Class-Specific Object Segmentation using Top-Down Attribute Support

Aron Yu
Department of Electrical Engineering
University of Texas at Austin
aron.yu@utexas.edu

David J. Garcia
Department of Computer Science
University of Texas at Austin
djchooy@utexas.edu

Abstract

Object localization and image segmentation can be considered as part of the same problem: dividing up an image in a meaningful way. However, there is a "semantic" gap between top-down semantics and bottom-up segmentations. Top-down approaches attempt to detect objects and to fit object-specific templates/statistics to image regions. Bottom-up approaches use local features to inform the segmentation by joining groups of similar pixels. In our approach, we use learned attributes to aid a bottom-up segmentation engine. Using attributes enables us to apply our method to a variety of object classes, including ones not present during attribute training. While some approaches consider bottom-up segmentation as a "preprocessing" step in a segmentation pipeline, we perform the segmentation while considering detected attribute locations. We show significant improvement over a bottom-up baseline segmentation routine.

1. Introduction

Semantically, it's intuitive to describe objects by their attributes [**?**]. Describing objects by their attributes has the additional benefit of extending, not only across members in the same object categories, but also across the categories themselves [**?**]. For example, members of the categories "Farm Animals" and "House Animals" have the "Legs" attribute. This idea might provide valuable context to the task of both object presence detection as well as object segmentation in an image. It may be possible to, without knowing the category, to describe an object with its attributes and potentially localize it in an image.

However, it is important to keep in mind that each object category has its own distribution of attributes. Horse and sheep have legs, for example, but unlike a sheep, a horse is not fluffy. Likewise, a wooden cart and a car both have wheels, but the wooden cart is not shiny. These problems can be mitigated with strong negative examples in training [**?**].

Another potentially complicating factor is the "Semantic Gap" between image content and the semantic meaning of that content [**?**]. For example, suppose the category "Bear" has two types of attributes: "Fuzzy" and "Eats Fish". The former attribute can be easily represented with a texture (a mid-level feature). The latter, however, is very difficult to represent with either low-level features. Typically, these types of problems are only tractable as a discriminative learning problem with a respective positive/negative training-set.

The typical end game motivation for these types of tasks is to accurately, and precisely, segment object/s of interest in images. There are various applications for this capability. Obviously, knowing **where** an object is, in an image, is related to knowing that an object is present in the image in the first place. Said another way, knowledge of object presence is a necessary, but not sufficient, condition for object localization. Object tracking, for example, requires extraordinarily precise location information [**?**]. Likewise, scene understanding benefits enormously from localization of constituent objects. A "gathering" activity necessarily requires knowledge of individual person locations in order to infer that they are converging on a central location.

....illustration...

Many object localization techniques have a pipeline that starts with low-level segmentation to produce superpixels/regions. Top-down information, usually obtained from a trained classifier, associates these mid-level features with each other to produce a final segmentation. The initial segmentation is little more than a preprocessing step: a process to produce a higher level feature that is easier to work with. This approach has the obvious advantage of not having to rely on some trained machine to do the low level processing. We investigate how one could use top-down information in this initial preprocessing step. Specifically, we look at how attribute location priors could inform bottom up image segmentation. We adapt the bottom-up parametric graph cuts technique to take into account the color distribution indicated by the attribute locations in the image [**?**]. Given good attribute location priors, we feel that an initial segmentation

is significantly improved. We also demonstrate the degradation in performance as attribute accuracy degrades respectively.

Include and overview of our method, accompanied by a figure.

1.1. Related Works

Our approach is the most similar to [?]. We investigate modifying the data-cost term by incorporating the location of detected attributes. Unlike the authors in [?], we propose that foreground seed placement can have a significant impact on the initial segmentation results. The work done by [?] classifies superpixels based upon relative location priors of the objects in the images. For example, if "cows" are typically "above" "grass", then there is a relatively high probability that a cow will be above grass in an image. Relative location priors are taken into account after superpixels are "initially" classified using lower level features. Essentially, the location priors are a refining technique.

In the work done by [?], the authors observe the potential for attributes to cross categories and thus suggest that a general classifier can be built with minimal training. [?] also distinguish between **Semantic Attributes** and **Discriminative Attributes**. The former captures ideas like shape (i.e. "boxy", "cylindrical", etc), part (i.e. "has head", "has leg",etc), and material (i.e. "has wood", "is furry", etc). The latter type of attributes is meant to enable discrimination between classes. Cats and dogs, for example, have many common semantic attributes. These attributes represent splits that effectively split classes from each other. In the work of [?], the authors show that there are important "Compositional Factors" that can bias where salient objects are in the image. [?] considers compositional factors to be the size and location of the salient object. Since our experimental goal is to essentially infer a foreground in images, our task is not unlike the task in [?]. We just use a standard color distribution for our features.

