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 categorization, 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. This method is computationally efficient and 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, containing no geometric or part information, and are unable to distinguish an object from its background. By first applying segmentation, then excluding the "bad" features, our approach seeks to improve on the bag-of-features methods.

2. Approach

In this paper, we choose several unrelated categories of objects, and create a training set consisting only of clean background images from each category of objects. We then process the images to isolate out the object of interest from its 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. A clustering algorithm is used to partition the SIFT features into bins and build the BoW dictionary. Once the BoW histograms are computed, a multi-class 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, consisting of images from the same categories as the training set. The test set is not constrained to clean background images. For these images, we will similarly compute the histograms, and collect the 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. We chose three unrelated synsets of objects to run train our detector. The synsets were *teapot* ("n04398044"), *revolver* ("n04086273"), and *scissors* ("n04148054"). Additionally, we picked two other synsets, *foldable chairs* ("n03376595") and *toyshop* ("n04462240") to create a training set for negatives, when none of the objects we trained on are present. We use the pre-computed densely sampled SIFT features available on all of the synsets to compute a histogram of words for each image.

2.2. Image Segmentation

We initially used a Normalized Cut algorithm implementation made available for research use [Cour]. The algorithm first resizes the image to a smaller, manageable size (up to 240 pixels on one side), then uses 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, and we configured the algorithm for one cut (two segments). The result is a matrix of segment labels corresponding to each pixel on 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.

Unfortunately, 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 normalized cut 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.

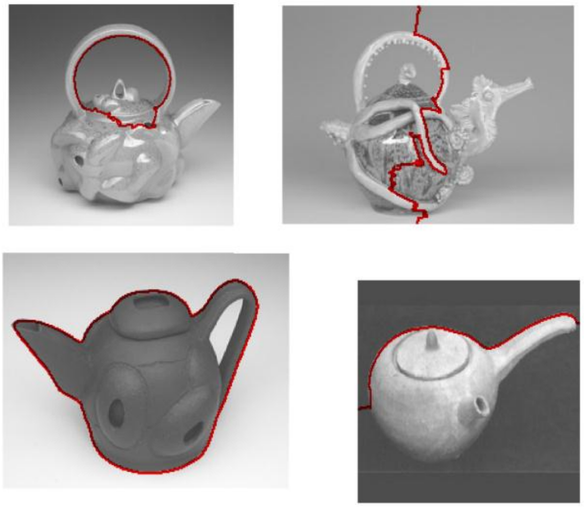

Figure 1 Normalized cut segmentation results. From top left, clockwise, (a), (b), (c), (d).

We resorted to an alternative method of segmentation that proved more effective. Instead of running a segmentation algorithm, we threshold SIFT descriptors based on their norm values. In a clean background image, higher norm values in descriptors generally correspond to points of interest, while low norm values indicate low energy, i.e. background areas or plain surface. By removing low norm densely samples features points, we consistently isolated out SIFT features corresponding to the shape of the object. See Figure 2.

A potential drawback with this approach is a loss of information regarding the texture of an object. For example, if an object is of plain texture, and this information is critical to describing the object, such as a refrigerator or a blackboard, then we would have removed critical information describing the object. However, this loss of information comes as a consequence of improved ability to detect objects of varying texture, such as teapots. In the case of teapot and many others, shape is more invariant and critical to the description of the object than texture, which changes from one instance of an object to another.

