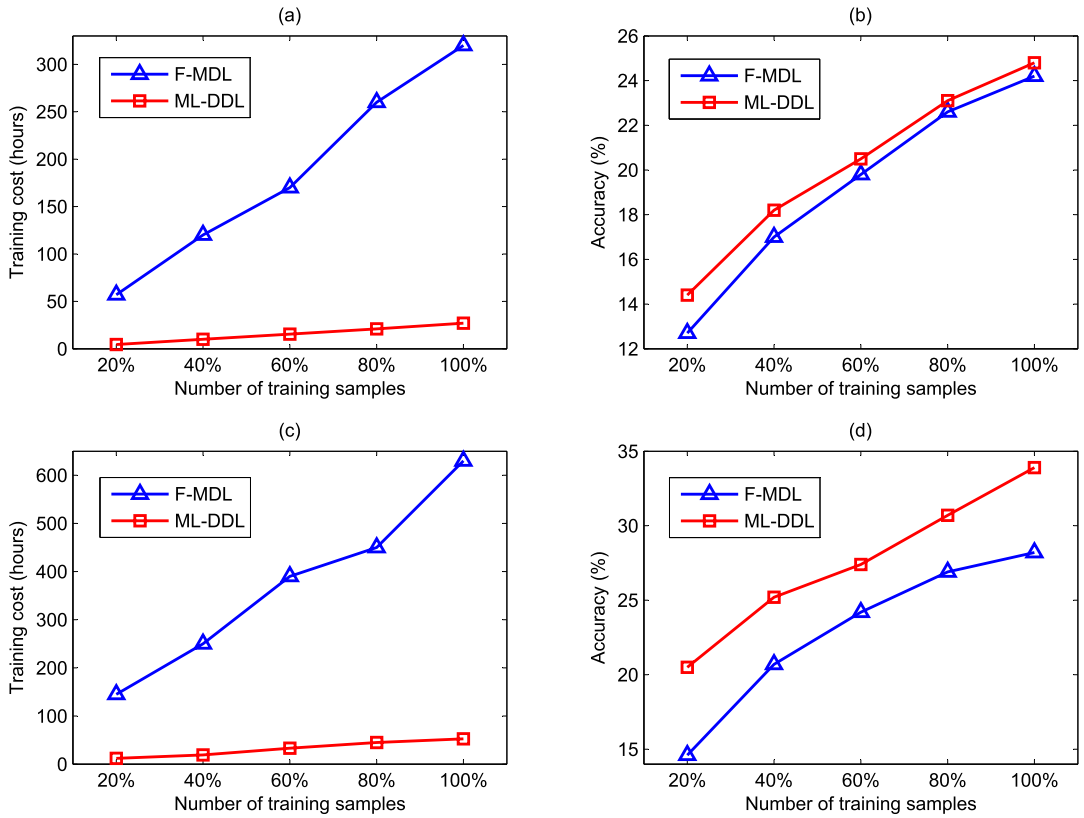


Fig. 7. Training cost (hours) and Accuracy (%) compared between ML-DDL and F-MDL with different sizes of training data. (a) Training cost on SUN397. (b) Accuracy on SUN397. (c) Training cost on ImageNet200. (d) Accuracy on ImageNet200.

