Discriminatively Trained And-Or Tree Models for Object Detection

Xi Song[†], Tianfu Wu[‡,*], Yunde Jia[†] and Song-Chun Zhu[‡]
[†]Lab of Intelligent Information Technology, Beijing Institute of Technology
[‡]Department of Statistics, University of California, Los Angeles
{songxi, jiayunde}@bit.edu.cn, {tfwu, sczhu}@stat.ucla.edu

Abstract

This paper presents a method of learning reconfigurable And-Or Tree (AOT) models discriminatively from weakly annotated data for object detection. To explore the appearance and geometry space of latent structures effectively, we first quantize the image lattice using an overcomplete set of shape primitives, and then organize them into a directed acyclic And-Or Graph (AOG) by exploiting their compositional relations. We allow overlaps between child nodes when combining them into a parent node, which is equivalent to introducing an appearance Or-node implicitly for the overlapped portion. The learning of an AOT model consists of three components: (i) Unsupervised sub-category learning (i.e., branches of an object Or-node) with the latent structures in AOG being integrated out. (ii) Weakly-supervised part configuration learning (i.e., seeking the globally optimal parse trees in AOG for each sub-category). To search the globally optimal parse tree in AOG efficiently, we propose a dynamic programming (DP) algorithm. (iii) Joint appearance and structural parameters training under latent structural SVM framework. In experiments, our method is tested on PASCAL VOC 2007 and 2010 detection benchmarks of 20 object classes and outperforms comparable state-of-the-art methods.

1. Introduction

1.1. Motivations, objectives and overview

In recent literature of object detection, compositional hierarchy and deformable templates are widely used and have shown improved performance. Most state-of-the-art methods focus on weakly-supervised latent structure learning such as the deformable part-based model (DPM) [10] and the stochastic And-Or templates [18]. By weakly-supervised learning or learning from weakly annotated data, it means that only the bounding boxes for whole objects and no parts are available in training, e.g., in the PASCAL VOC object detection benchmark [7]. As is emphasized in [10],

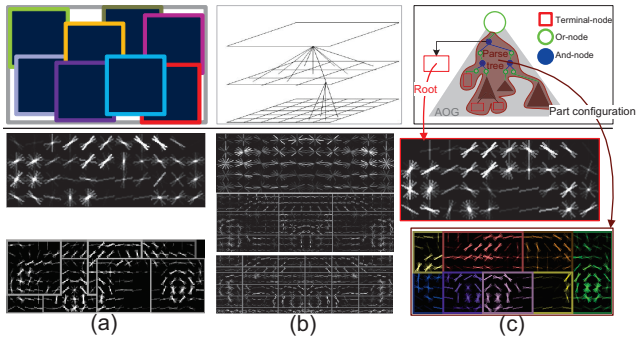

Figure 1. Illustration of three methods for weakly-supervised latent structure learning (*top*) and corresponding examples of learned car models from PASCAL VOC2007 dataset [7] (*bottom*). **(a)** shows the greedy pursuit method used by the deformable part-based model [10, 12], **(b)** the 3-layer quad-tree like decomposition method used in [27], and **(c)** the proposed method in this paper. The detection average precision (AP) are 54.2% for (a), 51.3% for (b) and 57.1% for (c) respectively. See text for details.

incorporating deformable parts is the major factor improving accuracy performance, however, how to find good part configurations (i.e., part shapes, sizes and locations which are latent variables given weakly annotated data) has not been addressed well in the literature. In existing work, there are two types of methods specifying latent part configurations: (i) *The greedy pursuit method* used by the DPM [10] where, as illustrated in Fig.1 (a), only a single part type (square with predefined size, e.g., 6×6) is adopted and the part configuration consists of a fixed number (often 8) of part instances placed by heuristic search [9, 12]. (ii) *The quad-tree like decomposition* adopted in [27] where the part configuration is predefined and the part types are fixed accordingly (see Fig.1 (b)).

Beside part configurations, another issue is how to learn sub-categories in an unsupervised manner to account for intra-class variations. Most existing work adopt k-mean clustering method based on aspect ratios of labeled object bounding boxes with k predefined (often $k = 3$) [10,12,27].

In this paper, we address the learning of sub-categories and part configurations from weakly annotated data in a

*Tianfu Wu is the corresponding author

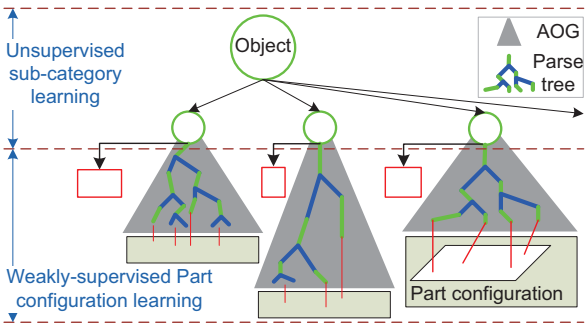

Figure 2. High-level illustration of the proposed And-Or Tree (AOT) model of an object category. See text for details.

principled way by learning reconfigurable And-Or Tree (AOT) models discriminatively.

