Noname manuscript No.
(will be inserted by the editor)

Learning Dictionary of Discriminative Part Detectors for Image Categorization and Cosegmentation

Jian Sun · Jean Ponce

Received: date / Accepted: date

Abstract This paper proposes a novel approach to learning mid-level image models for image categorization and cosegmentation. We represent each image class by a dictionary of part detectors that best discriminate that class from the background. We learn category-specific part detectors in a weakly supervised setting in which the training images are only annotated with category labels without part / object location information. We use a latent SVM model regularized using the $\ell_{2,1}$ group sparsity norm to learn the part detectors. Starting from a large set of initial parts, the group sparsity regularizer forces the model to jointly select and optimize a set of discriminative part detectors in a max-margin framework. We propose a stochastic version of a proximal algorithm to solve the corresponding optimization problem. We apply the learned part detectors to image classification and cosegmentation, and present extensive comparative experiments with standard benchmarks.

1 Introduction

Learning mid-level image representations is a promising approach to improving the performance of image recognition systems. Traditional recognition systems model the set of low-level features (e.g., SIFT [46], HOG [17]) by a mid-level bag-of-words model [16], sparse codes [81], Fisher vectors [63], etc. These approaches generally represent an image by a fixed-length image code through quantitizing the low-level feature space, then feed these image codes to classifiers for image recognition. They have been shown to be effective for image recognition.

Another category of popular mid-level representation decomposes objects or scenes into parts [2,8,20,26,49,67], and each part covers a discriminative region of an object / image, e.g., the head of dogs, the rear of cars. Successful examples of part-based models include deformable part models (DPMs) [26], poselets [8], discriminative patches [19,49,67] for object detection [26], action recognition [82], semantic segmentation [2], scene classification [19,49,67], etc.

Learning part-based models has, however, been a challenge. The essential question is how to efficiently learn and select object / image parts that are discriminative for an image / object category. The deformable part model (DPM) [26] learns a mixture of object templates in different poses represented by a few spatially deformable object parts using a discriminative latent-SVM learning framework. The positions and number of parts are heuristically initialized given the object bounding box. Other recent methods learn a much larger set of discriminative part detectors. For example, in the case of poselets [7,8] and discriminative patch (DP) models [19,20,67], a large number of part detectors are first learned by linear SVMs from image patch clusters. Discriminative parts are then selected by ranking the importance of image parts and discarding the unimportant ones. In the case of poselets, additional supervision in the form of keypoint labels is necessary.

In this work, we propose a principled approach to learning class-specific part detectors inspired by dictionary learning [24,47]. As illustrated by Figure 1, given a set of training images from the same category (Figure 1 (a)), we design a novel latent SVM model regularized by group sparsity to jointly select and optimize a set of discriminative part detectors. Given a large set of initial parts, the group sparsity regularizer forces the model to automatically select and

J. Sun
No.28, Xianning West Road, Xi'an, Shaanxi, 710049, P. R. China
Tel.: +86-29-82663153
E-mail: jiansun@mail.xjtu.edu.cn

J. Ponce
École Normale Supérieure / PSL Research University, 45, Rue d'Ulm, 75005 Paris, France.
E-mail: Jean.Ponce@ens.fr

(a) Example images of “Car” category. (b) Examples of learned part detectors.

Fig. 1 We learn discriminative part detectors for an image set with the same category label. The part detectors are applied to image classification and cosegmentation. (*Best viewed in color.*)

optimize a dictionary of discriminative part detectors. Our model tends to select the parts that appear more frequently and strongly in positive training images than in the negative ones. Examples of learned part detectors are shown in Figure 1 (b).

With our approach, part detectors are learned to reliably detect the image parts that best discriminate the category of interest from the background world. We have applied the learned part detectors to image classification and cosegmentation. For image classification, we encode an image using a fixed-length mid-level code by max-pooling the responses of the learned part detectors to the image, and achieve competitive performance in object, scene, and event classification over benchmark databases. We have observed that our discriminative part detectors are able to find the common object parts from a set of images containing the same object class, and therefore also propose a novel cosegmentation model in a discriminative clustering framework, using the object cues provided by the learned part detectors. We also report state-of-the-art results on large scale image benchmark datasets.

A preliminary version of this work appeared in [70]. This presentation extends [70] as follows: First, it provides more implementation details and several new experiments. We have also compared our approach to more recently published approaches. Second, we have re-implemented our algorithm for learning part detectors to take advantage of multi-scale image pyramids during training, and accordingly report improved classification results. Third, we have significantly extended the cosegmentation algorithm and achieved competitive cosegmentation results on large image sets by introducing an improved cosegmentation model. Fourth, we have tested the effect of the different part initialization methods on the recognition performance. Finally, the source codes of our algorithm are now publicly available online [1].

Nowadays, convolutional neural networks (CNN) [15, 39, 84] are popular in image recognition. Our part learning model is similar to a convolutional layer with ReLU transform [39] and max-pooling over the whole image region, followed by an SVM loss. But our part learning model is defined over HOG features, and regularized by a group sparsity term. More discussions on the relationship of our approach to CNNs are presented in Section 6.

1.1 Related Work on Image Representation for Recognition

Traditional image models are primarily based on quantized low-level features, including, bags of words (BoWs) [16], sparse coding [81], Fisher vectors [63], LLC coding [75], etc. The image is represented by spatially pooling the corresponding codes globally on a coarse grid or a spatial pyramid [41] for image classification. These approaches have achieved excellent results for image recognition. Contrary to them, however we model object or image caregories using a learned dictionary of discriminative mid-level parts in diverse poses / viewpoints.

1.1.1 Part-based Models

There is a large body of work on part-based models for recognition. The deformable part model (DPM) [26, 54] represents an object by a set of deformable parts organized in a tree structure and learned from object bounding boxes.

[1] `https://github.com/exploreman/discriminative_parts`

Strongly-supervised DPMs [3] further incorporated human-annotated object parts to improve performance. Weakly supervised DPMs [54] learn the deformable parts using only image-level categories as labels. Recently, DPMs [27] have been shown to be representable by convolutional neural networks, therefore the feature extractors and deformable parts can be learned by an end-to-end training procedure.

Learning a collection of part detectors has been an interesting topic in recognition. Reconfigurable models [56] learn a set of regions representing scenes. In poselets [7,8], a large number of object parts are learned and selected using SVMs trained over clusters of image patches with the aid of human-labelled 3D keypoints in different poses. Discriminative patch (DP) methods learn distinctive image patches using discriminative clustering [67] or extended mean-shift mode seeking [19]. In [1], discriminative HOG filters are selected from poselets or examplar SVMs by learning a ranking function with diversity constraints. The above approaches separately learn a set of part detectors using linear SVMs and select the distinctive ones by ranking their importance.

Compared to these approaches, ours has the following characteristics: First, instead of learning reconfigurable or deformable parts as in [54,56], we learn a large collection of category-specific part detectors, which are more flexible in capturing the diverse scene or object parts. Second, instead of separately learning and selecting part detectors [1, 8,19,67], our model provides a unified framework to jointly learn and select a dictionary of category-specific part detectors. The group sparsity regularizer plays the role of part selector, and allows us to select diverse part detectors best discriminating the positive training examples from negative background. Third, compared to poselet or examplar-SVMs, our approach works in a weakly supervised way and only requires training examples at the category level without any manually labeled keypoints or bounding boxes. In [57], a shared dictionary of object parts is learned for multiple object categories. This method is similar to ours in its use of group sparsity regularization for part selection, but it was published after the conference version [70] of our work, and we focus on learning class-specific part detectors.

1.1.2 Dictionary Learning

Our approach is also related to dictionary learning [24,31, 48,53], where image patches are encoded as a sparse linear combination of basis (dictionary) elements, optimized for image reconstruction [24,53] or classification [31,48]. Contrary to these approaches, we use the collection of part detectors themselves as dictionary, and represent each image by the total response of the part detectors, as will be discussed in Section 3.2. Our learning problem is modeled using a latent SVM with group sparsity regularization, which is significantly different from the discriminative sparse coding models of [31,48].

1.2 Related Work on Cosegmentation

Cosegmentation [13,37,51,74] is the problem of jointly dividing a set of images assumed to share the same type of prominent (foreground) objects into foreground and background regions. It is challenging since it only involves a weak form of supervision, i.e., the fact that images contain instances of the same object category. Its multi-class extensions [34,36] try to identify multiple prominent classes of objects in image collections. Recently, Rubinstein et al. [61] have also proposed a weakly supervised object segmentation approach to identify objects in large image sets collected from the Internet, which is beyond the capabilities of conventional cosegmentation approaches.

Related approaches include weakly supervised algorithms for semantic segmentation [72,73] and the segmentation propagation algorithm in ImageNet [40]. In the two-class case, weakly supervised semantic segmentation reduces to cosegmentation with known foreground category. In the multi-class case, it uses more supervision, i.e., the class labels for each image, than multi-class cosegmentation. The segmentation propagation method [40] propagates labels from annotated images to a large-scale image set in a semi-supervised manner.

In this paper, we address the original cosegmentation problem. We take the image set containing instances of the same object class as positive training data, and external background images as negative training data. Our approach can learn a dictionary of object part detectors which are discriminative and frequently appear in the positive training images. These part detectors provide object localization cues for object cosegmentation. In our approach, we use negative data when learning the part detectors, since negative data mining has been shown to be effective for learning discriminative features for object detection [26,68].

The rest of our paper is organized as follows. Section 2 defines our part model. Section 3 presents our model for learning discriminative part detectors. Sections 4 and 5 discuss the applications of discriminative part detectors to image classification and object cosegmentation respectively. Section 6 discusses the relationship between our work and CNN. Section 7 concludes the paper with a brief discussion.

2 Part Detector Definition

Given an image I, let us consider dense features extracted at fixed intervals over the image grid. An *image part* is a box whose top-left corner is positioned at z, and it is represented by a feature vector $\Phi(I,z)$ that concatenates all the feature

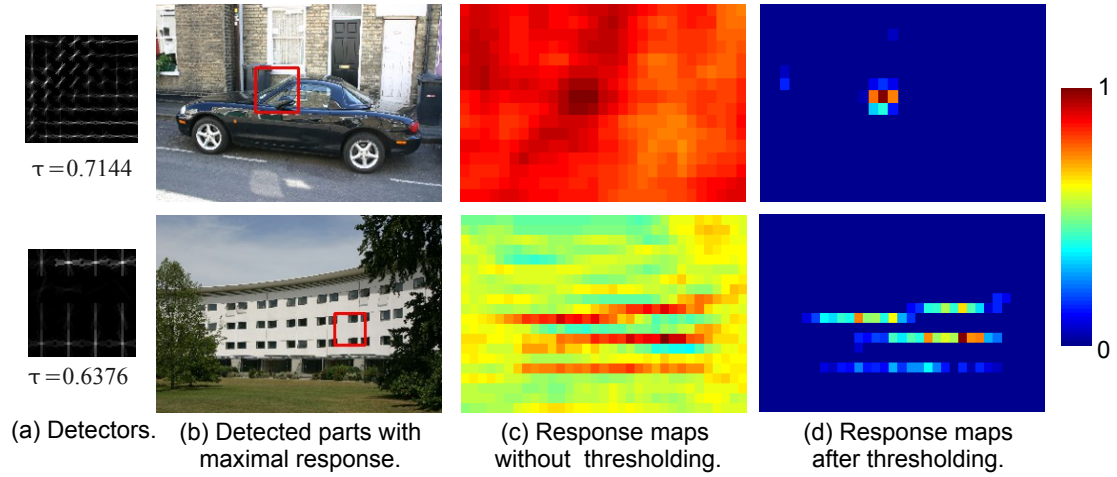


Fig. 2 Examples of part detectors. With the learned part thresholds, part detectors can produce clean responses to images. (*Best viewed in color.*)

vectors within the box. We further define a *part detector* $\Gamma_k = (\beta_k, \tau_k)$ $(k = 1, \cdots, K)$ as a *template / threshold* pair (β_k, τ_k), and define its response to image part $\Phi(I, z)$ as

$$r_z(\Gamma_k, I) = [S(\beta_k, \Phi(I, z)) - \tau_k]_+, \tag{1}$$

where $[a]_+ = \max(a, 0)$, and $S(\beta_k, \Phi(I, z))$ is the *matching score* between the part template β_k and the image part $\Phi(I, z)$. In this work, we simply define the matching score as the inner product between part template and normalized part feature vector:

$$\begin{aligned} S(\beta_k, \Phi(I, z)) &= \frac{< \beta_k, \Phi(I, z) >}{||\Phi(I, z)||_2} \\ &= < \beta_k, \frac{1}{||\Phi(I, z)||_2} \Phi(I, z) > . \end{aligned} \tag{2}$$

The normalization of part features prevents the part detector from being biased to have higher response to image box's feature vector with larger norm.

