HAL
archives-ouvertes.fr

Unsupervised Joint Object Discovery and Segmentation in Internet Images

Michael Rubinstein, Armand Joulin, Johannes Kopf, Ce Liu

▶ **To cite this version:**

Michael Rubinstein, Armand Joulin, Johannes Kopf, Ce Liu. Unsupervised Joint Object Discovery and Segmentation in Internet Images. CVPR 2013 - IEEE Conference on Computer Vision and Pattern Recognition, Jun 2013, Portland, Oregon, United States. pp.1939-1946, 10.1109/CVPR.2013.253 . hal-01064227

HAL Id: hal-01064227
https://hal.inria.fr/hal-01064227

Submitted on 15 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Unsupervised Joint Object Discovery and Segmentation in Internet Images

Michael Rubinstein[1,3] Armand Joulin[2,3] Johannes Kopf[3] Ce Liu[3]

[1]MIT CSAIL [2]INRIA [3]Microsoft Research

(a) Images downloaded from the Internet (b) Our automatic segmentation results (c) State-of-the-art co-segmentation results [8]

Figure 1. Image datasets collected from Internet search vary considerably in their appearance, and typically include many noise images that do not contain the object of interest (a small subset of the *car* image dataset is shown in (a); the full dataset is available in the accompanying material). Our algorithm automatically discovers and segments out the common object (b). Note how no objects are discovered for noise images in (b). Most previous co-segmentation methods, in contrast, are designed for more homogeneous datasets in which every image contains the object of interest, and, therefore, their performance degrades in the presence of noise (c).

Abstract

We present a new unsupervised algorithm to discover and segment out common objects from large and diverse image collections. In contrast to previous co-segmentation methods, our algorithm performs well even in the presence of significant amounts of noise images (images not containing a common object), as typical for datasets collected from Internet search. The key insight to our algorithm is that common object patterns should be salient within each image, while being sparse with respect to smooth transformations across images. We propose to use dense correspondences between images to capture the sparsity and visual variability of the common object over the entire database, which enables us to ignore noise objects that may be salient within their own images but do not commonly occur in others. We performed extensive numerical evaluation on established co-segmentation datasets, as well as several new datasets generated using Internet search. Our approach is able to effectively segment out the common object for diverse object categories, while naturally identifying images where the common object is not present.

1. Introduction

We consider the task of jointly segmenting multiple images containing a common object. The goal is to label each pixel in a set of images according to whether or not it belongs to the underlying common object, with no additional information on the images or the object class[1]. Such capability can be useful for automatic generation of large-scale training sets for object detectors/classifiers, data-driven image synthesis, as well as for improving image-to-text relevance and image search.

The task of simultaneously segmenting multiple images is known as *co-segmentation*, where joint segmentation essentially serves as a means of compensating for the lack of supervisory data, allowing to infer the visual properties of the foreground object even in the absence of *a priori* information about the object or the images.

While numerous co-segmentation methods have been proposed, they were shown to work well mostly on small datasets, namely MSRC and iCoseg, containing salient and similar objects. In fact, in most of the images in those datasets the foreground can be quite easily separated from the background based on each image alone (*i.e*. without co-segmentation, see Section 4.1).

However, Internet image collections, such as the ones returned by image search engines for a given user query, are significantly larger and more diverse (Figure 1(a)). Not only do the objects in images downloaded from the Internet exhibit drastically different style, color, texture, shape, pose, size, location and view-point; but such image collections also contain many *noise* images—images which do not contain the object of interest at all. These challenges, as we demonstrate, pose great difficulties on existing co-segmentation techniques (Figure 1(c)). In particular, most co-segmentation methods assume every image contains the object of interest, and hence are unable to handle dataset noise.

In this paper, we propose a novel correspondence-based *object discovery* and *co-segmentation* algorithm that performs well even in the presence of many noise images. Our algorithm automatically discovers the common object among the majority of images and computes a binary object/background label mask for each image. Images that do not contain the common object are naturally handled by returning an empty labeling (Figure 1(b), Figure 2).

Our algorithm is designed based on the assumption that pixels (features) belonging to the common object should be: (a) *salient*, *i.e*. dissimilar to other pixels within their image, and (b) *sparse*, *i.e*. similar to pixels (features) in other images with respect to smooth transformations between the images. Given an input image dataset, we build a large-scale graphical model connecting similar images, where dense pixel correspondences are used to capture the object's visual variability. These correspondences between images allow us to separate the common object from the background and visual noise.

We performed extensive evaluation of our proposed approach. Our algorithm produces state-of-the-art results on the established MSRC and iCoseg co-segmentation datasets[2], and provides considerable improvement over previous methods on several new challenging Internet datasets containing rigid and non-rigid object categories. Our Internet datasets, ground truth labels and results are available for the research community for further investigation at http://people.csail.mit.edu/mrub/ObjectDiscovery.

2. Related work

Object Discovery. Object discovery has been intensively studied in computer vision. In a supervised setup, objects were treated as topics and images as documents, and generative models such as Latent Dirichlet Allocation (LDA) and Hierarchical Pitman-Yor (HPY) have been used to learn the distribution and segmentation of multiple classes simultaneously [24, 22]. Winn and Jojic [26] propose a generative model for the distribution of mask, edge and color for visual objects with respect to a smooth deformation field. Although good object recovery results were reported, the model is limited to particular views of an object.

Recently, PageRank [7] was used to discover regions of interest in a bounding box representation [10], and self-similarities were used to discover a common pattern in several images [1]. Although in these works no generative models were used to learn the distribution of visual objects, reliable matching and saliency are found to be helpful for object discovery. The notions of matching and saliency were also successfully applied by Fakor *et al*. [5], a work done in parallel to ours, for unsupervised discovery of image categories.

