Object Detection Using Segmented Images

Naran Bayanbat
Stanford University
Palo Alto, CA
naranb@stanford.edu

Jason Chen
Stanford University
Palo Alto, CA
jasonch@stanford.edu

Abstract

Object detection is a long standing problem in computer vision. One of the common approaches to object detection is to train with segmented images. Intuitively, isolated foreground images should provide better training sets and improve the performance of the detection system. However, in practice, there are challenges associated with using segmentation for training data. Features located along segmentation borders in the training image assume for clean backgrounds, which makes the resulting detection not robust to noise in realistic images.

In this paper, we propose a way of excluding such features in order to obtain a generalized object detector that can perform cluttered background images by training on segmented images. We demonstrate the effectiveness of this approach by comparing the performance of our detector against that of a detector trained using ordinary, "dirty" background images.

1. Introduction

Object detection is one of the oldest problems in computer vision. One of the simplest ways to approach this problem is to training on a set of closely cropped images of the object. However, this approach has the drawback of including background pixels that are not necessarily a relevant part of the object itself. The image noise tends to degrade the performance of the detector. An alternative to this approach is to use segmentation to tightly crop out the object in the training set images and effectively remove the background clutter. As long as a set of clean images were chosen, it is possible to achieve a tight, accurate segmentation using the state of the art segmentation methods.

Unfortunately, there are also drawbacks associated with using segmented images. Features that are detected along the frame of the object, on its outer edges and corners, detract from the performance of the detector. Training on these features tunes the detector to clutter-free backgrounds. When applied to normal, noisy environment images, the detector performs poorly, even compared to a classifier that was trained on non-segmented images.

Our approach is to identify features (SIFT descriptors) that are located around the perimeter of the object segment, and exclude these from the training data. We hypothesize that removing the detracting features will improve the overall performance of the detector.

For the training set, we will use ImageNet to find clutter-free pictures of the object to train on. ImageNet has pre-categorizes images into "synsets," or semantic categories, that averages about 1000 images per category. A significant portion of the images on ImageNet are clean background images that will allow us to segment out the object easily.

We will evaluate our performance by training a SVM classifier on the modified set of features, and measuring the precision and recall rates of detection on a test set images. Additionally, we will also evaluate the performance of a classifier trained on "dirty", or non-segmented images, and provide plots of the performance metrics of both detectors for comparison. We hypothesize that our modified detector will outperform the "dirty" classifier.

1.1. Related Work

For the task of object detection, bag-of-features methods have been successfully applied in many instances. These approaches reduce an image into a collection of local features without preserving the geometrical structure of the underlying objects. Their application has largely been successful, allowing them to outperform more sophisticated methods that preserve the structure of objects [1, 2, 3]. However, they carry limited descriptive data, and are unable to segment an object out of its background. By first applying segmentation, then excluding the "bad" features, our approach seeks to improve on the bag-of-features methods.

2. Approach

Our initial approach is to choose a category of objects, and create a training set consisting only of clean background images of that category of objects. We run a segmentation algorithm (normalized cut) to isolate out the object from the background. For each image, SIFT features that are outside of the object's segment is filtered out, and the remaining

SIFT descriptors are used to compute a Bag-of-Words (BoW) histogram for that image. Mean shift clustering is used to partition the SIFT features into bins and build the BoW dictionary. Once the BoW histograms are computed, a SVM classifier is run on the training set to perform supervised learning.

To test the performance of our approach, we run the classifier on a separate test set of images. We will similarly compute the histogram of these images, and collect SVM's predictions. We use the resulting data to compute a precision and recall value for the performance of the classifier.

2.1. Data

We use images provided by Image-Net.org. Specifically, we chose one particular synset as our evaluation subject ("n04398044", Teapot) and one unrelated synset ("n03376595", Folding Chairs) as noise.

Pre-computed SIFT features are available for 1590 images in the Teapot synset, and 1537 images in the Folding Chair synset. We will use these SIFT descriptors to describe our images.

2.2. Normalized Cut Segmentation

We use a Normalized Cut algorithm made available for research use [Cour]. The algorithm first resizes the image to a smaller, manageable size (up to 240 pixels on one side), than used the normalized graph cut algorithm to segment the image. Since we are dealing with clean-background images, and there is only one subject on each of these images, we only output one cut (two segments) from the algorithm. The result is a matrix of segment labels for each pixel on of the image. Our heuristic for identifying the segment containing the object is simple – we partition the image first, and use the corner pixels to identify the background segment.

2.3. Bag-of-Words Histogram

To compute the Bag-of-Words (BoW) histograms, we first compute the vocabulary feature set. We do so by using mean-shift clustering over all the features from all the training images. Each cluster centroid represents a visual word. Then, for each image, we compute the BoW histogram over all the features in the image, using Euclidean distance to find the closest visual word to each feature. We use this histogram as descriptor for each image for training and testing.

