PartBook for Image Parsing

Kuiyuan Yang[1*] Lei Zhang[2] Yong Rui[2] Hong-Jiang Zhang[2]
[1]Dept. of Automation, University of Science and Technology of China,
[2]Microsoft Research Asia
[1]yky@ustc.edu, [2]{leizhang, yongrui, hjzhang}@microsoft.com

Abstract

Effective image parsing needs a representation that is both selective (to inter-class variations) and invariant (to intra-class variations). CodeBook from bag-of-visual-words representation addresses the invariance, and part-based models can potentially address the selectivity. However, existing part-based approaches either require expensive manual object-level labeling or make strong assumptions not applicable to real-world images. In this paper, we propose a PartBook approach that simultaneously overcomes the above two difficulties. Furthermore, we present an effective framework that integrates CodeBook and PartBook, which achieves both intra-class invariance and inter-class selectivity. Specifically, a set of candidate regions are first selected from heat map-like representations obtained by a SVM classifier trained for each category. Then the regions are clustered based on the dense matching-based similarity, and a part detector is learned from each cluster and further refined by utilizing a latent SVM. The learned PartBook summarizes the most representative mid-level patterns of each category, and can be readily used for image parsing tasks to identify not only objects but also different parts of an object. Extensive experimental results on real-world images show that the automatically learned parts are semantically meaningful, and demonstrate the effectiveness of ParkBook in image parsing tasks at different levels.

1. Introduction

Image representation plays a key role in all level of image parsing tasks. A good image representation should be both *selective* (large inter-class distance) and *invariant* (small intra-class distance). The CodeBook from bag-of-visual-words representation has been proven to be robust to *intra-class variations* [5], because it only uses small local features. On the other hand, as the object/part-based representations use bigger patches, capable of modeling spatial structures and carrying more semantic information, they are effective in handling *inter-class selectivity* in object detection tasks [8, 11, 22]. Intra-class invariance is well studied and has largely achieved good results [15, 20, 23]. In this paper, we will focus on providing an effective solution to the more challenging task of inter-class selectivity.

A part-based model typically consists of a set of part detectors learned from a set of aligned images, based on which the appearance likelihood and spatial consistency can be modeled and verified. Although there are several promising automatic methods for image-level annotation [18, 19], automatically aligning real-world images of a generic category is still an open problem. The challenges mainly come from two difficulties:
1. It is hard to select candidate regions for alignment as existing interest point detectors are only robust to affine transformations but not intra-class variations.
2. It is difficult to match regions due to the existence of large intra-class variations and other distracted regions from cluttered background.

The existing part-based approaches mainly fall into two categories.
1. *Manual labeling at object level*. Zhu et al. [24] proposed to specify a set of points on the target object boundary in training images with respect to a set of predefined parts (e.g., horse head, horse leg, etc.). Then, a hierarchical deformable template can be developed for robust object detection. Bourdev et al. [2] use detailed 3D human body annotations to learn body parts that are tightly clustered in both appearance and configuration space. Felzenszwalb et al. [8] instead obtained a set of initial part detectors by decomposing a global template learned from images with labeled object bounding boxes, and updated them by an iterative process of image alignment and detector learning. These approaches can potentially lead to part-based models, but the manual labeling is too costly to be scalable.
2. *Image level label but with restrictive image appearance*. To combat the above difficulties, the labeling information can be at image level. Ullman et al. [17] proposed to randomly select a set of candidate regions from some sample

*This work was performed when Kuiyuan Yang was visiting Microsoft Research Asia as research intern.

