Learning Compositional Models for Object Categories From Small Sample Sets

Jake Porway[*], Benjamin Yao[*], Song Chun Zhu[*†]

Abstract

In this chapter we present a method for learning a compositional model in a minimax entropy framework for modeling object categories with large intra-class variance. The model we learn incorporates the flexibility of a stochastic context free grammar (SCFG) to account for the variation in object structure with the neighborhood constraints of a Markov random field (MRF) to enforce spatial context. We learn the model through a generalized minimax entropy framework that accounts for the dynamic structure of the hierarchical model. We first learn the SCFG parameters using the frequencies of object parts, then pursue spatial relations in order of greatest information gain. The learned model can generalize from a small set of training samples ($n < 100$) to generate a combinatorially large number of novel instances using stochastic sampling. To verify our learning method and model performance, we present plots of KL divergence minimization as the algorithm proceeds, and show that samples from the model become more realistic as more spatial relations are added. We also show the model accurately predicting missing or undetected parts for top-down recognition along with preliminary results showing that the model can learn a large space of category appearances from a very small ($n < 15$) number of training samples. This process is similar to "recognition-by-components", a theory that postulates that biological vision systems recognize objects as composed from a dictionary of commonly appearing 3D structures. Finally, we discuss a compositional boosting algorithm for inference and show examples using it for object recognition.

This article is a chapter from the book **Object Categorization: Computer and Human Vision Perspectives**, edited by Sven Dickinson, Aleš Leonardis, Bernt Schiele, and Michael J. Tarr (Cambridge University Press).

[*]University of California Los Angeles, Los Angeles, CA.

[†]Lotus Hill Research Institute, EZhou, China.

1 Introduction

Modeling object categories is a challenging task due to the many structural variations between instances of the same category. There have been many non-hierarchical approaches to modeling object categories, all with limited levels of success. **Appearance based models**, which represent objects primarily by their photometric properties, such as global PCA, KPCA, fragments, SIFTs, and patches [18, 19, 25, 27], tend to disregard geometric information about the position of important keypoints within an object. Thus, they are not well-suited for recognition in scenarios where pose, occlusion, or part reconfiguration are factors. **Structure based models**, which include information about relative or absolute positions of features, such as the constellation model and pictorial structures [8, 10, 27], are more powerful than appearance based approaches as they can model relationships between groups of parts and thus improve recognition accuracy, but are rarely hierarchical and, as such, cannot account for radical transformations of the part positions.

Very recently there has been a resurgence in modeling object categories using grammars [14, 23, 32]. Work by Fu[11, 30] and Ohta[20] in the 70's and 80's, and later by Dickinson and Siddiqi [6, 15, 22] introduced these grammars to account for structural variance. Han[13] and Chen[3]used attributed graph grammars to describe rectilinear scenes and model clothes, but these models were hardcoded for one category of images.

Many of the problems in previous approaches come from lack of a good definition for *object category* that captures what is invariant between instances of the same class. We define an object category as an equivalence class where the object parts and their relations to one another are the invariants that make instances in the same category equivalent. Bikes always have wheels, clocks always have hands, and these parts are always related similarly to one another. We can capture the variation in these commonalities through a constrained compositional model that defines the set of instances for each object category.

From a human vision persepctive, the work we present is similar in concept to the theories put forth by Biederman [1] about how biological vision systems recognize objects. In his Recognition by Components theory, Biederman postulates that objects are represented by compositions of 3D objects called "geons". These simple, repeatable geons can be combined under different deformations to form a vast number of complicated objects. Our representation is a similar, though more general, theory of recognizing objects as compositions of parts related by spatial and appearance relationships.

In this chapter we provide a way to generalize the minimax entropy framework to learn such a model for objects with dynamic structure. This model combines a SCFG to capture the variance and hierarchy of object parts with the relational constraints of an MRF. This framework accounts for the dynamic structure of the model, in which constraints may not always exist from instance to instance, by pursuing relations according to their frequency of occurrence. The MRF constraints that we add to the SCFG match two main statistics of the model:

1. The frequencies of part ocurrences
2. The statistics on the spatial layouts of parts.

We also discuss an inference algorithm called *compositional boosting* that can recursively detect and compose object parts to identify whole objects even in the presence of occlusion or noise.

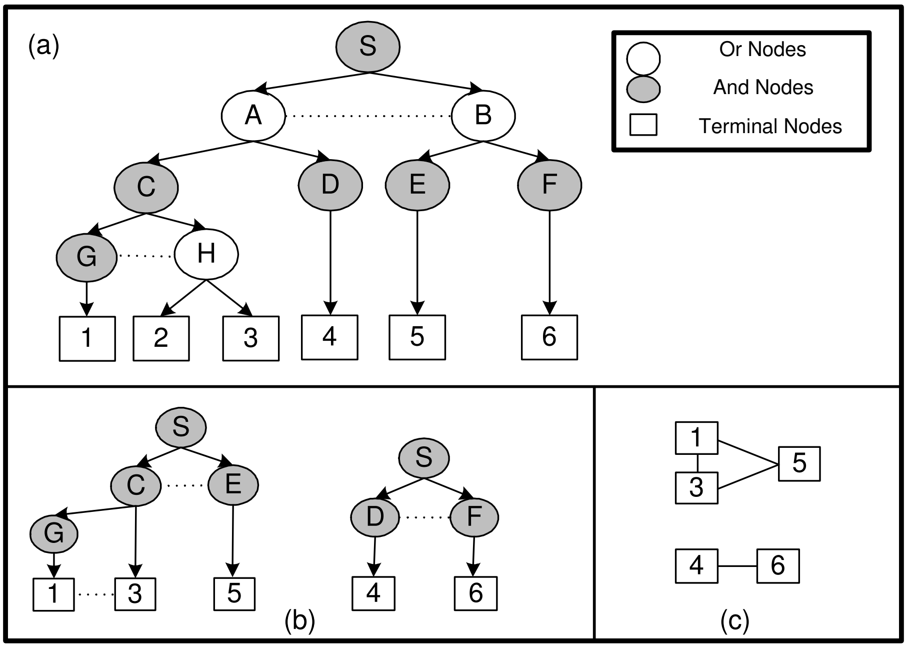


Figure 1: (a) An example of an "And-Or Graph". And nodes must decompose into all of their children, while Or nodes can only decompose into one. Constraints are shown as horizontal lines. (b) "Parse graphs", which are each one walk of the model. (c) "Configurations", which consist only of terminal nodes and any constraints they inherited.

