Learning Coarse-to-Fine Sparselets for Efficient Object Detection and Scene Classification

Gong Cheng[1], Junwei Han[1,*], Lei Guo[1], Tianming Liu[2]
[1]School of Automation, Northwestern Polytechnical University, Xi'an, China
[2]Department of Computer Science, The University of Georgia, Athens, GA
{gcheng,jhan,lguo}@nwpu.edu.cn, tliu@cs.uga.edu

Abstract

Part model-based methods have been successfully applied to object detection and scene classification and have achieved state-of-the-art results. More recently the "sparselets" work [1-3] were introduced to serve as a universal set of shared basis learned from a large number of part detectors, resulting in notable speedup. Inspired by this framework, in this paper, we propose a novel scheme to train more effective sparselets with a coarse-to-fine framework. Specifically, we first train coarse sparselets to exploit the redundancy existing among part detectors by using an unsupervised single-hidden-layer auto-encoder. Then, we simultaneously train fine sparselets and activation vectors using a supervised single-hidden-layer neural network, in which sparselets training and discriminative activation vectors learning are jointly embedded into a unified framework. In order to adequately explore the discriminative information hidden in the part detectors and to achieve sparsity, we propose to optimize a new discriminative objective function by imposing L0-norm sparsity constraint on the activation vectors. By using the proposed framework, promising results for multi-class object detection and scene classification are achieved on PASCAL VOC 2007, MIT Scene-67, and UC Merced Land Use datasets, compared with the existing sparselets baseline methods.

1. Introduction

Object detection and scene classification are two fundamental but challenging tasks in computer vision field. In recent years, since the seminal work of [4, 5] and with the success of deformable part models (DPMs) [6, 7], part model-based methods have demonstrated impressive performance for object detection [6-13] and scene classification [14-21]. Their success is largely owing to the usage of a number of part detectors to explicitly capture the locations, scales, and appearances of some representative and discriminative visual concepts.

However, as the number of part detectors grows (e.g., in [7] there are 960 part filters for PASCAL VOC 2007 dataset [22] and in [15] there are 13,400 part detectors for MIT Scene-67 dataset [23]), a major bottleneck to their wide applications is their heavy computational load caused by the exhaustive convolution operation between image feature pyramid and the large number of part detectors. This severely prevents them from scaling up to dealing with a large number of object or scene categories. In this situation, how to learn a universal and shared basis from a mass of part detectors is highly desirable due to the potential benefits for gaining computational efficiency.

To tackle this problem, Song *et al*. [1] introduced a new notion of "sparselets" to serve as a shared intermediate representation for multi-class object detection with DPM [7] thereby accelerating detection speed. In this application, the sparselets are defined as a universal set of shared basis learned from a number of part filters in a sparse coding framework, where each sparselet is regarded as a generic basis that is shared between all object categories. With this representation, the part filter responses of a DPM, for any object category, can be approximately reconstructed as sparse combinations of the sparselets with their corresponding activation vectors instead of exhaustive convolutions. However, as pointed out by [2] and [3], the method proposed by Song *et al*. [1] is brittle. In [1], sparselets and activation vectors were approximately obtained by using greedy orthogonal matching pursuit algorithm (OMP) [24, 25]. As sparsity increases, the errors between the original and the reconstructed part filters become larger, so the decision boundary of original models cannot be well preserved. Although these sparse coding based sparselets led to a great computational saving, they also resulted in a substantial loss in detection accuracy.

To address this drawback, [2] and [3] reformulated sparselets in a general structured output prediction framework and described a new training method to further improve the performance of [1]. The core idea of [2] and [3] is to learn which sparselets should be activated. Their solution was first learning sparselets in a sparse coding framework, then fixing the sparselets, and finally learning discriminative activation vectors for each predictor in a structural support vector machine (SVM) framework. The

*Corresponding author.

experimental results in [2] and [3] showed that discriminatively activated sparselets outperform the previous sparse coding-based sparselets significantly. However, there exists an intrinsic shortcoming in such method, that is, the discriminative activation vectors were learned by training and fixing sparselets firstly rather than fine-tuning or updating sparselets simultaneously, which did not sufficiently exploit the information hidden in the part detectors. This implies that the trained sparselets used for subsequent activation vectors learning must be as accurate as possible, but unfortunately, this requirement is very difficult to achieve in practical without jointly and simultaneously training sparselets and activation vectors.

Guided by this observation, in this paper we propose a novel solution to learn more effective sparselets for efficient multi-class object detection and scene classification. Our main contributions are as follows: First, a coarse-to-fine scheme is presented to train more effective sparselets. To be specific, we first train coarse sparselets by using an unsupervised single-hidden-layer auto-encoder to exploit the redundancy existing among part detectors. Then, we simultaneously and jointly train fine sparselets and discriminative activation vectors in a unified framework using a supervised single-hidden-layer neural network. In order to adequately explore the discriminative information hidden in the part detectors and to make the learned activation vectors to be sparse, we propose to optimize a new discriminative objective function by imposing L0-norm sparsity constraint on the activation vectors. Second, we construct and use different training samples from [1-3] to train sparselets. In [1-3] sparselets were learned using only part detectors themselves, while in our method we use high-confidence detections from all part detector to train sparselets and use the part detectors as validation set to prevent over-fitting. The variety of training samples makes the trained sparselets have good generalization ability. Third, by using the proposed framework, we obtain state-of-the-art performance for multi-class object detection and scene classification on PASCAL VOC 2007 dataset [22], MIT Scene-67 dataset [23], and UC Merced Land Use dataset [26], compared with the existing sparselets baseline methods [1-3].