Co-segmentation. Co-segmentation was first introduced by Rother *et al*. [19], who used histogram matching to simultaneously segment the same object in two different images. Since then, numerous methods were proposed to improve and refine the co-segmentation [16, 6, 2, 8], many of which work in the context of a pair of images with the exact same object [19, 16, 6] or require some form of user interaction [2, 4].

These techniques were later extended in various ways. Joulin *et al*. [8] used a discriminative clustering framework that can handle multiple images, and Kim *et al*. [12] proposed an optimization which scales up to even larger datasets. Vicente *et al*. [25] introduced the notion of "objectness" to the co-segmentation framework, showing that requiring the foreground segment to be an *object* often improves co-segmentation results significantly. All these techniques, however, maintain the strong assumption that the object is present in all of the images, which is not true for Internet image collections.

Other methods were proposed to handle images which might not contain the common object, either implicitly [9] or explicitly [11]. In particular, Kim and Xing [11] show promising results given additional user input, but do not show significant improvement in the unsupervised setting. It is clear that in the context of image search and web browsing, user input cannot be used.

Co-segmentation was also explored in weakly-supervised setups with multiple object categories [20, 13]. While image annotations may facilitate object discovery and segmentation, image tags are often noisy, and bounding boxes or class labels are usually unavailable. In this work we show that it is plausible to automatically discover visual objects from the Internet using image search alone.

3. Object Discovery and Segmentation

Let $\mathbf{I} = \{I_1, \ldots, I_N\}$ be the image dataset consisting of N images. Our goal is to compute the binary masks $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_N\}$, where for each image I_i and pixel $\mathbf{x} = (x, y)$, $\mathbf{b}_i(\mathbf{x}) = 1$ indicates foreground (the common object), and $\mathbf{b}_i(\mathbf{x}) = 0$ indicates background (not the object) at location $\mathbf{x}$.

Recall our assumption that for an object of interest the foreground pixels should be salient, *i.e*. dissimilar to other pixels within their image, and sparse, *i.e*. similar to nearest neighbors (with possible changes in color, size and position). We first define terms which capture these two properties, and then combine them within an optimization framework to solve for the most likely labels for all pixels in the dataset. An overview of the algorithm is shown in Figure 2.

Image saliency. The saliency of a pixel or a region in an image can be defined in numerous ways and exten-

[1] We note that while we call our method "unsupervised", we do assume that the input image dataset contains a common visual category. We use "unsupervised" to emphasize that, other than this assumption, the algorithm makes no further use of a priori information such as the common object's class or image annotations.

