in Proc. 11th IEEE International Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil, October 2007

Learning the Taxonomy and Models of Categories Present in Arbitrary Images

Narendra Ahuja and Sinisa Todorovic
Beckman Institute, University of Illinois at Urbana-Champaign
{ahuja, sintod}@vision.ai.uiuc.edu

Abstract

This paper proposes, and presents a solution to, the problem of simultaneous learning of multiple visual categories present in an arbitrary image set and their inter-category relationships. These relationships, also called their taxonomy, allow categories to be defined recursively, as spatial configurations of (simpler) subcategories each of which may be shared by many categories. Each image is represented by a segmentation tree, whose structure captures recursive embedding of image regions in a multiscale segmentation, and whose nodes contain the associated region properties. The presence of any occurring categories is reflected in the occurrence of associated, similar subtrees within the image trees. Similar subtrees across the entire image set are clustered. Each cluster corresponds to a discovered category, represented by the cluster properties. A (subcategory) cluster of small matching subtrees may occur within multiple clusters (categories) of larger matching subtrees, in different spatial relationships with subtrees from other small clusters. Such recursive embedding, grouping and intersection of clusters is captured in a directed acyclic graph (DAG) which represents the discovered taxonomy. Detection, recognition and segmentation of any of the learned categories present in a new image are simultaneously conducted by matching the segmentation tree of the new image with the learned DAG. This matching also yields a semantic explanation of the recognized category, in terms of the presence of its subcategories. Experiments with a newly compiled dataset of four-legged animals demonstrate good cross-category resolvability.

1. Introduction

This paper is about unsupervised extraction of subimages having similar appearances and topology from a given set of arbitrary images, as well as discovering the spatial relationships among subimages belonging to all discovered sets of similar subimages. Topology here refers to recursive embedding of homogeneous regions, captured in a multiscale image segmentation. Subimages are called 2D objects, and each set of similar subimages in the image set is said to define a category of objects. The categories, in general, have hierarchical mutual relationships. Thus, a category may be defined recursively by specifying properties and configurations of its subcategories. Such hierarchical category definitions may also include sharing of simple categories by more than one, complex categories. For example, category "leg" is shared by all legged animals, and, in turn, "leg" is an articulated combination of the simpler category of elongated shapes, which also occurs in the definitions of the categories of stools and scissors. It is reasonable to expect that simple categories occur more frequently in real-world images, and their occurrences exhibit smaller variations than encountered in more complex categories. This makes learning of simpler categories more robust. In turn, representation and learning of complex categories becomes more compact by exploiting the simpler descriptions of their subcategories, and more efficient as subcategory sharing makes the complexity of representation/learning of multiple categories sublinear in the number of categories. In this paper, we refer to the recursive representation of complex categories as spatial configurations of smaller, simpler subcategories as the taxonomy of the categories.

This paper is aimed at solving the following related problems: 1) simultaneous discovery of multiple categories of different complexities occurring in an arbitrary image set; 2) learning category-specific photometric (color), geometric (area, shape), and topological (recursive region embedding) properties; 3) identification of categories of different complexities and their relationships, i.e., learning the taxonomy; 4) simultaneous detection, recognition and segmentation of *all* objects from the learned categories present in a previously unseen image; and 5) retrieving the semantic explanation of why a category is found in a new image, i.e. in terms of the simpler categories detected and the learned taxonomy. Below, we first point out differences between this paper and prior work, and then present an overview and main contributions of our approach.

Prior work: In general, object recognition approaches consist of four major stages: feature extraction, category representation, training, and recognition. We review here the

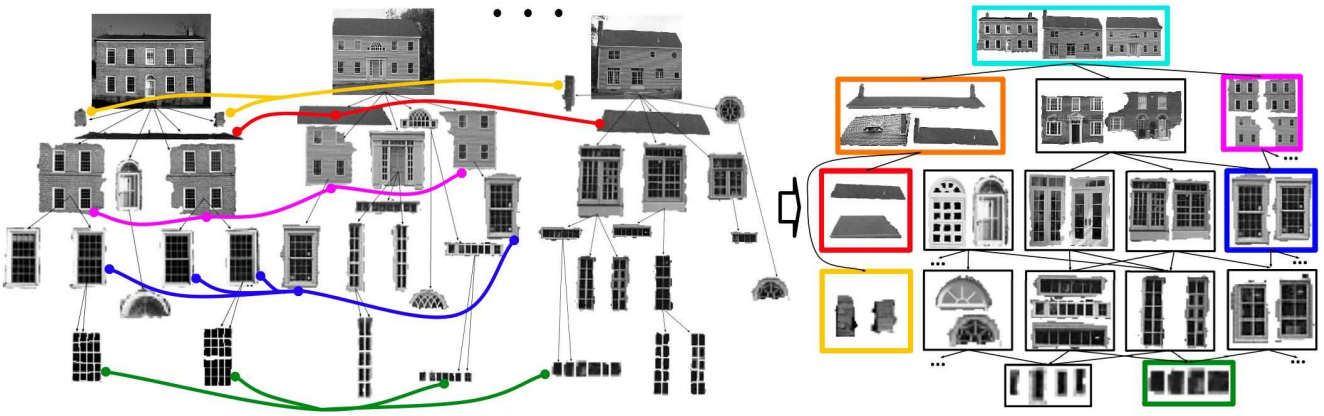


Figure 1. The results of our algorithms: (left) segmentation trees of house images; (right) learned taxonomy. The subtrees representing categories "roofs" (red), "windows" (blue), "window-panels" (green), etc., are each clustered, since they have similar photometric, geometric and topological properties. The subtrees of "window-panels" are contained within the subtrees of "windows" and "doors;" therefore, categories "windows" and "doors" share category "window-panels." The subtrees of "roofs" and "chimneys" are not contained within a larger subtree but co-occur in the segmentation trees; therefore, they define co-occurrence category "roof-chimney" (orange). Similarly, "roof-chimney," "quad-window-groups" (pink), and "windows-door" define co-occurrence category "house-front" (cyan).