978-1-4673-1612-5/12/$31.00 ©2012 IEEE

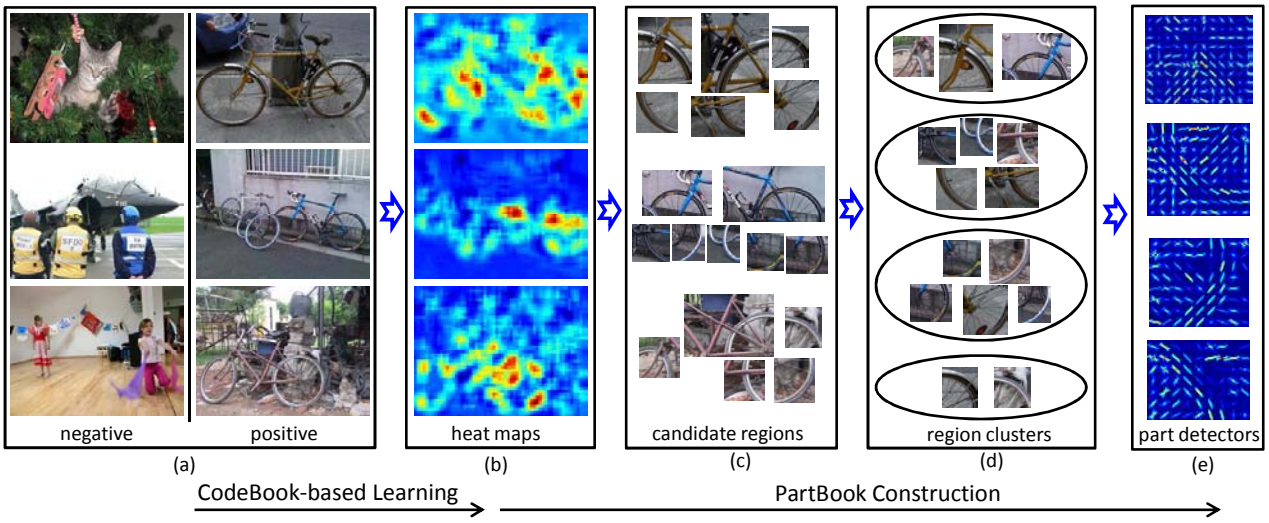

Figure 1. Schematic illustration of PartBook construction for *bicycle*. First, (a) we represent images by an improved version of bag-of-visual-words model and use SVM to learn a classifier. Then, (b) we utilize the learned SVM to relabel the positive images and generate a heap map for each positive image. After that, (c) we extract a set of heat regions from the heat maps and (d) group them based on a dense matching-based similarity. Finally, we learn a set of part detectors initialized by the region clusters.

images of the target object class, and use these candidates to search in every training image to choose the most informative ones as part candidates. Fergus et al. [9] instead used interest point detector to select a set of candidate regions from each training image and iteratively update the constellation model by testing their correspondence hypothesize. While these algorithms can automatically learn the parts, the high computational demands of their algorithms limits them to use very low image resolutions (e.g., 14x21 pixels) or fewer interesting points (20-30 local features per image). Instead of selecting parts from a large pool of candidates, multi-layer representations [13, 21] learn mid-level parts by summarizing all the image regions with the same size using low level patch features. However, these approaches work well only when foreground objects have few variations and the background is relatively simple.

In summary, to the best of our knowledge, no existing approaches that simultaneous satisfy the below requirements:
1. Only requires image-level label.
2. Handles real-world images with high resolution, diverse object variation and cluttered background.

In this paper, we present a novel approach that simultaneously meets the above requirement. This is done by introducing a key concept called *PartBook*–it is a set of representative parts in each category. With the PartBook, images of the same category can be implicitly aligned by applying the learned part detectors, and reliable part-based models can be built for object detection, image classification and other image parsing tasks. When combined with CodeBook, the proposed PartBook approach effectively handles both *intra-class invariance* and *inter-class selectivity*.

The inter-class selectivity of PartBook is obtained via gradually discriminative learning and common pattern abstracting, the process is illustrated in Fig. 1,
1. In the initial stage, we only know image-level labels and atomic patches in an image. For each category, we represent images by an improved version of bag-of-visual-words representation and use support vector machine (SVM) to learn a classifier [23]. The patch-level can produce a good cue for large structure selection [4, 16].
2. We next utilize the learned SVM coefficients to identify the most relevant regions to the category and use these regions as a good training set to learn initial part detectors. Because no part is predefined, we assume that similar regions belong to the same part and develop an unsupervised learning algorithm to group these regions. In this step, the region similarity is defined by employing a dense matching-based approach to take into account both the appearance similarity and the spatial consistency.
3. Finally, a set of part detectors is trained with the positive regions in each cluster, and further refined by utilizing a latent SVM.

The proposed PartBook approach leads to semantically meaningful representation. We conducted experiments on PASCAL VOC 2007 and 2010. The experimental results show that many of the learned mid-level parts look semantically meaningful and can provide deep image parsing beyond just image-level labels. For example, some wheel-related parts learned from *bicycle* are shown in Figure 1. We also tested a part detector of human head on the PASCAL VOC 2010 'person layout taster challenge', and achieved comparable results with the detectors learned from images

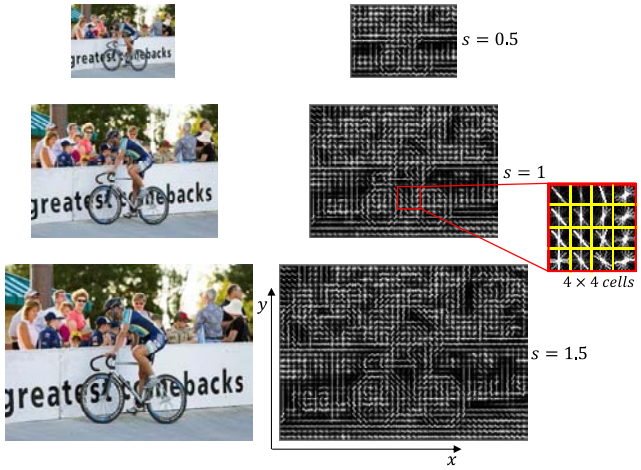

Figure 2. The HOG feature pyramid and a local feature of 4x4 cells.

with detailed labels.

The rest of this paper is organized as follows. We introduce image representation and PartBook construction in section 2 and present PartBook-based image parsing in section 3. Extensive experimental results are provided in section 4. Finally, we conclude this work in section 5.

