Active Learning for Object Detection on Satellite Images

Alberto Bietti
abietti@caltech.edu

January, 2012

Abstract

Active learning has been successfully used as a method for reducing labeling cost in a classification setting with a large amount of unlabeled data. This is done by starting the learning process with only a handful of labeled examples and then repeatedly find the most informative unlabeled examples, ask for their label and update the classifier with the new labeled examples. Active learning has been applied to image classification tasks, but has shown to be more challenging for object detection problems. We present an active learning system for object detection. Our system is built on top of a linear SVM classifier and relies on crowdsourcing for collecting labels. We used a dataset of satellite images to test our system. Our results show a significant improvement of the learning curve when using active learning to select training examples compared to random selection.

1 Introduction

One of the key problems in computer vision is object detection. Most recent approaches to object detection rely on classification methods from machine learning: you feed a set of training images to a classifier, some of which represent the object, some of which don't, the classifier then learns a way to discriminate between the images describing the object and the others, and then is able to generalize to new images. In the detection problem, the computer should also figure out where the objects are in an image, and generally this is accomplished using a classifier on windows inside the image and searching for objects by sliding a window across the image. This relies significantly on the availability of a good amount of annotated training images, which can sometimes be hard and costly to obtain, especially with the growing number of images available today.

In order to reduce the cost and time of annotation, researchers have started relying on crowdsourcing on platforms like Amazon Mechanical Turk [10, 9], to quickly obtain annotations from many humans at a lower cost. Since these annotation are not always precise and reliable, research has been done to find accurate annotations from the noisy labels by analyzing the behavior of the annotators [10].

The other way to reduce the annotation cost is to use active learning to reduce the number of labeled examples required: start off with few annotated images and then look at a pool of unlabeled examples (which are very cheap to obtain) and find the most informative ones, *i.e.* the ones which would most improve the performance of the classifier once their label has been obtained. This has been shown to significantly reduce the number of required labels [7, 6, 4]. Vijayanarasimhan and Grauman [8] have used active learning in conjunction with crowdsourcing techniques and web-crawled images to provide an end-to-end system for automatically learning object detectors.

We used active learning and crowdsourcing methods to perform object detection on satellite images. Not much work has been done lately on satellite images, yet these can provide insightful information and statistics about geographical regions, such as the number of swimming pools in a given region, which can be used for instance to check for tax evasion. One possible goal of our system would be to automatically learn an object detector that matches certain performance criteria specified by a client, which could be defined by a maximum number of errors on a dataset, or a minimum precision on part of a dataset. Our system uses a linear SVM classifier for detection and a simple scheme for active learning. Annotations are obtained using crowdsourcing on Amazon Mechanical Turk. This paper describes the different parts of our system and shows how it provides an improved learning curve over a standard system and therefore allows us to obtain a high performance classifier at a lower labeling cost.

2 Related Work

Linear Support Vector Machines have been successfully applied to many object detection problems. They have shown great performance on human detection tasks using Histogram of Oriented Gradient (HOG) features [3] and more generally on objects well defined by their shape. Although Linear SVMs might under-perform Kernel SVMs or other classification methods, they show great computational advantages such as linear training time in the number of training examples and constant classification time, which are particularly useful in an active learning context where the classifier needs to be trained many times and frequently predict scores of unlabeled examples.

Active learning has been proven useful in reducing annotation cost and time by selecting only the most important unlabeled examples to be queried for a label in order to improve the performance of the classifier with a minimum number of useful training examples [7, 4, 2]. Applications to image classification tasks have shown that active learning can effectively help reducing the labeling costs (*e.g.* [6, 5]). Similar active learning techniques would not scale well to window-based object detection problems, but simpler heuristics have sufficed to provide lower labeling costs in such problems [1, 8]. Vijayanarasimhan and Grauman [8] successfully applied the “simple margin” heuristic (which queries points closest to the SVM decision hyperplane) to large scale detection problems by using a hyperplane-hashing algorithm in conjunction with a sparse coding and max-pooling technique that makes it easier to find candidate image windows and doesn’t require a sliding-window approach.

