Category Specific Dictionary Learning for Attribute Specific Feature Selection

Wei Wang, Yan Yan, Stefan Winkler, *Senior Member, IEEE*, and Nicu Sebe, *Senior Member, IEEE*

***Abstract*— Attributes, as mid-level features, have demonstrated great potential in visual recognition tasks due to their excellent propagation capability through different categories. However, existing attribute learning methods are prone to learning the correlated attributes. To discover the genuine attribute specific features, many feature selection methods have been proposed. However, these feature selection methods are implemented at the level of raw features that might be very noisy, and these methods usually fail to consider the structural information in the feature space. To address this issue, in this paper, we propose a label constrained dictionary learning approach combined with a multilayer filter. The feature selection is implemented at dictionary level, which can better preserve the structural information. The label constrained dictionary learning suppresses the intra-class noise by encouraging the sparse representations of intra-class samples to lie close to their center. A multilayer filter is developed to discover the representative and robust attribute specific bases. The attribute specific bases are only shared among the positive samples or the negative samples. The experiments on the challenging Animals with Attributes data set and the SUN attribute data set demonstrate the effectiveness of our proposed method.**

***Index Terms*— Attribute learning, dictionary learning, dictionary bases.**

I. Introduction

THERE exist numerous object categories in the real world. In order to recognize the various objects and scenes, many machine learning approaches have been proposed. Current machine learning approaches heavily rely on the sufficiency of training data. However, the labeled data are often time-consuming and expensive to obtain. Besides, how to effectively annotate images and videos is still an open problem. In order to leverage the knowledge of annotated images to classify novel objects, visual attributes were proposed [1]. Visual attributes are mid-level descriptors which bridge the low-level features and high-level concepts. Various attributes are proposed for different applications. For example, attributes can be divided into binary attributes and relative attributes. The value of a binary attribute is either one or zero, while the value of a relative attribute is continuous. There are also semantic attributes and discriminative attributes. The semantic attributes have semantic meanings assigned to them while discriminative attributes do not have exact semantic meanings.

Attributes are used to describe the characteristics and quality of an object or scene, such as materials, appearances, and functions. Attributes can provide a more detailed description of an image [2], [3] and can make key-word based image search feasible (e.g., *young asian men with glasses*). Besides, attributes are also composable, they can be combined for different specificities, i.e., a consumer might want to find *high-heeled shiny* shoes. The most important property of attributes, such as color and shape, is that they can be transferred among different object categories. Zero-shot learning [4] is proposed based on this property. First, attribute classifiers are pre-learned from their related objects. Then the target object can be recognized based on its binary attribute representation, which requires no training examples. The attribute representation is a binary vector whose elements are either one or zero, indicating the presence or absence of a specific attribute [5]. The binary attributes can efficiently split the image space [6]. k binary attributes can split images into 2^k space. In addition, abnormality prediction can be achieved [7] by checking the absence of typical attributes or the presence of atypical attributes. However, the binary classifiers for attributes fail in capturing the relative strength of attributes between images. In order to capture more generative semantic relationships, relative attributes were introduced by Parikh and Grauman [8]. A ranking function is learned for each attribute whose output is a continuous score denoting the strength of attributes in an image. With the help of relative attributes, we can describe images relative to other images by comparing their attribute scores. A more recent study showed that the performance of relative attribute ranking functions can be improved by using local parts that are shared through categories instead of using global features [9], [10].

In most situations, attributes are predefined with semantic meanings. Attribute vocabulary can be manually designed, such as the 'Animal with Attributes' dataset [11] where 85 binary attributes about 50 animal classes are defined. However, human defined attributes might be insufficient and not discriminative, especially for the categories which are not well studied by linguists. To tackle this problem, Parikh and Grauman [12] proposed augmenting the vocabulary actively to ensure that the new attributes can be inter-class discriminative. The rich web data can also be utilized to mine attributes, which requires no human annotators.

Manuscript received July 9, 2015; revised November 25, 2015 and January 18, 2016; accepted January 19, 2016. Date of publication January 28, 2016; date of current version February 12, 2016. This work was supported in part by the xLiMe European Project and in part by the research grant for the Human-Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jing-Ming Guo.

W. Wang, Y. Yan, and N. Sebe are with the Department of Information Engineering and Computer Science, University of Trento, Trento 38123, Italy (e-mail: wei.wang@unitn.it; yan@disi.unitn.it; sebe@disi.unitn.it).

S. Winkler is with the Advanced Digital Sciences Center, Singapore 138632 (e-mail: stefan.winkler@adsc.com.sg).

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.2016.2523340

Berg *et al.* [13] proposed mining attribute vocabulary automatically from web images and noisy text descriptions. They also demonstrated that some attributes can be localized, i.e., attributes can be characterized into local or global ones. As the localized attributes can provide fine-grained information, they are more discriminative when the object categories are quite close to each other (e.g. bird species recognition). A local attribute discovery model was introduced by Duan *et al.* [14] to determine a local attribute vocabulary. In most situations, attributes are defined prior to learning their corresponding statistical models. We can also learn the models first, and then decide whether to assign semantic meanings to the learned models. For example, some discriminative attributes [5] without semantic meanings are proposed for object recognition. Thus, attributes do not have to be associated with semantic meanings.

Current attribute learning methods usually map the low-level features directly to attributes. The dimension of low-level feature vector is usually very high because of the concatenation of various features, such as SIFT, Color SIFT and HOG. Jayaraman *et al.* [15] pointed out that the performance of attribute classifiers could be improved through feature selection because of the intrinsic mappings between attributes and features. Take color attributes (*red, green, yellow, etc.*) for example, the color attributes can be better trained on the dimensions corresponding to color histogram bins, wheres texture attributes (*furry, silky, etc.*) prefer texture features.

Most works perform feature selection by adding different regularizers into the loss function to encourage sparsity selection of features, and the correlation between attributes is considered simultaneously [5], [15], [16]. For instance, l_1-norm encourages feature competition among groups, l_2-norm encourages feature sharing among groups, and $l_{2,1}$-norm encourages intra-group feature sharing and inter-group competition. Regardless of regularizer types, the underlying intuition remains the same, i.e., encourage the semantically close attributes to share similar feature dimensions. The semantic correlation is either measured according to the semantic distance mined from the web, e.g., using WordNet [17], or from attributes' co-occurrence probability as proposed by Han *et al.* [16]. However, it is hard to judge to what extent the visual appearance similarity can be reflected by semantic closeness, and there is no guarantee that the semantically close attributes are visually similar. For example, the semantic distance between *orange* and *apple* is 2.25 and 0.69 between *orange* and *mandarin*, which are calculated based on the Leacock-Chodorow similarity measurement from WordNet [17]. However, we could not say that *orange* is visually more similar to *apple* than *mandarin*. In fact, *orange* should be more visually similar to *mandarin* as they have the same shape and color. Furthermore, the raw features might be very noisy and feature selection [18] over the raw features discards the structure information as each feature dimension is treated independently.

To address this issue, we propose a novel framework which consists of a label constrained dictionary learning module and a multilayer filter to perform basis selection. Fig.1 shows the overview of the introduced framework. Different from the conventional methods which perform feature selection over the raw features, we adopt a multilayer filter to do feature selection at the dictionary level, as a dictionary is expected to capture the higher-level structure of images [19]. First, a label constrained dictionary is constructed by suppressing the intra-class training data. Second, we design a multilayer filter to perform basis selection for each attribute independently. The basis is regarded as attribute specific basis if only the positive or only the negative examples have large and stable distribution over it. The larger the distribution is, the more representative the basis is. The smaller the standard deviation is, the more robust the basis is. Therefore, in the multilayer filter, two filters are designed for attribute specific basis selection, namely, μ-Filter and σ-Filter. The μ-Filter selects the representative bases, and the σ-Filter select the robust bases from the representative bases. Common bases are marked if both the positive and negative examples have large distribution over them. The common bases are only used for the reconstruction while the attribute specific bases are used both for image reconstruction and attribute classifier learning. Finally, the attributes of an image are predicted by a set of linear SVM classifiers with its projection over the attribute specific bases. To sum up, this paper makes the following contributions:

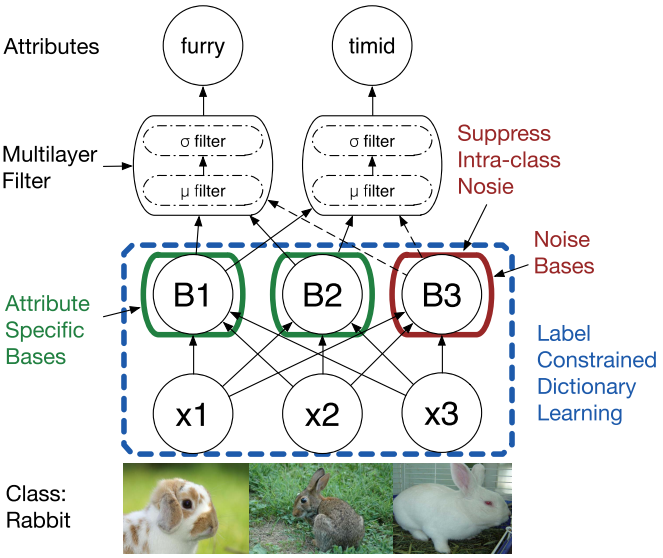


Fig. 1. **Overview of the Framework.** The label constrained dictionary learning module forces the dictionary to focus on learning the shared attribute specific bases by penalizing the intra-class variance. Then the mutilayer filter helps discover the representative and robust bases for each attribute.

- A novel label constrained dictionary learning method is proposed which suppresses intra-class noise and encourages the projections of intra-class training data to lie close by.
- A multilayer filter is designed for dictionary basis selection. Two filters, namely μ-Filter and σ-Filter are designed to select the robust and representative bases for each attribute.

This work is the extension of our previous work [3]. The paper is organized as follows. Section 2 reviews related work. Section 3 introduces our proposed framework. Experiments are described in Section 4, while Section 5 concludes this paper.

II. Related Work

In this section, we review the related work on attribute learning, feature selection and dictionary learning.