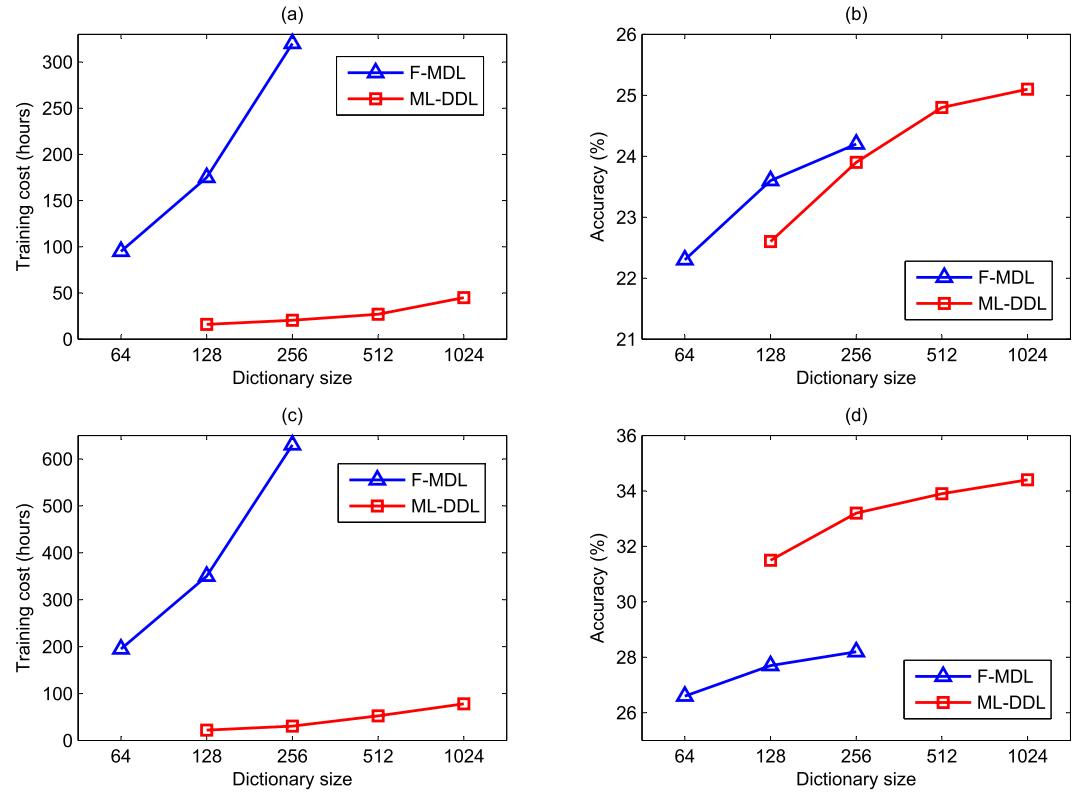


Fig. 8. Training cost (hours) and Accuracy (%) compared between ML-DDL and F-MDL with different dictionary sizes. (a) Training cost on SUN397. (b) Accuracy on SUN397. (c) Training cost on ImageNet200. (d) Accuracy on ImageNet200.

For hierarchical methods, different features tend to be exploited for the discrimination at different layers. Accordingly, significant improvement over F-UDL can be achieved. Moreover, MF-ML-DDL is clearly superior to MF-H-UDL on both datasets. In MF-ML-DDL, the subdictionaries can be adaptively tuned according to the training criterion.

TABLE VIII
THE COMPARISON ON ACCURACY (%) USING DIFFERENT POOLING STRATEGIES FOR TRAINING DICTIONARIES IN ML-DDL

Dataset	Average	Max
SUN397	23.5	24.8
ImageNet200	32.3	33.9

The flexible learning framework of ML-DDL ensures that discriminative information of multiple sources can be adequately extracted and exploited.

G. Experiment Revisit

In order to investigate ML-DDL thoroughly, we examine the effects of some factors such as pooling strategy, training data size and dictionary size on the algorithm in this section.

1) Pooling Strategy for Training Dictionary: When training dictionaries, we apply max spatial pooling, which has been demonstrated effectiveness in unsupervised dictionary learning methods [12]. To investigate its effect in this work, we compare it with another pooling strategy, average spatial pooling [32]. Experiments on SUN397 and ImageNet200 follow the same configuration in Table III except using different pooling strategies for training dictionaries. The results are shown in Table VIII. Spatial layout information is embedded in both two pooling strategies. Max spatial pooling produces better performance as shown, probably because it is robust to local spatial variations. Such property is helpful to capture discriminative information when training dictionaries.

2) Training Data Size: The size of training data plays an important role for the accuracy and training efficiency of system. For a comparison, we also evaluate the training cost and accuracy of discriminative dictionary learning method F-MDL, which achieves better performance than F-SDL. The set for testing is kept unchanged while the training samples per class are decimated with a fixed ratio. Particularly, each class is required to have at least two training samples in ImageNet200. The evaluation results on SUN397 and ImageNet200 are shown in Fig. 7.

Regarding the learning of discriminative dictionary, training cost is typically high. Decreasing the size of training data results in a faster convergence on optimization, i.e., lower training time cost. Compared with the experiments with total training samples, F-MDL achieves a much better efficiency with 20% data, however the time cost is still higher than the one of ML-DDL with total training samples. On the other hand, ML-DDL consistently outperforms F-MDL on accuracy although fewer training data leads to a lower accuracy. When increasing the size of training data, the generalization power of model can be improved. Both methods can substantially improve the accuracy accordingly. However, as the size of training data grows, the imbalance of sample distribution is more obvious in ImageNet200, which retards the increase of F-MDL. In contrast, ML-DDL shows a clearer advantage by transferring discriminative information in the hierarchy.