state of the art with respect to each of these stages. Regarding the first stage, most recent work uses local features (e.g., keypoints [9], and curve fragments [10]). There is also a significant number of region-based approaches [12, 13, 14]. Advantages of region features are that (i) they are higher-dimensional and thus in general richer descriptors than local features, (ii) their boundaries often coincide with relevant boundaries of objects, facilitating simultaneous object detection and segmentation, and (iii) they enable easy use of the constraints dealing with spatial cohesiveness and multiresolution structure of images. For these reasons, we use image regions as features. For category representation in the second stage, most approaches partition features into clusters, called "parts," whose boundaries are in general distinct from those of the true object parts. They represent the objects as either a planar, or hierarchical graph of these "parts." For example, the constellation model [3] is a planar graph with a user-specified number of "parts," configured in a known model structure. The hierarchical models of, e.g., [5, 4, 15] are derived by hierarchical clustering of features, where smaller feature clusters can be shared by larger ones. The structure of these hierarchical models is typically controlled by a pre-specified hierarchy depth. Our hierarchical model (i.e., taxonomy) differs in that it has an a priori unknown hierarchy depth, and arbitrary number of subparts forming arbitrary spatial layout configurations, all of which are learned from the image set. Also, our taxonomy encodes sharing of entire categories, while prior work is concerned only with feature sharing. Related to ours are approaches that use the graph-theoretic framework to learn hierarchical models of categories [8, 14]. In [14], the tree-union of a single category is learned. Instead, we simultaneously discover multiple categories, and learn a more general graph than tree-union, i.e., the taxonomy of shared categories. For training, most prior work requires each training image be labeled with the category or a few categories it contains. For example, [2] solves the problem of translation from visual features to semantics, provided training images are labeled with semantics. Recently, the required degree of supervision has been reduced such that each training image does not have to be labeled [13, 14]. However, unlike our approach, [13] requires specification of the total number of categories present. Object recognition, in stage four, is typically evaluated only through image classification (category present/absent) [3, 6]. Few approaches, like ours, precisely delineate the boundaries of detected objects [14, 13].

Overview of our approach: (1) An image is represented by a segmentation tree [1, 14] which captures the low-level, spatial and photometric image structure. Nodes at upper levels correspond to larger segments, while their children nodes capture embedded, smaller details (e.g., the quad-window-group nodes in Fig. 1 are parents to the window nodes). (2) Category instances (e.g., roofs, doors, windows in Fig. 1) appear as similar subimages, whose corresponding subtrees are accessible in the segmentation trees. To identify the instances, we measure the similarity of all segments across the image set, in terms of their intrinsic photometric, geometric and topological properties, as well as in terms of the same properties of their embedded subregions. (3) The identified similar subimages are clustered, and the resulting clusters are treated as evidence and exact instances of the categories present. The similar subimages within a cluster (e.g., of all doors) together provide for robust learning of the subtree properties characterizing the associated category. (4) The clusters containing less complex subimages are associated with more common, simple cate-

gories (rectangular panels). These subimages form components of the hierarchical definitions of subimages in other clusters representing more complex categories (windows, doors). The clusters inherit the containment properties of their constituent subimages, which allows us to establish hierarchical, containment links between the clusters (child link from the window cluster to the panel cluster), yielding a directed acyclic graph (DAG). The root nodes of the DAG represent the set of most complex categories, while those near the leaves represent the simplest, often most shared subcategories, as illustrated in Fig. 1. (5) The categories found in (3) may indeed represent different parts of a complex category. (roof and front wall of the house), and, may not belong to any single subtree in the segmentation tree. The detection of the parts can be used to encode such "co-occurrence" categories (house front marked cyan in Fig. 1). (6) To recognize the occurrence of any of the learned categories in a new image, its segmentation tree is searched for matches with the DAG. Any matches found denote the occurrences of the corresponding categories as well as all the associated subcategories. The subcategories, along with their hierarchical structure within the DAG, serve as a semantic (category-space) explanation of why the category is found. Simultaneously, the matches also specify the exact boundaries of the detected objects.

Contributions: 1) To our knowledge, this is the first solution to completely unsupervised learning of hierarchical and sharing relationships, or taxonomy, of visual categories. 2) Unlike in prior work, each unlabeled training image in our case may contain multiple instances of multiple target categories, whose total number is unknown. 3) Our approach derives a generative, hierarchical model of a category's image structure, instead of learning a classifier of pre-specified categories. 4) While prior work learns only sharing of features among known categories, and establishes similarity relationships between the categories with respect to the number of shared features, we instead learn sharing of entire categories. 5) Recognition capabilities of prior work are extended by providing a semantic basis of recognition. 6) We introduce a new co-occurrence category, which cannot be handled by most existing approaches (e.g., [14]). 7) We introduce a new graph similarity measure. 8) A new dataset of four-legged animals is compiled and used for evaluating resolving subtle cross-category differences.