2 Representation

The model we learn for object categories is a combination of a SCFG and an MRF, referred to as an "And-Or Graph" [1]. This model is like a language for an object category where parts of the object are analogous to words and parts of speech in language, and relational constraints model the context between them [3]. Figure 1(a) shows an example of an And-Or graph. An object is created by starting at the root of the graph and expanding nodes until only terminals remain, as in a SCFG. Node expansions in this structure can be thought of as "And" nodes, where one node expands into multiple nodes and "Or" nodes, which can only choose one of their child nodes to expand into. For example, node S is an And node, and expands into nodes A and B, which in turn are Or nodes and will only decompose into one child each. Figure 1(a) is a visualization of the following grammar

S → AB	C → GH	F → 6
A → C | D	D → 4	G → 1
B → E | F	E → 5	H → 2 | 3

The horizontal line between A and B represents a relational constraint. These constraints do not influence the node expansion, but act *a posteriori* on the selected nodes to

[1] The And-Or graph was previously used by Pearl in [21] for heuristic searches. In our work, we use it for a very different purpose and should not be confused with Pearl's work.

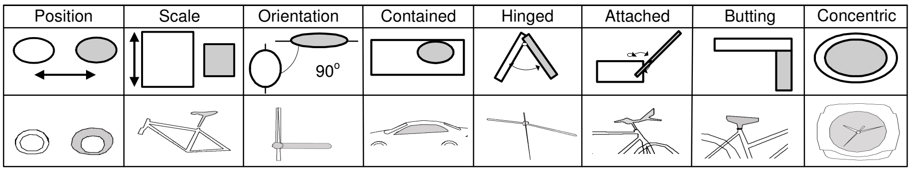


Figure 2: Visualization of possible pairwise relationships to be learned in the And-Or graph.

constrain their features, e.g. appearance. The constraints are inherited by any children of constrained nodes as well.

We define a *parse graph* **pg** as an instance drawn from the model, which is analogous to a sentence diagram in natural language. Figure 1(b) shows some *parse graphs* from our example model. The Or nodes are determined during a walk of the graph, fixing the parts, so only And nodes remain. We also define a *configuration* $\mathcal{C}$, which is simply the constrained parts from the parse graph with hierarchical information removed. This is equivalent to a sentence in natural language, and Figure 1(c) shows examples of configurations from our example.

This compositional model can be formalized as the tuple

$$G_{\text{and}-\text{or}} =< V_N, V_T, S, \mathcal{R}, \mathcal{P} > \tag{1}$$

$V_N = V^{Or} \cup V^{And}$ is the set of all non-terminal nodes, consisting of both the And and Or nodes. We define a switch variable $\omega(v)$ for $v \in V^{\text{or}}$, that takes an integer value to index its child node.

$$\omega(v) \in \{\emptyset, 1, 2, ..., n(v)\} \tag{2}$$

V_T represents a set of terminal nodes producable by the model. S is the root node that all instances start from. Together, these variables form the grammar structure of the model.

$\mathcal{R} = \{r_1, r_2, \ldots, r_{n(R)}\}$ is a dictionary of relationships between nodes. Each relationship can be thought of as a filter that operates on a set of nodes V, producing a response $\Phi_i = r_i(V)$. These responses can be pooled over a number of parse graphs G to create histograms that can be used as node constraints. The type of relationship will likely differ based on whether it is defined for one, two, or more nodes. Fig. 2 visualizes a set of relationships that could be used to augment an And-Or graph.

Altogether these variables form a language for an object category, $L(G)$, that can produce a combinatorial number of constrained object configurations.

Fig. 3 shows a parse graph for the bike category. For notational convenience, we denote the following components of a parse graph **pg**:

- $T(\mathbf{pg}) = \{t_1, ..., t_{n(\mathbf{pg})}\}$ is the set of leaf nodes in **pg**. For example, $T(\mathbf{pg}) = \{1, 3, 5\}$ for the parse graph shown at the top of Fig. 1(c).
- $V^{\text{or}}(\mathbf{pg})$ is the set of non-empty Or nodes that are used in **pg**. For instance, the left-hand parse graph in Fig. 1(b) has $V^{\text{or}}(\mathbf{pg}) = \{A, B, H\}$.
- $E(\mathbf{pg})$ is the set of links in **pg**.

The probability for **pg** takes the following Gibbs form,

$$p(\mathbf{pg}; \Theta, \mathcal{R}, \Delta) = \frac{1}{Z(\Theta)} \exp\{-\mathcal{E}(\mathbf{pg})\} \tag{3}$$

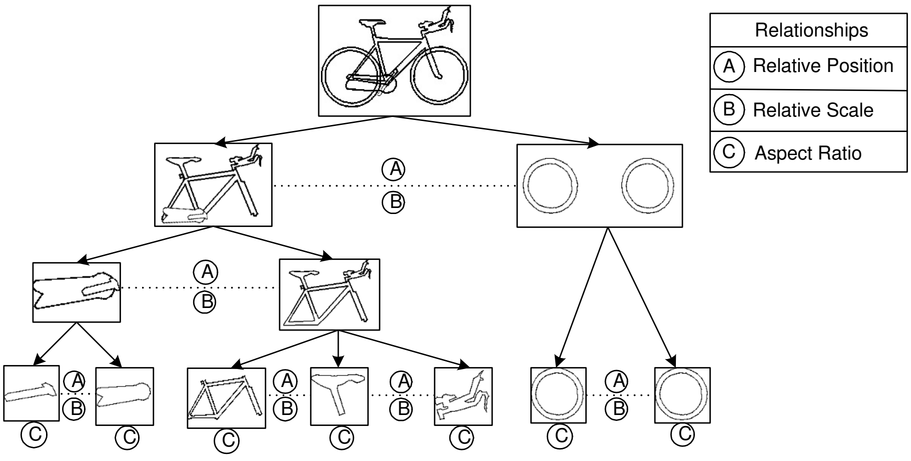


Figure 3: A parse graph of a bike. Nodes are composed via certain relations to form the representation of the bike.

where $\mathcal{E}(\mathbf{pg})$ is the total energy,

$$\mathcal{E}(\mathbf{pg}) = \sum_{v \in V^{\mathrm{or}}(\mathbf{pg})} \lambda_v(\omega(v)) + \sum_{t \in T(\mathbf{pg}) \cup V^{\mathrm{and}}(\mathbf{pg})} \lambda_t(\alpha(t)) + \sum_{(i,j) \in E(\mathbf{pg})} \lambda_{ij}(\alpha(v_i), \alpha(v_j)) \tag{4}$$

The model is specified by a number of parameters Θ, the relations set $\mathcal{R}$ and the vocabulary Δ. The first term in the energy is the same as that of a SCFG. It accounts for how frequently each Or node decomposes a certain way. The second and third terms are typical singleton and pair-clique energy for graphical models. The second term is defined on the geometric and appearance attributes $\alpha()$ of the image primitives, while the third term models the compatibility between the attributes of two related nodes.