2. Automatic PartBook Learning

Our models are built based on the Histogram of Oriented Gradient (HOG) features from [6]. Image representation is obtained by aligning an image to a CodeBook and a PartBook. A CodeBook consists of a set of visual words and captures small structures in an image. Large structures are captured by part detectors in the PartBook.

2.1. HOG Feature

We follow the construction in [6] with updates as suggested in [8] to define a dense representation of an image at a particular resolution. An image is first divided into non-overlapping regions of 8×8 pixels, namely *cells*. We represent each cell with a 31 dimensional HOG feature vector as described in [8]. To deal with objects with different scales, we define a HOG feature pyramid by computing HOG features at each level of a standard image pyramid (see Figure 2). Let H be a HOG pyramid and $l = (x, y, s)$ be a cell at (x, y) in the s-level of the pyramid. Let $\phi(H, l, w, h)$ denote a $w \times h \times 31$ dimensional feature vector obtained by concatenating the HOG features in the window of $w \times h$ cells with its top-left corner at l. Below we use $\phi(H, l)$ to simplify the notation $\phi(H, l, w, h)$ when the window size is clear from the context.

2.2. Data Set

Let $\mathcal{D} = \{(H_1, y_1), \ldots, (H_n, y_n)\}$ be a set of examples with image-level labels, where $y_i \in \{-1, 1\}$ and H_i specifies the feature pyramid for image i. Let $\mathcal{D}^+$,$\mathcal{D}^-$ be the positive and negative examples respectively. In following subsections we will introduce the procedure of learning the PartBook from images with only image-level labels.

2.3. First Layer

In the initial stage, the system has no knowledge to guide the selection of large structures. Then we start from representing each image with its atomic patches. Here, we employ an improved version of standard bag-of-visual-words model to aggregate the patch-level features to form the image-level features [23], which considers the appearance of each visual word in an image to avoid the quantization error. First, a set of patch-level features is densely sampled from each location of the HOG pyramid, the window size of each patch is set to be 4×4 cells (see Figure 2), and the patch-level feature at $l \in \mathcal{L}$ is $\phi(H, l) \in \mathbb{R}^{d_1}$, where $d_1 = 4 \times 4 \times 31$. As small variations of the local patch features, we use the k-means algorithm to partition the space $\mathbb{R}^{d_1}$ into A disjoint regions using 1 million randomly sampled local features, and denote the cluster centers by a generic CodeBook $\mathcal{C} = \{C_a; a = 1, \ldots, A\}, C_a \in \mathbb{R}^{d_1}$. Then each local feature $\phi(H, l)$ is assigned with the visual word by,

$$v_l = \arg \min_{a \in \{1,\ldots,A\}} \| \phi(H, l) - C_a \|_2 . \quad (1)$$

We denote the locations at which the visual words are C_a by:

$$\mathcal{L}_a = \{l : v_l = a\}. \quad (2)$$

The appearance of C_a in H is coded as

$$\phi(H; C_a) = \frac{1}{\sqrt{|\mathcal{L}_a|}} \sum_{l \in \mathcal{L}_a} \phi(H, l). \quad (3)$$

As suggested by [15, 23], $L1$-sqrt normalization is used for better performance. The image-level feature aggregated from the patch-level features is then represented as

$$\Phi_{\mathcal{C}}(H) = [\phi(H; C_1), \ldots, \phi(H; C_A)] . \quad (4)$$

We assume that each example H is scored by a linear classifier of the form,

$$f_{\beta_{\mathcal{C}}}(H) = \sum_{a=1}^{A} \beta_a \cdot \phi(H; C_a) = \beta_{\mathcal{C}} \cdot \Phi_{\mathcal{C}}(H), \quad (5)$$

where $\beta_{\mathcal{C}}$ is a vector of model parameters. The model parameters $\beta_{\mathcal{C}}$ are learned by passing the new constructed dataset $\{(\Phi_{\mathcal{C}}(H_1), y_1), \ldots, (\Phi_{\mathcal{C}}(H_n), y_n)\}$ into a classical SVM formulation,

$$\frac{1}{2} \| \beta \|^2 + C \sum_{i=1}^{n} \xi(y_i, \beta \cdot x_i), \quad (6)$$