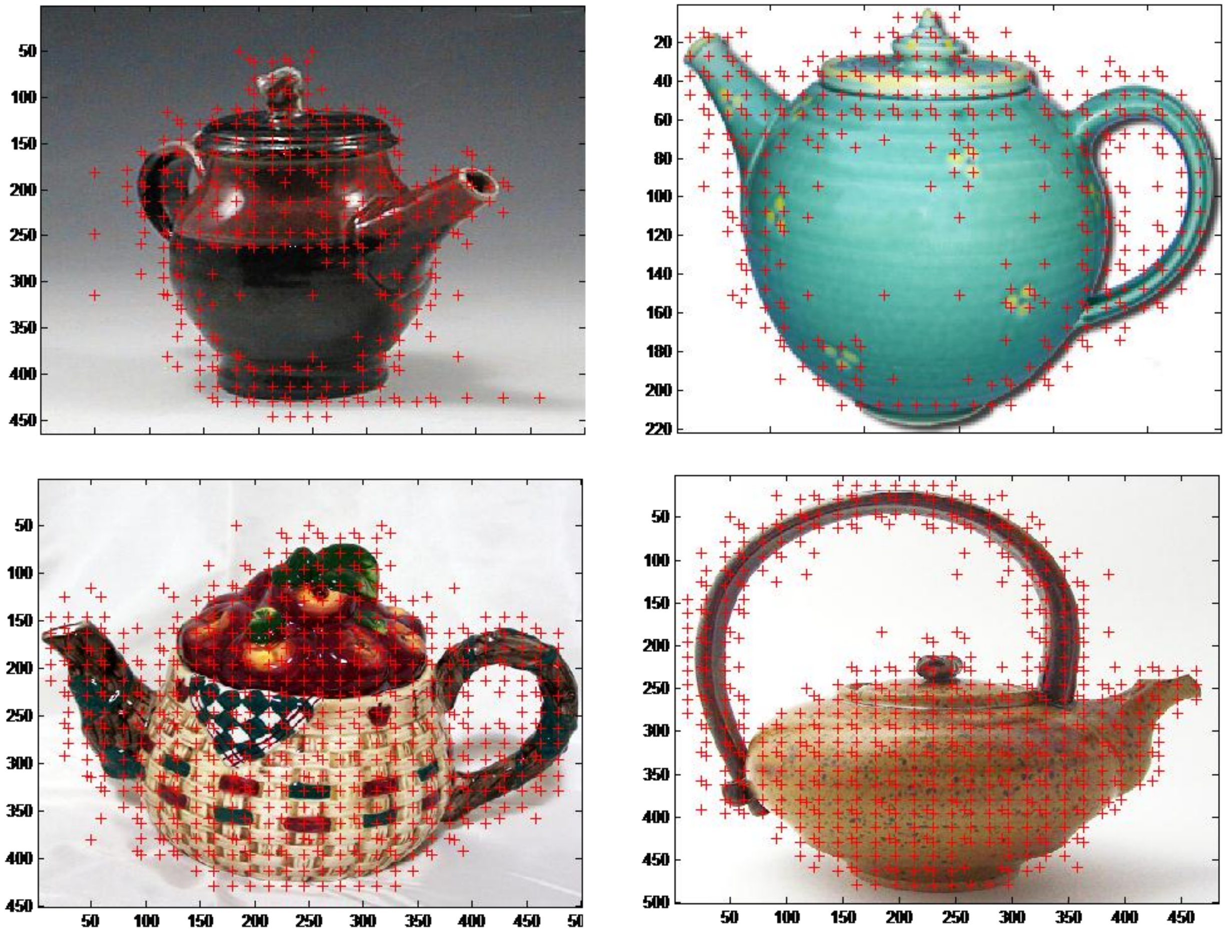

Figure 2. SIFT normal thresholding results. From top left, clockwise, (a), (b), (c), (d).

After the segmentation, we discard the part of image without SIFT features. For example, Figure 2(a) would be trimmed such that the image stretch from the leftmost SIFT feature to the rightmost, leaving a bounding box of the teapot. We believe this would improve the detection classifier as it would be invariant to the size of and amount of white space around the subject.

2.3. Histogram Computation

To compute the Bag-of-Words (BoW) histograms, we first compute the vocabulary feature set. We do so by using a clustering algorithm over a large number of randomly sampled SIFT descriptors from the training and test 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 L2 norm to find the closest visual word from each feature. We use this histogram as an image descriptor for training and testing.