Method overview. Fig.2 gives a high-level illustration of our AOT models. An object category (e.g., car) is represented by an Or-node (green circle) and it consists of an unknown number of alternative sub-categories (e.g., different viewpoints) which will be learned unsupervisedly. We unfold the space of all latent part configurations using a directed acyclic AOG (the solid gray triangle), and then seek the globally optimal parse tree in AOG using a DP algorithm. The globally optimal part configuration for a subcategory is obtained by collapsing the corresponding globally optimal parse trees onto image lattice (see the rightmost one). So, in the final model, each subcategory is represented by a root terminal node (red rectangle), and a collection of part terminal nodes (collapsed from the parse tree). Fig.1 (c) shows the learned part configuration for a sub-category of cars (side-view). In terms of the form of the final model, our AOT can be treated as a generalized representation for a mixture of DPMs [10].

Fig.4 shows an example of our directed acyclic AOG. We first quantize the image lattice using an overcomplete set of shape primitives (e.g., all rectangles with different sizes and aspect ratios enumerated in a given image lattice, see Fig.3 (a) and (b)), and then organize them into a directed acyclic AOG by exploiting their compositional relations (see Fig.3 (c)) [28]. We allow overlaps between child nodes when combining them into a parent node, which is equivalent to introducing an appearance Or-node implicitly for the overlapped portion (since there will be two different sets of appearance parameters in the final model) to account for the appearance variations. The constructed AOG can generate all possible part configurations. For example, without considering overlaps, the number of part configurations is listed in Table.1. The number will further increase geometrically if overlaps are considered.

Then, the learning of an AOT model consists of three components as follows.

(i) *Unsupervised sub-category learning*. We first compute the $n \times n$ similarity matrix of n training positive examples, and then utilize the affinity propagation clustering method [11] to learn the sub-categories automatically. We propose a method of measuring the similarity between any two positive examples by integrating out all the latent structures in AOG.

(ii) *Weakly-supervised part configuration learning*. For each sub-category, we create the AOG for the specified model grid. We first search the globally optimal parse tree by a DP algorithm, and then obtain the part configuration by collapsing the terminal nodes in the parse tree onto image lattice. The proposed DP algorithm consists of two phases: (1) The bottom-up phase factorizes the scoring function based on the depth-first search (DFS) of AOG. Appearance templates are discriminatively trained for terminal-nodes and their error rates on validation dataset are calculated. Then, each encountered Or-node selects the child node with the minimal error rate, and encountered And-nodes are treated as local deformable part-based models to calculate their error rates. (2) In the top-down phase, we retrieve the globally optimal parse tree using the error rates of nodes in AOG to guide the breadth-first search (BFS).

(iii) *Joint appearance and structural parameters learning*. Given the discovered sub-categories and their corresponding part configurations (i.e., an AOT), we train the parameters jointly using latent structural SVM [10, 24].

In experiments, we tested our method on PASCAL VOC2007 [7] and VOC2010 [8] detection benchmark of 20 object categories, and obtained better performance than state-of-the-art baseline methods [3, 9, 10, 12, 27] .

1.2. Related work and our contributions

In the literature of part-based object detection, there are four main directions on which researchers are working to improve performance.

(i) *Enriching the appearance features.* Many work extended the DPM [10] by incorporating other types of features complementary to HOG, such as local binary pattern (LBP) features [21,25], irregular-shaped image patches [15] and color attributes [14], which often increase the model complexity significantly. On the other hand, to make the model less complex, there are some representative work, including the steerable part models [17] which factorize the appearance parameters and share the linear subspace between parts, the sparselets [19] which build the sparse coding for the learned appearance parameters, and the hybrid image templates (HIT) [18] which integrate different complementary information (sketch, texture, flat and color) using the information projection principle, etc..

(ii) *Combining with contextual information or non-linear kernels.* Different types of contextual information are explored, such as the multi-category object layout context [6, 10, 15] and the image classification context [5, 21]. On the other hand, instead of only using linear SVM, nonlinear kernels are used to do object detection with cascade [22].

(iii) *Incorporating strong supervision.* Since introducing deformable parts is the major factor improving performance, there are some work extending DPM by providing strong supervision for parts, instead of treating them as latent variables. 2D semantic part annotations for animals are used in [2], keypoint annotations are used by the poselets [3], and 3D CAD car models are adopted by [16]. Another interesting work trains DPMs interactively [4].

(iv) *Exploring latent part configurations.* Beside the two methods listed above, the geometric And-Or quantization method was proposed for scene modeling recently in [23, 26], where non-overlapped shape primitives and generative learning are used. [20] adopted the idea in object detection.