2. Related Work

Part model-based object detection and scene classification: In the last few years the interest of visual recognition methods have moved from orderless models such as bag-of-words [27, 28] to part model-based methods. The impressive success of the DPMs of Felzenszwalb *et al*. [6, 7] is the most representative part model-based method for object detection, in which an object category is represented by a mixture model with six lower-resolution root filters and 48 higher-resolution part filters arranged in a flexible spatial configuration. Furthermore, some typical applications of part model-based methods on object detection can also be found in [8-13].

Inspired by the success of part model-based methods on object detection, finding discriminative visual parts to construct mid-level representation for scene classification has attracted much attention recently [14-21] and has achieved state-of-the-art results on challenging benchmarks. These advanced methods first learn a large number of discriminative part detectors (filters) and then represent images by a set of important mid-level visual elements detected by filtering in a convolution way. The mid-level visual elements are more informative than low-level visual words [27, 28] and meanwhile are easier to detect than high-level semantic objects (e.g. Object Bank [29]). For example, Doersch *et al*. [15] proposed a discriminative variant of mean-shift algorithm for finding mid-level visual elements, which learned 200 the most frequently-occurring elements per class, for a total of 13,400 part detectors, on the challenging MIT Scene-67 dataset [23], demonstrating state-of-the-art performance.

Although these part model-based methods have shown to perform very well for the tasks of object detection and scene classification, as the number of object or scene categories grows, individual part models are increasingly likely to become redundant [3]. Consequently, how to exploit the redundancy existing among the part models of multiple categories to save on computational cost has become a very urgent and important issue.

Part models sharing: The most related works and therefore also the baseline methods in our experiments are [1-3]. In [1], Song *et al*. introduced the sparselets work that implicitly shares part prototypes to provide a significant improvement in detection speed by compressing the effective number of parts used in standard DPMs. Girshick *et al*. [2] remedied the brittleness of the method in [1] and described a new framework for sparselet activation training that resulted in greater speedup factors while maintaining high task performance. In [3], Song *et al*. further integrated the preliminary version of sparselets work of [1] and [2] into a unified framework. Furthermore, Kokkinos [30] presented a method to learn a shared basis (i.e., shufflets) for the part and root filters of DPMs that are shared across different object categories by using shift-invariant sparse coding, which demonstrated better reconstructions of filters than that of the conventional sparse coding-based method. In addition, some other efforts have also been made to focus on the idea of using spared filters for visual recognition tasks, such as the work of [31-34].

Other related works will be cited throughout the paper.

3. Sparselets overview

Sparselets work was originally proposed in [1] and subsequently developed in [2] and [3] to serve as a new shared intermediate representation for the purpose of

accelerating detection speed. Briefly, sparselets is a shared dictionary $\mathbf{D}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\ldots,\boldsymbol{d}_K]\in\mathbb{R}^{m\times K}$ learned from N pre-trained part detectors $\mathbf{W}=[\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_N]\in\mathbb{R}^{m\times N}$ (to simplify the formulation we omit the bias term b), where each column $\boldsymbol{d}_k\in\mathbb{R}^m(k=1,2,\cdots,K)$ is called a sparselet, K is the sparselets dictionary size, and N is the total number of part detectors.

Part detector responses reconstruction: Denoting the histogram of oriented gradients (HOG) [35] feature pyramid of an image as Ψ, exhaustive convolution of Ψ with hundreds to thousands of part detectors is the major computational bottleneck in the tasks of part model-based object detection and scene classification. However, in the sparselets framework, the response of individual part detector $\boldsymbol{w}_i(i=1,2,\cdots,N)$ to Ψ can be approximately reconstructed as a sparse linear combination of sparselets convolutions by:

$$\Psi*\boldsymbol{w}_i\approx\Psi*\left(\sum_{k=1}^{K}\alpha_{ik}\boldsymbol{d}_k\right)=\sum_{k=1}^{K}\alpha_{ik}\left(\Psi*\boldsymbol{d}_k\right), \quad (1)$$

where $*$ denotes the convolution operator and $\boldsymbol{\alpha}_i=[\alpha_{i1},\alpha_{i2},\ldots,\alpha_{iK}]^{\mathrm{T}}\in\mathbb{R}^K$ is an activation vector of $\boldsymbol{w}_i$ with only a few nonzero elements. In this way, all part detector responses can be recovered via the following sparse matrix multiplication with the activation vectors instead of the exhaustive convolution operation [1]:

$$\Psi*\mathbf{W}=\begin{bmatrix}\Psi*\boldsymbol{w}_1\\ \Psi*\boldsymbol{w}_2\\ \vdots\\ \vdots\\ \vdots\\ \Psi*\boldsymbol{w}_N\end{bmatrix}\approx\begin{bmatrix}\boldsymbol{\alpha}_1\\ \boldsymbol{\alpha}_2\\ \vdots\\ \vdots\\ \vdots\\ \boldsymbol{\alpha}_N\end{bmatrix}\begin{bmatrix}\Psi*\boldsymbol{d}_1\\ \Psi*\boldsymbol{d}_2\\ \vdots\\ \Psi*\boldsymbol{d}_K\end{bmatrix}=\boldsymbol{\alpha}\boldsymbol{S}, \quad (2)$$

where $\boldsymbol{S}\in\mathbb{R}^K$ is a vector of all sparselet responses to Ψ, and $\boldsymbol{\alpha}=[\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\cdots,\boldsymbol{\alpha}_N]^{\mathrm{T}}\in\mathbb{R}^{N\times K}$ is a matrix of sparse activation vectors. Noting that the sparselet responses $\boldsymbol{S}=\Psi*\mathbf{D}$ are independent of any part detector, so their convolution cost can be amortized over all part detectors.