We initially used mean-shift clustering to compute the vocabulary set. However, with this approach, we observed a proclivity to cluster most of the provided SIFT descriptors around a single cluster point, with the outliers assigned to significantly smaller clusters around it. We attributed this problem to the fact that mean-shift clustering creates a single cluster at a time. The first cluster will generally have more points assigned to it than the subsequent ones, therefore creating a single dominant cluster in the beginning.

Our first attempted work-around this was to remove the dominant cluster from the vocabulary, and use the rest of the set. However, this approach brought inconclusive results. The regular clusters largely corresponded to outliers in the SIFT set, and did not provide sufficient description for regular SIFT points. The resulting histograms brought poor detection results.

Consequently, we switched to using k-means clustering instead. Since k-means employs effective heuristics for initializing clusters centers (k-means++), it is less prone to creating a single dominant cluster. See Figure 3 for comparison of histograms from vocabulary that used k-means clustering against histograms that used mean-shift clustered vocabulary.

Additionally, a spatial matching scheme as presented by Lazebnik et al. was used. Each image is duplicated into multiple layers. At each layer, the image is divided into increasingly fine resolution grids, and the histograms of the image in those bins were computed. Histograms at each level were weighed appropriately (higher resolution histograms weighted greater), and concatenated into combined histograms that were used to perform supervised learning.

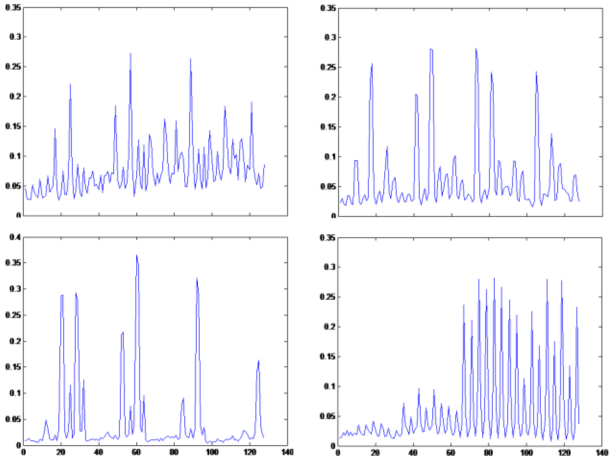

Figure 3. Example plots of cluster centroids from k-means clustering

To increase the robustness of the detection system, we also trained on each image multiple times, each time slightly shifting the object center around a 3x3 or a 5x5 grid. This produces the effect of small obstruction of the object around its edges, and allows the detection to become more robust to viewpoint obstruction.

2.4. Training and Testing

For our investigation, we use Crammer and Singer's multiclass SVM classifier to evaluate our hypothesis. We generate image histograms using the filtered SIFT features of that image. Each category of objects are defined a class. Additionally, we define one more class label for the *folding chairs* and *toyshop* synsets. This label corresponds to our detector not recognizing any of the other objects that it trained on. Effectively, this is a "no detection" label.

We use a separate test image set to evaluate the performance of our classifier. The images in the set also belong to the same categories of objects as the training set; however, they have arbitrary amounts of background clutter, and may present occlusion, rotation, scaling, illumination and viewpoint variations. The SVM predictions on the test set are matched against the correct labels of the set.

We define two other baselines to compare our method's performance. The first is to train our detector on clean background images without using object segmentation. The second baseline is to train our detector without the constraint of clean backgrounds in the training set. We predict that, in both cases, our approach will provide an improvement over the baseline.

3. Evaluation

3.1. Metric of Analysis

The main metric for analyzing the performance of our classifier is the accuracy of the multi-class SVM correctly predicting the synset from which the test images are chosen. Further, we compute the accuracy for our detection algorithm as the number of images detected with an object dividing the number of those from the synset.

Lastly, we compute rough bounding box for the desired object based on the most-confident decision value returned by the SVM model. Then we visually evaluate the result of the detection algorithm.