A. Attribute Learning

Attributes are middle level features which are shared through categories. Human naturally describe visual concepts with attributes. For instance, when we describe a person, we might say that he is a male, has short hair, and wear jeans. We also recognize objects or scenes through their attributes. For example, zebra has stripes. Recent studies revealed that the high performance of convolutional networks is ascribed to the attribute centric nodes within the net [20], and weakly supervised convolutional neural networks works well for attribute detection [10]. Besides, attributes usually provide more details of an image. In some situations, people may be interested not only in the object categories (*e.g., cat, dog, bike*), but also in the detailed information (*e.g., is silky, has legs, is cute*) of an image. In order to describe images with detailed information, Farhadi *et al.* [21] proposed describing an image based on semantic triples *<object, action, scene>*. The semantic triple links an image to a descriptive sentence. However, the method in [21] heavily relies on the object and scene classifiers to generate triples. Han *et al.* [22] proposed a hierarchical tree-structured semantic unit to describe an image at different semantic levels (*attribute level, category level, etc*). Thus, even if the object or scene classifier is unavailable, some attribute level information could still be provided.

As attributes are shared through categories, they also have great potential in object recognition tasks [1], [23], [24]. Latent attributes are utilized to improve the performance of object classifiers by taking the object-attribute relationship into consideration [25], [26]. Wang and Mori [27] took a further step to improve object classification performance by employing the attribute-attribute relationship. Besides, attributes can help recognize object when no training data is available. Lampert *et al.* [4] proposed zero-shot learning to predict unseen objects based on its binary attribute representation. Parikh and Grauman [8] improved the performance of zero-shot learning by utilizing relative attributes. Relative attributes can also be used to benefit interactive image search [28]. Based on the relative ranking scores, the system is enabled to adjust the strength of attributes to meet users' preferences. For active learning, attributes can propagate the impact of annotations through the entire model. Relative attributes can accelerate discriminative learning with few examples [29]–[31] as the mistake learned from one image can be transferred to many other images. For example, when the learner considers an image to be too open to be a forest, all other images more open than the current one will be filtered out. Attributes are also successfully applied into action recognition [32]–[34] and event detection [35]. Since attributes have wide applications, the performance of attribute classifiers are crucial.

B. Feature Selection for Attributes

There exist many different attribute groups, such as person-related attributes (*e.g., is male, has hat, has glasses*), scene attributes (*e.g., trees, clouds, leaves*) and animal attributes. In animal attributes group, there are also sub-groups, such as textures (*e.g., stripes, furry, spots*), part-of-body (*horns, claws, tusks*) and colors (*black, white, blue*). Jayaraman *et al.* [15] pointed out that the attribute classifiers would have different performances when different types of features were used because of the intrinsic relations between attributes and feature types.

The conventional methods learn attribute classifiers by mapping all the low-level raw features directly to each semantic attribute independently. However, many attributes are strongly correlated through the object categories. For example, most objects that *have wheels* are *made of metal*. Then when we try to learn *has wheel*, we may accidentally learn *made of metal*. To solve the correlation problem, various feature selection techniques are developed, most of which are implemented by integrating regularizers into the loss function. The underlying intuition behind feature selection is that only a portion of feature dimensions defines an attribute.

Thus, feature selection is an important process to improve the performance of attribute classifiers. Many works implement feature selection directly on the low-level raw features by using different regularizers, such as l_1-norm combined with l_2-norm, $l_{2,1}$-norm [15], [36], or $l_{2,p}$-norm, to encourage intra-group feature sharing and inter-group feature competition, as well as different loss functions, such as *linear regression* or *logistic regression*. Most regularizers are employed to get rid of the influence of attribute correlations. However, most current works revealed that the performance of attribute classifiers could be improved by harnessing attribute correlations rather than removing it [37], [38]. Han *et al.* [16] measured the attribute correlation through their co-occurrence probability among the object categories. A symmetric connected graph is constructed to represent the correlation between each pair of attributes, and the weights of the edges denote the quantified correlations. Then the correlation is put into l_1-norm regularizer. The relation between attributes does not have to be symmetric. For instance, the presence of necktie strongly indicates the presence of collar while the presence of collar does not indicate the presence of necktie. An asymmetric attribute correlation was defined in [39]. Usually, attribute correlation is regarded as an indicator of the feature sharing extent between attributes, and it is used to encourage feature sharing while feature competition is neglected. Regardless of the regularizer types, all these methods rely on regularizers to perform feature selection.

C. Dictionary Learning

Dictionary learning (or sparse coding) has been originally developed in order to explain the early visual processing in the brain [40]. An over-complete dictionary is built by minimizing the reconstruction error of the training samples where the learned bases are edges. Thus a more succinct and compact representation of an image can be obtained by its approximate decomposition over the dictionary bases. Based on sparse coding, hierarchical deep belief net model was proposed [41]. While learned bases in the first layer correspond to edges,

the learned bases in the second layer correspond to object components which are the combinations of edges. When multiple objects are used for training, the learned bases are the features shared across object classes. With the help of dictionary learning, the unlabeled data can be utilized to help supervised learning tasks, as usually the labeled data is very time-consuming to obtain. Dictionary learning allows us to use a small labeled training set to do a much better job at training classifiers [19].

More recently, dictionary learning has been applied to solve event detection [42], [43] and action detection problems [44]. Actions in videos are often atomic and largely defined by body poses, while events are composite and defined by objects and scenes. Qiu *et al.* [45] proposed learning a compact dictionary for actions, in which each basis is treated as an action attribute. In addition, dictionary learning can also be applied to image clustering tasks. Ramirez *et al.* [46] proposed learning multiple dictionaries for multiple categories to better embed the class information. The new data are assigned to the cluster whose dictionary can minimize the reconstruction error. Many different dictionary learning variants are studied by researchers, such as pairwise dictionary learning [47]. Another variant of dictionary learning was considered in [48] by integrating the manifold information and dictionary learning into the same framework.

Some work tried to bridge attributes and dictionary learning. Feng *et al.* [5] proposed an adaptive dictionary learning method for object recognition. Each image is reconstructed by a linear binary combination of dictionary bases, and each basis is regarded as one attribute. However, these attributes have no semantic meanings, and they can hardly be generalized to novel categories. Besides, the dictionary is usually trained by unlabeled data [19], [46] and a lot of noise bases that come from other unrelated objects are also learned. When labeled data are available, a label constrained dictionary can be learned, which is expected to encourage the sparse representation of intra-class data lie close by. In our work, this is implemented by a special regularizer and a modified Fast Iterative Soft-Thresholding Algorithm (FISTA) is adopted to solve the problem.

III. Label Constrained Dictionary Learning and Attribute Specific Basis Selection

In this section, we further discuss the underlying motivation of the proposed framework and present an overview of our approach. Then, our label constrained dictionary learning method is introduced. Finally, we elaborate the multilayer filter for basis selection.

A. Motivation and Overview

Most works employ feature selection to improve the performance of attribute classifiers. The underlying assumption is that an attribute is defined by a certain amount of feature dimensions. Thus, attributes are often learned jointly in a multi-task learning framework [49]–[52] in order to encourage feature sharing among correlated attributes.

However, feature selection discards the structural information of an image. Inspired by [5], we propose a label constrained dictionary learning method to decompose the images and the structural information is expected to be better preserved by dictionary bases. Then, we use the learned dictionary to reconstruct attributes. The motivation of our approach is that the objects containing the same attribute will have similar projections over the attribute specific bases. To help the dictionary focus on learning the shared attributes, label information is incorporated into the dictionary learning phase to minimize the intra-class noise. Qiu *et al.* [45] select a subset of dictionary to reconstruct all the actions and a better performance was yielded. Inspired by [45], we propose selecting attribute specific bases for attributes. Different from [45], we do basis selection for each attribute, and we implement it via a multilayer filter.

The proposed approach for training attribute classifiers is illustrated in Fig. 2. First, a label constrained dictionary is learned. This is implemented by penalizing the intra-class variance. Then the attribute specific bases are selected. Two types of attribute specific basis are considered: the basis that is only shared among the positive examples, and the one that is only shared among the negative examples. These two types of basis are named as *positive stimulus basis* which reflects what the attribute *has* and *negative stimulus basis* which reflects what the attribute does *not have*.

B. Label Constrained Dictionary Learning

The classical dictionary learning model which is aimed at minimizing reconstruction error and encouraging sparse projection is defined as follows:

$$\min_{\mathbf{D},\mathbf{C}} \|\mathbf{X}-\mathbf{D}\mathbf{C}\|_F^2+\lambda\sum_{i=1}^{N}\|\mathbf{c}_i\|_1$$

where the first term is in charge of minimizing the reconstruction error and the second term controls the sparsity. $\mathbf{X}\in\mathbb{R}^{M\times N}$, M is the dimension of training data, N is the number of data, $\mathbf{D}\in\mathbb{R}^{M\times L}$ is the dictionary, L is the number of bases, $\mathbf{C}\in\mathbb{R}^{L\times N}$ is the projection of training data, and $\mathbf{c_i}$ is the i-th column of $\mathbf{C}$, l_1-norm is the lasso constraint which encourages sparsity, and λ balances the trade-off between the reconstruction error and the sparsity.