Based on Eq.(1), the part detector Γ_k has non-zero response to image I at position z only when the matching score $S(\beta_k, \Phi(I, z))$ is higher than τ_k. Furthermore, we say that the part Γ_k *appears in an image* I when there exists at least one position z that satisfies $r_z(\Gamma_k, I) > 0$. Figure 2 shows examples of part detectors and the corresponding responses. As shown by this figure, after thresholding the matching scores using Eq.(1), irrelevant image parts are suppressed and only significantly similar image parts have non-zero responses.

3 Learning Part Detectors using Group Sparsity

In this section, we aim to learn a set of category-specific image part detectors that can best discriminate the images in the category of interest from the background images. As shown by Figure 3, the input of our approach is an image set composed of positive and negative training examples. First, we automatically pick an initial set of candidate part detectors associated with the image category. They frequently appear in the positive training images but may not be discriminative. Then we use a novel latent SVM model to select and optimize discriminative part detectors with group sparsity regularization.

3.1 Initialization

We first initialize a set of part detectors which will be taken as candidates for further optimization and selection by our learning model. We have tried two types of initialization methods, compared experimentally in Section 4.2.3.

Random Sampling. To initialize the candidate part detectors for an image category, we crop a fixed number of image parts randomly from the positive training images. More specifically, we randomly select image and part locations within the image. Assume that we have K sampled image parts, then we initialize K part detectors $\{\Gamma_k\}_{k=1}^K$, $\Gamma_k = \{\beta_k, \tau_k\}$. Each part template β_k is taken as the feature vector of the k-th patch, and the part threshold τ_k is initially set to 0.

Patch Clustering. An alternative initialization approach is based on patch clustering. We first randomly crop a large number of image parts (approximately ten thousands) from the positive training images, then we perform K-means clustering ($K = 1000$ clusters in our implementation) over these sampled image parts. We only retain clusters of size 10 or more. Assume that we have K clusters of image parts, then we initialize K part detectors $\{\Gamma_k\}_{k=1}^K$. The part template β_k and part threshold τ_k are initialized with the k-th cluster center and a zero value respectively. This procedure is similar to the construction of a visual word dictionary in BoWs.

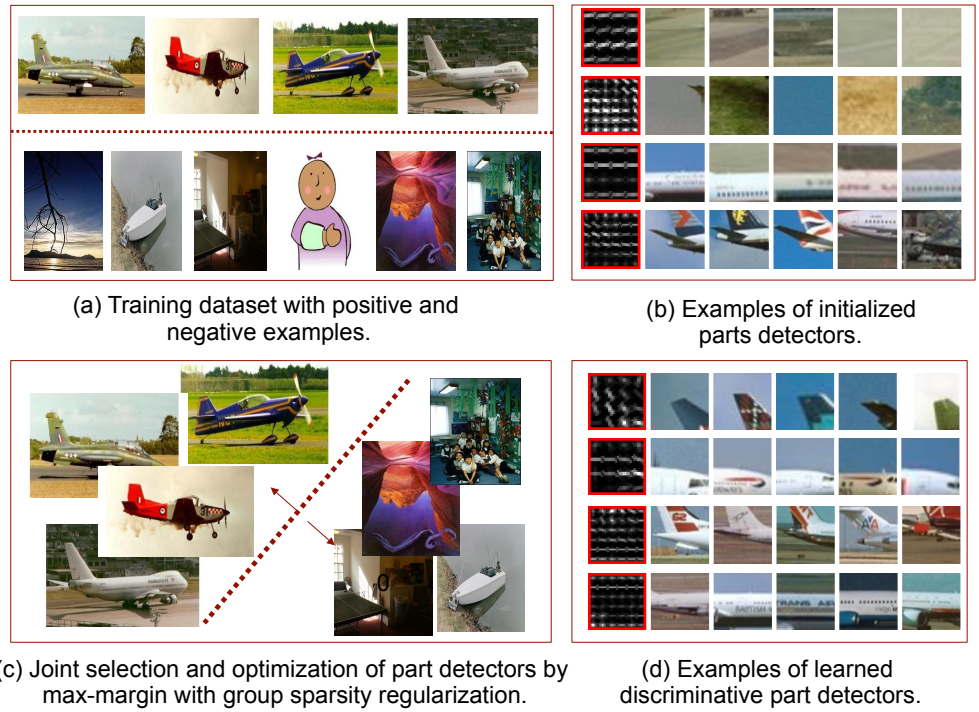


Fig. 3 An illustration of our learning framework. Given a training set of positive and negative images for an image category, we first initialize a set of part detectors as discussed in Section 3.1. Then we jointly select and optimize a set of part detectors (i.e., template / threshold pairs) using the novel latent SVM model regularized by group sparsity as discussed in Section 3.2.

The clustering-based initialization approach has also been shown to be effective for learning DPMs for object detection [64].

3.2 Learning Discriminative Part Detectors

With the above initialization, we now learn a set of part detectors that best discriminate the positive and negative training images. We require that the learned part detectors should appear more frequently and strongly in the positive training images than in the negative ones.

Before introducing our learning method, let us first define the confidence of image I belonging to the current category given class-specific part detectors $\Gamma = \{\Gamma_k\}_{k=1}^K$:

$$g(I,\Gamma) = \sum_{k=1}^{K} [\beta_k^T \Phi(I, z_k) - \tau_k]_+, \tag{3}$$

where z_k is a latent variable indicating the image part position with maximum response:

$$z_k = \text{argmax}_{z \in \Omega_I} \, \beta_k^T \Phi(I, z), \tag{4}$$

and Ω_I defines the set of all possible part positions in I. Observe from Eq.(3) that $g(I,\Gamma) \geq 0$ is defined as the sum of the maximum responses of all the part detectors to image I. Image I thus has higher confidence in belonging to the category of interest when more parts appear in I and have higher responses.

Next, we learn part detectors using a variant of the latent SVM model with group sparsity regularization. The basic idea is to jointly select and optimize the part detectors by maximizing the margin of the confidence value $g(I,\Gamma)$ on positive and negative training images. Denote the training image set as $\{I_n, y_n\}_{n=1}^N$ where $y_n = 1$ if I_n belongs to the category and $y_n = -1$ otherwise. The cost function is defined as:

$$E(\Gamma, b) = \frac{1}{N} \sum_{n=1}^{N} L(g(I_n, \Gamma), y_n, b) + \lambda R(B), \tag{5}$$

where $B = [\beta_1, \beta_2, \cdots, \beta_K]$ is the matrix whose columns are the vectorized part templates, L is the squared hinge loss function:

$$L\left(g(I,\Gamma), y, b\right) = [1 - y(g(I,\Gamma) + b)]_+^2, \tag{6}$$

and b is a bias term. We have chosen this loss function because it is differentiable w.r.t. g and b. We could have used other differentiable losses, e.g., a logistic function.

We use the regularization term $R(B)$ to impose group sparsity [83] over part templates, considering each template as a group. This forces the algorithm to automatically select

a few discriminative part detectors with non-zero templates from a large set of candidates. We use the $\ell_{2,1}$ structured sparsity norm [83] in this paper, i.e., $R(B) = \sum_{k=1}^{K} ||\beta_k||_2$, which is the sum of ℓ_2 norm of part templates, and is convex w.r.t. B. In summary, we learn the discriminative part detectors by solving:

$$\underset{\Gamma,b}{\operatorname{argmin}} \left\{ \frac{1}{N} \sum_{n=1}^{N} [1 - y_n(g(I_n, \Gamma) + b)]_+^2 + \lambda \sum_{k=1}^{K} ||\beta_k||_2 \right\}, \tag{7}$$

where $g(I_n, \Gamma)$ depends on latent variables in Eq.(4).

The above variant of the latent SVM model tries to enforce that $g(I, \Gamma) + b \geq 1$ if I is a positive training image, and $g(I, \Gamma) + b \leq -1$ if I is a negative one. This forces the learned part detectors to have larger responses to positive training images than to negative ones, and implies that the learned part detectors should be *discriminative*, i.e., more frequently and strongly trigger in the positive training images than in the negative ones. With group sparsity regularization, the optimization procedure will automatically discard the less discriminative part detectors among the initial ones.

Let us briefly compare our model to the latent SVM in [26]. Using the squared hinge loss instead of the regular one is a minor difference. More importantly, our proposed latent SVM model is regularized by group sparsity, which is able to automatically select discriminative part detectors from a large pool of initial detectors. Second, our learned part detectors are template / threshold pairs. With the thresholds, parts are not required to appear in every image of the category, which makes the detectors robust to intra-class variations caused by poses, sub-categories, etc.

3.3 Optimization Algorithm

The latent model of Eq.(7) is semi-convex [26] w.r.t. the part detectors Γ, i.e., it is convex for the negative examples and non-convex for the positive ones. This can be justified by the following facts. First, $g(I, \Gamma)$ is convex w.r.t. $\Gamma = \{\beta_k, \tau_k\}_{k=1}^{K}$. This can be easily shown by noting that $g(I, \Gamma) = \sum_{k=1}^{K} \max\{\tilde{\beta}_k^T \tilde{\Phi}(I, z_k), 0\}$, where $\tilde{\beta}_k = [\beta_k^T, \tau_k]^T$ and $\tilde{\Phi}(I, z_k) = [\Phi^T(I, z_k), -1]^T$, which is the maximum of linear functions. Second, the cost function in Eq.(7) is convex and non-decreasing w.r.t. $g(I, \Gamma)$ if I is a negative example (i.e., $y = -1$). Therefore the cost is convex w.r.t. Γ for the negative examples. However, it is non-convex for the positive examples.

