Learning Structured Appearance Models from Captioned Images of Cluttered Scenes *

Michael Jamieson[1] Afsaneh Fazly[1] Sven Dickinson[1] Suzanne Stevenson[1] Sven Wachsmuth[2]
[1]University of Toronto [2]Bielefeld University
{jamieson, afsaneh, sven, suzanne}@cs.toronto.edu swachsmu@techfak.uni-bielefeld.de

Abstract

Given an unstructured collection of captioned images of cluttered scenes featuring a variety of objects, our goal is to learn both the names and appearances of the objects. Only a small number of local features within any given image are associated with a particular caption word. We describe a connected graph appearance model where vertices represent local features and edges encode spatial relationships. We use the repetition of feature neighborhoods across training images and a measure of correspondence with caption words to guide the search for meaningful feature configurations. We demonstrate improved results on a dataset to which an unstructured object model was previously applied. We also apply the new method to a more challenging collection of captioned images from the web, detecting and annotating objects within highly cluttered realistic scenes.

1. Introduction

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.

Some of our previous work has shown that correspondences between image features and caption words can be used to guide the search for meaningful feature configurations [14, 10]. Building on this idea, we introduce a new word–appearance correspondence measure that uses language cues in addition to recurring visual patterns to incrementally construct strong object appearance models. In contrast to [14], our appearance models are composed of easily-extractable local features and can therefore detect exemplar objects in highly cluttered real-world images. Our new models capture more of the structured appearance of an object than the simple 'bag of features' models used in [10] while remaining robust to changes in scale and orientation, as well as to minor deformations. We demonstrate improved results on the set of images used in [10] and also discover meaningful word–appearance pairs in a larger and more challenging set of captioned hockey images.

2. Related Work

Work on learning relationships between text captions and image features to support automatic image annotation includes a variety of proposals [1, 4, 5, 7, 8]. 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 [4, 8] or simple shape information [7]. This type of representation

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

is less reliant on perceptual grouping than a shape or structured appearance model because color and texture are relatively robust to segmentation errors and the configuration of features is not critical. However, they tend to be less effective for objects that lack a consistent color or texture yet *do* have a characteristic shape or structured appearance. Other work avoids the perceptual grouping problem by focusing on a domain where there exists detailed prior knowledge of the appearance of the objects of interest, as in the task of matching names with faces [2].

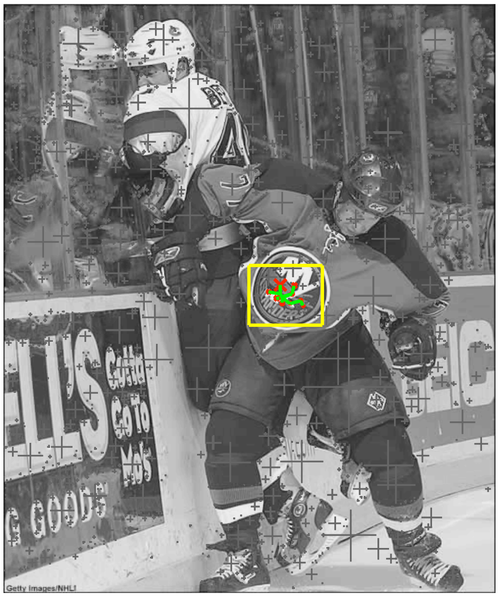
In contrast, Wachsmuth *et al*. [14] specifically target object classes defined by an unknown shape, so the perceptual grouping problem becomes central. Language information from a translation model, as in [7], is used to guide the combination of small image segments into identifiable shape categories. However, this shape model extraction has not yet been demonstrated on real images.

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*. [9] learn object appearance models consisting of a distinctive subset of local interest features and their relative positions. Crandall and Huttenlocher [6] use graph models in which vertices are oriented edge templates rather than local feature detections. These methods can learn an appearance model from very noisy training images; however, unlike most automatic annotation work, they are not designed for images containing multiple objects and multiple annotation words. In the domain of image annotation, our previous work [10] *can* handle such unstructured training sets, but uses a 'bag of features' appearance model with only a weak spatial proximity constraint.