In this paper, we follow the similar idea of And-Or quantization of image grid, but incorporate overlapping to account for the appearance "Or" implicitly, and adopt discriminative learning method.

Our contributions. This paper makes four main contributions to the weakly-supervised latent structure learning for object detection.

(i) It presents a directed acyclic AOG for exploring the space of latent structures effectively.

(ii) It presents an unsupervised method of learning subcategories of an object class.

(iii) It presents a DP algorithm to learn the reconfigurable part configuration efficiently.

(iv) It obtains better performance than comparable state-of-the-art methods on the detection benchmark of 20 object classes in PASCAL VOC2007 and VOC2010.

2. Unfolding the space of latent structures

In this section, we present how to construct the directed acyclic AOG to explore the space of latent structures.

Let Λ be the image grid with $W \times H$ cells, and assume rectangular shapes are used for parts. To decompose Λ, we need to specify (i) what the part types are (i.e. sizes and aspect ratios), (ii) where to place them, and (iii) how many instances each part type should have. Without posing some structural constrains, it is a combinatorial problem. As stated above, this is simplified by either adopting the greedy pursuit method with a single part type or using some predefined and fixed structure in existing work. We address this issue as follows.

Part types. A part type t is defined by its width and height (w_t, h_t). Starting from some minimal size (such as 2×2 cells), we enumerate all possible part types which fit the grid Λ, i.e., $2 \leq w_t \leq W$ and $2 \leq h_t \leq H$ (see A, B, C, D in Fig.3 (a) where A is of 2×2 cells).

Part instances. An instance of a part type t, denoted by t_i, is obtained by placing t at a position $(x_{t_i}, y_{t_i}) \in \Lambda$. So, it is defined by a bounding box in Λ, $(x_{t_i}, y_{t_i}, w_t, h_t)$. The set of all valid instances of a part type t is then defined by $\{(x_{t_i}, y_{t_i}, w_t, h_t) | (x_{t_i}, y_{t_i}) \in \Lambda, (x_{t_i} + w_t, y_{t_i} + h_t) \in \Lambda\}$.

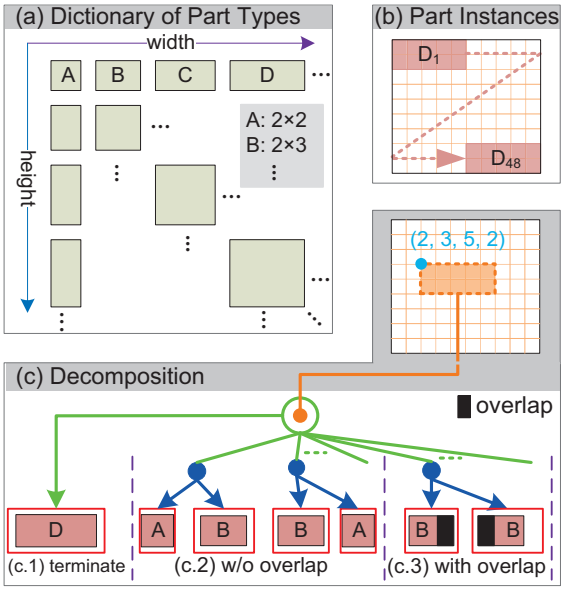

Figure 3. Illustration of (a) the dictionary of part types (i.e., all rectangles with different sizes and aspect ratios enumerated in a given image grid), and (b) part instances generated by placing a part type in image grid. Given the part instances, (c) shows how a sub-grid can be decomposed in different ways. We allow overlap between child nodes (see (3) in (c)). See text for details.

Fig.3 (b) shows the example of placing part type D (2×5 cells) in a 9×10 image grid.

The AOG organization. For any sub-grid $(x, y, w, h) \subseteq \Lambda$ (e.g., $(2, 3, 5, 2)$ in the right-top of Fig.3 (c)), we can either terminate it directly to the corresponding part instance (see Fig.3 (c.1)), or decompose it into two smaller sub-grids using either horizontal or vertical cut. Depending on the side length of (w, h), we may have multiple valid cuts along both directions (see Fig.3 (c.2)). When cutting either side we allow overlaps between the two sub-grids up to some ratio (see Fig.3 (c.3)). Then, we represent the sub-grid as an Or-node, which has a set of child nodes including a terminal-node (i.e. the part instance directly terminated from it), and a number of And-nodes (each of which represents a valid decomposition). This procedure is done recursively for the obtained two smaller sub-grids. By starting from the whole grid Λ and using BFS, we construct the AOG. Denote by $\mathcal{G} =< V, E >$ an AOG where the node set $V = V_T \cup V_{Or} \cup V_{And}$ consists of the terminal-node set V_T, the Or-node set V_{Or} and the And-node set V_{And}, and E is the edge set. We summarize the algorithm for creating the AOG in Alg.1 (which takes less than 1 second for typical grids, e.g. 20×18). See Fig.4 for an example.