In order to easily label and annotate images, researchers have often relied on crowdsourcing techniques, usually based on tools like Amazon Mechanical Turk [10, 11, 8], as well as custom designed online games [9]. Welinder *et al.* [10] used Bayesian inference techniques to find structure in the noisy annotations and discover the different qualities of each Mechanical Turk annotator and consequently use this information to obtain more precise annotations. Crowdsourcing has also been used to train an object detector live in an active learning setting [8].

In our work, we applied similar methods to satellite images, where objects are very small relative to the size of the image, and where many objects of interest can be found in a single image, unlike the images used in most active learning for object detection settings, such as [8].

3 The Detector

3.1 The Classifier

Our detector uses a linear SVM classifier in a sliding window approach. Given an input image window I_i, we compute a vector of features $x_i = \phi(I_i) \in \mathbb{R}^n$. Let $(x_i, y_i) \in \mathbb{R}^n \times \{-1, 1\}$ for $i = 1, \dots, m$ be our set of labeled training examples. Training the SVM classifier solves the following optimization problem:

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2} w^\top w + C \sum_{i=1}^{m} \xi_i \\ s.t. \quad & y_i(w^\top x_i) \geq 1 - \xi_i, i = 1, \dots, m \\ & \xi_i \geq 0, i = 1, \dots, m \end{aligned}$$

The resulting decision function is given by $sign(h(x))$, where $h(x) = w^\top x$ is the score function, corresponding to the distance from x to the decision plane. A bias term can be included by adding a constant feature equal to 1 in vector x.

3.2 The Features

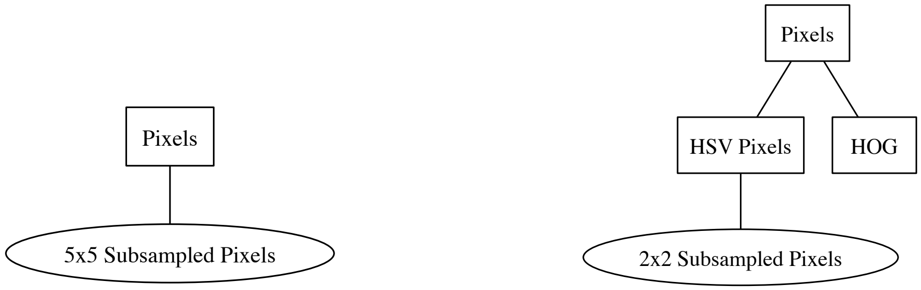
We used three main types of features for our detector: subsampled pixel values, color histograms, and histogram of gradient (HOG) features.

Subsampled pixel values The subsampling was done using bilinear interpolation for simple spatial pooling. This allows the classifier to rely mostly on colors and less on shape, which is well-suited for objects like pools in satellite images. Using a fixed subsampling size can also allow the system to work across different scales and resolutions.

Color histograms These are obtained on RGB or HSV pixels, either independently on each color channel, or as 2D/3D histograms on different channels, or by first finding a few relevant clusters in 3D color space using K-means on a few images from the dataset and then binning each pixel to its closest cluster center to construct the histogram. Using normalized histograms can also work across different scales.

Histogram of gradient (HOG) This method (Dalal and Triggs [3]) works particularly well for objects well determined by their shape. Unfortunately, HOG isn't invariant to rotation and doesn't perform very well when objects appear rotated at different angles, which is often the case in satellite images. HOG features can be computed either globally (so that image windows are matched to the nearest HOG boxes) or locally for each image window.

Combining features To combine multiple kinds of features, we used a simple tree-based structure where each node computes new global or local features from its parent's features. Our algorithm then concatenates the resulting feature arrays at the specified image locations, in a way that only extracts global features once for each image. Figure 1 shows two example of trees: square nodes indicate global features while rounded nodes denote local features. The image gets split into windows at the specified locations once an edge between a square and a rounded node is reached (or after a leaf square node), and descending feature nodes are applied to these local windows.



