# Using Language to Learn Structured Appearance Models for Image Annotation

Michael Jamieson, *Student Member, IEEE,* Afsaneh Fazly, Suzanne Stevenson, Sven Dickinson, *Member, IEEE,* Sven Wachsmuth, *Member, IEEE*

***Abstract*— Given an unstructured collection of captioned images of cluttered scenes featuring a variety of objects, our goal is to simultaneously learn the names and appearances of the objects. Only a small fraction of local features within any given image are associated with a particular caption word, and captions may contain irrelevant words not associated with any image object. We propose a novel algorithm that uses the repetition of feature neighborhoods across training images and a measure of correspondence with caption words to learn meaningful feature configurations (representing named objects). We also introduce a graph-based appearance model that captures some of the structure of an object by encoding the spatial relationships among the local visual features. In an iterative procedure we use language (the words) to drive a perceptual grouping process that assembles an appearance model for a named object. Results of applying our method to three data sets in a variety of conditions demonstrate that from complex, cluttered, real-world scenes with noisy captions, we can learn both the names and appearances of objects, resulting in a set of models invariant to translation, scale, orientation, occlusion, and minor changes in viewpoint or articulation. These named models, in turn, are used to automatically annotate new, uncaptioned images, thereby facilitating keyword-based image retrieval.**

***Index Terms*— Language–Vision integration, Image annotation, Perceptual grouping, Appearance models, Object recognition**

## I. Motivation

Manual annotation of new images in large image collections is prohibitively expensive for commercial databases, and overly time-consuming for the home photographer. However, low-cost imaging, storage and communication technologies have already made accessible millions of images that are meaningfully associated with text in the form of captions or keywords. It is tempting to see these pairings of visual and linguistic representations as a kind of distributed Rosetta Stone from which we may learn to automatically translate between the names of things and their appearances. Even limited success in this challenging project would support at least partial automatic annotation of new images, enabling search of image databases by both image features and keywords that describe their contents.

Any such endeavor faces the daunting challenge of the perceptual grouping problem. Regardless of the type of image feature used, a word typically refers not to a single feature, but to a configuration of features that form the object of interest. The problem is particularly acute since any given image may contain multiple objects or configurations; moreover, the meaningful configurations may be easily lost among a huge number of irrelevant or accidental groupings of features. Without substantial bottom-up grouping hints, it is a nearly hopeless task to glean the meaningful feature configurations from a single image–caption pair. Given a *collection* of images, however, one can look for patterns of features that appear much more often than expected by chance. Usually, though, only a fraction of these recurring configurations correspond to salient objects that are referred to by words in the captions. Our system searches for *meaningful* feature configurations that appear to correspond to a caption word. From these starting points it iteratively constructs flexible appearance models that maximize word–model correspondence.

Our approach is best-suited to learning the appearance of objects distinguished by their structure (*e.g.*, logos or landmarks) rather than their color and texture (*e.g.*, tigers or blue sky). By detecting the portions of an object with distinctive structure, we can find whether the object is present in an image and where (part of) the object appears, but we do not determine its full extent. Therefore our system is appropriate for annotation but only limited localization. Our specific focus is on learning correspondences between the names and appearances of exemplar objects from relatively noisy and complex training data rather than attempting to learn the more highly-variable appearance of object classes from less ambiguous training sets. However, our framework and the structure of our appearance model are designed to learn to recognize any objects that appear as multiple parts in a reasonably consistent configuration. We therefore believe that with the right choice of features, our framework could be adapted to learn the appearance of object classes such as cars, jets, or motorcycles.

### A. Background

The literature on automatic image annotation describes a large number of proposals [1], [3]–[8], [10], [12], [13], [17], [19], [21], [23]. Like our previous efforts ([15], [16]) and the work presented here, these systems are designed to learn meaningful correspondences between words and appearance models from cluttered images of multiple objects paired with noisy captions, *i.e.*, captions that contain irrelevant words.

Many of these approaches associate a caption word with a probability distribution over a feature space dominated by color and texture (though the set of features may include position [1], [6], or simple shape information [2], [12]). This type of representation is less reliant on perceptual grouping than a shape model or a structured appearance model because color and texture are relatively robust to segmentation errors and the configuration of features is not critical. This type of representation is a good fit for relatively structureless materials such as grass, sand or water. In addition, they can often be more effective than more rigid models for object classes such as animals (*e.g.*, Berg and Forsyth [4]) that *have* a consistent structure but are so variable

Manuscript received <month> dd, yyyy; revised <month> dd, yyyy.

M. Jamieson, A. Fazly, S. Stevenson, and S. Dickinson are with the Department of Computer Science at the University of Toronto; S. Wachsmuth is with the Faculty of Technology at Bielefeld University.

in articulation and pose that the structure is difficult to discern in 2D images. However, highly structured objects, such as buildings and bicycles, that may lack distinctive color or texture may still be recognized if individually ambiguous parts of appearance can be grouped together into a meaningful configuration.

A number of researchers have addressed the problems of perceptual grouping and learning of configurations in automatic annotation. For instance, Barnard *et al.* [2] acknowledge that coarse granularity features, such as regions, may be oversegmented during feature extraction and propose a ranking scheme for potential merges of regions based on a similarity of word–region associations. It is possible, however, for different parts of an object to be associated with very different words, and hence such an approach is problematic for grouping the object's parts in such cases. In a similar vein, Quattoni *et al.* [23] use the co-occurrence of caption words and visual features to merge together "synonymous" features. This lowers the dimensionality of their bag-of-features image representation and therefore allows image classifiers to be trained with fewer labeled examples. Such a system, which models visual structure as a mixture of features, can represent complex objects as the co-occurrence of distinctive parts within an image [6], [8], [13], [17], [21], [23]. However, these models contain no spatial relationships (even proximity) between parts that would allow them to represent true part configurations. Carbonetta *et al.* [7] use a Markov random field model that can successfully recognize a set of adjacent regions with widely varying appearance as being associated with a given word. Their method is not capable of learning structured configurations of features, however, since the spatial relationships between the object parts are not part of the learned region–word pairings. The multi-resolution statistical model proposed by Li and Wang [19] can represent configurations of visual features across multiple scales. However, the system does not perform grouping, as each semantic class is associated with a layout for the entire image, where the division into parts is predefined. Other work avoids the perceptual grouping problem by focusing on domains where there exists detailed prior knowledge of the appearance of the objects of interest, as in the task of matching names with faces [4]. Our work focuses on using correspondences between image features and caption words to guide the grouping of image features into explicit, meaningful configurations.

Methods for grouping individual features of various types into meaningful configurations are reasonably common in the broader object recognition literature. For instance, Fergus *et al.* [14] learn object appearance models consisting of a distinctive subset of local interest features and their relative positions, by looking for a subset of features and relationships that repeat across a collection of object images. Crandall and Huttenlocher [11] use graph models in which vertices are oriented edge templates rather than local feature detections, and edges represent spatial relationships. Sharing elements with both of the above approaches, our appearance models (carried over from [16]) are collections of distinctive local interest features tied together in a graph in which edges represent constrained spatial relationships. This type of representation is sufficiently flexible to handle occlusion, minor changes in scale and viewpoint, and common deformations. While both [14] and [11] can learn an appearance model from very noisy training images, the methods (like most object recognition systems) require training images in which a single object of the desired category occupies a large portion of the image. Our method makes these structured appearance models more applicable to automatic image annotation by relaxing that constraint.

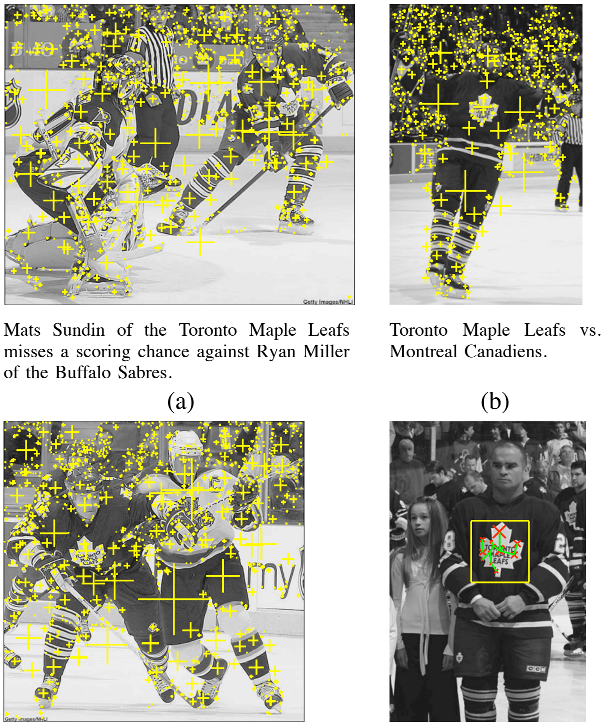
While many structured appearance models use features designed for object classes, our system uses features that are best suited to learning the appearance of exemplar objects (such as St. Paul's Cathedral) rather than a broad class of objects (such as cathedrals in general). The world is full of exemplars, and there has been a great deal of work in sorting and annotating exemplar images, such as the method proposed by Simon *et al.* [24] for organizing collections of related photographs into labeled canonical views. While our current detection method is not as scalable as high-performance exemplar image retrieval systems such as that proposed by Philbin *et al.* [22], our use of language can improve text-based querying and link together widely different appearances or views of a single exemplar.

The proposed learning algorithm here is an extension and a refinement of the algorithms presented in our previous work, [15], [16]. In [15] we represented appearance as an unstructured local collection of features and used a translation model to find correspondences between words and appearance. In [16] we added spatial relationships to the appearance model and introduced a more direct word–model correspondence measure. Here, we introduce a novel unified framework for evaluating both the goodness of a detection and the appropriateness of associating the detection with a caption word. We have also modified the improvement mechanism to learn more spatial relationships between features and present extensive new evaluation and analysis.

### *B. An Overview of Our Approach*

The goal of this work is to annotate exemplar objects appearing in images of cluttered scenes, such as the images shown in Figure 1(a). A typical such image, with hundreds (or even thousands) of local features, contains a huge number of possible feature configurations, most of which are noise or accidental groupings. A complex configuration of features that occurs in many images is unlikely to be an accident, but may still correspond to common elements of the background or other unnamed structures. The only evidence on which to establish a connection between words and configurations of visual features is their co-occurrence across the set of captioned images. The key insight is that this evidence can guide not only the annotation of complex feature configurations, but also the search for meaningful configurations themselves. Accordingly, we have developed a novel algorithm that uses language cues in addition to recurring visual patterns to incrementally learn strong object appearance models from a collection of noisy image–caption pairs (as in Figure 1(a-c)). The result of learning is a set of exemplar object appearance models paired with their names, which can be used for annotating similar objects in new (unseen and uncaptioned) images; a sample annotation is shown in Figure 1(b).

The overall structure of our proposed system is depicted in Figure 2. The system is composed of two main parts: a learning component, and an annotation component. Both components work with a particular representation of the images and objects, and use the same algorithm to detect instances of an object model in an image. Details of our image and object representation, as well as the detection algorithm, are presented in Section II. The two stages of the learning component, including methods for expanding and evaluating appearance models, are elaborated on in Section III. Finally, in Section IV, we present the annotation



Mats Sundin of the Toronto Maple Leafs misses a scoring chance against Ryan Miller of the Buffalo Sabres.

(a)

Toronto Maple Leafs vs. Montreal Canadiens.

(b)

Florida's Olli Jokinen gets bumped by Alexei Ponikarovsky of the Maple Leafs.

(c)

"Maple Leafs"

(c)

Fig. 1. (a-c) A sample input image–caption collection, where each image contains hundreds of local (SIFT) features (yellow crosses). From the input training collection, associations between structured subsets of local features and particular nouns are learned. (d) A sample output of our system, where the object (the Maple Leafs logo) is detected (shown with red features and green relationships in a yellow box), and annotated with its name ("Maple Leafs"). The annotation is performed using a word–appearance association discovered from the training image–caption collection.

component, demonstrating the promising performance of our system in discovering meaningful word–appearance pairs.

