Learning Object Category Shape from Captioned Images

by

Tom Sie Ho Lee

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

Learning Object Category Shape from Captioned Images

Tom Sie Ho Lee

Master of Science

Graduate Department of Computer Science

University of Toronto

2011

Given a collection of unsupervised captioned images of cluttered scenes, we learn shape models of object categories by finding image features that co-occur with words. Instead of relying on prior object localization (*e.g.*, bounding boxes), we use perceptual grouping cues of closure, continuity, and proximity to learn a parts-based model of spatially related contours from cluttered images. We implement a recently proposed framework that learns a graph model part-by-part subject to grouping constraints, and extend it with bottom-up segmentation cues for part initialization. We show that shape features are more effective than appearance features (*e.g.*, SIFT) at modelling object categories and present encouraging results on the ETHZ dataset.

Contents

Chapter 1

Introduction

Most object recognition methods are trained with labelled bounding boxes or segmentations to separate out background and other objects. A more realistic and scalable learning scenario would be to mine collections of captioned images for the visual appearances of object categories and their names without any manual supervision. Difficulty in such an endeavour, however, arises from the high level of ambiguity in cluttered images—the key to this problem is the repetition of visual features and caption language across multiple images, leading to a visual-linguistic correspondence, *e.g.*, between a word and a subset of image features. In this paper, we address the problem of learning *shape* representations of named object categories from captioned images without bounding boxes.

In merging shape recognition with captioned image data, we extend and apply a recent framework proposed by Jamieson *et al.* [11]. In this framework, an object is modelled as a graph over object parts (vertices) constrained by pairwise spatial relations (edges). Each part is a local invariant image feature, and detection is done by matching features under spatial constraints. Initially, a given word w is used to group those images likely to yield a consistent subset of image features. By using an a priori grouping cue of proximity between object parts, the learning algorithm efficiently finds a structured configuration of features that maximally co-occurs with the word w across a set of captioned images.

Whereas object categories have sometimes been modelled with distinctive appearance features (Fergus *et al*. [6], Crandall & Huttenlocher [4]) or grouped using such features (Lee & Grauman [13]), surface characteristics of colour and texture are generally not specific to categories. Studies in visual perception (Biederman & Ju [3]) have shown that household objects presented in full-colour photographs provide no advantage over line-drawings with respect to latency of recognition. Shape representations, which were common in early computer vision research, have re-emerged in the last few years in category recognition. Local invariant features that encode only image contours are available, and make possible a direct application of the framework of Jamieson *et al*. [11].

Our application of the framework to shape categories involves the following contributions. We have augmented the learning procedure with bottom-up segmentations as an additional grouping cue to focus search on promising image contours, thus eliminating the need for bounding boxes. Multiple segmentations are extracted per image to reduce dependency on any specific one, and the region boundaries are used only as initial cues, rather than hard constraints on the model as in other approaches. Secondly, in applying the framework to shape features, we have designed a more stable version of kAS local contour features [7] by defining a canonical ordering of geometric components, and show that they are more effective than appearance features (*e.g.*, SIFT) for categorization. Finally, we have adapted proximity grouping to contours by taking into account their long, curvilinear nature.

Following a review of related work in Section 2, we continue in Section 3 with an overview of Jamieson's framework and its application. A description of the object model and its constituent contour parts and spatial relations is found in Section 4, followed by object detection in Section 5. The learning algorithm is covered in Sections 6 and 7, which describe the co-occurrence objective and the learning algorithm. We conclude with a discussion of encouraging results and future directions.

Chapter 2

Related work

Approaches to shape-based object category recognition have often been supervised or semi-supervised with manual class labelling and bounding boxes, *e.g.*, Shotton *et al.* [18], Ferrari *et al.* [9]. Other approaches do not assume any supervision, but use distinctive appearance features (Kim *et al.* [12]), sometimes in conjunction with contours (Lee & Grauman [13]). More recently, Payet & Todorovic [17] showed that shape alone is sufficiently distinctive for unsupervised learning by finding clusters of spatial configurations of contours. Distinct object categories are automatically found over cluttered images via a probabilistic colouring of a graph over matching pairs of contours and spatial relations. While our visual representation also consists only of contours, we take an integrated approach where categorization is guided by both bottom-up segmentation and image caption text. In particular, the presence of linguistic regularities across captioned images provides a comparatively efficient way to initialize visual clusters.

Language-vision integration seeks correlations between words and visual features in a set of image-text pairs. Barnard *et al.* [1], and Duygulu *et al.* [5] learned distributions over words and visual features of segmented regions described by colour, texture, appearance, and global shape. A single image segmentation, however, is prone to error as a grouping mechanism. Even when oversegmented regions are merged (Barnard *et al.* [2]), grouping is still limited by the accuracy of region boundaries. Jamieson *et al*. [11] does not rely on bottom-up segmentation, and instead learns correspondences between words and spatial configurations of appearance features by grouping nearby features together. Recognition approaches that exclude language have also relied on bottom-up segmentation for grouping, and reduce dependency on any one segmentation via multiple segmentations. Russell *et al*. [17], used text analysis methods to rank regions from multiple segmentations per image, where regions were described by their interior appearance. Gu *et al*. [10] used a region tree segmentation with a richer description including contours. In comparison to these methods, we use the boundaries of multiple segmentations only as initial cues for promising image contours, and do not limit learning or recognition to their accuracy.

Since we model objects as a graph over related parts, we briefly review other graphical models. While Fergus *et al*. [6] and Crandall & Huttenlocher [4] learn graph models from weakly supervised images, graphs are constrained to a star structure. Shotton *et al*. [18] and Opelt *et al*. [16] learn centroid-voting shape-based parts, although a subset of training images are assumed to be labelled. In using the framework of Jamieson *et al*. [11] we learn from unsupervised captioned images a graph representation of local shape with no structural constraints.

Chapter 3

Overview of approach

Given a word w, a corresponding visual representation of an object is learned by maximizing the co-occurrence score $C(w, M)$ with respect to the object model M, over captioned images. The object model M is a flexible graph representation (Figure 3.1) over object parts (contour features) with pairwise spatial relations, providing local representation for matching under occlusion and local variations. Since objects are spatially coherent, we require the graph to be connected, but impose no further constraints on the connectivity and number of parts. Part relations add distinctiveness and coherence to an otherwise structureless bag-of-features model, which is more likely to yield accidental detections. Relations are spatially invariant, allowing the model to inherit the spatial invariance of

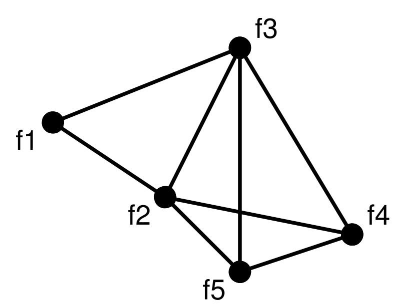


Figure 3.1: An object is modelled as a graph over spatially related parts, which are local invariant features, *e.g.*, $F = \{f_1, \ldots, f_5\}$.

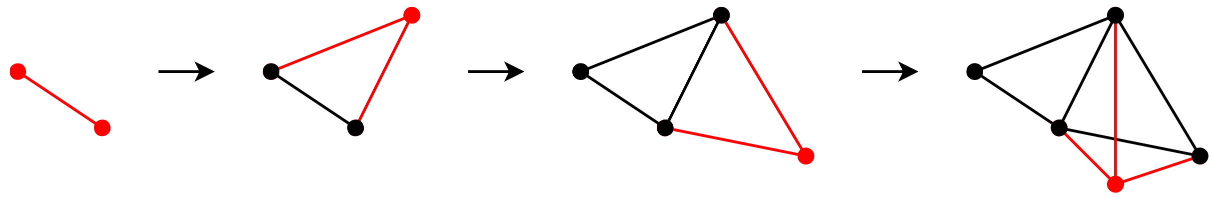


Figure 3.2: The learning procedure starts with a small set of spatially related local features (far left), and iteratively expands by a related feature until the co-occurrence score converges.

constituent parts. Objects are detected efficiently by using spatial relations to prune the search for a set of matching image features.

Due to high scene variability, learning a parts-based model from unlabelled images is potentially very expensive. The framework of Jamieson *et al*. [11] reduces this complexity using 1) a greedy learning procedure that constructs the model part-by-part, and 2) a grouping constraint based on feature proximity. The learning algorithm greedily constructs a model part-by-part (Figure 3.2), where each iteration finds one additional part given existing parts. Thus, the model grows from a small, weak set of related parts, to a larger, more distinctive set. The iterative nature of learning allows proximity constraints to be applied in a straightforward manner: a new part is learned from only those features in the vicinity of existing parts. By constraining model parts to be in close proximity to each other, image features likely to be mutually irrelevant are efficiently disregarded. The initial set of parts is learned by finding clusters of similar feature neighbourhoods.

A major extension of the framework is an additional grouping cue for initialization using multiple bottom-up segmentations. This is a natural way to focus search on image contours that are likely to correspond to object boundaries a priori. Subsequent parts are not constrained to segmentations, and thus our approach is not limited by the accuracy of bottom-up segmentation. Furthermore, the initial segmentation constraint is of benefit to the greedy learning algorithm, in which model parts remain unchanged once they are added. Clearly, it is important to add only object parts to the model, and this is particularly difficult in the early stages due to the low specificity of shape: we are hoping to find only *small* contour portions that are relevant and stand out from background clutter across images. The effect of using multiple bottom-up segmentations is to bring out the potential regularities, making them easier to find and ensuring that the model grows from a good initialization.