This model can be derived using the maximum entropy principle under two types of constraints on the statistics of a training set. One constraint matches the frequency at each Or node, like a SCFG, and the other matches the relation statistics, such as the histograms modeling relative appearance or co-occurrence. Θ is the set of parameters in the energy,

$$\Theta = \{\lambda_v(), \lambda_t(), \lambda_{ij}();\ \forall v \in V^{\mathrm{or}}, \forall t \in V_T, \forall (i,j) \in \mathcal{R}\}. \tag{5}$$

Each $\lambda()$ above is a potential function, not a scalar, and is represented by a vector created by discretizing the function in a non-parametric way, as was done in the FRAME model for texture [33]. Δ is the vocabulary for the generative model. The partition function is summed over all parse graphs in the And-Or graph $G_{\mathrm{and-or}}$

$$Z = Z(\Theta) = \sum_{\mathbf{pg}} \exp\{-\mathcal{E}(\mathbf{pg})\}. \tag{6}$$

3 Learning and Estimation with the And-Or Graph

Suppose we have a set of observed parse graphs that follow f, the true distribution governing the objects.

$$D^{\text{obs}} = \{(\mathbf{I}_i^{\text{obs}}, \mathbf{pg}_i^{\text{obs}}) : \ i = 1, 2, ..., N\} \sim f(\mathbf{I}, \mathbf{pg}). \tag{7}$$

The parse graphs $\mathbf{pg}_i^{\text{obs}}$ are from a ground truth database or other labeled source. The objective is to learn a model p which approaches f by minimizing the Kullback-Leibler divergence $KL(f||p)$. This is equivalent to the ML estimate for the optimal vocabulary Δ, relation $\mathcal{R}$, and parameter Θ.

Learning the probability model includes two phases, both of which follow the same principle above.

1. Estimating the parameters Θ from training data D^{obs} for given $\mathcal{R}$ and Δ,
2. Learning and pursuing the relation set $\mathcal{R}$ for nodes in G given Δ.

3.1 Maximum Likelihood Learning of Θ

For a given And-Or graph hierarchy and relations, the estimation of Θ follows the MLE learning process. Let $\mathcal{L}(\Theta) = \sum_{i=1}^{N} \log p(\mathbf{I}_i^{\text{obs}}, \mathbf{pg}_i^{\text{obs}}; \Theta, \mathcal{R}, \Delta)$ be the log-likelihood. By setting $\frac{\partial \mathcal{L}(\Theta)}{\partial \Theta} = 0$, we have the following two learning steps.

1. Learning the λ_v at each Or node $v \in V^{\text{or}}$. The switch variable at v has $n(v)$ choices $\omega(v) \in \{\emptyset, 1, 2, ..., n(v)\}$ and is $\emptyset$ when v is not included in the $\mathbf{pg}$. We compute the histogram $\mathbf{h}_v^{\text{obs}}(\omega(v))$ in all the parse graphs in $\Omega_{\mathbf{pg}}^{\text{obs}}$. Thus,

$$\lambda_v(\omega(v) = i) = -\log \mathbf{h}_v^{\text{obs}}(\omega(v) = i), \ \ \forall v \in V^{\text{or}}. \tag{8}$$

This is simply the sample frequency for the Or node decompositions, as shown in [4].

2. Learning the potential functions $\lambda_t()$ at the terminal node $t \in V_T$ and $\lambda_{ij}()$ for each pair relation $(i, j) \in \mathcal{R}$. $\frac{\partial \mathcal{L}(\Theta)}{\partial \lambda_t} = 0$ and $\frac{\partial \mathcal{L}(\Theta)}{\partial \lambda_{ij}} = 0$ lead to the statistical constraints,

$$E_{p(\mathbf{pg};\Theta,\mathcal{R},\Delta)}[\mathbf{h}(\alpha(t)] = \mathbf{h}_t^{\text{obs}}, \ \forall t \in V_T \tag{9}$$

$$E_{p(\mathbf{pg};\Theta,\mathcal{R},\Delta)}[\mathbf{h}(\alpha(v_i), \alpha(v_j))]] = \mathbf{h}_{ij}^{\text{obs}}, \ \forall (i, j) \in \mathcal{R}. \tag{10}$$

In the above equation $\mathbf{h}()$ is a statistical measure of the attributes of the nodes in question, such as a histogram pooled over all the occurrences of those nodes in $\Omega_{\mathbf{pg}}^{\text{obs}}$. The λ parameters, when learned, will weight the $\mathbf{h}$ histograms so that $p(\mathbf{pg}; \Theta, \mathcal{R}, \Delta)$ is normalized correctly.

The equations (8), (9) and (10) are the constraints for deriving the Gibbs model $p(\mathbf{pg}; \Theta, \mathcal{R}, \Delta)$ in equation (3) through the maximum entropy principle.

Due to the coupling of the energy terms, equations (9) and (10) are solved iteratively through a gradient method. In a general case, we follow the stochastic gradient method adopted in learning the FRAME model [33], which approximates the expectations $E_p[\mathbf{h}(\alpha(t))]$ and $E_p[\mathbf{h}(\alpha(v_i), \alpha(v_j))]$ by sample means from a set of synthesized examples. This is the method of analysis-by-synthesis adopted in our texture modeling paper [33].

3.2 Learning and Pursuing the Relation Set

In addition to learning the parameters Θ, we can also augment the relation set $\mathcal{R}$ in an And-Or Graph, thus pursuing the energy terms in $\sum_{(i,j)\in E(\mathbf{pg})} \lambda_{ij}(\alpha(v_i), \alpha(v_j))$ in the same way as filters and statistics were pursued in texture modeling by the minimax entropy principle [33].

Suppose we start with an empty relation set $\mathcal{R} = \emptyset$, creating a parse graph with just its SCFG componenent defined, $p = p(\mathbf{pg}; \lambda, \emptyset, \Delta)$. We define a greedy pursuit where, at each step, we add a relation e_+ to $\mathcal{R}$ and thus augment model $p(\mathbf{pg}; \Theta, \mathcal{R}, \Delta)$ to $p_+(\mathbf{pg}; \Theta, \mathcal{R}_+, \Delta)$, where $\mathcal{R}_+ = \mathcal{R} \cup \{e_+\}$.

e_+ is selected from a large pool $\Delta_{\mathcal{R}}$ so as to maximally reduce KL-divengence,

$$e_+ = \arg\max KL(f||p) - KL(f||p_+) \ = \arg\max KL(p_+||p), \tag{11}$$

Thus we denote the information gain of e_+ by

$$\delta(e_+) \overset{def}{=} KL(p_+||p) \approx f^{\mathrm{obs}}(e_+) d_{\mathrm{manh}}(\mathbf{h}^{\mathrm{obs}}(e_+), \mathbf{h}_p^{\mathrm{syn}}(e_+)). \tag{12}$$

In the above formula, $f^{\mathrm{obs}}(e_+)$ is the frequency that relation e_+ is observed in the training data, $\mathbf{h}^{\mathrm{obs}}(e_+)$ is the histogram for relation e_+ over training data D^{obs}, and $\mathbf{h}_p^{\mathrm{syn}}(e_+)$ is the histogram for relation e_+ over the synthesized parse graphs according to the current model p. $d_{\mathrm{manh}}()$ is the Mahalanobis distance between the two histograms.