## II. Representing and Matching Objects

Our learning framework allows a one-to-many relationship between words and appearance models. It is thus not necessary that a single model capture object appearance from all possible viewpoints. Moreover, since we deal with exemplar objects, our method need not handle the changes in texture and structural detail that are possible within a class of objects. In order to serve as a robust object detector, however, it is important that the appearance model representation be invariant to reasonable changes in lighting, scale, orientation, articulation and deformation. The representation must also be reliably detectable, in order to avoid false annotations.

We represent our images using easily-extractable local interest features that can be used reliably to detect exemplar objects in highly cluttered scenes. We represent small patches of an instance of an object using such local features, and capture its structure using the pairwise spatial relationships between the patches. From the object instances, we then construct an abstract object appearance model in the form of a graph, by modeling the recurrent local features as vertices and the recurrent spatial relationships between pairs of local features as edges. The model that is built this way is a reliable descriptor of the object appearance and at the same time flexible enough to handle common deformations. Details of our choices for the representation of images and objects are given in Sections II-A and II-B, respectively. Given an object model (in the form of a graph), we need a method for detecting instances of the object in an image—that is, to find a matching between model vertices and local image features. Our detection algorithm is presented in detail in Section II-C.

### *A. Image Representation*

We represent an image as a set $I$ of local interest points $p_m$, *i.e.*, $I = \{p_m | m = 1 \ldots |I|\}$, referred to hereafter as points or image points. These points are detected using Lowe's SIFT method [20], which defines a point $p_m$ in terms of its Cartesian position $\mathbf{x}_m$, scale $\lambda_m$ and orientation $\theta_m$. In addition to spatial parameters, for each point we also extract a feature vector $\mathbf{f}_m$ that encodes a portion of the image surrounding the point. Since $\mathbf{f}_m$ is extracted relative to the spatial coordinates of $p_m$, it is invariant to changes in position, scale and orientation. While our approach is not dependent on a particular point detection method or feature encoding, we use the PCA-SIFT feature encoding developed by Ke and Sukthankar [18] because it allows for fast feature comparison and low memory requirements. This feature encoding is reasonably robust to lighting changes, minor deformations and changes in perspective. Since individual features capture small, independent patches of object appearance, the overall representation is robust to occlusion and articulation.

The continuous feature vector $\mathbf{f}_m$ is supplemented by a quantized descriptor $c_m$ for each image point, in order to support the ability to quickly scan for potentially matching features. Following Sivic and Zisserman [25], we use the K-means algorithm to generate a set of cluster centers, $C = \{\mathbf{f}_c | c = 1 \ldots |C|\}$, from a set of features randomly selected from a stock image collection. The index of the cluster center closest to $\mathbf{f}_m$ is used as the descriptor $c_m$ associated with $p_m$.

In addition to describing each point individually, we also attempt to capture the local spatial configuration of points using neighborhoods that describe the local context of a point. Each point $p_m$ is associated with a neighborhood $\mathbf{n}_m$ that is the set of its spatially closest neighbors $p_n$, according to the $\Delta x_{mn}$ distance measure taken from Carneiro and Jepson [9]:

$$\Delta x_{mn} = \frac{\|\mathbf{x}_m - \mathbf{x}_n\|}{\min(\lambda_m, \lambda_n)} \quad (1)$$

This normalized distance measure makes neighborhoods more robust to changes in scale, as newly-introduced fine-scale points are less likely to push coarse-scale points out of the neighborhood when the scale of an object increases.

To summarize, each image is represented as a set of points, $I = \{p_m | m = 1 \ldots |I|\}$, in which $p_m$ is a 6-tuple of the form $(\mathbf{f}_m, \mathbf{x}_m, \lambda_m, \theta_m, c_m, \mathbf{n}_m)$. In addition, a vector of transformation-invariant spatial relationships $r_{mn}$ is defined between each pair of neighboring points, $p_m$ and $p_n$, including the relative distance between the two points ($\Delta x_{mn}$), the relative scale difference between them ($\Delta\lambda_{mn}$), the relative heading from $p_m$ to $p_n$ ($\Delta\phi_{mn}$), and the relative heading in the opposite direction ($\Delta\phi_{nm}$). That is, $r_{mn} = (\Delta x_{mn}, \Delta\lambda_{mn}, \Delta\phi_{mn}, \Delta\phi_{nm})$, where the spatial relationships are taken from Carneiro and Jepson [9], and are calculated as in Equations (1) above, and (2) and (3)

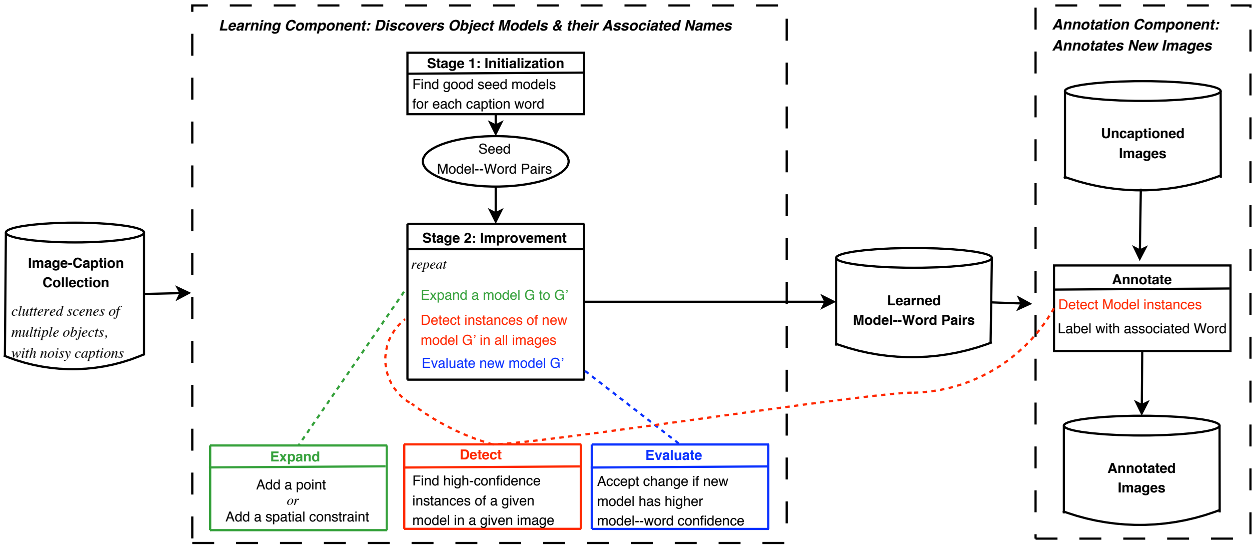

Learning Component: Discovers Object Models & their Associated Names
Stage 1: Initialization
Find good seed models for each caption word
Seed Model--Word Pairs
Stage 2: Improvement
repeat
Expand a model G to G'
Detect instances of new model G' in all images
Evaluate new model G'
Image-Caption Collection
cluttered scenes of multiple objects, with noisy captions
Expand
Add a point
or
Add a spatial constraint
Detect
Find high-confidence instances of a given model in a given image
Evaluate
Accept change if new model has higher model--word confidence
Learned Model--Word Pairs
Annotation Component: Annotates New Images
Uncaptioned Images
Annotate
Detect Model instances
Label with associated Word
Annotated Images

Fig. 2. A pictorial representation of our system for learning and annotating objects.

below:

$$\Delta\lambda_{mn} = \frac{\lambda_m - \lambda_n}{\min(\lambda_m, \lambda_n)} \tag{2}$$

$$\Delta\phi_{mn} = \Delta_\theta(\tan^{-1}(\mathbf{x}_m - \mathbf{x}_n) - \theta_m) \tag{3}$$

where $\Delta_\theta(.) \in [-\pi, +\pi]$ denotes the principle angle. The spatial relationships are not part of the stored image representation, but are calculated on demand when object appearance models are being built (see Section II-B) or detected (see Section II-C).

## B. Object Appearance Model

An object model describes the distinctive appearance of an object as a particular set of local features that have a more-or-less structured arrangement. We represent this structured configuration of features as a graph $G = (V, E)$. Each vertex $v_i \in V$ is composed of a continuous feature vector $\mathbf{f}_i$, and a cluster index vector $\mathbf{c}_i$ containing indices for the $|\mathbf{c}_i|$ nearest cluster centers to $\mathbf{f}_i$, *i.e.*, $v_i = (\mathbf{f}_i, \mathbf{c}_i)$. Associating each model vertex with a set of clusters allows for fast comparison of features during model detection while minimizing the effects of quantization noise. Note that model vertices, unlike image points, do not include spatial information (*i.e.*, position, orientation, and scale), because the model must be invariant to translation, rotation, and scale. Each edge $e_{ij} \in E$ encodes the expected spatial relationship between two vertices $v_i$ and $v_j$, in four parts: $e_{ij} = (\Delta x_{ij}, \Delta\lambda_{ij}, \Delta\phi_{ij}, \Delta\phi_{ij})$ (as defined in Equations (1)–(3) above).

We assume objects are spatially coherent, and hence two nearby points on an object are expected to be related non-accidentally, *i.e.*, through a geometric relation. Thus in our framework only a connected graph is considered to be a valid object appearance model; edges are allowed between any pair of vertices (and thus models are not restricted to trees). Models that can encode all the non-accidental relationships between pairs of nearby image points are generally more distinctive, and more robust to occlusion and inconsistent point detection, than models that are restricted to trees.

The parameters of a model's vertices ($\mathbf{f}_i$ and $\mathbf{c}_i$) as well as those of the edges ($\Delta x_{ij}$, $\Delta\lambda_{ij}$, $\Delta\phi_{ij}$, and $\Delta\phi_{ij}$) are calculated from the corresponding parameters of the images (points and the spatial relationships between them) through an iterative process of model construction and detection (see Sections II-C and III).

## C. Detecting Instances of an Object

We use a heuristic algorithm that searches for high-confidence instances of an appearance model in an image. Though each model is intended to robustly describe an object's appearance, no observed instance of an object is expected to fit its model exactly. Deformation, noise, and changes in perspective can distort the features encoded at points and/or the spatial relationships between them. Our detection algorithm should thus be capable of finding partial matches of a model (representing some visible part of an object). At the same time, the algorithm should distinguish a partial match which has only a few observed vertices from an accidental collection of background elements. Our algorithm thus assigns, to each observed instance of a model, a confidence score that meets the above requirements, thereby determining how well the instance fits the model. Below we first explain our proposed detection confidence score, and then present the details of our detection algorithm.

*1) Detection Confidence:* An observed instance $O$ of an object appearance model $G$ is a set of vertex–point associations, $O = \{(v_i, p_m) | v_i \in V, p_m \in I\}$, where $O$ defines a one-to-one mapping between a subset of the model vertices and some subset of points in an image. Our detection confidence score defines the goodness of fit between a model $G$ and an observation $O$ as the likelihood of $O$ being a true instance of $G$ and not a chance configuration of background features. Let $H_G$ be the hypothesis that $O$ is a true instance of the appearance model $G$, while $H_B$ is the competing hypothesis that $O$ is a chance assortment of background features. The *detection confidence* is:

$$\begin{aligned} \text{Conf}_{detect}(O, G) &= P(H_G|O) \\ &= \frac{p(O|H_G)P(H_G)}{p(O|H_G)P(H_G) + p(O|H_B)P(H_B)} \end{aligned} \tag{4}$$

where $p(O|H_B)$ is the background likelihood of $O$, and $p(O|H_G)$ is the model likelihood of $O$. We use $P(.)$ to indicate probability functions and $p(.)$ to indicate probability density functions.