Model growth converges when no more parts can be added that increase the co-occurrence score $C(w, M)$. Although this criterion does not guarantee that the complete bounding and interior contours of an object are learned, it ensures that parts are collectively distinctive to maximally distinguish between images with and without w in their captions.

Chapter 4

Object model

In this section we describe the local invariant geometric contour features that constitute the model parts $F = \{f_1, \ldots, f_T\}$, and the relations $S = \{S_{f_i,f_j}\}$ that encode the change in spatial properties between parts.

It will be useful below to distinguish a model feature from an image feature, hence we use f to refer to the former, and ϕ to refer to the latter. Despite the notational difference, both f and ϕ refer to the contour feature described in this section.

4.1 Local invariant contour features

We represent local contours with geometric, invariant descriptors derived from a linearization of edgel chains, which are very similar to kAS features [8]. Given edgels from the Pb detector [15], we obtain a linearization via the Contour Segment Network by Ferrari *et al*. [7] which chains edgels together, bridges edgel chains over potential contour gaps, and partitions the resulting chains into linear segments. The result is a branching network of line segments linked between endpoints and junction points. Contour descriptions are obtained by grouping linked line segments together. An overview of feature extraction is shown in Figure 4.1.

Ferrari *et al*. [8] extract kAS features by taking all groups of k linked line segments.

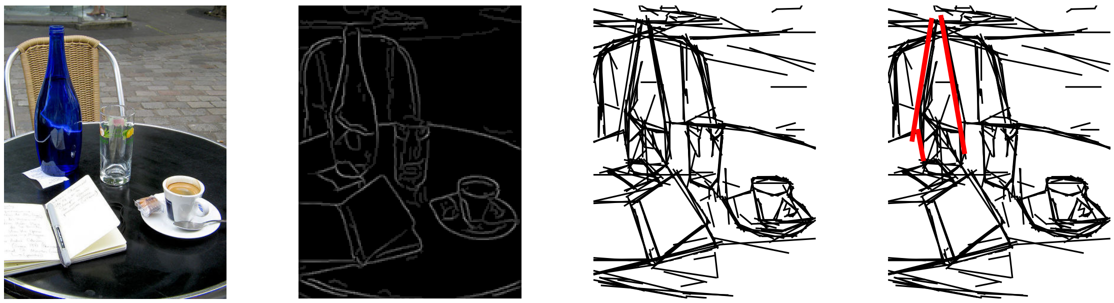


Figure 4.1: The feature extraction process from 1) image pixels, to 2) contour edgels, to 3) overlapping line segments at multiple scales, 4) from which one extracted contour feature is highlighted in red (consisting of 3 line segments).

Since links exist between endpoints as well as junction points, kAS features describe a rich set of contour configurations including paths, T-, and Y-junctions. With such a variety of shapes, however, comes the difficulty of defining a stable internal ordering of segments. The centroid- and axis-based ordering used by kAS is neither stable nor robust to changes in orientation, and could negatively affect performance when detection is feature-based. We obtain a stable *continuity*-based ordering while keeping a sufficiently expressive set of shapes. By extracting only paths of line segments, a canonical ordering is achieved[1]. Furthermore, this ordering is orientation-independent, leading directly to rotation-invariance, if desired.

Our feature descriptor ϕ is identical in form (but not in order) to that of kAS features [8]. It is a vector encoding of k line segments, denoted $s_1, \ldots, s_k$ such that s_i precedes s_{i+1} in a path (we use $k = 3$). Each segment is described by its relative position $\vec{p}_i$, orientation ψ_i, and length ℓ_i:

$$\phi = (\vec{p}_2, \ldots, \vec{p}_k, \psi_1, \ldots, \psi_k, \ell_1, \ldots, \ell_k). \tag{4.1}$$

Relative positions are measured with respect to the first segment position $\vec{p}_1$, which is

[1]The ordering is canonical up to the two possible directions in a path. The disambiguation of these two possibilities is discussed in Appendix B.1.

therefore omitted. (If rotation-invariance is desired, orientations are measured relative to ψ_1, which would also be omitted.) The descriptor ϕ is scale-invariant with respect to the distance z between the *furthest* two segment midpoints, *i.e.*, the relative positions and lengths are normalized by z. Contour similarity accounts for shape deformation and is measured using the kAS distance [8] between two contour descriptors ϕ and ϕ':

$$d(\phi, \phi') = w_r \sum_{i=2}^{k} ||\vec{p}_i' - \vec{p}_i|| + w_\psi \sum_{i=1}^{k} \angle(\psi_i', \psi_i) + w_l \sum_{i=1}^{k} |\log(\ell_i'/\ell_i)|\,. \tag{4.2}$$

Feature matching and clustering (Sections 5, 7) are facilitated with a codebook Q of feature codewords (or "visual vocabulary" [19]), where each image feature ϕ is quantized into its nearest representative $q \in Q$ at extraction time. By pre-computing feature distance at the codeword level, recognition and learning is more efficient. Construction of the codebook via clustering of background image features is described in Appendix B.2.

4.1.1 Multi-scale image representation

Significant changes in contour curvature leading to distinct features may arise from perceptually insignificant changes. For example, viewpoint distance, object size, or local changes in scale or detail can result in qualitatively different linearizations, and hence features that are not repeatable. To increase the chance that regularities are found, we compute linearizations at multiple scales to yield a multi-scale image representation. (Due to practical limitations, linearizations were computed over rescaled images.) Unlike a layered representation that handles only global image variation (*e.g.*, image pyramid), we allow different model parts to match at different scales by mixing features from different scales together (Figure 4.1). Our image representation is thus a rich set of contour features that increases matching availability in its redundancy.

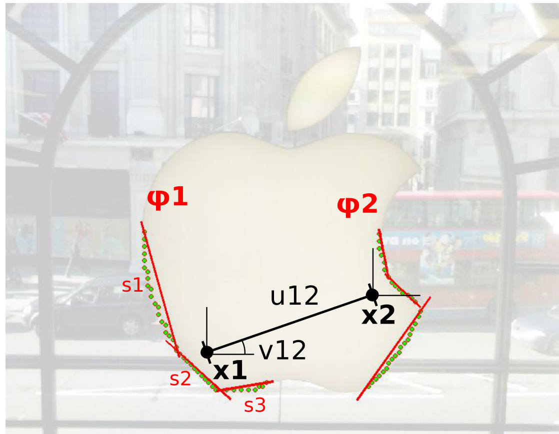


Figure 4.2: Distance (u_{12}) and relative orientation (v_{12}) components of the spatial relation between two image contour features ϕ_1 and ϕ_2. Green circles indicate the underlying edgel-chains of each contour, which have been linearized into adjoining segments.

4.2 Pairwise spatial relations

Each contour has spatial properties with respect to image coordinates, namely its position $\vec{x}$ and scale s in the image. Spatial properties are defined identically to those of kAS features: position is the average midpoint of all line segments, and scale is the value z, defined above. (A rotation-invariant version also has an image orientation θ, defined as the orientation of the first line segment, ψ_1.) The spatial relation between two features ϕ_i and ϕ_j is a vector S_{ϕ_i,ϕ_j} encoding the change in spatial properties

$$S_{\phi_i,\phi_j} = (u_{ij}, v_{ij}, w_{ij}) \tag{4.3}$$

with components of distance u_{ij}, direction v_{ij}, and relative scale w_{ij}. Figure 4.2 illustrates selected feature relations. To preserve feature invariance, each component itself is spatially invariant. The distance u_{ij} between features is normalized by $\lambda = \min(s_i, s_j)$, hence

$$u_{ij} = \frac{||\vec{x}_j - \vec{x}_i||}{\lambda}. \tag{4.4}$$

The direction v_{ij} of ϕ_j with respect to ϕ_i, measured in $[0, 2\pi)$, is given by

$$v_{ij} = \arctan(\vec{x}_j - \vec{x}_i), \tag{4.5}$$

and change in scale w_{ij} is given by

$$w_{ij} = \frac{s_j - s_i}{\lambda}. \tag{4.6}$$

Aside: Relations between rotation-invariant features require an additional degree of freedom for feature orientation with respect to the image. This can be achieved as in Jamieson *et al.* [11], where direction is captured by two *headings* v_{ij}, v_{ji} as follows:

$$v_{ij} = \angle(\theta_i, \arctan(\vec{x}_j - \vec{x}_i)) \tag{4.7}$$

$$v_{ji} = \angle(\theta_j, \arctan(\vec{x}_i - \vec{x}_j)). \tag{4.8}$$

This completes the description of image features and pairwise relations, carrying over to model features $F = \{f_1, \ldots, f_T\}$ and relations $S = \{S_{f_i, f_j}\}$. We do not require $M = (F, S)$ to be a complete graph so that spatial relations can be described at a simpler, local level. Edge redundancy (*i.e.*, beyond a spanning tree), however, is useful for maintaining model coherence under partial matching (*e.g.*, due to occlusion).