Intuitively, $\delta(e_+)$ is large if e_+ occurs frequently and creates a large difference between the histograms of the observed and the synthesized parse graphs. Large information gain means e_+ is a significant relationship.

4 Experiments on Learning and Sampling

We tested our learned model on 24 categories of objects, shown with representative samples in Figure 5. Categories were selected from the Lotus Hill Database [29], and between 40 and 60 training instances were collected for each category. The training data for each image consisted of the labeled boundaries of a pre-determined set of object parts. The object parts were selected by hand for each category (e.g. a teapot consists of a spout, base, handle, and lid) and outlined by hand. Each part category consists of about 5 - 10 different types (e.g. round handles vs. square handles), which are also selected by hand. Our dictionary of relationships Δ_R consists of the following relations: (aspect ratio, relative position, relative scale, relative orientation, overlap). Aspect ratio is the one singleton relationship, while all other relationships are measured over pairs of parts. When initializing the learning algorithm, we measure each relationship across every possible pair of parts. For an object category consisting of k parts and a dictionary of m singleton relationships and n pairwise relations, the algorithm begins with a pool of $m{*}k{+}n{*}(\frac{k(k-1)}{2})$ potential relationships.

Learning By Random Synthesis Figure 4 shows samples drawn from the model at each stage of the relationship pursuit in 3.2. At each stage, a new relationship was added and the parameters were estimated using the process in 3.1. Figure 4 shows this process for the clock and bicycle categories. The initial samples are wild and unconstrained, while the objects appear more coherent as relationships are added. Figure 5 shows samples from the learned model for 24 different object categories. We can see that the samples are perceptually equivalent to their category examples, even if slightly different on a part-by-part basis.

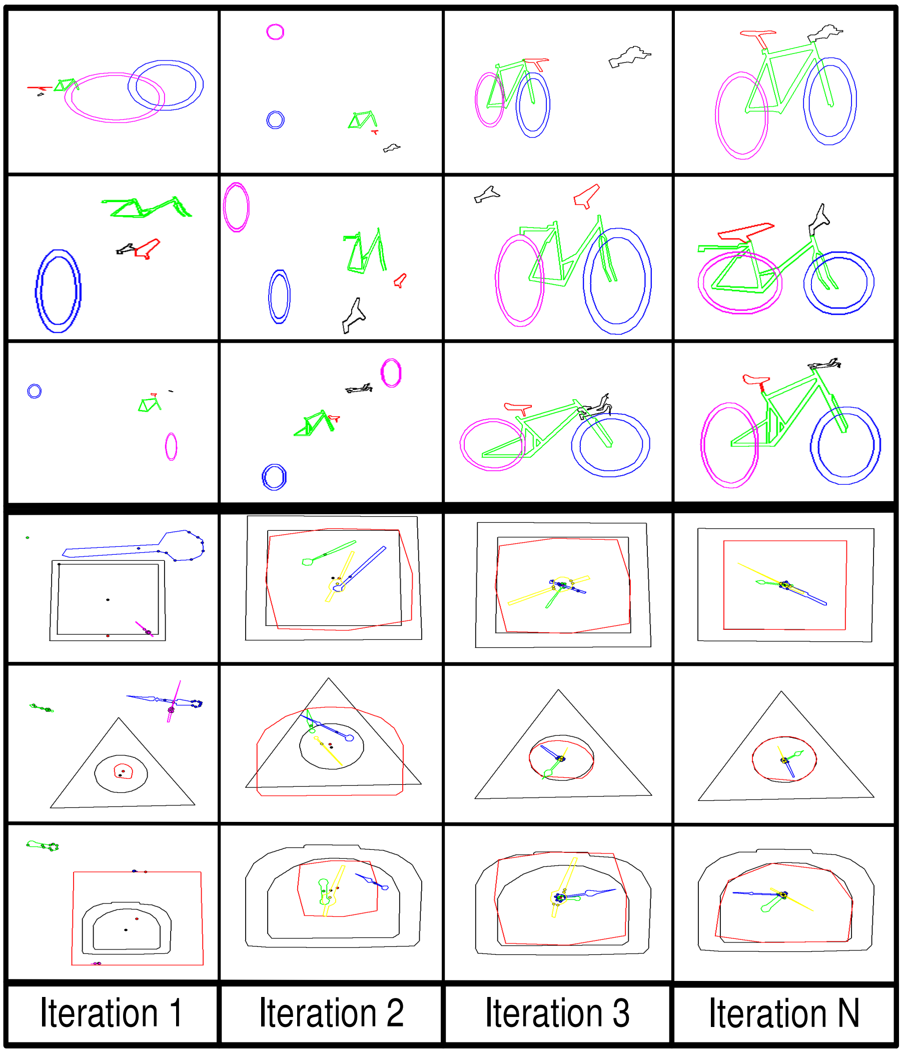


Figure 4: Samples from p during each stage of the relationship pursuit. Objects become more coherent as new relationships are added.

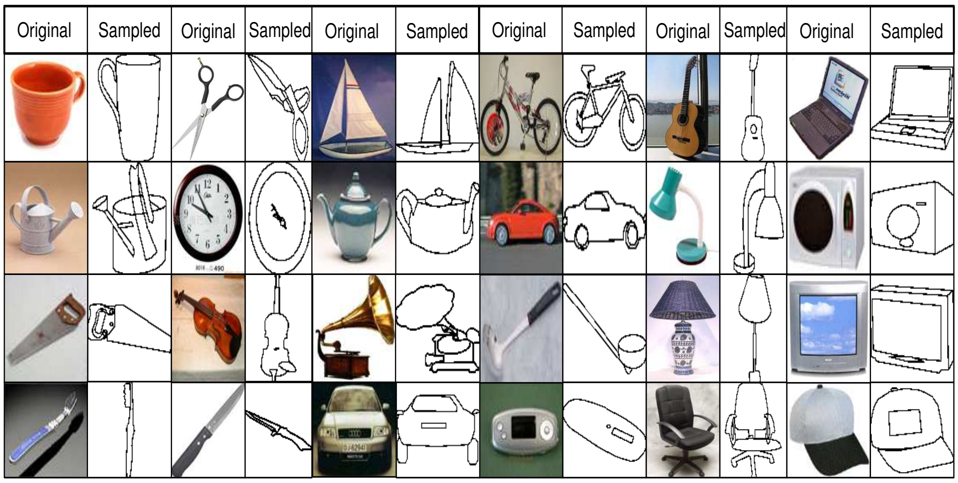


Figure 5: Twenty-four object categories with high intra-class variability and their corresponding samples.