Following [26], we optimize Eq. (7) by iteratively performing the following two steps: First, we update the latent variables for all the positive examples based on Eq. (4). Second, given the set of latent variables for all the positive examples (denoted as Z_p), we optimize part detectors $\{\beta_k, \tau_k\}_{k=1}^{K}$ and bias term b by minimizing the convex cost $E(\Gamma, b; Z_p)$ which is the cost function in Eq.(7) with fixed latent variables for positive examples. We stop the iterations when a maximal number of steps is reached or when the parameters do not change significantly anymore.

We now discuss how to minimize $E(\Gamma, b; Z_p)$ given Z_p. This cost function is smooth for b and piecewise-smooth for Γ. Therefore, we utilize a gradient descent method to optimize b and a subgradient method to optimize $\Gamma = \{\beta_k, \tau_k\}_{k=1}^{K}$ simultaneously. Due to the group sparsity regularization on $\{\beta_k\}$, we can not use gradient descent algorithm to optimize part templates because the $\ell_{2,1}$ regularizer is non-smooth. We therefore utilize a stochastic version of a proximal method (specifically, the FISTA algorithm [4]) for the optimization of part detectors by minimizing the convex cost $E(\Gamma, b; Z_p)$. Proximal methods are known to be effective in optimizing convex loss functions with sparse regularization. An optimization problem of the form $\min_B\{L(B)+\lambda R(B)\}$ where L is a convex loss function, and $R(B)$ is the above group sparsity regularization can be efficiently optimized by updating the parameters using a proximal operator [4]:

$${\beta_k}^{t+1} = \operatorname{Prox}_{\lambda\gamma}(\beta_k^t - \gamma \frac{\partial L(B)}{\partial {\beta_k}^t}), \tag{8}$$

where $\operatorname{Prox}_{\mu}(\boldsymbol{\beta_k}) = \frac{1}{||\boldsymbol{\beta_k}||_2} \boldsymbol{\beta_k} [||\boldsymbol{\beta_k}||_2 - \mu]_+$ for $\ell_{2,1}$ group sparsity regularizer.

In summary, given a training image set, we minimize the energy $E(\Gamma, b; Z_p)$ by iteratively updating the parameters:

$${\beta_k}^{t+1} = \operatorname{Prox}_{\lambda\gamma}(\beta_k^t - \gamma \frac{1}{N} \sum_{n=1}^{N} \frac{\partial L_n}{\partial {\beta_k}^t}), \tag{9}$$

$$b^{t+1} = b^t - \gamma \frac{1}{N} \sum_{n=1}^{N} \frac{\partial L_n}{\partial b^t}, \tag{10}$$

$${\tau_k}^{t+1} = \tau_k^t - \gamma \frac{1}{N} \sum_{n=1}^{N} \frac{\partial L_n}{\partial \tau_k}, \tag{11}$$

where γ is the step size determined by the back-tracking method in the FISTA algorithm [4], and $L_n = L(g(I_n, \Gamma), y_n, b)$. The gradient (w.r.t. b) and sub-gradients (w.r.t. β_k, τ_k) involved are computed as follows.

$$\frac{\partial L_n}{\partial b} = \begin{cases} -\eta_n y_n & \text{if } y_n(g(I_n, \Gamma) + b) < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

$$\frac{\partial L_n}{\partial \beta_k} = \begin{cases} -\eta_n y_n \Phi(I_n, z_{n,k}) & \text{if } \mathbf{C} \text{ is satisfied} \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

$$\frac{\partial L_n}{\partial \tau_k} = \begin{cases} \eta_n y_n & \text{if } \mathbf{C} \text{ is satisfied} \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

where $\eta_n = 2(1 - y_n(g(I_n, \Gamma) + b))$, $z_{n,k}$ is the k-th latent variable for image I_n, $\mathbf{C}$ denotes the conditions of

$$\beta_k^T \Phi(I_n, z_{n,k}) > \tau_k \text{ and } y_n(g(I_n, \Gamma) + b) < 1.$$

Algorithm 1 Algorithm for discriminative learning of class-specific part detectors.

Input: Training images $\mathbb{S} = \{I_n, y_n\}_{n=1}^N$. Maximum iterations T_{in} and T_{out}.
Output: Learned part detectors $\Theta = \{\beta_k, \tau_k\}_{k=1}^K$.

Initialize part detectors $\Gamma^0 = \{\beta_k^0, \tau_k^0\}_{k=1}^K$ as in Section 3.1, bias term $b = 0$ and $t_{out} = 0$;
while $t_{out} < T_{out}$ **do**
  Compute latent variables of all part detectors over positive training images by Eq.(4), then optimize part detections by the following FISTA iterations.
  Initialize $s_1 = s_0 = 1; \overline{\Theta}^1 = \overline{\Theta}^0 = \Theta^0 = \Gamma^{t_{out}}; t = 0$.
  while $t < T_{in}$ **do**
    Sample training examples $\mathbb{S}_t \subset \mathbb{S}$ (six positive and negative examples respectively).
    Compute latent variables for part detectors over sampled negative examples by Eq.(4).
    Compute the estimated average gradients of parameters (denoted as $\frac{\partial \hat{L}}{\partial \beta_k^t}, \frac{\partial \hat{L}}{\partial \tau_k}, \frac{\partial \hat{L}}{\partial b}$) using Eqs. (12-14) over S_t; Estimate Lipschitz constant γ^t by backtracking as in the FISTA algorithm [4];
    Update parameters: $\overline{\beta}_k^t = \mathrm{Prox}_{\lambda/\gamma_t}(\beta_k^t - \frac{1}{\gamma_t}\frac{\partial \hat{L}}{\partial \beta_k^t})$; $\overline{\tau}_k^t = \tau_k^t - \frac{1}{\gamma_t}\frac{\partial \hat{L}}{\partial \tau_k}$; $\overline{b}^t = b^t - \frac{1}{\gamma_t}\frac{\partial \hat{L}}{\partial b}$, for $k = 1, \cdots K$. Then assign $\overline{\Theta}^t \leftarrow \{\overline{\beta}_k^t, \overline{\tau}_k^t\}_{k=1}^K$;
    $s_{t+1} = \frac{1+\sqrt{1+4s_t^2}}{2}$;
    $\Theta^{t+1} = \overline{\Theta}^t + \frac{s_t - 1}{s_{t+1}}(\overline{\Theta}^t - \overline{\Theta}^{t-1})$;
    $t = t + 1$.
  end while
  $\Gamma^{t_{out}} = \Theta^{T_{in}}; t_{out} = t_{out} + 1$.
end while
Output the learned part detectors set Θ which is composed of the part detectors in $\Gamma^{t_{out}}$ with non-zero norms in the part templates.

The optimization of $E(\Gamma, b; Z_p)$ is a large-scale and high-dimensional convex optimization problem. To make it tractable, we propose to use a stochastic algorithm in which a subset of training images is sampled to approximate the gradients / subgradients [23].

Algorithm 1 presents the detailed optimization procedures. After optimization, we discard these part detectors having part templates with zero norms and derive a set of discriminative part detectors. Since they are compact and discriminative, these part detectors are enforced to diversely represent complex objects or images in different poses, subcategories, etc.

3.4 Implementation Details

To learn part detectors, we extract dense HOG features at eight-pixel intervals, and each image part is represented as the concatenation of all HOG features in the corresponding region. We set the number of iterations $T_{out} = 8$, $T_{in} = 80$ in Algorithm 1, and 6 positive and negative images are randomly selected in each step of the stochastic proximal algorithm. The training time for learning part detectors for one category is about 2-3 hours on a single CPU , and the training procedures for multiple categories is implemented in parallel on multiple CPUs.

Multi-Scale Discriminative Part Learning: In the conference version [70], we learned the part detectors based on the dense HOG features of training images of their original resolution when we optimize Eq.(7). We have now re-implemented Algorithm 1 using the dense HOG features of training images in a multi-scale pyramid. In this setting, each training image is represented by a pyramid in thirteen successive scales ($\{2^{-2}, 2^{-1.75}, \cdots, 2^{0.75}, 2\}$), and HOG features are extracted from the image pyramid of each scale. The part detectors are then learned with the training images in HOG pyramids using Algorithm 1, i.e., the latent part location variable is computed by finding the max-response location of a part detector to the HOG pyramids for each image. This change allows us to learn object parts robust to the scales. As shown in the following paragraphs, this multi-scale implementation consistently produces significantly improved results for image classification.

3.5 Total Response Maps of Part Detectors

To illustrate the learned part detectors, we define the *response map of a part detector* Γ_k to an image I as the weighted sum of all the detected parts appearing in the image pyramid by resizing the image to multi-scale resolutions, i.e.,

$$Q(\Gamma_k, I) = \sum_s \sum_{z \in \Omega_{I^s}} r_z(\Gamma_k, I^s) M_z(I^s), \tag{15}$$

where I^s is the image at scale s, $r_z(\Gamma_k, I^s)$ is the response value defined in Eq.(1), $M_z(I^s)$ is the binary mask of I^s indicating the region occupied by image part located at position z. The part mask $M_z(I^s)$ is rescaled by $\frac{1}{s}$, therefore the response map $Q(\Gamma_k, I)$ has the same resolution as I. In our implementation, we construct an image pyramid using thirteen scaling factors, i.e., $s \in \{2^{-2}, 2^{-1.75}, \cdots, 2^{0.75}, 2\}$. The *total response map* to an image is defined as the sum of