(a) Subsampled Pixels

(b) HSV Histogram + HOG

Figure 1: Examples of feature trees

4 Active Learning

We would like to minimize the cost of labeling the numerous images in the dataset. A common approach is to use active learning techniques in order to actively select which training examples (in our case, training windows inside an image) will be most informative and will improve the performance of the classifier.

4.1 The Simple Margin Method

A simple heuristic for active learning is to query the points closest to the SVM decision boundary since they are highly uncertain (a small shift of the decision boundary might change their class). This is sometimes referred to as the "simple margin" querying method (see [7]) and has been successfully applied to detection problems (see [1, 8]). We used this heuristic in simulation by selecting a pool of unlabeled training windows x (although we keep track of their actual label) extracted in a sliding-window manner from a set of training images, and iteratively selecting the T windows with smallest margin $|h(x)|$ (we used $T = 50$), after after an initial training of the classifier h with a certain number of labeled training windows $(x, y) \in \mathbb{R}^n \times \{-1, 1\}$.

4.2 Other Methods

Other active learning methods use different techniques to find informative examples to query for their label.

Kapoor *et al.* used Pyramid Match Kernel Gaussian Processes to perform active learning in object categorization tasks [6]. The probabilistic framework makes it possible to explicitly determine a measure of uncertainty in categorizing an unlabeled image, and therefore to easily select uncertain images to label in an active learning setting. Unfortunately, this method quickly becomes computationally intractable when the available number of examples is larger than 100-200, which is certainly too small for object detection tasks where the number of unlabeled training windows (mostly negatives) is larger by at least one order of magnitude.

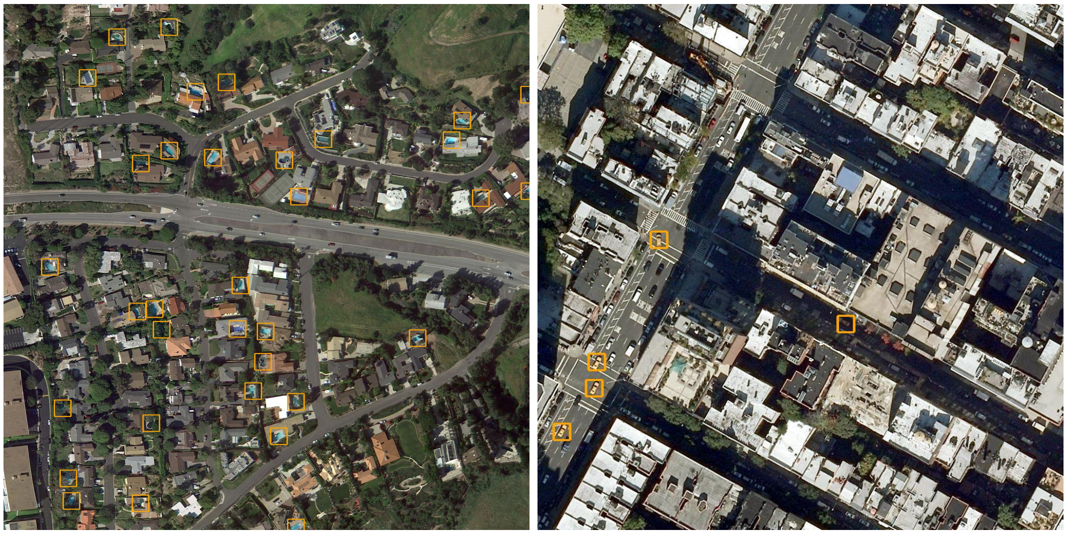


Figure 2: Examples of satellite images from the pools (left) and taxis (right) datasets. Orange boxes indicate ground truth.