Next, we describe our image representation and estimation of similarity between subimages in Sec. 2, clustering of similar subimages in Sec. 3, organizing the clusters of similar subimages into a DAG in Sec. 4, and finally experimental evaluation of these algorithms in Sec. 5.

2. Locating Subimages of Potential Categories

This section describes Steps 1 and 2 of our approach. Since these steps are similar to those used in [14], we here only point out the major differences.

Images are represented by trees obtained by a multi-scale segmentation algorithm, presented in [1, 14]. Any cutset of the segmentation tree corresponds to one possible image segmentation, while parent-child node relationships capture recursive region embedding. The number of nodes ($\approx$ 150–200), branching factor ($\approx$ 0–5), and the number of levels ($\approx$ 10–15) in different parts of the segmentation tree are image dependent. A vector $\boldsymbol{\psi}_v$ of region properties is associated with each node v in the tree, defined relative to the corresponding properties of v's parent-node u, to allow scale and rotation-in-plane recognition invariance. The region's principal axis is estimated as the eigenvector of matrix $\frac{1}{\mu_{00}}\left[\begin{smallmatrix}\mu_{20} & \mu_{11}\\ \mu_{11} & \mu_{02}\end{smallmatrix}\right]$ associated with the larger eigenvalue, where μ_{pq} are the region's standard central moments. The components of $\boldsymbol{\psi}_v$ are as follows: (1) gray-level contrast g_v between v and its parent u; (2) normalized intensity variance $\sigma_v^2 = \frac{\text{var}(v)}{\text{var}(u)}$; (3) normalized area $a_v \triangleq \frac{\text{area}(v)}{\text{area}(u)}$; (4) area dispersion $\text{AD}_v \triangleq \sum_{w\in C(v)}(a_w - \bar{a}_w)^2$ over v's children $w \in C(v)$; (5) bending energy (a measure of boundary jaggedness) $\text{BE}_v \triangleq \frac{1}{L_v}\sum_{i=1}^{L_v}\kappa_i^2$, where L_v is the length in pixels of v's boundary, and $\{\kappa_i\}$ is an array of curvature values computed at each boundary pixel from the standard 8-connected chain code; (6) squared perimeter over area, $\text{PA}_v \triangleq \frac{L_v^2}{\text{area}(v)}$; (7) angle φ_v between the principal axes of v and u; (8) normalized displacement $\overrightarrow{\Delta}_v \triangleq \frac{d_{uv}}{\sqrt{\text{area}(u)}}\overrightarrow{r}_{uv}$, where d_{uv} ($\measuredangle \overrightarrow{r}_{uv}$) is the distance (unit vector) from the centroid of u to that of v; (9) context vector $\overrightarrow{\Phi}_v \triangleq \sum_{w\in\mathcal{N}_v}\frac{\text{area}(w)}{d_{vw}^2}\overrightarrow{r}_{vw}$ that records the general direction in which v sees its sibling regions $w \in \mathcal{N}_v$, and disallows matching of scrambled layouts of regions. Each entry of $\boldsymbol{\psi}_v$ is normalized to take a value in the interval $[0, 1]$.

Having obtained the tree representation of a given image set, $\mathbb{T} = \{t_1, t_2, \ldots t_N\}$, we proceed with estimating the similarity between all pairs of subimages (i.e., subtrees) in $\mathbb{T}$. Accordingly, we define a similarity measure, $\mathcal{S}_{vv'}$, between two regions (nodes), v and v', in terms of their intrinsic region properties, $\boldsymbol{\psi}_v$ and $\boldsymbol{\psi}_{v'}$, as well as the properties of their embedded subregions, w and w', i.e., descendant nodes underneath v and v' in the segmentation trees. $\mathcal{S}_{vv'}$ is computed by the well-known tree matching algorithm presented in [11, 16, 14], which for two given trees finds their common subtrees. Given two trees $t = (V_t, E_t)$ and $t' = (V_{t'}, E_{t'})$, where V and E are the sets of nodes and edges, the goal of matching is to find the topologically consistent subtree isomorphism, $f: U_t \rightarrow U_{t'}$, where $U_t \subseteq V_t$ and $U_{t'} \subseteq V_{t'}$, which maximizes their similarity measure

$$\mathcal{S}_{tt'} \triangleq \max_f \sum_{(v,v')\in f}(r_v + r_{v'} - m_{vv'}), \qquad (1)$$

where r_v is the saliency of region v, and $m_{vv'} \triangleq |r_v - r_{v'}|$ is the cost of matching $v \in t$ and $v' \in t'$ in bijection f. The

region saliency is defined as a linear combination of region properties $r_v \triangleq \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\psi}_v$, where $\boldsymbol{\xi}$ is a vector of weighting coefficients so that $||\boldsymbol{\xi}||=1$ and $\boldsymbol{\xi}\geq 0$. From (1), for all node pairs $(v, v')\in t\times t'$, $\mathcal{S}_{vv'}$ can be computed recursively, bottom-up:

$$\mathcal{S}_{vv'} = r_v + r_{v'} - m_{vv'} + \textstyle\sum_{(w,w')\in\mathcal{C}_{vv'}} \mathcal{S}_{ww'}, \quad (2)$$

where $\mathcal{C}_{vv'}$ is the maximum weight clique of the association graph constructed from all descendants (w, w') of the node pair (v, v') [11, 16, 14]. If each of the two trees has no more than $|t|$ nodes, complexity of their matching is $O(|t|^4)$.