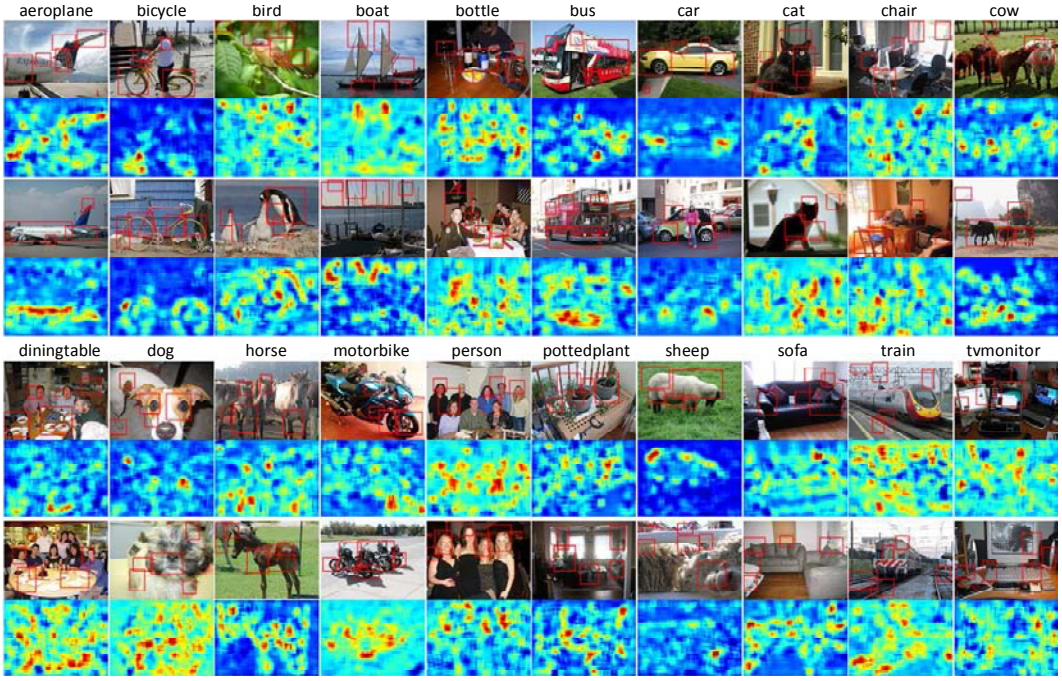

Figure 3. Examples of heat maps from the 20 categories in Pascal VOC 2010. For clarity, only the heat map of the HOG feature map at the level of the original image scale is shown. In each image, the top five high score regions are marked with red bounding boxes. (Better viewed in color.)

where $\xi(y_i, \beta \cdot x_i) = \max(0, 1 - y_i \beta \cdot x_i)$ is the hinge loss function, and $C > 0$ is the regularization constant.

The learned classifier provides the mapping between the aggregated patch-level features and the category. As the classifier is a linear operator on the image-level features, which are linearly aggregated from the patch-level features, the classification score of each image can be decomposed into the sum of patch-level scores, we follow the deduction in [23] to obtain the score of each patch as follows,

$$\begin{aligned} f_{\beta_C}(H) &= \sum_{a=1}^{A} \beta_a \cdot \phi(H; C_a) = \sum_{a=1}^{A} \frac{1}{\sqrt{|\mathcal{L}_a|}} \sum_{l \in \mathcal{L}_a} \beta_a \cdot \phi(H, l) \\ &= \sum_{l \in \mathcal{L}} \sum_{a=1}^{A} \frac{1}{\sqrt{|\mathcal{L}_a|}} \delta[v_l = a] \beta_a \cdot \phi(H, l) \\ &= \sum_{l \in \mathcal{L}} s(l) \\ s(l) &= \sum_{a=1}^{A} \frac{1}{\sqrt{|\mathcal{L}_a|}} \delta[v_l = a] \beta_a \cdot \phi(H, l) \end{aligned} \tag{7}$$

where $s(l)$ is the score of the patch at l, $\delta[v_l = a]$ is an indicator function that takes on value 1 if $v_l = a$ and 0 otherwise. The score of a cell at l denoted by $c(l)$ is averaged from all the patches that contain it. With the score of each cell, we can create the heat map for each HOG feature map. In Fig. 3, we visualize some heat maps from the 20 categories in PASCAL VOC 2010. From the heat maps, we observe that many common patterns in positive images are reinforced whereas unrelated background patterns are suppressed. This property provides a good guidance to select large structures.

The score of a region Ω is naturally defined as $s(\Omega) = \sum_{l \in \Omega} c(l)$. And high score regions are more positive according to the category classifier. Here, we use a threshold to detect the high score regions,

$$\mathcal{O}(l) = \begin{cases} 1, & if\ c(l) > \alpha \\ 0, & otherwise \end{cases}, \tag{8}$$

Empirically, we set α to be the average score of all the cells in an image. Then, the 4-connected foreground regions in $\mathcal{O}(l)$ are extracted. We denote the regions extracted from image i as $\{(\phi(H_i, l_r), s_{ir}); r = 1, \ldots, R_i\}$, where l_r is location of the top-left corner of the bounding box of the rth region, $\phi(H_i, l_r) \in \mathbb{R}^{d_{ir}}, d_{ir} = w_{ir} \times h_{ir} \times 31$, s_{ir} is the score of the rth region and R_i is the number of regions extracted from image i. We rank the regions extracted from all the positive images according to their region scores and keep the top 1000 regions as candidates, and denote them by $\{\phi_k; k = 1, \ldots, 1000\}$, ϕ_k is the HOG feature map of the kth region. Note that ϕ_k might have different dimensions as the region size (w_{ir} and h_{ir}) may be not the same.

2.4. Second Layer

The candidate regions selected from the first layer serve as good training data to construct part detectors. In this subsection, we will detail the process of learning part detectors.

