Unsupervised Detection of Regions of Interest
Using Iterative Link Analysis

Gunhee Kim
School of Computer Science
Carnegie Mellon University
gunhee@cs.cmu.edu

Antonio Torralba
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
torralba@csail.mit.edu

Abstract

This paper proposes a fast and scalable alternating optimization technique to detect regions of interest (ROIs) in cluttered Web images without labels. The proposed approach discovers highly probable regions of object instances by iteratively repeating the following two functions: (1) choose the exemplar set (i.e., a small number of highly ranked reference ROIs) across the dataset and (2) refine the ROIs of each image with respect to the exemplar set. These two subproblems are formulated as ranking in two different similarity networks of ROI hypotheses by link analysis. The experiments with the PASCAL 06 dataset show that our unsupervised localization performance is better than one of state-of-the-art techniques and comparable to supervised methods. Also, we test the scalability of our approach with five objects in Flickr dataset consisting of more than 200K images.

1 Introduction

This paper proposes an unsupervised approach to the detection of regions of interest (ROIs) from a Web-sized dataset (Fig.1). We define the regions of interest as highly probable rectangular regions of object instances in the images. The extraction of ROIs is extremely helpful for recognition and Web user interfaces. For example, comparative studies in [3, 5] show that the ROI detection is useful to learn more accurate object models, which lead to nontrivial improvement of classification and localization performance. In the indoor scene recognition [17], the local regions that contain objects may have special meaning to characterize the scenes. Also, many Web services allow a user to attach notes on user-specified regions in a cluttered image (e.g., Flickr Notes). Our method can ease this cumbersome annotation by automatically suggesting the regions that a user may be interested in.

Our solution to the problem of unsupervised ROI detection is inspired by an alternating optimization. Alternating optimization is one of widely used heuristics where optimization over two sets of variables is not straightforward, but optimization with respect to one while keeping the other fixed is much easier and solvable. This approach has been successful to solve a wide range of problems such as K-means, Expectation-Maximization, and Iterative Closest Point algorithm [2].

Figure 1: Detection of regions of interest (ROIs). Given a Web-sized dataset, our algorithm detects bounding box-shaped ROIs that are statistically significant across the dataset in an unsupervised manner. The yellow boxes are groundtruth labels, and the red and blue ones are ROIs detected by the proposed method.
The unsupervised ROI detection can be thought of as a chicken-and-egg problem between (1) finding exemplars of objects in the dataset and (2) localizing object instances in each image. If class-representative exemplars are given, the detection of objects in images is solvable (i.e., a conventional detection or localization problem). Conversely, if object instances are clearly annotated beforehand, the exemplars can be easily obtained (i.e., a conventional modeling or ranking problem).

Given an image set, we first assume that each image itself is the best ROI (i.e., the most confident object region). Then a small number of highly ranked ones among the selected ROIs are chosen as exemplars (called hub seeking), which serve as references to refine the ROIs of each image (called ROI refinement). We repeat these two updates until convergence. The two steps are formulated as ranking in two different similarity networks of ROI hypotheses by link analysis. The hub seeking corresponds to finding a central and diverse hub set in a network of the selected ROIs (i.e., inter-image level). The ROI refinement is the ranking in a bipartite graph between the hub sets and all possible ROI hypotheses of each image (i.e., intra-image level).

Our work is closely related to topics on ROI detection [3, 5, 17, 14], unsupervised localization [9, 24, 21, 18, 1, 12], and online image collection [13, 19, 6]. The ROI detection and unsupervised localization share a similar goal of detecting the regions of objects in cluttered images. However, most previous work has been successful for standard datasets with thousands of images. On the other hand, our goal is to propose a simple and fast method that can take advantage of enormous amounts of Web data. The main objective of online image collection is to collect relevant images from highly noisy data queried by keywords from the Web. Its main limitation is that much of the previous work requires additional assumptions such as a small number of seed images in the beginning [13], texts and HTML tags associated with images [19], and user-labeled images [6]. On the other hand, no additional meta-data are required in our approach.