The main deficiency of the similarity measure, given by (2), versus our objectives is that the measure depends on the size of the hierarchies being matched. In particular, the matches of more structured image regions are favored over those of simple, homogeneous regions. It is not clear if and to what extent should the similarity between two nodes depend on their subtree depths and branching factors. For the purpose of this paper, we have chosen to make the match quality depend only on the intrinsic matches between the paired nodes, without any direct dependence on the subtree depths. To this end, we weight the contributions to similarity of each node-pair in (2) as follows:

$$\tilde{\mathcal{S}}_{vv'} \triangleq \rho_{vv'}(v, v')(r_v + r_{v'} - m_{vv'}) + \sum_{\mathcal{C}_{vv'}} \rho_{vv'}(w, w')\tilde{\mathcal{S}}_{ww'}, \quad (3)$$

where the weights $\rho_{vv'}(w, w')$ make the contributions of the regions w and w' proportional to the relative areas they occupy within v and v'. We define $\rho_{vv'}(w, w')$ as the total outer-ring area of $\{w\cup w'\}$ that is not occupied by the other descendants of v and v' in $\mathcal{C}_{vv'}$, expressed as a percentage of the total area of $\{v \cup v'\}$. With this new similarity measure, the matching algorithm yields a set of pairs of matched subimages drawn from images in the entire set, along with their similarity values, which are then used for identifying different categories present.

3. Discovering Multiple Categories

The matched subtree pairs obtained above link multiple occurrences of the same category with high similarity values. Thus, all occurrences of the same category across the image set are expected to be transitively connected by a sequence of high-value links. This section describes the next step (Step 3 in Sec. 1) in our algorithm, which is aimed at clustering together all highly similar subimages. The result is one cluster per category thus discovered. Since we do not know how many categories exist in the data, and what the extent of their intra- and inter-category variations are, we conduct hierarchical, agglomerative, binary clustering. The result of this hierarchical clustering can be easily transformed into a particular categorization, given a desired degree of cross-category resolvability (e.g., by merging together all clusters whose similarities are closer than the specified level of sensitivity).

We conduct the standard, complete linkage, agglomerative clustering over the entire set of regions $v\in\mathbb{T}$ from all the images, where the two most similar clusters are merged into a larger one at each stage, provided that none of the nodes within the clusters has a descendant or ancestor present in the other. This is done until there are no more clusters that can be merged. The pairwise cluster merging is based on the minimum intercluster similarity value (Hausdorff distance). Some of these mergers may combine two clusters containing instances of the same category, while others might force two different populations to merge. Although each merger selects the best candidates available for merging, in the latter case it combines two categories which we may want to keep as separate, because the difference in their geometric, photometric and topological properties is above our desired sensitivity level. In contrast, in the former case, the merger is desirable and enlarges the set of samples in the common category of the merged clusters. To formally evaluate the validity of agglomerative merging of two clusters at any given stage, we will assume that similarity values $\tilde{\mathcal{S}}_{vv'}$ within a cluster are samples drawn from a probability density function (pdf) characteristic of the associated category. Then, erroneous merging of two distinct categories, into an artifact category, would amount to treating two different pdf's as the same.

Distinguishing Categories: To prevent erroneous category merging, we use the well-known Kolmogorov-Smirnov test (KS-test). The null hypothesis for the KS-test is that the two sample sets of similarity values are drawn from the same continuous pdf, while the alternative hypothesis is that they are drawn from different pdf's. The null hypothesis is rejected if the test is significant at level α, which we set to the standard value of $\alpha=5\%$, thus quantifying our level of sensitivity to inter-category differences. The attractive characteristics of the KS-test are that it does not require assumptions about the distribution of data, and binning of the samples (as, e.g., χ^2-test), and that the distribution of the KS-test statistic itself does not depend on the underlying pdf being tested. Rejection of the null hypothesis results in the retention of both clusters as distinct categories. Since the KS-test is more reliable over large clusters, we first create the complete binary merger tree, and then prune erroneous mergers top-down, which results in a forest of binary cluster mergers. The pruning process ends when no null hypothesis is rejected. The roots of the agglomerative clustering hierarchy (i.e., the largest clusters) that remain at the end of the pruning process are taken as representing the categories discovered in the image set. Each cluster root is guaranteed to be a category by itself, because it has passed the KS-test for being distinct from all others. Each category discovered is assigned a label, c, and a vector, $\boldsymbol{\psi}_c$, which is the mean

of the property vectors $\boldsymbol{\psi}_v$ associated with all subtrees v contained within cluster c.

4. Taxonomy of All Categories Discovered

This section presents Steps 4 and 5 of our approach (Sec. 1) aimed at organizing the clusters of similar subtrees (described in Sec. 3) into a DAG, and thus obtaining the taxonomy of the discovered categories. Each cluster contains the transitive closure of matched pairs of subtrees across the image set. Subtrees in one cluster may contain those in another cluster. These subtree containment relationships from the original segmentation trees can be directly extended to the clusters (i.e., categories). If a subtree in cluster c_1 is contained within a larger subtree in cluster c_2, then c_1 becomes a child of c_2. When subtrees in c_1 are contained within larger subtrees in a number of other clusters – the case of sharing a simpler category by many more complex categories – c_1 may have more than one parent cluster. This can be represented by a directed acyclic graph (DAG), whose nodes are the categories and edges capture their parent-child relationships. Each category may have an arbitrary number of child and parent links emanating from it. The property vector $\boldsymbol{\psi}_c$ of category c (explained in Sec. 3) is associated with the node c in the DAG.