Instead of learning multiple dictionaries, we learn one single label constrained dictionary for all categories. Thus, the shared attribute specific bases among the objects can be learned. To encourage the projections of intra-class data to lie close by, we propose the following optimization model:

$$\min_{\mathbf{D},\mathbf{C}} \|\mathbf{X}-\mathbf{D}\mathbf{C}\|_F^2+\alpha\sum_{i=1}^{N}\|\mathbf{c}_i\|_1+\beta\sum_{s=1}^{K}\|\mathbf{C}^{(s)}-\overline{\mathbf{C}^{(s)}}E_s\|_F^2 \qquad (1)$$

The first two terms remain the same as the classical dictionary learning. The third term helps decrease the intra-class distribution variances. K is the number of categories. $\mathbf{C^{(s)}}=[\mathbf{c_1^{(s)}},\mathbf{c_2^{(s)}},\ldots,\mathbf{c_{s_t}^{(s)}}]$ denotes the projections of data from category s. $\overline{\mathbf{C^{(s)}}}$ is the mean of $\mathbf{C^{(s)}}$. $\mathbf{E_s}=[1,1,\ldots]_{1\times s_t}$,

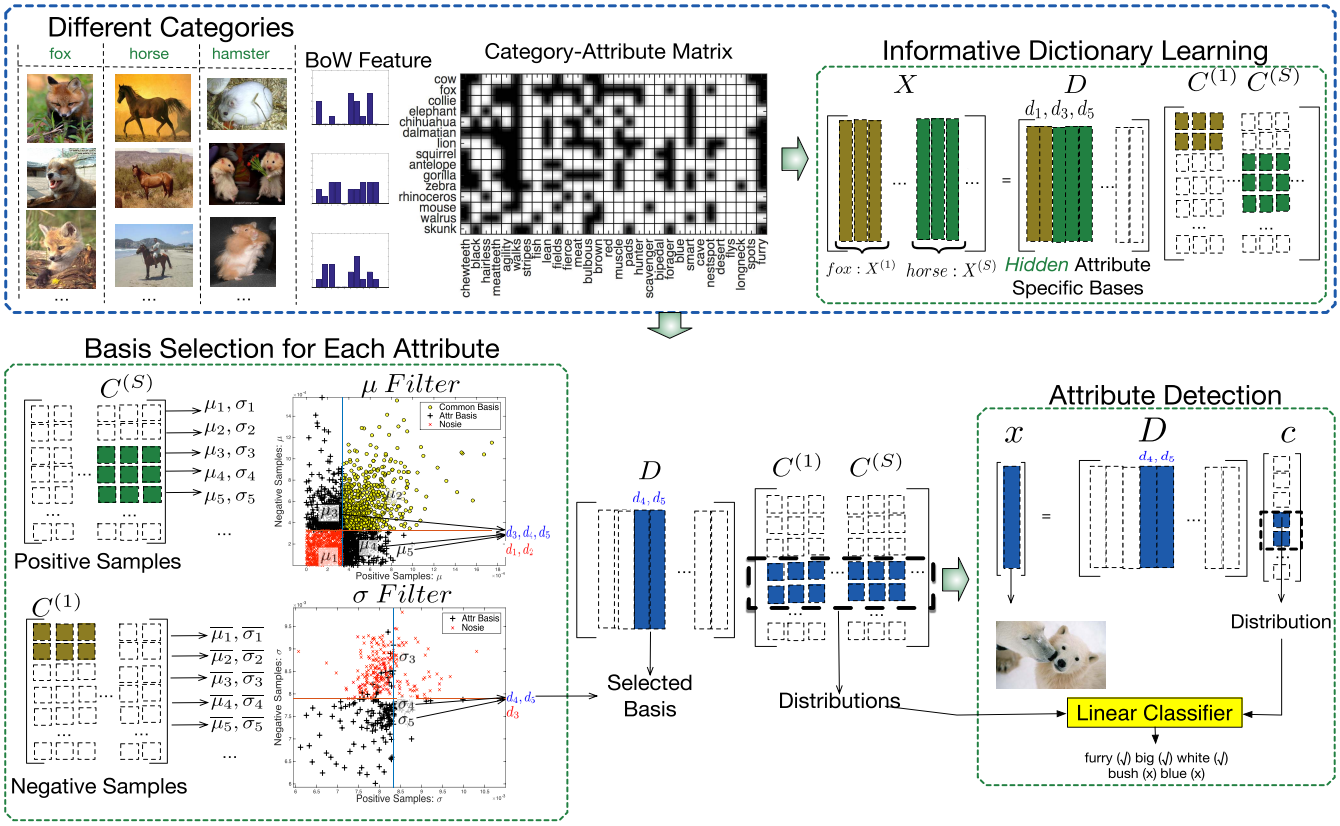


Fig. 2. **Pipeline Overview.** (top) A label constrained dictionary is learned by encouraging intra-class samples to lie close by. (bottom left) Multilayer filter: μ-Filter & σ-Filter are designed to select a set of robust and representative attribute specific bases to reconstruct each attribute. (bottom right) Attributes are predicted by linear SVM classifiers using the distributions over the attribute specific bases.

where s_t is the number of data from category s. α balances the reconstruction error against sparsity penalty while β denotes the weight of intra-class variance penalty. The third term forces all the intra-class data to lie close to the category center which is defined as the mean of the projection. Thus, the model will focus on learning the shared attributes among the intra-class data and the noise, such as background information will be suppressed. In addition, learning all the bases in the same dictionary, instead of multiple dictionaries, allows us to learn bases that are shared across different categories. Thus, the attribute specific bases can be identified by mining the shared bases across different categories containing that specific attribute.

Algorithm 1 Solution Structure

Initialization: $\mathbf{D} \leftarrow \mathbf{D_0}$, $\mathbf{C} \leftarrow \mathbf{C_0}$
repeat
  fix $\mathbf{D}$, *update* $\mathbf{C}$:
  for $\mathbf{C}^{(s)} \in \mathbf{C}$ **do**
    $ratio \leftarrow 1$
    while $ratio > threshold$ **do**
      run ***modified FISTA***
      update $ratio$
    end while
  end for
  fix $\mathbf{C}$, *update* $\mathbf{D}$
until converges

C. Optimization

The proposed optimization problem in Eqn.(1) is nonconvex. However, when one of the variables is fixed, the problem becomes convex with respect to the other one. Thus, we solve the problem by optimizing the objective by fixing one of the variables alternatively until the loss function converges. To learn the projection for each category, we decompose the objective into sub-objectives, and we adopt a modified Fast Iterative Soft-Thresholding Algorithm (FISTA) [53] algorithm to solve the sub-objectives. FISTA algorithm was proposed to solve the classical dictionary learning problem and it converges very fast. A soft-threshold step is incorporated to guarantee the sparseness of the solution. It converges in function values as $O(1/k^2)$ [53], in which k denotes the iteration times, while for the traditional ISTA method, the complexity is $O(1/k)$. The details are shown in Algorithm 1.

Initialization in Algorithm 1: we employ k-means clustering to find k centroids as the initial bases in dictionary $\mathbf{D}_0$. $\mathbf{C}_0$ is set to $\mathbf{0}$.

The **loop** in Algorithm 1 consists of two parts:

(1) Fix C, Optimize D: By setting the derivative of Eqn.(1) with respect to $\mathbf{D}$ equal to $\mathbf{0}$, we obtain,

$$(\mathbf{DC} - \mathbf{X})\mathbf{C}^T = \mathbf{0} \Rightarrow \mathbf{D} = \mathbf{X}\mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}$$

In case $\mathbf{CC}^T$ is singular, we use the following equation to update $\mathbf{D}$.

$$\mathbf{D} = \mathbf{XC}^T(\mathbf{CC}^T + \lambda\mathbf{I})^{-1}$$

λ is a small constant to guarantee that the matrix $\mathbf{CC}^T + \lambda\mathbf{I}$ is invertible when $\mathbf{CC}^T$ is singular.

(2) Fix D, Optimize C: To update $\mathbf{C}$, we decompose the objective into a set of sub-objectives. Each sub-objective corresponds to one category.

Note that when $\mathbf{D}$ is fixed, $L(\mathbf{D}; \mathbf{C}^{(s)}; \mathbf{X}^{(s)})$ is independent from each other with respect to s. Then the objective Eqn.(1) can be written as:

$$\min_{\mathbf{C}} \sum_{s=1}^{K} L(\mathbf{D}; \mathbf{C}^{(s)}; \mathbf{X}^{(s)}) = \sum_{s=1}^{K} \min_{\mathbf{C}^{(s)}} L(\mathbf{D}; \mathbf{C}^{(s)}; \mathbf{X}^{(s)})$$

Thus the original objective function is decomposed into a set of sub-objective functions with respect to each category. The third term in Eqn.(1) makes the $\mathbf{c}_i$ and $\mathbf{c}_j$ within the same category become dependent on each other. Thus, $\mathbf{c}_i$ and $\mathbf{c}_j$ must be updated simultaneously in order to make the whole system converge. We modify the FISTA algorithm to tackle the problem. The new sub-objective in our model is as follows,

$$F = \sum_{c \in C^{(s)}} \|\mathbf{Dc} - \mathbf{x}\|^2 + \alpha\|\mathbf{c}\|_1 + \beta\|\mathbf{c} - \frac{1}{N}\sum_{c_k \in C^{(s)}} \mathbf{c_k}\|^2$$

From the equation above, we can find that, for training data $\mathbf{x} \in \mathbf{X^{(s)}}$, its distribution $\mathbf{c}$ ($\mathbf{c} \in \mathbf{C^{(s)}}$) depends on other $\mathbf{c_k}$ ($\mathbf{c_k} \in \mathbf{C^{(s)}}$). Thus, the sub-objectives cannot be optimized independently. We modify the FISTA algorithm to optimize the sub-objectives from the same group simultaneously. The sub-objectives are grouped together if the training data belong to the same group. Then when updating $\mathbf{C}^{(s)}$, all $\mathbf{c}_j \in \mathbf{C}^{(s)}$ are updated simultaneously for $j = 1, \ldots, s_t$.

$$\mathbf{c_j} := \mathbf{c_j} - \gamma \frac{\partial F}{\partial \mathbf{c_j}}$$

Please refer to [53] for the details about how to select the appropriate γ, as well as the following soft thresholding process to delete the small values in $\mathbf{c_j}$. This updating procedure of $\mathbf{C^{(s)}}$ continues until convergence. To judge whether all the $\mathbf{c_j}$ in the same category converge or not, we refer to the metric $ratio$, which is defined as:

$$ratio = \min_{\mathbf{c_j} \in \mathbf{C^{(s)}}} \|\mathbf{c_j} - \widehat{\mathbf{c}}_\mathbf{j}\|^2 / \|\widehat{\mathbf{c}}_\mathbf{j}\|^2$$

in which $\widehat{\mathbf{c}}_\mathbf{j}$ denotes the updated value of $\mathbf{c_j}$. The threshold controls the number of iterations of each category. If $ratio < threshold$, the update procedure for the category will be terminated. We run the same procedure for each category. In Algorithm 1, line 4 to line 10 represent the pseudo-code to update $\mathbf{C}$. The setting of the parameter values is available at the end of section 4.3.2. The convergence condition required in step 12 of the algorithm is similar to the ratio defined in the FISTA algorithm.