Recently, link analysis techniques on visual similarity networks were successfully exploited in computer vision problems [12, 15, 11, 16]. [15] applied the random walk with restart technique to the auto-captioning task. However, their work is a supervised method requiring annotated caption words for the segmented regions in training images. [12] is similar to ours in that the unsupervised classification and localization are the main objectives. However, their method suffers from a scalability issue, and thus their experiments were performed using only 600 images. [11] successfully applied the PageRank technique to a large-scale image search, but unlike ours their approach is evaluated with quite clean images and sub-image level localization is not dealt with. Likewise, [16] also exploited the matching graph of a large-scale image set, but the localization was not discussed.

The main advantages of our approach are summarized as follows. First, the proposed method is extremely simple and fast, with compelling performance. Our approach shows superior results over a state-of-the-art unsupervised localization method [18] for the PASCAL 06 dataset. We proposed a simple heuristic for scalability to make the computation time linear with the data size without severe performance drop. For example, the localization of about 200K images took only 4.5 hours with naive matlab implementation on a single PC equipped with Intel Xeon 2.83 GHz CPU (once image oversegmentation and feature extraction were done). Second, our approach is dynamic thanks to the evolving network representation. At every iteration, new ROI hypotheses are added and trivial ones are removed from the network while reusing a large portion of previously computed information. Third, unlike most previous work, our approach requires neither human annotation, meta-data, nor initial seed images. Finally, we evaluate our approach with a challenging Flickr dataset of up to 200K images. Although some work [22] in image retrieval uses millions of images, this work has a different goal from ours. The objective of image retrieval is to quickly index and search the nearest images to a given query. On the other hand, our goal is to localize objects in every single image of a dataset without supervision.

2 ROI Candidates and Description

The input to our algorithm is a set of images $\mathcal{I} = \{I_1, \ldots, I_N\}$. The first task is to define a set of ROI hypotheses $\mathcal{R} = \{R_1, \ldots, R_N\}$ from the image set $\mathcal{I}$. Ideally, the set of ROI hypotheses $R_m = \{r_{m1}, \ldots, r_{mn}\}$ of an image $I_m$ enumerates all plausible bounding boxes, and at least one of them is supposed to be a good object annotation. Fig.2 shows the procedure of ROI hypotheses generation. Given an image, 15 segments are extracted by Normalized cuts [20]. The minimum rectangle to enclose each segment is defined as initial ROI hypotheses. Since the over-segmentation
is unavoidable in most cases, the combinations of the initial hypotheses are also considered. We first compute pairwise minimum paths between the initial hypotheses using the Dijkstra algorithm. Then the bounding boxes to enclose those minimum paths are added to the set of ROI hypotheses. Finally, a largely overlapped pair of ROIs is merged if \( r_{ai} \cap r_{aj} \neq 0 \). Note that the hypothesis set always includes the image itself as the largest candidate, and the average set size is about 50.

Each ROI hypothesis is represented by two types of descriptors, which are spatial pyramids of visual words [17] and HOG [3]. As usual, the visual words are generated by vector quantization to randomly selected SIFT descriptors. K-means is applied to form a dictionary of 200 visual words. A visual word is assigned to each pixel of an image by finding nearest cluster center in the dictionary, and then binned using a two-level spatial pyramid. The oriented gradients are computed by Canny edge detection and Sobel mask. Then the HOG descriptor is discretized into 20 orientation bins in the range of \([0°,180°]\) by following [3]. The pyramid level is up to three. The similarity measure between a pair of ROIs is cosine similarity, which is simply calculated by dot product of two \( L^2 \) normalized histograms. Here both descriptors are equally weighted.