The image set may also contain categories that are more complex than those at the highest level of the taxonomy obtained. One such type of a complex category may be defined by simultaneous occurrence of some of the discovered categories in the images (house front in Fig. 1). Such a co-occurrence category appears as a forest of disjoint subtrees in the segmentation trees, and thus could not be discovered by using the similarity measure defined in (3), since it accounts only for the substructure within given regions. Discovering co-occurrence categories can be easily addressed by explicitly checking for simultaneous appearance of already discovered categories. In case such a category is discovered, we introduce a new node in the DAG, and connect it as a parent to its co-occurring subcategories. The newly obtained co-occurrence categories are recursively checked if they concurrently appear with any other categories.

Given a new image, all instances of the learned categories present in the image are simultaneously identified by matching the segmentation tree of the new image with the DAG, using the same algorithm as used in Sec. 2.

5. Results

Experiments are designed to evaluate the algorithm's capability to: (i) extract the taxonomy from a given set of unlabeled training images; (ii) simultaneously detect, recognize and segment all instances of the learned categories present in a test image; (iii) resolve small cross-category differences; and (iv) provide a semantic explanation as to why the categories are found in the test image. To this end, we use two benchmark datasets, and another newly compiled one. Specifically, we use 40 categories from Caltech-101 [3] (including 435 faces, 800 motorbikes, 800 airplanes, 526 cars-rear), as well as 108 UIUC multiscale car images. Each Caltech-101 image contains only a single, prominently featured object from the category. The Caltech cars-rear and the UIUC cars-side increase complexity, since the images contain multiple cars, which appear at different scales, have low contrast with the textured background, and may be partially occluded. UIUC images also contain other frequently occurring categories (e.g., trees, buildings), allowing us to test identifying multiple instances of multiple categories per image. However, the main deficiency of these benchmark sets is that their categories significantly differ in appearance and topology, and thus are not convenient for evaluating how well the algorithm resolves subtle cross-category differences, and identifies subcategory sharing. To address this issue, we have compiled a new dataset, referred to as Animals, containing 200 images of horses, cows, camels, deer, sheep and goats (Fig. 4). This dataset is the most challenging of the three, since each image contains multiple instances of several very similar categories (e.g., horses and deer), co-occurring in the images at different scales, possibly partially occluded. Since the animals are similar, they share a number of similar parts, which should be captured by our model. The animals also have category-specific, discriminative subcategories (e.g., only deer have antlers), which allow for categorization, and thus should be learned as non-shared subcategories in the taxonomy.

Multiple-category learning with the Caltech dataset is carried out on a training set that contains a total of N_{cat} target categories, where $N_{\text{cat}}=\{4, 10, 20, 30, 40\}$ but is unknown to the algorithm. When $N_{\text{cat}}=4$, we use the Caltech faces, motorbikes, airplanes, and cars-rear. For $N_{\text{cat}}\geq 10$, in addition to these four, the training set contains a mix of other randomly drawn categories. A number of images, $N_{\text{train}}^{\text{cat}}=\{5, 10, 15, 20, 25, 30\}$, are randomly drawn per each category, resulting in the training set of size $N_{\text{train}}=N_{\text{cat}}\times N_{\text{train}}^{\text{cat}}$. From the remaining images, the test set is randomly drawn so that it contains $N_{\text{test}}^{\text{cat}}=50$ images per each target category, including the background category, totaling $N_{\text{test}}=(N_{\text{cat}}+1)\times N_{\text{test}}^{\text{cat}}$ test images. People in the Caltech test and training images for faces are different. Varying $N_{\text{train}}^{\text{cat}}$ and N_{cat} allows us to test the algorithm's performance against the number of available training samples, and its sensitivity to the total number of categories to be learned. As for UIUC cars and Animals, the training sets contain $N_{\text{train}}=\{10, 40\}$ randomly drawn images from the entire dataset, respectively, while the remaining images are used for testing. Detection, recognition and segmentation is performed simultaneously, by matching the learned DAG (i.e., taxonomy) with the test-image trees. Each experiment

is repeated 10 times to estimate the average performance.