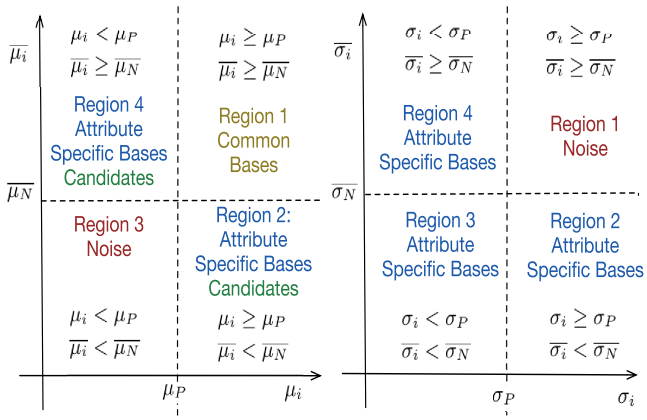


Fig. 3. μ-Filter & σ-Filter design for basis selection. μ-Filter selects discriminative bases and σ-Filter selects robust bases.

D. Multilayer Filter-Basis Selection

After learning the label constrained dictionary, we rely on the statistics of the projection $\mathbf{C}$ to divide the bases into 3 groups, the **common bases**, **attribute specific bases**, and **noise bases**. *Common Bases* are the bases over which both the positive and negative examples have large and stable distributions. *Attribute Specific Bases* are the bases over which only the positive or only the negative samples have large and stable distributions. *Noise Bases* are the remained ones.

Two metrics, the mean μ and the standard deviation σ, are used to characterize the distribution of samples. Thus, we design a two-layer filter for basis selection which consists of μ-Filter and σ-Filter.

Let $c_{i,j}$ denote the distribution of j-th sample over i-th basis. Then the mean of positive samples over the i-th basis is $\mu_i = \frac{1}{|P|}\sum_{j \in P} c_{i,j}$. P is the set of positive samples and $|P|$ is the cardinality of the set. Similarly, we have $\overline{\mu_i} = \frac{1}{|N|}\sum_{j \in N} c_{i,j}$ for the negative set N. The basis selection criterion is illustrated in Fig.3. μ_P, $\overline{\mu_N}$, σ_P and $\overline{\sigma_N}$ are threshold values that control the ratio of selected bases. The first layer filter, μ-Filter, filters out part of the noise bases and all the common bases and only the candidates for attribute specific bases are left. The second layer σ-Filter further filters out the unstable bases. Thus, only the stable candidates are selected as attribute specific bases.

μ-Filter The candidates of positive stimulus bases are the ones which are located in **region 2** (in the μ-Filter section of Fig. 3) over which only the positive samples have large mean distribution. The candidates of negative stimulus are located in **region 4** over which only the negative samples have large mean distribution. The common bases are located in **region 1**, and the noise bases are located in **region 3**. The candidates of attribute specific bases will be further processed by the second layer filter, σ-Filter, in where only the robust candidates can pass and be selected as attribute specific bases.

σ-Filter is the second layer filter. Given a candidate of positive stimulus basis, it will be selected as a positive stimulus only if the standard deviation of positive examples over the basis is small while there is no requirement for the negative samples. The positive stimulus bases are located in **region 4** & **region 3** in the σ-Filter section of Fig.3. The robust negative

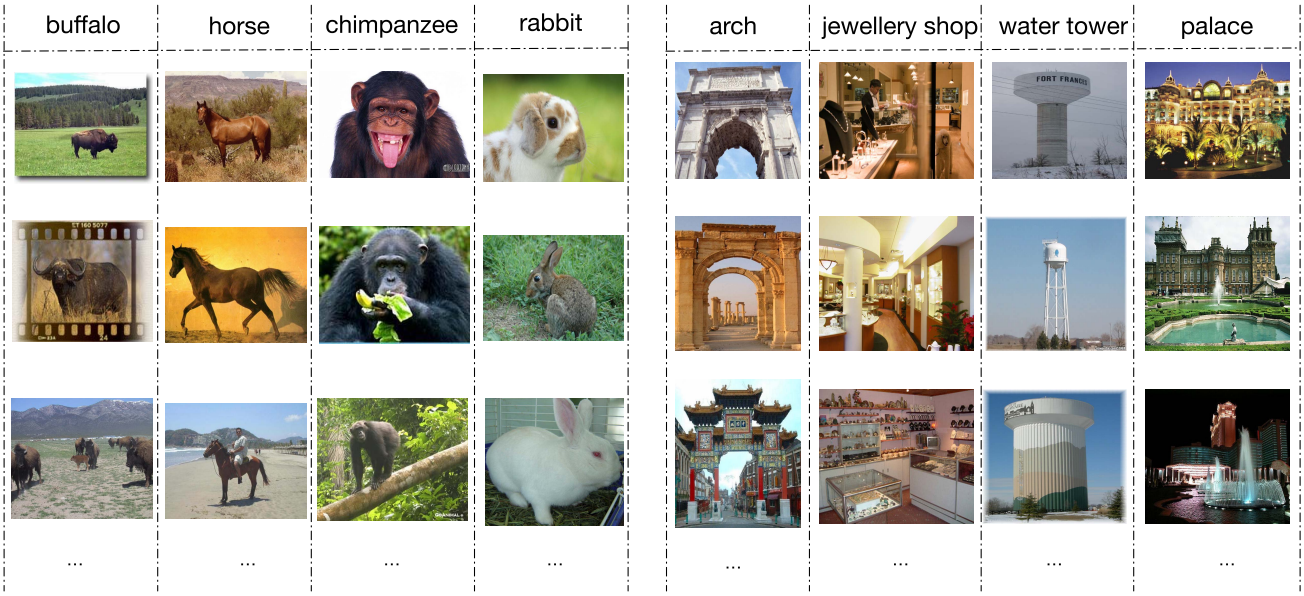


Fig. 4. (left) Images from the AwA dataset. (right) Images from the SUN attribute dataset. It is worth noticing that the attributes in the AwA dataset are class-wise, and there is no intra-class attribute variance. The attributes in the SUN attribute dataset are class-agnostic, and there exists intra-class attribute variance.

stimulus bases are selected in a similar manner. The negative stimulus bases located in **region 2** & **region 3**. The unstable candidates are classified as noise bases.

E. Attribute Classifier & Evaluation Metric

After obtaining the attribute specific bases, we adopt linear SVM as attribute classifier. The training data for the classifiers are the sparse representations of samples over the attribute specific bases. To detect attributes for a new image, the image is first decomposed by the dictionary to get a more compact representation. Then its distribution over the attribute specific bases will be used to perform attribute detection.

The testing data are very biased, and we use F_1 score to evaluate the performance of our method. F_1 is the harmonic mean of precision and recall:

$$F_1 = 2\frac{precision \cdot recall}{precision + recall}$$

After obtaining the F_1 scores of multiple attributes, the mean F_1 score is adopted as the evaluation metric.

IV. Experiments

In this section, we conduct extensive experiments to evaluate our proposed method.

A. Datasets

We evaluate our proposed framework with two datasets. The 'Animal with Attributes' (AwA) dataset introduced by Lampert *et al.* [11], and the 'SUN Attribute Database' introduced by Patterson and Hays in [54]. Fig. 4 shows examples from the AwA dataset and SUN dataset. The AwA dataset contains 50 animal categories, which are separated into 2 parts: 40 seen animal categories and 10 unseen animal categories. 85 semantic attributes are defined in the dataset, which are grouped into 9 groups (*color, texture, shape, etc.*). The attributes are mapped to the categories according to the attribute-category matrix. The features are provided along with the dataset which include SIFT, Color SIFT, Pyramid HOG and Decaf features which are generated at the fully connected layer (fc7) from CaffeNet.

The Sun attribute database is a large-scale scene database which includes 102 discriminative continuous attributes which describes scenes' materials, surface properties, lighting, functions, affordances, and spatial layout properties. It consists of 14340 images from 717 classes (20 images per class on average). The authors of [54] also provide image features which are GIST, HOG, self-similarity, and Geometric color histograms. For our experiment, we rely on these features.

Different from the binary class-wise category-attribute matrix in the AwA dataset, the attribute presence probability in the scene-attribute matrix is continuous. Each images is labeled by 3 annotators. The image-attribute element is set to 1 if the annotator believes that such attribute is present in the image. Otherwise it is set to 0. Finally, the value of the image-attribute matrix is set by taking the average of the presence scores from 3 annotators. In order to convert the continuous value of the probability into a binary one, we set the value in the image-attribute matrix to 1 if two or more annotators vote for the presence of an attribute in an image and set it to 0 if it receives 0 vote for its presence. If there is only one vote for the presence of an attribute in an image, the image will be neglected for this attribute, as this implies that the image is in a transitional state between the two states (presence and absence). Fig.5 shows the category-wise attribute matrix. In the AwA dataset, each attribute has at least one class of positive training samples. But the attributes in the SUN attribute dataset are category agnostic, and intra-class attribute differences are allowed. Thus the training samples for some attributes in the SUN attribute dataset can be extremely biased.