Chapter 5

Object occurrence

An occurrence of an object $M = (F, S)$ is found by selecting a subset of image features that match part-wise to model features F subject to spatial constraints S, as shown in Figure 5.1. Only a subset of model features may be matched (*e.g.*, due to occlusion). Detection is performed efficiently by exploiting the spatial constraints between model parts via pruning. We score detections as in Jamieson *et al.* [11], though here we attempt a more concise formulation by incorporating partial matching into the likelihood model.

For notation, let $\mathbf{h}$ be a mask over model features indicating which ones are matched, and let $F(\mathbf{h}) \subset F$ indicate the subset of matched model features, and $S(\mathbf{h}) \subset S$ the model relations induced by $F(\mathbf{h})$. A (partial) match is denoted by $\Phi = \{\phi_f : f \in F(\mathbf{h})\}$, where ϕ_f indicates correspondence to the model feature f.

5.1 Detection score

The detection score $D(M, \Phi) \in [0, 1]$ is defined in terms of a common formulation combining similarity to the object model M and dissimilarity to a constant background model B via the likelihood ratio

$$\frac{p(\Phi|M)}{p(\Phi|B)}. \tag{5.1}$$

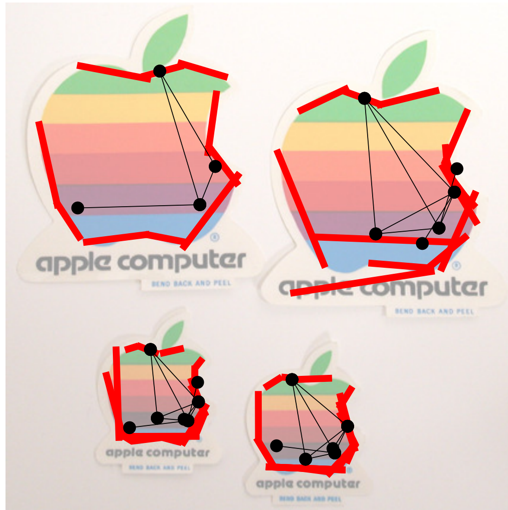


Figure 5.1: Example detections annotated with the line segment representation of contour features. Feature positions ($\vec{x}$'s) are indicated with black vertices, and edges indicate spatial relations. Not all features are matched due to imperfect recall.

We define $D(M, \Phi)$ as the posterior probability of the object, which is an increasing function of the likelihood ratio as shown below:

$$\begin{aligned} D(M, \Phi) &= p(M|\Phi) \\ &= \frac{p(\Phi|M)p(M)}{p(\Phi|M)p(M) + p(\Phi|B)p(B)} \\ &= \frac{1}{1 + \frac{p(\Phi|B)}{p(\Phi|M)}\frac{p(B)}{p(M)}}. \end{aligned} \tag{5.2}$$

We assume independence among model components and factor the object likelihood $p(\Phi|M)$ into a part matching term, a spatial relation term, and a term for partial matching, respectively:

$$p(\Phi|M) = p(\Phi|F)p(\Phi|S)p(\mathbf{h}|M). \tag{5.3}$$

The feature and spatial terms factor into their respective components:

$$p(\Phi|F) = \prod_{f \in F(\mathbf{h})} p(\phi_f|f) \tag{5.4}$$

$$p(\Phi|S) = \prod_{(f_i, f_j) \in S(\mathbf{h})} p(S_{\phi_{f_i}, \phi_{f_j}}|S_{f_i, f_j}). \tag{5.5}$$

Feature probabilities $p(\phi_f|f)$ are modelled with a Gaussian distribution with variance σ_{f}^2. Spatial relations $p(S_{\phi_{f_i},\phi_{f_j}}|S_{f_i,f_j})$ are also Gaussian-distributed around mean (u_{ij}, v_{ij}, w_{ij}) with diagonal variances $(\sigma_{\mathrm{u}}^2, \sigma_{\mathrm{v}}^2, \sigma_{\mathrm{w}}^2)$. Partial matches are assumed to arise from features that independently match with probability $\alpha \in [0, 1]$, hence the following factorization of $p(\mathbf{h}|M)$ in Equation 5.3:

$$p(\mathbf{h}|M) \quad = \quad \alpha^{|F(\mathbf{h})|+|S(\mathbf{h})|}(1-\alpha)^{|\overline{F(\mathbf{h})}|+|\overline{S(\mathbf{h})}|}. \tag{5.6}$$

Spatial relations in the exponents in Equation 5.6 reflect independence among model components, so the first factor corresponds to the matched portion of the model, and the second factor the unmatched portion (with the bar indicating set complement).

The background likelihood of a match similarly factors into three terms:

$$p(\Phi|B) \quad = \quad p(\Phi|f_B)p(\Phi|S_B)p(\mathbf{h}|B) \tag{5.7}$$

where

$$p(\Phi|f_B) \quad = \quad \prod_{f\in F(\mathbf{h})} p(\phi_f|f_B) \tag{5.8}$$

$$p(\Phi|S_B) \quad = \quad \prod_{(f_i,f_j)\in S(\mathbf{h})} p(S_{\phi_{f_i},\phi_{f_j}}|S_B). \tag{5.9}$$

The background feature likelihood $p(\phi|f_B)$ represents how likely ϕ occurs accidentally, which we have approximated with a uniform distribution (alternatives are discussed in Appendix A.2). Background distance and relative scale are Gaussian-distributed with (wide) variances $\sigma_{\mathrm{bu}}^2, \sigma_{\mathrm{bv}}^2$, while background direction is uniformly distributed over $[0, 2\pi)$. The term $p(\mathbf{h}|B)$ is the likelihood that the whole model is unmatched, thus

$$p(\mathbf{h}|B) \quad = \quad \alpha^{|F|+|S|}. \tag{5.10}$$

Substituting the object and background likelihoods into the ratio (Equation 5.1), the complete likelihood ratio is

$$\frac{p(\Phi|M)}{p(\Phi|B)} \quad = \quad \left(\frac{1-\alpha}{\alpha}\right)^{|\overline{F(\mathbf{h})}|+|\overline{S(\mathbf{h})}|} \prod_{f\in F(\mathbf{h})} \frac{p(\phi_f|f)}{p(\phi_f|f_B)} \prod_{(f_i,f_j)\in S(\mathbf{h})} \frac{p(S_{\phi_{f_i},\phi_{f_j}}|S_{f_i,f_j})}{p(S_{\phi_{f_i},\phi_{f_j}}|S_B)}. \tag{5.11}$$

The three components of Equation 5.11 can be interpreted as 1) the penalty for unmatched features, 2) the likelihood ratio for matched features, and 3) the likelihood ratio for relations between matched features, respectively.

5.2 Detection algorithm

In a cluttered image with thousands of features, it is impractical to consider all possible matches to different subsets of parts. Rather, a match $(\Phi, \mathbf{h})$ of a given model $M = (F, S)$ is found using the relational constraints S to efficiently prune the search space. We only give an overview here, and refer the reader to Algorithm 1 and Jamieson *et al.* [11] for details.

Algorithm 1: finding a detection $(\Phi, \mathbf{h})$ of model $M = (F, S)$ in image I

$((f_i, f_j), (\phi_{f_i}, \phi_{f_j})) \leftarrow$ find_best_matching_edge(I, M)

if no edge found **then return** $(\varnothing, \mathbf{0})$

$\Phi \leftarrow \{\phi_{f_i}, \phi_{f_j}\}$

$\mathbf{h} \leftarrow \mathbf{0}; h_i, h_j \leftarrow 1, 1$

while $\sum \mathbf{h} < |F|$ **do**

  $((f, f_k), (\phi_f, \phi_{f_k})) \leftarrow$ find_best_edge_expansion(I, M, Φ) such that $\phi_f \in \Phi$

  if no expansion found **then break**

  $\Phi \leftarrow \Phi \cup \{\phi_{f_k}\}$

  $h_k \leftarrow 1$

end

if $D(\Phi, \mathbf{h}) > t$ **then return** $(\Phi, \mathbf{h})$; **else return** $(\varnothing, \mathbf{0})$

Detection is performed greedily by first seeking for the best matching pair of adjacent model vertices, then iteratively expanding the match along edges incident to the currently matched subgraph. The matching criterion at each step is to maximize the likelihood ratio (Equation 5.11), with α set to $\frac{1}{2}$ so that there is no penalty for partial matching in intermediate steps. In mathematical notation, suppose that $\Phi^{(0)} = \{\phi_{f_i}, \phi_{f_j}\}$ is initially the best matching pair of adjacent model vertices (f_i, f_j, and their relation). Each subsequent iteration, indexed by τ, expands the previous match $\Phi^{(\tau-1)}$ by considering each edge incident to the features in $\Phi^{(\tau-1)}$. The best matching adjacent feature ϕ_k is added to obtain $\Phi^{(\tau)} = \Phi^{(\tau-1)} \cup \{\phi_k\}$. A match stops expanding (*i.e.*, is pruned) when no such edge increases the likelihood ratio, or when no more edges are available (in which case the match is complete). The final match $(\Phi, \mathbf{h})$ is scored with $D(M, \Phi)$. Multiple detections in an image are found by restricting subsequent searches to remaining image features. Matches whose bounding boxes overlap more than 20% are disambiguated by removing the lower-scoring match.