3 The Algorithm

3.1 Similarity Networks and Link Analysis Techniques

All inferences in our approach are based on the link analysis of \( k \)-nearest neighbor similarity network between ROI hypotheses. The similarity network is a weighted graph \( G = (V, E, W) \), where \( V \) is the set of vertices that are ROI hypotheses. \( E \) and \( W \) are edge and weight sets discovered by the similarity measure in the previous section. Each vertex is only connected to its \( k \)-nearest neighbors with \( k = a \cdot \log |V| \) [23], where \( a \) is a constant set to 10. It results in a sparse network, which is more advantageous in terms of computational speed and accuracy. It guarantees that the complexity of network analysis is \( \mathcal{O}(|V| \log |V|) \) at worst. The network is row normalized so that the edge weight from note \( i \) and \( j \) indicates the probability of a random surfer jumping from \( i \) to \( j \). The link analysis technique we use is the PageRank [4, 10]. Given a similarity matrix \( G \), it computes the same length of PageRank vector \( p \), which assigns a ranked score to each vertex of the network. Intuitively, the PageRank scores of the network of ROI hypotheses are indices of the goodness of hypotheses.

3.2 Overview of the Algorithm

Algorithm 1 summarizes the proposed algorithm. The main input is the set of ROI hypotheses \( R \) generated by the method of section 2. The output is the set of selected ROIs \( S^* \) (\( \subset R \)). In each image, usually one or two, and rarely more than three, of the most promising ROIs are chosen.

The basic idea of our approach is to jointly optimize the ROI selection of each image and the exemplar detection among the selected ROIs. Exemplars correspond to hubs in our network representation. We begin with images themselves as an initial set of ROI selection \( S^{(0)} \) (Step 1). Even though this initialization may be poor for many images, highly ranked hubs among the ROIs are likely to be much more reliable. They are detected by the function Hub seeking (Step 3). Then, the hub sets are exploited to refine the ROIs of each image by the function ROI refinement (Step 4). In turn, those refined ROIs are likely to lead to a better hub set at next iteration. The alternating iterations of those two functions are expected to reach a convergence for not only the best ROI selection of each image but also the most representative ROIs of the data set as the exemplar set. Fig.4.(c) shows an example of refining ROI selections at every iteration. Although our algorithm forces to select at least one ROI for each image, the PageRank vector by ROI refinement can indicate the confidence of each ROI, which can be used to filter out wrongly selected ROIs later. Conceptually, both func-
we design the hub seeking inspired by Mean Shift [7]; given data points, the algorithm creates a
criteria are optimization with image patches is linearity with small incremental displacement (e.g.
mergence, sensitivity to initial guess, and quality of our solution. One widely used assumption in the
Since we deal with discrete patches from unordered natural images on the Web, it is extremely diffi-
objects in the center with a significant size. Obviously, they are excellent initialization candidates.
this argument in our dataset. Although images in our datasets are highly variable, majority of pic-
reveal the dominant statistics of the image set. The more repetitive visual information may get more
work [12]. If each visual entity votes for others that are similar to itself, this democratic voting can
ages, the hub images are likely to be good references. This is based on the finding of our previous
Inherently, a good initialization is essential for alternating optimization. Our key assumption here
is as follows: Provided that the similarity network is built from a sufficiently large number of im-
images from five objects of our Flickr datasetss and all \{train+val\} images from two objects of the PASCAL06.

Algorithm 1 The Algorithm
Input: The set of ROI hypotheses \( \mathcal{R} \) for the input image set \( \mathcal{I} \).
Output: The set of selected ROIs \( S^*(\subset \mathcal{R}) \) and the exemplar set \( \mathcal{H}^*(\subset S^*) \) when converged at \( T \).
1: \( S^{(0)} \leftarrow \text{largest ROI hypothesis in each image.} \)
while \( S^{(t-1)} \neq S^{(t)} \) or maximum iterations are not reached yet do
2: \( \mathcal{H}^{(t)} \leftarrow \text{Hub seeking}(G^{(t)}), \) where the hub set \( \mathcal{H}^{(t)} \subset S^{(t)} \)
3: \( \mathcal{H}^{(t)} \leftarrow \text{Hub seeking}(G^{(t)}), \) where the hub set \( \mathcal{H}^{(t)} \subset S^{(t)} \)
4: \( s^{(t)}_a \leftarrow \text{ROI refinement}(\mathcal{H}^{(t)}, R_a), \) where \( s^{(t)}_a : \) ROI selection of \( I_a, R_a : \) ROI hypotheses of \( I_a \).
5: \( S^{(t)} \leftarrow S^{(t)} \cup s^{(t)}_a \setminus s^{(t-1)}_a . \)
end for
end while