The number of part configurations. Given an AOG, to count the number of all possible part configurations, we traverse through it using DFS: (i) For each encountered terminal node, the number of configuration is one, (ii) For each And-node, the number of configurations is the sum of

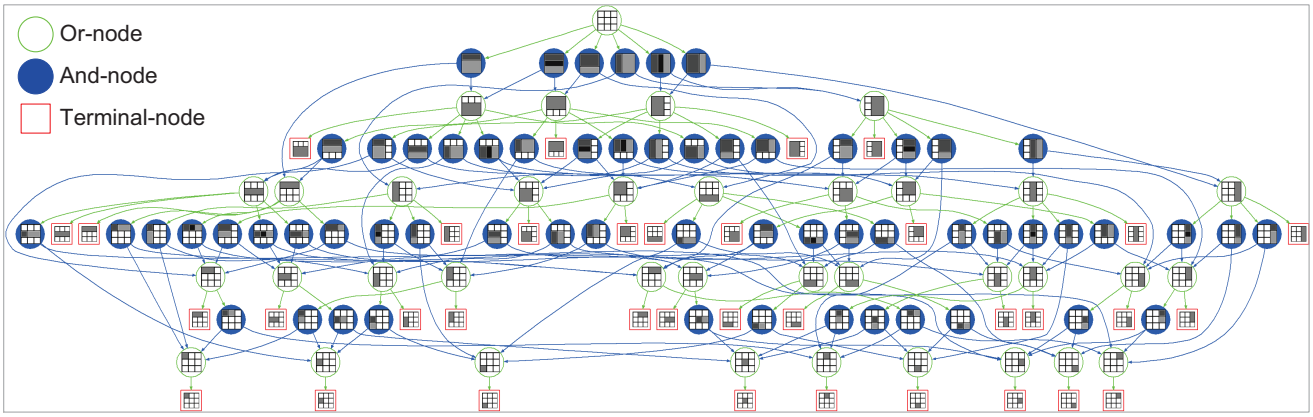

Figure 4. Illustration of the directed acyclic And-Or Graph (AOG) proposed to explore the space of latent structures of objects in this paper. For clarity, we show the AOG structure constructed for unfolding part configurations in a 3×3 grid. The AOG can generate all possible part configurations (the number is often huge for typical grid sizes, see Tabel.1), while allowing efficient exploration with a DP algorithm due to the property of being directed acyclic. See text for details. (Best viewed in color and magnification)

Input: Image grid Λ with $W \times H$ cells; Minimal size of a part type (w_0, h_0); Maximal overlap ratio r between two sub-grids.
Output: The And-Or graph $\mathcal{G} =< V, E >$ (see Fig.4)
Initialization: Create an Or-node O_Λ for the grid Λ, $V = \{O_\Lambda\}, E = \emptyset$, BFSqueue$= \{O_\Lambda\}$;
while *BFSqueue is not empty* **do**
  Pop a node v from the BFSqueue;
  if v *is an Or-node* **then**
    i) Add a terminal-node t (i.e. the part instance) $V = V \cup \{t\}$, $E = E \cup \{< v, t >\}$;
    ii) Create And-nodes A_i for all valid cuts; $E = E \cup \{< v, A_i >\}$;
    if $A_i \notin V$ **then**
      $V = V \cup \{A_i\}$;
      Push A_i to the back of BFSqueue;
    end
  else if v *is an And-node* **then**
    Create two Or-nodes O_i for the two sub-grids; $E = E \cup \{< v, O_i >\}$;
    if $O_i \notin V$ **then**
      $V = V \cup \{O_i\}$;
      Push O_i to the back of BFSqueue;
    end
  end
end

Algorithm 1: Building the And-Or Graph using BFS

that of its two child nodes, and (iii) For each Or-node, the number of configurations is the product of that of its child nodes with the double-counting between the child nodes subtracted. Table.1 lists some cases from which we can see that our AOG can cover a large number of part configurations using a relatively small set of part instances.

Grid	min. part	#Config.	#Term.	#And
3×3	1×1	319	35	48
5×5	1×1	76,879,359	224	600
10×12	2×2	3.8936e+009	1409	5209

Table 1. The number of part configurations generated from our AOG without considering the overlapped compositions.

3. Learning an AOT Model

In this section, we present the method of learning an AOT model from weakly annotated data.

The weakly annotated data. Denote by $D^+ = \{(I_1, B_1), \cdots, (I_n, B_n)\}$ the set of n positive training images for a given object category where B_i represents the labeled bounding box of an object instance in I_i (without loss of generality we assume each image I_i contains only one object instance). Denote by D^- a set of m negative images (i.e., images in which no object instances of the given object class appear). **Appearance feature.** In this paper, we use the modified HOG appearance feature [10] only for better understanding the performance improvement by exploring the latent structures using our AOG.