Chapter 6

Object co-occurrence with word

Our learning objective is to maximize the co-occurrence score $C(w, M)$ of the object model M with a given word w, where occurrences are determined via object detection (Section 5) and caption string matching. Perfect co-occurrence over N images would allow us to conclude with high confidence that the word and visual representation are in correspondence, but this is a highly unlikely scenario due to factors arising from word ambiguity, object occlusion and orientation, and reliability of the captions. Jamieson's framework uses a naive Bayes model to determine the posterior probability that w and M are in correspondence, given their observations over N images. We indicate word observations by

$$\mathbf{w} = \{w_1, \ldots, w_N\}, \tag{6.1}$$

where w_n is 1 if the word w occurs in the caption of the nth image, and 0 otherwise. Observations of the object via M are indicated by

$$\mathbf{m} = \{m_1, \ldots, m_N\}, \tag{6.2}$$

where m_n is a confidence score in $[0, 1]$. If M is not detected in the nth image, m_n is 0; otherwise, m_n is equal to $D(M, \Phi)$. When there are multiple detections in the image, object occurrence is indicated by the maximum of the detection scores.

We evaluate the likelihood of the observed values under two hypotheses: 1) correspondence, $p(\mathbf{w},\mathbf{m}|G)$, and 2) non-correspondence $p(\mathbf{w},\mathbf{m}|H)$. Under non-correspondence, observations of the word and object are mutually accidental, thus the likelihood factors independently:

$$p(\mathbf{w},\mathbf{m}|H) = \prod_{n=1}^{N} p(w_n|H)p(m_n|H) \tag{6.3}$$

The probabilities $p(w_n|H)$ and $p(m_n|H)$ are determined empirically from training data by frequency of word and object detection, respectively.

Under correspondence, observations of a word and object are expected to correlate with one another. To model this probabilistically, we introduce a conditional hidden variable $\mathbf{o} \in \{0,1\}^N$, where o_n is 1 if the object is present in the scene, and 0 otherwise:

$$\begin{aligned} p(\mathbf{w},\mathbf{m}|G) &= \prod_{n=1}^{N} p(w_n, m_n|G) \\ &= \prod_{n=1}^{N} \sum_{o_n} p(w_n, m_n|o_n)p(o_n|G) \\ &= \prod_{n=1}^{N} \sum_{o_n} p(w_n|o_n)p(m_n|o_n)p(o_n|G) \end{aligned} \tag{6.4}$$

In Equation 6.4 the probabilities $p(w_n|o_n = 1)$ and $p(w_n|o_n = 0)$ express the uncertainty of word occurrence depending on whether the object is present. Words may have multiple meanings corresponding to different objects, and there may be multiple synonymous names for an object. As with words, detections of objects do not necessarily correspond to the presence of the object in the scene. When an object is present, the probability $p(m_n|o_n = 1)$ accounts for uncertainty in detecting the object, which may be fully or severely occluded, or captured from an unusual viewpoint. The probability $p(m_n|o_n = 0)$ accounts for accidental detections of an object that is absent from the scene. We set $p(o_n|G)$ to 0.5, and give further details of $p(w_n|o_n)$ and $p(m_n|o_n)$ in Appendix A.1.

We can now define the co-occurrence score $C(w, M)$ as the posterior probability of correspondence $p(G|\mathbf{w},\mathbf{m})$, given priors $p(G)$ and $p(H)$, and the likelihoods as defined above:

$$
\begin{aligned}
C(w, M) &= p(G|\mathbf{w}, \mathbf{m}) \\
&= \frac{p(\mathbf{w}, \mathbf{m}|G)p(G)}{p(\mathbf{w}, \mathbf{m}|G)p(G) + p(\mathbf{w}, \mathbf{m}|H)p(H)}.
\end{aligned}
\tag{6.5}
$$

In the following section we use $C(w, M)$ as a scoring function to find a model M that maximally co-occurs with a word w over captioned images.

Chapter 7

Learning co-occurring object models

An object model M that corresponds to w is learned in a greedy part-by-part process, beginning with a small, initial model $M^{(0)}$ and iterating over successively larger models $M^{(1)}, M^{(2)}, \ldots, M^{(\tau)}, \ldots$ until $C(w, M^{(\tau)})$ converges. Each expansion of the model is learned over image subregions defined in the vicinity of previous model detections, thus focusing search on only those features likely to be related to the object. Learning is summarized in Algorithm 2 and the iterative step is illustrated in Figure 7.1.

7.1 Part expansion

Each iteration takes as input the model $M^{(\tau-1)}$, and learns a larger model $M^{(\tau)}$ such that co-occurrence is (maximally) increased:

$$C(w, M^{(\tau)}) \quad > \quad C(w, M^{(\tau-1)}). \tag{7.1}$$

Greediness arises from the larger model being a supergraph of the given model:

$$M^{(\tau)} \quad = \quad M^{(\tau-1)} \cup E. \tag{7.2}$$

The model *expansion* E consists of a new part f^* (a vertex) that is spatially related (via edges) to the parts of $M^{(\tau-1)}$:

$$E \quad = \quad (f^*, \{S_{f_i, f^*}\}). \tag{7.3}$$

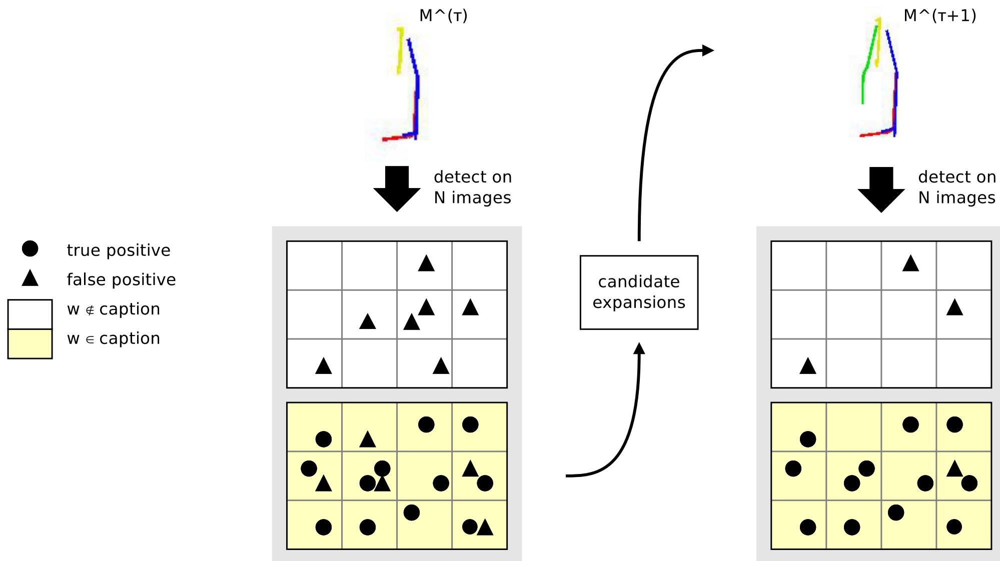


Figure 7.1: Instances of $M^{(\tau)}$ are detected with many false positives. Detections on images captioned with w yield candidate expansions, from which one is selected to obtain $M^{(\tau+1)}$. The co-occurrence of $M^{(\tau+1)}$ is higher due to fewer false positive detections.

7.1.1 Proximity constraint

Learning complexity is reduced by identifying image features that are non-accidentally grouped by proximity to the given model, and restricting learning to these features. The *neighbourhood* $N(\cdot)$ of a contour feature ϕ is the set of contour features within a maximum spatial distance of q, measured by $d_{\text{nbh}}(\phi, \cdot)$:

$$N(\phi) \quad = \quad \{\phi' : d_{\text{nbh}}(\phi, \phi') < q\}. \tag{7.4}$$

Distance between contours is defined in terms of their long, curvilinear structure. Intuitively, two contours that together span a large part of an image but are touching have zero distance between them. We thus define the distance $d_{\text{nbh}}(\cdot, \cdot)$ between two contours as the *minimum* distance between their underlying image edgels. In practice (due to practical issues with the linearization software), we compute an approximation using

$$d_{\text{nbh}}(\phi, \phi') \approx \min_{\substack{s \in \{s_1, \ldots, s_k\} \\ t \in \{s'_1, \ldots, s'_k\}}} d_{\text{line}}(s, t), \tag{7.5}$$

Algorithm 2: Learning model M from captioned images $\{I_n\}_{n=1}^{N}$ given word w

$\tau \leftarrow 0$

$\{I_n^+\}_{n=1}^{N+} \leftarrow \{I_n : w \text{ in caption}\}$

$\{J_n^+\}_{n=1}^{N+} \leftarrow \{\{\phi \in I_n^+ : \phi \text{ masked by segmentation boundaries}\}\}$

$M^{(0)} \leftarrow \text{initialize}(\{J_n^+\})$

$\{\Phi_{M^{(0)}}\} \leftarrow \text{detect}(\{I_n^+\}, M^{(0)})$

repeat

$\tau \leftarrow \tau + 1$

$\{J_n^+\}_{n=1}^{N+} \leftarrow \{N(\Phi) : \Phi \in \{\Phi_{M^{(\tau-1)}}\}\}$

$\{E_c\}_{c=1}^{C} \leftarrow \text{expansion-candidates}(\{J_n^+\})$

$M^{(\tau)} \leftarrow \arg\max_{c=1}^{C} C(w, M^{(\tau-1)} \cup E_c)$

until $M^{(\tau-1)}$ converged;

return $M := M^{(\tau-1)}$

where the s_i's are the line segments of the respective features (Section 4.1) and $d_{\text{line}}(\cdot,\cdot)$ is the distance between two lines. The maximum contour distance q is linear in the scale of ϕ, thus allowing contours with a larger spatial extent to have a spatially larger neighbourhood.