2.4.1 Region Similarity

To summarize the common patterns from the candidate regions, we need to define a similarity measure to group similar regions. However, these regions are much larger than local patches and their features represented as ϕ_k do not have the same dimension. We observe that these category-related regions are with fewer variations and less-cluttered background. This allows us to adopt the similarity measure defined by explicitly feature matching. Many algorithms have proved the effectiveness of similarity measure defined by explicitly feature matching between two images with few variations [1, 3, 7, 14]. In this work, we adopt PatchMatch [1] to establish dense feature matching between two regions for its efficiency. The algorithm is driven by the key insight that some good feature matches can be found via random matching, and the natural coherence in the imagery allows it to propagate such matches quickly to surrounding areas [1]. After running PatchMatch between two regions ϕ_i and ϕ_j, for the cell at $p = (x, y)$ in region ϕ_i, we have the location of its matched cell in region ϕ_j by $p + d_p$, where $d_p = (dx_p, dy_p)$. The similarity measure based on the PatchMatch result is defined as:

$$\text{sim}(\phi_i, \phi_j) = \sum_{p \in \Omega_i} \parallel \phi_i(p) - \phi_j(p + d_p) \parallel^2 + \lambda \sum_{(p,q) \in \mathcal{E}} \parallel d_p - d_q \parallel^2, \tag{9}$$

which contains an appearance similarity term and a spatial consistency term. We add the spatial consistency term based on the fact that the regions from the same part tend to be more spatially consistent than the ones from different parts. The appearance vector is normalized to $L2$ unit length, and the spatial vector is normalized to $[0, 1]$ according to the image size. The tradeoff is chosen to be $\lambda = 1$.

2.4.2 PartBook Construction

After the similarity matrix is computed, we use Affinity Propagation to group similar regions because it only requires the similarity matrix (need not to be symmetric) and the preferences as input and can identify a subset of representative examples [10]. The method exchanges messages between data points until a good set of exemplars and corresponding clusters gradually emerges. We assume that all regions are equally considered to be exemplars. Hence the preferences are set to a common value–the median of $\text{sim}(\phi_i, \phi_j)$. After the regions are clustered, the set of representative regions is named as PartBook and denoted by $\mathcal{P} = \{P_b; b = 1, \ldots, B\}$, where $P_b \in \mathbb{R}^{d_b}, d_b = w_b \times h_b \times 31$.

2.4.3 PartBook Refinement

Each part (or region cluster) in the PartBook can be used to implicitly align the unseen images by part detection. It is desired that the learned part detectors are of good generalization ability. However, each part in the PartBook is just a specific region instance identified from positive images and cannot generalize well. We enhance the generalization ability of each part by training a latent SVM by detecting more positive instances (many are missed in the bottom-up process of Layer 1) and separating enormous negative instances [8]. We assume each image H is scored by part b as follows,

$$f_{\beta_b}(H) = \max_{l \in \mathcal{L}} \beta_b \cdot \phi(H, l), \tag{10}$$

where $\beta_b \in \mathbb{R}^{d_b}$ is a vector of model parameters, and $\mathcal{L}$ is all possible locations of HOG cells to place the part. In analogy to the classical SVM in Eq. 6, we would like to train β_b from the image-level labeled dataset $\mathcal{D}$ by optimizing the following objective function,

$$\begin{aligned} &\min_{\beta_b} \frac{1}{2} \parallel \beta_b \parallel^2 + C \sum_{i=1}^{n} \xi_i \\ &s.t.\ \max_{l \in \mathcal{L}_i} \beta_b \cdot \phi(H_i, l) \leq -1 + \xi_i, \forall i \in \mathcal{D}^- \\ &\quad \max_{l \in \mathcal{L}_i} \beta_b \cdot \phi(H_i, l) \geq +1 - \xi_i, \forall i \in \mathcal{D}^+ \\ &\quad \xi_i \geq 0 \end{aligned} \tag{11}$$

The constraint for each negative image can be equivalently replaced by $|\mathcal{L}_i|$ linear constraints, i.e., $\forall l \in \mathcal{L}_i, \beta_b \cdot \phi(H_i, l) \leq -1 + \xi_i$. This involves too large a number of linear inequality constraints to be optimized over explicitly. This is a common problem and has been well solved in structural SVM learning. Here we use the well-tuned solver cutting plane method to solve the problem [12].

The constraint for each positive image requires $\exists l \in \mathcal{L}_i, \beta_b \cdot \phi(H_i, l) \geq 1 - \xi_i$. These constraints imply a set of sub optimization problems, each one is formed by specifying a location for each positive image and denoted by $\mathcal{L}_+ = \{l_i; l_i \in \mathcal{L}_i\}, i = 1, \ldots, n_+$, where l_i is the location specified for positive image i, $n_+ = |\mathcal{D}^+|$. Each sub optimization is a classical SVM. However, as the total number of sub optimization problem is $\prod_{i=1}^{n_+} |\mathcal{L}_i|$, it is impractical to solve all the sub optimization problem to optimize Eq. 11. Actually, most the sub optimization problems are meaningless because their part locations are not correctly specified. In practice, latent SVM only solves a few sub optimization problems by an iterative process:

Step 1: Holding β_b fixed, select the sub optimization problem, $l_i = \arg\max_{l \in \mathcal{L}_i} \beta_b \cdot \phi(H_i, l), \forall i \in \mathcal{D}^+$.