Note we also made the assumption that the algorithm's parameters are independent. Therefore, we can optimize the accuracy over each parameter individually.

These metrics are compared against two baseline algorithms. One, we run the same procedure except without performing segmentation as described and, two, we train only on images with cluttered background.

3.2. Clustering

As discussed in Section 2.3, for mapping the SIFT features into BoW histograms, we examined using mean-shift clustering or k-means clustering. Due to the tendency for mean-shift to produce a single dominant cluster, we decided to employ kmeans clustering for the analysis. Figure 4 shows the change in classifier accuracy as we tune number of clusters in k-means. Due to the time constraint, we could not collect more data on the changing number of clusters with k-means. Therefore, we concluded a cluster number of 300 is optimal and would be used for optimizing the other parameters.

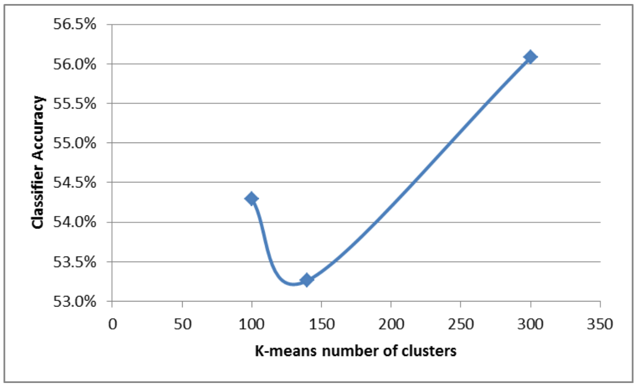

Figure 4 K-means cluster parameter tuning

3.3. Levels of Spatial Pyramid

Another parameter to tune is the number of levels to divide the training images into in computing the feature histogram. The hypothesis is that, given more levels in computing the spatial pyramid, more spatial information will be preserved and therefore improving our accuracy. As one can see from Figure 5, this is indeed the case. However, since every subsequent level in the spatial pyramid exponentially increases the number of dimensions in our feature space, we decided to stop before level 4, where number of dimensions would increase from 21 to 75, and the marginal increase in accuracy is projected to be less than one percent.

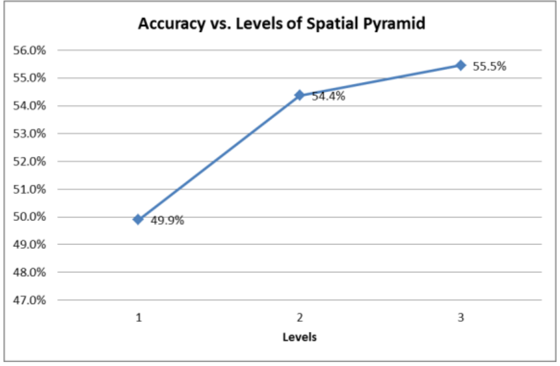

Figure 5 Levels of Spatial Pyramid parameter tuning

3.4. Jitter

We also experimented with different sizes of the jitter grid, and the jitter amount. The result is summarized in Table 1. We concluded that having jitter improves the performance slightly, but the marginal increase is low and the resulting is inconclusive given the amount of data. Further, in depth analysis with more data is needed to conclude the true optimal jitter grid size and jitter amount.

Jitter grid size	Jitter Amount	Accuracy
1	0.0625	54.4234%
3	0.0625	53.2647%
5	0.0625	55.1039%
5	0.05	56.7071%

Table 1 Effect of Jitter Parameters on Accuracy

3.5. Baseline Comparison

Table 2b shows the comparison of classifier accuracy between the methods we have previously described. Baseline 1 is the algorithm with norm threshold of zero, and Baseline 2 is the algorithm trained on cluttered-background images only. The accuracy of our implementation is based on the optimized parameters summarized in Table 2a.