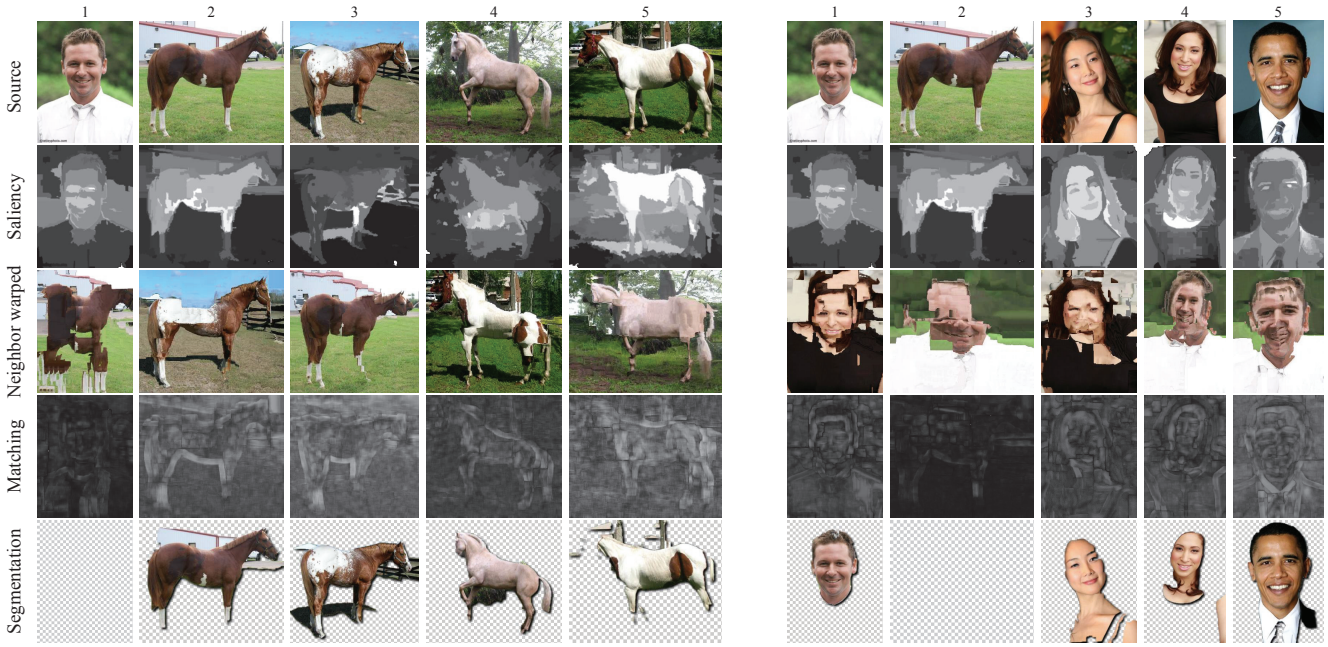


Figure 2. **One of these things is not like the others.** An illustration of joint object discovery and segmentation by our algorithm on two small datasets of five images each. The images are shown at the top row, with two images common to the two datasets – the face and horse images in columns 1 and 2, respectively. Left: when adding to the two common images three images containing horses (columns $3-5$), our algorithm successfully identifies *horse* as the common object and *face* as "noise", resulting in the horses being labeled as foreground and the face being labeled as background (bottom row). Right: when adding to the two common images three images containing faces, *face* is now recognized as common and *horse* as noise, and the algorithm labels the faces as foreground and the horse as background. For each dataset, the second row shows the saliency maps, colored from black (less salient) to white (more salient); the third row shows the correspondences between images, illustrated by warping the nearest neighbor image to the source image; and the fourth row shows the matching scores based on the correspondences, colored from black (worse matching) to white (better matching).

sive research in computer and human vision has been devoted to this topic. In our experiments, we used an off-the-shelf saliency measure—Cheng *et al.*'s Contrast-based Saliency [3]—that produced sufficiently good saliency estimates for our purposes, but our formulation is not limited to a particular saliency measure and others can be used.

Briefly, Cheng *et al.* [3] define the saliency of a pixel based on its color contrast to other pixels in the image (how different it is from the other pixels). Since high contrast to surrounding regions is usually a stronger evidence for saliency of a region than high contrast to far away regions, they weigh the contrast by the spatial distances in the image.

Given a saliency map, $\widehat{M}_i$, for each image I_i, we first compute the dataset-wide normalized saliency, M_i (with values in $[0,1]$), and define the term

$$\Phi^i_{saliency}(\mathbf{x}) = -\log M_i(\mathbf{x}). \tag{1}$$

This term will encourage more (resp. less) salient pixels to be labeled foreground (resp. background) later on.

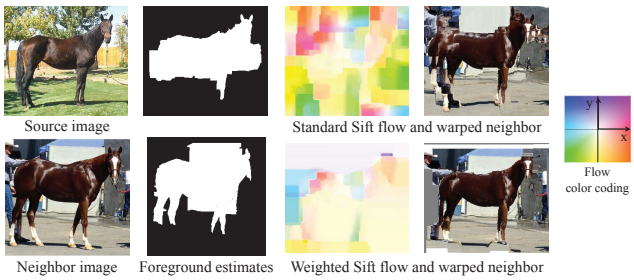
Pixel Correspondence. To exploit the dataset structure and similarity between image regions, we need to establish reliable correspondences between pixels in different images. This enables us to determine a pixel as background even when it may be very salient within its own image. We do this using SIFT flow [15], which has been successfully applied in the past for label propagation [14, 20]. However, instead of establishing the correspondence between all pixels in a pair of images, as done by previous work, we solve and update the correspondences based on our estimation of the foreground regions. This helps in ignoring background clutter and ultimately improves the correspondence between foreground pixels (Figure 3).

Formally, let $\mathbf{w}_{ij}$ denote the flow field from image I_i to image I_j. Given the binary masks $\mathbf{b}_i, \mathbf{b}_j$, the SIFT flow objective function becomes

$$\begin{aligned} &E(\mathbf{w}_{ij}; \mathbf{b}_i, \mathbf{b}_j) = \\ &\sum_{\mathbf{x}\in\Lambda_i} \mathbf{b}_i(\mathbf{x}) \Big(\mathbf{b}_j(\mathbf{x}+\mathbf{w}_{ij}(\mathbf{x})) \left\| S_i(\mathbf{x}) - S_j(\mathbf{x}+\mathbf{w}_{ij}(\mathbf{x})) \right\|_1 \\ &+ (1 - \mathbf{b}_j(\mathbf{x}+\mathbf{w}_{ij}(\mathbf{x}))C_0 + \sum_{\mathbf{y}\in\mathcal{N}^i_{\mathbf{x}}} \alpha \left\| \mathbf{w}(\mathbf{x}) - \mathbf{w}(\mathbf{y}) \right\|_2 \Big), \end{aligned} \tag{2}$$

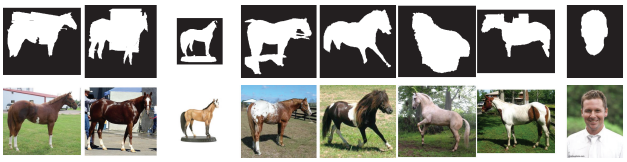
where S_i are the dense SIFT descriptors of image I_i, $x \mapsto \|x\|_p$ is the L_p distance for $p = 1$ and 2, Λ_i is image I_i's lattice, $N^i_{\mathbf{x}}$ is the neighborhood of $\mathbf{x}$, α weighs the smoothness term, and C_0 is a large constant. We then denote by $\mathbf{W}$ the set of all pixel correspondences in the dataset: $\mathbf{W} = \cup_{i=1}^N \cup_{I_j\in\mathcal{N}_i} \mathbf{w}_{ij}$.

The difference between this objective function and the original SIFT flow [15] is that it encourages matching foreground pixels in image I_i with foreground pixels in image I_j. We also use an L_2-norm for the smoothness term instead of the truncated L_1-norm in the original formulation [15] to make the flow more rigid in order to surface mismatches between the images. Figure 3(a) shows



(a) Comparison between standard and weighted Sift flow.

(b) Nearest neighbor ordering (left to right) for the source image in (a), computed with the standard Gist descriptor.



(c) Nearest neighbor ordering (bottom row; left to right) for the source image in (a), computed with a weighted Gist descriptor using the foreground estimates (top row).