Algorithm 2 Hub seeking function
Input: (1) Network \( G^{(t)} \), (2) Window size: \( d \).
Output: (1) Hub set \( \mathcal{H}^{(t)} \),
1: Compute PageRank vector \( \mathbf{p} \) of \( G^{(t)} \),
for all vertex \( v \in G^{(t)} \) do
2: Find the neighbor set of \( v \): \( \mathcal{N}_v = \{ u \} \) max reachable probability from \( v \) to \( u > d \).
3: Find local maximum node of \( v \): \( m(v) = \arg \max_u p(\mathcal{N}_u) \) where \( u \in \mathcal{N}_v \).
4: \( \mathcal{H}^{(t)} \leftarrow \{ v \} \text{ if } v = m(v) . \)
end for

Algorithm 3 ROI refinement function
Input: (1) Hub set \( \mathcal{H}^{(t)} \), (2) \( R_a \), ROI hypotheses of \( I_a \)
Output: (1) The selected ROIs \( s^{(t)}_a (\subset R_a) \).
1: Generate \( k \)-NN self-similarity matrix \( W_v \) of \( R_a \) and \( k \)-NN similarity matrix \( W_o \) between \( R_a \) and \( \mathcal{H}^{(t)} \). Both of them are row-normalized.
2: Generate augmented bipartite graph \( W = \begin{pmatrix} \alpha W_v & (1 - \alpha)W_o \\ W_o & 0 \end{pmatrix} \).
3: Compute PageRank vector \( \mathbf{p} \) of \( W \).
4: \( s^{(t)}_a = \arg \max r_{a_j} p(r_{a_j}) \) where \( r_{a_j} \in R_a \).

3.3 Hub Seeking with Centrality and Diversity
The goal of this step is to detect a hub set \( \mathcal{H}^{(t)} \) from \( S^{(t)} \) by analyzing the network \( G^{(t)} \). The main criteria are centrality and diversity. In other words, the selected hub set should be not only highly ranked but also diverse enough not to lose various aspects of the dataset. To meet this requirement, we design the hub seeking inspired by Mean Shift [7]; given data points, the algorithm creates a fixed-radius window at each point. Then each window iteratively moves into the direction of the
For better understanding, let us first consider a pure bipartite graph with $\alpha$ (the dark gray car) having significant values. With nonzero $\alpha$ shows the result of PageRank values from left and right. The values of leftmost and rightmost are 0.0081 and 0.0024, respectively. The hub set successfully covers various views of the car class. (b) The effect of the augmented bipartite graph. The left image is with $\alpha = 0$ and the right with $\alpha = 0.1$. The ranking of hypotheses is represented by jet colormap from red (high) to blue (low). In the left, the weights from the red box to the blue one are (0.052, 0.050, 0.049, 0.049, 0.049); in the right, (0.060, 0.060, 0.059, 0.059, 0.057). (c) An example of ROI evolution. At $T = 0$, the selected ROI is an image itself and is converged to the real object after $T = 5$.

maximum increase in the local density function until it reaches a local maximum. Those local maxima become the modes, and the data points that converge to the same maxima are clustered.

The proposed algorithm 2 works in the same manner. For each vertex, we define the search window in the form of maximum reachable probability $d$ (Step 2). The window covers the vertices whose maximum reachable probability is larger than $d$. For example, given $d = 0.1$, $w_{ij} = 0.6$, $w_{jk} = 0.2$, the probability of vertices $i$ to $k$ is $0.6 \times 0.2 = 0.12 > d$. Thus, $k$ is considered inside the search window of $i$. For the density function, we use the PageRank vector, whose values are proportional to the vertex degrees if the graph is symmetric and connected [25]. In Step 3, we compute the vector $m$ that assigns the local maximum vertex within the window of each vertex. If $v = m(v)$, the $v$ is a local maximum, and it is added to $H^{(t)}$. Additionally, we can easily perform the clustering from $m$. For each node, the search window keeps moving the maximum direction indicated by $m$ until it reaches the local maximum. Then the nodes that converge to the same maxima are clustered.