Three steps in learning. Since only the object bounding boxes B_i's are given in positive images, to learn the AOT model, we utilize three steps as follows:

(i) To account for the intra-category variations (e.g., viewpoints and poses which are not labeled), we initialize the latent sub-category label for each positive example by clustering. We first measure the similarity between any two positive examples by integrating out all the latent structures in AOG, and then do clustering based on the affinity propagation algorithm [11].

(ii) For each sub-category And-node, we select the model grid Λ and construct its AOG to unfold the latent structures, and then search the globally optimal part configuration in the AOG by proposing a DP algorithm.

(iii) Jointly training the whole AOT model on D^+ and D^- under the latent structural SVM framework [10, 24].

3.1. Learning sub-categories

In existing work, a predefined number of sub-categories is used, and the assignment of sub-category label for each positive example I_i is initialized based on the aspect ratio of bounding box B_i [10, 12, 27]. The limitations of using aspect ratios only are pointed out in several work [1, 2, 13] recently. We address this problem by our AOG.

Measuring similarity by integrating out all the latent structures. We first select a prototype size which is smaller than at least 90% bounding boxes in D^+, and then normalize all the positive examples to that size with their own aspect ratios unchanged. For simplicity of notation, we still use B_i's to denote the normalized bounding boxes.

For any two positive examples (I_i, B_i) and (I_j, B_j) $(1 \leq i < j \leq n)$, we obtain all the boxes of interest (BOI), denoted by Ω_{BOI}, which can overlap more than 70% with both B_i and B_j. This amounts to explore the unknown alignment. The similarity, $\text{Sim}(B_i, B_j)$, is defined by

$$\text{Sim}(B_i, B_j) = \max_{\Lambda \in \Omega_{\text{BOI}}} \text{Sim}(\Lambda | I_i, I_j), \tag{1}$$

where by definition $\text{Sim}(B_i, B_j) = 0$ if Ω_{BOI} is empty, and $\text{Sim}(\Lambda | I_i, I_j)$ is computed by integrating out the AOG constructed for decomposing Λ. We have,

(i) For each terminal-node $t \in V_T$, denote by $\Lambda_t = (x_t, y_t, w_t, h_t) \in \Lambda$ the sub-grid occupied by it. Let $F_i(\Lambda_t)$ and $F_j(\Lambda_t)$ the HOG feature vectors extracted for Λ_t in image I_i and I_j respectively. Then, we obtain

$$\text{Sim}(\Lambda_t | I_i, I_j) = \frac{|\Lambda_t|}{d(F_i, F_j)}, \tag{2}$$

where $d(\cdot, \cdot)$ is the Euclidean distance.

(ii) For each Or-node $O \in V_{Or}$, we compute $\text{Sim}(\Lambda_O | I_i, I_j) = \text{mean}(\text{Sim}(\Lambda_v | I_i, I_j))$, averaging over its child nodes $v \in ch(O) \subset (V_T \cup V_{And})$ (i.e. integrating out all the latent structures).

(iii) For each And-node $A \in V_{And}$, we compute $\text{Sim}(\Lambda_A | I_i, I_j) = \sum_{v \in ch(A)} \text{Sim}(\Lambda_v | I_i, I_j)$.

Then, we follow DFS order of nodes in the AOG to do the calculations above, and obtain $\text{Sim}(\Lambda | I_i, I_j)$ from the root Or-node. Note that aspect ratios of B_i's are also taken into account implicitly due to the construction of Ω_{BOI}.

Clustering by affinity propagation. Denote by $\mathcal{S}$ the symmetric similarity matrix for positive examples in D^+ where $\mathcal{S}(i, j) = \mathcal{S}(j, i) = \text{Sim}(B_i, B_j)$ (see an example in the top-left of Fig.5 computed for the horse class in PASCAL VOC2007). Given $\mathcal{S}$, we use the affinity propagation clustering method [11] to initialize the assignment of sub-category label for each positive example. We search the number of clusters over $k = 2 : 12$ in our experiments when using the affinity propagation, and choose the best number of clusters K which gives the largest improvement of the net similarity (which is returned from affinity propagation clustering) relative to its previous one (see the plot in the top-right of Fig.5).

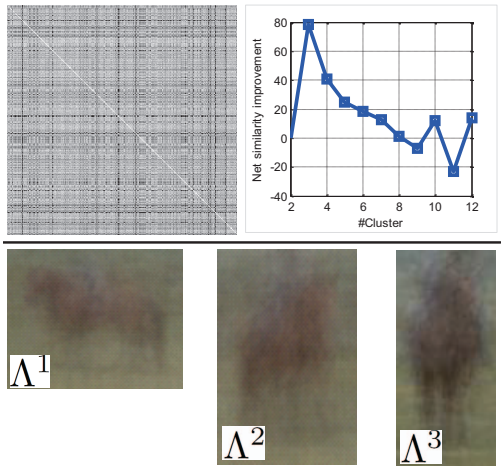

Figure 5. Illustration of learning sub-categories by clustering (horses in PASCAL VOC2007). *Top*: The similarity matrix and the plot of net similarity improvements using the affinity propagation clustering (based on which we choose 3 clusters). *Bottom*: We show the average images of the three sub-categories.