Step 2: Holding l_i fixed for each positive image, optimize β_b by solving the sub optimization problem.

Both steps always improve or maintain the value of the objective function in Eq. 11. The most crucial part of training latent SVM is the initialization of β_b, which guides sub optimization problem selection in Step 1. According to the infinite monkey theorem [1], for a randomly initialized β_b, infinite amount of time will be needed to almost surely choose

[1] http://en.wikipedia.org/wiki/Infinite_monkey_theorem

the right position for the part in each positive image. Fortunately, the regions in cluster b serve as good training data to initialize β_b. First, we warp each region in cluster b to its representative region P_b according to the dense feature matching result to form the initial positive examples. This is to ensure all the positive examples are well aligned and have the same dimensionality. Then, a set of regions with the same size with P_b are sampled from negative images as negative examples. After that, we train an initial model β_b for part b by passing the training data into the classical SVM in Eq. 6. With the well initialized part detector, we start up the latent SVM to further refine it. On a single CPU, the entire training process takes 3 to 4 hours per object category in the PASCAL datasets, including the initialization of the parts. Figure 4 shows the most discriminative parts (i.e., with the smallest classification error) learned from the 20 categories in PASCAL VOC 2010. From the figure, one can see that many semantically meaningful parts are automatically learned from training images with only image-level labels, for example, wheels learned from *bicycle*, *bus*, *car* and *motorbike*, noses learned from *cow* and *dog*, heads learned from *cat* and *person* etc.

3. PartBook-based Image Parsing

The learned PartBook summarizes the most representative mid-level patterns of a category, and can be directly used for image parsing tasks to identify not only objects but also the parts of an object. Also, the PartBook can be used to enhance the selectivity of the image representation. In section 2, a two layer image representation is constructed to achieve both invariance and selectivity. The first layer captures small structures by a CodeBook and has a good intra-class invariance property. The second layer is based on a PartBook, which captures mid-level structures and has a good inter-class selectivity property. The final classifier for an image is essentially a linear model,

$$f_\beta(H) = \beta \cdot \Phi(H), \tag{12}$$

where

$$\begin{aligned} \beta &= [\beta_\mathcal{C}, \beta_\mathcal{P}] \\ \Phi(H) &= [\Phi_\mathcal{C}(H), \Phi_\mathcal{P}(H)], \end{aligned} \tag{13}$$

$\Phi_\mathcal{C}(H)$ is the feature vector from the first layer defined in Eq. 4, and $\Phi_\mathcal{P}(H)$ is the feature vector from the second layer according to the PartBook $\mathcal{P}$,

$$\Phi_\mathcal{P}(H) = [\phi(H; P_1), \ldots, \phi(H; P_B)], \tag{14}$$

where $\phi(H; P_b)$ is the appearance of part b in H and is defined as,

$$\begin{aligned} \phi(H; P_b) &= \phi(H, l_b) \\ l_b &= \arg\max_{l \in \mathcal{L}} \beta_b \cdot \phi(H, l). \end{aligned} \tag{15}$$

Fig. 5 illustrates the part detection results on two images. It can be seen that the part detectors generate strong responses at their corresponding positions in the images.

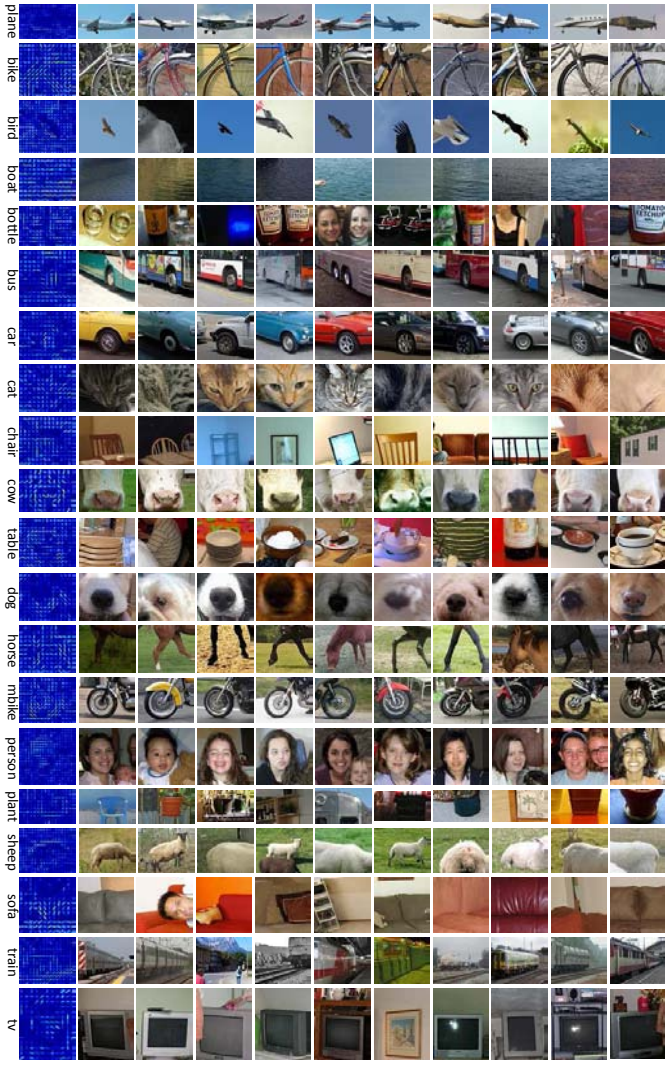

Figure 4. The most discriminative part detector learned from the 20 categories of PASCAL VOC 2010. Each row shows the part detector in the most left column, followed by its top 10 detections.