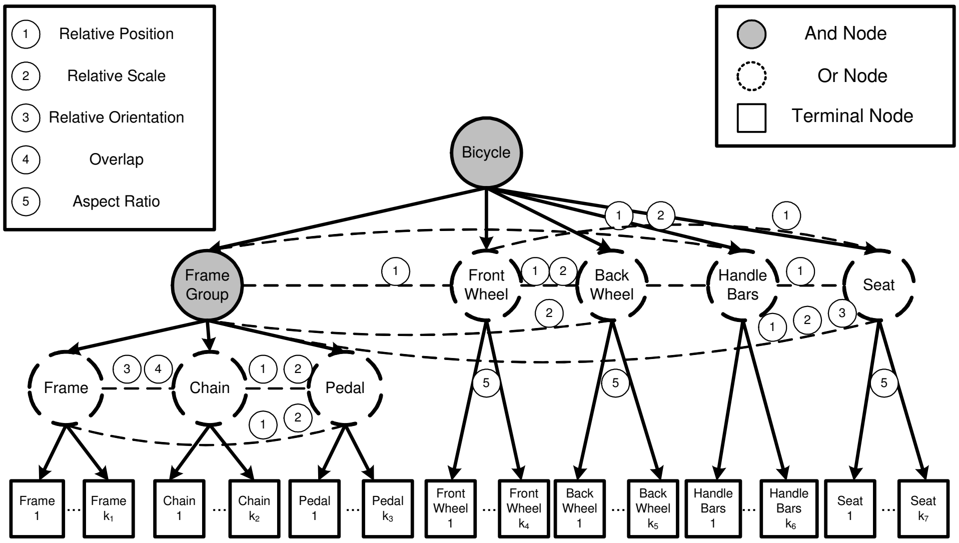


Figure 6: The learned And-Or graph for the bicycle category, showing the relationships added between pairs of nodes.

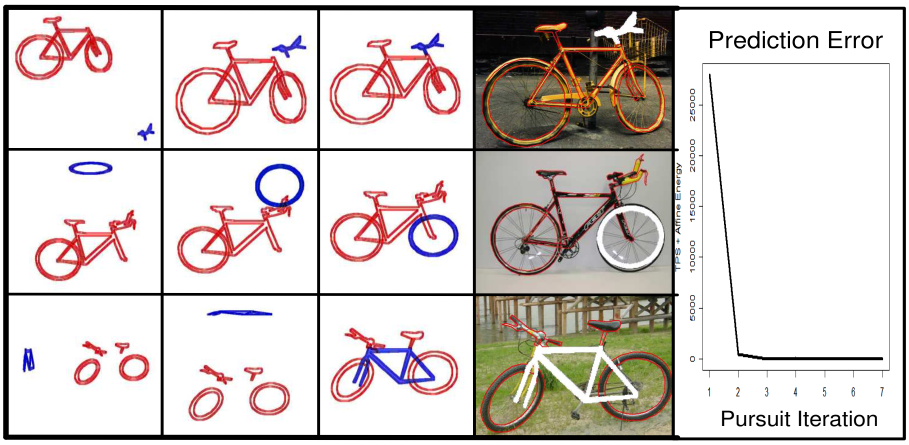


Figure 7: Top-down prediction of missing parts at each stage of the relationship pursuit. A neighborhood of parts is fixed and the remaining parts are Gibbs sampled. The accuracy of the prediction is measured by the Thin-plate-spline + affine transformation needed to move the predicted part to its true position. We can see that this energy decreases drastically as we add more relations to the model.

Figure 6 shows the And-Or graph that was learned for the bicycle category and the relationships that were added to pairs and single parts. We can almost see causal chains appearing in the relationships learned (the frame constrains the chain's position and scale, which in turn constrains the pedal's position and scale). Each part decomposes into one of a number of terminals. These terminals could be represented by an appearance model for each part, though in our case we used exemplars from the initial dataset, scaled, oriented, and positioned to create new object configurations.

Predicting Missing Parts Using Learned Model The sampling process can be used to provide top-down proposals for inference. Figure 7 shows the results of removing the wheel, frame, and handle bars from a perfect sketch of a bike and predicting the true part positions at each stage of the learning process. The missing parts of the structure are first reintroduced, and then their constraints are sampled for 100 iterations. The model with few constraints does not yield good results, while the full model predicts the part locations perfectly, as shown overlaid in the original image. The error is measured as the sum of the thin-plate-spline deformation energy [2] and affine energy needed to move the predicted part to its true location, and is shown in the last column. This shows that the compositional model provides strong top-down cues for part positions and appearances, and can predict the presence of parts that are occluded, missing, or had weak bottom-up cues for recognition and detection tasks.

Small Sample Set Generalization Due to the consistency of many of the perceptual similarities between objects in the same class, e.g. relative position, we can learn our model from a very small sample set. Fig. 8 shows samples drawn from the model learned from just 6 training instances. Despite there being so few training instances, their parts can be reconfigured and adjusted according to the model to produce radically different instances. Note that the part configurations and appearances in the samples differ greatly from those in the training set, yet the objects are still coherent. This is useful for recognition tasks, where new instances can be recognized despite not appearing in the training

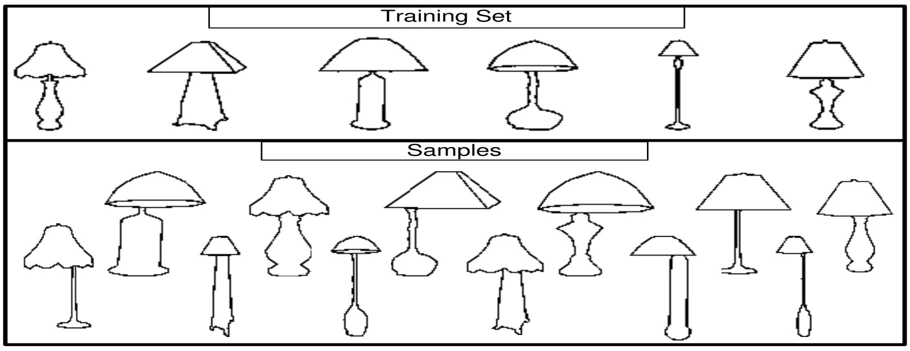


Figure 8: Demonstration of the model's generalizability. The model learned from only 6 training instances can produce the varied samples below.

data. One can also generate large amounts of training data for discriminative tasks using this model, learned from a small, easily obtained set of images. Such an experiment was done comparing recognition results using two different datasets. The first was fully hand-collected, while the second consisted of hand-collected data and samples generated from our model. The latter classifier showed a 15% improvement over solely hand-collected data, likely because there were more varied data samples available in the second dataset [16]. Further work is being done on this topic.

5 Inference with the And-Or-Graph

This section contains a brief introduction to our compositional inference algorithm for recognizing objects by parts. We refer readers to [13, 32, 17] for a more detailed explanation.