Now, for each positive example $(I_i, B_i) \in D^+$, we have assigned to it the initial sub-category label $\ell_i \in \{1, \cdots, K\}$, and we have the initial data set $D_k^+ = \{(I_i, B_i, \ell_i) | \ell_i = k\}$ for each sub-category And-node.

The initial AOT consists of an Or-node for the object class, and K root terminal-nodes. Based on the initial sub-category assignment, we use the most frequent aspect ratio in D_k^+ as the model aspect ratio, and then choose model size such that it can recover more than 80% examples in D_k^+ when the model is placed in the HOG feature pyramid, similar to [10, 12]. Denote by Λ^k the model grid for the sub-category And-node k. We use latent SVM [10] to train the initial appearance parameters for the K root terminal-nodes, $\theta^{app}_{\Lambda^k}$, and we obtain the updated D_k^+'s where the sub-category And-node assignment for each positive example is based on the trained initial AOT.

3.2. Learning the full AOT

3.2.1 Learning part configurations by DP

For each Λ^k, we double its size to find part configurations to account for finer appearance information, and create the AOG $\mathcal{G}^k$ for Λ^k using the method presented in Sec.2. To learn the part configuration for a sub-category And-node k, we need to specify (i) how to evaluate the goodness of configurations generated from AOG, and (ii) how to seek the globally optimal parse tree in AOG. In this paper, we evalu-

ate a part configuration in terms of its discriminative power (i.e., the error rate calculated using D_k^+ and D^-), and propose a DP algorithm to find the globally optimal parse tree.

The scoring function of an AOG. We first augment our AOG by introducing appearance and deformation parameters for terminal-nodes $t \in V_T$ and we have

$$\mathcal{G} =< V, E, \Theta^{app}, \Theta^{def} >, \tag{3}$$

where each terminal node t has its own appearance parameter $\theta_t^{app} \in \Theta^{app}$ and deformation parameter $\theta_t^{def} \in \Theta^{def}$.

Appearance features and parameters. We use HOG features. Given D_k^+ and D^-, we can train a linear SVM using the HOG feature for each terminal-node individually to obtain θ_t^{app}. In practice, for simplicity, we initialize them based on the learned $\theta_{\Lambda^k}^{app}$ above.

Deformation features. The anchor position of terminal node t is $(x_t, y_t) \in \Lambda$. Denote by $\delta = (dx, dy)$ the displacement in Λ. The deformation features are defined by a quadratic function $\Phi(\delta) = (dx^2, dx, dy^2, dy)$ which is also used in DPM and its variants [10, 27].

We use linear functions to evaluate both the appearance scores and the deformation scores. Given the appearance and deformation parameters and an input image I_Λ, we can compute the scores for nodes in $\mathcal{G}^k$ (as illustrated in Fig.6),

(i) For a terminal-node $t \in V_T$, we have

$$\text{Score}(t|I_\Lambda) = \max_{\delta}[\theta_t^{app} \cdot F(I_{\Lambda_t \oplus \delta}) - \theta_t^{def} \cdot \Phi(\delta)], \tag{4}$$

where $\Lambda_t \oplus \delta$ means the anchor box is displaced by δ.

(ii) The score of an Or-node $O \in V_{Or}$ is defined by

$$\text{Score}(O|I_\Lambda) = \max_{v \in ch(O)} \text{Score}(v|I_\Lambda). \tag{5}$$

(iii) The score of an And-node $A \in V_{And}$ is defined by

$$\text{Score}(A|I_\Lambda) = \sum_{v \in ch(A)} \text{Score}(v|I_\Lambda). \tag{6}$$

Deformation parameters. Above, we initialize $\theta_t^{def} = \vec{0}$ and set δ to the half size of model grid to obtain the local maximal appearance scores and corresponding deformed positions for terminal-nodes on D_k^+ and D^-. Then, we estimate θ_t^{def}'s using linear discriminant analysis (LDA), and recompute the scores of terminal-nodes.