Figure 3. **Weighted Gist and Sift flow for improved image correspondence.** We use the foreground mask estimates to remove background clutter when computing correspondences (a), and to improve the retrieval of neighbor images (compared to (b), the ordering in (c) places right-facing horses first, followed by left-facing horses, with the (noise) image of a person last).

the contribution of this modification for establishing reliable correspondences between similar images.

For small datasets, we can estimate the correspondences between any pair of images, however for large datasets such computation is clearly prohibitive. Therefore, we first find for each image I_i a set of similar images, N_i, based on *global* image statistics that are more efficient to compute, and estimate pixel correspondences with those images only. For each image I_i, we fix the size of N_i to the same constant, K. We use the Gist descriptor [17] in our implementation, and similarly modify it to account for the foreground estimates by giving lower weight in the descriptor to pixels labeled as background. Figure 3(b–c) demonstrate that better sorting of the images is achieved when using this *weighted Gist* descriptor, which in turn improves the set of images with which pixel correspondences are computed.

Based on the computed correspondences, we define the matching term

$$\widehat{\Phi^i_{match}}(\mathbf{x}) = \frac{1}{|N_i|} \sum_{j \in N_i} \left\| S_i(\mathbf{x}) - S_i(\mathbf{x} + \mathbf{w}_{ij}(\mathbf{x}) \right\|_1, \quad (3)$$

where smaller values indicate higher similarity to the corresponding pixels. Similarly to the saliency, we compute a dataset-wide normalized term (with values in $[0, 1]$), Φ^i_{match}.

Foreground Likelihood. We use the above saliency and matching terms to define the likelihood of a pixel label:

$$\Phi^i(\mathbf{x}) = \begin{cases} \Phi^i_{saliency}(\mathbf{x}) + \lambda_{match} \Phi^i_{match}(\mathbf{x}), & \mathbf{b}_i(\mathbf{x}) = 1, \\ \beta, & \mathbf{b}_i(\mathbf{x}) = 0, \end{cases} \quad (4)$$

where β is a constant parameter for adjusting the likelihood of background pixels. Decreasing β makes every pixel more likely to belong to the background, thus producing a more conservative estimation of the foreground.

Regularization. We would like the masks $\mathbf{b}_i$ to be spatially consistent within each image, i.e. neighboring pixels are encouraged to have the same label, subject to the image structures. We thus define the *intra-image* compatibility between adjacent pixels $\mathbf{x}, \mathbf{y}$ in image I_i as [14, 19]

$$\Psi^i_{int}(\mathbf{x}, \mathbf{y}) = [\mathbf{b}_i(\mathbf{x}) \neq \mathbf{b}_i(\mathbf{y})] \exp\left(-\|I_i(\mathbf{x}) - I_i(\mathbf{y})\|_2^2\right), \quad (5)$$

where the indicator function $[\cdot]$ is 1 when its argument is true, and 0 otherwise.

We would also like the labeling to be consistent *between* images, and so we add a term accounting for the *inter-image* compatibility between a pixel $\mathbf{x}$ in image I_i and its corresponding pixel $\mathbf{y} = \mathbf{x} + \mathbf{w}_{ij}(\mathbf{x})$ in image I_j:

$$\Psi^{ij}_{ext}(\mathbf{x}, \mathbf{y}) = [\mathbf{b}_i(\mathbf{x}) \neq \mathbf{b}_j(\mathbf{y})] \exp\left(-\|S_i(\mathbf{x}) - S_j(\mathbf{y})\|_1\right). \quad (6)$$

Notice that SIFT features are used for the inter-image similarity metric in Equation 6 whereas RGB intensities are used for the intra-image similarity in Equation 5.

Finally, once we have an estimate of $\mathbf{b}_i$, we can learn the color histograms of the background and foreground regions of image I_i, denoted $\mathbf{h}^0_i$ and $\mathbf{h}^1_i$, respectively. We also denote $\mathbf{h}_i = \left(\mathbf{h}^0_i, \mathbf{h}^1_i\right)$, and $\mathbf{H} = \cup_{i=1}^N \mathbf{h}_i$. We add the term $\Phi^i_{color}(\mathbf{x})$ accounting for the contribution of the pixel $\mathbf{x}$ to the foreground or background color model based on the segmentation estimate $\mathbf{b}_i(\mathbf{x})$:

$$\Phi^i_{color}(\mathbf{x}, \mathbf{h}_i) = -\log \mathbf{h}_i^{\mathbf{b}_i(\mathbf{x})}(\mathbf{x}). \quad (7)$$

We use 3D histograms in color space (with 64 bins in each dimension) to model the color distributions instead of the Gaussian mixture models used in [19].

By combining all the aforementioned terms, we obtain a cost function, $E(\mathbf{B}; \mathbf{W}, \mathbf{H})$, for the segmentations $\mathbf{B}$ given the correspondences $\mathbf{W}$ and the color models $\mathbf{H}$:

$$E(\mathbf{B}; \mathbf{W}, \mathbf{H}) = \sum_{i=1}^{N} \sum_{\mathbf{x} \in \Lambda_i} \Big(\Phi^i(\mathbf{x}) + \lambda_{color} \Phi^i_{color}(\mathbf{x}, \mathbf{h}_i) + \sum_{\mathbf{y} \in \mathcal{N}^i_{\mathbf{x}}} \lambda_{int} \Psi^i_{int}(\mathbf{x}, \mathbf{y}) + \sum_{j \in \mathcal{N}_i} \lambda_{ext} \Psi^{ij}_{ext}(\mathbf{x}, \mathbf{x} + \mathbf{w}_{ij}(\mathbf{x})) \Big). \quad (8)$$

Optimization. Our algorithm alternates between optimizing the correspondences $\mathbf{W}$ (Equation 2), and the binary masks $\mathbf{B}$ (Equation 8). Instead of optimizing Equation 8 jointly over all the dataset images, we use coordinate descent that already produces good results. More specifically, at each step we optimize for a single image by fixing the segmentation masks for the rest of the images. Note that our cost function is non-convex and is not guaranteed to reach the global minimum. After propagating labels from other images, we optimize each image using a Grabcut-like [18] alternation between optimizing Equation 8 and estimating the color models $\mathbf{h}_i$. The algorithm then recomputes neighboring images and pixel correspondences based on the current foreground estimates, and the process is repeated for a few iterations until convergence (we typically used $5 - 10$ iterations).