Other more theoretical methods rely on maintaining a *version space* (a set of candidate hypotheses) and query labels so as to quickly and efficiently reduce its size [2, 7]. These methods generally involve looking at each unlabeled example and comparing the version spaces V^+ and V^- obtained by classifying the example as either positive or negative in order to decide whether to ask for its true label. This means the classifier needs to be trained twice for each unlabeled example, which is quite intractable unless some type of online learning scheme is used, and applying this to detection problems with many unlabeled examples remains an issue.

5 Experiments and results

5.1 The Datasets

The datasets we used are collections of satellite images of different cities from Google Maps. The objects we looked for are swimming pools in Los Angeles and Athens, as well as yellow taxis in Manhattan. Although the images are of relatively good quality, they do present a few problems: the detail and contrast sometimes varies significantly among the different images in the same dataset, leaving us with pools of many different colors (including some empty pools that appear gray), and some yellow taxis that look almost white in some images. Many of the taxis in the Manhattan images are in the shade and are therefore very dark and hard to spot. Other problems included a certain number of strange artifacts, especially in Athens but also in Manhattan, and occluded objects (taxis or pools hiding below trees). Figure 2 shows examples of images from the datasets.

The ground truth annotations were obtained in two steps. Images were first sent out to Amazon Mechanical Turk, where a few users were asked to click on the pools and taxis on each image. We then used a simple GUI to browse through the noisy annotations obtained with Amazon Mechanical Turk and provide more accurate ground truth. Sometimes accurate annotations turned out to be much harder than expected, therefore we decided to specify a confidence level with each annotation (through the GUI).

5.2 Detector Results

The best detector performance on satellite images was obtained using subsampled pixel regions. Figure 3 shows the performance of the detector at different subsampling sizes on our Los Angeles swimming pools dataset and Figure 4 shows some examples of false detections and misses. Color histograms showed poor performance with a linear SVM: they are more useful when a non-linear kernel is used, especially the histogram intersection kernel or even a histogram pyramid match kernel, which has additional benefits in terms of localizing features in the detection window. HOG performed quite poorly on swimming pools since the shape of pools varies a lot.

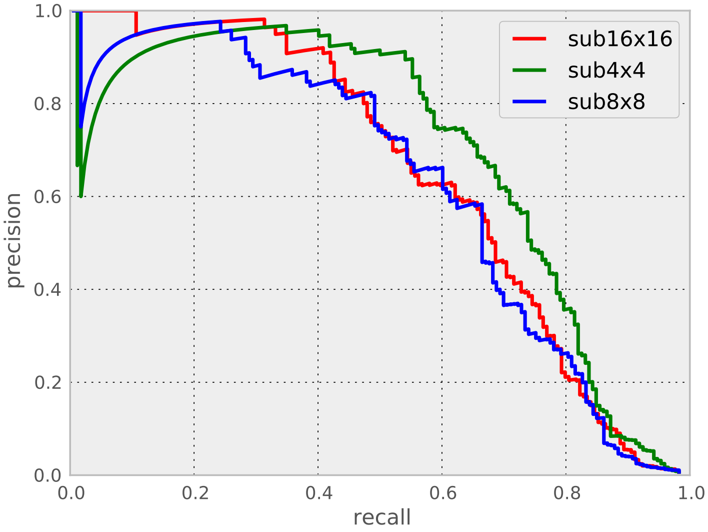
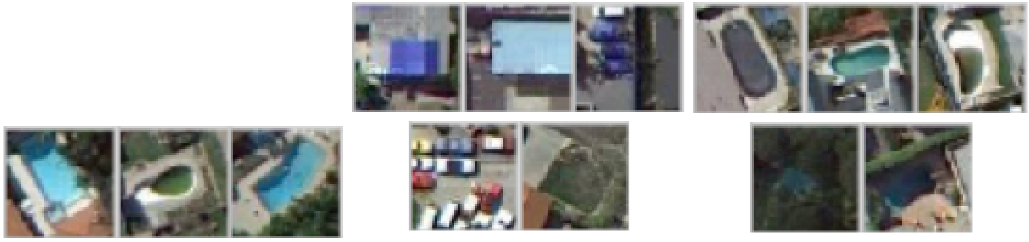