3) Dictionary Size: We also investigate the effect of dictionary size on the two methods ML-DDL and F-MDL. In the experiments on ML-DDL, we try four sizes, 128, 256, 512 and 1024, for the specific dictionaries. For each dictionary in F-MDL, we use three sizes, 64, 128 and 256, respectively.

The increase of dictionary size enriches the encoded properties, which enhances the discriminative power of feature representation. As shown in Fig. 8 (b) and (d), the accuracy of both methods can be improved with a larger dictionary, however the improvement is marginal when the dictionary size grows further. On the other hand, as dictionary size grows, more time cost is spent on representation computing and dictionary updating in each iteration, which is accumulated on total training cost. As shown in Fig. 8 (a) and (c), with respect to F-MDL, the training cost is intractable when the dictionary size is large. Conversely, ML-DDL can be developed with a much larger dictionary, thus shows a better scalability of model complexity over F-MDL.

VII. CONCLUSION

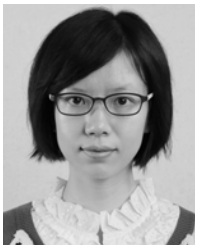
In this paper, we present a novel multi-level discriminative dictionary learning method, ML-DDL, and apply it to large scale image classification. Hierarchical dictionary learning and dictionary inheritance are exploited to encode multi-level discriminative information. The joint learning of dictionaries and associated model parameters helps to improve the performance of classification. Besides, ML-DDL is flexible in extending the dictionaries to multi-feature channels, which further enhances the accuracy. The experimental results demonstrate that ML-DDL can take advantage of category hierarchy to effectively capture discriminative information via the sparse coding technique, and is capable of dealing with large scale image classification.

Our method relies on a given hierarchy, which may not facilitate the information transfer among classes at the best. In the future, we will incorporate hierarchy learning/ construction into the framework.

REFERENCES

[1] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.
[2] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 27–38, Mar. 2011.
[3] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Proc. NIPS*, 2009, pp. 2223–2231.
[4] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE CVPR*, Jun. 2010, pp. 3360–3367.
[5] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image classification using super-vector coding of local image descriptors," in *Proc. ECCV*, 2010, pp. 141–154.
[6] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, 2013.
[7] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.
[8] B. Wen, S. Ravishankar, and Y. Bresler, "Structured overcomplete sparsifying transform learning with convergence guarantees and applications," *Int. J. Comput. Vis.*, Oct. 2014.
[9] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.
[10] A. Y. Yang, Z. Zhou, A. G. Balasubramanian, S. S. Sastry, and Y. Ma, "Fast ℓ_1-minimization algorithms for robust face recognition," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3234–3246, Aug. 2013.