Theoretical speedup factor analysis: The speed up factor is defined as the ratio between the time needed to perform part model-based visual recognition tasks without and with the usage of sparselets framework, respectively. To be specific, let λ denote the average number of nonzero elements in $\boldsymbol{\alpha}$, for sparselets dictionary size K, sparselet dimensionality m, total number of part detectors N, an exhaustive convolution based detection scheme requires approximately Nm additions and multiplications per feature pyramid location; while the sparselets-based method only requires $Km+N\lambda$ operations, where the first term Km is a shared cost for computing the sparselet responses $\boldsymbol{S}$ and the second term $N\lambda$ is a cost for reconstructing all part detector responses in terms of $\boldsymbol{\alpha}\boldsymbol{S}$. Consequently, if we ignore the time for low-level features extraction (which actually can be pre-computed), the theoretical speedup factor η provided by the sparselets work can be written as:

$$\eta=Nm/\left(Km+N\lambda\right). \quad (3)$$

To enlarge the speedup factor, the dictionary size K should be as smaller as possible than the total number of part detectors N, and the average number of nonzero coefficients λ should be much less than the sparselet size K. Since Km is independent of the number of detectors and depends only on the dictionary size which is fixed, as the number of detectors grows, the cost of computing sparselet responses becomes fully amortized which leads to a maximum theoretical speedup of m/λ [1]. This analysis shows that the sparselets work is more applicable to part model-based multiple visual recognition methods which contain a large number of part detectors.

4. Method

As we mentioned in section 1, although the existing sparselets work [1-3] have obtained a great computational saving, some intrinsic drawbacks still exist in these methods. To learn more effective sparselets from a large number of pre-trained part detectors, in this section we propose a novel solution by constructing a coarse-to-fine training framework. Better results are obtained on the tasks of object detection and scene classification, compared with the existing sparselets baseline [1-3]. Fig. 1 gives the framework of the proposoed coarse-to-fine sparselets training, which consists of two stages: coarse sparselets training and fine sparselets and discriminative activation vectors training. In the first stage, coarse sparselets are trained to exploit the redundancy existing among different part detectors by using an unsupervised single-hidden-layer auto-encoder [36-38]. The parameters between input layers and single-hidden-layers denote the to-be-learned coarse sparselets and the number of neurons in the hidden layer corresponds to the sparselets dictionary size. In the second stage, we simultaneously train fine sparselets and discriminative activation vectors in a unified framework, using a single-hidden-layer neural network with L0-norm sparsity constraint, to adequately explore the discriminative information hidden in the part detectors.

4.1. Coarse sparselets training

The proposed coarse sparselets training is based on an unsupervised single-hidden-layer auto-encoder (SA), as shown in Fig. 1(a) and Algorithm 1. In the following text, we will use the same symbols as sparselets description rather than the conventional symbols of SA to help readers better understand our method.

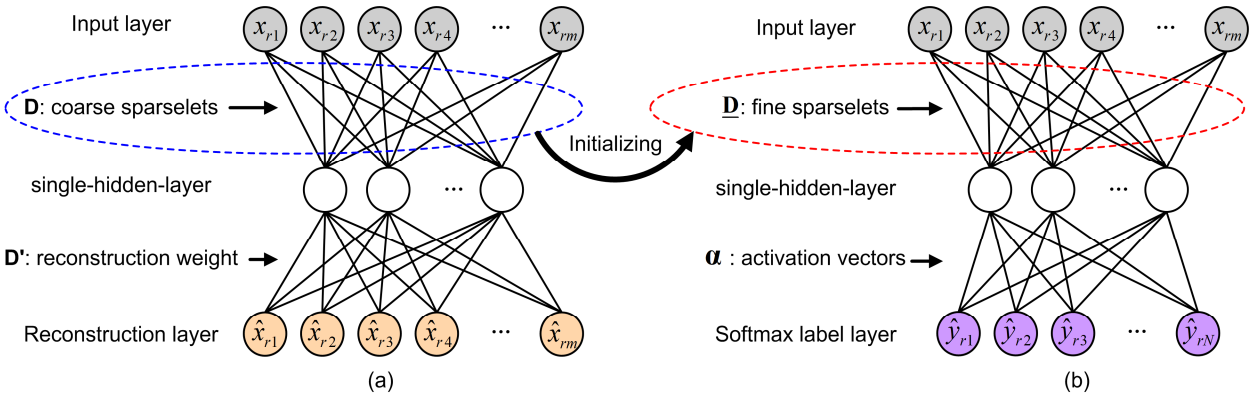

Figure 1. Framework of our proposed coarse-to-fine sparselets training: (a) coarse sparselets training based on a single-hidden-layer auto-encoder; (b) fine sparselets and discriminative activation vectors training based on a single-hidden-layer neural network.