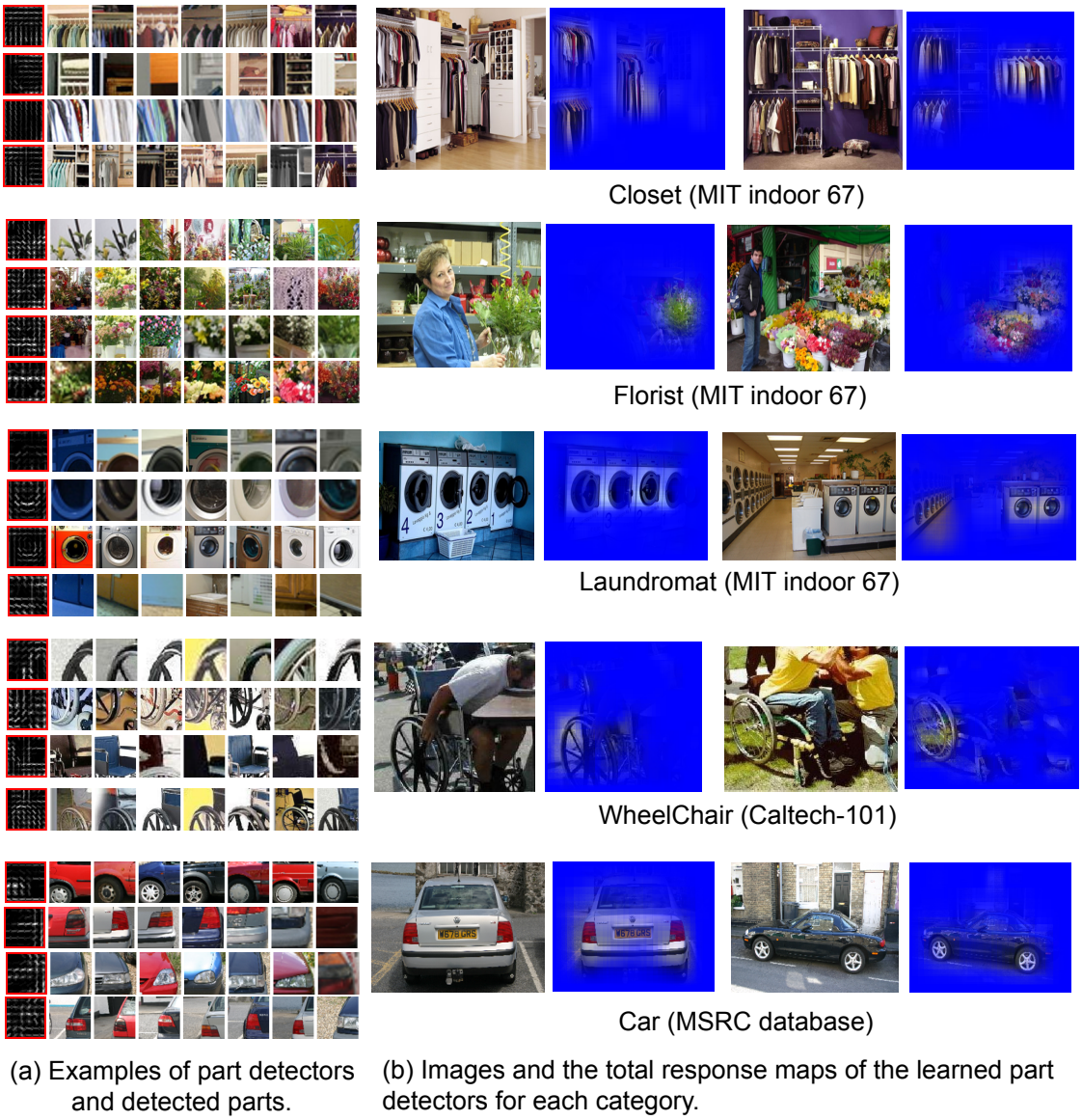


Fig. 4 Examples of learned part detectors, detected parts and total response maps of part detectors to images. The learned part detectors have higher responses to the discriminative regions in each category. Response maps are shown as the original images masked by the linearly normalized total response maps in the [0,1] range. (*Best viewed in color.*)

all the response maps of the derived part detectors:

$$Q(\Gamma, I) = \sum_k Q(\Gamma_k, I). \tag{16}$$

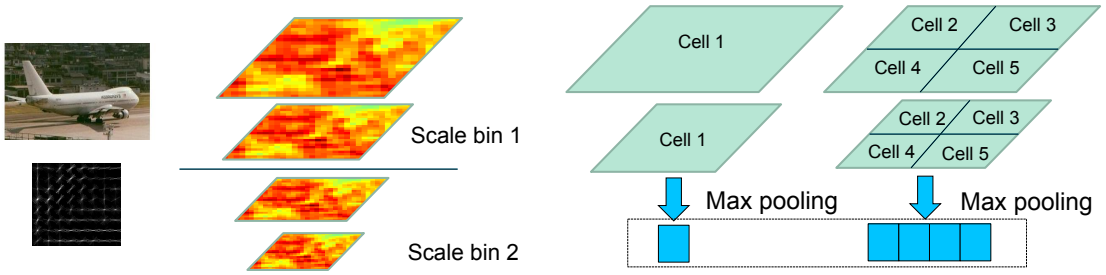
Figure 4 shows examples of learned part detectors and detected parts. As shown in Figure 4(a), the learned detectors are discriminative for the categories considered. For example, in categories of closet, florist, laundarmat, wheelchairs and cars, they commonly represent the important parts of these categories. Note that, even though we are not given any part localization information in training, our approach automatically learns the discriminative parts in these categories. Figure 4(b) shows total response maps of part detectors.

4 Application to Image Classification

Discriminative part detectors provide a mid-level and discriminative representation for an image category. We now apply them to image classification.

4.1 Image Coding

Given an image database, we learn class-specific part detectors for each category using one-vs-all training. We denote all the learned part detectors from different categories as $\Gamma = \{\Gamma_k\}_{k=1}^K$, K is the total number of part detectors. Based on our learning method for part detectors, an image I



(a) Multi-scale image responses to a part detector. (b) Feature max-pooling (per scale bin).

Fig. 5 An illustrative example of feature max-pooling for image coding. Given an image, we compute its multi-scale response maps to each part filter. We discretize the scales uniformly into scale bins as in (a). As shown in (b), for each scale bin, we perform max-pooling of the response values over spatial cells to produce a code. The final image code is the concatenation of these codes computed for all scale bins and part detectors.

can be naturally encoded by a vector of codes $\{c_k\}_{k=1}^K$, and each code $c_k = [\max_{z\in\Omega_I} \beta_k^T \Phi(I,z) - \tau_k]_+$, corresponding to max-pooling over the responses of part detector Γ_k to all the image parts in I.

Following object-bank [42], we improve the above coding method in a multi-scale scheme using the following steps. We resize the original image resolution in 13 scaling factors ($\{2^{-2}, 2^{-1.75}, \cdots, 2^{0.75}, 2\}$) to capture image parts in different scales. Then we uniformly quantize these scales into S bins. In each scale bin, we use spatial pyramid matching (SPM) [41] by dividing the image region into spatial cells in three levels ($1\times1, 2\times2, 4\times4$). The response values in each spatial cell are max-pooled to produce the image code for each part detector (please refer to Figure 5 for an illustrative example). Finally, the image I is coded by concatenating all the codes computed over all part detectors and scale bins. This coding method produces a feature vector with a length of SMK, where M is the number of cells in spatial pyramid. Given the image codes, we use a linear SVM classifier to produce the classification results.

4.2 Evaluation

In this section, we first present our experimental setting, then quantitative results for image classification.

4.2.1 Experimental setting

The discriminative part detectors are learned in one-vs-all mode for each dataset. When training the part detectors, we use multiple template sizes (8×8, 6×6, 4×4 feature cells) to capture features at different scales. 1000 part detectors are initialized for each category. The regularization parameter λ controls the sparsity of the solution. We have fixed it to 0.005 in all experiments, which retains about 10-15% of the part detectors after optimization. It takes around 4-5 seconds on one CPU to compute one image code when using 3000 learned part detectors, and the image coding procedures can be implemented in parallel.

We test our classification method on four representative image databases for object recognition (Caltech-101 [25], Caltech-256), scene categorization (15-Scenes [41], MIT-indoor [59]), and event categorization (UIUC-Sports [43]). We use mean accuracy (i.e., the average of per-class accuracies in a database) to measure classification performance. In all the experiments, "Ours_singleScale" denotes the results produced by our previous implementation [70], and "Ours_multiScale" denotes the results produced by our current multi-scale version.

Because our approach utilizes the conventional HOG feature, we mainly compare it with the approaches also using conventional features (e.g., HOG, SIFT, etc.). We also include results obtained by deep learning methods for completeness, although they benefit from features learned specifically for classification tasks over large annotated datasets.

Caltech-101. This database [25] contains 101 categories of objects and 40 to 800 images per category. We randomly split the database into train / test sets, with 30 images for training per class. Table 1 compares the results of our approach with other algorithms. Our learned discriminative part detectors achieve a competitive result of 81.6% mean accuracy on this database using a single type of HOG feature. Graph matching [22] performs comparable to ours (80.3% vs. ours 81.6%) on Caltech-101 using a kernel method designed for dense matching. However, it achieves significantly lower results on 15-Scenes as shown in Table 3, probably because objects in Caltech-101 are well aligned and can be densely matched with higher accuracy.

Caltech-256. This database [29] contains 30607 images belonging to 256 object categories, and each category has at least 80 images. It is much more challenging than Caltech-101 database due to the larger number of categories, large variations in object poses, scales or complex backgrounds. We randomly split the database into train / test set and each category has 30 and 60 images for training respectively. As

Table 1 Comparison on Caltech-101 database. DeCAF denotes the result by deep convolutional neural network in [21].

Single feature		*Multiple features*	
Methods	Accuracy	Methods	Accuracy
EMK [6]	70.1 ± 0.8	M-HMP [5]	82.5
Graph-matching[22]	80.3 ± 1.2	MKL [80]	**84.3**
LC_KSVD[32]	73.6	SubCategory [71]	81.9
LLC [76]	73.4		
Macro-feature [9]	75.7 ± 1.1		
Multi-way pooling[10]	77.1 ± 0.7		
P-FV [65]	80.1		
Sparse-coding [81]	73.2 ± 0.5		
SPM [41]	64.4 ± 0.8		
Ours_singleScale [70]	78.8 ± 0.5		
Ours_multiScale	$\mathbf{81.6 \pm 0.6}$		
DeCAF [21]	86.9		

Table 2 Comparison on Caltech-256 database.

Single feature			*Multiple features*		
Methods	ntrain = 30	ntrain = 60	Methods	ntrain = 30	ntrain = 60
EMK [6]	30.5 ± 0.4	37.6 ± 0.6	M-HMP [5]	**50.7**	**58.0**
Graph-matching [22]	38.1 ± 0.6				
IFK [58]	40.8 ± 0.1	47.9 ± 0.4			
LC_KSVD [32]	34.3				
LLC [76]	41.2	47.7			
Multi-way pooling[10]	41.7 ± 0.8				
P-FV[65]	44.9	52.6			
Sparse-coding [81]	34.0 ± 0.4	40.1 ± 0.9			
Ours_multiScale	$\mathbf{48.0 \pm 0.5}$	$\mathbf{55.1 \pm 0.4}$			
DeepFeats [84]	70.6 ± 0.2	74.2 ± 0.3			

shown in Table 2, our learned discriminative part detectors achieve a competitive result of 48.0% and 55.1% mean accuracies on this database based on single type of HOG feature. Our results are significantly higher than improved Fisher kernel [58] using dense SIFT, a recently published approach P-FV [65] using Fisher vector coding of Pyramid-SIFT descriptors, and the label consistent KSVD (LC_KSVD) approach in [32]. The state-of-the-art approach using conventional features is multipath Hierarchical Matching Pursuit (M-HMP) [5] that combines a collections of hierarchical sparse features.

15-Scenes. This database [41] is composed of 15 categories of indoor and outdoor scenes with 4485 images. We use 10 splits of train / test data to measure the mean and standard deviation of accuracies across different categories. In each split, 100 random images are taken as training images for each category and all the other images are taken as test images. Table 3 shows comparison results on 15-Scenes by different algorithms. Our discriminative part detectors perform significantly better than the low-level visual words in [81,41] and high-level object detectors in [42]. The highest result (92.9%) is achieved by label-consistent KSVD (LC_KSVD) [32]. But, as shown in the above paragraphs, our approach achieves much higher results than LC_KSVD on Caltech-101 (81.6% vs. 73.6%) and Caltech-256 (48.0% vs. 34.32%). The approach in [86] achieves 92.8% accuracy by combining DSFL features and deep features. Using DSFL features only, this approach achieves a mean accuracy of 84.2% which is lower than ours (87.2%) using a single type of HOG feature.