The DP algorithm. Denote by $\mathcal{C}(\Lambda^k)$ a configuration generated from AOG $\mathcal{G}^k$ which consists of a set of selected terminal-nodes (i.e. part instances). We define the globally optimal part configuration in AOG by,

$$\mathcal{C}^*(\Lambda^k) = \arg\min_{\mathcal{C} \in \mathcal{G}^k} \text{Err}(\mathcal{C}(\Lambda^k)) \tag{7}$$

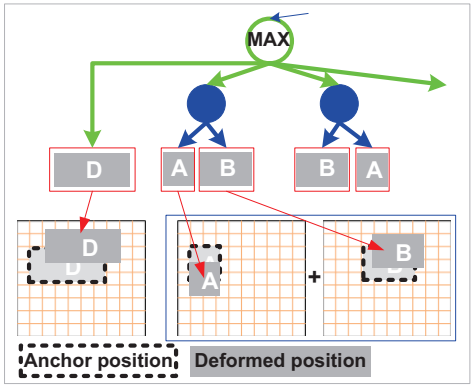

Figure 6. Illustration of computing scores for nodes in an AOG where A, B and D represent different part types.

where $\text{Err}(\cdot)$ is the error rate of a configuration computed using D_k^+ and D^-.

The DP algorithm has two phases: (i) With Eqn.4 and Eqn.6, we can calculate the error rates for each terminal-node and And-node. The Or-nodes take the minimal error rate of its child nodes. (ii) In the top-down pass, based on BFS, we start from the root Or-node to retrieve the globally optimal parse tree, and then obtain the part configuration.

3.2.2 Jointly training the parameters of full AOT

After the part configurations are initialized, we adopt the latent structural SVM [10, 24] to retrain the full AOT jointly similar to [10, 27]. More details of training by latent structural SVM are referred to [10, 24].

4. Experiments

We evaluate our method on the detection benchmark of 20 object categories in both PASCAL VOC2007 [7] and VOC2010 [8], and follow their experimental protocols in training and testing. Overall, we obtain better performance than the baseline methods.

Baseline methods. We compare with 4 baseline methods, the original DPM work [10], the latest two versions of the DPM (i.e. voc-release4 [9] and voc-release5 [12]), and the 3-layer DPM work [27]. The comparison are fair since all the methods use the same appearance features with the objective to learn better part configuration.

Parameter setting. In creating AOG, the two parameters are the minimal part size and the allowed overlapping ratio, which are set to 3×3 and 0.5 respectively in our experiments. In training the parameters by latent structural SVM, we use the same settings as in [10, 27] for fair comparison.

Detection results We summarize the detection results and performance comparison in Table.2 for PASCAL VOC2007 and Table.3 for PASCAL VOC2010.

In PASCAL VOC2007, we obtain the best performance for 17 out of 20 object categories, and for some object classes we made significant improvement relative to the

Table 2. Performance comparison using Average Precision (AP) for the 20 object categories in PASCAL VOC2007 dataset (using the protocol, competition "comp3" trained on VOC2007). All the 5 models use the HOG feature only, and the performance are obtained without post-processing such as bounding box prediction or layout context rescoring. We obtain better performance for 17 object classes.

	aero	bike	boat	bttle	bus	car	mbik	train	bird	cat	cow	dog	hrse	sheep	pers	plant	chair	tble	sofa	tv	avg.
DPM [10]	29	54.6	13.4	26.2	39.4	46.4	37.8	34	0.6	16.1	16.5	5	43.6	17.3	35	8.8	16.3	24.5	21.6	39	26.3
voc-r4 [9]	29.6	57.3	**17.1**	25.2	47.8	55	46.5	44.5	10.1	18.4	24.7	11.2	57.6	18.6	42.1	12.2	21.6	23.3	31.9	40.9	31.8
voc-r5 [12]	32.4	57.7	15.7	25.3	51.3	54.2	47.5	44.2	10.7	17.9	24	11.6	55.6	**22.6**	**43.5**	14.5	21	25.7	34.2	41.3	32.5
3-layer [27]	29.4	55.8	14.3	28.6	44	51.3	38.4	36.8	9.4	21.3	19.3	12.5	50.4	19.7	36.6	15.1	20	25.2	25.1	39.3	29.6
Ours	**35.3**	**60.2**	16.6	**29.5**	**53**	**57.1**	**49.9**	**48.5**	**11**	**23**	**27.7**	**13.1**	**58.9**	22.4	41.4	**16**	**22.9**	**28.6**	**37.2**	**42.4**	**34.7**

Table 3. Performance comparison using Average Precision (AP) for the 20 object categories in PASCAL VOC2010 dataset (using the protocol, competition "comp3" trained on VOC2010). All the performance are obtained without post-processing such as bounding box prediction or layout context rescoring for the first two methods. We also compare with the poselet [3] in the third row since the objective of poselet is aslo to find better parts (but their results are obtained by following a different protocol, competition "comp4", trained on own data, with additional keypoint annotations used).

	aero	bike	boat	bttle	bus	car	mbik	train	bird	cat	cow	dog	hrse	sheep	pers	plant	chair	tble	sofa	tv	avg
voc-r5 [12]	42.9	47.2	11.1	26.3	**48.4**	40.2	44	39	10.3	**22.9**	22.9	19.9	41.5	28.3	41	7.6	17	10.2	18.2	32.9	28.6
Ours	**44.6**	48.5	**12.9**	26.3	47.5	**41.6**	45.3	39	**10.8**	21.6	**23.6**	**22.9**	40.9	30.4	37.9	**9.6**	17.3	11.5	**25.3**	31.2	**29.4**
poselet [3]	33.2	**51.9**	8.5	8.2	34.8	39	**48.8**	22.2	-	20.6	-	18.5	**48.2**	**44.1**	**48.5**	9.1	**28**	**13**	22.5	**33**	

runner-up (2.5% for bike, 2.1% for car, 4% for train, and 2.9% for dining table).