Given an input image $\mathbf{I}$, we would like to compute a parse graph $\mathbf{pg}$ that maximizes the posterior probability

$$\mathbf{pg}^* = \arg\max_{\mathbf{pg}} p(\mathbf{I}|\mathbf{pg}; \Delta_{\mathrm{sk}}) p(\mathbf{pg}; \Theta, \Delta). \tag{13}$$

The likelihood model is based on how well the terminal nodes match the image, and the prior is defined by the grammar model in equation 3. In our implementation, our likelihood model follows that of the primal sketch [12].

Because the And-Or graph can be defined recursively, so too can the inference algorithm, which largely simplifies the algorithm design and makes it easily scalable to arbitrarily large number of object categories. This algorithm is known as a *compositional boosting* algorithm [28].

Consider an arbitrary And node A. Let us assume that A can be decomposed into $n(A) = 3$ parts, or can just terminate at a low-resolution version of itself.

$$A \to A_1 \cdot A_2 \cdot A_3 \,|\, t_1 \,|\, \cdots \,|\, t_n. \tag{14}$$

This recursive unit is shown in Fig. 9.

This representation borrows from some common concepts in artificial intelligence [21]. An **Open List** stores a number of weighted particles (or hypotheses) generated by bottom-up detectors for A. A **Closed List** stores a number of instances for A which have been

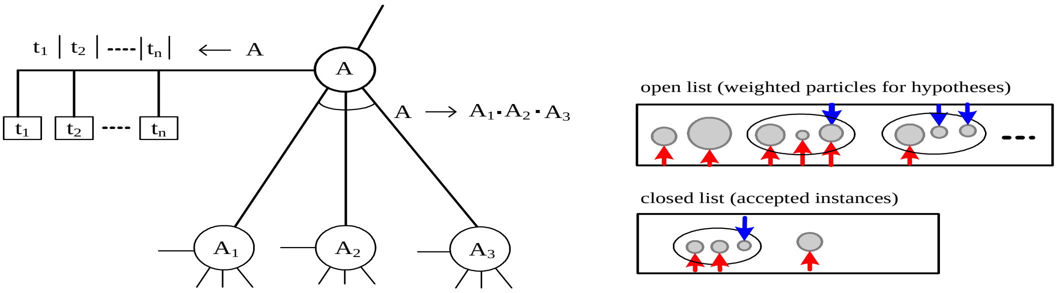

t₁ | t₂ | ----| tₙ | ← A
A
t₁
t₂
----
tₙ
A → A₁·A₂·A₃
A₁
A₂
A₃
open list (weighted particles for hypotheses)
closed list (accepted instances)

Figure 9: Data structure for the recursive inference algorithm on the And-Or graph.

accepted in the top-down process. These instances are nodes in the current parse graph **pg**.

The bottom-up process creates the particles in the Open lists using two methods.

(i) Generating hypotheses for A directly from images using bottom-up processes, such as Adaboosting[9, 26] or Hough transforms, to detect various terminals $t_1, ..., t_n$. The weight of a detected hypothesis is the logarithm of some local marginal posterior probability ratio given a small image patch Λ^i,

$$\omega_A^i = \log \frac{p(A^i|\mathbf{I}_{\Lambda^i})}{p(\bar{A}^i|\mathbf{I}_{\Lambda^i})} \approx \log \frac{p(A^i|F(\mathbf{I}_{\Lambda^i}))}{p(\bar{A}^i|F(\mathbf{I}_{\Lambda^i}))} = \hat{\omega}_A^i. \tag{15}$$

$\bar{A}$ represents a competing hypothesis. For computational effectiveness, the posterior probability ratio is approximated using local features $F(\mathbf{I}_{\Lambda^i})$ rather than the image $\mathbf{I}_{\Lambda^i}$.

(ii) Generating hypotheses for A by binding k of A's children from the existing Open and Closed lists. The binding process tests the relationships between these child nodes for compatibility and quickly rules out obviously incompatible compositions. The weight of a bound hypothesis is the logarithm of some local conditional posterior probability ratio. Suppose a particle A^i is bound from two existing parts A_1^i and A_2^i with A_3^i missing, and Λ^i is the domain containing the hypothesized A. Then the weight will be

$$\begin{aligned} \omega_A^i = & \log \frac{p(A^i|A_1^i, A_2^i, \mathbf{I}_{\Lambda^i})}{p(\bar{A}^i|A_1^i, A_2^i, \mathbf{I}_{\Lambda^i})} = \log \frac{p(A_1^i, A_2^i, \mathbf{I}_{\Lambda^i}|A^i)p(A^i)}{p(A_1^i, A_2^i, \mathbf{I}_{\Lambda^i}|\bar{A}^i)p(\bar{A}^i)} \\ \approx & \log \frac{p(A_1^i, A_2^i|A^i)p(A^i)}{p(A_1^i, A_2^i|\bar{A}^i)p(\bar{A}^i)} = \hat{\omega}_A^i. \end{aligned} \tag{16}$$

where $\bar{A}$ represents a competing hypothesis.

The top-down process validates the bottom-up hypotheses in all the Open lists following the Bayesian posterior probability. During this process it needs to maintain the weights of the Open lists.

(i) Given a hypothesis A^i with weight $\hat{\omega}_A^i$, the top-down process validates it by computing the true posterior probability ratio ω_A^i stated above. If A^i is accepted into the Closed list of A then the current parse graph **pg** moves to a new parse graph $\mathbf{pg}_+$. In a reverse process, the top-down process may also select a node A in the Closed list and either delete it (putting it back into the Open list) or disassemble it into independent parts.

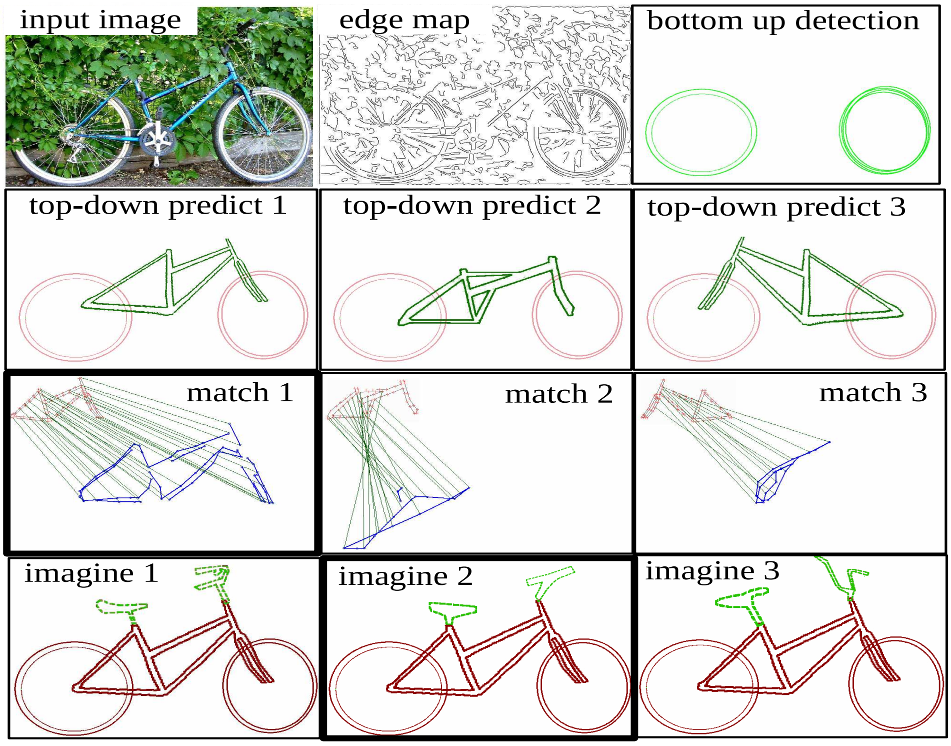


Figure 10: The combination of bottom-up and top-down influences for detecting an occluded bicycle (presented in [17]).