Specifically, suppose we have N part detectors $\mathbf{W}=[\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_N]\in\mathbb{R}^{m\times N}$ and each detector has n high-confidence detections obtained from training dataset used for part detectors training, the input of the SA is Nn m-dimensional HOG features of detections while we use N part detectors for validation set to prevent over-fitting. Let $\mathbf{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{Nn}]\in\mathbb{R}^{m\times Nn}$ denote the Nn input data, $\boldsymbol{x}_r=[x_{r1},x_{r2},\cdots,x_{rm}]^{\mathrm{T}}\in\mathbb{R}^m(r=1,2,\cdots,Nn)$ denote an m-dimensional HOG feature vector of each detection and $\hat{\boldsymbol{x}}_r=[\hat{x}_{r1},\hat{x}_{r2},\cdots,\hat{x}_{rm}]^{\mathrm{T}}\in\mathbb{R}^m$ denote the reconstruction of $\boldsymbol{x}_r$, our objective is to learn $\mathbf{D}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\ldots,\boldsymbol{d}_K]\in\mathbb{R}^{m\times K}$ and $\mathbf{D}'=[\boldsymbol{d}'_1,\boldsymbol{d}'_2,\ldots,\boldsymbol{d}'_K]\in\mathbb{R}^{m\times K}$ to make the output of the reconstruction layer to be as close to the input layer as possible, i.e., $\hat{\boldsymbol{x}}_r\approx\boldsymbol{x}_r$, by minimizing the following objective function $F_1(\mathbf{D},\mathbf{D}';\boldsymbol{x})$ with activation sparsity constraint to hidden layer:

$$F_1(\mathbf{D},\mathbf{D}';\boldsymbol{x})=\frac{1}{2Nn}\sum_{r=1}^{Nn}\|\boldsymbol{x}_r-\hat{\boldsymbol{x}}_r\|_2^2+\beta\sum_{k=1}^{K}\mathrm{KL}(\rho\,\|\,\hat{\rho}_k), \quad (4)$$

$$\hat{\boldsymbol{x}}_r=\frac{\mathbf{D}'}{1+\exp(-\mathbf{D}^{\mathrm{T}}\boldsymbol{x}_r)}, \quad (5)$$

$$\mathrm{KL}(\rho\,\|\,\hat{\rho}_k)=\rho\log\frac{\rho}{\hat{\rho}_k}+(1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_k}, \quad (6)$$

$$\hat{\rho}_k=\frac{1}{Nn}\sum\nolimits_{r=1}^{Nn}\left(1+\exp(-\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{x}_r)\right)^{-1}, \quad (7)$$

where $\mathbf{D}$ is the to-be-learned coarse sparselets with its elements subjected to $\|\boldsymbol{d}_k\|_2=1(k=1,2,\cdots,K)$, $\mathbf{D}'$ is a reconstruction weight matrix used to reconstruct the input layer from the hidden layer, K is the number of neurons in the hidden layer which corresponds to the sparselets dictionary size, β is the weight of the sparsity penalty, ρ is the target average activation of the hidden nodes, and $\hat{\rho}_k$ is the average activation of the k-th hidden node over the Nn training data. The Kullback-Leibler divergence $\mathrm{KL}(\cdot)$ is a standard function for measuring how different two distributions are, which provides the sparsity constraint. Here we set $\beta=3$ and ρ=0.05 as suggested in [36].

We can easily see that the objective function given by (4) mainly measures an average reconstruction error between the input data $\boldsymbol{x}_r$ and the reconstruction data $\hat{\boldsymbol{x}}_r$. If the model achieves a good reconstruction using $\mathbf{D}$ and $\mathbf{D}'$, we can be sure that the learned sparselets have preserved most of the information of part detectors. In practice, we solve this optimization problem by using the L-BFGS algorithm [39] which enables to address large-scale data with limited memory. Details of the solution can be found in many related works [36].

4.2. Fine sparselets and discriminative activation vectors training

Notice that we have a large number of training examples with confident part detector labels. In order to incorporate this information to sufficiently exploit the discriminative information hidden in the training examples, we propose to further fine-tune the trained coarse sparselets to enhance their generalization capability and simultaneously train discriminative activation vectors in a unified framework, by building a supervised single-hidden-layer neural network (SNN), as illustrated in Fig. 1(b) and Algorithm 2. Meanwhile, to make the learned activation vectors to be discriminative and sparse, we propose to optimize a new objective function by imposing L0-norm sparsity constraint on the activation vectors.

Different from reconstruction layer of Fig. 1(a), the output layer is now a binary vector with a softmax unit that allows one element to be 1 out of N-dimensions for N-way classification problem. The fine sparselets are now not only learned from reconstructing the input data, but also from a softmax classifier predicting the labels. A discriminative objective function computes an average classification loss

Algorithm 1 Coarse sparselets training based on SA

Input: a library of part detectors $\mathbf{W}=[\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_N]$ and their corresponding Nn high-confidence detections with HOG features $\mathbf{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{Nn}]$

Output: a dictionary of coarse sparselets $\mathbf{D}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\ldots,\boldsymbol{d}_K]$

1: **begin**
2: Initialize $\mathbf{D}$ with its elements subjected to $\|\boldsymbol{d}_k\|_2=1$
3: Initialize reconstruction weight $\mathbf{D}'=[\boldsymbol{d}'_1,\boldsymbol{d}'_2,\ldots,\boldsymbol{d}'_K]$
4: **while** stopping criterion has not been met **do**
5: compute average activation $\hat{\rho}_k$ and $\mathrm{KL}(\rho\|\hat{\rho}_k)$ for each hidden node using (7) and (6)
6: compute reconstruction layer outputs $\hat{\boldsymbol{x}}_r$ using (5)
7: compute objective function $F_1(\mathbf{D},\mathbf{D}';\boldsymbol{x})$ using (4)
8: update $\mathbf{D}$ and $\mathbf{D}'$ using L-BFGS algorithm
9: **end while**
10: **return** $\mathbf{D}$
11: **end begin**