The above equation can be rewritten as:

$$\mathrm{Conf}_{detect}(O,G) = \frac{\frac{p(O|H_G)\,P(H_G)}{p(O|H_B)\,P(H_B)}}{\frac{p(O|H_G)\,P(H_G)}{p(O|H_B)\,P(H_B)}+1} \quad (5)$$

Thus the detection confidence can be calculated from the prior likelihood ratio, $P(H_G)/P(H_B)$, and the observation likelihood ratio, $p(O|H_G)/p(O|H_B)$. The prior likelihood ratio is set to a fixed value, empirically determined from experiments on a held-out subset of the training data (see the Appendix). Next, we explain how we estimate the observation likelihood ratio from a set of training image–caption pairs.

The background likelihood of an observation $O$ (*i.e.*, $p(O|H_B)$) is independent of the model $G$ and depends only on the features $\mathbf{f}_m$ of the observed points and their spatial relationships $r_{mn}$. In other words, $p(O|H_B)$ can be estimated from the background feature probability, $p(\mathbf{f}_m|H_B)$, and the background distribution of spatial relationships, $p(r_{mn}|H_B)$. $p(\mathbf{f}_m|H_B)$ reflects how common a feature is across a set of stock images with a wide variety of objects, while $p(r_{mn}|H_B)$ reflects the distribution of relationships observed between neighboring points across the stock image set.

According to the background hypothesis, all point feature vectors $\mathbf{f}_m$ are i.i.d. (independent and identically distributed). While some local features represent one of a few very common visual patterns, other local feature values are quite rare, and therefore more distinctive. A Gaussian mixture model (GMM) allows us to approximate such a structured background feature distribution:

$$p(\mathbf{f}_m|H_B) = \sum_{c=1}^{|C|} \omega_c \cdot n(\mathbf{f}_m|\mathbf{f}_c, \sigma_c) \quad (6)$$

where $\omega_c$ is a weight term ($\sum \omega_c = 1$) and $n(\mathbf{f}_m|\mathbf{f}_c, \sigma_c)$ is a multivariate normal distribution with mean $\mathbf{f}_c$ and diagonal covariance values $\sigma_c^2$. Each mean $\mathbf{f}_c$ is a member of the cluster centroid set $C$, found using K-means as explained in Section II-A above. Given these fixed means, the weights $\omega_c$ and standard deviations $\sigma_c$ are determined using the $EM$ algorithm on the same (large) set of stock images that we use to find clusters in $C$. This approach provides a smoother background feature distribution than using the statistics of the K-means clusters directly and is less computationally expensive than full GMM clustering.

According to the background hypothesis, all spatial relationship vectors $r_{mn}$ between neighboring points are also i.i.d. Since histograms of relative distance ($\Delta x_{mn}$) and relative scale ($\Delta\lambda_{mn}$) are unimodal in the stock set, we model them with a normal distribution:

$$p(\Delta x_{mn}|H_B) \;=\; n(\Delta x_{mn}|\mu_{xB}, \sigma_{xB}) \quad (7)$$

$$p(\Delta\lambda_{mn}|H_B) \;=\; n(\Delta\lambda_{mn}|\mu_{\lambda B}, \sigma_{\lambda B}) \quad (8)$$

where the means ($\mu_{xB}$ and $\mu_{\lambda B}$) and standard deviations ($\sigma_{xB}$ and $\sigma_{\lambda B}$) are the sample statistics for pairs of neighboring points in the stock image set. For the two relative heading terms ($\Delta\phi_{mn}$, $\Delta\phi_{nm}$) we did not observe any tendency to a particular value in the stock image set, hence we model them as uniformly distributed.

Calculation of the model likelihood of an observation, $p(O|H_G)$, is more complex, since the components of the appearance model are not identically distributed. In order to account for model vertices that are occluded or lost due to inconsistent interest point detection, we assume each vertex $v_i \in V$ is observed with probability $\alpha_v$.[1] In matching an image point $p_m$ to a model vertex $v_i$, we do not require the feature vector $\mathbf{f}_m$ of $p_m$ to be precisely equal to $\mathbf{f}_i$, but we assume that $\mathbf{f}_m$ is normally distributed with mean $\mathbf{f}_i$ and standard deviation $\sigma_f$. This vertex feature deviation $\sigma_f$ is approximately equal to the background's mean cluster variance $\overline{\sigma_c}$. While a graph vertex $v_i \in V$ defines the expected feature vector of a model observation, a graph edge $e_{ij} \in E$ defines the expected spatial relationships between certain pairs of observed points. If $O$ contains observations $(v_i, p_m)$ and $(v_j, p_n)$ and $e_{ij} \in E$, then the elements of the spatial relationship $r_{mn}$ are independent and normally distributed with means $(\Delta x_{ij}, \Delta\lambda_{ij}, \Delta\phi_{ij}, \Delta\phi_{ji})$ and standard deviations $(\sigma_x, \sigma_\lambda, \sigma_\phi, \sigma_\phi)$. If the model does not specify a relationship between the vertices ($e_{ij} \notin E$), then the elements of $r_{mn}$ are distributed according to the background hypothesis.

From the above formulations for the background and model likelihoods of an observation, we can calculate the observation likelihood ratio as:

$$\frac{p(O|H_G)}{p(O|H_B)} = \prod_{v_i \notin O}(1-\alpha_v) \prod_{(v_i,p_m)\in O} \alpha_v \frac{p(\mathbf{f}_m|\mathbf{f}_i)}{p(\mathbf{f}_m|H_B)} \prod_{e_{ij}\ \mathrm{in}\ O} \frac{p(r_{mn}|e_{ij})}{p(r_{mn}|H_B)} \quad (9)$$

where $p(\mathbf{f}_m|\mathbf{f}_i)$ and $p(r_{mn}|e_{ij})$ reflect how well the observed points and their spatial relationships match the values expected by the appearance model $G$ (as explained in the paragraph above). We consider $e_{ij}$ to be in $O$ if both $(v_i, p_m) \in O$ and $(v_j, p_n) \in O$. Note that the observation likelihood ratio takes into account the size of the observed and unobserved portions of the model, the background likelihood of the observed features and spatial relationships, as well as how well the observed points and spatial relationships fit $G$.

*2) Detection Algorithm:* To detect an instance of an object model in an image, we need to find a high-confidence association between the model vertices and image points. Given that a typical image contains thousands of points, determining the optimal association is potentially quite expensive. We thus propose a greedy heuristic that efficiently searches the space of possible associations for a nearly-optimal solution. Individual vertices of a model may be unobserved (*e.g.*, due to occlusion), and therefore some edges in the model may also not be instantiated. Our detection heuristic thus allows for a connected model to be instantiated as disconnected components. To reduce the probability of false detections, the search for disconnected parts is confined to the neighborhood of observed vertices, and isolated singleton points are ignored. That is, in a valid model instance, each observed vertex shares a link with at least one other observed vertex. Also, our detection algorithm only reports those observations $O$ with a detection confidence greater than a threshold, *i.e.*, $\mathrm{Conf}_{detect}(O,G) \geq T_d$. We set $T_d$ to be quite low so that potentially meaningful detections are not overlooked.

Algorithm 1 presents the greedy heuristic that detects instances of an appearance model $G$ within the image representation $I$. The actual implementation can detect more than one instance of $G$ within $I$ by suppressing points in previously detected instances.

[1] More accurately, we treat $\alpha_v$ not as the empirically observed probability of model vertex survival, but rather as what we would like it to be (i.e., as a user-defined parameter to guide detection). We chose $\alpha_v = 0.5$ because it is a reasonable compromise between a high $\alpha_v$ which requires that almost all elements of a model be reproduced, and a low $\alpha_v$ where a great majority of the model vertices can be absent with little impact on detection confidence.

The first step is to find the set $\mathcal{A}$ of potential associations between image points and model vertices, that is, all vertex–point pairs whose match ($p(\mathbf{f}_m|\mathbf{f}_i)$) is higher than expected by chance. The algorithm then calculates the set of potential model edges $\mathcal{L}$ linking these vertex–point pairs. The set of observed correspondences, $O$, is assembled across several iterations by greedily adding the vertex–point pairs from $\mathcal{A}$ that maximize the detection confidence, $\text{Conf}_{detect}(O, G)$. Each time $O$ is expanded, elements of $\mathcal{A}$ and $\mathcal{L}$ that are incompatible with the current observed set are pruned away. The greedy expansion of $O$ continues until there are no available correspondences that could increase $\text{Conf}_{detect}(O, G)$.

**Algorithm 1** Detects instances of $G$ in $I$

```
FindModelInstance($G$,$I$)
 1) Find the set $\mathcal{A}$ of all potential vertex–point associations:
      $\mathcal{A} = \{(v_i, p_m) \,|\, c_m \in \mathbf{c}_i,\ p(\mathbf{f}_m \,|\, \mathbf{f}_i) > p(\mathbf{f}_m|H_B)\}$
 2) Find the set $\mathcal{L}$ of all potential links between elements of $\mathcal{A}$:
      $\mathcal{L} = \{((v_i, p_m), (v_j, p_n)) \,|\, p_n \in \mathbf{n}_m,\ p(r_{mn}|e_{ij}) > p(r_{mn}|H_B)\}$
 3) Set the initial instance $O$ to the pair $\{(v_i, p_m), (v_j, p_n)\}$, such
    that the link $((v_i, p_m), (v_j, p_n)) \in \mathcal{L}$ and $\text{Conf}_{detect}(O, G)$ is
    maximum.
 4) Remove $(v_i, p_m)$ from $\mathcal{A}$ if either $v_i$ or $p_m$ are part of another
    vertex–point association $\in O$.
 5) Remove $((v_i, p_m), (v_j, p_n))$ from $\mathcal{L}$ if neither end is in $\mathcal{A}$.
 6) Let $\mathcal{A}_{adj}$ be the subset of $\mathcal{A}$ that is linked through $\mathcal{L}$ with an
    element of $O$.
 7) If $\mathcal{A}_{adj}$ contains associations that could increase
    $\text{Conf}_{detect}(O, G)$, add to $O$ the association that leads to
    the greatest increase, and go to step 4.
 8) Let $\mathcal{L}_{neigh}$ be the subset of $\mathcal{L}$ within the union of the neigh-
    borhoods of observed points in $O$.
 9) If $\mathcal{L}_{neigh}$ contains observed links that could increase
    $\text{Conf}_{detect}(O, G)$, add to $O$ the pair of associations with the
    link that produces the greatest increase, and go to step 4.
10) Return $(O, \text{Conf}_{detect}(O, G))$.
```

## III. Discovering Word–Appearance Associations

We propose an unsupervised learning algorithm that builds structured appearance models for the salient objects appearing in a set of training image–caption pairs. Salient objects are those that appear in many images, and are often referred to in the captions. Because each image contains many features of non-salient objects, and each caption may contain words irrelevant to the displayed objects, the algorithm has to discover which image features and words are salient. The algorithm learns object models through discovering strong correspondences between configurations of visual features and caption words. The output is a set of appearance models, each associated with a caption word, which can be used for the annotation of new images.

The learning algorithm has two stages. First, an initialization stage determines a structured seed model for each caption word, by finding recurring neighborhoods of features that also co-occur with the word. Second, an improvement stage iteratively expands each initial seed model into an appearance model that covers a larger portion of the object of interest, and at the same time is more strongly associated with the corresponding caption word. The two stages of learning use a novel measure of correspondence between a caption word and an appearance model, which is explained in Section III-A. We then explain the initialization and improvement stages of the learning algorithm in Sections III-B and III-C, respectively.

### A. Word–Appearance Correspondence Confidence

Here, we explain the word–appearance correspondence score used by our learning algorithm. The learning algorithm seeks pairs of words and appearance models that are representations of the same object in different modalities (linguistic and visual). We assume that both the word and the model instance are present in an image because the *object* is present. We thus define the correspondence score as a measure of confidence that a given appearance model is a reliable detector for the object referred to by a word. In other words, the correspondence score reflects the amount of evidence, available in a set of training images, that a word and an object model are generated from a common underlying source object.