### 3.4 ROI Refinement

Formally, this step is to define a nonparametric function for each image $f_a : R_a \rightarrow \mathbb{R}^+$ (positive real number) with respect to the hub set $H^{(t)}$. Then the hypothesis with the maximum ranking value is chosen as the best ROI. In order to solve this problem, we first construct an augmented bipartite graph $W$ between the hub set $H^{(t)}$ and all ROIs $R_a$ as shown in Step 2 of Algorithm 3 (see Fig 4(a)). For better understanding, let us first consider a pure bipartite graph with $\alpha = 0$. Then the matrix $W$ represents the similarity voting between the ROI candidates and the hub set. If the PageRank vector $p$ of $W$ is computed, then $p(R_a)$ summarizes the relative importance of each ROI hypothesis with respect to the $H^{(t)}$, which is the ranking function $f_a$ we want. Yet, instead of using a pure bipartite graph ($\alpha = 0$), we augment it by nonzero $\alpha$. Fig.4,(b) explains the effects of $\alpha$. The left image shows the result of $\alpha = 0$. Even though the red hypothesis is the maximum, several hypotheses near the dark gray car have also significant values. With nonzero $\alpha = 0.1$ in the right, those hypotheses augment each other, so the maximum ROI is changed to a hypothesis on the car. In terms of link analysis, if a random surfer visits nodes of ROI hypotheses ($R_a$), it jumps to other hypotheses with probability $\alpha$ or to hubs with $1 - \alpha$. Since the nearby hypotheses share large portions of rectangles, they have higher similarity one another, which results in more votes for overlapping hypotheses.

### 3.5 Scalability Setting

The bottleneck of our approach is the Step 3 of Algorithm 1. The network generation requires quadratic computation of cosine similarity of $S^{(t)}$. In order to bound the computational complexity, we limit the maximum number of images to be considered each run of Algorithm 1 by constant number $N$. $N$ should be small enough not to suffer from computational burden. Simultaneously, it should be large enough to successfully detect the meaningful statistics from an extremely variable
dataset. (In experiments, \( N \) is set to 10,000.) If the dataset size \(| \mathcal{I} | > N \), we randomly sample \( N \) images from \( \mathcal{I} \) and construct initial consideration set \( \mathcal{I}_c \subset \mathcal{I} \). Algorithm 1 is applied to the image set \( \mathcal{I}_c \) to obtain \( \mathcal{S}_c^* \). Then we generate new \( \mathcal{I}_c \) by sampling from unvisited images of \( \mathcal{I} \). In order to reuse the result of \( \mathcal{S}_c^* \) for the new \( \mathcal{I}_c \), we sample \( x\% \) of \( N \) from previous \( \mathcal{S}_c^* \) based on the PageRank values of the network \( G^* \) of \( \mathcal{S}_c^* \). In other words, the highly ranked (i.e. highly confident) ROIs in the previous step are reused to expedite the convergence of next iteration. We iterate the above strategy until all images are examined. This simple heuristic allows our technique to analyze an extremely large dataset in a linear time without significant performance drop.

4 Results

We evaluate our approach with two different experiments, (1) performance tests with PASCAL VOC 2006\(^1\) and (2) scalability tests with Flickr images. The PASCAL dataset provides groundtruth labels, so our approach is quantitatively evaluated and compared with other approaches. Using Flickr images, we examine the scalability of our method in a real-world problem. The images are collected by a query that consists of one object word and one context word. We downloaded the Flickr dataset, and thus all of the images labels, so our approach is quantitatively evaluated and compared with other approaches. Usingcomparison, we ran publicly available code of one of the state-of-the-art techniques proposed by challenging so that only very rare previous work has used it for unsupervised localization. For not only localization but also classification according to object types. The PASCAL 06 dataset is weakly supervised.