between the actual label $\boldsymbol{y}_r=[y_{r1},y_{r2},\cdots,y_{rN}]^{\mathrm{T}}\in\mathbb{R}^N$ and the predicted label $\hat{\boldsymbol{y}}_r=[\hat{y}_{r1},\hat{y}_{r2},\cdots,\hat{y}_{rN}]^{\mathrm{T}}\in\mathbb{R}^N$. By imposing L0-norm sparsity constraint on the activation vectors, the new discriminative objective function $F_2(\underline{\mathbf{D}},\boldsymbol{\alpha};\boldsymbol{x},\boldsymbol{y})$ can be rewritten as:

$$F_2(\underline{\mathbf{D}},\boldsymbol{\alpha};\boldsymbol{x},\boldsymbol{y})=\frac{1}{2Nn}\sum_{r=1}^{Nn}\|\boldsymbol{y}_r-\hat{\boldsymbol{y}}_r\|_2^2+Z(\boldsymbol{\alpha}), \quad (8)$$

$$\hat{\boldsymbol{y}}_r=\mathrm{softmax}\left(\boldsymbol{\alpha}\underline{\mathbf{D}}^{\mathrm{T}}\boldsymbol{x}_r\right), \quad (9)$$

$$Z(\boldsymbol{\alpha})=\frac{\gamma}{2}\sum_{i=1}^{N}\|\boldsymbol{\alpha}_i\|_2^2 \ \text{ s.t. } \ \|\boldsymbol{\alpha}_i\|_0\le\lambda,\forall i=1,2,\cdots,N, \quad (10)$$

where $\underline{\mathbf{D}}=[\underline{\boldsymbol{d}}_1,\underline{\boldsymbol{d}}_2,\ldots,\underline{\boldsymbol{d}}_K]\in\mathbb{R}^{m\times K}$ is the to-be-learned fine sparselets with its elements subjected to $\|\underline{\boldsymbol{d}}_k\|_2^2=1\,(k=1,2,\cdots,K)$, $\boldsymbol{\alpha}$ is the to-be-learned discriminative activation vectors, γ is a weight decay parameter controls the relative importance of the two terms which is set to be 0.001 as suggested in [36], λ is the number of nonzero elements in each activation vector, and $\mathrm{softmax}(a_i)=\exp(a_i)/\sum_{i'}\exp(a_{i'})(i=1,2,\cdots,N;a\in\mathbb{R}^N)$.

In the new objective function of (8), the first term is a supervised goal ensuring the learned sparselets to be discriminative between different part detectors. The second term is a regularization term that tends to decrease the magnitude of the activation vectors and helps to prevent over-fitting, while with L0-norm constraint to achieve sparsity. Similar to coarse sparselets training, we solve this optimization problem by using the L-BFGS algorithm [39].

However, as the second term is an NP-hard optimization problem, we adopt a similar solution as [2] to approximately minimize it by employing a two-step process. To be specific, in the first step, based on the learned coarse sparselets $\mathbf{D}$, we initialize the activation vectors $\boldsymbol{\alpha}$ by minimizing the average reconstruction error between all part detectors and their reconstruction approximation via the following formulation:

$$\min\frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{w}_i-\mathbf{D}\boldsymbol{\alpha}_i\|_2^2 \ \text{ s.t. } \|\boldsymbol{\alpha}_i\|_0\le\lambda,\forall i=1,2,\cdots,N. \quad (11)$$

In our work, we use the OMP algorithm [24, 25, 40] implemented in the SPArse Modeling Software (SPAMS) package [40] to optimize Eq. (11). In the second step, the initialization of nonzero variables is fixed, which leads to the satisfaction of the sparsity constraint and results in a convex optimization problem to solve. We then learn the selected variables discriminatively according to (8).

Algorithm 2 Fine sparselets and discriminative activation vectors training based on SNN

Input: a library of N part detectors $\mathbf{W}=[\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_N]$ and their corresponding Nn high-confidence detections with HOG features $\mathbf{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{Nn}]$ and their labels $\mathbf{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_{Nn}]$, coarse sparselets $\mathbf{D}$

Output: a dictionary of fine sparselets $\underline{\mathbf{D}}=[\underline{\boldsymbol{d}}_1,\underline{\boldsymbol{d}}_2,\ldots,\underline{\boldsymbol{d}}_K]$, and discriminative activation vectors $\boldsymbol{\alpha}=[\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\cdots,\boldsymbol{\alpha}_N]^{\mathrm{T}}$

1: **begin**
2: Initialize $\underline{\mathbf{D}}$ to be the same as $\mathbf{D}$
3: Initialize activation vectors $\boldsymbol{\alpha}$ using (11)
4: **while** stopping criterion has not been met **do**
5: compute the predicted label $\hat{\boldsymbol{y}}_r$ using (9)
6: compute regularization term $Z(\boldsymbol{\alpha})$ using (10)
7: compute objective function $F_2(\underline{\mathbf{D}},\boldsymbol{\alpha};\boldsymbol{x},\boldsymbol{y})$ using (8)
8: update $\underline{\mathbf{D}}$ and $\boldsymbol{\alpha}$ using L-BFGS algorithm
9: **end while**
10: **return** $\underline{\mathbf{D}}$ and $\boldsymbol{\alpha}$
11: **end begin**

5. Experiments

We comprehensively evaluated our method on three well known benchmarks: the PASCAL VOC 2007 dataset [22], the MIT Scene-67 dataset [23], and the UC Merced Land Use dataset [26], where the former one is used for 20-class object detection and latter two are used for 67-class indoor scene classification and 21-class aerial scene classification. The performance of object detection and scene classification is evaluated by mean Average Precision (AP) and average classification accuracy (ACA) over all object classes and scene classes in each dataset, respectively.