We restrict expansion learning to features that are in proximity to instances of $M^{(\tau-1)}$. An instance neighbourhood J (an image subregion) is defined by taking the *union* of the neighbourhoods of matching features. Note, however, that not all instance neighbourhoods may contain an example of an expansion. For example, false positive detections do not correspond to true object instances, and so are not expected to yield consistent features nearby. As models grow in size, however, the increase in distinctiveness reduces the rate of false neighbourhoods, thus allowing learning to bootstrap on successively more reliable spatial constraints.

A further constraint arises from our multi-scale image representation. Recall from Section 4.1.1 that features are combined from multiple linearizations to increase repeatability across local variations. To ensure that expansions represent novel object parts, the neighbourhood $N(\phi)$ omits contour features that completely overlap with ϕ.

7.1.2 Expansion learning

The instance neighbourhoods $\{J_n\}$ are taken to be a set of independent image subregions containing examples of expansions. Learning is restricted to images whose captions contain the word w, as other images are not expected to contain object instances. The two components of an expansion $E = (f^*, \{S_{f_i,f^*}\})$ are learned in succession by 1) finding recurring features via codebook voting, then 2) using mean-shift to identify stable relations to existing features of the model.

Given the set of instance neighbourhoods $\{J_n\}$, the voting space $\mathbb{N}^{|Q|}$ counts occurrences of codewords in each $J \in \{J_n\}$. To ensure that regularities are found *across* (rather than within) neighbourhoods, each neighbourhood $J \in \{J_n\}$ contributes a maximum of one vote per codeword $q \in Q$. Codewords with the most votes are selected to be candidates $f_1^*, \ldots, f_c^*$ for the new part f^*, where $c \leq 50$.

A new part f^* is selected for an expansion by learning its relations $\{S_{f_i,f^*}\}$ to the given model. Pairwise relations are modelled independently and learned in succession. We first seek a stable relation to one of the model features whose neighbourhood contains f^*. Examples for spatial relations, $\mathbf{S} = (S_1, \ldots, S_K)$, are drawn from the instance neighbourhoods $\{J_n\}$, and comprise the data for density estimation. Each $S_k \in \mathbf{S}$ is a spatial relation vector (Equation 4.3) encoding distance, relative scale, and relative orientation. A spatial relation mode is found using mean-shift over $\mathbf{S}$ by initializing with points of high density, and accepted if its score exceeds a minimum threshold. This procedure is repeated with respect to other model features. If there is no acceptable mode for a particular model feature, then no relation exists to that feature.

This completes a list of expansion candidates $E_1, \ldots, E_c$, from which a selection is made to maximize the learning objective $C(w, M^{(\tau)})$. We choose the expansion E such that $M^{(\tau)} := M^{(\tau-1)} \cup E$ has the highest co-occurrence, or, if there is no such expansion due to convergence, then the final model $M := M^{(\tau-1)}$ is returned.

7.2 Part initialization

The initial model $M^{(0)}$ represents a small portion of an object's contours that is learned without any prior spatial information. To ensure that a relevant model is initialized from noisy images, we derive *boundary hypotheses* from multiple segmentations to constrain initialization over only a subset of promising image features (Figure 7.2). While a range of structures are possible with varying levels of distinctiveness, we initialize a graph of two related parts, *i.e.*, $M^{(0)} = (\{f_1, f_2\}, S_{f_1,f_2})$.

7.2.1 Bottom-up segmentation constraint

The Superpixel Closure segmentation algorithm by Levinshtein *et al*. [14] uses a gap-to-area criterion to select a contiguous region of superpixels for figure-ground segmentation. Multiple segmentations at different image scales and locations, corresponding to different hypothetical objects, are obtained by varying a weight parameter. Given a set B of boundary edgels, we are interested in any contour feature ϕ that is within a minimum distance $d_{\text{bnd}}(B, \phi)$ to the boundary. To obtain only those contours that fall entirely in the immediate vicinity of the boundary we use the Hausdorff distance:

$$d_{\text{bnd}}(B, \phi) \quad = \quad \max_{e \in \text{edgels}(\phi)} \left(\min_{b \in B} ||e - b|| \right). \tag{7.6}$$

As shown in Figure 7.2, initial part learning is restricted to image boundary features, obtained by thresholding the distance $d_{\text{bnd}}(\cdot, \cdot)$ to multiple segmentation boundaries.

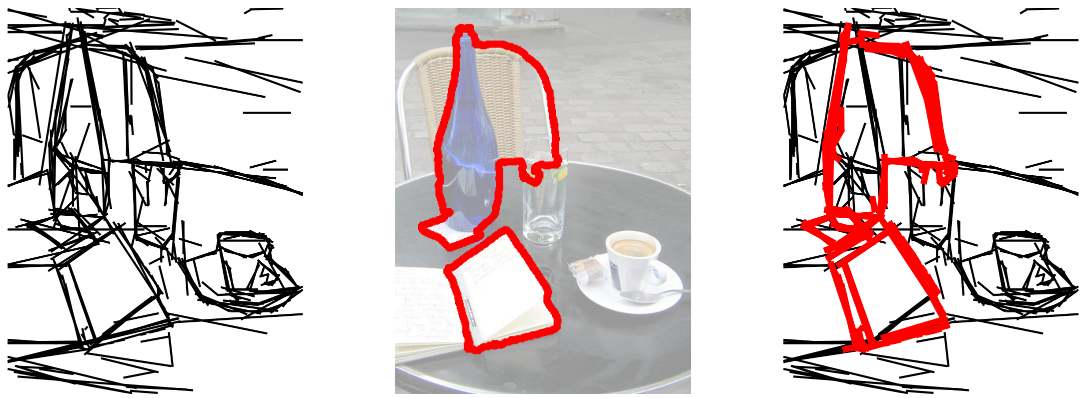


Figure 7.2: Initial segmentation constraint: 1) extracted image features, 2) figure-ground segmentation hypotheses, and 3) image features masked by the boundary hypotheses.

7.2.2 Initial learning

In a procedure similar to the iterative step, image features are restricted according to constraints, and parts are learned via codebook voting and mean-shift. Given features masked by boundary hypotheses, an initial model $M^{(0)} = (\{f_1, f_2\}, S_{f_1,f_2})$ is learned by 1) finding recurring feature pairs via codeword voting over a joint vote space $\mathbb{N}^{|Q|} \times \mathbb{N}^{|Q|}$, and 2) mean-shift over the induced relation examples.

Chapter 8

Evaluation

We evaluate object localization on the benchmark dataset ETHZ [7] which contains 5 shape categories appearing in 255 cluttered scenes. In this evaluation, image category labels are treated as image captions. It is possible to simulate caption noise by random re-assignment of a subset of category labels, although we have not done so in our experiments. A random half of the images (per category) were used as training examples, from which we extracted bottom-up segmentations and learned the models without using bounding boxes. For each category, up to 20 models were learned from independent initializations, from which the one with the highest co-occurrence score was selected. Performance is evaluated on the test set (the remaining half) using precision and recall.

Thresholded detections are counted as true positives when the detection bounding box BB_d overlaps at least 50% with the ground truth bounding box BB_{gt}, where overlap is measured by intersection-over-union $(BB_{gt} \cap BB_d)/(BB_{gt} \cup BB_d)$. The detection bounding box BB_d is defined with the following observation in mind: ground truth bounding boxes capture not only correct localization, but also the correct spatial extent of the object. Because our learned models do not always correspond to the entire object boundary, BB_d needs to be defined so that localization can be evaluated independently of object completeness. (For example, if only a portion of a Giraffe was learned, localization