MIT-indoor. This database contains 15620 images belonging to 67 categories of indoor scenes. It is challenging because of the large ambiguities between categories. We use the same split of train / test data as in [59], with around 80 images for training, and 20 images for testing for each category. Table 4 shows a comparison of our method with the state of the art. We learn a total of 6372 (9.5% of the number of initial detectors) part detectors for 67 classes, and achieve 58.1% in mean accuracy using a single type of HOG features. Compared to related mid-level feature learning algorithms, our part detectors perform significantly better than discriminative patches learned by discriminative clustering [67], bags of parts [35], and multiple instance dictionary learning [77]. Though we achieve lower mean accuracy than the mode seeking algorithm [19], our result is produced by 6372 part detectors, which is much less than the 13400 elements in [19]. Moreover, compared to the visual element discovery [19] and bag-of-parts models [35], we learn and select discriminative parts in a more principled way by optimizing a latent SVM model.

UIUC-Sports. This database [43] contains eight categories of sport events, e.g., rowing, badminton, polo, rock

Table 3 Comparison on 15-Scenes database.

Single feature		*Multiple features*	
Methods	Accuracy	Methods	Accuracy
DSS [66]	85.5 ± 0.6	BSPR [79]	88.9 ± 0.6
Graph-matching [22]	82.1 ± 1.1	DSFL [86]	84.2
Hybrid-Parts [85]	84.7	DSFL + DeCAF [86]	**92.8**
ISPR [44]	85.1 ± 0.01	Hybrid-Parts + Gist-color+SP [85]	86.30
LC_KSVD[32]	$\mathbf{92.9}$	ISPR + FV [44]	91.6 ± 0.05
LPR [62]	85.8	Object-bank [42]	80.9
MIDL [77]	86.35	SContext [69]	87.8 ± 0.5
Sparse-coding [81]	80.3 ± 0.9	SUN [78]	88.1
SPM [41]	81.4 ± 0.5		
Ours_singleScale [70]	86.0 ± 0.8		
Ours_multiScale	87.2 ± 0.5		
DeCAF [21]	88.0		

Table 4 Comparison on MIT-indoor 67 scenes categorization.

Single feature		*Multiple features*	
Methods	Accuracy	Methods	Accuracy
BoP [35]	43.6	DSFL [86]	52.2
DPM [55]	30.4	DeCAF+ DSFL [86]	**76.2**
DiscPatches [67]	38.1	DPM + GIST + SPM [55]	43.1
Hybrid-parts [85]	39.8	Hybrid-parts + GIST + SPM [85]	47.2
LPR-LIN [62]	44.8	M-HMP [5]	51.2
MIDL [77]	50.2	Object-bank [42]	37.6
Mode Seeking [19]	**64.0**		
Ours_singleScale [70]	51.4		
Ours_multiScale	58.1		
DeCAF [21]	58.5		
DeepFeats_MP [28]	68.9		
VGG_VD_FV [15]	81.0		

Table 5 Comparison on UIUC-Sports database.

Single feature		*Multiple features*	
Methods	Accuracy	Methods	Accuracy
Hybrid-parts [85]	84.5	DSFL [86]	86.5
LPR [62]	86.25	DSFL + DeCAF [86]	**96.8**
LSA[45]	82.3 ± 1.8	Hybrid-parts+GIST+SPM [85]	86.3
MIDL[77]	88.5 ± 2.3	Object-bank [42]	76.3
Sparse-coding [81]	82.7 ± 1.7		
Ours_singleScale [70]	86.4 ± 0.9		
Ours_multiScale	$\mathbf{86.8 \pm 1.0}$		
DeCAF [21]	93.9		

climbing, etc. Following [43], we randomly take 70 images per category for training and the remaining data for testing in 10 rounds. Table 5 shows comparison results on this database. Our algorithm achieves significantly higher results than the hybrid-parts [85], object bank [42], sparse coding [81], LPR [62] and LSA [45]. Though our result is lower than the multiple instance dictionary learning (MIDL) algorithm [77] on this database, our results are higher than MIDL on the other two databases of MIT-indoor and 15-Scenes as shown in Tables 3 and 4.

Summary of the comparisons: Our classification method using discriminative part detectors achieve accuracies consistently among the best approaches using a single type of conventional feature on standard benchmark databases. Using multi-scale part learning consistently improves our classification results compared to the single scale learning approach in the conference version [70]. Figure 6 shows examples of the total response maps of the learned category-specific part detectors for various images sets. It shows that the learned part detectors response well to the discriminative regions in each category, and the non-informative background clutters are removed.

In the past few years, deep features have achieved significantly improved results in image recognition. As a mid-level part learning approach, our method could leverage deep features by substituting them to HOG features. This promising direction of research is left for future work.

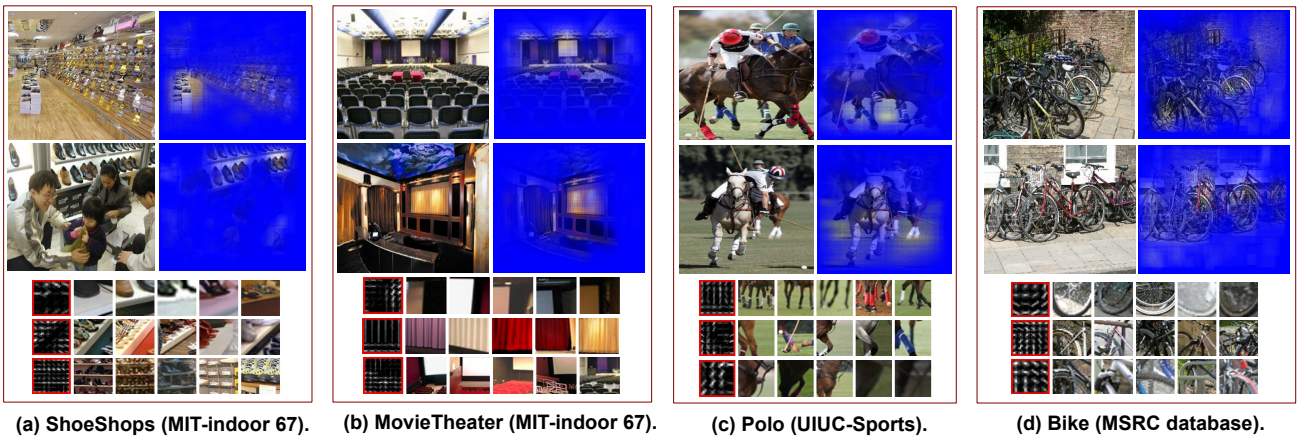


Fig. 6 More examples of the total response maps of images to the learned class-specific part detectors. (*Best viewed in color.*)

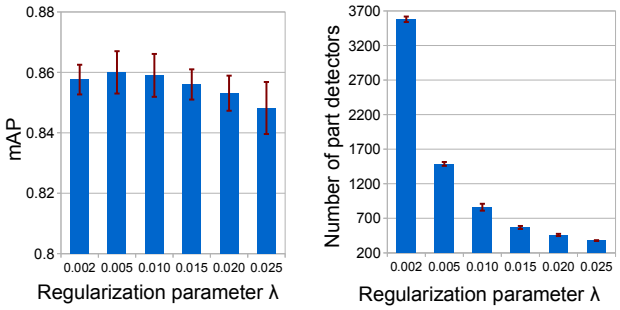


Fig. 7 Effect of the regularization parameter λ on classification performance for the 15-Scenes database.

4.2.2 Effect of the parameter λ on performance

The regularization parameter λ in Eq.(7) determines the degree of sparsity imposed on the part detectors. Theoretically, increasing λ imposes higher sparsity on the part detectors, i.e., the selection of fewer part detectors with non-zero part templates. Figure 7 shows the effect of different λ values on performance for the 15-Scenes database. With the increase of λ, we observe that the classification accuracy slightly increases then decreases. However, it is quite stable to the exact value of λ in the interval $[0.002, 0.015]$. On the other hand, with the increase of λ, the number of selected part detectors decreases fast as shown in the right part of Figure 7. We achieve a competitive result of 84.8% with only 378 part detectors (much fewer than the number of words in BoWs model [41]) with $\lambda = 0.025$.

4.2.3 Effect of part initialization on performance

In the above experiments, we initialize the part detectors using the patch clustering method of Section 3.1. Now we compare classification performance using random initialization instead. In both cases, we initialize 1000 initial part detectors for each category. Using random initialization (i.e., randomly cropping image parts from positive training images as the initial part templates), the final learned part detectors produced 85.8% mean accuracy on the 15-Scenes database, which is lower than 87.2% using patch clustering for initialization. This is reasonable, because the cluster centers of image parts in positive training images can represent these images in a more compact and complete way than randomly cropped positive patches.

We have also tested the effect of the number of initial part detectors on the final classification results. Using patch clustering for part initialization, we have learned and tested the part detectors for image classification with 300, 900, 1500, 2100, 2700 initialized part detectors for each category on 15-Scenes database. With the same split of train / test data, we obtain 86.7% 86.5%, 87.1%, 86.5% and 86.6% in mean accuracy respectively. This shows that classification performance is stable with respect to the number of initial part detectors.

5 Application to Object Cosegmentation

In this section, we will apply the learned part detectors to object cosegmentation. We first present our basic cosegmentation model, then improve it using image correspondences as constraints. Finally, we evaluate the proposed models.

5.1 Basic Model

We aim to segment the common objects in an image set with the same category label. Given an image set $\{I_n\}_{n=1}^N$ from the same category, we first learn discriminative part detectors $\Gamma = \{\Gamma_k\}_{k=1}^K$ from a training set with the input images as positive examples and a set of diverse background

images as negative examples. As shown in Figure 4(b) and Figure 8(b), the discriminative part detectors response more strongly and frequently to the common objects of the image set, which provides a high-level common object cue for cosegmentation.

For each image I in the image set, we aim to assign labels $X = \{\boldsymbol{x}_i\}$ to pixels with $\boldsymbol{x}_i = 1$ for a foreground pixel and $\boldsymbol{x}_i = 0$ for a background pixel. This can be considered as a weakly supervised clustering problem. In particular, discriminative clustering has achieved state-of-the-art performance on cosegmentation [34,33]. In this work, we design a novel cosegmentation algorithm by embedding the object cue provided by part detectors into the discriminative clustering framework.

We denote by v_i the image data associated with pixel i, and by $\Psi(v_i)$ the feature associated v_i into a high-dimensional Hilbert space $\mathcal{F}$. Discriminative clustering [33] tries to jointly infer the segment labels X and non-linear separating surface $f \in \mathcal{F}$ based on kernel SVM by minimizing:

$$E_c(X, f, d|I) = \sum_{i \in \Omega_I} [1 - \boldsymbol{x}_i(f^T \Psi(v_i) + d)]_+ + \alpha_c ||f||^2, \quad (17)$$

where d is a bias term, and α_c is a regularization parameter.