5.1. Multi-class object detection

We first validate our coarse-to-fine sparselets training method on multi-class object detection with DPMs on PASCAL VOC 2007 dataset [22], which is the most

popular dataset for object detection consisting of 9,963 images of 20 different object classes with 5,011 training images and 4,952 testing images. The task is to predict bounding boxes of the objects of interest if they are present in the images. As the best object detection performance reported in [41, 42] is obtained based on DPMs with the combination of other attributes, we mainly focus on demonstrating our learning method for accelerating DPM detection speed. It can be implied that state-of-the-art detection accuracy can be achieved if we use the proposed method to replace DPMs in [41, 42].

Like [1] we used the off-the-shelf part filters of DPMs from voc-release4 [43] to train the sparselets for our method and [1, 2] (the results of [2] and [3] are the same so we only report one of them and this is the same for the following scene classification), and meanwhile, we set the sparselet size to be the same as the part filter size. In [43], there are total 960 part filters with the same 6×6 HOG cells (1152-dimension). We set sparsity level (i.e., the rate of zero entries in the matrix of sparse activation vectors $\boldsymbol{\alpha}$) to be 0.9 and set sparselets dictionary size K to be the set of $\{100,150,200,250,300\}$. Table 1 and Fig. 2 present the mean AP values and actual speedup factors (averaged over 3 PCs with different configurations and this is the same for the following scene classification) obtained with three different sparslets training methods, together with the DPM baseline. As can be seen from Table 1 and Fig. 2, among the three sparselets training methods, our method obtained the best results under the same speedup factor. Especially, when K=300, we obtained almost the same accuracy as original DPM baseline yet with a 1.59 speedup factor. Considering the small number of object classes (only 20) on PASCAL VOC 2007 dataset, the potential benefit of computation gain is very considerable and appealing for bigger dataset such as ImageSearch-100k [44] which contains 100,000 object concepts and more than a million part filters.

Meanwhile, it should be pointed out that the speedup factors are obtained by only using our sparselets framework, higher speedup factors can be easily obtained by combining our method with other existing accelerating algorithms, such as Cascade [45], FFT [46], branch-and-bound [47], coarse-to-fine [12], winner-take-all hashing [44] and so on.

5.2. Scene classification

MIT Scene-67 dataset This is a dataset of 15,620 images over 67 indoor scenes assembled by [23]. On this dataset, we followed the original experimental setting in [23] which is widely adopted by [14, 15, 17, 18, 20, 21, 23, 29, 31, 48-53], where each scene category has about 80 training and 20 test images. Since [15] has demonstrated state-of-the-art performance on this challenging dataset by learning 200 part detectors for the most frequently occurring elements per class, for a total of 13,400 part detectors, in our work we directly used these off-the-shelf part detectors of [15] to train the sparselets for our method and [1-3] and then implemented the scene classification using the framework in [15]. The detectors in [15] are all with the same 6×6 HOG cells (1188-dimension).

Table 1. Results on PASCAL VOC 2007 dataset [22]

Methods	mean AP (%)	Speedup factor
DPM voc-release4 [7]	32.30	baseline
Reconstructive sparselets [1]	24.13-30.64	2.43-1.59
Discriminative sparselets [2]	29.21-31.89	
Our coarse-to-fine sparselets	30.32-32.25	

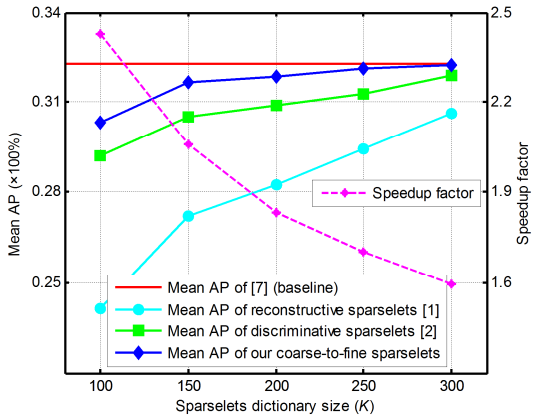

Figure 2. Mean AP (the left Y-axis) and speedup factor (the right Y-axis) obtained with different methods and different sparselets dictionary sizes on PASCAL VOC 2007 dataset.

In our experiments, we set sparsity level to be 0.9 and set sparselets dictionary size K to be the set of $\{100,200,300,400,500\}$. Fig. 3 shows the ACA and actual speedup factors at various sparselets dictionary sizes, obtained with three different sparslets training methods and the baseline of [15]. As our goal is to validate the effectiveness of the proposed sparselets training method, Table 2 only summarizes the results of various state-of-the-art single-feature approaches [14, 15, 17, 18, 20, 21, 23, 29, 31, 48-53], together with our and the existing sparselets methods [1, 2]. Higher accuracy can be easily achieved by combining our method with the currently best-performing methods. As can be seen from Fig. 3 and Table 2: (1) With the same speedup factors at all sparselets dictionary sizes, our proposed method achieved much better performance than the existing sparselets methods of [1] and [2] in terms of ACA. (2) Our highest accuracy could outperform the state-of-the-art baseline method of [15] by 0.33% while with a significant speedup factor of more than 7.75.