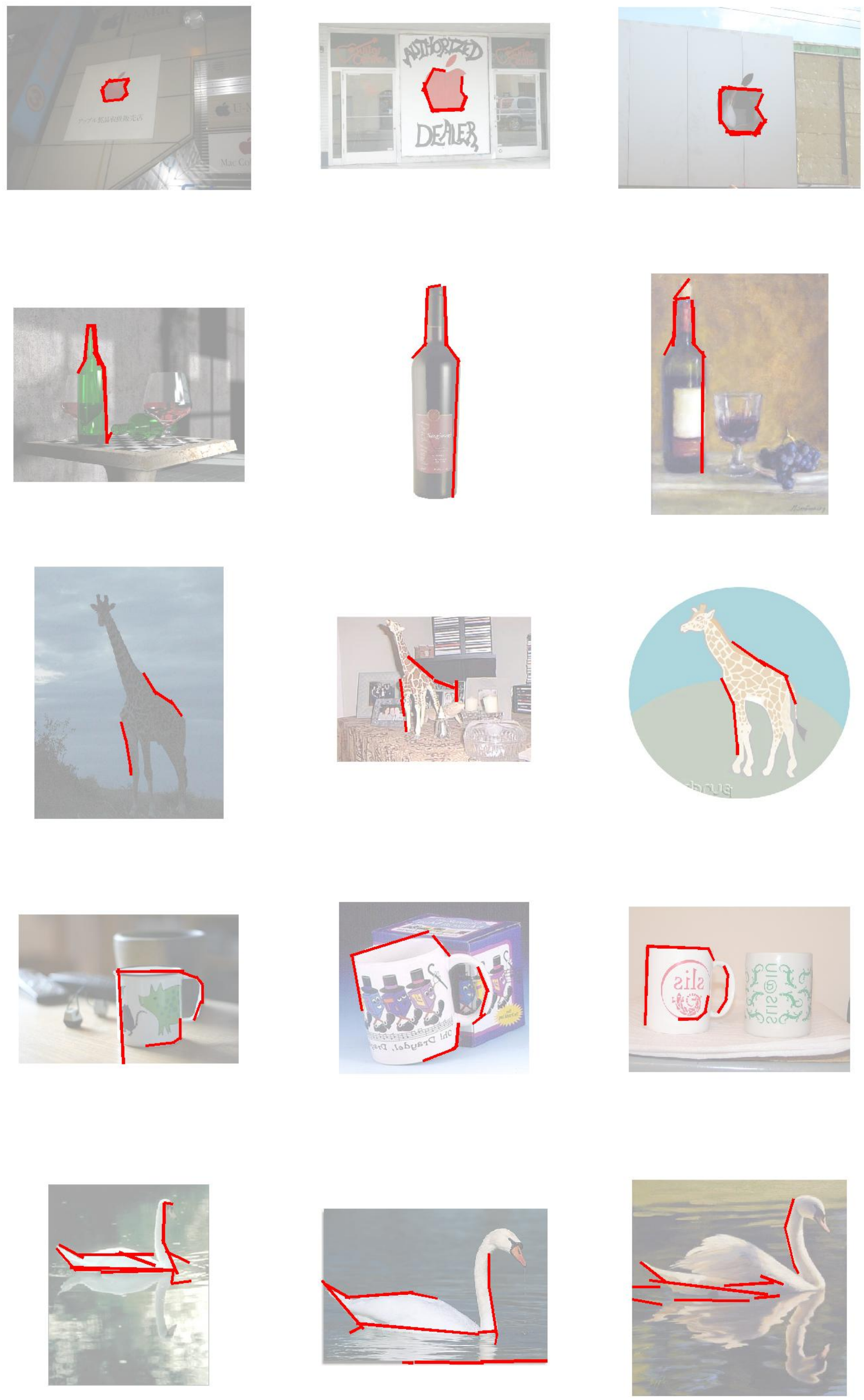


Figure 8.1: Example detections of the 5 ETHZ categories in cluttered images. Detections are annotated with the line segment representation of contours.

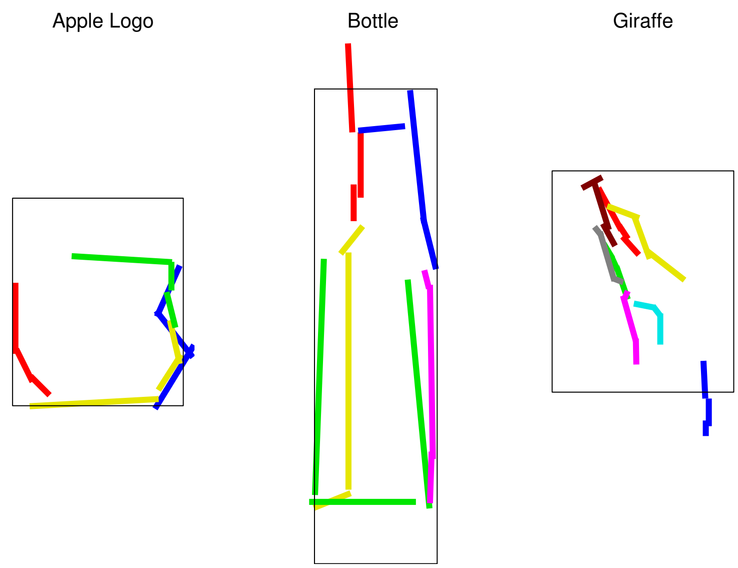
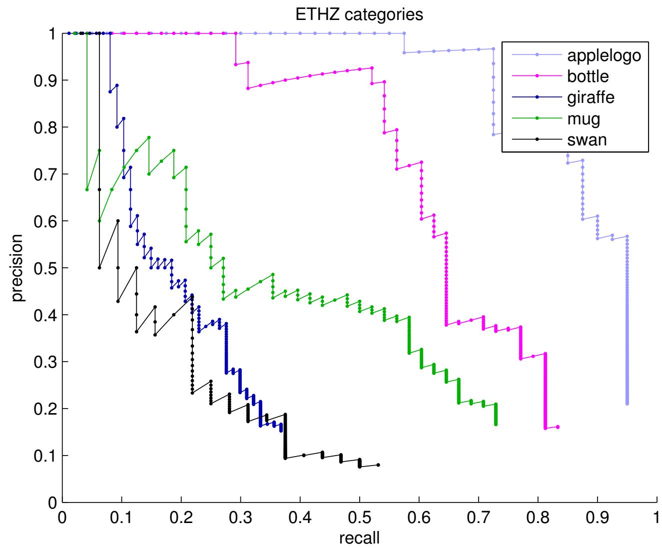


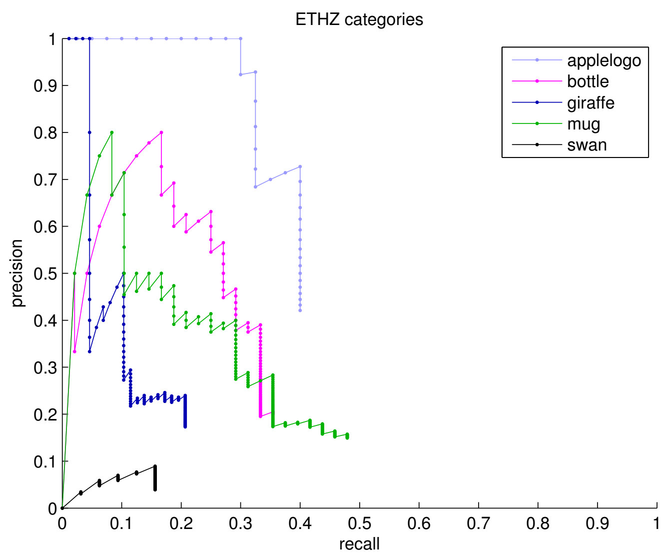
Figure 8.2: Examples of prototype bounding boxes for each model. Models are shown in their line segment representation.

performance would be systematically lower due to smaller bounding boxes.) As shown in Figure 8.2, this is accomplished by associating a full object bounding box (a prototype of BB_d) with each model via its parts. Each part stores an estimate of BB_d relative to itself, and detected parts yield the actual BB_d via the average of the estimates. The estimates of BB_d themselves are obtained from training data by taking relevant model detections and finding the average transformation from part bounding boxes to the ground truth bounding boxes. Note that bounding boxes are used only for evaluation purposes, and are learned after training the object models.

Results in Figure 8.3 show significant differences in performance over the 5 categories. The best precision and recall was achieved on the Apple Logos, which often appear nicely segmented with only slight changes in orientation, while the worst-performing were the Giraffes and Swans, which come with articulation and shape deformation. Annotation on novel images presents a significant challenge as shown in the performance decrease. To demonstrate the advantage of shape features, we present a comparison with the SIFT



(a) Training performance over all images.



(b) Annotation performance on 50% of images.

Figure 8.3: Precision-recall of ETHZ shape categories under two conditions: (a) performance on training images using the entire dataset, and (b) performance on novel images by models trained on 50% of the dataset.