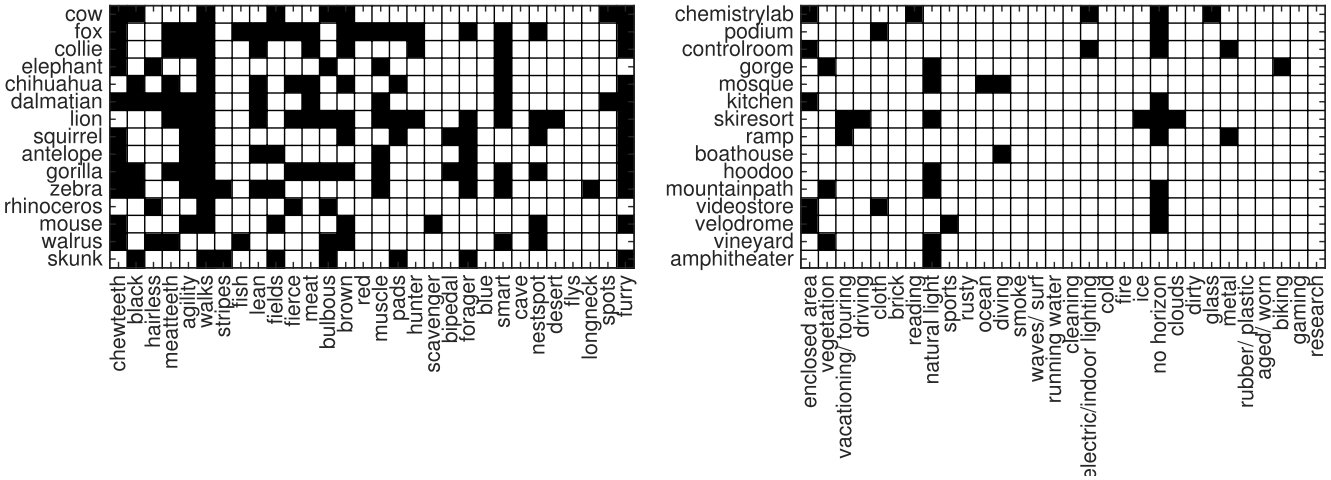


Fig. 5. (left) Binary animal-attribute matrix from the AwA dataset: 15×30 extracted from the complete 50×80 matrix. (right) Binary scene-attribute matrix from the SUN attribute dataset: 15×30 extracted from the complete 102×717 matrix.

B. Experimental Settings

1) Data Split for Training and Testing: For the AwA dataset, the label constrained dictionary is trained with the data from the 40 seen categories. The parameters of the multilayer filter and the linear SVM classifier of the binary attributes are learned jointly based on the seen categories. 5-fold cross validation is implemented to select the optimal classification parameters. The remaining 10 unseen categories are used to evaluate the generalization properties of the attribute specific bases. We use all the samples in the seen categories for training and all the samples in the unseen categories for testing. During the classification training phase, the weights of the miss-labeling penalty for the negative and positive data are set inversely proportional to the size of negative and positive data to make up for the bias of training data.

For the SUN attribute dataset, the seen and unseen categories are not predefined. We randomly select 358 categories as seen categories, and 359 categories as unseen categories. The label constrained dictionary and basis selections filters are designed and calibrated based on the seen categories. The training data in the SUN attribute dataset are more biased than the data in the AwA dataset. To prevent attribute detection from being influenced by attribute popularity, we fix the ratio of positive and negative samples both for training and testing. Each classifier is trained on 300 images and test on 100 images. The attributes whose positive samples are less than 200 are excluded. Thus, 87 attributes in the SUN attribute attribute dataset are selected to evaluate our methods.

2) Parameter Settings: In the label constrained dictionary learning phase, α and β are tuned from $[10^{-3}, 10^{-2}, \ldots, 10^{3}]$. Dictionary size varies within the range of $[0.5, 1, 1.5, \ldots, 3]\times 10^{3}$. For the multilayer filter, the threshold values $\mu_P, \overline{\mu_N}, \sigma_P, \overline{\sigma_N}$ are tuned from $[10\%, 20\%, \ldots, 100\%]$. In the attribute classifier training phase, the penalty parameter C in the SVM classifier is tuned from $[10^{-3}, 10^{-2}, \ldots, 10^{3}]$.

C. Results

1) Evaluation of Label Constrained Dictionary Learning and Basis Selection: To evaluate the performance of our introduced method, we compare our method with label constrained dictionary learning without basis selection, the classical dictionary learning, as well as the raw feature. Linear SVM classifiers are employed, and the mean F1-score is employed as the evaluation metric.

Fig.6 (left) shows the mean F1-score in the AwA dataset. In Fig.6 (left), deep feature is employed as the raw feature. Similarly, Fig.6 (right) shows the mean F1-score in the SUN attribute dataset where the GIST [55] feature is employed.

From Fig.6 we can observe that our method outperforms all the baselines for both datasets. The classical dictionary learning has similar performance with the raw feature. The label constrained dictionary learning outperforms both the raw feature and classical dictionary learning method. For the AwA dataset, the label constrained dictionary learning outperforms the raw feature by 3.5%. However, for the SUN attribute dataset, the improvement is very small (0.9%). There is no surprise that the label constrained dictionary learning has a more remarkable effect on the AwA dataset compared with the SUN attribute dataset. This is because the attributes in the AwA dataset are class-wise. Then, there is no intra-class attribute variance. However, for the SUN attribute dataset, the attribute is class agnostic. Then, there exists a certain amount of intra-class attribute variance. Our label constrained dictionary learning is aimed at suppressing the intra-class noise. Consequently, the performance of the label constrained dictionary learning is restricted by the intra-class attribute variance. The reason why the label constrained dictionary learning still outperforms the raw feature in the SUN attribute dataset is that most images within the same class still share the same attributes. Thus, the label constrained dictionary learning can still help focus on learning those attributes which are shared through the whole class.

We can also observe that basis selection further improves the performance of label constrained dictionary learning by 7.89% on the AwA dataset, and 4.4% on the SUN dataset.

2) Multilayer Filter Parameter Settings and Convergence Study: We employ the AwA dataset to study the sensitivity of the multilayer filter parameters. Fig.7 shows the grid plot

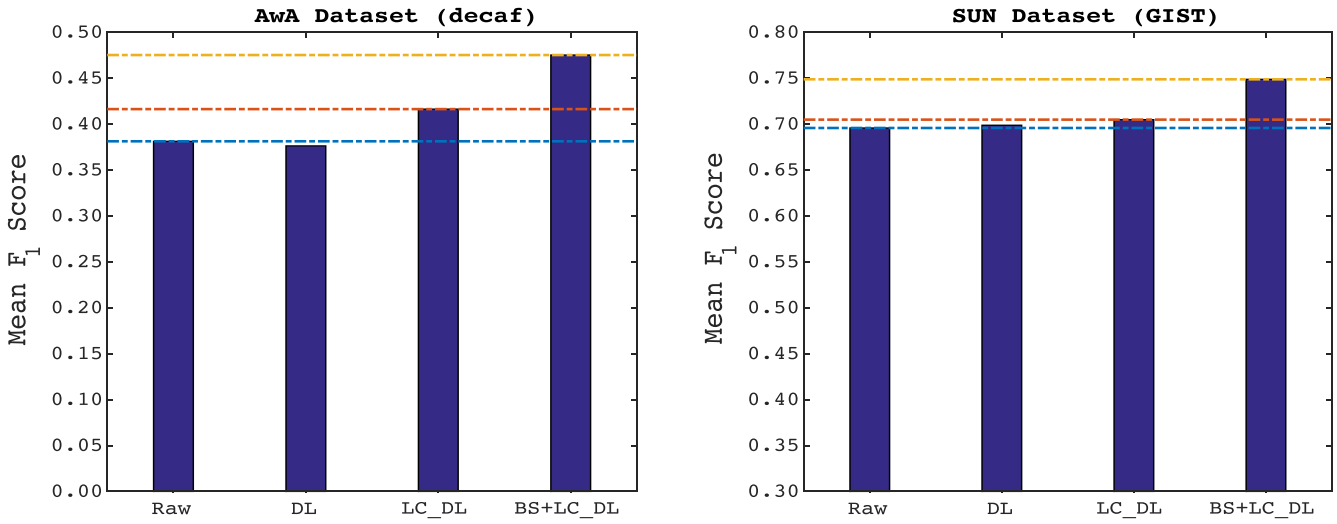


Fig. 6. Performance comparison between **Raw Feature, Classical Dictionary Learning approach (DL), Label Constrained Dictionary Learning approach (LC_DL) and Label Constrained Dictionary Learning combined with basis selection (BS+LC_DL)** on the AwA dataset with GIST feature and the SUN attribute dataset with decaf feature.

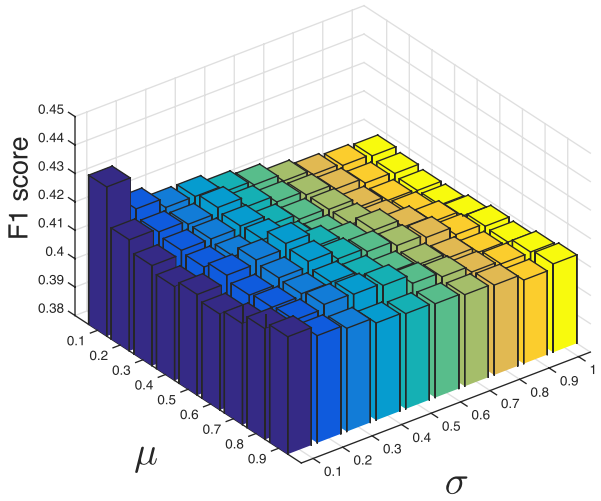


Fig. 7. Mean F_1 Score of 85 attributes for the AwA dataset over different μ & σ settings.

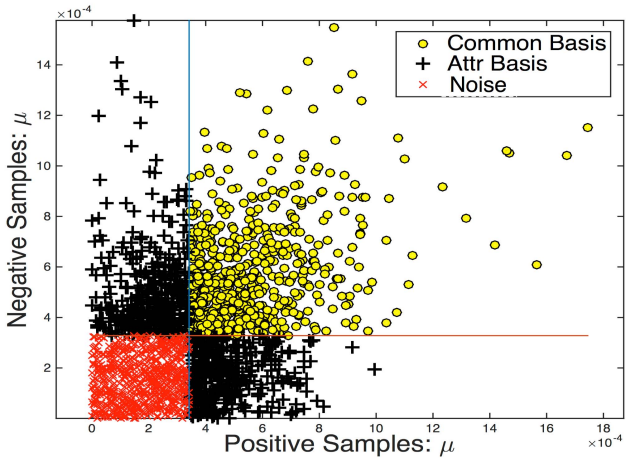


Fig. 8. Scatter Plot of bases over positive and negative samples in μ-Filter (20% bases are selected).

of the mean F_1 score with respect to different filter parameters. The dictionary size is set to be 2000. The μ-Filter and σ-Filter control the ratio of selected bases. The ratio of the selected bases ranges from 10% to 100%. The performance is measured on the unseen object categories. The value of the bar is the mean F_1 score of all the 85 attributes. From Fig.7, we observe that when more bases are selected either by the μ-Filter or by the σ-Filter, the mean F_1 score tends to decrease. The maximum F_1 score is obtained when both the μ-Filter and the σ-Filter only select 10% bases. From this observation, we can conclude that, the basis selection improves the performance of attribute detectors, and the best ratio of the basis selection lies close to 10% which could be mined out by doing a fine-grained search of the ratio. The optimal filter parameter settings for the SUN attribute dataset are configured in the same way. As the training samples in the SUN attribute dataset is relatively small compared with that in the AwA dataset, its dictionary size is set to be 500.