2.4. Training and Testing

For our investigation, we use a simple SVM classifier to evaluate our hypothesis. We initially compute BoW histograms using the filtered SIFT features of an image. Each histogram from our subject synset (in this case, "Teapot"), is a positively labeled sample in our SVM data space. We also obtain images from unrelated synsets, compute BoW histograms for this set, and use the resulting histograms as negative training samples.

We compare this approach to one where BoW histograms in the training are computed over all the SIFT features, i.e. an approach that does not employ segmentation. For both cases, we compute the precision and recall values and compare the results.

3. Current State

3.1. Segmentation

Normalized cut performs under expectation even on images with clean backgrounds. A valid segmentation is necessary to filtering the SIFT descriptors and circumvent some of the challenges of using a clean background image for object detection.

Some examples of segmentation results are shown in Figure 1. Image in (a) displays an example of a successful cut. However, in many cases, segmentation returns undesirable results. In (b), the object itself is partitioned into two parts as parts of the object and the background almost blend together. In (d), segmentation is only partially successful, as it successfully partitions along the object boundaries, although the result is not what we expected. A potential way around this is to segment the image into more than two partitions, although that approach has challenges of its own.

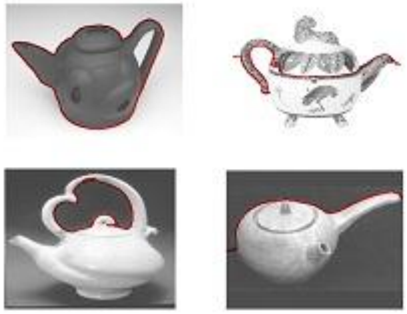

Figure 1 Segmentation results. From top left, clockwise, (a), (b), (c), (d).

3.2. Mean Shift Clustering

For mapping the SIFT features into BoW histograms, we decided to employ mean shift clustering as the number of clusters that the SIFT descriptors should be divided into was not immediately clear. In order to achieve a desirable number of clusters (too few would underdescribe while too many would overfit), we tuned the window size of the mean shift clustering to 0.7, resulting in 349 clusters.

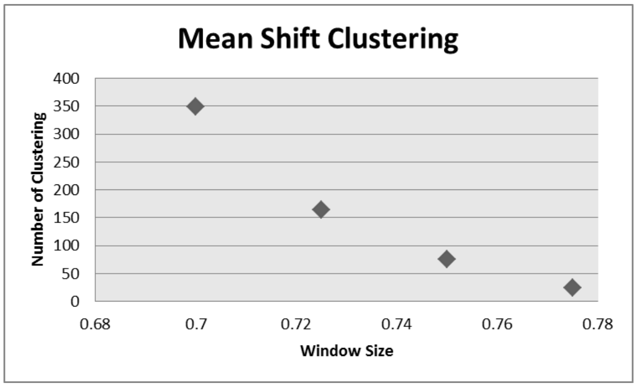

Figure 2 Mean shift cluster parameter tuning

4. Future Work

4.1. Improving the Model

Because Normalized Cut has less-than-satisfactory performance even when used on our clean-background images, we will investigate other options such as GrabCut [4]. However, since segmentation is a premise and not a goal for this investigation, we will use pre-segmented images and assume it can be done beforehand.

4.2. Detection

After improving the performance of our classifier, we would attempt to use a weak spatial model to detect the location of the target object in the test image similar to Spatial Pyramid Matching [1].

To achieve this, given a test image, we divide the image into grids of various scales. For each region at each scale, we compute the BoW histogram. We then append each histogram into an N-by-K test matrix, where K is the number of words in the vocabulary, and each row represents the histogram at a certain scale. Note N will always be 2-^{0}+2^{1}+...+2^{L}, where L is the total number of levels we are taking.

We hypothesize that the object of interest in the test image will match with one of the training image at some scale. By leveraging the fact that our training image contains no "distraction" and all features are from within the object of interest, we can achieve scale invariance and find a coarse bounding box of the object in our test image.

Future Distribution Permission

The author(s) of this report give permission for this document to be distributed to Stanford-affiliated students taking future courses.

References

[1] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, 2006.

[2] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *Proc. ICCV*, volume 1, pages 257–264, 2003.

[3] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *ICPR Workshop on Learning for Adaptable Visual Systems*, 2004.

[4] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. In ACM SIGGRAPH 2004 Papers (SIGGRAPH '04), Joe Marks (Ed.). ACM, New York, NY, USA, 309-314. DOI=10.1145/1186562.1015720 http://doi.acm.org/10.1145/1186562.1015720

5. Appendix

This project is part of the ImageNet research effort in providing a large-scale image database for researchers and educators around the world. The detection system developed in this paper aims to improve the tagging of relevant portions of the images in each semantic category ("synsets").