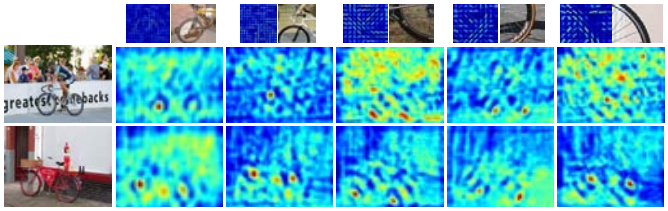

Figure 5. The first row shows the learned part detectors for *bicycle* and their most positive instances. The last two rows show the responses of part detector over the entire image. (Better viewed in color.)

4. Experimental Results

We evaluate the newly proposed PartBook in the context of detection and classification on two publicly avail-

Figure 6. The most left image is the head detector and head region is marked with a red bounding box. The detections are ranked by the detection scores, displayed in scan-line order (left to right, top to bottom).

Table 1. Performance of head detection on PASCAL VOC 2010 'person layout challenge' testset.

Method	OpenCV_Head	DPM_Head	PartBook_Head
AP(%)	22.5	42.3	40.0

able PASCAL VOC challenges datasets, i.e. 2007 [2] and 2010 [3]. The images in both datasets contain objects from 20 object categories in realistic scenes. The datasets are extremely challenging due to the wide varieties of appearances and poses of objects, and cluttered background. PASCAL VOC 2007 consists of 9,963 images which are divided into three subsets: training data (2501 images), validation data (2510 images), and testing data (4952 images). PASCAL VOC 2010 consists of 21,738 images and correspondingly are divided into three subsets: training data (4998 images), validation data (5823 images), and testing data (9637 images). The performance is evaluated using the Average Precision (AP) measure, the standard metric used by PASCAL VOC challenges, which computes the area under the Precision/Recall curve.

4.1. Detection

In this section we evaluate the PartBook in the context of detection.

Part Detection As no ground truth of the learned parts is available in the datasets, we indirectly evaluate the part detectors using the ground truth of object bounding boxes. To be considered as a correct part detection, the predicted part bounding box B_p must be no larger than the ground truth object bounding box B_{gt} and $\frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} > 0.5$. Since the ground truth object bounding boxes of VOC 2010 are still confidential, we only evaluated the part detectors on PASCAL VOC 2007 testing set.

We report performance of the top 3 part detectors of each category in Table 2. We also list the performance of Dalal-Triggs model for reference [6]. From the table, one can see that a single part detector for each category achieves comparable performance with the Dalal-Triggs model. The part detectors perform well on rigid objects such as *bicycle* and *car* as well as highly deformable objects such as *cat* and *dog*.

[2] http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
[3] http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html

Head Detection From the learned parts for *person*, we observed that some of them are about head. By coincidence, there are ground truth data for human head in PASCAL VOC 2010 'person layout challenge'. The test set of the challenge contains 320 images and 505 humans annotated. Bounding boxes are provided around every human figure and the task is to predict the location of head. The prediction of a head is considered to be correct if the overlap ratio with the ground truth is larger than 0.5. We use one learned part about head and mark out the exact head area as head detector.

Figure 6 illustrates the high score detections from the test set. From the figure, we can see that the learned head detector is relatively robust to the changes of pose and expression.

In Table 1, we compare our head detector (PartBook_Head) with the general frontal face detector (OpenCV_Head) provided by OpenCV and the head detector (DPM_Head) learned from images with human bounding boxes [8]. The PartBook_Head performs better than OpenCV_Head and is comparable with DPM_Head on the test set. Note that our head detector directly learned from image-level labeled data. This makes our approach more capable of being applied to other parts such as car wheel and cow nose.

4.2. Classification

To evaluate the usefulness of PartBook in the context of image classification tasks, we compare the following three methods:

Layer 1: an improved version of bag-of-visual-words representation as described in section 2.3, which is the state-of-the-art single feature method [23]. The codebook size A is 1024.

+DPM: the responses of 20 part-based detectors provided by [8] are added into the Layer 1 representation.

+PartBook: the responses of part detectors in the PartBook are added into the Layer 1 representation.

The results are summarized in Table 3. From the results, one can see that both part-based object detectors and PartBook improve the CodeBook-based image representation, which demonstrates the effectiveness of mid-level parts in image classification tasks. PartBook achieves a comparable improvement (only 0.6% lower in mAP on VOC 2007

Table 2. Performance of part detection on the PASCAL VOC 2007 test set.