In PASCAL VOC2010, we obtain better performance than voc-release5 [12] for 13 out of 20 classes. We also compare with the poselet work [3] since the objective of poselet is also to find better parts (note that the detection results of poselet are obtained by following different protocol, i.e. trained on their own data with additional keypoint annotations used). We obtain better performance for 10 out of 18 object categories on which poselet tested.

Fig.7 shows some examples of detection results for the 20 object categories in PASCAL VOC2007.

5. Conclusion

This paper presents a method of learning AOT models discriminatively from weakly annotated data for object detection, with two main issues being addressed: (i) A directed acyclic AOG is proposed to explore the space of latent structures effectively, and (ii) A DP algorithm is used to search the huge space of latent structures efficiently to find the globally optimal AOT. The proposed method was evaluated on the detection benchmark of 20 object classes in both PASCAL VOC2007 and VOC2010 and obtained better performance than state-of-the-art baseline methods.

In our on-going work, we are studying (i) how to incorporate the appearance Or-node for different types of features complementary to HOG (such as LBP and color) into our AOG explicitly, and (ii) how to share both part appearance and part configuration in learning with AOG among different sub-categories with-in the same object class, and between different object classes.

Acknowledgment X. Song and Y. Jia are supported by NSFC 90920009 and Specialized Research Fund for the Doctoral Program 20121101110035, X. Song is also supported by 973 Program of China 2012CB316300, and T. Wu and S.-C. Zhu are supported by DARPA MSEE project FA 8650-11-1-7149, MURI ONR N00014-10-1-0933, and NSF IIS 1018751.

References

[1] O. Aghazadeh, H. Azizpour, J. Sullivan, and S. Carlsson. Mixture component identification and learning for visual recognition. In *ECCV*, 2012.

[2] H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *ECCV*, 2012.

[3] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010.

[4] S. Branson, P. Perona, and S. Belongie. Strong supervision from weak annotation: Interactive training of deformable part models. In *ICCV*, 2011.

[5] R. G. Cinbis1 and S. Sclaroff. Contextual object detection using set-based classification. In *ECCV*, 2012.

[6] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. *IJCV*, 95(1):1–12, 2011.

[7] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[8] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[9] P. Felzenszwalb, R. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. "http://people.cs.uchicago.edu/ pff/latent-release4/".

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

[11] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[12] R. Girshick, P. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. "http://people.cs.uchicago.edu/ rbg/latent-release5/".

Figure 7. Examples of detection results of the 20 object categories on PASCAL VOC2007 testing dataset. For clarity, we show the object bounding boxes only for some detection results of objects which are relatively small in images.

[13] C. Gu, P. Arbelaez, Y. Lin, K. Yu, and J. Malik. Multi-component models for object detection. In *ECCV*, 2012.

[14] F. Khan, R. Anwer, J. van de Weijer, A. Bagdanov, M. Vanrell, and A. M. López. Color attributes for object detection. In *CVPR*, 2012.

[15] R. Mottaghi. Augmenting deformable part models with irregular-shaped object patches. In *CVPR*, 2012.

[16] B. Pepik, M. Stark, P. V. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *CVPR*, 2012.

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

[18] Z. Si and S.-C. Zhu. Learning and-or templates for object recognition and detection. *TPAMI*, 2013.

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

[20] X. Song, T. Wu, Y. Xie, and Y. Jia. Learning global and reconfigurable part-based models for object detection. In *ICME*, 2012.

[21] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan. Contextualizing object detection and classification. In *CVPR*, 2011.

[22] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

[23] S. Wang, Y. Wang, and S.-C. Zhu. Hierarchical space tiling for scene modelling. In *ACCV*, 2012.

[24] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, 2009.

[25] J. Zhang, K. Huang, Y. Yu, and T. Tan. Boosted local structured hog-lbp for object localization. In *CVPR*, 2011.

[26] J. Zhu, T. Wu, S.-C. Zhu, X. Yang, and W. Zhang. Learning reconfigurable scene representation by tangram model. In *WACV*, 2012.

[27] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010.

[28] S. C. Zhu and D. Mumford. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, 2(4):259–362, 2006.