Of special note is the work of [?]. The authors model image segmentation as a four-way graph cut problem with respect to K number of classifications any one pixel can take. The choice of pixel classification is essentially reduced to min-cut/max-flow problem. The optimal classification is the one that results from removing links that, when added together, have the lowest energy possible. Also of note in this work is restriction on the data cost function D_p. This function captures how well the label f_p fits label p given the observed data. In our case, the observed data is simply the color of the pixel with respect to foreground and background color distributions. Again, this is very similar to [?]. The entire energy function that is to be minimized is of the form:

$$\mathrm{E}(f^C) = \sum_{p \in \mathrm{P}} D_p(f_p^C) + \sum_{\{p,q\} \in \mathrm{N}} V(f_p^C, f_q^C)$$

f_p^C is the label that we give to pixel p for the minimal graph cut C. $V(f_p^C, f_q^C)$ is the function that describes the pairwise labelings between any two pixels p and q. It is understood that N only contains edges between adjacent pixels.

2. Top-Down Attributes

Explain how we're using attributes - trained attribute detectors causes pixels to respond - cluster the pixels into key attribute points

2.1. Attribute Detectors

Talk about the attribute detector from the Farhadi paper. - works pretty well on localized objects, finding unusual attributes, etc Briefly talk about the base features and the semantic features used - attempt at avoiding attribute transfer

Manually selected the attributes for each class - picked only semantic attributes - list out the attribute for each class (maybe a table) with reference to the dataset section

2.2. Attribute Points

Back traced the locations of each attributes - minor details about down sampling the color features to be fair Clustering the feature locations Show good sample images of the back traced features Note that the detectors are not meant for localization - divided into grids that suppose to benefit from object placement - trained on localized objects - talk about dominance of the color features

Simply took the raw result points from the attribute detector and tried to detect without using part detector like [1].

Measure the quality of attribute localization by counting the number of dots that lands on the objects.

2.3. Performance

Show images of the attribute points Problems with approach, no the best - not completely decoorelated but we can deal with the result since we're not looking for perfect localization

Show images of objects detected using wrong set of attributes

3. Bottom-Up Segmentation

Use seeds as starting point

3.1. Graph Cut

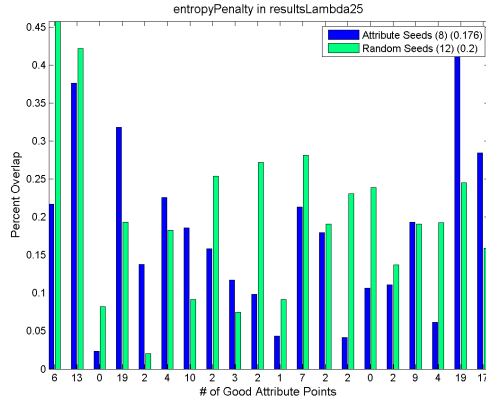In section **1.1** we briefly discuss the minimization of

Figure 1. Scores for Cars using a bias of 25.

$$\mathrm{E}(f^C) = \sum_{p \in \mathrm{P}} D_p(f_p^C) + \sum_{\{p,q\} \in \mathrm{N}} V(f_p^C, f_q^C)$$

For simplification, we define the pixe

Explain the use of Graph Cut for segmentation - use of starting seeds, obtaining foreground/background color histogram - various biases

3.2. Cost Functions

The formation, rationale behind our choices
Sample Equation

$$\mathrm{L2\ Score}(u_s) = T_s \left[\sum_{k \in \mathrm{followers}} P(u_s \text{ influences } u_k) \right]$$

3.3. Raw Segmentation

not doing ranking and fast elimination sample images of the segmentation results

4. Experiments

Testing with 3 data cost functions and 4 background bias values for each class and then compute the overlapping score with ground truth Trying to see how "good" attribute points can help with segmentation

4.1. Dataset

Used VOC2008 dataset (cite the dataset) - because that's what the attribute detectors were trained on - used the segmentation dataset as testset due to the availability of ground truth - tried to pick ones with only 1 class in the image - formed a smaller 6-class dataset with 20 images per class

Method	Frobnability
Theirs	Frumpy
Yours	Frobbly
Ours	Makes one's heart Frob

Table 1. Results. Ours is better.

4.2. Score Evaluation

Compute the overlap score, intersection over union, of the segmentation and the GT Since we don't have perfect segmentation, we look at the relative performance

Bulk of the discussion - compare the 12 sets of results, which one is better, why and why not, etc - show score bar graph comparison - consider the number of "good" attribute points

4.3. Extra Cases

Good Attributes
Manually select the images that achieved good attribute points from all classes - evaluated the overall result
Multiple Objects
Tried with images containing multiple objects just to see result - objects from different classes

5. Discussion

conclusion - we've shown that good attribute points leads to better segmentation - better attribute detector would lead to better results

future work 1) Better attribute detector 2) Try cross-category generalization 3) Implement the full system 4) Try different cost functions, besides color histogram 5) Retrain attribute detector with less inter-class correlation

5.1. Illustrations, graphs, and photographs

All graphics should be centered. Please ensure that any point you wish to make is resolvable in a printed copy of the paper. Resize fonts in figures to match the font in the body text, and choose line widths which render effectively in print. Many readers (and reviewers), even of an electronic copy, will choose to print your paper in order to read it. You cannot insist that they do otherwise, and therefore must not assume that they can zoom in to see tiny details on a graphic.

When placing figures in LaTeX, it's almost always best to use \includegraphics, and to specify the figure width as a multiple of the line width as in the example below

```
\usepackage[dvips]{graphicx} ...
\includegraphics[width=0.8\linewidth]
               {myfile.eps}
```

References

[1] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010.