Figure 3: Performance of the detector with different subsampling sizes on LA swimming pools



(a) Examples of pools

(b) False positives

(c) Misses

Figure 4: Results on pools

5.3 Active Learning Results

Our active learning experiment starts by selecting a pool of unlabeled training windows x (although we keep track of their actual label) extracted in a sliding-window manner from a set of training images. The algorithm trains an initial classifier with a certain number of labeled training windows (we used 10 positive and 10 negative examples in this experiment). It then repeats the following steps: select the T windows with smallest margin (we used $T = 50$), find their label (this corresponds to querying for the label of these windows), add these to the training set and re-train the classifier.

Figure 5 shows the evolution of the performance of the classifier during the active learning process on the LA swimming pools dataset, Figure 6 shows the distribution of positive and negative examples selected in each iteration and Figure 7 shows some results of this active learning approach compared to other querying methods. We can see that the active learning strategy greatly improves the performance of the classifier within a few iterations of active selection by choosing the most informative examples. In comparison, passive selection finds very few interesting examples (and very few positives) and stagnates at low performance.

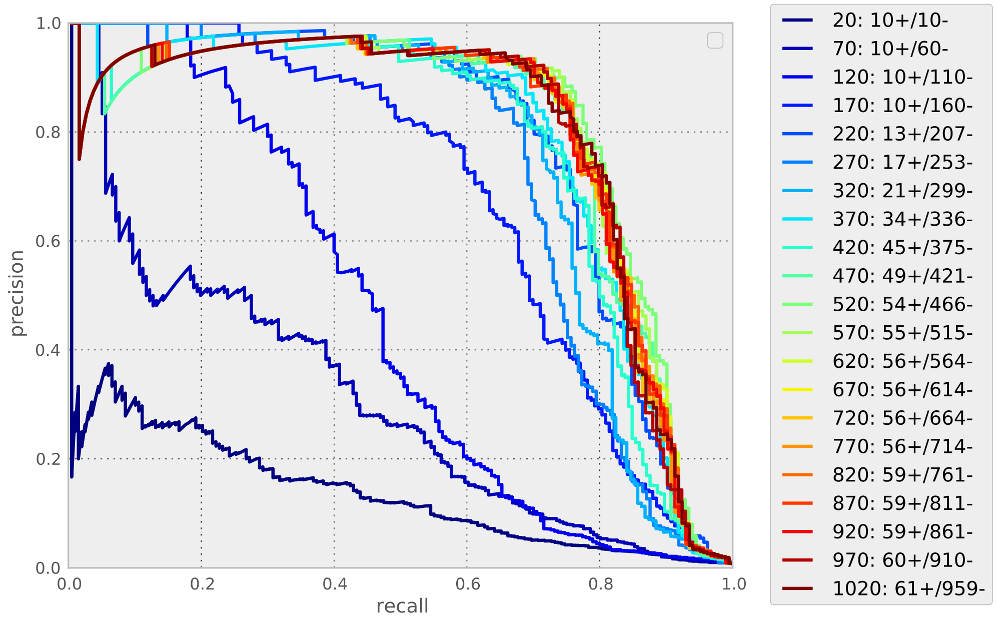


Figure 5: Evolution of the precision-recall curve of the simple margin active learner. The legend shows the total number of training windows followed by the number of positives and negatives.