Consider a set of $k$ (training) captioned images. We represent the occurrence pattern of a word $w$ in the captions of these images as a binary vector $\mathbf{r}_w = \{r_{wi}|i = 1, \ldots, k\}$. Similarly, we use a binary vector, $\mathbf{q}_G = \{q_{Gi}|i = 1, \ldots, k\}$ to indicate, for each training image, whether it contains at least one true observation of model $G$. However, even if we detect model $G$ in the $i^{th}$ training image, we cannot be certain that this is a true observation of $G$ ($q_{Gi} = 1$) instead of a random assortment of background features ($q_{Gi} = 0$). Therefore, we treat $q_{Gi}$ as a hidden variable and associate it with an observed value, $o_{Gi} \in [0, 1]$, that reflects the likelihood of model $G$ being present in image $i$. We set $o_{Gi}$ to the maximum of the detection confidence scores, $\text{Conf}_{detect}(O, G)$, over all the detected instances of a given object model $G$ in a given image $i$.

It is always possible that the word occurrence pattern, $\mathbf{r}_w$, and the observed confidence pattern, $\mathbf{o}_G = \{o_{Gi}|i = 1, \ldots, k\}$, are independent (the null hypothesis or $H_0$). Alternatively, instances of the word $w$ and model $G$ may both derive from a hidden common source object (the common-source hypothesis or $H_C$). According to $H_C$, some fraction of image–caption pairs contain a hidden source $s$, which may emit the word $w$ and/or the appearance model $G$. The existence of the appearance model in turn influences our observed confidence values, $o_{Gi}$. We define the correspondence between $w$ and $G$ as the likelihood $P(H_C|\mathbf{r}_w, \mathbf{o}_G)$, and rewrite it as in Equation (11) below:

$$\begin{aligned}\text{Conf}_{corr}(G, w) &= P(H_C|\mathbf{r}_w, \mathbf{o}_G) && (10)\\ &= \frac{p(\mathbf{r}_w, \mathbf{o}_G|H_C)P(H_C)}{p(\mathbf{r}_w, \mathbf{o}_G|H_C)P(H_C) + p(\mathbf{r}_w, \mathbf{o}_G|H_0)P(H_0)}\end{aligned}$$

where:

$$p(\mathbf{r}_w, \mathbf{o}_G|H_C) = \prod_i \sum_{s_i} P(s_i)\, P(r_{wi}|s_i)\, p(o_{Gi}|s_i) \quad (11)$$

$$p(\mathbf{r}_w, \mathbf{o}_G|H_0) = \prod_i P(r_{wi})\, p(o_{Gi}) \quad (12)$$

where $s_i \in \{0, 1\}$ represents the presence of the common source in image–caption pair $i$. To calculate the likelihoods of the observed confidence values under the two competing hypotheses ($p(o_{Gi}|s_i)$ and $p(o_{Gi})$) we marginalize over the unobserved variable $q_{Gi}$:

$$\begin{aligned}p(o_{Gi}|s_i) &= p(o_{Gi}|q_{Gi} = 1)P(q_{Gi} = 1|s_i)\\ &\quad + p(o_{Gi}|q_{Gi} = 0)P(q_{Gi} = 0|s_i) && (13)\\ p(o_{Gi}) &= p(o_{Gi}|q_{Gi} = 1)P(q_{Gi} = 1)\\ &\quad + p(o_{Gi}|q_{Gi} = 0)P(q_{Gi} = 0) && (14)\end{aligned}$$

where we choose $p(o_{Gi}|q_{Gi} = 1)$ to be proportional to the detection confidence $o_{Gi}$ (hence $p(o_{Gi}|q_{Gi} = 0)$ will be proportional to $1 - o_{Gi}$). The intuition is that $o_{Gi}$ is usually high when the model $G$ is present in image $i$, and is low when the model is not present.

To get the likelihood of observed data under $H_C$, defined in Equations (11) and (13), we also need to estimate the parameters $P(s_i), P(r_{wi}|s_i)$, and $P(q_{Gi}|s_i)$. $P(r_{wi}|s_i)$ and $P(q_{Gi}|s_i = 0)$ are given fixed values according to assumptions we make about the training images, which we elaborate on in the Appendix. $P(s_i)$ and $P(q_{Gi}|s_i = 1)$ are given maximum likelihood estimates (MLEs) determined using expectation maximization over the training data. The MLEs for parameters under $H_0$ are more straightforward. $P(r_{wi})$ in Equation (12) is the observed probability for word occurrence in the training data while $P(q_{Gi})$ is the inferred probability of model occurrence: $\sum_i o_{Gi}/k$.

### B. Model Initialization

The goal of the model initialization stage is to quickly find for each word a set of seed appearance models that are fruitful starting points for building strong object detectors. Later, the seed object models are iteratively expanded and refined into larger and more distinctive appearance models using image captions as a guide. The existence of good starting points strongly affects the final outcome of the iterative improvement process. Nonetheless, a balance must be reached between time spent searching for good seeds and time spent in the improvement stage refining the seeds.

The most straightforward starting points are singleton features, but their relationship to an object may be too tenuous to provide effective guidance for building strong object models [15]. At the same time, trying all possible configurations of even a small number of features as seeds is impractical. The neighborhood pattern introduced by Sivic and Zisserman [25] roughly describes the appearance of a point's local context as a bag of features. The neighborhood pattern is more distinctive than a singleton but less complex than our configurational models. Our initialization module uses neighborhood patterns with potentially meaningful word correspondences to help construct seed appearance models.

Recall that each point $p_m$ in an image has associated with it a vector of neighboring points $\mathbf{n}_m$. The neighborhood pattern $\eta_m$ is a sparse binary vector that denotes which quantized feature descriptors $c$ are present in $p_m$'s neighborhood (including $c_m$). Thus $\eta_m$ roughly captures the types of feature vectors present within the small portion of the image centered at $p_m$. Two neighborhood patterns $\eta_m$ and $\eta_l$ are considered *similar* ($\eta_m \approx \eta_l$) if they have at least $t_\eta$ quantized feature descriptors in common. We use a modified version of the two-stage clustering method described in [25] to identify clusters of similar neighborhood patterns in the training images. The first stage identifies potential cluster centers with relatively low similarity threshold ($t_\eta = 6$) but requires that the quantized descriptor of the central feature of the neighborhoods match. The second pass that greedily forms the clusters has a higher similarity threshold ($t_\eta = 8$) but does not require a matching central descriptor.

Each resulting *neighborhood cluster* $\mathcal{N}_m$ is a set of neighborhoods with patterns similar to $\eta_m$ ($\mathcal{N}_m = \{\mathbf{n}_l | \eta_l \approx \eta_m\}$). Each neighborhood cluster represents a weak, unstructured appearance context that occurs multiple times across the training image collection. We represent the occurrence pattern of a given neighborhood cluster $\mathcal{N}$ in the training images as a binary vector $\mathbf{q}_\mathcal{N} = \{q_{\mathcal{N}i} | i = 1, \cdots, k\}$, where $q_{\mathcal{N}i} = 1$ if image $i$ contains a member of $\mathcal{N}$, and zero otherwise.

The initialization module measures the association between each caption word $w$ and neighborhood cluster $\mathcal{N}$, using the correspondence confidence score of Equation (11) above, *i.e.*, $\text{Conf}_{corr}(\mathcal{N}, w)$.[2] We assume that occurrences of a neighborhood cluster $\mathcal{N}$ that has a better-than-chance correspondence to a word $w$ may spatially overlap the object referred to by $w$. We therefore select for each word $w$ the 20 neighborhood clusters $\mathcal{N}$ with the strongest correspondence score and attempt to extract from each cluster a seed appearance model with the best possible correspondence with $w$. Since the simplest detectable and structured appearance model is a single pair of linked vertices, we search for two-vertex seed appearance models $G$ that could explain the presence of frequently-occurring point pairs in $\mathcal{N}$ and that also have strong correspondence with $w$.

Two neighboring points are considered to be a pair in a neighborhood cluster $\mathcal{N}$ if at least one end of the pair belongs to one of the neighborhoods in $\mathcal{N}$. We extract groups of "similar" pairs in $\mathcal{N}$, *i.e.*, those that share the same appearance cluster pairs $(c_m, c_n)$. Each such group $\mathcal{P}$ is viewed as a set of observations for a potential two-vertex appearance model $G$. For each $\mathcal{P}$ with members appearing in more than one image, we propose up to 20 distinct appearance models whose features and spatial relationships are randomly drawn (without replacement) from the pairs in $\mathcal{P}$. For each model $G$, we calculate a score that reflects how well $\mathcal{P}$ supports model $G$, by treating the $K$ pairs in the group as observations $O_k$:

$$\text{Supp}(\mathcal{P}, G) = \sum_{k=1,\ldots,K} p(O_k | H_G) \qquad (15)$$

We keep the model with the highest $\text{Supp}(\mathcal{P}, G)$ for each group of pairs. Then, for as many as 50 models from different pair groups with the highest support, we calculate $\text{Conf}_{corr}(G, w)$. The model with the highest correspondence confidence is selected as word $w$'s seed model for the current neighborhood cluster $\mathcal{N}$. Each starting seed model is therefore initialized from a different neighborhood cluster. While it is possible that different seed models may converge on the same appearance during the following iterative improvement stage, ideally the set of learned appearance models will cover different distinctive parts of the object and also provide coverage from a variety of viewpoints.

### C. Iterative Improvement

The improvement stage iteratively makes simple changes to the seed object models found at the initialization stage, guided by the correspondence between caption words and models. More specifically, the improvement algorithm starts with a seed model $G$ for a given word $w$, makes a simple modification to this model (*e.g.*, adds a new vertex), and detects instances of the new model $G'$ in the training images (using the detection algorithm presented in Section II-C). The new model is accepted as a better object detector and replaces its predecessor if it has a higher correspondence score with $w$, *i.e.*, if $\text{Conf}_{corr}(G', w) > \text{Conf}_{corr}(G, w)$. In other words, the improvement algorithm performs a greedy search through the space of appearance models to find a reliable object detector for any given word.

At each iteration, the algorithm tries to expand the current model by adding a new vertex and linking it with one of the

[2] In our calculation of $\text{Conf}_{corr}(\mathcal{N}, w)$, we replace $\mathbf{q}_G$ with $\mathbf{q}_\mathcal{N}$ indicating the occurrences of $\mathcal{N}$ in the training images.

existing vertices. Vertex candidates are drawn from points that fall within the neighborhood of the detected instances of the current model. To ensure a strong correspondence between the growing model and its associated word, only model detections that occur within images with the desired caption word $w$ are considered for this purpose. Consider a detected point $p_m$ that corresponds to a model vertex $v_i$. Each point $p_n$ that is in the neighborhood of $p_m$ but not part of the detection is a candidate to form a new vertex $v_j$. The corresponding edge candidate, $e_{ij}$, inherits the spatial relationship vector $r_{mn}$ between the two points $p_m$ and $p_n$. As in the initialization stage, from the observations and their neighboring points, we form groups $\mathcal{P}$ each containing observations of a potential one-vertex extension to the current model. The observations are pairs $(p_m, p_n)$ with their first points being observations of the same existing vertex $v_i$, and their second points sharing the same cluster index $c_n$. For each pair group $\mathcal{P}$, we propose up to 20 two-vertex incremental models $\Delta G = (\{v_i, v_j\}, e_{ij})$ that bridge the gap between the existing model and the new vertex. We keep the incremental model with the highest $\text{Supp}(\mathcal{P}, \Delta G)$. The incremental models from different pair groups form a queue of potential (vertex, edge) additions, prioritized by their degree of support.