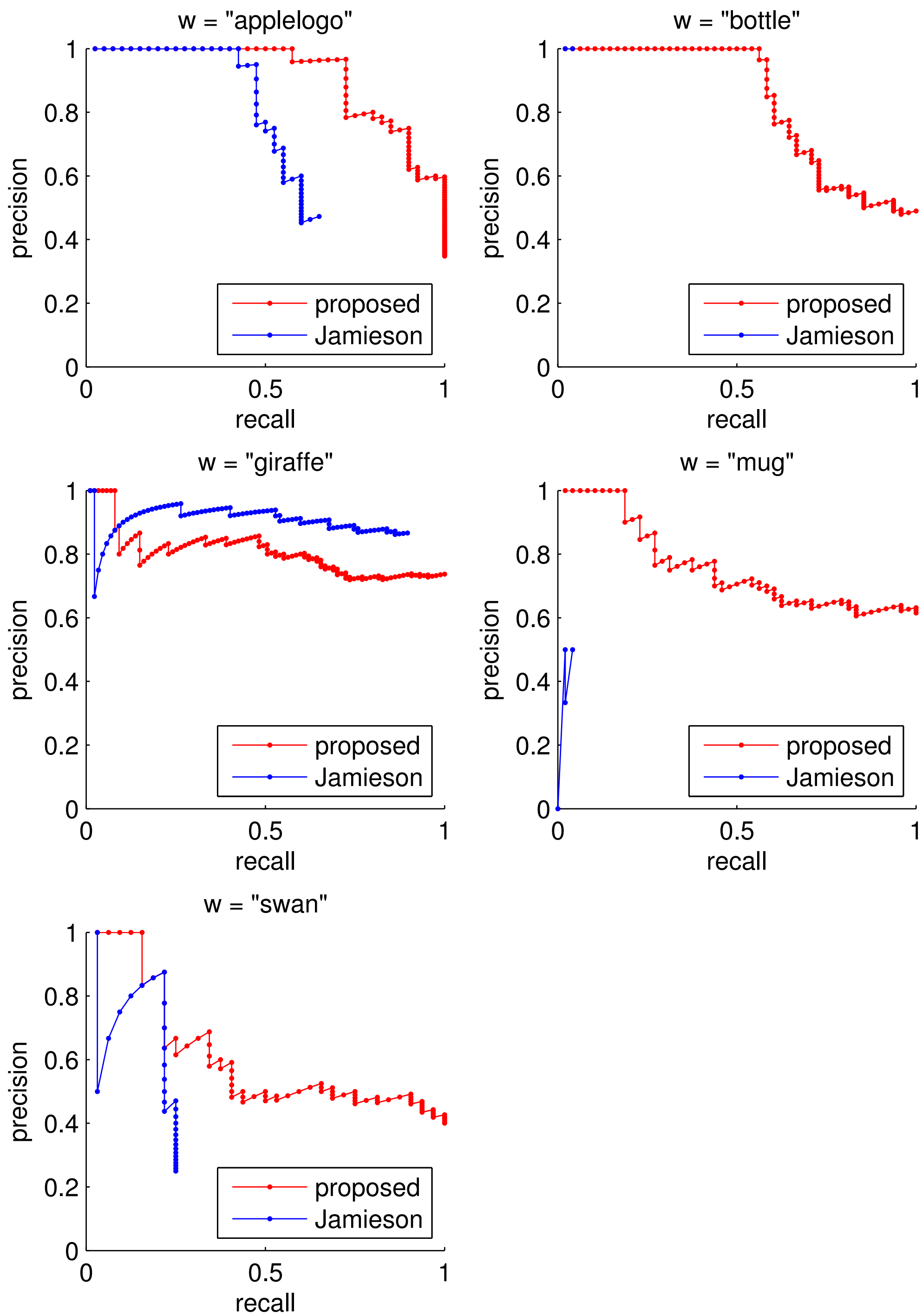


Figure 8.4: Comparison with Jamieson *et al.* [11] under model-word co-occurrence (no localization), over the entire ETHZ dataset.

feature model of Jamieson *et al*. [11] in Figure 8.4. As expected, higher co-occurrence is achieved with shape features. This is especially evident for Bottles and Mugs, where appearance features could not be found that were stable across images. It is interesting to note that among the learned appearance models were company slogan text for Apple Logos (which did not appear with each logo); skin texture for Giraffe images (which had higher co-occurrence than our shape model); and water ripples for Swan images.

Results in Figure 8.5 show that adding bounding box supervision to all stages of learning improves the performance of every category, though only marginally. Insight into this may be gained by considering our bottom-up segmentation constraint, which is sufficiently strong to yield regularities that already appear within bounding boxes. Furthermore, the closed region specified by a bounding box is only a stronger form of proximity grouping already inherent in our learning approach.

Future tests: It would be beneficial to know how well our approach performs under such supervised conditions, hence we intend to compare with the state-of-the-art in the immediate future. We expect to only approach the performance of Ferrari *et al*. [9] due to the absence of a more refined shape model, but achieve similar results compared to their first stage of learning, which performs only Hough voting over kAS features. Additionally, by depriving the state-of-the-art of bounding boxes, we expect to show comparatively better performance, as well as demonstrate the reliance on bounding boxes in other systems. Finally, a direct comparison with kAS features is also necessary to evaluate our descriptor stability improvements.

Limitations of our approach: Since expansions derive from the detections of predecessor models, final performance is dependent on the performance of predecessors. While initial models are expected to yield a precision increase as they grow to be more distinctive, an initial low recall can be a limiting factor for subsequent expansions. Low recall may be due to missing features (*e.g.*, image clutter or occlusion causing contours to break), but there is generally a deficiency in capturing within-category and viewpoint

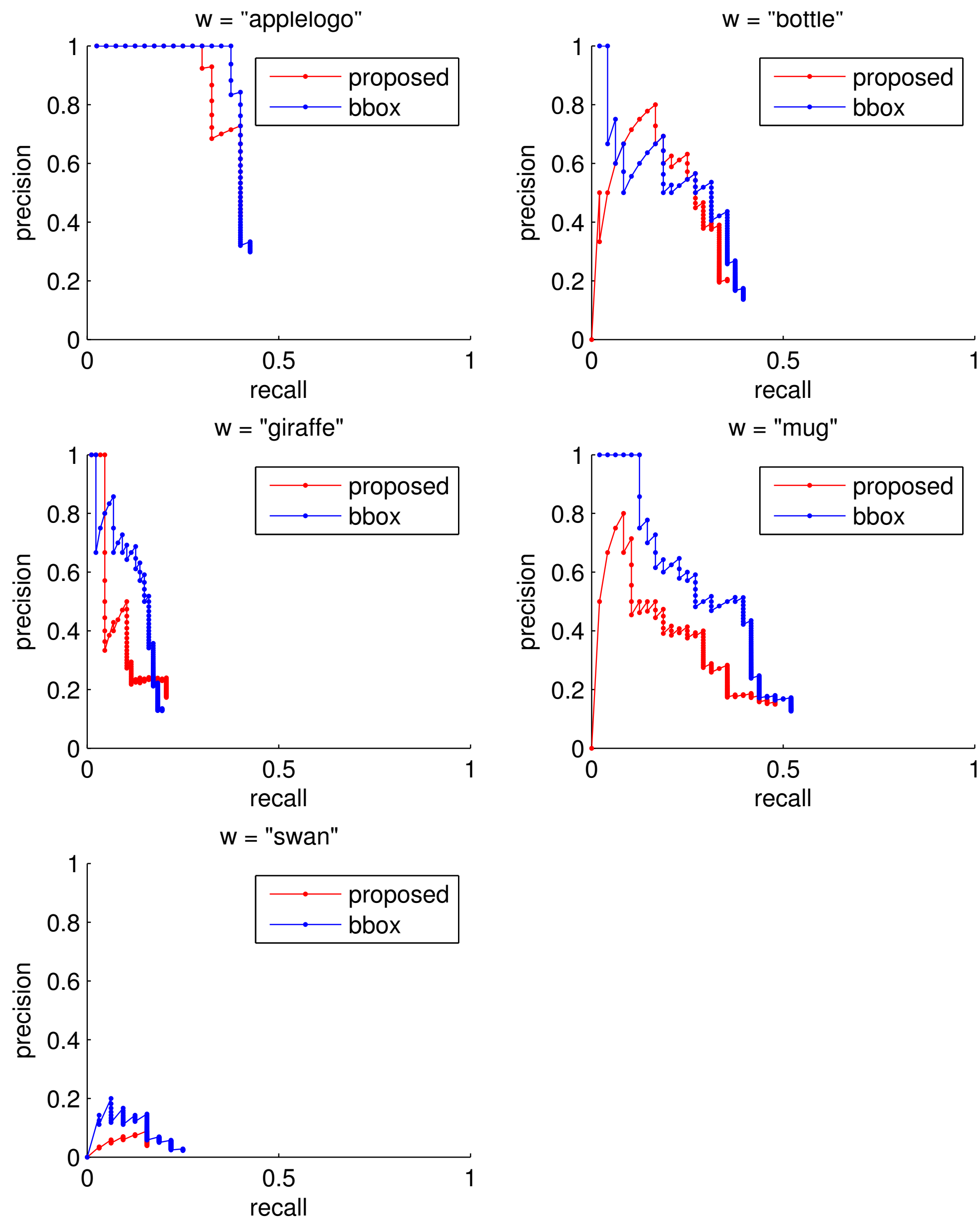


Figure 8.5: Comparison with the addition of bounding box supervision during training.

variation. Unexpected feature representations can arise from shape deformation or slight changes in viewpoint that transform image curvature with unstable contour linearization. Breakpoints may jump from place to place, causing lines to suddenly merge, split, or cover substantially different contour segments. We have addressed a subclass of these variations via multi-scale linearization (Section 4.1.1), thus increasing feature repeatability across a range of scales, although it was impractical to cover instances appearing at very small scales (*e.g.*, Apple Logos).

The shape representation is prone to low precision in the following way. Because contour fragments are spatially interrelated only via their midpoints, there is weak continuity between the ends of multiple contour fragments. This is true at the model level, where the relation between two matching contour features may be numerically similar even though the two contours are mutually discontinuous (*e.g.*, due to false positive features from clutter). Similarly, weak continuity exists at the feature level, where line segments do not necessarily coterminate, despite our efforts to restrict configurations to paths (Section 4.1, and details in Appendix B.1).