In this paper, we encode the appearance of an exemplar object through a set of local features connected by robust pairwise spatial relationships. This model is less prone than [10] to false-positive detections and is invariant to scale and rotation. Moreover, whereas the method described in [10] requires a relatively distinctive singleton seed feature for initialization of the object models, our new approach begins with a group of features, none of which has to be individually distinctive. Finally, while both [10] and [14] use a translation model between words and image features to guide grouping of the features into object models, our current method uses a more efficient word–feature correspondence metric that enables us to more effectively explore the space of configurations.

3. Learning to Annotate Exemplar Objects

The goal of this work is to annotate exemplar objects appearing in images of cluttered scenes. A typical such image, with more than a thousand 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 of [10] and [14] is that this evidence can guide not only the annotation of complex feature configurations, but also the search for meaningful configurations themselves.



New York **Islanders**' defenseman Alexei Zhitnik mashes Vancouver Canucks' right wing Todd Bertuzzi into the glass.

Figure 1. From a set of image–caption pairs, each containing hundreds of local features (gray crosses), our algorithm has discovered an association between a team name (shown in red) and its logo (red features and green relationships in a yellow box).

Accordingly, we look for recurring configurations of features that also strongly co-occur with certain words, hence simultaneously finding objects and annotating them. To illustrate, Figure 1 shows a sample image containing hundreds of local features, paired with a caption containing many irrelevant words. Our algorithm has learned both an appearance model (here representing the team's *logo*), as well as its association with a caption word (here the team name *Islanders*). To begin, the learning algorithm finds a set of simple recurring object models, evaluates how strongly they correspond to each caption word, and accordingly chooses a set of good 'seed' appearance models for each word. Each seed model is then iteratively expanded (if possible) into a more reliable object detector, guided at each step by the change in its strength of association with the corresponding caption word.

We use a learning framework that 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 use local interest features to represent small patches of appearance and use the pairwise spatial relationships between the patches to construct a connected graph model for the object. 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 described in Section 4.

The initial stage of our learning algorithm provides a set of seed appearance models to use as starting points for the detection of objects mentioned in the captions. 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. At the same time, trying all possible configurations of even a small number of features as seeds is impractical. Instead, our initialization method generates structured seed models by looking at recurring neighborhoods of features that also co-occur with particular caption words.

The initial seed models are fed into an iterative improvement stage, which expands them into appearance models that cover a larger portion of the object. In previous work, the guidance for the iterative improvement of an initial model is provided through a probabilistic translation method [10, 14]. However, it is expensive to relearn all of the translation probabilities every time a new configuration of features is formed. Here, we use a simpler and more efficient measure of correspondence between a caption word and an appearance model. The measure reflects the amount of evidence, available in a set of training images, that the word and the model are generated from a common underlying source object. Section 5 elaborates on the correspondence measure, as well as the initial and the iterative improvement stages that draw on this measure.

4. Image and Object Representations

4.1. Image Representation

We represent an image as a set I of interest points p_m, *i.e.*, $I = \{p_m | m = 1 \dots |I|\}$. These points are detected using Lowe's SIFT method [12], which defines a point in terms of its Cartesian position $\mathbf{x}_m$, scale σ_m and orientation θ_m. In addition to these, for each interest 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. For this, we use the PCA-SIFT feature encoding developed in [11] 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.

In addition to the continuous feature vector $\mathbf{f}_m$, we also use a quantized descriptor c_m for each image point, in order to quickly scan for potentially matching features. Following [13], we use K-means to generate a set of cluster centers from a set of features randomly selected from our training images. The index of the cluster center closest to $\mathbf{f}_m$ is the descriptor c_m associated with p_m.

Each point p_m is also associated with a neighborhood $\mathbf{n}_m$ that is the set of spatial neighbors of the point. The distance measure used to construct a neighborhood is:

$$\Delta x_{ij} = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\sqrt{\sigma_i^2 + \sigma_j^2}} \tag{1}$$

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 the object increases.