Discriminative clustering is weakly-supervised method for cosegmentation. In our approach, we incorporate the object cue provided by part detectors and label smoothness into the above formulation. The corresponding optimization problem is to minimize:

$$E(X, f, d|I) = E_c(X, f, d|I) + \sum_{i \in \Omega_I} E_o(\boldsymbol{x}_i|\Gamma, I) + \alpha_s \sum_{i \in \Omega_I} \sum_{j \in N(i)} E_s(\boldsymbol{x}_i, \boldsymbol{x}_j|I), \quad (18)$$

where $N(i)$ is the neighborhood of i. This cost function is defined for an image I in the given image set, and E_o is defined based on the common object cue shared by the image set:

$$E_o(\boldsymbol{x}_i|\Gamma, I) = \begin{cases} R_i(\Gamma_k, I) - \zeta \text{ if } \boldsymbol{x}_i = 0 \\ 0 \text{ if } \boldsymbol{x}_i = 1, \end{cases} \quad (19)$$

where R_i is the value of response map in Eq.(16) at pixel i. Obviously, this model prefers to assign a foreground label to a pixel with $\sum_k R_i(\Gamma_k, I) > \zeta$, and ζ is automatically set for each image by enforcing that pixels above this threshold occupy at most 40% of the image area. E_s is a smoothness term defined as $E_s(\boldsymbol{x}_i, \boldsymbol{x}_j|I) = |\boldsymbol{x}_i - \boldsymbol{x}_j| \exp(-\frac{||v_i^c - v_j^c||_2^2}{2\sigma})$ as in [60], where v_i^c is color vector at pixel i, and σ is the mean of the squared distances between adjacent colors over the image. E_s is submodular and encourages the segmentation boundary to align with strong edges.

We minimize $E(X, f, d|I)$ in Eq.(18) by alternatively inferring the SVM parameters $\{f, d\}$ and the segmentation label X. Given X, $\{f, d\}$ can be found by minimizing E_c since it is the only term that depends on f and d in Eq.(18). This can be done by a standard kernel SVM algorithm. Given $\{f, d\}$, the segmentation label X can be computed by minimizing Eq.(18) with fixed f, d, which can be done efficiently using graph cuts [11]. We initialize X by solving:

$$\text{argmin}_X \{ \sum_{i \in \Omega_I} [E_o(\boldsymbol{x}_i|\Gamma, I) + \alpha_s \sum_{j \in N(i)} E_s(\boldsymbol{x}_i, \boldsymbol{x}_j|I)] \}, \quad (20)$$

which is based on the object cue and label smoothness.

We take the feature vector v as the concatenation of HOG features v^h and color features v^c with length L_h and L_c respectively. Color values are scaled to [0, 1]. In kernel SVM, we use the kernel $K(v_i, v_j) = \exp(-\lambda_c(\frac{1}{L_h}||v_i^h - v_j^h||_2^2 + \frac{1}{L_c}||v_i^c - v_j^c||_2^2))$ with $\lambda_c = 5$. It is a valid kernel since it is the product of two radial basis kernels.

Figure 8 illustrates our cosegmentation procedure for an image set containing street signs. Assume that we already learned a set of discriminative part detectors for the given cosegmentation image set, we first compute the total response map of an image to these part detectors (Fig. 8(b)), then produce the initial segmentation result by optimizing Eq.(20) (Fig. 8(c)). Starting from this initial segmentation, we iteratively optimize the cosegmentation model to produce the final cosegmentation result (Fig. 8(d)).

5.2 Model Using Image Correspondences

We next present our improved model using image correspondences as constraints. Let $\mathbf{I} = \{I_1, I_2, \cdots, I_N\}$ denote an image set consisting of N images, we now jointly segment the image set by incorporating image correspondences into our basic model. The segmentation label set of $\mathbf{I}$ is $\mathbf{X} = \{X^p\}$, and $X^p = \{\boldsymbol{x}_i^p\}_{i \in \Omega_p}$ is the binary segmentation label for pixel i in image I_p ($p = 1, \cdots, N$), Ω_p is the pixel set of image I_p.

Following [61], we further incorporate segmentation consistency among the similar images into our model. For each image I_p to be segmented, we search its k-nearest neighboring images $\{I_q\}_{q \in N_m(p)}^k$ in the input image set using similarity of global features (e.g., Gist [52]), where $N_m(p)$ is the index set of neighboring images of I_p. Then we match I_p to each of its neighboring images I_q using dense feature matching [38]. We further introduce a correspondence term that constrains the densely matched pixels between I_p and I_q to have similar segmentation labels:

$$E_m(\boldsymbol{x}_i^p, \boldsymbol{x}_{m(i)}^q) = |\boldsymbol{x}_i^p - \boldsymbol{x}_{m(i)}^q| \exp(-\frac{||u_i^p - u_{m(i)}^q||_2}{\sigma_m}), \quad (21)$$

where $m(i)$ is the pixel in I_q that matches the pixel i in I_p. The exponential term measures the matching similarity of

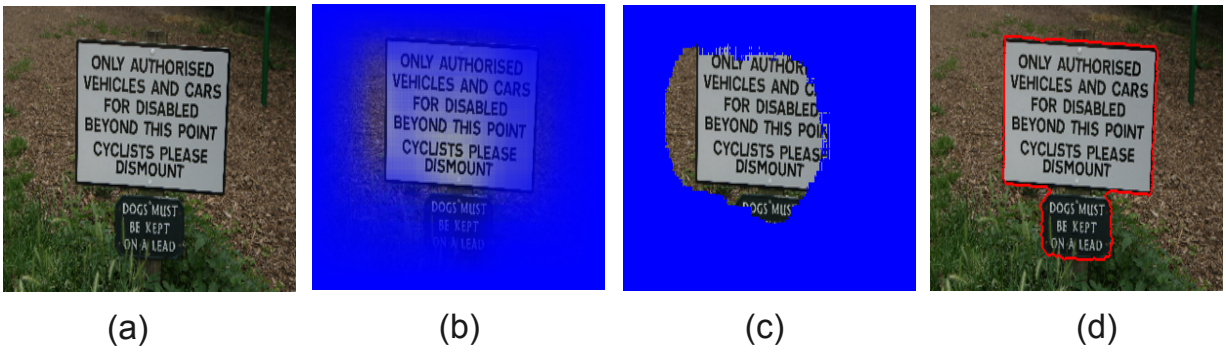


Fig. 8 Cosegmentation example (the image comes from "sign" category of MSRC database). (b) Total response map. (c) Initial segmentation mask. (d) Final segmentation boundary.

two pixels using their SIFT features denoted as u_i^p and $u_{m(i)}^q$ respectively. $\sigma_m = 0.02$ is a variance term.

By incorporating the correspondence term into our basic model, the objective function for the image set $\mathbf{I}$ can be written as

$$E(\mathbf{X},\mathbf{f},\mathbf{d}|\mathbf{I}) = \sum_{p=1}^{N}\left[E_c(X^p,f^p,d^p|I^p) + \sum_{i\in\Omega_{I_p}}\left(\alpha_o E_o(\boldsymbol{x}_i^p|\Gamma,I_p) + \alpha_s\sum_{j\in N(i)}E_s(\boldsymbol{x}_i^p,\boldsymbol{x}_j^p) + \frac{\alpha_m}{|N_m(p)|}\sum_{q\in N_m(p)}E_m(\boldsymbol{x}_i^p,\boldsymbol{x}_{m(i)}^q)\right)\right],$$

where $N_m(p)$ is the index set of the nearest neighboring images of I_p. The first and second terms are unary terms measuring per-pixel labeling cost using discriminative clustering and total response map of learned part detectors. The third and fourth terms are binary and enforce label smoothness of neighboring pixels within an image, and matched pixels across similar images.

We iteratively minimize $E(\mathbf{X},\mathbf{f},\mathbf{d}|\mathbf{I})$ as follows. First, the segmentation of each image in $\mathbf{I}$ is initialized by optimizing the model in Eq. (20) only using the total response maps of object parts. Then we optimize $E(\mathbf{X},\mathbf{f},\mathbf{d}|\mathbf{I})$ using coordinate descent [61]. At each iteration, instead of jointly optimizing over all images in $\mathbf{I}$, we optimize for each image I_p in $E(\mathbf{X},\mathbf{f},\mathbf{d}|\mathbf{I})$ by fixing the segmentations of all the neighboring images. This sub-problem for each image I_p can be optimized using the approach in Section 5.1, i.e., updating the separating surface $\{f^p,d^p\}$ using a kernel SVM, and then optimizing the segmentation mask for I_p using graph cuts. Though the energy function of $E(\mathbf{X},\mathbf{f},\mathbf{d}|\mathbf{I})$ is not convex, we found that our optimization algorithm converges fast, i.e., the segmentation masks do not change significantly in about 4-5 iterations.

5.3 Implementation Details

For the discriminative clustering term in Eq. (17), it is inefficient to use per-pixel features (HOG and color) for learning the kernel SVM, because the size of the kernel matrix ($N_p \times N_p$ if N_p is the number of pixels) is huge. In our implementation, we compute the HOG and color features with pixel intervals of 6×6, then use these features to learn a SVM separating surface. When updating the segmentation mask, the energy values of Eq. (17) are up-sampled to the full resolution using bilinear interpolation. We use the liblinear library [2] to optimize the kernel SVM problem, and the regularization parameter α_c in the SVM model is simply set to 1.

We next discuss the parameter settings in the model. We first analyze the range of values for each energy terms. The energy term E_c in Eq.(17) is the discriminative clustering term, and the hinge loss values is in the range of [0, 1]. The energy term E_o in Eq.(19) is also within the range of [0, 1] because we linearly scale the total response map to the range of [0, 1]. The energy term E_m in $E(\mathbf{X},\mathbf{f},\mathbf{d}|\mathbf{I})$ measures the label consistency between matched pixels, which is in [0, 1]. Because the labels of matched pixels are taken as constant in our coordinate descent optimization, this term is also a unary term during optimization. Since these energy terms are in the same order of magnitude, we make them equally contribute to the total energy as unary terms by simply setting $\alpha_m = 1$, $\alpha_o = 1$. The smoothness coefficient α_s in graph cuts is set to 5. All the evaluation results are produced using the above parameter settings.

5.4 Experimental Evaluation

The total response maps of category-specific part detectors provide common object cues for image set containing the same objects and diverse backgrounds. We next test our weakly supervised object cosegmentation algorithm on challenging datasets, i.e., the Internet object database [61] and a subset of the ImageNet database, collected from the Internet with diverse backgrounds. We also test our algorithm on the conventional MSRC database with restricted foreground / back-

[2] http://www.csie.ntu.edu.tw/ cjlin/liblinear/

Table 6 Comparison of object cosegmentation methods on Internet object dataset. The accuracies in each table cell are intersection-over-union and precision values. "w/o corr." and "w/ corr." indicate algorithms without and with image correspondence cue respectively.

Datasets	Images	w/o corr.				w/ corr.				Sub-categorization					
		ObjSegm [61]		Ours		ObjSegm [61]		Ours		eLDA [30]		NEIL [12]		ObjSubCate [13]	
Car	6381	46.1	72.3	**61.3**	**81.1**	63.4	83.4	**67.5**	**84.3**	**70.6**	85.6	63.1	85.5	64.7	**87.1**
Horse	4347	50.1	74.9	**52.1**	**86.4**	53.9	83.7	**57.4**	**88.8**	57.0	85.9	51.5	83.0	**57.6**	**89.0**
Airplane	4542	**51.2**	80.5	32.4	**86.8**	**55.6**	86.1	36.7	**89.0**	55.3	85.3	50.0	85.2	**60.0**	**90.2**

Table 7 Comparison of object cosegmentation methods on a subset of the Internet object dataset. The two values in each table cell are intersection-over-union and precision values.