Lack of precision is ultimately mitigated by the distinctiveness of final models, but the greedy nature of the learning procedure requires that smaller, intermediate models also satisfy a certain level of distinctiveness. Preliminary tests with less distinctive, rotation-invariant features show that an initial model of only two parts, $M^{(0)} = (\{f_1, f_2\}, S_{f_1,f_2})$, may not provide enough precision for weaker features, and thus hinder learning by propagating the same problem to the next expansion. Note that our inclusion of bottom-up segmentations relaxes the initial need for distinctiveness because they help bring out the regularities. Even so, this was insufficient for rotation-invariant features, and so it would have been necessary to use other approaches such as grouping more line segments together (with $k > 3$) to obtain more distinctive features, or reducing greediness by initializing and expanding larger chunks of parts.

Chapter 9

Conclusions and future work

We have presented an approach for learning category shape from captioned images in an unsupervised manner, and demonstrated encouraging results on the ETHZ dataset. By using shape information from bottom-up segmentations, we achieve a natural and powerful alternative to supervised learning with bounding boxes, without being limited by segmentation accuracy. We have also introduced a contour feature derived from the kAS feature with stability improvements, and shown that shape features are more effective than appearance features for object categorization.

An immediate priority is a thorough quantitative evaluation. To summarize the future tests outlined in Section 8, we intend to compare performance with the state-of-the-art under commonplace supervised conditions (*e.g.*, Ferrari *et al*. [9]), and also expect to show superior performance when the state-of-the-art is deprived of bounding box supervision. A direct comparison with kAS features [9] is also needed to evaluate our contour features.

Finally, a variety of possible extensions follow from the multiple approaches that we integrated:

- A number of additional perceptual grouping methods can be integrated, *e.g.*, symmetry, repetition, and continuity. In particular, continuity alone can be exploited for a more focused expansion guide along the boundary of an object.

- The word representation can be made more flexible, *e.g.*, by allowing objects to have multiple names (possibly a hierarchical description), and using context to disambiguate words with multiple meanings. However, a more integrated learning algorithm would be necessary to carry out the simultaneous grouping of words and visual features.

- Recognition performance could be improved by incorporating a refinement stage, such as that by Ferrari *et al*. [9]. While kAS-like features allow for efficient discovery of regularities, a finer and more concise object representation would be less sensitive to within-category variation.

- The overall design and performance could potentially be improved by reducing greediness in both learning and detection. While greedy algorithms are optimal for problems that have a greedy structure, it is not clear that this assumption applies or is necessary to the current extent for this task.

Appendix A

Probability model parameters

A.1 Co-occurrence score

The likelihood that a word w occurs when the object is present in the scene, $p(w|o = 1)$, can be determined from ground truth data if available, otherwise it may be specified per-word. The likelihood when the object is absent from the scene, $p(w|o = 0)$, is similarly determined. We use $p(w|o = 1) = 0.99$ and $p(w|o = 0) = 0.01$.

The likelihood of a model instance m given object presence in the scene is more complicated to determine. Since $m \in [0, 1]$, the likelihood $p(m|o)$ is continuous and may not be easy to obtain from ground truth data. However, we can specify the endpoint probabilities, $p(m = 1|o)$ (and thus $p(m = 0|o) = 1 - p(m = 1|o)$), as was done for words, and then assume a linear interpolation between them (and checking that the density sums to 1). This linear relation reflects the expectation that when the object is in the scene, high confidence detections are more likely than low confidence ones, while the opposite is true when the object is absent.

A.2 Detection score

The background feature likelihood $p(\phi|f_B)$ represents the natural frequency of occurrence in background images. While the ϕ exists in the space of all possible contours, some contours occur more frequently than others, while others occur more rarely and are thus more distinctive. For simplicity, however, we have approximated the density with a uniform distribution. A more accurate distribution could have been obtained by considering the variances of individual codewords.

The background mean distance μ_{bu} and mean relative scale μ_{bv}, and their variances $\sigma^2_{\mathrm{bu}}, \sigma^2_{\mathrm{bv}}$ are determined empirically by sampling from feature neighbourhoods. The prior probabilities of the object and background models, $p(M)$ and $p(B)$, respectively, are determined empirically from training data by counting word occurrence.

Appendix B

Computational details

B.1 Line segment ordering

Two issues arose in the restriction of line segment grouping to paths, whose motivation is given in Section 4.1. First, our line segment ordering is canonical, but only up to the two path directions. Contour similarity needs to be invariant to these two possibilities, so we compute similarity twice for each pair of features (ϕ_1, ϕ_2), once with respect to the forward encoding of ϕ_2, and once with respect to the backward encoding of ϕ_2. The direction is disambiguated by choosing the highest of the two similarities. Both encodings are pre-computed for each feature. (For rotation-invariant features, the spatial relation encoding between ϕ_1 and ϕ_2 depends on the ordering of both features through θ_1 and θ_2. Disambiguation of the relation encoding is provided by the ordering of the respective common codewords.)

Secondly, we were unable to group line segments directly from the linearization since our kAS software version did not document internal data. Instead, we restricted grouping to paths by taking a subset of extracted kAS features. The "pathness" of a kAS feature is a soft notion because line segments are linked rather than coterminating, and line segments were often found to be only "almost" coterminating. We determined the "pathness" of a configuration of line segments by finding the best path through the segments, where the total gap length required to bridge the path across segments was minimized. Each kAS feature was given a continuity score between 0 and 1 based on the total gap length, and the feature subset was chosen by thresholding the score at 0.75.

B.2 Codebook construction

Our codebook is constructed by finding clusters of similar contour features from background images (all images in the ETHZ dataset). Features are clustered using the K-means algorithm, and each cluster is represented by the centre-most member in kAS distance. The target number of clusters is set to $K = 700$. We found that the exact choice of K had little effect on performance; however, we manually examined codewords to verify that their neighbours were visually neither too similar nor too dissimilar.

By using K-means clustering we make the assumption that our contour features are comparable with Euclidean distance between descriptor vectors (Equation 4.1). Due to the circular angle space and component weights in the kAS distance function, however, more accurate clustering results could have been obtained with kAS distance directly, *e.g.*, with spectral clustering or a clique partitioning approach.

B.3 Distance computations

In a typical image with thousands of contour features, distance computations for neighbourhoods ($d_{\text{nbh}}(\phi, \phi')$), overlap, and boundary hypotheses masks ($d_{\text{bnd}}(B, \phi)$) are expensive. The number of distance computations is quadratic in the number of features, and each distance computation is quadratic in the number of their respective edgels. We compute distances efficiently by assigning contour edgels to spatial bins corresponding to a grid over the image. Pairwise distances between bins are pre-computed, effectively creating a look-up table for edgel distances.

Bibliography

[1] K Barnard, P Duygulu, D Forsyth, N De Freitas, DM Blei, and MI Jordan. Matching words and pictures. *The 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. *CVPR*, 2003.

[3] I Biederman and G Ju. Surface versus edge-based determinants of visual recognition. *Cognitive Psychology*, 20(1):38–64, 1988.

[4] D Crandall and D Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. *Computer Vision–ECCV 2006*, pages 16–29, 2006.

[5] P Duygulu, K Barnard, J De Freitas, and D Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *Computer Vision—ECCV 2002*, pages 349–354, 2002.

[6] R Fergus, P Perona, and A Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *International journal of computer vision*, 71(3):273–303, 2007.

[7] V Ferrari, T Tuytelaars, and L Van Gool. Object detection by contour segment networks. *ECCV Proceedings*, pages 14–28, 2006.

[8] Vittorio Ferrari, Loic Fevrier, Frederic Jurie, and Cordelia Schmid. Groups of adjacent contour segments for object detection. *IEEE PAMI*, pages 1–16, Nov 2008.

[9] Vittorio Ferrari, Frederic Jurie, and Cordelia Schmid. From images to shape models for object detection. *IJCV*, 87(3):284–303, May 2010.

[10] C Gu, JJ Lim, P Arbeláez, and J Malik. Recognition using regions. *CVPR Proceedings*, 2009.

[11] M Jamieson, A Fazly, S Stevenson, S Dickinson, and S Wachsmuth. Using language to learn structured appearance models for image annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):148–164, 2010.

[12] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories using link analysis techniques. *CVPR 2008*, 2008.

[13] Y Lee and K Grauman. Shape discovery from unlabeled image collections. *CVPR 2009*, 2009.

[14] A Levinshtein, C Sminchisescu, and S Dickinson. Optimal contour closure by superpixel grouping. *Computer Vision–ECCV 2010*, pages 480–493, 2010.

[15] D Martin, C Fowlkes, and J Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE PAMI*, pages 1–20, Jul 2004.

[16] A Opelt, A Pinz, and A Zisserman. A boundary-fragment-model for object detection. *Computer Vision–ECCV 2006*, pages 575–588, 2006.

[17] B Russell, W Freeman, A Efros, J Sivic, and A Zisserman. Using multiple segmentations to discover objects and their extent in image collections. *CVPR*, 2:1605–1614, Apr 2006.

[18] J Shotton, A Blake, and R Cipolla. Contour-based learning for object detection. *ICCV Proceedings*, pages 1–8, Jul 2005.

[19] J Sivic and A Zisserman. Video google: Efficient visual search of videos. *Toward Category-Level Object Recognition*, pages 127–144, 2006.