To summarize, each point p_m in an image representation I is a 6-tuple of the form $(\mathbf{f}_m, \mathbf{x}_m, \sigma_m, \theta_m, c_m, \mathbf{n}_m)$.

4.2. Appearance Model Representation

We represent an appearance model using a graph $G = (V, E)$. Each vertex $v_i = (\mathbf{f}_i, \mathbf{c}_i)$ is composed of a continuous feature vector $\mathbf{f}_i$, and a cluster index vector $\mathbf{c}_i$ containing indexes for the $|\mathbf{c}_i|$ nearest cluster centers to $\mathbf{f}_i$. 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 because the appearance model must be invariant to scale, translation and rotation. Instead, each edge in G encodes a spatial relationship between vertices in four parts: $e_{ij} = (\Delta x_{ij}, \Delta\sigma_{ij}, \Delta\phi_{ij}, \Delta\phi_{ji})$, where Δx_{ij} is the relative distance between v_i and v_j, $\Delta\sigma_{ij}$ is the relative scale difference between them, $\Delta\phi_{ij}$ is the relative heading from v_i to v_j, and $\Delta\phi_{ji}$ is the heading in the opposite direction. These relationships are taken from Carniero and Jepson [3], and are calculated as in (1) above, and (2) and (3) below:

$$\Delta\sigma_{ij} = \frac{\sigma_i - \sigma_j}{\sqrt{\sigma_i^2 + \sigma_j^2}} \quad (2)$$

$$\Delta\phi_{ij} = \Delta_\theta(\tan^{-1}(\mathbf{x}_i - \mathbf{x}_j) - \theta_i) \quad (3)$$

where $\Delta_\theta(.) \in [-\pi, +\pi]$ denotes the principle angle.

An observed instance of a model 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 local interest points 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. Deformations, noise and changes in perspective can distort the features encoded at local interest points and the spatial relationships between them. Also, a model may be only partially observed, with vertices occluded or lost due to inconsistent detection of points.

We introduce an energy function, $H(G, I, O)$, that measures how well the observed instance O in image representation I matches the object appearance model G. The function is defined so that observed configurations of points that closely fit the model have low energy, whereas configurations that are very different from the model have high energy. Details are given in Appendix A.

To detect an instance of an object model, we need to find a low-energy association of its vertices to points in an image. Given that a typical image can contain thousands of interest points, determining the optimal associations is potentially quite expensive. We thus propose a greedy heuristic that efficiently searches the space of possible associations for a nearly-optimal solution. Since individual vertices of a model may be unobserved, our detection heuristic 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 model edge with at least one other observed vertex. Also, only those observations O with $H(G, I, O)$ below a learned model-specific threshold, Δ_H, are considered valid instances for annotation. The detection algorithm is given in Appendix B.

5. Using Words to Learn Appearance Models

Here, we propose an unsupervised learning algorithm that constructs 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 the caption contains 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.

5.1. A Measure of Word–Model Correspondence

We seek 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 appearance model are present in an image because the *object* is present. We thus define and use a measure of confidence that a given appearance model is a reliable detector for the object referred to by a word.

Consider a set of k captioned images. The occurrence pattern of a word w in the captions of these images may be represented as a binary vector $\mathbf{r}_w = \{r_{wi} | i = 1, \ldots, k\}$. Similarly, we can represent the occurrence of a model G with another binary vector, $\mathbf{q}_G = \{q_{Gi} | i = 1, \ldots, k\}$. It is always possible that the two patterns of occurrence are independent (the null hypothesis or H_0). Alternatively, they may have been derived 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. We define the correspondence between w and G as the log-likelihood ratio of generating the observed patterns, $\mathbf{r}_w$ and $\mathbf{q}_G$, under H_C and H_0:

$$\mathrm{Corr}(w, G) = \log \frac{P(\mathbf{r}_w, \mathbf{q}_G | H_C)}{P(\mathbf{r}_w, \mathbf{q}_G | H_0)} \quad (4)$$

$$P(\mathbf{r}_w, \mathbf{q}_G | H_C) = \prod_i \sum_{s_i} P(s_i) P(r_{wi} | s_i) P(q_{Gi} | s_i) \quad (5)$$

$$P(\mathbf{r}_w, \mathbf{q}_G | H_0) = \prod_i P(r_{wi}) P(q_{Gi}) \quad (6)$$

where $s_i \in \{0, 1\}$ represents the presence of the common source in image–caption pair i.