An augmented model, $G'$, is constructed by removing the top incremental model $\Delta G$ from the queue and incorporating it into the existing model ($G' = G \cup \Delta G$). If the augmented model does not improve the correspondence score with $w$, then the change is rejected and the next iteration begins with the next candidate $\Delta G$ to be tested. If the augmented model *does* improve the correspondence score, the change is accepted, and the algorithm attempts to establish additional edges between the existing vertices and the new vertex. The edge candidates are prioritized based on their support among detections of the new model $G'$, as this reflects the number of times the two end point vertices have been observed together, as well as the consistency of their spatial relationship over those observations. New edges are tested sequentially and those that improve the correspondence score are added. Generally, if the model vertices have very consistent spatial relationships across the current detections, the model will tend to accept many edges. If the underlying object is more deformable or viewed from a variety of perspectives, fewer edges are likely to be accepted.

Once a new vertex is added and connected to the model, and additional edges are either accepted or rejected, a new iteration begins with the new model as the starting point. If none of the proposed model extensions are accepted, the model $G$ paired with the caption word $w$ is added to the set of discovered word–appearance associations to be used for future annotation of new images.

## IV. Evaluation: Annotating Objects in Uncaptioned Images

In previous sections, we have presented the components of our learning framework for discovering object models and their associated names from a set of training image–caption pairs. Ultimately, we want to use the discovered model–word pairs to detect and annotate new instances of the objects in unseen and uncaptioned images. For detection of new instances of a given object model, we use the detection algorithm presented in Section II-C.2. Our confidence that a word $w$ is appropriate to annotate a detected object $O$ in an image depends on two factors: (i) our confidence that the observed instance $O$ is a true instance of the given model $G$ and not a chance configuration of background features—*i.e.*, the detection confidence $\text{Conf}_{detect}(O, G)$; and (ii) our confidence that the appearance model $G$ and the word $w$ represent the same object—*i.e.*, the correspondence confidence $\text{Conf}_{corr}(G, w)$. We can thus associate an *annotation confidence* to every object instance that is to be labeled in a new image. The annotation confidence is defined as the product of the detection confidence and the correspondence confidence, as in:

$$\text{Conf}_{annotate}(O, w, G) = \text{Conf}_{detect}(O, G) \times \text{Conf}_{corr}(G, w) \tag{16}$$

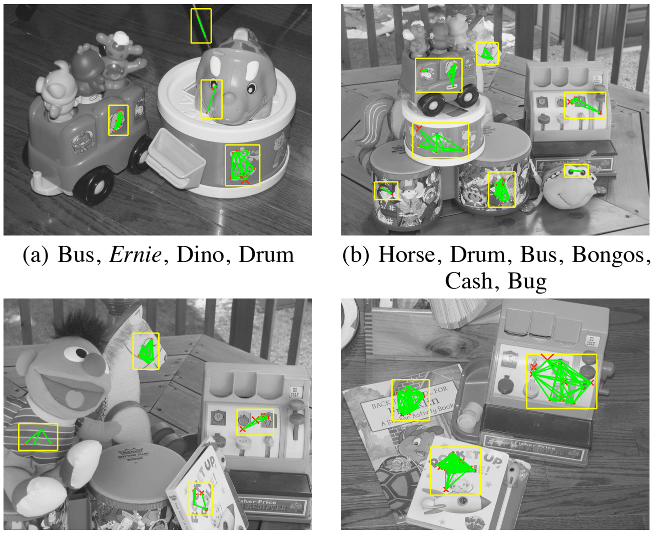
We annotate a new instance of an object only if the annotation confidence is greater than a threshold, *i.e.*, if $\text{Conf}_{annotate}(O, w, G) > T_a$. The value of $T_a$ could be determined by a user, depending on whether they desire more detections (higher recall) or fewer detections with higher confidence (higher precision). In all experiments reported here, we set the threshold very high, *i.e.*, $T_a = 0.95$.

The following sections present the results of applying our learning, detection, and annotation algorithms to two types of data sets: a small set of real images of toys that we captured and annotated ourselves (Section IV-A) and two larger and more challenging sets of real-world images of sports scenes and landmarks, respectively, downloaded from the web (Section IV-B). Our choices for the parameters involved in the confidence scores and the algorithms are given in the Appendix.

### A. Experiments on a Controlled Data Set

Here, we report the performance of our method applied to a set of 228 images of arrangements of children's toys, generated under controlled conditions. The data set was first used in our experiments presented in an earlier paper [15]. Throughout this article, we refer to this data set as the TOYS data set. The original color photographs are converted to intensity images with a resolution of 800x600. Most images contain 3 or 4 toy objects out of a pool of 10, though there are a handful of examples of up to 8 objects. The objects are not arranged in any consistent pose and many are partially occluded. The images are collected against approximately 15 different backgrounds of varying complexity. The pool of 228 images is randomly divided into a training set of 128 and a test set of 100. Each training image is annotated with the unique keyword for each object of interest shown and between 2 and 5 other keywords uniformly drawn from a pool of distractor labels. Note that the objects of interest never appear individually and the training data contains no information as to the position or pose of the labeled objects.

Figure 3 displays some example images from the test set and the associated annotations produced by our system. False annotations are written in italics and missed annotations in parentheses. While there is no direct limit on the size of learned appearance models, they tend to cover small, distinctive patches of the objects. In many cases, the size of learned models is limited by the availability of sufficient repeatable visual structure. Some objects with large areas of sufficient detail, such as the 'Cash' object and the two books 'Franklin' and 'Rocket', can support larger models (as in Figure 3(d)). Our method learns multiple appearance models for many of the objects, such as the two models shown in Figure 3(b) for 'Bongos'. It is thus possible to detect an object even when a distinctive part of it is occluded. The



(a) Bus, *Ernie*, Dino, Drum

(b) Horse, Drum, Bus, Bongos, Cash, Bug

(c) Ernie, Horse, Rocket, Cash, (Bongos)

(d) Franklin, Rocket, Cash

Fig. 3. Sample detections of objects in the TOYS test set. False detections are shown in italics while missed detections are placed in parentheses.

system can sometimes detect planar objects such as the 'Rocket' book under significant perspective distortion (as in Figure 3(c)).

Our earlier work ([15], [16]) has shown promising results on the TOYS data set, while using simpler appearance models and/or less efficient learning or detection algorithms. To evaluate the improvement in annotation performance, we performed a single training run for our method on the same 128 training images used in previous work and compared results on the 100-image test set. Figure 4 shows the precision–recall curves for four systems: the current system (with and without spatial relationships), the ICCV system presented in [16], and the CVPR system described in [15]. To implement a system without spatial relations, we remove all spatial contributions to the detection confidence measure, $\text{Conf}_{detect}(O, G)$. Therefore, edge position and connectivity play no role in the detection mechanism. The only remaining spatial constraint in the resulting bag-of-features model is that each vertex added to a detection must fall within the neighborhood of a vertex that is already part of the detection (a constraint of the underlying detection algorithm). Our new system achieves the high annotation precision of ICCV with about 15% higher overall recall due to our new detection confidence measure. Training without spatial relationships generates approximately a 12% penalty in recall, indicating that more distinctive object models can be constructed by finding recurring spatial relationships among image points. While the CVPR system and the 'No Spatial' variant of our current system both represent appearance as an unstructured collection of local features, a variety of changes in the initialization mechanism, the representation of individual features and the detection and correspondence measures greatly improve overall performance.[3]

Table I shows the per-object precision and recall values of our current system. In this and subsequent tables, the Frequency column shows the number of captions within the test set that contain at least one instance of the corresponding word. All of the precision and recall values we report are based on word occurrence in the captions of the test set; if the system does not detect a word that appears in the caption, that instance is counted as a false positive, even if the named object does not actually appear in the image. The method performs best on objects that have large, roughly planar surfaces and distinctive structural details, such as 'Rocket', 'Franklin', 'Drum' and 'Horse'. The only instances of these objects that cannot be detected with very high precision either are highly occluded or are viewed from an atypical angle (such as edge-on for a book). Our system has the greatest difficulty with the 'Ernie' and 'Dino' objects, perhaps because they lack fine-scale surface details and distinctive textures. For instance, the striped pattern of the shirt of the 'Ernie' doll is somewhat distinctive within the image set, but the face lacks sufficient keypoints to build a reliable model. The 'Dino' object is particularly difficult, as specular reflections significantly alter the local-feature description of its surface appearance depending on perspective and lighting.

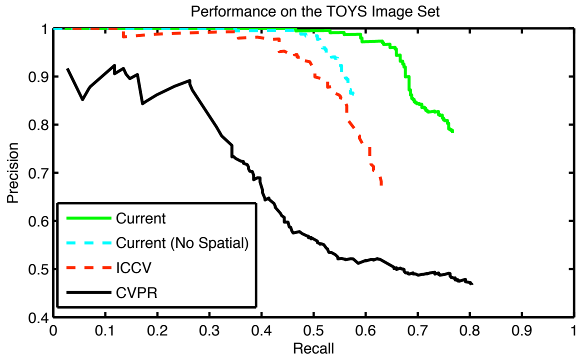


Fig. 4. A comparison of precision–recall curves over the TOYS test set, for four systems: current, current without spatial relationships, ICCV [16], and CVPR [15]. CVPR used a local bag-of-features models that were initialized with singleton features. ICCV added spatial relationships and neighborhood initialization. The current system adds detection and annotation confidence scores and builds models with more spatial relationships.

| Name | Precision | Recall | Frequency |
|---|---|---|---|
| Horse | 1.00 | 0.82 | 28 |
| Rocket | 1.00 | 0.82 | 44 |
| Drum | 1.00 | 0.81 | 32 |
| Franklin | 1.00 | 0.73 | 33 |
| Bus | 1.00 | 0.60 | 57 |
| Bongos | 1.00 | 0.53 | 36 |
| Bug | 1.00 | 0.53 | 51 |
| Dino | 1.00 | 0.12 | 42 |
| Cash | 0.97 | 0.76 | 46 |
| Ernie | 0.94 | 0.41 | 39 |

TABLE I

PERFORMANCE RESULTS ON THE TOYS TEST SET; $T_a = 0.95$.

Precision and recall indicate whether the system is detecting objects in the correct images, but not how often the models are detected on the objects themselves. To evaluate the locational accuracy of the detections we manually defined bounding boxes for all named objects in the test image set (this data was not available to the system). A detection was considered accurate if all of the detected model vertices were located within the correct bounding box. Overall, 98.8% of above-threshold detections on the TOYS data set are completely within the boundaries defined for the object. The lowest accuracy was for the 'Franklin' object, on which 5.2% of detections were partially outside the object bounds.

[3] The improved precision of the ICCV system over the CVPR system is due in part to the addition of spatial relationships to the appearance models, as well as to improvements in the correspondence confidence measure and initialization method.

### *B. Experiments on Web Data Sets*

This section reports the performance of our system on two larger and more challenging sets of images downloaded from the web. The first set, which we refer to as the HOCKEY data set, includes 2526 images of National Hockey League (NHL) players and games, with associated captions, downloaded from a variety of sports websites. The second set, which we refer to as the LANDMARK data set, contains 3258 images of 27 well-known buildings and locations, with associated tags, downloaded from the Flickr website[4]. Due to space considerations, our analysis focuses mainly on the results on the HOCKEY data set (IV-B.1 through IV-B.3), though most of the same phenomena also appear in the LANDMARK results (IV-B.4).

*1) Annotation Performance on the* HOCKEY *Data Set:* The HOCKEY set contains examples of all 30 NHL teams and is divided into 2026 training and 500 test image–caption pairs. About two-thirds of the captions are full sentence descriptions, whereas the remainder simply name the two teams involved in the game (see Figure 1, page 3 for examples of each type). We automatically process captions of the training images, removing capitalization, punctuation, and plural indicators, and dropping words that occur in less than 1% of the captions. Captions of the test images are only used for evaluation purposes.