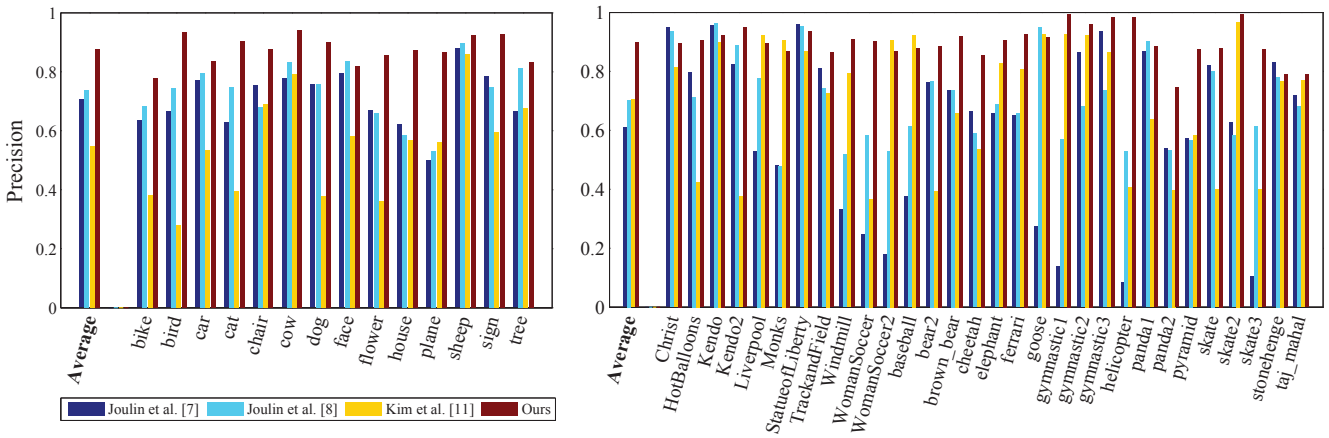


Figure 4. **Segmentation accuracy on MSRC (left) and iCoseg (right)**, measured as the ratio of correctly labeled pixels (both foreground and background), and compared to state-of-the-art co-segmentation methods (we performed a separate comparison with Object Cosegmentation [25]; see the text and Table 1). Each plot shows the average per-class precision on the left, followed by a breakdown of the precision for each class in the dataset.

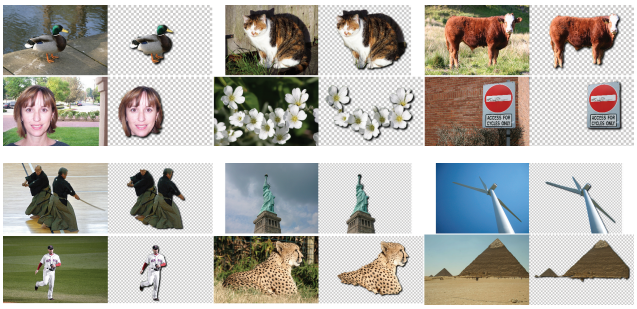


Figure 5. **Sample results on MSRC (top two rows) and iCoseg (bottom two rows)**. For each image we show a pair of the original (left) and our segmentation result (right). More results and qualitative comparisons with state-of-the-art are available in the supplementary material.

4. Results

We conducted extensive experiments to verify our approach, both on standard co-segmentation datasets and image collections downloaded from the Internet. We tuned the algorithm's parameters manually on a small subset of the Internet images, and vary β to control the performance. Unless mentioned otherwise, we used the following parameter settings: $\lambda_{match} = 4, \lambda_{int} = 15, \lambda_{ext} = 1, \lambda_{color} = 2, \alpha = 2, K = 16$. Our implementation of the algorithm is comprised of distributed Matlab and C++ code, which we ran on a small cluster with 36 cores.

We present both qualitative and quantitative results, as well as comparisons with state-of-the-art co-segmentation methods on both types of datasets. Quantitative evaluation is performed against manual foreground-background segmentations that are considered as "ground truth". We use two performance metrics: precision, P (the ratio of correctly labeled pixels, both foreground and background), and Jaccard similarity, J (the intersection over union of the result and ground truth segmentations). Both measures are commonly used for evaluation in image segmentation research. We show a sample of the results and comparisons in the paper, and refer the interested reader to many more results that we provide in the supplementary material.

4.1. Results on Co-segmentation datasets

We report results for the MSRC dataset [23] (14 object classes; about 30 images per class) and iCoseg dataset [2] (30 classes; varying number of images per class), which have been widely used by previous work to evaluate co-segmentation performance. Both datasets include human-given segmentations that are used for the quantitative evaluation.