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
Dalal-Triggs	17.9	36.5	1.8	0.8	15.3	29.1	28.0	1.4	12.2	16.6	16.0	9.8	21.7	24.1	17.2	11.3	13.9	11.8	17.1	29.9	16.6
Part1	10.3	28.9	6.8	9.2	3.0	19.4	26.1	20.4	1.2	9.5	14.0	16.2	13.6	21.6	18.9	9.5	9.6	11.4	13.5	15.0	13.9
Part2	10.2	27.3	3.9	3.4	1.3	19.3	26.0	18.0	1.0	9.3	13.0	12.9	11.7	20.3	18.4	9.3	9.6	7.1	8.8	13.7	12.2
Part3	10.1	26.8	3.7	3.1	0.3	11.3	25.1	16.4	0.6	5.1	12.0	10.7	11.6	19.8	18.4	6.3	5.3	5.3	6.1	9.5	10.4

Table 3. Performance of image classification on the PASCAL VOC 2007 and VOC 2010 test set.

Classification on VOC 2007

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
Layer 1	75.5	66.4	48.0	69.1	33.3	68.1	78.6	57.7	53.0	47.4	57.0	48.5	77.0	65.5	83.7	29.3	46.5	57.0	80.3	57.0	59.9
+DPM	76.2	76.1	45.9	69.9	49.9	70.2	84.2	64.4	56.6	47.5	58.4	47.3	77.9	69.3	88.4	46.2	51.0	61.2	76.4	66.9	64.2
+PartBook	73.1	75.4	45.6	69.6	47.2	70.0	81.5	64.2	56.7	48.3	60.2	47.9	79.2	69.2	87.0	44.7	51.1	59.8	76.2	64.3	63.6

Classification on VOC 2010

Layer 1	88.8	64.1	59.2	70.9	34.1	78.4	69.4	66.6	54.7	49.9	51.1	60.2	65.0	69.4	83.3	25.4	56.0	54.2	82.0	60.7	62.2
+DPM	86.8	73.4	57.8	69.6	47.2	79.7	73.9	69.3	57.7	50.2	52.0	60.5	65.6	74.2	88.2	38.2	58.1	49.1	79.0	68.3	64.9
+PartBook	84.9	71.0	55.8	70.3	46.5	79.0	71.8	69.2	57.2	51.6	51.1	60.6	64.9	73.0	86.4	37.6	58.1	50.9	79.5	64.6	64.2

and 0.7% lower on VOC 2010), but significantly reduces the manual labeling effort as PartBook directly works on image-level labeled data.

5. Conclusion

In this paper, we have presented a novel framework that automatically learns a set of mid-level parts for each category from images with only image-level labels. The algorithms in this framework gradually learn and abstract parts in a bottom-up manner, and further refine them in a top-down manner. Many of the learned mid-level parts look semantically meaningful, and can be readily used in image parsing tasks such as human head detection and wheel detection. Also, they can be used to enhance the selectivity of image representations. Experimental results on challenging benchmark data suggest the effectiveness of the learned parts in image parsing tasks at different levels.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 2009.

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

[3] B. Caputo and L. Jie. A performance evaluation of exact and approximate match kernels for object recognition. In *Electronic Letters on Computer Vision and Image Analysis*, 2009.

[4] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, 2007.

[5] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, 2004.

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

[7] O. Duchenne, A. Joulin, and J. Ponce. A graph-matching kernel for object categorization. In *ICCV*, 2011.

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[9] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[10] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 2007.

[11] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *CVPR*, 2009.

[12] T. Joachims. Making large-scale svm learning practical. *Advances in Kernel Methods Support Vector Learning*, 1999.

[13] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, 2009.

[14] C. Liu, J. Yuen, and A. Torralba. SIFT flow: dense correspondence across different scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[15] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.

[16] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *ICCV*, 2007.

[17] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 2002.

[18] M. Wang, X. Hua, R. Hong, J. Tang, G. Qi, and Y. Song. Unified video annotation via multigraph learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 2009.

[19] M. Wang, X.-S. Hua, J. Tang, and R. Hong. Beyond distance measurement: Constructing neighborhood similarity for video annotation. *IEEE Transactions on Multimedia*, 2009.

[20] K. Yang, L. Zhang, M. Wang, and H.-J. Zhang. Semantic point detector. In *International Conference on Multimedia*, 2011.

[21] K. Yu, Y. Lin, and J. Lafferty. Learning image representation from pixel level via hierarchical sparse coding. In *CVPR*, 2011.

[22] S. Zhang, Q. Tian, Q. Huang, and W. Gao. Objectbook construction for large-scale semantic-aware image retrieval. In *IEEE International Workshop on Multimedia Signal Processing*, 2011.

[23] X. Zhou, K. Yu, T. Zhang, and T. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV*, 2010.

[24] L. Zhu, Y. Chen, and A. Yuille. Learning a hierarchical deformable template for rapid deformable object parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.