UC Merced Land Use dataset To demonstrate the stability and adaptability of our method, the UC Merced

Table 2. Results on MIT Scene-67 dataset [23]

Methods	ACA (%)	Speedup factor
ROI + Gist [23]	26.05	--
MM-scene [53]	28.00	--
DPM [20]	30.40	--
CENTRIST [52]	36.90	--
Object Bank [29]	37.60	--
RBoW [51]	37.93	--
D-patches [18]	38.10	--
LPR [50]	44.84	--
BoP [17]	46.10	--
VC [14]	46.40	--
ISPR [48]	50.10	--
MMDL [49]	50.15	--
D-part detectors [21]	51.40	--
DSFL [31]	52.24	--
Mid-level visual elements [15]	64.03	baseline
Reconstructive sparselets [1]	43.12-56.45	24.72-7.75
Discriminative sparselets [2]	53.84-61.21	
Our coarse-to-fine sparselets	59.87-64.36	

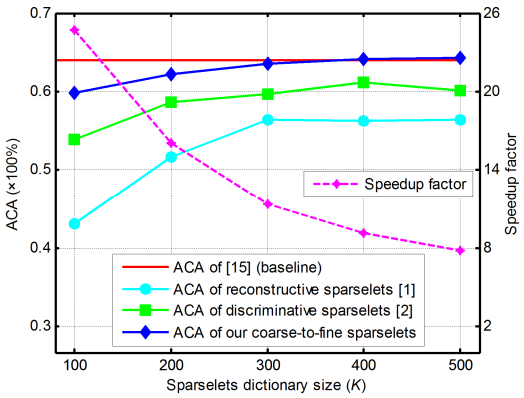

Figure 3. ACA (the left Y-axis) and speedup factor (the right Y-axis) obtained with different methods and different sparselets dictionary sizes on MIT Scene-67 dataset.

Land Use dataset [26] was selected in our validation experiments. Different from PASCAL VOC 2007 dataset [22] and MIT Scene-67 dataset [23] in which images are all natural scene (non-overhead view) images, this is a new yet challenging high-spatial-resolution overhead images dataset collected by [26], which consists of 21 aerial scenes with complex geospatial objects and various spatial structures. Each class contains 100 images of 256×256 pixels in a resolution of one foot per pixel. Five-fold cross-validation was adopted to report the classification results according to the experimental setting in [26].

We adopted the work of [16] as our baseline, in which there are total 3,093, 3,140, 3,072, 3,110, and 2,822 part detectors on all five held-out sets, respectively. In our work we directly used these detectors to train the sparselets for our method and [1-3] and then implemented the scene classification using the framework in [16].

Table 3. Results on UC Merced Land Use dataset [26]

Methods	ACA (%)	Speedup factor
BOVW [26]	71.86	--
SCK [28]	72.52	--
SPCK++ [26]	77.38	--
BRSP [54]	77.80	--
COPD [16]	91.33	baseline
Reconstructive sparselets [1]	64.52-83.24	14.78-4.44
Discriminative sparselets [2]	79.32-88.02	
Our coarse-to-fine sparselets	85.23-91.46	

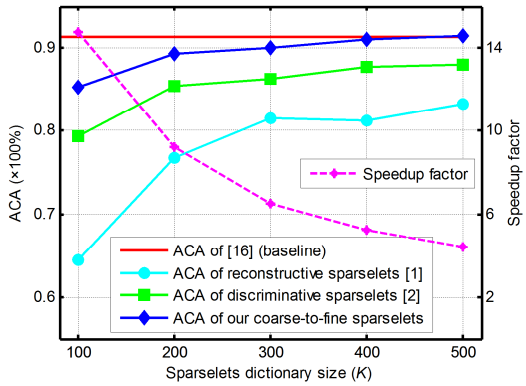

Figure 4. ACA (the left Y-axis) and speedup factor (the right Y-axis) obtained with different methods and different sparselets dictionary sizes on UC Merced Land Use dataset.

The same as MIT Scene-67 dataset, we set sparsity level to be 0.9 and set sparselets dictionary size K to be the set of $\{100,200,300,400,500\}$. Fig. 4 shows the ACA and actual speedup factors at various sparselets dictionary sizes, obtained with three different sparslets training methods and the baseline of [16]. Table 3 lists the results of some previously published methods [26, 28, 54], together with our and the existing sparselets methods [1, 2]. As can be seen from Fig. 4 and Table 3: (1) Compared with the existing sparselets methods [1, 2], our method obtained the best classification accuracy at all sparselets dictionary sizes. (2) When we set K to be 400 and 500, our method yielded almost the same accuracy as the baseline method of [16] (our highest accuracy even outperformed [16] slightly) yet with big computational gains.

6. Conclusions

In this paper, we proposed a novel scheme to learn more effective sparselets by constructing a coarse-to-fine training framework, which can be used for efficient part model based object detection and scene classification. In this new framework, coarse sparselets were firstly trained to exploit the redundancy that exists among different part

detectors by using an unsupervised single-hidden-layer auto-encoder. Then, fine sparselets and discriminative activation vectors were jointly and simultaneously trained in a unified framework by building a supervised single-hidden-layer neural network and optimizing a new discriminative objective function with L0-norm sparsity constraint, making the information hidden in the training samples to be exploited adequately.

By using the proposed framework, promising results were achieved on PASCAL VOC 2007 dataset [22], MIT Scene-67 dataset [23], and UC Merced Land Use dataset [26], compared with the existing sparselets baseline methods [1-3]. Experimental results demonstrate that the proposed coarse-to-fine training framework opens a new window for future subsequent development of sparselets work, showing sparselets-based methods' huge potential for fast and accurate visual recognition tasks.