6 Conclusion and Discussion

Our experiments show that simple active learning techniques can successfully be applied to object detection problems, by using the simple margin method on a linear SVM with a set of simple features. However, this approach has a few limitations. One has to do with computational tractability:

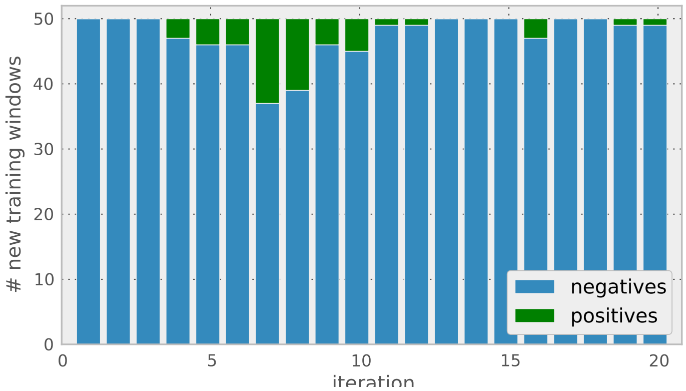


Figure 6: Number of positive and negative training windows added by the simple margin active learner in each iteration.

we currently hold a pool of image windows extracted in a sliding-window/grid-based manner; this means that if the objects in question are small, each image can contain several thousand windows, and if the number of unlabeled images gets large, the pool of features becomes larger and larger. In this case, selecting the T examples closest to the margin can be prohibitively expensive (even though it can be done in $O(n)$). One can avoid this computational cost by only looking at subsets of the pool, but since positive examples are very rare compared to negative ones, they may never be found. Finding positive examples is a major difficulty in this context of object detection, and this leads to one of the main problems of the simple margin method, which is that it may never query some groups of examples which are far from the decision boundary and classified incorrectly.

There are two main points requiring improvements: the first is the active learning sampling scheme, which could be more accurate than the simple margin heuristic but also scale well to detection problems; the second is the search of good candidate windows, which, as in [8], needs to find relevant windows more efficiently and scale better than the sliding-window approach.

References

[1] Y. Abramson and Y. Freund. Active learning for visual object detection. In *CVPR*, 2005.

[2] A. Beygelzimer, D. Hsu, J. Langford, and T Zhang. Agnostic active learning without constraints. In *NIPS*, 2010.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] S. Dasgupta, D.J. Hsu, , and C. Monteleoni. A general agnostic active learning algorithm. In *NIPS*, 2007.

[5] A. Joshi, F. Porikli, , and N. Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, 2009.

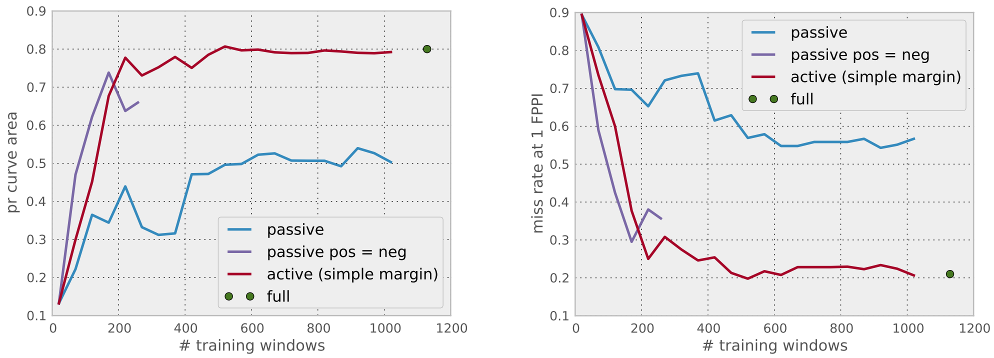


Figure 7: Learning curves for different querying methods on the LA-1 pools dataset. The straight lines represent the performance of fully supervised classifiers. *passive* queries passively at random whereas *passive pos* = *neg* adds an equal number of positives and negatives on each iteration. *full* uses the maximum number of positives (129) and 1000 negatives.

[6] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *ICCV*, 2007.

[7] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.

[8] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. In *CVPR*, 2011.

[9] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *CHI*, 2004.

[10] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *NIPS*, 2010.

[11] P. Welinder and P. Perona. Online crowdsourcing: Rating annotators and obtaining cost-effective labels. In *ACVHL*, 2010.