We ran our method on these datasets both with and without the inter-image components in our objective function (*i.e.* when using the parameters above, and when setting $\lambda_{match} = \lambda_{ext} = 0$, respectively), where the latter effectively reduces the method to segmenting every image independently using its saliency map and spatial regularization (combined in a Grabcut-style iterative optimization). Interestingly, we noticed that using the inter-image terms had negligible effect on the results for these datasets. Moreover, this simple algorithm—an off-the-shelf, low-level saliency measure combined with spatial regularization—which does not use co-segmentation, is sufficient to produce accurate results (and outperforms recent techniques; see below) on the standard co-segmentation datasets!

The reason is twofold: (a) all images in each visual category in those datasets contain the object of interest, and (b) for most of the images the foreground is quite easily separated from the background based on its relative saliency alone. A similar observation was recently made by Vicente *et al.* [25], who noticed that their *single image* classifier outperformed recent co-segmentation methods on these datasets, a finding that is reinforced by our experiments. We thus report the results when the inter-image components are disabled. Representative results from a sample of the classes of each dataset are shown in Figure 5.

Comparison with co-segmentation methods. We compare our results with three previously proposed methods [8, 9, 12]. For all three methods we used the original implementations by the authors that are publicly available, and verified we are able to reproduce the results reported in their papers when running their code. The per-class precision is shown in Figure 4 and the Jaccard similarities are available in the supplemental material. Our overall precision (87.66% MSRC, 89.84% iCoseg) shows significant

Method	MSRC		iCoseg	
	$\bar{P}$	$\bar{J}$	$\bar{P}$	$\bar{J}$
Vicente et al. [25]	90.2	70.6	85.34	62.04
Ours	**92.16**	**74.7**	**89.6**	**67.63**

Table 1. **Comparison with Object Cosegmentation [25] on MSRCV and iCoseg.** $\bar{P}$ and $\bar{J}$ denote the average precision and Jaccard similarity, respectively. The per-class performance and visual results are available in the supplementary material.

improvement over [9] (73.61% MSRC, 70.21% iCoseg) and [12] (54.65% MSRC, 70.41% iCoseg).

Comparison with Object Cosegmentation [25]. Vicente *et al.*'s method [25] is currently considered state-of-the-art on these datasets[2]. Their code is not publicly available, however they provided us with the segmentation masks for the subsets of MSRC and iCoseg they used in their paper. We performed a separate comparison with their method using only the subset of images they used. Our method outperforms theirs on all classes in MSRC and $9/16$ of the classes in iCoseg (see supplementary material), and our average precision and Jaccard similarity are slightly better than theirs (Table 1). We note that despite the incremental improvement over their method on these datasets, our results in this case were produced by segmenting each image separately using generic, low-level image cues, while their method segments the images jointly and requires training.

4.2. Results on Internet Datasets

Using the Bing API, we automatically downloaded images for three queries with query expansion through Wikipedia: *car* (4,347 images), *horse* (6,381 images), and *airplane* (4,542 images). With $K = 16$ nearest neighbors, it took 10 hours on average for the algorithm to process each dataset.

Some discovery results are shown in Figure 6. Overall, our algorithm is able to discover visual objects despite large variation in style, color, texture, pose, scale, position, and viewing-angle. For the objects under a uniform background or with distinct colors, our method is able to output nearly perfect segmentation. Many objects are not very distinctive from the background in terms of color, but they were still successfully discovered due to good correspondences to other images. For *car*, some car parts are occasionally missing as they may be less salient within their image or not well aligned to other images. Similarly, for *horse*, the body of horses gets consistently discovered but sometimes legs are missing. More flexible transforms might be needed for establishing correspondences between horses. For *airplane*, saliency plays a more important role as the uniform skies always match best regardless of the transform. However the algorithm manages to correctly segment out airplanes even when they are less salient, and identifies noise images, such as that of plane cabins and jet engines, as background, since those have an overall worse matching to other images in the dataset.

For qualitative evaluation, we collected partial human labels for each dataset using the LabelMe annotation toolbox [21] and a combination of volunteers and Mechanical Turk workers, resulting in 1,306 car, 879 horse, and 561 airplane images labeled. All labels were manually inspected and refined.

[2]While writing this paper, Kuettel *et al.* [13] managed to improve the state-of-the-art precision on the iCoseg dataset (91.4%).

Method	**Car** (7.5%)		**Horse** (7.8%)		**Airplane** (16%)	
	P	J	P	J	P	J
Without corr.	72.25	46.10	74.88	50.06	80.53	51.18
With corr.	**83.38**	**63.36**	**83.69**	**53.89**	**86.14**	**55.62**

Table 2. **Segmentation accuracy on the Internet datasets**, with and without utilizing image correspondences. Next to the name of each dataset is its percentage of noisy images (images that do not contain the object). P denotes precision and J denotes Jaccard similarity. Qualitative results for these datasets are shown in Figure 6 and the supplementary material.

Method	**Car** (11%)		**Horse** (7%)		**Airplane** (18%)	
	P	J	P	J	P	J
Baseline 1	68.91	0	81.54	0	87.48	0
Baseline 2	31.09	34.93	18.46	19.85	12.52	15.26
Joulin et al. [8]	58.7	37.15	63.84	30.16	49.25	15.36
Joulin et al. [9]	59.2	35.15	64.22	29.53	47.48	11.72
Kim et al. [12]	68.85	0.04	75.12	6.43	80.2	7.9
Ours	**85.38**	**64.42**	**82.81**	**51.65**	**88.04**	**55.81**

Table 3. **Comparison with previous co-segmentation methods on the Internet datasets.**

In Table 2 we show the precision and Jaccard similarity of our method on each dataset, with and without using image correspondences. The performance on airplane is slightly better than horse and car as in many of the images the airplane can be easily segmented out from the uniform sky background. Image correspondences helped the most on the *car* dataset (+11% precision, +17% Jaccard similarity), probably because in many of the images the cars are not that salient, while they can be matched reliably to similar car images to be segmented correctly.