Fig.8 illustrates the first layer filter, namely, the μ-Filter. The two decision boundaries control the ratio of the selected candidates of the attribute specific bases before putting them into σ-Filter. The decision boundaries in the μ-Filter are determined by two threshold values, μ_P, $\overline{\mu_N}$ which correspond to the mean of the distribution of the positive samples, and to the mean of the distribution of the negative samples respectively. The two boundaries divide the bases into four regions. However, only the bases in the upper-left region and the lower-right region are selected as representative bases. The bases in the upper-left region represent what the attribute does *not have*. The bases in the lower-right region represent what the attribute *has*. The bases in lower-left region are regarded as noise as both positive and negative samples have small distributions over them. By setting the boundaries to different values, different amount of bases can be selected.

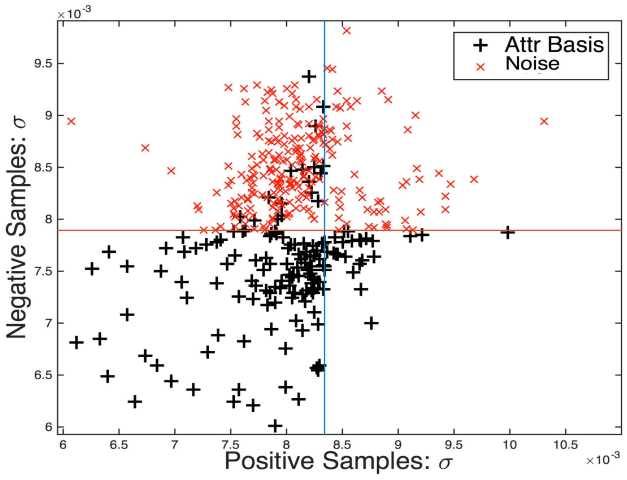


Fig. 9. Scatter Plot of bases over positive and negative samples in σ-Filter (20% bases are selected).

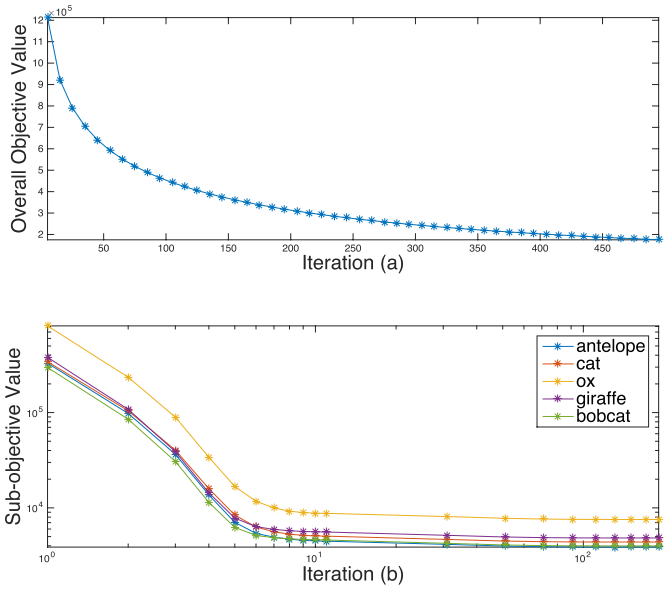


Fig. 10. (a) Convergence curve of the overall function. (b) Convergence curve of sub-objective function for each animal category $\mathbf{C}^{(s)}$.

For μ filter, we sort the bases in ascending order with respect to μ and $\hat{\mu}$ separately. Then we select the desired amount of percentage (10%, 20%, etc) with respect to μ and $\hat{\mu}$ separately. The attribute specific bases are then selected from these bases by removing their overlap (noise bases).

Fig.9 is the scatter plot of basis selection with the σ-Filter. After the selection of representative bases, the σ-Filter is applied to discover the candidates which are robust enough to be attribute specific bases. The decision boundaries in the σ-Filter are determined by two threshold values, σ_P and $\overline{\sigma_N}$ which correspond to the standard deviation of positive samples, and the standard deviation of negative samples respectively. If either the positive samples or the negative samples have large mean and small standard deviations over the representative bases, the bases will be selected as attribute specific bases. Otherwise, they will be filtered out. Thus, the robust and representative attribute specific bases are obtained.

We also study the convergence of our algorithm with the AwA dataset. We rely on K-means to select K most representative basis to initialize the dictionary. Fig.10 (a) shows the convergence curve of the overall function. The threshold controls the number of iterations of the algorithm for each category. We set the threshold to 0.01. Fig.10 (b) shows the log plot of the loss when updating $\mathbf{C}^{(s)}$ for five categories. It shows that all the five sub-objectives converge very fast. The threshold could be adjusted to a smaller number if we expect the algorithm to have more iterations.

3) Comparison With Baselines: After performing basis selection for the 85 attributes with the multilayer filter, the next step is to make use of the attribute specific bases to train the classifiers and we test these classifiers with the unseen categories. We divide the baselines into two groups, namely, the **non-dictionary learning group** and **dictionary learning group**. For the non-dictionary learning group, we use the following baselines:

- 1) The lib-svm classifiers combined with raw features.
- 2) The inter-group feature competition and intra-group feature sharing multi-task learning framework with $l_{2,1}$-norm regularizer [15], which is referred to as **Attr-Attr Relationship** in the tables.

For the dictionary learning group:

- 1) Classical dictionary learning (**DL**) method.
- 2) Label constrained dictionary learning without basis selection (**LC_DL**).
- 3) The label constrained dictionary learning which performs feature selection randomly (**RBS+LC_DL**).
- 4) Other dictionary learning frameworks which integrate the dictionary learning process and classifier training process, such as supervised dictionary learning [56], label consistent dictionary learning [57], as well as discriminative dictionary learning [58].

Fig.11 illustrates the performance of different approaches for some attributes in the AwA dataset. It shows the F_1 score for each attribute using decaf feature. From Fig.11, we can observe that for most attributes, our method outperforms the other baselines. In general, our method outperforms other baselines in 64 out of 85 attributes. When different features are employed, the performance may vary a bit. Our method has inferior performance over some attributes, such as "newworld" and "oldworld" in Fig.12. This is probably because these abstract attributes rely on the global features while our basis selection strategy harms the global information. This problem might be solved by integrating global features as an extra channel with the selected basis. In the future, we will explore how to integrate the global features into the attribute specific basis.

Table I shows the performance of different approaches with different features on the attributes from the AwA dataset. Two metrics are employed to measure the performance, namely, the average F_1 score of the 85 attributes, as well as the mean precision. We can see from Table I that the performance of our method outperforms other baselines.

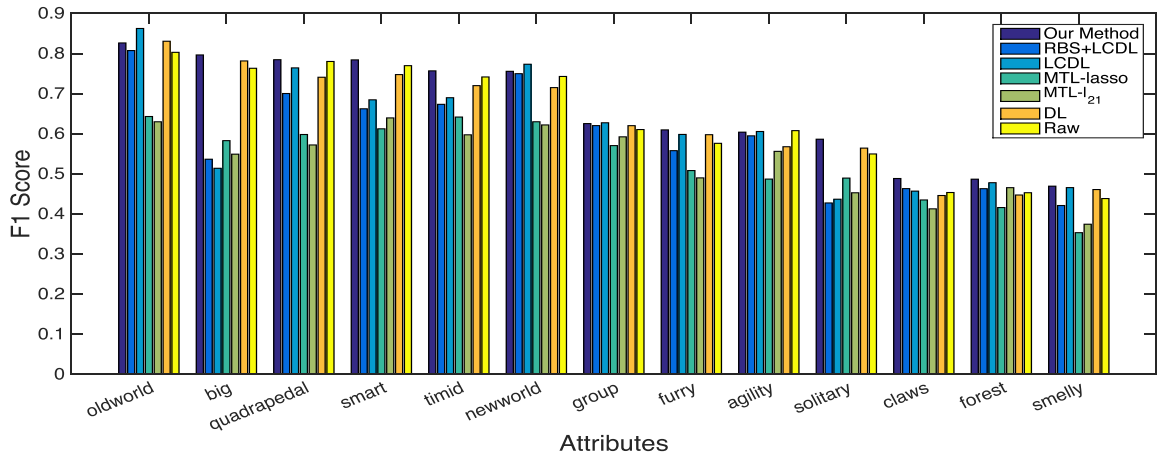


Fig. 11. AwA dataset: Mean F_1 Score over the attributes on unseen animal categories. Our method (Basis Selection + Label Constrained Dictionary Learning) outperforms other baselines for most of the attributes.