For quantitative evaluation, we define detection, segmentation and recognition errors. We use manually delineated outer contours of each category instance appearing in the test images as ground truth. Those matched subtrees in the test images whose similarity measure is larger than a specified threshold are adjudged as detected objects. The threshold is varied to plot the recall-precision curves, while for the purposes of showing specific results in tables and figures, we use use the similarity-measure threshold yielding the highest F-measure, where $F \triangleq 2 \cdot \text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall})$. Let the area that a matched subtree covers in the test image be A_d, and the ground-truth object area be A_g. Then, the matched subtree is said to be false positive (FP) if $\frac{A_d \cap A_g}{A_d \cup A_g} < 0.5$. The remaining cases are declared true positives (TP). Segmentation error is defined as the ratio $\frac{\text{XOR}(A_d, A_g)}{A_d \cup A_g}$. The recognition performance is evaluated only on the TP's as follows. Each node in the DAG represents a cluster of subimages from one learned category. For testing purposes, the meaning of each learned category is assigned manually, by observing the majority of entries in the corresponding cluster. Thus, for example, if mostly faces are grouped in cluster c, then category c will mean faces. Then, recognition is done by assigning to each TP this user-specified meaning of the matched node in the DAG, and if different from the ground truth (verified by visual inspection) the TP is declared erroneously recognized. Depending on specific training images in each experiment, a different number of categories of varying complexities are discovered. For testing purposes here, we focus only on labeled categories in the Caltech dataset (i.e., faces, motorbikes, etc.), cars in the UIUC dataset, and the six animal categories in Animals. We will call them target categories. Evaluation of other discovered categories (road, grass, sky, etc.) is omitted for brevity.

Qualitative evaluation – Segmentation: Figs. 2–5 demonstrate high accuracy in simultaneous object detection and segmentation on Caltech, UIUC and Animals images, for the training sets containing $N_{\text{train}} = \{40, 10, 40\}$ images, respectively. Detected instances of the target categories declared TP's are shown in Figs. 2, 3 and 5 by drawing their outer contours on the original image, and in Fig. 4 by masking undetected image parts. Each TP in the figures is correctly recognized. Segmentation performance is good even in cases when object boundaries are jagged and blurred (e.g., motorbikes in Fig. 3), when objects are partially occluded (e.g., faces in Fig. 3), and when objects from the same category occlude each other, forming a complex region topology with low-intensity contrasts (e.g., small and large camel in Fig. 4). Objects that are not detected, for the most part, have low intensity contrasts with the surround, and thus do not form category-characteristic subtrees in the segmentation tree that can be matched with the DAG.

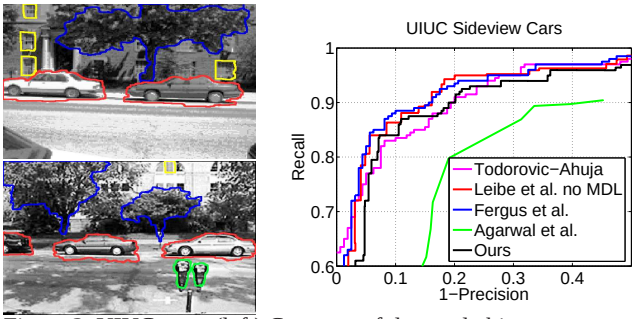


Figure 2. UIUC cars: (left) Contours of detected objects are overlaid on the original. In addition to "cars" (red), the DAG encodes "windows" (yellow), "park-meters" (green), "trees" (blue) among other categories. $N_{\text{train}} = 10$. (right) Recall-precision curves for UIUC cars. The comparison is with [14, 3, 7].

Qualitative evaluation – Semantic Explanation: Fig. 5 illustrates a part the DAG learned over the training set shown in Fig. 4a. Specifically, the matched parts of a given test image, showing a horse and five cows, depict the corresponding DAG nodes. As can be seen, the rider and horse are matched with a DAG node representing "rider-on-horse" category, learned from the training images that do indeed contain horseback riding scenes. This complex category is found, because its subcategories "rider" and "partial-horse" are identified. Similarly, only four cows are detected, where the three are recognized as category "cow," and one as "spotted cow," which is a co-occurrence category learned from frequent co-occurrences of disjoint cow parts. The DAG also provides an explanation that "horse" and "cow" share learned subcategories "hind leg" and "muzzle," and, further down the taxonomy, "limbs." We do detect and recognize "hind leg" of the occluded, leftmost cow, and do not confuse its contours with those of the occluding cow in front. Such identification of subcategory instances can be used in some applications with a higher level of supervision for indicating the presence of partially visible parent categories. Dogs appearing in the test image are not detected, as they are not present in the training set, and thus are not learned. In Fig. 4b we also depict subcategories of each of the six target categories, which are not shared among them. These subcategories are discriminative, category-specific, and facilitate cross-category resolvability. These results suggests that the discovered taxonomy is meaningful.

Quantitative evaluation: Averages of object detection, segmentation, and recognition errors are summarized in Table 1. In comparison with [14], we outperform their 9.3% segmentation error obtained for the simpler, single-scale UIUC cars images, and have similar performance to their 6.8% segmentation error on Caltech faces (within a standard deviation). Also, our segmentation error is close to 6.0% reported in [17] for a much simpler dataset of sideview cars,