Parameter	**Optimal Value**
K-means clusters	300
SIFT Norm threshold	4.3
Spatial pyramid levels	3
Jitter Grid Size	5
Jitter Amount	0.05

	Approach	Baseline 1	Baseline 2
Classifier accuracy	56.707%	50.120%	29.323%

Table 2: (a) top, tuned parameters for the final comparison (b) bottom, resulting classifier accuracy of the algorithm and the baseline implementations

The poor performance of Baseline 2 is likely due to the high amount of distraction in cluttered-background images. With four classification labels, 29% is not significantly better than random guessing. Since our target object is highly variable (different shapes, design, texture), it is difficult to find dominant feature to distinguish the subject from the background.

The improvement in accuracy from filtering with a norm threshold (Baseline 1) can be explained by spatial pyramid scheme's weak location constraints. Since a teapot can be anywhere within a test image, without segmenting and locating the features corresponding to the teapot, the algorithm will be much more strict on the location of, and amount of white space around, the object. This confirms our original hypothesis.

Figure 6 shows the resulting bounding boxes from the sliding-window detection algorithm. Since we are approximating sliding-window by increasing the size of the spatial pyramid grids such that they overlap, the actual detected bounding boxes are 1/2 longer in each dimension. Also note these bounding boxes are the “most confident” locations of the teapots, and there may be other areas that responded to detection.

As one can see, the algorithm detected fairly accurately in Figure 6(a) and Figure 6(b), while less accurately in the others. This is likely the artifact of two reasons. First, the subject must conform to at least one of the sliding window sizes to be detected. Additionally, if the sliding step, i.e. overlap between the windows, is too wide, then subjects that fall between two windows are unlikely to be detected. This is further amplified by the decision to approximate sliding window with spatial pyramid scheme.

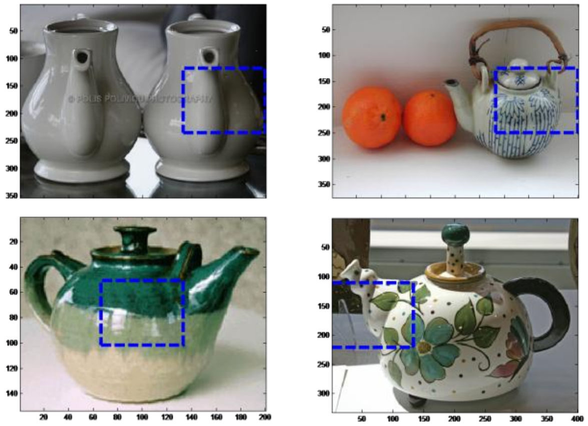

Figure 6 from top left clockwise (a), (b), (c), (d). Detected bounding boxes for teapots

4. Future Work

4.1. Improving the Model

Using grid-based spatial pyramid provides weak spatial information and therefore preserves the relationship between different parts of the subject. This is an inexpensive way to improve performance of object classification utilizing the speed of global descriptors. However, this is at the cost of robustness against geometric changes [1]. We hypothesize that, since we have an isolated object, we can improve robustness against object rotation, by choosing polar coordinate binning as described in Shape Context [5].

4.2. Detection

For simplicity, we used SVM classifier with sliding window detection, and additionally approximated sliding-window detection by overlapping spatial pyramid grids. 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. However, this method requires many levels of window sizes to accommodate different possible sizes of the object inquired, and is not robust to occlusion and objects lying along the borders of these grids.

5. Conclusion

In this paper, we have discussed an algorithm using a discriminatory classifier (multi-class SVM) trained on segmented images to be used as an object detector. Our goal is to compare the performance of such detector trained on cluttered background images versus it trained on segmented (foreground detected) images. The result is that training on segmented images outperforms both training on cluttered background images and on clean-background images without segmentation. The conclusion would be helpful because such detectors, using only global image cues, can be inexpensively implemented compared to more sophisticated part-base detectors.

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] Belongie, S.; Malik, J.; Puzicha, J.; , "Shape matching and object recognition using shape contexts," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.24, no.4, pp.509-522, Apr 2002

6. 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”).