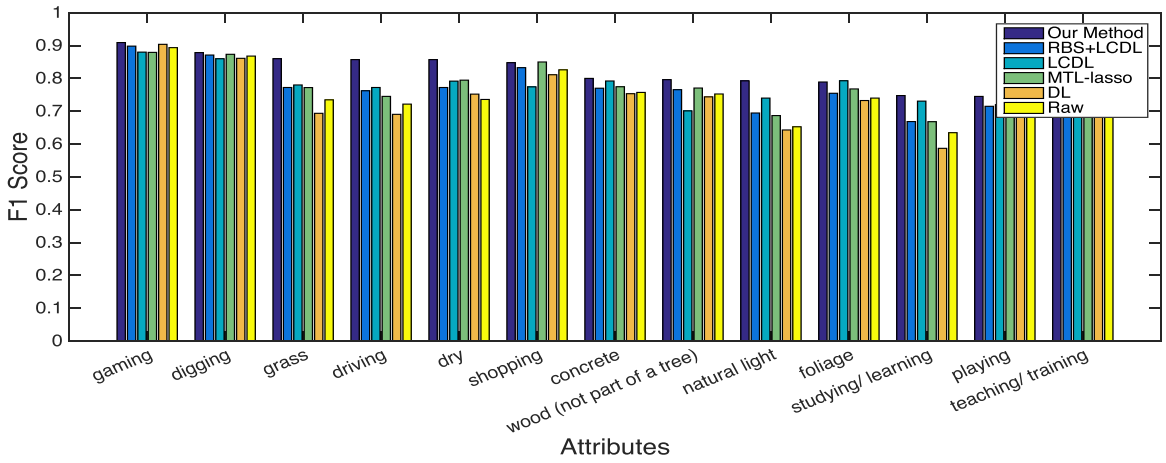


Fig. 12. SUN attribute dataset: Mean F_1 Score over the attributes. Our method (Basis Selection + Label Constrained Dictionary Learning) outperforms other baselines for most of the attributes. The improvement is relatively small in comparison with the AwA dataset.

TABLE I

AWA DATASET: PERFORMANCE COMPARISON WITH BASELINES BASED ON DIFFERENT FEATURES

Methods		Features							
		SIFT		Color-SIFT		Pyramid-HOG		DeCaf	
		F_1 score	Precision	F_1 score	Precision	F_1 score	Precision	F_1 score	Precision
Non-Dictionary Methods	Lib SVM + Raw Feature	0.4034	0.4145	0.3739	0.3976	0.4264	0.4083	0.4039	0.4373
	Attr-Attr Relationship [15]	0.4107	0.4097	0.4167	0.4112	0.4166	0.3860	0.4432	0.4386
Dictionary Methods	DL [19]	0.4024	0.4185	0.3909	0.3805	0.4289	0.4128	0.4133	0.4364
	LC_DL	0.4011	0.4165	0.3981	0.4064	0.4270	0.4193	0.4163	0.4346
	RBS+LC_DL	0.4122	0.4140	0.4070	0.4071	0.4273	0.4233	0.4159	0.4179
	Jiang [57]	0.4103	0.4036	0.3898	0.3993	0.4134	0.4130	0.4039	0.4266
	Zhang [58]	0.4088	0.4001	0.3788	0.3834	0.4099	0.4038	0.4003	0.4197
	Mairal [56]	0.4174	0.4346	0.4006	0.4247	0.4396	0.4213	0.4219	0.4455
	Our method	**0.4789**	**0.4493**	**0.4465**	**0.4348**	**0.4815**	**0.4328**	**0.4752**	**0.4568**

To show the effectiveness of our method, we also use the SUN attribute dataset to evaluate our method. Fig.12 shows the performance of different approaches for some attributes in the SUN attribute dataset and Table II shows the performance of different approaches with different features. Similarly, our method still outperforms other baselines on this dataset. However, the performance improvement is less significant when compared with the AwA dataset. This is because the attributes in the SUN attribute dataset are class-agnostic and there exists intra-class attribute variance. Thus, the intra-class attribute variance weakens the performance of the label constrained dictionary learning which is aimed at minimizing the intra-class variance.

For the test data without labels, we follow the settings in [57]. The novel regularizer will be neglected. Thus, the model becomes a standard dictionary learning model. We show

TABLE II

SUN ATTRIBUTE DATASET: PERFORMANCE COMPARISON WITH BASELINES BASED ON DIFFERENT FEATURES

Methods		Features							
		GIST		HOG		Self-Similarity		Geometric Color Hist	
		F_1 score	Precision	F_1 score	Precision	F_1 score	Precision	F_1 score	Precision
Non-Dictionary Methods	Lib SVM + Raw Feature	0.7049	0.6984	0.7162	0.7069	0.7091	0.7044	0.6865	0.6671
	Attr-Attr Relationship [15]	0.5887	0.5321	0.5354	0.6431	0.5686	0.5633	0.5897	0.6182
Dictionary Methods	DL [19]	0.6575	0.6791	0.6906	0.6363	0.6488	0.6885	0.6743	0.6606
	LC_DL	0.7049	0.6984	0.7162	0.7069	0.7091	0.7044	0.6865	0.6671
	RBS+LC_DL	0.6958	0.6868	0.7072	0.7006	0.7052	0.6914	0.6946	0.6827
	Jiang [57]	0.6944	0.6865	0.7043	0.6923	0.6946	0.6994	0.6683	0.6527
	Zhang [58]	0.6896	0.6875	0.6987	0.6887	0.6839	0.6823	0.6645	0.6498
	Patterson [54]	0.7163	0.7038	0.7199	0.7085	0.7104	0.7064	0.6913	0.6752
	Mairal [56]	0.7241	0.7177	0.7194	0.7263	0.7282	0.7141	0.7025	0.6699
	Our method	**0.7488**	**0.7454**	**0.7916**	**0.7644**	**0.7657**	**0.7493**	**0.7349**	**0.7298**

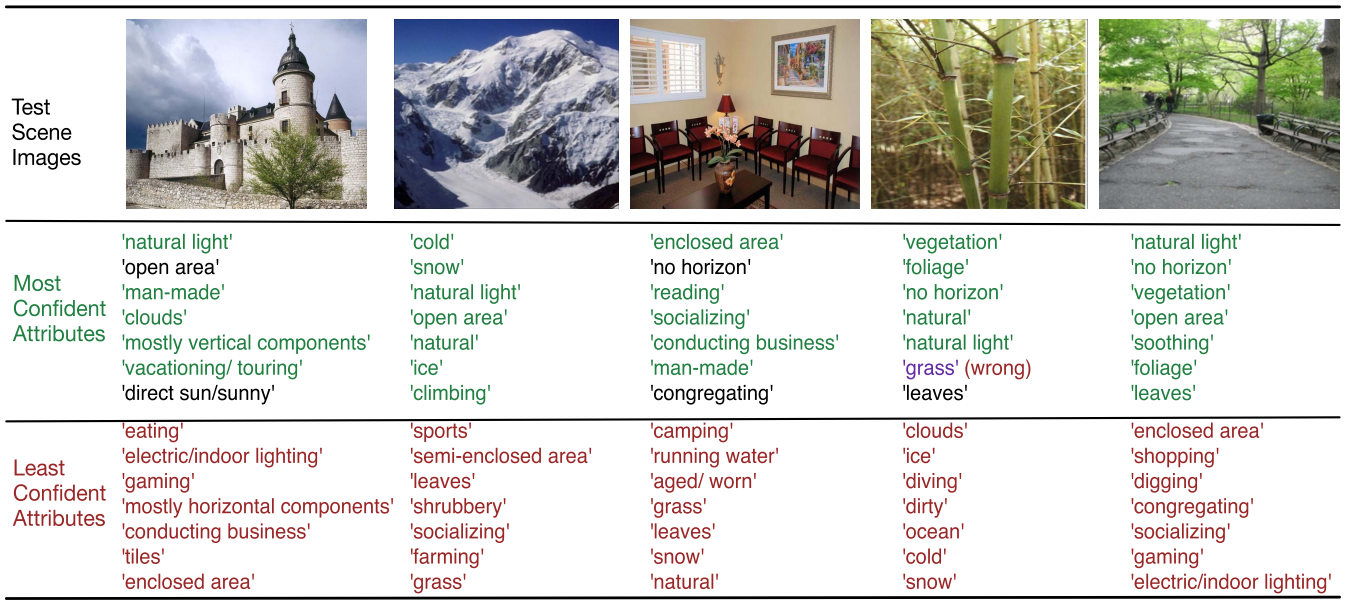


Fig. 13. Attribute Detection for SUN attribute dataset. For each query image, 7 most confidently recognized attributes (green and black) and 7 least confidently recognized attributes (red) are listed. The black ones are the attributes which only receive 1 vote from 3 annotators and the green ones receive at least 2 votes from 3 annotators.

our qualitative results of our attribute classifiers in Fig.13. Most of the attributes which have high confidences received at least 2 votes from 3 annotators, and a small portion of the attributes receives 1 vote. The attributes with low confidences are indeed absent in the image. For the forth image in Fig.13, there is a false positive attribute *grass*. This is because this image is visually similar to the *grass* as it has *dirt* and visually *green*. It is very interesting that some function attributes can be recognized with very high confidences even though these functions are very abstract and hard to define visually. For example, **socializing, conducting business** in the third image are detected successfully.

V. CONCLUSIONS

In this paper, we propose a label constrained dictionary learning method to improve the performance of attribute detectors. First, we learn a label constrained dictionary which encourages the sparse representations of intra-class data lie close by and suppresses the intra-class noise. Then, we design a multilayer filter, the μ-Filter and σ-Filter, to mine out a set of robust and representative attribute specific bases for each attribute. We test our method on both the AwA dataset and the SUN attribute dataset, the extensive experimental results demonstrate effectiveness of our proposed method, and it outperforms other important baselines on average. In recent years, convolutional neural network (CNN) is widely used in many tasks, and Zeiler *et. al* pointed out that the third convolutional layer in the Alex Net corresponds to attribute [59]. Thus, the convolutional neural network (CNN) may also benefit attribute detection task.

Overall, the proposed label constrained dictionary learning is novel for attribute detection. Most attributes considered in both the AwA dataset and the SUN attribute dataset are global attributes (function attributes) while some may be localized (material attributes in the SUN dataset, texture attributes in the AwA dataset). Thus, attribute localization techniques might help improve the performance of those attributes who have spatial support. Besides, the attributes are learned independently without considering the attribute correlations. But in reality, some attributes are closely correlated (smoke and fire

in the SUN dataset, swim and water in AwA dataset). Thus, the multi-attribute classification method which considers the attribute correlation may improve the performance by learning the attribute classifiers jointly. In the future, we would further explore attribute correlations to improve attribute detection accuracy.

REFERENCES

[1] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. CVPR*, 2009, pp. 1778–1785.

[2] J. Cai, Z.-J. Zha, M. Wang, S. Zhang, and Q. Tian, "An attribute-assisted reranking model for Web image search," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 261–272, Jan. 2015.