The input of our algorithm consists of unlabeled images, which may include a single object (called as weakly supervised) or multiple objects (called unsupervised). For unsupervised cases, we perform not only localization but also classification according to object types. The PASCAL 06 dataset is challenging so that only very rare previous work has used it for unsupervised localization. For comparison, we ran publicly available code of one of the state-of-the-art techniques proposed by Russell et al\(^2\) [18] in the identical setting with ours.

The PASCAL dataset consists of \( \{ \text{train+val+test} \} \). However, our approach requires only images as an input, and thus all of the \( \{ \text{train+val+test} \} \) images are used without discrimination between them. Note that our task is an image annotation not a learning problem that requires training and test steps. The performance is evaluated by following the protocol of PASCAL evaluation: (1) The accuracies are measured from only the \( \{ \text{test} \} \) set. In practice, there is very little performance difference between analysis of all \( \{ \text{train+val+test} \} \) and \( \{ \text{test} \} \) only. (2) The detection is considered correct if the overlap between the prediction and ground truth exceeds 50%.

**Weakly supervised localization.** Fig.5 shows the detection performance as Precision-Recall (PR) curves. For [18], we iterate experiments by changing the number of topics from two to six, and report the best results. For fair comparison between our results and [18], we select only the single best bounding box in each image. We also present the best result of each object in VOC06 competition. Strictly speaking, it is not a fair comparison because the experimental setup of VOC06 competition is supervised while ours is unsupervised. However, we include them as references to show how closely our approach can reach the best supervised methods in VOC 06 for the localization. Although the performance varies according to objects, our approach significantly outperformed [18] except in cow. Promisingly, the performances of our approach for bicycle and motorbike are comparable, and those for bus, car, and dog objects are superior to the bests of the supervised methods in VOC06.

**Unsupervised classification and localization.** Here we evaluate how well our approach works for unsupervised classification and localization tasks (i.e. images of multiple objects are given without any annotation). Since both our method and [18] aim at sub-image level classification and detection, we first find out the most confident region of each image, and run the LDA clustering for [18] and spectral clustering [20] for our method. Fig.6 shows ROC curves as the evaluation of classification by following the VOC06 protocol. We also show the best of the VOC06 submissions for supervised classification as reference. As shown in Fig.6.(a)–(c), our method and [18] present similar ROC performance. In other words, both methods are quite good at ranking for classification. However, the classification rates of our method are better by about 10% for both 3-object and 4-object cases. (Ours: 69.08%; [18]: 59.05% for \{bicycle, car, dog\}. Ours: 59.51%; [18]: 50.99% for

\(^1\)The dataset is available at http://www.pascal-network.org/challenges/VOC/.

\(^2\)The code is available at http://www.di.ens.fr/~russell/projects/mult_seg_discovery/index.html
Figure 5: Results of weakly supervised localization. PR curves for the \{test\} sets of all objects in the PASCAL 06 dataset for ours (blue), [18] (red), and the best of VOC06 (green). Note that our localization and that of [18] are unsupervised, whereas the VOC06 localization is supervised. (X-axis: recall; Y-axis: precision).

Figure 6: Results of unsupervised classification and localization. (a)−(c) ROC curves for the \{test\} set of \{bicycle, car, dog\} for ours (blue), [18] (red), and the best of VOC06 (green). The AUCs of ours, [18], and the best of VOC06 are as follows; bicycle: (0.892, 0.869, 0.948), car: , and dog: (0.932, 0.954, 0.876), respectively. (X-axis: false positive rates, Y-axis: true positive rates). (d)−(f) PR curves for unsupervised localization of ours (blue) and [18] (magenta). As references, we also represent the results of our weakly supervised localization (red) and the best of VOC 06 (green). (X-axis: recall, Y-axis: precision).