$\mathrm{Corr}(w, G)$ reflects the degree to which the common-source hypothesis explains the observed patterns of occurrence for a word w and a model G. To get the likelihood of observed data under H_C, we 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 in Section 6.1. $P(s_i)$ 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 simply the observed probabilities for word and model occurrence.

5.2. Model Initialization

The goal of the model initialization stage is to quickly find 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.

Even searching the space of small models for good starting points is not a trivial task. We thus look for good seed models using the more tractable neighborhood patterns introduced by Sivic and Zisserman [13]. A neighborhood pattern $\mathbf{p}_m$ is a vector containing the quantized descriptors c of a point p_m and all of its neighbors $\mathbf{n}_m$. We use the clustering method described in [13] to find commonly recurring neighborhoods across the training images.

To find distinctive starting points that also correspond to named objects, the initialization module finds associations between a caption word w and a cluster of similar neighborhoods $\mathcal{N} = \{\mathbf{p}_m\}$, using $\mathrm{Corr}(w, \mathcal{N})$ in Equation 4 above.[1] Neighborhoods within a cluster $\mathcal{N}$ that strongly corresponds to a word w are likely to spatially overlap the object referred to by w. We thus construct seed appearance models potentially corresponding to w, using recurring points and their spatial relationships within $\mathcal{N}$. Since the simplest detectable appearance model is a single pair of vertices, we identify neighboring pairs of points with consistent spatial relationships that have appeared in the elements of $\mathcal{N}$ in more than one distinct image. The pair with the lowest summed energy (Equation 7 in Appendix A) across all detections is adopted as a seed appearance model.

5.3. Iterative Improvement

The improvement stage iteratively makes changes to a seed object model, guided by the correspondence between caption words and models. More specifically, the improvement algorithm starts with a seed model for a given word, makes a simple modification to this model (e.g., adds a new vertex), and detects instances of the new model in the training images. The new model is accepted as a better object detector under either of two conditions: it has a stronger correspondence with the word (according to Equation 4), or it has the same correspondence, but a lower total detected energy (according to Equation 7).

On alternating iterations, the algorithm randomly tries to either adjust the energy threshold, or expand the model by adding a vertex or an edge. For the former, the threshold is increased (decreased) by a small percentage ($\pm 25\%$), in order to leave out false detections or bring in true instances. When expanding the model, candidates for addition include: (i) points that consistently appear in the neighborhoods of the currently-detected vertices—to be added as new vertices; and (ii) consistently appearing pairwise spatial relationships between the currently-detected vertices—to be added as new edges.

The algorithm first tries the addition that would result in the greatest decrease in total energy over the currently detected instances. This gives priority to neighboring points that are more common and have the most consistent spatial relationships with detected model vertices. Since adding a vertex to the model also adds an edge, it is often associated with a greater decrease in energy than adding a new internal edge. Consequently, the algorithm tends to favor expanding the scope of the model over adding spatial constraints. Nonetheless, if a relationship between vertices consistently appears in many detected instances, it will have priority over a point that appears in few instances.

The initial two-vertex model often is not large enough to encode the visual structure of the initial neighborhood cluster $\mathcal{N}$ and so may occur in many other distracting contexts. In order to give the model a chance to grow to better-characterize its seed context, the first few iterations of the improvement stage are confined to search for the model in the starting cluster of neighborhoods, $\mathcal{N}$, rather than the complete training set.