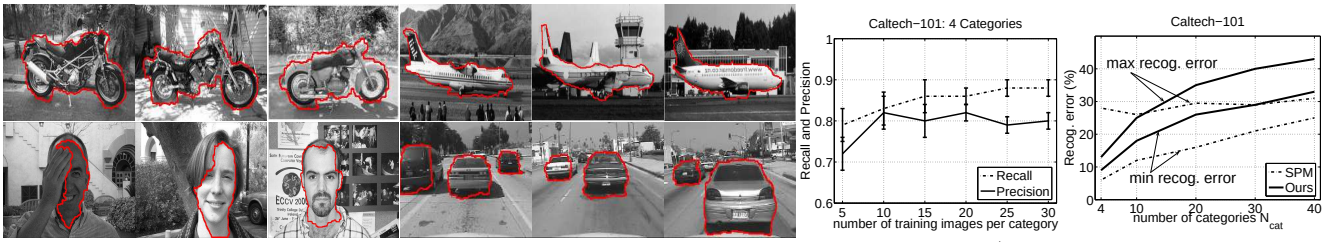
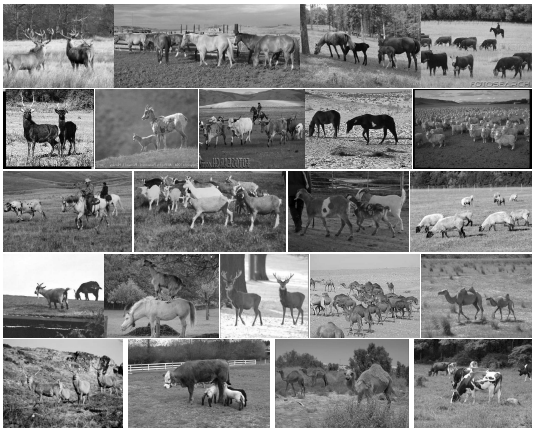
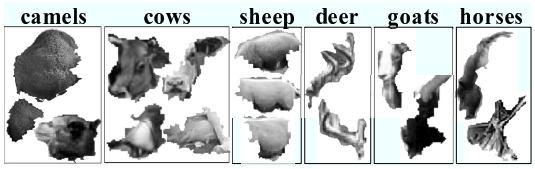


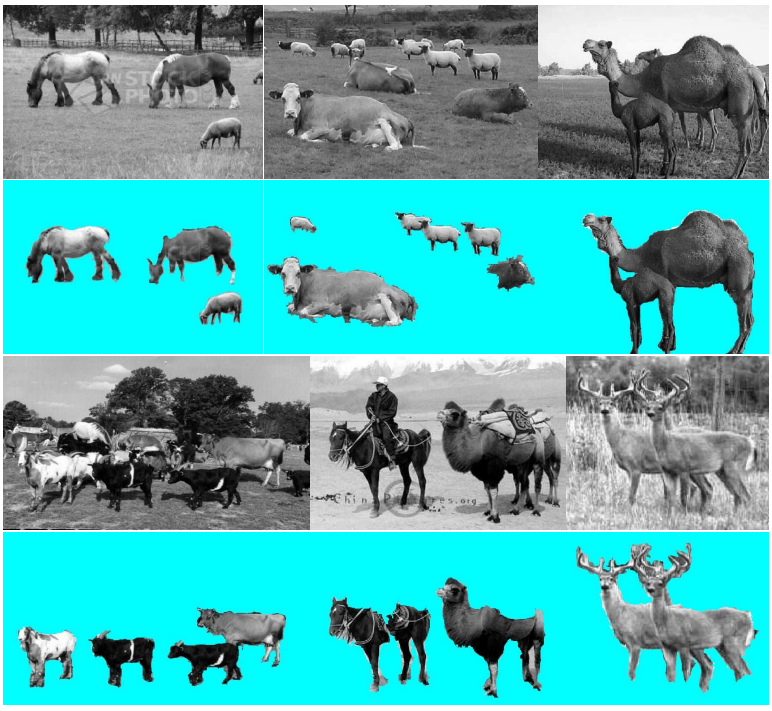
Figure 3. Caltech-101: (left) Contours of detected objects are overlaid on the original. N_{cat}=4 and $N^{\text{cat}}_{\text{train}}$=10. (right) The plots of recall-precision rates against the number of training images per category for N_{cat}=4, and recognition errors versus the number of randomly drawn categories present in the training set for $N^{\text{cat}}_{\text{train}}$=10. The comparison is with the spatial pyramid matching (SPM) approach [6].



(a) Sample training images from Animals: each image may contain multiple instances of very similar animals, posing a challenge for cross-category resolvability.



(b) Learned discriminative, non-shared subcategories of the target categories.



(c) Masked out are the image parts not occupied by detected instances of target categories.

Figure 4. Animals: detection, recognition and segmentation on the test images shown in (c; rows 1,3) using the DAG learned on the training images shown in (a). The DAG successfully resolves the subtle differences among the animals, since it learns the subcategories of each of these six that are not shared, namely: camel's hump and head, cow's udder and head, deer's antlers, goat's beard and horns, horse's reins and mane; shown in (b) are the parts of the test images in (c) that got matched with the non-shared DAG nodes.

as compared to the UIUC multiscale dataset we use. Our recall and precision rates for the Caltech faces are similar to those of [14], reporting recall 84.6% and precision 78.2%, but they learn "faces" as a single category, while we simultaneously learn N_{cat}=4 categories. Recall-precision curves over the UIUC cars are compared with those of [14, 3, 7] in Fig. 2. Despite the fact that most of these methods use simpler and more forgiving evaluation metrics (e.g., bounding boxes containing detected objects), our detection rates can be seen to be very close to the state of the art. For the Caltech faces, motorbikes, airplanes and cars-rear, we also plot recall-precision rates against the number of training images per category in Fig. 3. As the training set becomes larger, we get only modest improvements after reaching a certain size of the training set ($N^{\text{cat}}_{\text{train}}$>20 for N_{cat}=4 Caltech categories). Finally, we plot the recognition error versus the number of categories present in the training set, randomly selected from the Caltech database, in Fig. 3. These results are compared against the best classifier on Caltech-101 that uses spatial pyramid matching (SPM) [6].