Datasets	Images	DC [33]		mDC [34]		Diffusion [37]		ObjSegm [61]		ObjSubCate [13]		Ours	
Car	100	37.2	58.7	35.2	59.2	0.04	68.9	64.4	85.4	64.86	**87.65**	**73.4**	87.0
Horse	100	30.2	63.8	29.5	64.2	6.43	75.1	51.7	82.8	33.39	86.16	**54.7**	**87.6**
Airplane	100	15.4	49.3	11.7	47.5	7.9	80.2	**55.8**	88.0	**40.33**	**90.25**	36.3	88.6

ground object categories for cosegmentation [33,37,50]. To learn object part detectors, we take the input image set as the positive training set, and the remaining images in the dataset as negative training data. For the Internet object databases, we also include the VOC 2007 database (excluding the same category of objects) as additional negative training data. We utilize precision and intersection-over-union scores [34] to measure the segmentation accuracy, which are respectively defined as the ratio of correctly labeled pixels and the intersection over union of the ground-truth and estimated segmentation masks. With the learned part detectors, it takes around 20 seconds to segment each image on average on a single CPU.

Internet object dataset [61]. This is a dataset collected from Internet for objects of "Car", "House" and "Airplane". The datasets of "Car", "House" and "Airplane" contain 6381, 4347 and 4542 images respectively with considerably varying appearances, poses and backgrounds. Ground-truth object segmentation masks are provided for quantitatively measuring the accuracies. The data also includes many noisy images that do not contain the object of interest. Therefore it is a challenging dataset for testing the accuracy and scalability of our discriminative part-based algorithm.

In Table 6, we compare our approach to the state-of-the-art weakly supervised object segmentation algorithms in [61] (ObjSegm), and sub-categorization approaches using eLDA [30], NEIL [12] and a two-step sub-categorization method [13] (ObjSubCate). Method of "ObjSegm" is a similar approach to ours that combines the cues of object saliency [14], color distribution and image correspondences in a MRF framework. As shown in Table 6, we achieve significantly higher accuracies on the three datasets both with and without image correspondence cues [3]. When we do not use image correspondences, the major difference between our algorithm and "ObjSegm" is that we utilize the total response map of part detectors as an objectness cue instead of saliency. In this case, we achieve precisions of 81.1, 86.4 and 86.8, which are much higher than 72.3, 74.9 and 80.5 in [61] for the three categories. Our improved model in section 5.2 performs significantly better than the basic model. The sub-categorization based approaches categorize the image set into a few clusters and perform cosegmentation on these clusters. Our approach performs better than eLDA [30], NEIL [12], and is comparable to a well-designed sub-categoriziation approach in [13].

In Table 7, we also compare our algorithm to cosegmentation algorithms [33,34,37], object segmentation algorithm [61], and sub-categorization based algorithm [13] on subset of each image set used in [61], because the cosegmentation algorithms [33,34] are not well scaled to large scale dataset. Our results are among the state of the art on this dataset.

ImageNet [18]. We test our algorithm on a subset of ImageNet challenge 2012 with 13 categories of objects that was also used in [37]. We use all the images with ground-truth bounding-box annotations (around 500-1000 images) for each category. The ground-truth object segmentation masks are not available for ImageNet. In [37], the segmentation accuracies are directly measured using intersection-over-union w.r.t. the ground-truth bounding-boxes. We found that this is not an ideal accuracy measurement, because the ground-truth segmentation masks are within and occupy a fraction (denoted as r_{seg} in average over an image set) of the bounding-box areas. For example, $r_{seg} = 0.41$ for "horse" in Internet object dataset. This means that the average accuracy for the perfect segmentation masks are $r_{seg} < 1$, and the value of 1 does not imply the best segmentation result. We instead measure the accuracy using intersection-over-union computed between the bounding-box tightly enclosing the estimated foreground mask and the ground-truth bounding-box. This measurement guarantees that the ground-truth segmentation mask produces accuracy of 1. Because the cosegmentation algorithms [33,34] typically do not scale well to large image sets, we compare them on the subset of the

[3] In our approach, image correspondence cues can be disabled by setting $\alpha_m = 0$.

Table 8 Comparison of object cosegmentation methods on 13 objects in ImageNet. The value in each table cell is intersection-over-union value using bounding-box.

Datasets	Subset: 100 images in each category					Full: all images in each category		
	DC [33]	mDC [34]	Diffusion [37]	ObjSeg [61]	Ours	Diffusion [37]	ObjSeg [61]	Ours
Barn spider	56.3	57.8	42.5	57.2	**63.3**	42.8	57.3	**64.7**
Hognose snake	45.3	47.5	33.5	48.3	**54.0**	24.2	46.0	**54.0**
Coral	60.5	66.1	37.2	62.1	**67.0**	35.9	61.7	**66.4**
St Bernard	56.1	58.7	47.5	60.4	**64.0**	49.1	64.5	**65.5**
Basenji	53.3	56.4	45.9	60.2	**61.6**	41.9	59.2	**62.2**
Tabby cat	64.8	**68.0**	55.3	61.3	67.8	52.2	62.4	**67.5**
Jaguar	64.8	68.0	54.8	54.0	**70.1**	51.8	69.1	**72.1**
Lion	57.5	64.7	38.0	65.2	**65.3**	37.2	63.2	**65.5**
Starfish	40.9	46.2	36.7	47.5	**52.2**	34.4	52.3	**58.4**
Polecat	47.9	50.8	47.6	55.4	**56.5**	38.5	56.6	**59.0**
Badger	43.4	51.3	38.7	52.4	**57.5**	38.6	56.4	**59.6**
Orangutan	59.8	63.2	44.3	68.0	**70.0**	47.8	**66.6**	64.8
Guenon monkey	47.4	61.7	46.5	63.8	**66.4**	49.8	64.7	**65.3**
Average	53.7	58.5	43.7	58.1	**62.7**	41.9	60.0	**63.5**

Table 9 Comparison of object cosegmentation methods on MSRC dataset. The two values in each table cell are intersection-over-union and precision values.

Datasets	Images	DC [33]		mDC [34]		Diffusion[37]		ObjSegm [61]		Subspace [51]		Ours	
Bike	30	38.5	63.6	46.4	68.3	30.2	38.0	**54.1**	**77.9**	–	–	49.8	71.4
Bird	30	28.2	66.6	36.8	74.4	17.9	28.0	**67.3**	93.5	–	**94.8**	34.0	68.8
Car	30	58.0	77.0	62.1	79.4	36.8	53.4	**66.7**	**83.7**	–	80.1	60.2	77.8
Cat	24	33.7	62.8	45.1	74.6	21.2	39.5	**66.2**	**90.4**	–	–	52.1	78.4
Chair	30	46.2	75.5	39.9	68.2	31.7	69.0	**62.2**	**87.6**	–	–	49.8	75.6
Cow	30	52.7	77.9	61.3	83.2	35.9	79.1	**79.4**	**94.1**	–	87.8	54.1	77.0
Dog	26	47.2	75.8	47.2	75.7	22.3	37.6	**67.5**	90.0	–	**93.5**	44.4	74.9
Face	30	56.0	79.6	70.0	68.8	83.5	40.1	**58.3**	**82.1**	–	–	55.2	77.4
Flower	30	46.9	67.0	46.1	65.9	24.4	36.0	**71.4**	**85.5**	–	**86.5**	68.9	80.4
House	30	43.5	62.1	40.7	58.4	41.5	56.7	**72.8**	**87.2**	–	–	60.4	78.3
Plane	30	17.9	49.9	22.6	52.9	21.6	56.1	**56.7**	86.6	–	**87.1**	38.3	72.4
Sheep	30	68.1	88.0	55.3	74.9	63.3	86.1	**78.9**	**92.4**	–	89.0	60.1	83.6
Sign	30	56.3	78.5	52.3	74.9	38.3	59.6	**82.3**	**92.8**	–	–	73.0	89.0
Tree	30	40.1	66.6	69.0	81.3	50.8	67.8	**69.9**	**83.4**	–	–	66.9	79.6
Average		45.2	70.8	50.7	73.6	34.0	54.7	**68.1**	**87.7**	–	–	54.8	77.5

dataset as in [61], i.e., 100 randomly sampled images in each image category. We also compare the state-of-the-art object segmentation algorithms [37,61] for large scale image sets on the full set of images in each image category. Table 8 shows the results of different algorithms on the subset and full set of the database. In both cases, our approach achieves consistently better results than the other algorithms. For the algorithm in [37], we divide each image set into 100 clusters, and each cluster of images are cosegmented by running its source code using eight segments. The binary foreground and background assignments are produced by normalized cut among the eight sets of segments, as discussed in [37]. Because the source code of [61] is not available, we have re-implemented it using the same dense matching approach [38] for image correspondence as in our approach.

MSRC dataset [4]. It is a small cosegmentation dataset with restricted object categories in image foregrounds and backgrounds. The per-pixel segmentation masks are labeled for each image. Table 9 shows the comparison results. Our algorithm achieves higher accuracies on this database than the approaches in [33,34,37] using discriminative clustering or other segmentation approaches. The algorithm in [51] reports results on a subset of the database. It achieves higher accuracies but depends on human supervision in the form of labeled foreground regions on selected representative training images. This prevents its application to the larger datasets, e.g., the Internet object dataset, with thousands of images in each category. The algorithm in [61] also achieves higher accuracies than ours by simply using a regional contrast saliency [14] as the unary term in a MRF segmentation model separately processed for each image. This simple approach works well on this dataset because the images in MSRC commonly contain objects of interest with high color contrast to backgrounds, and color saliency is effective for detecting objects in a single image. But for the Internet image set, as shown in Tables 6-8, the saliency-based approach [61] achieves lower accuracies than our approach.

[4] `http://research.microsoft.com/en-us/projects/objectclassrecognition/`

Fig. 9 Examples of our cosegmentation results.

Figure 9 shows several cosegmentation results on different datasets. For these images with objects in diverse poses, our approach can effectively segment out the common objects from complex backgrounds.

Our weakly supervised cosegmentation approach produces competitive results on large scale image sets, i.e., Internet object dataset and ImageNet. We believe that this is mainly because we can learn diverse object part detectors that produce objectness maps better identifying objects of interest from complex backgrounds. Approaches using color saliency [61] or joint segmentation using color / shape features [34,37] perform well on the somewhat artificial dataset of MSRC, which contains objects of interest with strong contrast to the backgrounds. But for internet datasets, these approaches have difficulties in recognizing the foreground objects due to large variations of object appearances and backgrounds.

6 Relationship Between Our Work and CNN

As many visual recognition architecture including spatial pyramids and CNNs, our approach combines multiple stages of non-linear coding and pooling, followed by an SVM stage [9]. Not surprisingly, our part learning model can thus be interpreted as a convolutional layer over the HOG features, followed by an SVM loss. The template β_k of a part detector (β_k, τ_k) can be seen as a filter, and the threshold τ_k can be seen as a bias term. The confidence function for an image belonging to a category in Eqn. (3) can be decom-

posed into the following operations: convolutions of all the category-specific part templates over the HOG feature map, max-pooling over the whole image, substracting the thresholds, ReLU non-linear transform. The confidence values after these operations are fed into the SVM loss for categorization.

Our method differs from conventional CNNs as follows. First, our model is defined over HOG feature map for mid-sized patches, while CNNs learn successive small filters with non-linear operations such as max-pooling. Second, we impose a group-sparsity regularization over the part templates (i.e., the filters in an equivalent convolutional layer), where we only retain a small number of part detectors. In a conventional convolutional layer, the L_2-norm regularization is commonly used to prevent over-fitting. Third, our model is optimized using a stochastic version of proximal method, but a CNN is commonly optimized using back-propagation. Fourth, our model is defined in a weakly-supervised setting for learning object parts, where only the image-level category label is provided. A conventional CNN [21,39] commonly learns hierarchical image features for classification from a large-scale annotated dataset.