6. Using Models to Annotate New Instances

For evaluation, we use the models our algorithm has learned from training image–caption pairs to detect and annotate new instances of objects in previously unseen (and uncaptioned) test images. For detection of new instances, we use the same algorithm we use in learning (Appendix B). To annotate a detected instance of an object, we use the word associated with the learned object model.

In Section 6.1 we explain our choices for the parameters of our algorithms. We then compare results with our previous work [10] by looking into the performance of our learning and annotation methods on a data set of 228 real images of toys in Section 6.2. Finally, we present the results of applying our methods to a larger and more challenging set of 1240 real-world images from the web in Section 6.3.

6.1. Parameter Settings

For our image representation described in Section 4.1, 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 since this was empirically found to be an appropriate tradeoff between having distinctiveness and locality in a neighborhood. Recall from Section 4.2 that in an appearance model, each vertex is associated with a vector of neighboring cluster centers, $\mathbf{c}_i$; we set $|\mathbf{c}_i| = 10$ to minimize the chance of missing a matching feature due to quantization noise.

[1] For this, we replace $\mathbf{q}_G$ in (4) with $\mathbf{q}_{\mathcal{N}}$ indicating the occurrences of $\mathcal{N}$ in the training images.

We set the parameters of the word–model correspondence measure, $\mathrm{Corr}(w, G)$, according to the following assumptions: Reflecting high confidence in the captions of training images, $P(r_{wi} = 1|s_i = 1) = 0.95$ and $P(r_{wi} = 1|s_i = 0) = 0.05$. During model learning, $P(q_{Gi} = 1|s_i = 0) = 0.01$ in order to decrease the likelihood of false detections. When evaluating correspondences between words and neighborhoods to find good seed models, we set this target false-positive rate higher at 0.05, as we do not want to overlook potentially-strong seed models.

At the initialization stage, all word–model pairs with $\mathrm{Corr}(w, G) > 0$ are selected as seed models, further modified in up to 100 stages of iterative improvement. For annotation of new instances, we only use learned models G that are considered reliable according to our correspondence measure ($\mathrm{Corr}(w, G) \geq 10$ for some word w). Since we consider precision more important than recall for annotation, we set the energy threshold during annotation to twice the threshold learned during training. The value of this threshold could be determined by a user, depending on whether they desire more detections (higher recall) or fewer detections with higher confidence (higher precision).

6.2. Experiments on Toy Images

We use the 128 training and 100 test images described in [10]. Each image in this data set contains 3 or 4 toy objects (out of a pool of 10), arranged in different poses, and partially occluded in many cases. Each training image is paired with a manually-generated caption that contains names of all objects in the image plus a few distractor names.

Figure 2 shows the precision–recall curves of both the new detection method and that of [10] on the test images. Our new method consistently demonstrates substantially higher annotation precision for equivalent recall; in addition, we learn strong models for 3 of the objects for which the original method of [10] was unsuccessful. These results confirm our hypothesis that we can build more distinctive object models by finding recurring spatial relationships among the recurring image points. Improved performance is also due to the choice of better starting points for model construction. Moreover, our simpler correspondence measure speeds up the training phase by a factor of 10 ($\sim$ 1–2 hours for the new method vs. $\sim$22 hours for that of [10]).

6.3. Experiments on Web Data

The web data set includes images of National Hockey League (NHL) players and games, with associated captions, downloaded from various web sites, and randomly divided into 850 training and 390 test image–caption pairs. About a third of the captions are full sentence descriptions (as in Figure 1, page 2), whereas the remainder simply name the two teams involved in the game (*e.g.*, 'Maple Leafs vs Senators'). Most images are on-ice shots and display multiple players in a variety of poses and scales.