Most images are on-ice shots and display multiple players in a variety of poses and scales. We thus expect our system to learn distinctive appearance models for the players of each team, and to discover meaningful associations between the models and the corresponding team names. A team's logo is perhaps the most distinctive appearance model that our system could learn for the team. In addition, there may be other visual appearances that unambiguously identify a particular team, such as shoulder patches or sock patterns. Note that we do not incorporate any prior knowledge into our system about which objects or words are of interest. The system is expected to learn these from the pairings of the images and captions in the training data. The only information we provide to our system is a link between a team's name (*e.g.*, Bruins) and its city name (*e.g.*, Boston) because most NHL teams are referred to by both names. Our system thus treats the two words (team name and city name) as the same word when learning model–word associations. The final vocabulary extracted from the training captions contains 237 words, of which only 30 are team designations. As we will see later, our system learns appearance models for many of these words, including team names as well as other words.

We experiment with different degrees of spatial constraints imposed by our detection algorithm, in order to see how that affects the learning and annotation performance of our system. Our detection confidence score, presented in Section II-C, has a set of parameters corresponding to the model spatial relationship variances. We set each of these to a fraction $1/\Gamma$ of the corresponding background variance, where $\Gamma$ is the spatial tolerance parameter that determines the amount of spatial constraints required by the detection algorithm to consider an observation as a true instance of a given model.

High values of $\Gamma$ thus imply a narrow acceptability range for spatial relationships between any two connected vertices of a model, resulting in tighter spatial constraints when detecting instances of the model. In contrast, a low value of $\Gamma$ translates into looser spatial constraints in detection. Figure 5 shows the overall precision–recall curves for various settings of the spatial tolerance parameter $\Gamma$, including a version in which the spatial relationships between points do not contribute to the detection confidence function (No Spatial). We use the notation $\Gamma_i$ to refer to an implementation of the system where $\Gamma$ is given the value $i$.

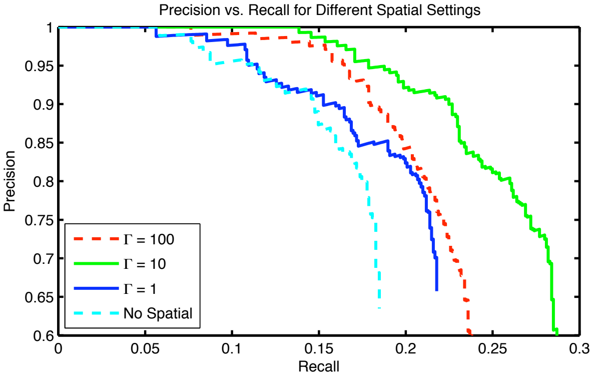


Fig. 5. Precision–recall curves of our current system on test HOCKEY images, over a wide range of spatial settings. Although the system can detect objects even without spatial constraints (the 'No Spatial' case), the use of moderate spatial constraints (when $\Gamma = 10$) offers the best performance.

The curves indicate that the implementation that ignores spatial factors (No Spatial) and $\Gamma_1$ (where model spatial variances are identical to background spatial variances) are roughly equivalent. Remaining differences are due to the fact that $\Gamma_1$ can learn some edge connection structure while the 'No Spatial' model cannot. Very strong spatial constraints, as in $\Gamma_{100}$, may result in brittle model detections, but still the system is capable of producing reasonably good results (*e.g.*, substantially better than a bag-of-features model; see Figure 5). Nonetheless, results confirm that moderate spatial constraints are generally more effective, as with $\Gamma_{10}$. One might expect that stronger spatial constraints (higher values of $\Gamma$) would always lead to higher precision. This is not necessarily the case, because even a configuration of relatively few observed vertices may have high detection confidence if their spatial relationships conform to the model's tight constraints. Moreover, many of the false annotations on test images are not the result of incorrect model detection at annotation time, but are due to learning a spurious word–appearance correspondence. Finally, a model that requires a rigid spatial configuration among its components may not grow to the size of a model that is more accommodating. The resulting smaller models may be less distinctive, even though each edge is more precise in capturing a particular spatial configuration.

To analyze the sensitivity of our method to the precise value of the spatial tolerance parameter $\Gamma$, we perform experiments with a smaller range of values for this parameter. Figure 6 shows the precision–recall curves for $\Gamma$ set to 5, 10, and 20. The results confirm that our method is not sensitive to small changes in the value of $\Gamma$. This might indicate that the iterative improvement process can optimize appearance models to compensate for different spatial tolerance values within a reasonable range. It is also possible that the models themselves are effective across a range of $\Gamma$ values, such that, for example, system performance would not be adversely affected if different values of $\Gamma$ were used for training and annotation.

[4] www.flickr.com

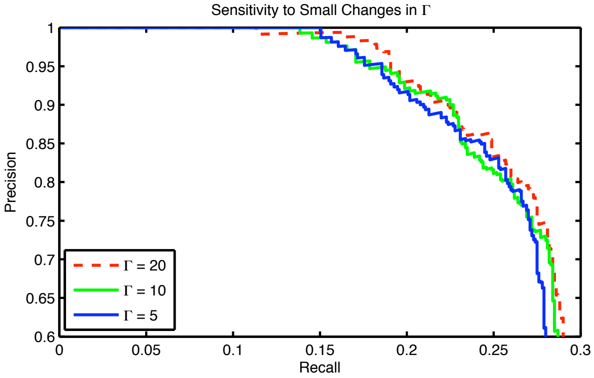


Fig. 6. Precision–recall curves for our current system are relatively unaffected by modest changes in the spatial tolerance parameter Γ.

| Name | Precision | Recall | Frequency |
|---|---|---|---|
| Tampa Bay Lightning | 1.00 | 0.51 | 49 |
| Los Angeles Kings | 1.00 | 0.33 | 36 |
| New York Islanders | 1.00 | 0.32 | 60 |
| Calgary Flames | 1.00 | 0.31 | 26 |
| Dallas Stars | 1.00 | 0.31 | 42 |
| Minnesota Wild | 1.00 | 0.26 | 35 |
| Chicago Blackhawks | 1.00 | 0.24 | 35 |
| New York Rangers | 1.00 | 0.21 | 42 |
| Buffalo Sabres | 1.00 | 0.19 | 32 |
| Carolina Hurricane | 1.00 | 0.17 | 30 |
| Nashville Predators | 1.00 | 0.10 | 20 |
| Colorado Avalanche | 1.00 | 0.09 | 23 |
| Toronto Maple Leafs | 0.96 | 0.30 | 73 |
| Ottawa Senators | 0.90 | 0.16 | 58 |
| Pittsburg Penguins | 0.89 | 0.28 | 29 |
| Atlanta Thrashers | 0.86 | 0.17 | 35 |
| New Jersey Devils | 0.85 | 0.19 | 59 |
| Detroit Red Wings | 0.83 | 0.24 | 42 |
| San Jose Sharks | 0.83 | 0.22 | 23 |
| Florida Panthers | 0.83 | 0.20 | 25 |
| Montreal Canadiens | 0.80 | 0.17 | 23 |
| Vancouver Canucks | 0.57 | 0.10 | 40 |
| Boston Bruins | 0.44 | 0.24 | 17 |

TABLE II

INDIVIDUAL PRECISION AND RECALL VALUES FOR 23 OF THE TEAM NAMES (OF 30) DETECTED WITH HIGH CONFIDENCE IN THE HOCKEY TEST SET; $\Gamma = 10$ AND $T_a = 0.95$.

*2) An Analysis of the Annotation Results for the* HOCKEY *Set:* Our results presented above show that $\Gamma_{10}$ has the best overall performance. We thus focus on $\Gamma_{10}$, analyzing various aspects of its performance. Table II shows the annotation performance of $\Gamma_{10}$ on the test images, focusing on the team names only. The system has high-confidence detections for 23 of the 30 teams. (There are 2 additional teams for which the system learns high-confidence models, but does not detect them in the test images.) Results show that the annotation of the test images generally has high precision but low recall. The low recall is partly because of our choice of a high annotation threshold, but also due to the fact that the captions that we use as ground truth often mention teams that are not visible in the image. In addition, a hockey player has a highly variable appearance depending on viewing angle and pose. A model that captures the appearance of the front logo will not help annotate a view of a player from the side.

An important factor that determines whether the system can learn high-confidence models for a team, and detect instances of it in the test images, is the number of training examples that contain both the team's name and some appearance of it. It is hard for the system to learn a high-confidence model for a team if the team's logo appears in a small number of training images. Even if the system learns a model for such a team, the model may be too specific to be useful for detecting new instances (which may differ in viewpoint, for example). On average, teams detected in the test set are mentioned in the captions of 114 training images, while teams with no test set detections are mentioned in the captions of 57 training images. The system detects only two teams (the Bruins and the Canadiens) with fewer than 60 caption-mentions, and fails to detect only one team (the Flyers) with more than 60 caption-mentions in the training set.

Note also that a team name's mention in the caption of an image does not necessarily mean that the corresponding object (*e.g.*, the team's logo) appears in the image. After analyzing 15 of the teams in Table II, we found that only 30–40% of the training images that mention a team's name also contain a visible (less than half obscured) instance of one of the team's logos. Therefore, a team with fewer than 60 training examples will almost always have fewer than 24 usable instances of the team logo. In addition, in some cases these instances will display various versions of the team's logo. For example, images of the Maple Leafs regularly show three different versions of their logo. The training set contains four completely different Buffalo Sabres chest logos, and many of these are placed on different backgrounds for home and away jerseys. As we see later in Figure 7(d), the system does not learn the new Sabres logo, but it does have a strong model for the older version of the logo displayed by the fan in the background.

For teams detected with relatively high recall, the system tends to learn separate models for each variation of the chest logo and often additional models for other distinctive parts of a player's uniform, such as shoulder patches or sock patterns. In some cases, the system learns several high-confidence models for the same logo. Some of these are redundant models that have nearly identical detection patterns, while in other cases the models describe the logo in different modes of perspective distortion. Teams with lower recall tend to only have one or two high-confidence appearance models describing a single logo variation. The training restriction of only 20 seed appearance models for each named object is perhaps ill-advised, given that multiple appearance models are helpful and only about 1 in 6 seed models leads to a high-confidence object appearance model.

The visual properties of a team's logo also affect whether the system learns an association between its appearance and the team's name. For example, whereas the Philadelphia Flyers are mentioned in the captions of 168 training images, their primary logo lacks detailed texture and so attracts relatively few interest points. This may explain why the system did not learn a reliable detector for this logo.

*3) Sample Annotations from the* HOCKEY *Set:* Figure 7 shows some example annotations in the test images for $\Gamma_{10}$. As expected, team logos tend to be the most useful patches of appearance for recognizing the teams. The system is often able to detect learned logos that are distorted or partially occluded. In some cases, however, the system fails to detect learned logos that are relatively clean and undistorted. These might be due to contrast-reversal from a logo appearing against a different background, or perhaps missed detections of the underlying interest point detector.

While the team logos tend to be the most distinctive aspect of a player's appearance, the system has learned alternate models for a variety of teams. Figure 8 displays two such examples, including a shoulder patch and sock pattern. The system might also learn an association between a team name and an appearance patch that is not part of a player at all.