Comparison with co-segmentation methods. We also compared our results with the same three state-of-the-art co-segmentation methods as in Section 4.1 by running them on our datasets. Since the competing methods do not scale to large datasets, we randomly selected 100 of the images with available ground truth labels from each dataset. We re-ran our method on these smaller datasets for a fair comparison. We also compared to two baselines, one where all the pixels are classified as background ("Baseline 1"), and one where all pixels are classified as foreground ("Baseline 2"). Table 3 summarizes this comparison, showing again that our method produces much better results according to both performance metrics (ours results are not exactly the same as in Table 2 bottom row, since only subsets of the full datasets are used here). The largest gain in precision by our method is on the airplane dataset, which has the highest noise level of these three datasets. Some visual comparisons are shown in Figure 7 and more are available in the supplementary material.

4.3. Discussion and Limitations

Some failures of the algorithm are shown in Figure 6 (last row of each dataset) and the supplemental material. False positives include a motorcycle and a headlight in the *car* dataset, and a tree in the *horse* dataset. This indicates that although matching image structures often leads to object-level correspondence, exceptions occur especially when context is not taken into account.

The algorithm also fails occasionally to discover objects with unique views or background. This is because Gist is a global image descriptor, and unique view and background make it difficult to retrieve similar objects in the dataset.

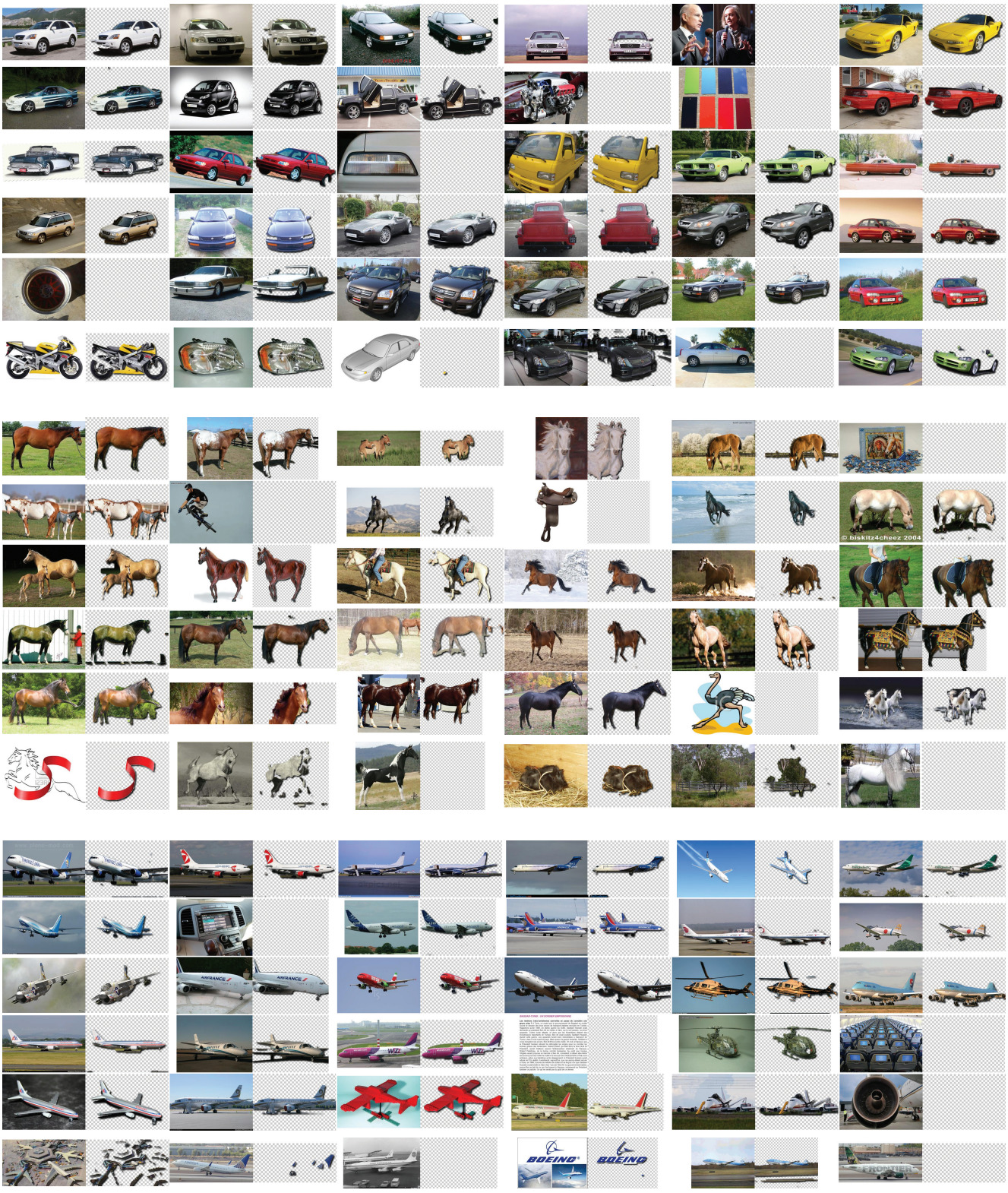
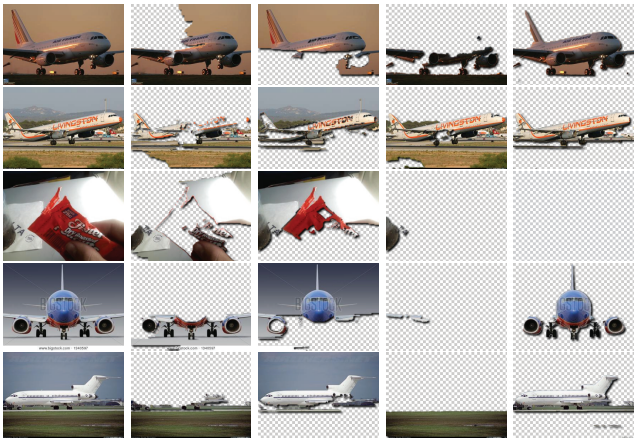