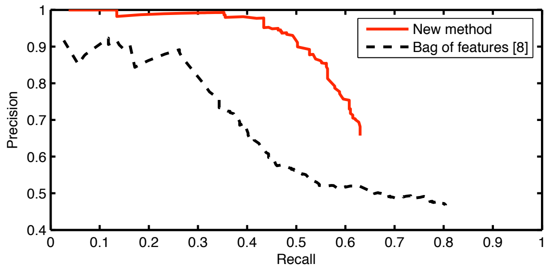


Figure 2. Precision–recall curves for our new detection method and our earlier bag-of-features approach [10].

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. Since NHL teams are referred to by both their team name (*e.g.*, 'Bruins') and their city name (*e.g.*, 'Boston'), we treat an occurrence of either as an instance of the team name. This link between team names and city names is the only prior knowledge available to our learning algorithm. The final vocabulary extracted from the training captions contains 105 words, of which only 30 are team designations. Note that the algorithm has only these caption words and the contents of the images to guide its search for meaningful word–appearance pairs.

From the training image–caption pairs, our algorithm learns one or more strong appearance models (i.e., those with Corr $\geq$ 10) for 9 team names (out of 30). The strength of learned models is highly influenced by the number of times the object appears in the training set. The 9 teams for which strong models are learned are on average mentioned in 108 captions, whereas the other 21 teams are on average mentioned in 34 captions. In addition, a team does not have to be visible in an image in order to be mentioned in the caption. Thus, in most cases, fewer instances of an object are accessible to the learning algorithm. For instance, the team name Vancouver Canucks is mentioned in 52 training captions but players only appear in 20 images and the logo is only visible in 14 of these.

Of the 9 teams with strong learned models, 8 were detected in the test images. Figure 3 shows a test image with a detection of a learned appearance model associated with the Toronto Maple Leafs. Figure 4 shows several other detections of learned appearance models (and their associated team names) on test images. Table 1 gives the precision and recall of detections of each of the 8 teams, calculated based on whether the model predicted the presence of the correct word in the corresponding test caption. (Note that we use the test captions only for evaluation.) Test image annotation generally has high precision but low recall. This reflects the fact that teams mentioned in the captions are not always visible and that a hockey player has a highly variable

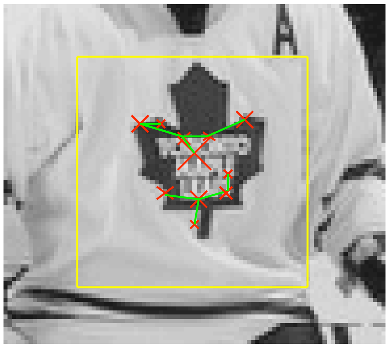
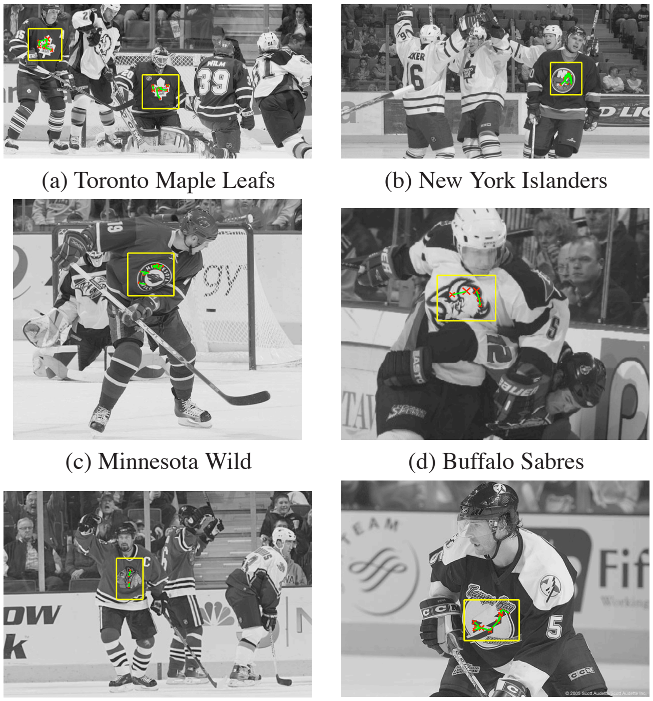


Figure 3. Detection of a model associated with the Toronto Maple Leafs. Observed vertices are in red; edges in green.