[11] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model," *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 661–677, Feb. 2013.
[12] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE CVPR*, Jun. 2009, pp. 1794–1801.
[13] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE CVPR*, Jun. 2006, pp. 2169–2178.
[14] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[15] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.
[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Proc. NIPS*, 2008, pp. 1033–1040.
[17] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE CVPR*, Jun. 2010, pp. 2559–2566.
[18] J. Yang, K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *Proc. IEEE CVPR*, Jun. 2010, pp. 3517–3524.
[19] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.
[20] N. Zhou, Y. Shen, J. Peng, and J. Fan, "Learning inter-related visual dictionary for object recognition," in *Proc. IEEE CVPR*, Jun. 2012, pp. 3490–3497.
[21] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification," *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 209–232, 2014.
[22] G. Griffin and P. Perona, "Learning and using taxonomies for fast visual categorization," in *Proc. IEEE CVPR*, Jun. 2008, pp. 1–8.
[23] S. Bengio, J. Weston, and D. Grangier, "Label embedding trees for large multiclass task," in *Proc. NIPS*, 2010, pp. 163–171.
[24] J. Deng, S. Satheesh, A. C. Berg, and L. Fei-Fei, "Fast and balanced: Efficient label tree learning for large scale object recognition," in *Proc. NIPS*, 2011, pp. 567–575.
[25] L.-J. Li, C. Wang, Y. Lim, D. M. Blei, and L. Fei-Fei, "Building and using a semantivisual image hierarchy," in *Proc. IEEE CVPR*, Jun. 2010, pp. 3336–3343.
[26] T. Gao and D. Koller, "Discriminative learning of relaxed hierarchy for large-scale visual recognition," in *Proc. IEEE ICCV*, Nov. 2011, pp. 2072–2079.
[27] R. Salakhutdinov, A. Torralba, and J. Tenenbaum, "Learning to share visual appearance for multiclass object detection," in *Proc. IEEE CVPR*, Jun. 2011, pp. 1481–1488.
[28] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei, "What does classifying more than 10,000 image categories tell us?" in *Proc. ECCV*, 2010, pp. 71–84.
[29] T. Deselaers and V. Ferrari, "Visual and semantic similarity in ImageNet," in *Proc. IEEE CVPR*, Jun. 2011, pp. 1777–1784.
[30] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Unsupervised learning of hierarchical representations with convolutional deep belief networks," *Commun. ACM*, vol. 54, no. 10, pp. 95–103, 2011.
[31] N. Zhou and J. Fan, "Jointly learning visually correlated dictionaries for large-scale visual recognition applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 715–730, Apr. 2014.
[32] L. Shen, S. Wang, G. Sun, S. Jiang, and Q. Huang, "Multi-level discriminative dictionary learning towards hierarchical visual categorization," in *Proc. IEEE CVPR*, Jun. 2013, pp. 383–390.
[33] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. NIPS*, 2006, pp. 801–808.
[34] K. Yu, Y. Lin, and J. Lafferty, "Learning image representations from the pixel level via hierarchical sparse coding," in *Proc. IEEE CVPR*, Jun. 2011, pp. 1713–1720.
[35] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for hierarchical sparse coding," *J. Mach. Learn. Res.*, vol. 12, pp. 2297–2334, Feb. 2011.
[36] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Proc. NIPS*, 2006, pp. 609–616.
[37] L. Yang, R. Jin, R. Sukthankar, and F. Jurie, "Unifying discriminative visual codebook generation with classifier training for object category recognition," in *Proc. IEEE CVPR*, Jun. 2008, pp. 1–8.
[38] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proc. IEEE CVPR*, Jun. 2008, pp. 1–8.
[39] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE CVPR*, Jun. 2010, pp. 3501–3508.
[40] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proc. IEEE ICCV*, Nov. 2011, pp. 543–550.
[41] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *Proc. IEEE CVPR*, Jun. 2011, pp. 1697–1704.
[42] O. Dekel, J. Keshet, and Y. Singer, "Large margin hierarchical classification," in *Proc. ICML*, 2004, p. 27.
[43] S. Gopal, Y. Yang, B. Bai, and A. Niculescu-Mizil, "Bayesian models for large-scale hierarchical classification," in *Proc. NIPS*, 2012, pp. 2420–2428.
[44] D. Zhou, L. Xiao, and M. Wu, "Hierarchical classification via orthogonal transfer," in *Proc. ICML*, 2011, pp. 801–808.
[45] S. J. Hwang, K. Grauman, and F. Sha, "Learning a tree of metrics with disjoint visual features," in *Proc. NIPS*, 2011, pp. 621–629.
[46] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Incremental algorithms for hierarchical classification," *J. Mach. Learn. Res.*, vol. 7, pp. 31–54, Jan. 2006.
[47] B. Zhao, L. Fei-Fei, and E. Xing, "Large-scale category structure aware image categorization," in *Proc. NIPS*, 2011, pp. 1251–1259.
[48] V. Vural and J. G. Dy, "A hierarchical method for multi-class support vector machines," in *Proc. ICML*, 2004, p. 105.
[49] B. Liu, F. Sadeghi, M. Tappen, O. Shamir, and C. Liu, "Probabilistic label trees for efficient large scale image classification," in *Proc. IEEE CVPR*, Jun. 2013, pp. 843–850.
[50] M. J. Choi, A. Torralba, and A. S. Willsky, "A tree-based context model for object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 240–252, Feb. 2012.
[51] A. Zien and C. S. Ong, "Multiclass multiple kernel learning," in *Proc. 24th ICML*, 2007, pp. 1191–1198.
[52] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *Proc. IEEE 12th ICCV*, Sep./Oct. 2009, pp. 606–613.
[53] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proc. IEEE 12th ICCV*, Sep./Oct. 2009, pp. 221–228.
[54] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*. New York, NY, USA: Springer-Verlag, 2006.
[55] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE CVPR*, Jun. 2010, pp. 3485–3492.
[56] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE CVPR*, Jun. 2009, pp. 248–255.
[57] A. Berg, J. Deng, and L. Fei-Fei. (2010). *Large Scale Visual Recognition Challenge 2010*. [Online]. Available: http://image-net.org/challenges/LSVRC/2010/index
[58] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural SVMs," *Mach. Learn.*, vol. 77, no. 1, pp. 27–59, 2009.
[59] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[60] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. IEEE CVPR*, Jun. 2007, pp. 1–8.
[61] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press, 1998.
[62] Y. Lin *et al.*, "Large-scale image classification: Fast feature extraction and SVM training," in *Proc. IEEE CVPR*, Jun. 2011, pp. 1689–1696.
[63] A. Bergamo and L. Torresani, "Meta-class features for large-scale object categorization on a budget," in *Proc. IEEE CVPR*, Jun. 2012, pp. 3085–3092.