Acknowledgements

This work was supported in part by the National Science Foundation of China under Grant 61401357, Grant 61473231, and Grant 61333017, and in part by the China Postdoctoral Science Foundation under Grant 2014M552491.

References

[1] H. O. Song, S. Zickler, T. Althoff, R. Girshick, M. Fritz, C. Geyer, P. Felzenszwalb, and T. Darrell. Sparselet models for efficient multiclass object detection. In *ECCV*, 2012.

[2] R. Girshick, H. O. Song, and T. Darrell. Discriminatively activated sparselets. In *ICML*, 2013.

[3] H. Song, R. Girshick, S. Zickler, C. Geyer, P. Felzenszwalb, and T. Darrell. Generalized sparselet models for real-time multiclass object recognition. *TPAMI,* 2014.

[4] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Comput.,* 22(1): 67-92, 1973.

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV,* 61(1): 55-79, 2005.

[6] P. Felzenszwalb, D. Mcallester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI,* 32(9): 1627-1645, 2010.

[8] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009.

[9] I. Endres, K. J. Shih, J. Jiaa, and D. Hoiem. Learning collections of part models for object recognition. In *CVPR*, 2013.

[10] G. Cheng, J. Han, L. Guo, X. Qian, P. Zhou, X. Yao, and X. Hu. Object detection in remote sensing imagery using a discriminatively trained mixture model. *ISPRS J. Photogramm. Remote Sens.,* 85: 32-43, 2013.

[11] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[12] M. Pedersoli, A. Vedaldi, and J. Gonzalez. A coarse-to-fine approach for fast deformable object detection. In *CVPR*, 2011.

[13] J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In *CVPR*, 2014.

[14] Q. Li, J. Wu, and Z. Tu. Harvesting mid-level visual concepts from large-scale internet images. In *CVPR*, 2013.

[15] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *NIPS*, 2013.

[16] G. Cheng, J. Han, P. Zhou, and L. Guo. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS J. Photogramm. Remote Sens.,* 98: 119-132, 2014.

[17] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: distinctive parts for scene classification. In *CVPR*, 2013.

[18] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012.

[19] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes Paris look like Paris? In *SIGGRAPH*, 2012.

[20] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011.

[21] J. Sun and J. Ponce. Learning discriminative part detectors for image classification and cosegmentation. In *ICCV*, 2013.

[22] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV,* 88(2): 303-338, 2010.

[23] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009.

[24] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.,* 41(12): 3397-3415, 1993.

[25] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.,* 11: 19-60, 2010.

[26] Y. Yang and S. Newsam. Spatial pyramid co-occurrence for image classification. In *ICCV*, 2011.

[27] F. F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[28] Y. Yang and S. Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *ACM SIGSPATIAL GIS*, 2010.

[29] L. Li, H. Su, E. P. Xing, and F. Li. Object Bank: a high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010.

[30] I. Kokkinos. Shufflets: shared mid-level parts for fast object detection. In *ICCV*, 2013.

[31] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, and X. Jiangy. Learning discriminative and shareable features for scene classification. In *ECCV*, 2014.

[32] H. Pirsiavash and D. Ramanan. Steerable part models. In *CVPR*, 2012.

[33] P. Ott and M. Everingham. Shared parts for deformable part-based models. In *CVPR*, 2011.

[34] M. Jaderberg, A. Vedaldi, and A. Zisserman. Speeding up convolutional neural networks with low rank expansions. In *BMVC*, 2014.

[35] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[36] A. Ng. CS294A lecture notes: Sparse autoencoder. Stanford University, 2010.
[37] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science,* 313(5786): 504-507, 2006.
[38] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature,* 323: 533-536, 1986.
[39] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Math. Comput.,* 35(151): 773-782, 1980.
[40] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, 2009.
[41] W. Voravuthikunchai, B. Crémilleux, and F. Jurie. Histograms of pattern sets for image classification and object recognition. In *CVPR*, 2014.
[42] F. Shahbaz Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez. Color attributes for object detection. In *CVPR*, 2012.
[43] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/~pff/latent-release4/.
[44] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *CVPR*, 2013.
[45] P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *CVPR*, 2010.
[46] C. Dubout and F. Fleuret. Exact acceleration of linear object detectors. In *ECCV*, 2012.
[47] I. Kokkinos. Rapid deformable object detection using dual-tree branch-and-bound. In *NIPS*, 2011.
[48] D. Lin, C. Lu, R. Liao, and J. Jia. Learning important spatial pooling regions for scene classification. In *CVPR*, 2014.
[49] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu. Max-margin multiple-instance dictionary learning. In *ICML*, 2013.
[50] F. Sadeghi and M. F. Tappen. Latent pyramidal regions for recognizing scenes. In *ECCV*, 2012.
[51] S. N. Parizi, J. G. Oberlin, and P. F. Felzenszwalb. Reconfigurable models for scene recognition. In *CVPR*, 2012.
[52] J. Wu and J. M. Rehg. CENTRIST: A visual descriptor for scene categorization. *TPAMI,* 33(8): 1489-1501, 2011.
[53] J. Zhu, L.-J. Li, F.-F. Li, and E. P. Xing. Large margin learning of upstream scene understanding models. In *NIPS*, 2010.
[54] Y. Jiang, J. Yuan, and G. Yu. Randomized spatial partition for scene recognition. In *ECCV*, 2012.