[3] W. Wang, Y. Yan, and N. Sebe, "Attribute guided dictionary learning," in *Proc. ICMR*, 2015, pp. 211–218.

[4] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 453–465, Mar. 2014.

[5] J. Feng, S. Jegelka, S. Yan, and T. Darrell, "Learning scalable discriminative dictionary with sample relatedness," in *Proc. CVPR*, Jun. 2014, pp. 1645–1652.

[6] M. Rastegari, A. Farhadi, and D. Forsyth, "Attribute discovery via predictable discriminative binary codes," in *Proc. ECCV*, 2012, pp. 876–889.

[7] B. Saleh, A. Farhadi, and A. Elgammal, "Object-centric anomaly detection by attribute-based reasoning," in *Proc. CVPR*, 2013, pp. 787–794.

[8] D. Parikh and K. Grauman, "Relative attributes," in *Proc. ICCV*, 2011, pp. 503–510.

[9] R. N. Sandeep, Y. Verma, and C. V. Jawahar, "Relative parts: Distinctive parts for learning relative attributes," in *Proc. CVPR*, 2014, pp. 3614–3621.

[10] S. Shankar, V. K. Garg, and R. Cipolla, "Deep-carving: Discovering visual attributes by carving deep neural nets," in *Proc. CVPR*, 2015, pp. 3403–3412.

[11] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. CVPR*, 2009, pp. 951–958.

[12] D. Parikh and K. Grauman, "Interactively building a discriminative vocabulary of nameable attributes," in *Proc. CVPR*, 2011, pp. 1681–1688.

[13] T. L. Berg, A. C. Berg, and J. Shih, "Automatic attribute discovery and characterization from noisy Web data," in *Proc. ECCV*, 2010, pp. 663–676.

[14] K. Duan, D. Parikh, D. Crandall, and K. Grauman, "Discovering localized attributes for fine-grained recognition," in *Proc. CVPR*, 2012, pp. 3474–3481.

[15] D. Jayaraman, F. Sha, and K. Grauman, "Decorrelating semantic visual attributes by resisting the urge to share," in *Proc. CVPR*, 2014, pp. 1629–1636.

[16] Y. Han, F. Wu, X. Lu, Q. Tian, Y. Zhuang, and J. Luo, "Correlated attribute transfer with multi-task graph-guided fusion," in *Proc. 20th ACM MM*, 2012, pp. 529–538.

[17] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[18] Y. Yan, H. Shen, G. Liu, Z. Ma, C. Gao, and N. Sebe, "Glocal tells you more: Coupling glocal structural for feature selection with sparsity for image and video classification," *Comput. Vis. Image Understand.*, vol. 124, pp. 99–109, Jul. 2014.

[19] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. 24th ICML*, 2007, pp. 759–766.

[20] V. Escorcia, J. C. Niebles, and B. Ghanem, "On the relationship between visual attributes and convolutional networks," in *Proc. CVPR*, 2015, pp. 1256–1264.

[21] A. Farhadi *et al.*, "Every picture tells a story: Generating sentences from images," in *Proc. 11th ECCV*, 2010, pp. 15–29.

[22] Y. Han, Y. Yang, Z. Ma, H. Shen, N. Sebe, and X. Zhou, "Image attribute adaptation," *IEEE Trans. Multimedia*, vol. 16, no. 4, pp. 1115–1126, Jun. 2014.

[23] A. Farhadi, I. Endres, and D. Hoiem, "Attribute-centric recognition for cross-category generalization," in *Proc. CVPR*, 2010, pp. 2352–2359.

[24] R. Tao, A. W. Smeulders, and S.-F. Chang, "Attributes and categories for generic instance search from one example," in *Proc. CVPR*, 2015, pp. 177–186.

[25] C.-N. J. Yu and T. Joachims, "Learning structural SVMs with latent variables," in *Proc. 26th ICML*, 2009, pp. 1169–1176.

[26] Y. Gao, R. Ji, W. Liu, Q. Dai, and G. Hua, "Weakly supervised visual dictionary learning by harnessing image attributes," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5400–5411, Dec. 2014.

[27] Y. Wang and G. Mori, "A discriminative latent model of object classes and attributes," in *Proc. 11th ECCV*, 2010, pp. 155–168.

[28] A. Kovashka and K. Grauman, "Attribute pivots for guiding relevance feedback in image search," in *Proc. ICCV*, 2013, pp. 297–304.

[29] A. Biswas and D. Parikh, "Simultaneous active learning of classifiers & attributes via relative feedback," in *Proc. CVPR*, 2013, pp. 644–651.

[30] B. Qian, X. Wang, N. Cao, Y.-G. Jiang, and I. Davidson, "Learning multiple relative attributes with humans in the loop," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5573–5585, Dec. 2014.

[31] X. You, R. Wang, and D. Tao, "Diverse expected gradient active learning for relative attributes," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3203–3217, Jul. 2014.

[32] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *Proc. CVPR*, 2011, pp. 3337–3344.

[33] L. Lin, Y. Lu, Y. Pan, and X. Chen, "Integrating graph partitioning and matching for trajectory analysis in video surveillance," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4844–4857, Dec. 2012.

[34] F. S. Khan, J. van de Weijer, R. M. Anwer, M. Felsberg, and C. Gatta, "Semantic pyramids for gender and action recognition," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3633–3645, Aug. 2014.

[35] J. Liu *et al.*, "Video event recognition using concept attributes," in *Proc. WACV*, 2013, pp. 339–346.

[36] S. Wang, X. Chang, X. Li, Q. Z. Sheng, and W. Chen, "Multi-task support vector machines for feature selection with shared knowledge discovery," *Signal Process.*, vol. 120, pp. 746–753, Mar. 2014.

[37] Q. Zhang, L. Chen, and B. Li, "Max-margin multiattribute learning with low-rank constraint," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 2866–2876, Jul. 2014.

[38] S. Huang, M. Elhoseiny, A. Elgammal, and D. Yang, "Learning hypergraph-regularized attribute predictors," in *Proc. CVPR*, 2015, pp. 409–417.

[39] H. Chen, A. Gallagher, and B. Girod, "Describing clothing by semantic attributes," in *Proc. 12th ECCV*, 2012, pp. 609–623.

[40] 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.

[41] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief net model for visual area V2," in *Proc. NIPS*, 2008, pp. 873–880.

[42] Y. Yan *et al.*, "Complex event detection via event oriented dictionary learning," in *Proc. AAAI*, 2015, pp. 3841–3847.

[43] Y. Yan *et al.*, "Event oriented dictionary learning for complex event detection," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1867–1878, Jun. 2015.

[44] J. Luo, W. Wang, and H. Qi, "Group sparsity and geometry constrained dictionary learning for action recognition from depth maps," in *Proc. ICCV*, 2013, pp. 1809–1816.

[45] Q. Qiu, Z. Jiang, and R. Chellappa, "Sparse dictionary-based representation and recognition of action attributes," in *Proc. ICCV*, 2011, pp. 707–714.

[46] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. CVPR*, 2010, pp. 3501–3508.

[47] H. Guo, Z. Jiang, and L. S. Davis, "Discriminative dictionary learning with pairwise constraints," in *Proc. ACCV*, 2013.

[48] X. Zhu, H.-I. Suk, and D. Shen, "Matrix-similarity based loss function and feature selection for alzheimer's disease diagnosis," in *Proc. CVPR*, 2014, pp. 328–342.

[49] L. Zhang, M. Wang, R. Hong, B. Yin, and X. Li, "Large-scale aerial image categorization using a multitask topological codebook," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 535–545, Feb. 2016.

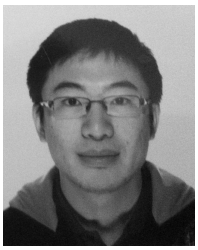
[50] Y. Yan, E. Ricci, R. Subramanian, and G. Liu, "Multitask linear discriminant analysis for view invariant action recognition," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5599–5611, Dec. 2014.

[51] Y. Yan, E. Ricci, G. Liu, and N. Sebe, "Egocentric daily activity recognition via multitask clustering," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 2984–2995, Oct. 2015.

[52] Y. Yan, E. Ricci, R. Subramanian, O. Lanz, and N. Sebe, "No matter where you are: Flexible graph-guided multi-task learning for multi-view head pose classification under target motion," in *Proc. ICCV*, 2013, pp. 1177–1184.

[53] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
[54] G. Patterson, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *Proc. CVPR*, 2012, pp. 2751–2758.
[55] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
[56] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Proc. NIPS*, 2009, pp. 1033–1040.
[57] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.
[58] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. CVPR*, 2010, pp. 2691–2698.
[59] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2014, pp. 818–833.

Wei Wang received the master's degree from the University of Southern Denmark. He is currently pursuing the Ph.D. degree with the Multimedia and Human Understanding Group, University of Trento, Italy. His research interests include machine learning and its application to computer vision and multimedia analysis.



Yan Yan received the Ph.D. degree from the University of Trento, Trento, Italy, in 2014. He is currently a Post-Doctoral Researcher with the Multimedia and Human Understanding Group, University of Trento. His research interests include machine learning and its application to computer vision and multimedia analysis.

Stefan Winkler is currently a Principal Scientist and the Director of the Video and Analytics Program with the University of Illinois' Advanced Digital Sciences Center, Singapore. His research interests include video processing, computer vision, perception, and human–computer interaction. He is also an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, a member of the Image, Video, and Multidimensional Signal Processing Technical Committee of the IEEE Signal Processing Society, and the Chair of the IEEE Singapore Signal Processing Chapter.

Nicu Sebe is currently a Professor with the University of Trento, Italy, leading the research in the areas of multimedia information retrieval and human-behavior understanding. He is a fellow of the International Association for Pattern Recognition. He was the General Co-Chair of the IEEE Face and Gesture Conference in 2008 and the Association for Computing Machinery (ACM) Multimedia 2013, and the Program Chair of the International Conference on Image and Video Retrieval in 2007 and 2010, and ACM Multimedia in 2007 and 2011. He is the Program Chair of European Conference on Computer Vision 2016 and International Conference on Computer Vision 2017 and a General Chair of ACM International Conference on Multimedia Retrieval 2017.