Li Shen received the B.S. degree from Nankai University, in 2009. She is currently pursuing the Ph.D. degree with the School of Computer and Control Engineering, University of Chinese Academy of Sciences. Her research interests include object recognition, feature learning, and transfer learning.

Gang Sun received the B.S. degree from Nankai University, in 2009. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. His research interests include computer vision and deep learning.

Qingming Huang received the bachelor's degree in computer science and the Ph.D. degree in computer engineering from the Harbin Institute of Technology, China, in 1988 and 1994, respectively. He was a Post-Doctoral Fellow with the National University of Singapore from 1995 to 1996, and served as a member of the Research Staff with the Institute for Infocomm Research, Singapore, from 1996 to 2002. He visited the University of Toronto as a Visiting Fellow in 2000. He joined the Chinese Academy of Sciences as a Professor under the Science100 Talent Plan in 2003, and was granted by the China National Funds for Distinguished Young Scientist in 2010. He is currently a Professor with the University of Chinese Academy of Sciences and an Adjunct Research Professor with the Institute of Computing Technology, Chinese Academy of Sciences. He has authored over 300 academic papers in prestigious international journals, including the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MULTIMEDIA, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and top-level conferences, such as ACM Multimedia, ICCV, CVPR, and ECCV. He holds 30 patents in U.S., Singapore, and China. His research area includes multimedia computing, pattern recognition, machine learning, and computer vision. He served as the Program Chair and organization/TPC members in various well-known international conferences, including ACM Multimedia, ICCV, ICME, and PSIVT.

Shuhui Wang received the B.S. degree in electronics engineering from Tsinghua University, Beijing, China, in 2006, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2012. He is currently an Assistant Professor with the Institute of Computing Technology, Chinese Academy of Sciences. He is also with the Key Laboratory of Intelligent Information Processing, Chinese Academy of Sciences. His research interests include large-scale Web data mining, visual semantic analysis, and machine learning.

Zhouchen Lin (M'00–SM'08) received the Ph.D. degree in applied mathematics from Peking University, in 2000. He was a Guest Professor with Shanghai Jiao Tong University, Beijing Jiaotong University, and Southeast University. He was also a Guest Researcher with the Institute of Computing Technology, Chinese Academic of Sciences. He is currently a Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. He is also the Chair Professor with Northeast Normal University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He was an Area Chair of CVPR 2014. He is an Area Chair of ICCV 2015, NIPS 2015, and AAAI 2016. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the *International Journal of Computer Vision*.

Enhua Wu (M'97) received the B.Sc. degree from Tsinghua University, Beijing, in 1970, and the Ph.D. degree from the University of Manchester, U.K., in 1984. He has been with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, since 1985, and has been a Full Professor with the University of Macau since 1997. His research interests include realistic image synthesis, physically based simulation, and virtual reality. He is a member of the Association for Computing Machinery, and a fellow of the China Computer Federation. He has been invited to chair or co-chair various conferences or program committees, including Pacific Graphics 2005, CASA 2006, ACM VRST 2010, 2013, and 2015, and WSCG 2012, and as a Keynote Speaker of CyberWorlds 2006, ACM VRST 2010 and 2011, and WSCG 2012. He has been the Associate Editor-in-Chief of the *Journal of Computer Science and Technology* since 1995, and the Editorial Board Member of *The Visual Computer*, *Computer Animation and Virtual Worlds*, the *International Journal of Image and Graphics*, the *International Journal of Virtual Reality*, and the *International Journal of Software and Informatics*.