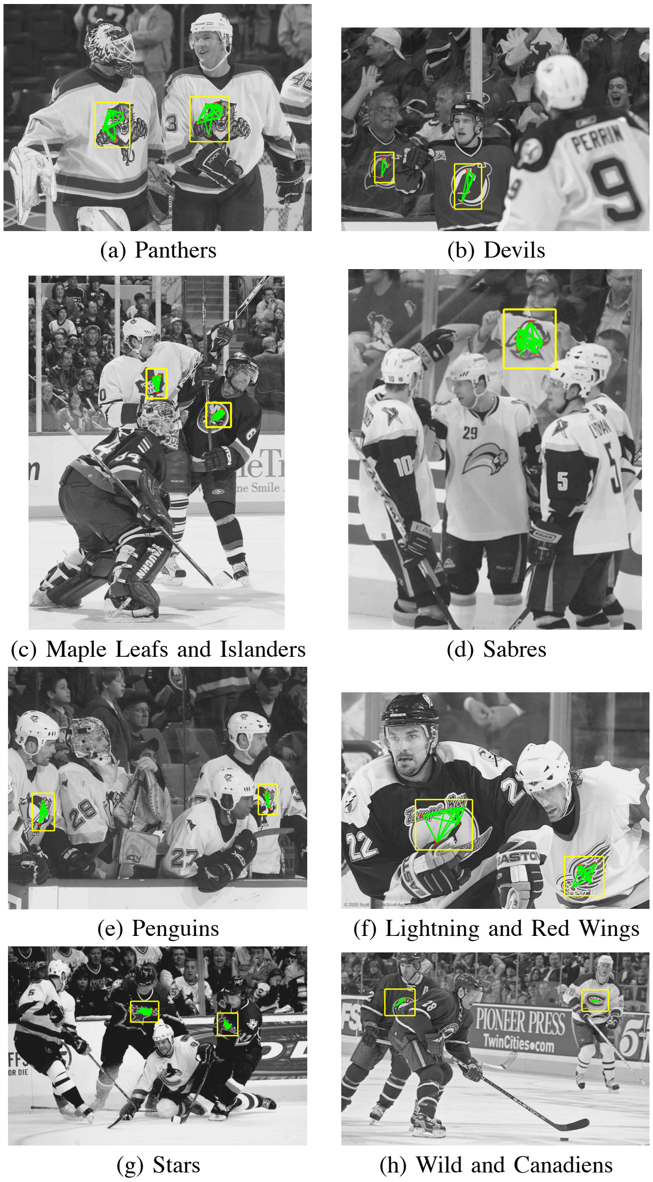

(a) Panthers
(b) Devils
(c) Maple Leafs and Islanders
(d) Sabres
(e) Penguins
(f) Lightning and Red Wings
(g) Stars
(h) Wild and Canadiens

Fig. 7. Sample annotations of our system in the hockey test images. The system automatically discovers chest logos as the most reliable method for recognizing a team.

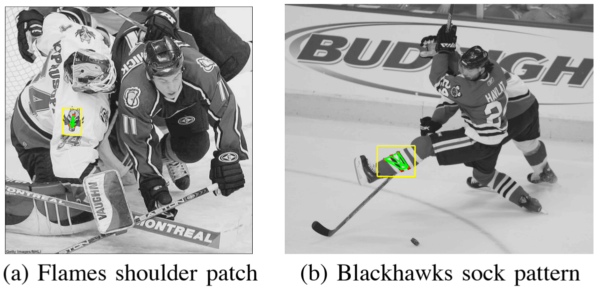

(a) Flames shoulder patch
(b) Blackhawks sock pattern

Fig. 8. Alternate models can be learned, such as shoulder patches or sock patterns.

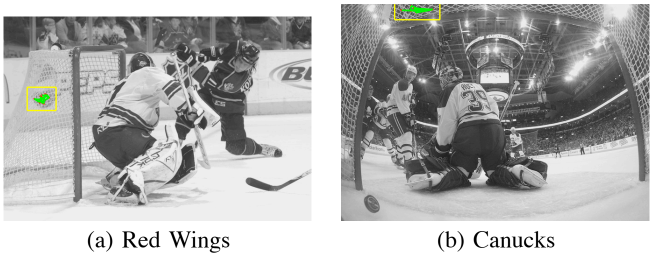

(a) Red Wings
(b) Canucks

Fig. 9. Due to overfitting, our system learns bad models (a white net on a dark background) for several teams.

Figure 9 displays instances of a particular type of false annotation. Variants of a white net pattern against a dark background are learned independently as reasonably high-confidence appearance models for several teams. This appears to be a result of overfitting. While this back-of-the-net appearance is quite common across the NHL image collection, certain variations of this appearance have a weak initial correspondence with particular teams. During iterative improvement, the system is able to tailor each team's version of the net appearance model to fit only the specific instances of the net appearance in the training data that are annotated with the desired team name. In this way, a model can have few false positives in the training set, even though the described appearance is not meaningfully associated to the team name. We are currently considering approaches for reducing the occurrence of such overfitting while still allowing the improvement stage to encode meaningful differences in appearance.

As mentioned previously, the system is not restricted to finding visual correspondences solely for team names. Figure 10 shows a few example detections of models learned for other words. Two of the models associate the distinctive pads of the Calgary Flames goalie Mikka Kiprusoff with the labels 'mikka' and 'kiprusoff'. Most of the other high-confidence model–word associations link the appearance of a team logo with other words associated with the team. For instance, Figure 10(b) shows a learned association between the Islanders logo and words related to their home arena of "Nassau Coliseum in Uniondale, New York". Other associations are with fragments of a team's name that are not among the words we manually linked to identify a team. For instance, the system learned associations between the words *wing* and *los* and the logos of the Detroit Red Wings and the Los Angeles Kings. We associate the words *angeles* and *kings* with the same caption token, so that either word could be used to represent the team. The word *los* was not one of the words we linked, but it still has a strong correspondence in our training set with the Los Angeles Kings logo. The fact that the components of a multi-word name can independently converge on similar visual descriptions means that it may be possible to automatically learn such compound names based on visual model similarity, instead of manually linking the components prior to learning.

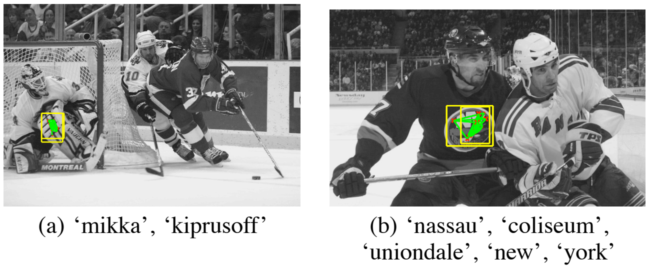

(a) 'mikka', 'kiprusoff'
(b) 'nassau', 'coliseum', 'uniondale', 'new', 'york'

Fig. 10. Words other than team names for which the system discovered strong models include the name of the Calgary Flames' goalie, the name and location of the New York Islanders' arena.

| Name | Precision | Recall | Frequency |
|---|---|---|---|
| ‘york’ | 1.00 | 0.22 | 81 |
| ‘new’ | 0.95 | 0.16 | 120 |
| ‘los’ | 0.92 | 0.34 | 32 |
| ‘maple’ | 0.86 | 0.16 | 37 |
| ‘uniondale’ | 0.85 | 0.31 | 35 |
| ‘coliseum’ | 0.85 | 0.20 | 35 |
| ‘nassau’ | 0.83 | 0.29 | 35 |
| ‘wings’ | 0.73 | 0.17 | 47 |
| ‘rutherford’ | 0.70 | 0.33 | 21 |
| ‘canada’ | 0.47 | 0.16 | 20 |
| ‘california’ | 0.40 | 0.20 | 20 |

TABLE III

PRECISION AND RECALL VALUES FOR 11 WORDS WITH AT LEAST 10 DETECTIONS IN THE TEST IMAGES; $\Gamma = 10, T_a = 0.95$.

Table III gives precision and recall values for all words (other than the team names) with at least 10 detections in the test images. These results may be somewhat understated, as the test image captions are not as consistent in mentioning these background words as they are with the team names. For instance, a detection for the word *maple* of the Toronto Maple Leafs logo will count as a true positive if the caption of the test image is “Maple Leafs vs Lightning”, but count as a false positive if the caption is the semantically equivalent “Toronto vs Tampa Bay”.

*4) Annotation Performance on the* LANDMARK *Data Set:* The LANDMARK data set includes images of 27 famous buildings and locations with some associated tags downloaded from the Flickr website, and randomly divided into 2172 training and 1086 test image–caption pairs. Like the NHL logos, each landmark appears in a variety of perspectives, scales and (to a lesser extent) orientations. Whereas a hockey logo is deformable and may appear in several different versions, the appearance of the landmarks can be strongly affected by varying lighting conditions, and different faces of the structure can present dramatically different appearances. Another difference is that the HOCKEY captions usually mention two teams, but each LANDMARK image is only associated with a single landmark.

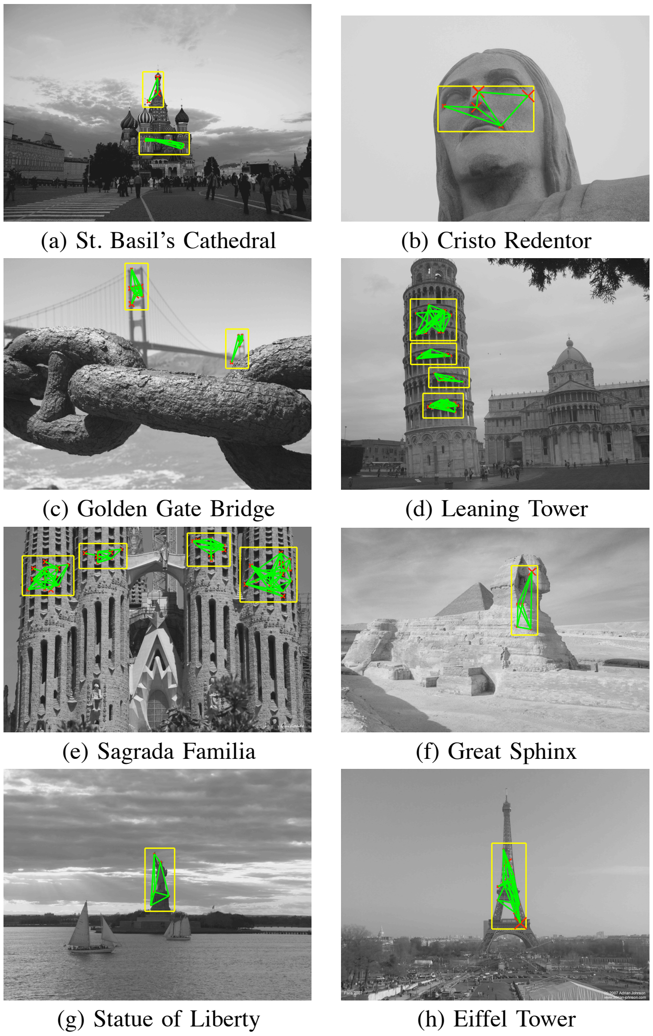
Table IV presents precision and recall information for the 26 of 27 landmarks for which high-confidence detections were made in the test set. Compared to the HOCKEY results (Table II), recall is generally higher. This is probably because test image labels used as ground truth are somewhat more reliable in the LANDMARK set; there are fewer instances where a test caption mentions a landmark that is not present in the image. The only landmark of the set that was not learned to some degree was the Sydney Opera House, perhaps due to the smooth, relatively textureless exterior of the building. Our system recognized only one of the CN Tower images in the test set, probably because the more distinctive pattern of the observation level makes up such a small fraction of the overall tower.

Figure 11 shows 8 examples of the detected landmarks. In general, the learned models tend to cover highly textured areas of a landmark. In some cases, however, the profile of a structure such as the towers of the Golden Gate Bridge or the Statue of Liberty is distinctive enough to form a strong model. Since many of the building details display different shadowing at different times of the day, multiple models are often learned to cover these different appearances. Due to the repetitive surface structure of many of the landmarks, an image often contains multiple detections of a single appearance model (*e.g.*, Figure 11(c-e)). The system also learned associations between landmark names and some nearby objects, such as an adjacent office tower that the system associated with the Empire State building.

| Name | Precision | Recall | Frequency |
|---|---|---|---|
| Rushmore | 1.00 | 0.71 | 35 |
| St. Basil’s Cathedral | 1.00 | 0.69 | 35 |
| Statue of Liberty | 1.00 | 0.61 | 36 |
| Great Sphinx | 1.00 | 0.45 | 40 |
| Notre Dame Cathedral | 1.00 | 0.40 | 40 |
| Stonehenge | 1.00 | 0.36 | 42 |
| St. Peter’s Basilica | 1.00 | 0.15 | 41 |
| Chichen Itza | 1.00 | 0.05 | 37 |
| CN Tower | 1.00 | 0.03 | 34 |
| Golden Gate Bridge | 0.97 | 0.73 | 45 |
| Christo Redentor | 0.96 | 0.55 | 44 |
| Eiffel Tower | 0.95 | 0.61 | 33 |
| Taj Mahal | 0.89 | 0.52 | 33 |
| Big Ben | 0.88 | 0.68 | 44 |
| Colosseum | 0.87 | 0.33 | 39 |
| Tower Bridge | 0.82 | 0.79 | 47 |
| White House | 0.81 | 0.38 | 45 |
| US Capitol | 0.80 | 0.80 | 45 |
| Reichstag | 0.80 | 0.53 | 45 |
| St. Paul’s Cathedral | 0.75 | 0.69 | 48 |
| Arc De Triomphe | 0.71 | 0.57 | 42 |
| Parthenon | 0.71 | 0.29 | 35 |
| Burj Al Arab | 0.71 | 0.23 | 43 |
| Leaning Tower | 0.70 | 0.93 | 43 |
| Empire State Building | 0.62 | 0.55 | 38 |
| Sagrada Familia | 0.54 | 0.40 | 35 |

TABLE IV

INDIVIDUAL PRECISION AND RECALL VALUES FOR THE 26 OF 27 LANDMARKS DETECTED WITH HIGH CONFIDENCE IN THE LANDMARK TEST SET; $\Gamma = 10$ AND $T_a = 0.95$.