\{bicycle, car, dog, sheep\}. We also show the unsupervised localization performance as PR-curves in Fig.6.(d)−(f). As references, we also represent the results of our weakly supervised experiments and the bests of VOC 06 for corresponding objects. The nontrivial performance drop is observed because the unsupervised setting is more challenging than the weakly supervised one due to the classification errors and distraction by other objects in the dataset.

4.2 Scalability Tests

It is an open question how to evaluate the results of a large number of Web images that have no ground-truth. For a quantitative evaluation, we manually annotated 0.5\% randomly selected images of datasets, and they are used as limited but approximate indices of performance measures. According to the data sizes used in experiments, we randomly pick \(x\)% from the annotated set and \((100 - x)\)% from the non-annotated set. The \(x\) is \{20, 10, 5, 1, 0.5, 0.5\} for the dataset sizes of \{500, 5K, 10K, 50K, 100K, 200K\}, respectively.

Weakly supervised localization. One interesting question we address here is how performances and computation times vary as a function of data sizes. The experiments are repeated ten times for each dataset size, and the median (i.e. fifth-best) performance scores are reported. Similarly to previous tests, we select only the single best ROI per image. As shown in Fig.7, the performances of 500 images highly fluctuate, but those of the dataset sizes above 5K are stable. As dataset sizes increase, a small performance improvement is observed. Since the maximum number of images at each execution of the algorithm is bounded by \(N = 10,000\), the computation times are linear to the number of images, and the performances of the data sizes above \(N\) are similar one another.

Perturbation tests. Here we test the goodness of selected ROIs from a different point of view: the robustness of ROI detection against random network formation. For example, given an image \(I_\alpha\), we can generate 100 sets of 200 randomly selected images including \(I_\alpha\). If the ROI selection for \(I_\alpha\) is repetitive across 100 different sets, we can say the ROI estimator for \(I_\alpha\) is confident. This procedure is similar to bootstrapping or cross-validation.
Figure 7: Weakly supervised localization. (a) PR curves for five objects of our Flickr dataset by varying
dataset sizes from 500 to 200K. (b) The log-log plot between the number of images and the computation time
for the car object. The slope of each range is \{1.23, 2.05, 0.95, 1.05, 1.28\} from left to right, respectively.

Figure 8: Examples of the perturbation tests. The histograms summarize how many times each ROI is selected
in 100 random sets. The frequencies of ROIs are represented in the images by the thickness of bounding boxes
and the jet colormap from red (high) to blue (low). From left to right, the entropies of the distributions are
\{0.2419, 1.6846, 2.4331\}, respectively. (X-axis: ROI hypotheses; Y-axis: frequencies).

Fig.8 shows some examples of the perturbation tests. The histogram indicates how many times each
ROI hypothesis is chosen among 100 random sets. From the left image to the right, one can see the
increase of the difficulty of ROI detection. A peak is observed for the obvious left image, but the
distribution is wider for the challenging right image. The entropy of the distribution in the caption
of Fig.8 can be an index of the measure of difficulty or the confidence of the estimator for the image.

More localization examples. Fig.9 shows more examples of localization in our approach. The third
row illustrates some typical examples of failure. Frequently co-occurred objects can be detected
instead such as flowers in butterfly images, insects on sunflowers, other animals in the zoo, and
persons everywhere. Another common case of failure is that our approach sometimes detects small
multiple instances or a part of an object as one ROI (e.g. a giraffe face rather than the whole body).

5 Discussion

We proposed an alternating optimization approach for scalable unsupervised ROI detection by an-
alyzing the statistics of similarity links between ROI hypotheses. Both tests with PASCAL 06 and
Flickr datasets showed that our approach is not only comparable to other unsupervised and super-
vised techniques but also applicable to challenging images on the Web.

Acknowledgement. Funding for this research was provided by NSF Career award (IIS 0747120).

Figure 9: More examples of ROI discovery. The first and second rows represent successful detections, and the
third row illustrates some typical failures. The yellow boxes are groundtruth labels, and the red and blue ones
are ROIs detected by the proposed method.
References