(a) Toronto Maple Leafs (b) New York Islanders

(c) Minnesota Wild (d) Buffalo Sabres

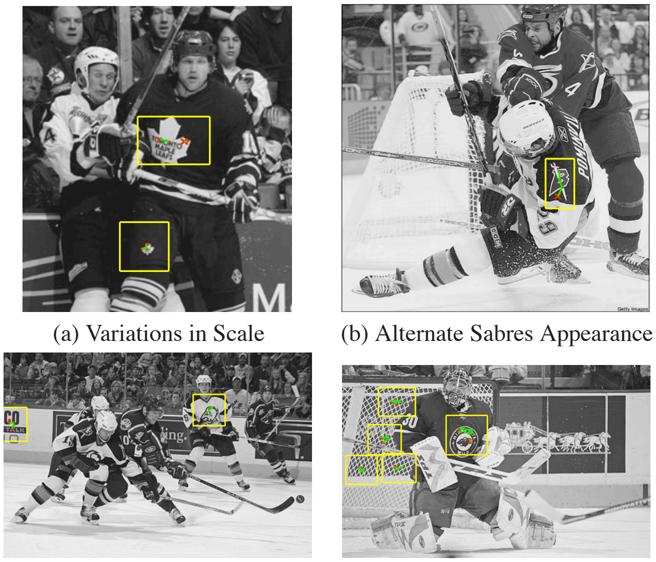
(e) Chicago Blackhawks (f) Tampa Bay Lightning

Figure 4. Sample detections of team logos in test images.

Name	Precision	Recall	Frequency
Tampa Bay Lightning	0.92	0.73	48
Maple Leafs	1.00	0.15	80
Minnesota Wild	1.00	0.22	51
New York Islanders	1.00	0.20	20
Buffalo Sabres	1.00	0.14	22
Chicago Blackhawks	0.67	0.16	37
Dallas Stars	1.00	0.08	50
Ottawa Senators	1.00	0.05	20

Table 1. Precision and recall of detection for 8 team names in test images, and the number of test captions each name appears in.

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.



(a) Variations in Scale (b) Alternate Sabres Appearance

(c) Minnesota Wild Arena (left) (d) Detections of ‘vs’

Figure 5. Some interesting detections in test images.

Figure 5 illustrates several interesting types of model detections. Part (a) demonstrates that the learned appearance models can be detected across a wide range of scales. In (b), the detected Buffalo Sabres shoulder-patch model is an example of multiple distinct models being associated with a single word (cf. Figure 4(d)). The model detection on the left of (c) shows the appearance of an advertisement along the boards of the Minnesota Wild arena, learned as a second model for this team name. (d) shows a detection of the Minnesota Wild logo and several background detections associated with the word ‘vs’. Due to the probabilistic nature of our algorithm, it learns strong associations between background features in images, such as parts the net, and function words, such as *vs* and *the*. These types of detections can be easily avoided by ignoring words that occur frequently across many captions, which are less distinctive.

7. Conclusions

We have proposed an unsupervised method that uses language both to discover salient objects and to construct 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. 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.

References

[1] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *ICCV-01*. 1

[2] 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-04*. 2

[3] G. Carniero and A. Jepson. Flexible spatial models for grouping local image features. In *CVPR-04*. 3

[4] G. Carniero and N. Vasconcelos. Formulating semantic image annotation as a supervised learning problem. In *CVPR-05*. 1

[5] M. L. Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the World Wide Web. In *Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries*, 1998. 1

[6] D. J. Crandall and D. P. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *ECCV-06*. 2

[7] 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-02*. 1, 2

[8] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *CVPR-04*. 1

[9] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In *CVPR-05*. 2

[10] M. Jamieson, S. Dickinson, S. Stevenson, and S. Wachsmuth. Using language to drive the perceptual grouping of local image features. In *CVPR-06*. 1, 2, 3, 5, 6