(a) St. Basil’s Cathedral

(b) Cristo Redentor

(c) Golden Gate Bridge

(d) Leaning Tower

(e) Sagrada Familia

(f) Great Sphinx

(g) Statue of Liberty

(h) Eiffel Tower

Fig. 11. Sample annotations of our system in the landmark test images. Multiple detections of a single model within an image can be seen in (c-e).

### C. Summary of Results

The results presented in this section show that our system is able to find many meaningful correspondences between word labels and visual structures in three very different data sets. While the system can work equally well over a range of spatial parameter settings, the addition of spatial relationships to the appearance model can dramatically improve the strength and number of the learned correspondences. Our framework can deal with occlusion and the highly variable appearance of some objects by associating multiple visual models with a single word, but this depends on the presence of a sufficient number and a variety of training examples.

## V. Conclusions

We have proposed an unsupervised method that uses language both to discover salient objects and to build distinctive appearance models for them, from cluttered images paired with noisy captions. The algorithm simultaneously learns appropriate names for these object models from the captions. Note that we do not incorporate any prior knowledge into our system about which objects or words are of interest. The system has to learn these from the pairings of the images and captions in the training data. We have devised a novel appearance model that captures the common structure among instances of an object by using pairs of points together with their spatial relationships as the basic distinctive portions of an object. We have also introduced a novel detection method that can be reliably used to find and annotate new instances of the learned object models in previously unseen (and uncaptioned) test images. Given the abundance of existing images paired with text descriptions, such methods can be very useful for the automatic annotation of new or uncaptioned images, and hence can help in the organization of image databases, as well as in content-based image search.

At this juncture, there are two complimentary directions for future work. First, we would like to use insights from computational linguistics to move beyond individual words to groups of words (*e.g.*, word combinations such as compound nouns and collocations, or semantically-related words) that correspond to distinct visual patterns. Second, though local features provide a strong basis for detecting unique objects, they are less ideal for detecting object categories or general settings. However, the grouping problem persists for most types of visual features and most forms of annotation. We expect that many of the mechanisms for addressing the grouping problem for local features will also apply to other feature classes, such as contours.

## Appendix

This section elaborates on how we set the values of various parameters required by our model and methods, including: parameters of our image and object representation, and those of our detection confidence score from Section II, and parameters of the model–word correspondence confidence score (used in learning) from Section III.

For our image representation presented in Section II-A, we use a cluster set of size 4000 to generate the quantized descriptors $c_m$ associated with each image point. We set the neighborhood size, $|\mathbf{n}_m|$, to 50 as experiments on the CVPR data set indicated this was an appropriate tradeoff between having distinctiveness and locality in a neighborhood. (Neighborhoods in the range of $40-70$ points produced roughly equivalent results.) Recall from Section II-B that in an appearance model, each vertex is associated with a vector of neighboring cluster centers, $\mathbf{c}_i$. We set $|\mathbf{c}_i| = 20$ to minimize the chance of missing a matching feature due to quantization noise, at the expense of slowing down the model detection function.

We set the parameters of the word–appearance correspondence measure, $\mathrm{Corr}(w, G)$, according to the following assumptions: Reflecting high confidence in the captions of training images, we set $P(r_{wi} = 1|s_i = 1) = 0.95$, since we expect to observe an object's name in a caption whenever the object is present in the corresponding image. Similarly, we expect that few images will be labelled $w$ when the object is not present. However, even setting a low background generation rate ($P(r_{wi} = 1|s_i = 0) = 0.5$) could lead to a relatively large number of false $w$ labels if few image–caption pairs contain the object ($P(s_i = 1)$ is low). We expect the false labels to be a small *fraction* of the true labels. Since the expected rate of true word labels is $P(r_{wi} = 1|s_i = 1)P(s_i = 1)$ and the expected rate of false word labels is $P(r_{wi} = 1|s_i = 0)P(s_i = 0)$, we therefore set:

$$P(r_{wi} = 1|s_i = 0) = 0.05 \cdot \frac{P(r_{wi} = 1|s_i = 1)P(s_i = 1)}{P(s_i = 0)}. \quad (17)$$

Similarly, during model learning, we wish to assert that a high-confidence model should have a low ratio of false-positives to true-positives. Therefore we set:

$$P(q_{Gi} = 1|s_i = 0) = 0.05 \cdot \frac{P(q_{Gi} = 1|s_i = 1)P(s_i = 1)}{P(s_i = 0)} \quad (18)$$

However, when evaluating correspondences between words and neighborhoods to find good seed models, we set this target number of false-positives as a fraction of the true positives much higher at 1. We do so because the starting seed models are expected to be less distinctive, hence even promising seeds may have a large number of false positives. Performing experiments on a held-out validation set of 500 HOCKEY training images, we found the precise value of $P(H_0)/P(H_C)$ to have little effect on the overall performance; we thus set $P(H_0)/P(H_C) = 100,000$.

At the initialization stage, up to 20 neighborhood clusters $\mathcal{N}$ are selected to generate seed models, which are further modified in up to 200 stages of iterative improvement.

If the detection threshold $T_d$ is set too low, the detection algorithm can report a large number of low-probability detections, which can impose a computation burden on later stages of processing. However, the threshold should not be set so high that detections that might lead to model improvements are ignored. We err on the side of caution and set $T_d = 0.001$. The parameters of the background likelihood (used in measuring detection confidence) are set according to sample statistics of a collection of 10,000 commercial stock photographs. Based on this wide sample of visual content, the distance variance $\sigma^2_{xB}$ is set to 200, and the scale variance $\sigma^2_{\lambda B}$ is set to 5. Each of the model spatial relationship variances is set to a fraction $1/\Gamma$ of the corresponding background variance, *i.e.*:

$$\sigma^2_x = \frac{\sigma^2_{xB}}{\Gamma},\ \sigma^2_\lambda = \frac{\sigma^2_{\lambda B}}{\Gamma},\ \text{and}\ \sigma^2_\phi = \frac{\pi^2}{3\Gamma}.$$

where $\Gamma$ is the spatial tolerance parameter that determines the expected degree of spatial variability among observed object models. In most experiments, we set $\Gamma = 10$, a value we found to result in reasonably good performance on the HOCKEY validation set. After estimating the mixture of Gaussian distribution based on the stock photo collection, we observed that average variance among the background feature clusters was 1.31, so we set the model feature variance to be slightly higher at $\sigma_f^2 = 1.5$. Based on empirical findings on the held-out subset of the HOCKEY training data, we set $P(H_B)/P(H_G) = 100,000$, though we found a broad range of values to yield similar results.

## Acknowledgment

The authors gratefully acknowledge the support of Idée Inc., CITO and NSERC Canada.

## References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[2] K. Barnard, P. Duygulu, R. Guru, P. Gabbur, and D. Forsyth. The effects of segmentation and feature choice in a translation model of object recognition. In *CVPR*, 2003.

[3] K. Barnard and Q. Fan. Reducing correspondence ambiguity in loosely labeled training data. In *CVPR*, 2007.

[4] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y.-W. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *CVPR*, 2004.

[5] T. L. Berg and D. A. Forsyth. Animals on the web. In *CVPR*, 2006.

[6] D. Blei and M. Jordan. Modeling annotated data. In *SIGIR*, 2003.

[7] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *ECCV*, 2004.

[8] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):394–410, 2007.

[9] G. Carneiro and A. Jepson. Flexible spatial configuration of local image features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2089–2104, 2007.

[10] M. L. Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the World Wide Web. In *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries*, June 1998.

[11] D. J. Crandall and D. P. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *ECCV*, 2006.

[12] 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*, volume 4, pages 97–112, 2002.

[13] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *CVPR*, 2004.

[14] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In *CVPR*, 2005.

[15] M. Jamieson, S. Dickinson, S. Stevenson, and S. Wachsmuth. Using language to drive the perceptual grouping of local image features. In *CVPR*, 2006.

[16] M. Jamieson, A. Fazly, S. Dickinson, S. Stevenson, and S. Wachsmuth. Learning structured appearance models from captioned images of cluttered scenes. In *ICCV*, 2007.

[17] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR*, 2003.

[18] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR*, 2004.

[19] J. Li and J. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(9):1075–1088, 2003.

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

[21] F. Monay and D. Gatica-Perez. Modeling semantic aspects for cross-media image indexing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(10):1802–1817, 2007.

[22] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[23] A. Quattoni, M. Collins, and T. Darrell. Learning visual representations using images with captions. In *CVPR*, 2007.

[24] I. Simon, N. Snavely, and S. M. Seitz. Scene summarization for online image collections. In *ICCV*, 2007.

[25] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *CVPR*, 2004.

**Michael Jamieson** Michael Jamieson received the B.A.Sc. and M.A.Sc. degrees in Systems Design Engineering from the University of Waterloo in 1999 and 2003 respectively. From 1999 to 2000 he helped design radar countermeasures at Vantage Point International and since 2003 he has developed visual search algorithms at Idée Inc. He is currently a Ph.D candidate in the Computer Science department at the University of Toronto. His research interests include visual search and automatic image annotation.

**Afsaneh Fazly** Afsaneh Fazly received bachelor's and master's degrees in Computer Engineering from Sharif University of Technology (Iran), and master's and Ph.D. degrees in Computer Science from the University of Toronto. She is currently a postdoctoral fellow at the University of Toronto. Dr. Fazly's primary area of research is computational linguistics and cognitive science, with an interest in approaches that integrate these disciplines with others in artificial intelligence, such as computational vision.

**Suzanne Stevenson** Suzanne Stevenson received a B.A. in Computer Science and Linguistics from William and Mary, and M.S. and Ph.D. degrees from the University of Maryland, College Park. From 1995-2000, she was on the faculty at Rutgers University, holding joint appointments in the Department of Computer Science and in the Rutgers Center for Cognitive Science. She is now at the University of Toronto, where she is Associate Professor of Computer Science, and Vice-Dean, Students, Faculty of Arts and Science.

**Sven Dickinson** Sven Dickinson received the B.A.Sc. degree in systems design engineering from the University of Waterloo, in 1983, and the M.S. and Ph.D. degrees in computer science from the University of Maryland, in 1988 and 1991, respectively. He is currently Professor of Computer Science at the University of Toronto's Department of Computer Science. His research interests include object recognition, shape representation, object tracking, vision-based navigation, content-based image retrieval, and language-vision integration.

**Sven Wachsmuth** Sven Wachsmuth received the Ph.D. degree in computer science from Bielefeld University, Germany, in 2001. He now holds a staff position there as a senior researcher in Applied Informatics. In 2008 he joined the Cluster of Excellence Cognitive Interaction Technology at Bielefeld University and is heading the Central Lab Facilities. His research interests include cognitive and perceptive systems, human-machine communication, and the interaction of speech and visual processing.