(ii) Given two competing hypotheses A and A' which overlap in a domain Λ_o, accepting one hypothesis will lower the weight of the other. Therefore, whenever we add or delete a node A in the parse graph, all the other hypotheses whose domains overlap with that of A will have to update their weights.

The acceptance of a node can be performed using a greedy algorithm that maximizes the posterior probability. At each iteration, the particle whose weight is the largest among all Open lists is accepted. This continues until the largest weight is below a certain threshold.

6 Experiments on Object Recognition Using the And-Or Graph

We apply our inference algorithm to five object categories – clock, bike, computer, cup/bowl, and teapot. Fig. 10 shows an example of inferring a partially occluded bicycle.

In Fig. 10. The first row shows the input image, an edge map, and bottom-up detection of the two wheels using a Hough transform. The second row shows some top-down predictions of bike frames based on the two wheels, sampled from the learned MRF model.

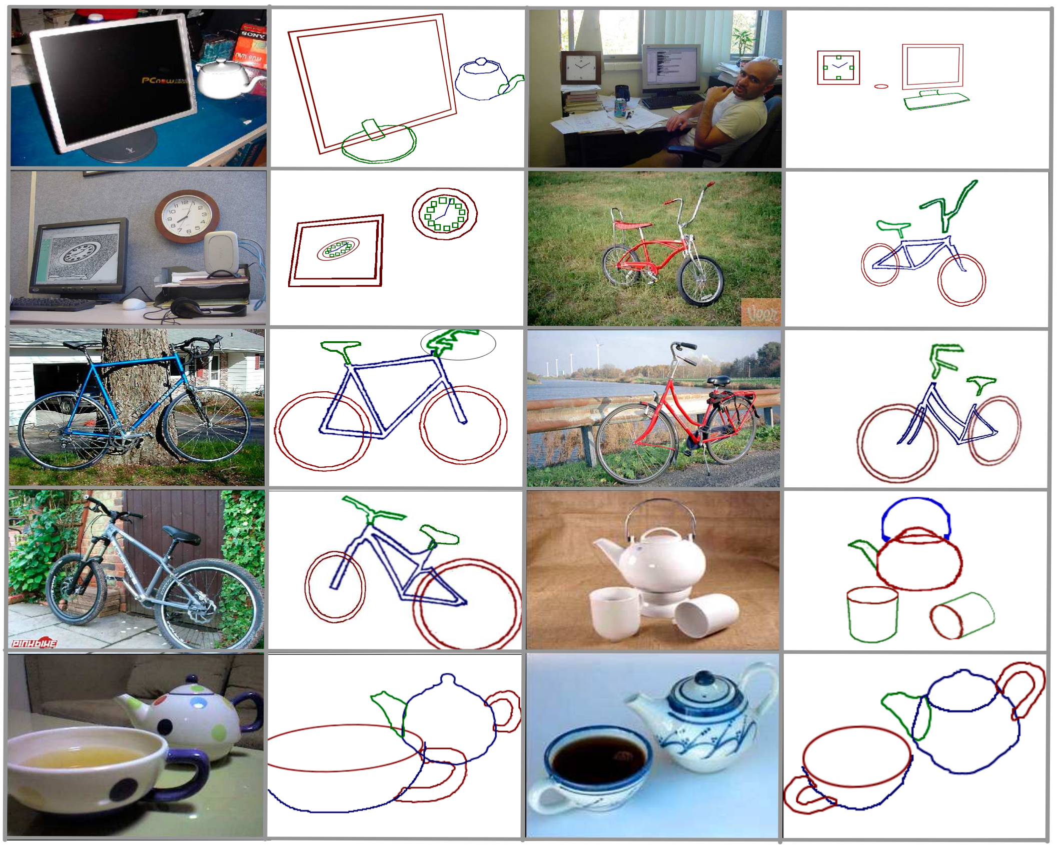
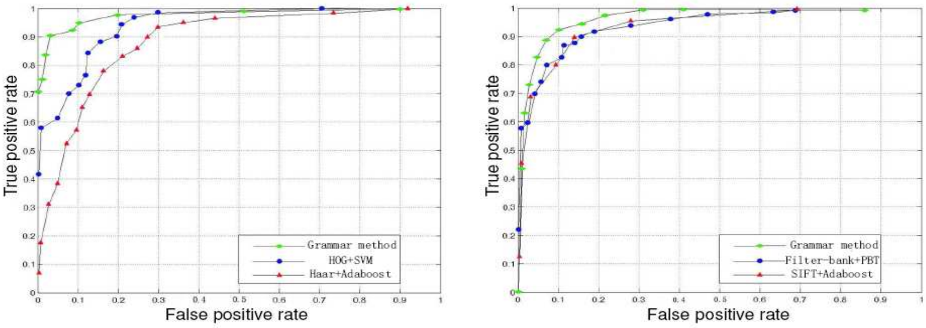


Figure 11: Recognition experiments on 5 object categories (presented in [17]).

The third row shows the template matching process that matches the predicted frames (in red) to the edges (in blue) in the image. The frame with minimum matching cost is selected. The fourth row shows the top-down hallucinations for the seat and handlebar (in green), which are randomly sampled from the And-Or graph model.

Fig. 11 shows recognition results for the five categories. For each input image, the image on the right shows the recognized parts from the image in different colors. It should be mentioned that the recognition algorithm is distinct from most of the classification algorithms in the literature. It interprets the image as a parse graph, which includes the classification of categories and parts, matches the leaf templates to images, and hallucinates occluded parts.

Some recent work was done to show the accuracy of our method versus common recognition and classification techniques in [17]. This work uses our methodology for learning but includes additional experimental results for the inference algorithm. In this work, our model was trained on the categories of rear car and bicycle and these models were used to recognize objects in the Lotus Hill Database [29] as well as the Caltech 101 dataset [7]. Figure 12 shows the results of this experiment. Our model was compared



(a) Bicycle category

(b) Rear-car category

Figure 12: ROC curves for recognizing bicycles (from the LHI dataset) and rear-cars (from the Caltech 101 dataset). Our model outperforms a HOG-based SVM and Adaboost on bicycles and PBT and SIFT-based boosting for rear-cars (presented in [17]).

against a HOG-based SVM [5] and a Haar-feature-based Adaboost model [26]. One can see that our model performs much better than either of these two methods. For the Caltech 101 dataset, we trained our model and ran it against a Probabilistic Boosting Tree [24] as well as a SIFT-based boosting method [31], both of which performed worse than our grammar-based approach.