In summary, our model can be seen as a non-conventional convolutional layer with SVM loss defined on HOG features in a weakly-supervised setting. This idea bridges the conventional feature extraction approach and modern CNN approach. It may be useful for applications where we could extract multiple types of features (either by conventional feature extraction or CNNs computed for pixels / super-pixels / boxes), and combine them with additional convolutional layers for mid-level representation learning.

7 Conclusion

In this work, we have proposed a novel latent SVM with group sparsity to learn discriminative part detectors for image recognition and cosegmentation. Given image-level category labels, we have shown that our model is able to learn a small number of part detectors that best discriminate the image category from the background. Contrary to related algorithms, e.g., discriminative patches or bag-of-parts models, our approach is able to optimize and select the part detectors simultaneously in an efficient and principled way. We have experimentally demonstrated that our learned model achieves competitive results for image classification and cosegmentation.

In the future, we are interested in the following research directions. First, for multiple image categories, how to learn a shared dictionary of parts for more compact image representation? Second, how can we incorporate the spatial or geometric information among part detectors in a graph structure for object localization and recognition? Third, how can we combine our approach with deep features for learning mid-level discriminative parts? In our current work, we use dense HOG features for part templates, and a simple solution would be to substitute deep features to their HOG counterparts. Fourth, we will investigate using these learned mid-level part detectors for fine-grained recognition or attributes recognition.

Acknowledgement

Jian Sun was supported by NSFC (No. 61472313, 11131006), the 973 program (2013CB329404), NCET-12-0442, and NSFC (No. 61303121). Jean Ponce's work was supported in part by European Research Council (VideoWorld project) and the Institut Universitaire de France.

References

1. Ahmed, E., Shakhnarovich, G., Maji, S.: Knowing a good hog filter when you see it: Efficient selection of filters for detection. In: ECCV (2014)
2. Arbeláez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., Malik, J.: Semantic segmentation using regions and parts. In: CVPR (2012)
3. Azizpour, H., Laptev, I.: Object detection using strongly-supervised deformable part models. In: ECCV (2012)
4. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences **2**(1), 183–202 (2009)
5. Bo, L., Ren, X., Fox, D.: Multipath sparse coding using hierarchical matching pursuit. In: CVPR (2013)
6. Bo, L., Sminchisescu, C.: Efficient match kernel between sets of features for visual recognition. In: NIPS (2009)
7. Bourdev, L., Maji, S., Brox, T., Malik, J.: Detecting people using mutually consistent poselet activations. In: ECCV, pp. 168–181 (2010)
8. Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3d human pose annotations. In: ICCV (2009)
9. Boureau, Y., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: CVPR (2010)
10. Boureau, Y., Le Roux, N., Bach, F., Ponce, J., LeCun, Y.: Ask the locals: multi-way local pooling for image recognition. In: ICCV (2011)
11. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(9), 1124–1137 (2004)
12. Chen, X., Shrivastava, A., Gupta, A.: Neil:extracting visual knowledge from web data. In: ICCV (2013)
13. Chen, X., Shrivastava, A., Gupta, A.: Enriching visual knowledge bases via object discovery and segmentation. In: CVPR (2015)
14. Cheng, M.M., Zhang, G.X., Mitra, N.J., Huang, X., Hu, S.M.: Global contrast based salient region detection. In: CVPR (2011)
15. Cimpoi, M., Maji, S., Vedaldi, A.: Deep filter banks for texture recognition and segmentation. In: CVPR (2015)
16. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV workshop on statistical learning in computer vision (2004)
17. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
18. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)

19. Doersch, C., Gupta, A., Efros, A.A.: Mid-level visual element discovery as discriminative mode seeking. In: NIPS (2013)
20. Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.: What makes paris look like paris? ACM Transactions On Graphics **31**(4), 101:1–101:9 (2012)
21. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: ICML (2014)
22. Duchenne, O., Joulin, A., Ponce, J.: A graph-matching kernel for object categorization. In: ICCV (2011)
23. Duchi, J., Singer, Y.: Efficient learning using forward-backward splitting. In: NIPS (2009)
24. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. Image Processing, IEEE Transactions on **15**(12), 3736–3745 (2006)
25. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: CVPR workshop on generative-model based vision (2004)
26. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9), 1627–1645 (2010)
27. Girshick, R., Iandola, F., Darrell, T., Malik, J.: Deformable part models are convolutional neural networks. In: CVPR (2015)
28. Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. In: ECCV (2014)
29. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category data set (2007)
30. Hariharan, B., Malik, J., Ramanan, D.: Discriminative decorrelation for clustering and classification. In: ECCV
31. Jiang, Z., Lin, Z., Davis, L.S.: Learning a discriminative dictionary for sparse coding via label consistent k-svd. In: CVPR (2011)
32. Jiang, Z., Lin, Z., Davis, L.S.: Label consistent k-svd: Learning a discriminative dictionary for recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(11), 2651–2664 (2013)
33. Joulin, A., Bach, F., Ponce, J.: Discriminative clustering for image co-segmentation. In: CVPR (2010)
34. Joulin, A., Bach, F., Ponce, J.: Multi-class cosegmentation. In: CVPR (2012)
35. Juneja, M., Vedaldi, A., Jawahar, C., Zisserman, A.: Blocks that shout: Distinctive parts for scene classification. In: CVPR (2013)
36. Kim, G., Xing, E.P.: On multiple foreground cosegmentation. In: CVPR (2012)
37. Kim, G., Xing, E.P., Fei-Fei, L., Kanade, T.: Distributed cosegmentation via submodular optimization on anisotropic diffusion. In: ICCV (2011)
38. Kim, J., Liu, C., Sha, F., Grauman, K.: Deformable spatial pyramid matching for fast dense correspondences. In: CVPR (2013)
39. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
40. Kuettel, D., Guillaumin, M., Ferrari, V.: Segmentation Propagation in ImageNet. In: ECCV (2012)
41. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
42. Li, L., Su, H., Xing, E., Fei-Fei, L.: Object bank: A high-level image representation for scene classification and semantic feature sparsification. In: NIPS (2010)
43. Li, L.J., Fei-Fei, L.: What, where and who? classifying events by scene and object recognition. In: ICCV (2007)
44. Lin, D., Lu, C., Liao, R., Jia, J.: Learning important spatial pooling regions for scene classification. In: CVPR (2014)
45. Liu, L., Wang, L., Liu, X.: In defense of soft-assignment coding. In: ICCV (2011)
46. Lowe, D.G.: Object recognition from local scale-invariant features. In: CVPR (1999)
47. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: ICML (2009)
48. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Discriminative learned dictionaries for local image analysis. In: CVPR (2008)
49. M.Juneja, Vedaldi, A., Jawahar, C.V., Zisserman, A.: Blocks that shout: Distinctive parts for scene classification. In: CVPR (2013)
50. Mukherjee, L., Singh, V., Peng, J.: Scale invariant cosegmentation for image groups. In: CVPR (2011)
51. Mukherjee, L., Singh, V., Xu, J., Collins, M.D.: Analyzing the subspace structure of related images:concurrent segmentation of image sets. In: ECCV (2012)
52. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. International Journal of Computer Vision **42**(3), 145–175 (2010)
53. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision research **37**(23), 3311–3325 (1997)
54. Pandey, M., Lazebnik, S.: Scene recognition and weakly supervised object localization with deformable part-based models. In: ICCV (2011)
55. Pandey, M., Lazebnik, S.: Scene recognition and weakly supervised object localization with deformable part-based models. In: ICCV (2011)
56. Parizi, S.N., Oberlin, J.G., Felzenszwalb, P.F.: Reconfigurable models for scene recognition. In: CVPR (2012)
57. Parizi, S.N., Vedaldi, A., Zisserman, A., Felzenszwalb, P.: Automatic discovery and optimization of parts for image classification. In: ICLR (2015)
58. Perronnin, F., Sanchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: ECCV (2010)
59. Quattoni, A., A.Torralba: Recognizing indoor scenes. In: CVPR (2009)
60. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Transactions On Graphics **23**(3), 309–314 (2004)
61. Rubinstein, M., Joulin, A., Kopf, J., Liu, C.: Unsupervised joint object discovery and segmentation in internet images. In: CVPR (2013)
62. Sadeghi, F., Tappen, M.F.: Latent pyramidal regions for recognizing scenes. In: ECCV (2012)
63. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: Theory and practice. International journal of computer vision **105**(3), 222–245 (2013)
64. Santosh K. Divvala, A.A.E., Hebert, M.: How important are deformable parts in the deformable parts model? In: ECCV Workshop on Parts and Attributes (2012)
65. Seidenari, L., Serra, G., Bagdanov, A.D., Bimbo, A.D.: Local pyramidal descriptors for image recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(5), 1033–1040 (2014)
66. Sharma, G., Jurie, F., Schmid, C.: Discriminative spatial saliency for image classification. In: CVPR (2012)
67. Singh, S., Gupta, A., Efros, A.: Unsupervised discovery of mid-level discriminative patches. In: ECCV (2012)
68. Siva, P., Russell, C., Xiang, T.: In defence of negative mining for annotating weakly labelled data. In: ECCV (2012)
69. Su, Y., Jurie, F.: Visual word disambiguation by semantic contexts. In: ICCV (2011)
70. Sun, J., Ponce, J.: Learning discriminative part detectors for image classification and cosegmentation. In: ICCV (2013)
71. Todorovic, S., Ahuja, N.: Learning subcategory relevances for category recognition. In: CVPR (2008)

72. Vezhnevets, A., Buhmann, J.M.: Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In: CVPR (2012)
73. Vezhnevets, A., Ferrari, V., Buhmann, J.M.: Weakly supervised structured output learning for semantic segmentation. In: CVPR (2012)
74. Vicente, S., Rother, C., Kolmogorov, V.: Object cosegmentation. In: CVPR (2011)
75. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR (2010)
76. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality constrained linear coding for image classification. In: CVPR (2010)
77. Wang, X., Wang, B., Bai, X., Liu, W., Tu, Z.: Max-margin multiple-instance dictionary learning. In: ICML (2013)
78. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: CVPR (2010)
79. Yan, S., Xu, X., Xu, D., Lin, S., Li, X.: Beyond spatial pyramids: A new feature extraction framework with dense spatial sampling for image classification. In: ECCV (2012)
80. Yang, J., Li, Y., Tian, Y., Duan, L., Gao, W.: Group-sensitive multiple kernel learning for object categorization. In: CVPR (2009)
81. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)
82. Yao, B., Jiang, X., Khosla, A., Lin, A., Guibas, L., Fei-Fei, L.: Human action recognition by learning bases of action attributes and parts. In: ICCV (2011)
83. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B **68**(1), 49–67 (2005)
84. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV (2014)
85. Zheng, Y., Jiang, Y.G., Xue, X.: Learning hybrid part filters for scene recognition. In: ECCV (2012)
86. Zuo, Z., Wang, G., Shuai, B., Zhao, L., Yang, Q., Jiang, X.: Learning discriminative and shareable features for scene classification. In: ECCV (2014)