[11] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR-04*. 3

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 3

[13] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *CVPR-04*. 3, 5

[14] S. Wachsmuth, S. Stevenson, and S. Dickinson. Towards a framework for learning structured shape models from text-annotated images. In *HLT-NAACL03 Workshop on Learning Word Meaning from Non-Linguistic Data*, 2003. 1, 2, 3

A. Energy Function

The energy function has two components, looking at vertices and edges, respectively:

$$H(G,I,O) = \sum_{(v_i,p_m)\in O} h_V(v_i,p_m) + \sum_{e_{ij}\text{ in }O} h_E(e_{ij},p_m,p_n) \tag{7}$$

where e_{ij} is in O if both $(v_i,p_m)\in O$ and $(v_j,p_n)\in O$.

$h_V(v_i,p_m)$ measures the fit between a model vertex v_i and the observed image point p_m, and is calculated as:

$$h_V(v_i,p_m) = \delta_V + \alpha_f \|\mathbf{f}_i - \mathbf{f}_m\|^2 \tag{8}$$

where δ_V is the maximum energy reward for observing a vertex, and α_f is the rate the reward decays with feature dissimilarity.

The component h_E measures how well spatial relationships between observed points fit expectations set by the model. It is calculated as:

$$\begin{aligned} h_E(e_{ij},p_m,p_n) \;=\; \delta_E \;&+\; \alpha_\sigma(\Delta\sigma_{ij} - \Delta\sigma_{mn})^2 \\ &+\; \alpha_x(\Delta x_{ij} - \Delta x_{mn})^2 \\ &+\; \alpha_\phi(\Delta\phi_{ij} - \Delta\phi_{mn})^2 \end{aligned} \tag{9}$$

where δ_E is the maximum energy reward for matching the expected spatial relationship between vertices, and α_σ, α_x and α_ϕ control the rates of reward decay. In our experiments, these parameters are set as follows: $\delta_V = -8$, $\delta_E = -5$, $\alpha_f = 0.16$, $\alpha_\sigma = 25$, $\alpha_x = 0.25$ and $\alpha_\phi = 2.5$. By experimenting with several pairs of training images, we determine thresholds for spatial and feature variation that capture most corresponding interest point pairs while excluding most false positives. The δ_V and δ_E parameters roughly reflect the log probability of two random point pairs falling within their respective thresholds while the α values are adjusted so that $h_V < 0$ and $h_E < 0$ within the allowed variational range.

B. Model Instance Detection Algorithm

Algorithm 1 uses a greedy heuristic to detect 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 observed image points.

Algorithm 1 Detects instances of G in I

```
FindModelInstance($G$,$I$)

1. Find the set of potential vertex–point associatiations $A = \{(v_i, p_m)\}$,
   where $c_m \in \mathbf{c}_i$ and $h_V(v_i, p_m) < 0$.
2. Find the set $L$ of potential links $((v_i, p_m), (v_j, p_n))$ between elements of
   $A$, where $p_n \in \mathbf{n}_m$ and $h_E(e_{ij}, p_m, p_n) < 0$.
3. Set the initial instance $O$ to the pair $\{(v_i, p_m), (v_j, p_n)\}$, such that the link
   $((v_i, p_m), (v_j, p_n)) \in L$ and $H(G, I, O)$ is minimum.
4. Remove $(v_i, p_m)$ from $A$ 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 $L$ if neither end is in $A$.
6. Let $A_{adj}$ be the subset of $A$ that shares an edge in $L$ with $O$.
7. If $A_{adj}$ contains associations that could decrease $H(G, I, O)$, add to $O$ the
   association with greatest decrease in $H$, and go to step 4.
8. Let $L_{neigh}$ be the subset of $L$ within the union of the neighborhoods of ob-
   served points in $O$.
9. If $L_{neigh}$ contains observed links that could decrease $H(G, I, O)$, add to $O$
   the pair of associations with the link that produces the greatest decrease in $H$,
   and go to step 4.
10. Return $O$.
```