Figure 6. **Automatic discovery of cars, horses and airplanes downloaded from the Internet**, containing $4,347$, $6,381$ and $4,542$ images, respectively. For each image, we show a pair of the original (left) and the segmentation result (right). Notice how images that do not contain the object are labeled as background. The last row of each dataset shows some failure cases where no object was discovered or where the discovery is wrong or incomplete. Quantitative results are available in Table 2, and more visual results can be found in the supplementary material.



Source [8] [9] [12] Ours

Figure 7. **Comparison with state-of-the-art co-segmentation methods on the *airplane* Internet dataset.** More comparisons can be found in the accompanying material.

Finally, our algorithm makes the implicit assumption of non-structured dataset noise. That is, repeating visual patterns are assumed to be part of some "common" object. For example, had a dataset of 100 *car* images contained 80 images of cars and 20 images of car wheels, then using $K = 16$ neighbor images by our algorithm may result in intra-group connections, relating images of cars to other images of cars and images of wheels with others alike. In such case the algorithm may not be able to infer that one category is more common than the other, and both cars and wheels would be segmented as foreground. Fortunately, the fixed setting of K we used seems to perform well in practice, however in the general case K needs to be set according to what the user considers as "common".

5. Conclusion

We explored automatic visual object discovery and segmentation from the Internet using one query of an object category. Image datasets resulting from such queries are significantly more diverse and noisy than the ones used to develop and evaluate previous co-segmentation work. The common object often differs drastically in appearance, and a significant portion of the images may not contain the object at all. We demonstrated that existing co-segmentation algorithms do not perform well in such cases, and presented a new algorithm that is able to naturally handle the visual variation and noise in Internet images. We model the sparsity and saliency properties of the common object, and construct a large-scale graphical model to jointly infer a binary mask for each image. We demonstrated improvement over existing co-segmentation techniques on standard co-segmentation datasets and several challenging Internet datasets.

Acknowledgments. We thank Antonio Torralba for his help in collecting human foreground-background segmentations for our Internet datasets. This work was done while Michael Rubinstein and Armand Joulin were interns at Microsoft Research Redmond. Michael Rubinstein is supported by the Microsoft Research PhD Fellowship. Armand Joulin is supported by the European Research Council (SIERRA and VIDEOWORLD projects).

References

[1] S. Bagon, O. Brostovski, M. Galun, and M. Irani. Detecting and sketching the common. In *CVPR*, pages 33–40, 2010. 2

[2] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010. 2, 5

[3] M. Cheng, G. Zhang, N. Mitra, X. Huang, and S. Hu. Global contrast based salient region detection. In *CVPR*, pages 409–416, 2011. 3

[4] M. Collins, J. Xu, L. Grady, and V. Singh. Random walks based multi-image segmentation: Quasiconvexity results and gpu-based solutions. In *CVPR*, 2012. 2

[5] A. Faktor and M. Irani. clustering by composition–unsupervised discovery of image categories. In *ECCV*, pages 474–487. 2012. 2

[6] D. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009. 2

[7] Y. Jing and S. Baluja. VisualRank: Applying pagerank to large-scale image search. *TPAMI*, 30(11):1877 –1890, 2008. 2

[8] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 2, 5, 6, 8

[9] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012. 2, 5, 6, 8

[10] G. Kim and A. Torralba. Unsupervised detection of regions of interest using iterative link analysis. In *NIPS*, 2009. 2

[11] G. Kim and E. Xing. On multiple foreground cosegmentation. In *CVPR*, 2012. 2

[12] G. Kim, E. Xing, L. Fei-Fei, and T. Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, 2011. 2, 5, 6, 8

[13] D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation propagation in imagenet. In *ECCV*, pages 459–473. Springer, 2012. 2, 6

[14] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *TPAMI*, 33(12):2368–2382, 2011. 3, 4

[15] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *TPAMI*, 33(5):978–994, 2011. 3

[16] L. Mukherjee, V. Singh, and C. Dyer. Half-integrality based algorithms for cosegmentation of images. In *CVPR*, 2009. 2

[17] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42:145–175, 2001. 4

[18] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH*, volume 23, pages 309–314, 2004. 4

[19] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR*, 2006. 2, 4

[20] M. Rubinstein, C. Liu, and W. T. Freeman. Annotation propagation in large image databases via dense image correspondence. In *ECCV*, pages 85–99, 2012. 2, 3

[21] B. Russell, A. Torralba, K. Murphy, and W. Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 77(1):157–173, 2008. 6

[22] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 2

[23] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *ECCV*, pages 1–15, 2006. 5

[24] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *ICCV*, 2005. 2

[25] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, pages 2217–2224, 2011. 2, 5, 6

[26] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *ICCV*, volume 1, pages 756–763, 2005. 2