7 Summary

In this chapter we have discussed a minimax learning algorithm for a compositional model of object categories. The grammar structure of this model accounts for the variability of part configurations, while the relational cosntraints at each level capture the variability of part appearances. Samples drawn from the model are visually similar to the training data, yet novel instances can be created from even very small training sets. With the use of compositional boosting, objects can be reliably recognized and parsed in new images as well. Our algorithms are similar to theories of recognition in the human-vision system and yield encouraging results for the implementation of part-based recognition theories for biological vision. These are promising results and we plan to continue studying using grammars for representing visual knowledge.

Acknowledgments

This work was funded by NSF grant IIS 0713652. Projects done at the Lotus Hill Research Institute were supported by an 863 program No. 2006AA012121.

References

[1] I. Biederman, "Recognition-by-Components: A theory of Human Image Understanding", *Psychological Review*, vol. 94: 115-147, 1987.

[2] F. L. Bookstein, “Principal Warps: Thin-Plate Splines and the Decomposition of Deformations”, *Pattern Analysis and Machine Intelligence*, vol. 11: 567-585, 1989.

[3] H. Chen, Z. Xu, Z. Liu, S.C. Zhu, “Composite Templates for Cloth Modeling and Sketching”, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1: 943-950, 2006.

[4] Z. Chi, S. Geman, “Estimation of Probabilistic Context-Free Grammars”, *Computational Linguistics*, vol. 24, n.2, June 1998.

[5] N. Dalal, B. Triggs, C. Schmid. Human Detection Using Oriented Histograms of Flow and Appearance, *Proc. of European Conference on Computer Vision*, vol. 2: 428-441, 2006.

[6] S. Dickinson, A. Pentland, A. Rosenfeld, “From volume to views: an approach to 3D object recognition”, *CVGIP: Image Understanding*, 55(2): 130-154, 1992.

[7] L. Fei-Fei, P. Perona. “A Bayesian Hierarchical Model for Learning Natural Scene Categories”, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2: 524-531, 2005.

[8] P. Felzenszwalb, D. Huttenlocher, ”Pictorial Structures for Object Recognition”, *International Journal of Computer Vision*, 61(1): 55-79, 2005.

[9] J. Friedman, T. Hastie, R. Tibshirani, “Additive logistic regression: a statistical view of boosting”, *Annals of Statistics*, 38(2): 337-374, 2000.

[10] M. Fischler, R. Elschlager, “The representation and matching of pictorial structures”, *IEEE Transactions on Computers*, 22(1): 67-92, 1973.

[11] K.S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice Hall, 1981.

[12] C.E. Guo, S.C. Zhu, Y.N. Wu, “Primal sketch: integrating texture and structure”, *Proc. Int’l Conf. on Computer Vision*, vol. 106: 5-19, 2003.

[13] F. Han, S.C. Zhu, “Bottom-up/top-down image parsing by attribute graph grammar”, *Proc. of Int’l Conf. on Computer Vision*, vol. 2, 2005.

[14] Y. Jin, S. Geman, “Context and hierarchy in a probabilistic image model”, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2: 2145-2152, 2006.

[15] Y. Keselman, S. Dickinson, “Generic model abstraction from examples”, *Pattern Analysis and Machine Intelligence*, vol. 27: 1141-1156, 2001.

[16] L. Lin, S. Peng, J. Porway, S.C. Zhu, Y. Wang, “An Empirical Study of Object Category Recognition: Sequential Testing with Generalized Samples”, *Proc. of Int’l Conf. on Computer Vision*, pp. 1-8, 2007.

[17] L. Lin, T. Wu, J. Porway, Z. Xu, “A Stochastic Graph Grammar For Compositional Object Representation and Recognition”, *Under review for Pattern Recognition, 2009.*

[18] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”,*International Journal of Computer Vision*, 60(2), pp. 91-110, 2004.

[19] S. K. Nayar, H. Murase, S. A. Nene, “Parametric Appearance Representation”, in S. K. Nayar and T. Poggio, (eds) *Early Visual Learning*, 1996.

[20] Y. Ohta, *Knowledge-based interpretation of outdoor natural color scenes*, Pitman, 1985.

[21] J. Pearl, *Heuristics: intelligent search strategies for computer problem solving*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1984.

[22] K. Siddigi, A. Shokoufandeh, S.J. Dickinson, S.W. Zucker, “Shock graphs and shape matching”, *International Journal of Computer Vision*, 35(1), 13-32, 199.

[23] S. Todorovic, N. Ahuja, “Extracting subimages of an unknown category from a set of images”, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, 927 - 934, 2006.

[24] Z. Tu, “Probabilistic Boosting Tree: Learning Discriminative Models for Classification, Recognition, and Clustering”, *Proc. of Int’l Conf. on Computer Vision*, vol. 2: 1589-1596, 2005.

[25] S. Ullman, E. Sali, M. Vidal-Naquet, “A Fragment-Based Approach to Object Representation and Classification”. *Proc. 4th Int’l Wkshp. on Visual Form*, Capri, Italy, 2001.

[26] P. Viola, M. Jones, “Rapid object detection using a boosted cascade of simple features,”, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 511-518, 2001.

[27] M. Weber, M. Welling, P. Perona, “Towards automatic discovery of object categories”, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2: 101-108, 2000.

[28] T.F. Wu, G.S. Xia, S.C. Zhu, “Compositional Boosting for Computing Hierarchical Image Structures”, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1-8, June, 2007.

[29] B. Yao, X. Yang, S.C. Zhu, “Introduction to a large scale general purpose groundtruth dataset: methodology, annotation tool, and benchmarks,” *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer LNCS Vol. 4697: 169-183, 2007.

[30] F.C You, K.S. Fu, “Attributed Grammar: A Tool for Combining Syntatic and Statistical Approaches to Pattern Recognition”, *IEEE Trans. on SMC*, vol. 10, 1980.

[31] W. Zhang, B. Yu, G.J. Zelinsky, D. Samaras, “Object class recognition using multiple layer boosting with multiple features”, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2: 323-330, 2005.

[32] S.C. Zhu, D. Mumford, “A Stochastic Grammar of Images”, *Foundation and Trends in Computer Graphics and Vision*, 2:4: 259-362, 2006.

[33] S. C. Zhu, Y. N. Wu, D. Mumford, “Minimax entropy principle and its application to texture modeling”, *Neural Computation*, v.9 n.9, p.1627-1660, Nov. 15, 1997