Parameters and Run-time: Since the entries in region property vector $\boldsymbol{\psi}_v$ are chosen to represent distinct characteristics of regions, we set $\boldsymbol{\xi}=\mathbf{1}/|\boldsymbol{\psi}_v|$, where $|\boldsymbol{\psi}_v|$ is the number of components in $\boldsymbol{\psi}_v$. The computation time of our training (steps presented in Sec. 2, 3, and 4) for the 40 Caltech training images took 4.5 hours on a 2.4GHz, 2GB RAM PC. Matching the DAG model with the test-image segmentation tree takes approximately 10-30s, depending

	Faces	Motorbikes	Airplanes	Cars rear	UIUC cars side	Horses	Cows	Deer	Sheep	Goats	Camels
Recall %	88.6±7.3	80.1±3.5	84.5±8.2	82.6±12.3	87.6±6.9	78.9±12.3	75.6±14.8	84.3±5.9	78.2±10.4	72.1±9.5	86.6±8.1
Precision %	78.1±5.8	87.6±3.8	87.1±11.4	78.6±11.3	81.6±6.4	82.8±7.5	79.9±11.7	82.2±4.9	78.1±7.2	78.8±5.3	86.2±7.2
Seg. error %	9.7±6.5	16.6±6.9	16.3±9.5	19.7±14.3	8.5±3.4	16.1±7.3	18.1±4.2	12.2±7.24	25.9±8.2	21.3±11.2	12.1±4.2
Rec. error %	6.4±4.6	7.7±7.3	4.7±4.5	8.6±4.8	4.7±2.8	8.6±3.2	7.2±4.1	9.2±2.4	9.2±6.1	15.9±6.4	3.6±4.9

Table 1. Average recall, precision, segmentation, and recognition error (in %) on the Caltech, UIUC, and Animals datasets for the highest F-measure; for Caltech-101 N_{cat}=4, $N_{\text{train}}^{\text{cat}}$=10; for UIUC cars N_{train}=10; for Animals N_{train}=40.

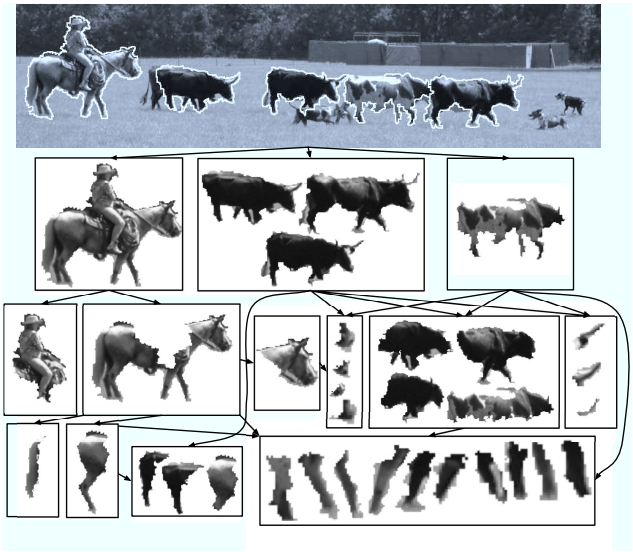


Figure 5. A part of the taxonomy of animals learned over the training images shown in Fig. 4. Contours of detected objects are overlaid on the original. Segments represent the matches of the DAG nodes with the corresponding parts of the test image. The DAG learned that cows and horses share hind legs and muzzles, while their respective non-shared subcategories are horns and tails.

on the number of nodes in these graphs.

6. Conclusions

We have proposed the problem of simultaneous learning of multiple visual categories present in an arbitrary image set, and their hierarchical relationships or taxonomy. Our solution yields state-of-the-art recognition and segmentation of all instances of multiple categories present in a test image. Moreover, a semantic explanation of each category found is provided in terms of the presence of its constituent subcategories.

Acknowledgment

The support of the Office of Naval Research under grant N00014-06-1-0101 is gratefully acknowledged.

References

[1] N. Ahuja. A transform for multiscale image segmentation by integrated edge and region detection. *IEEE TPAMI*, 18(12):1211–1235, 1996.

[2] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.

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

[4] S. Fidler, G. Berginc, and A. Leonardis. Hierarchical statistical learning of generic parts of object structure. In *CVPR*, 2006.

[5] Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. In *CVPR*, 2006.

[6] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[7] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV*, 2004.

[8] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Learning hierarchical shape models from examples. In *EMMCVPR, Springer LNCS*, 2005.

[9] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.

[10] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, 2006.

[11] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching hierarchical structures using association graphs. *IEEE TPAMI*, 21(11):1105–1120, 1999.

[12] B. Perrin, N. Ahuja, and N. Srinivasa. Learning multiscale image models of 2D object classes. In *ACCV, Springer LNCS*, 1998.

[13] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.

[14] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images. In *CVPR*, 2006.

[15] S. Todorovic and M. C. Nechyba. Dynamic trees for unsupervised segmentation and matching of image regions. *IEEE TPAMI*, 27(11):1762–1777, 2005.

[16] A. Torsello and E. R. Hancock. Computing approximate tree edit distance using relaxation labeling. *Pattern Recogn. Lett.*, 24(8):1089–1097, 2003.

[17] J. Winn and N. Jojic. Locus: learning object classes with unsupervised segmentation. In *ICCV*, 2005.