In Proceedings of the Asian Conference on Computer Vision (ACCV), 2014.

Which Image Pairs Will Cosegment Well? Predicting Partners for Cosegmentation

Suyog Dutt Jain and Kristen Grauman

University of Texas at Austin

Abstract. Cosegmentation methods segment multiple related images *jointly*, exploiting their shared appearance to generate more robust foreground models. While existing approaches assume that an oracle will specify which pairs of images are amenable to cosegmentation, in many scenarios such external information may be difficult to obtain. This is problematic, since coupling the "wrong" images for segmentation—even images of the same object class—can actually deteriorate performance relative to single-image segmentation. Rather than manually specify partner images for cosegmentation, we propose to automatically *predict* which images will cosegment well together. We develop a learning-to-rank approach that identifies good partners, based on paired descriptors capturing the images' amenability to joint segmentation. We compare our approach to alternative methods for partnering images, including basic image similarity, and show the advantages on two challenging datasets.

1 Introduction

In the cosegmentation problem, we are given two or more images containing related content, and must segment them each into regions. Because the inputs are known to share some visual relationship—for example, they contain the same foreground object, or instances of the same object class—the algorithm has valuable cues about which pixels might go together. At a high level, the idea is to detect any common appearance/shapes, exploit that association to determine likely foreground regions, then use a "shared" foreground model to jointly guide the region estimates in all input images [1–8]. In contrast, such cues are not available in the traditional single-image segmentation task, where the system must rely solely on bottom-up features to perform the grouping.

Methods for cosegmentation have a variety of potential applications. They are valuable when working with "weakly supervised" data for object recognition, since they make it possible to automatically isolate the foreground object in training images in spite of cluttered backgrounds. This is quite practical given the availability of tagged Web photos, which are often curated to form recognition datasets but lack foreground annotations. Furthermore, cosegmentation methods can be applied to discover the re-occurring patterns in an image database and summarize its key visual themes, or focus on the foreground for content-based image retrieval tasks. Cosegmentation of batches of related photos (or video frames) can help initialize an interactive method for rotoscoping, allowing designers to composite the foreground object onto novel backgrounds.

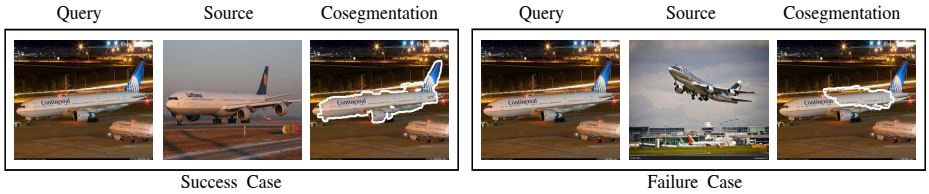

Fig. 1: Motivation for our approach. When an image pair share strong foreground similarity, their cosegmentation is successful (left). However, when incompatible images are used—even from the same object category—cosegmentation fails and can even deterioriate the single-image results (right).

Researchers have made substantial progress on the cosegmentation problem in recent years. While initially the problem was defined to entail two input images showing very same object against distinct backgrounds [1], recent work broadens the problem definition to include batches of input images known only to contain instances of the same object class [2–10]. This is also referred to as *weakly supervised* or *joint foreground segmentation*: each input image is known to contain an instance from the same object category, but its localization within the background is unknown.[1] Some work further relaxes the two-region (foreground/background) assumption to tackle k-region segmentation [11, 7, 6]. Furthermore, eager to capitalize on large collections of weakly labeled images, methods are being developed to account for both noisily labeled instances [11, 8] and scalable optimization [12, 7, 6].

Nonetheless, intra-class appearance variation remains a major obstacle to accurate cosegmentation. In the ideal "clean" scenario, the input batch of images would contain very similar-looking objects, making each image mutually valuable to the rest for building a shared foreground model. However, in many realistic scenarios, the input batch is not so clean. The foreground object may actually look quite different in some images, whether due to image tagging errors, viewpoint variations, or simply diversity in that category's visual appearance. As a result, *not all images are mutually valuable for cosegmentation*. In fact, for this very reason, recent studies report the discouraging outcome that, on some datasets, standard single-image segmentation actually exceeds its cosegmentation counterpart—despite the latter's presumed advantage of having access to a batch of weakly labeled data [4, 8]. See Figure 1.

This motivates us to reconsider the standard assumption that all images are created equal for cosegmentation. Instead, we propose to *predict* which pairs of images are likely to successfully cosegment together. Given an input image and a pool of candidate images sharing the same weak label (e.g., a batch of "car" images), the goal is to find the candidate that, when coupled with the input image, will most boost its foreground accuracy if they are jointly segmented. To this end, we introduce a learning approach that uses a paired description of two

[1] We use the terms *cosegmentation*, *joint segmentation*, and *weakly supervised segmentation* interchangeably.

images to predict their degree of cosegmentation success. The paired description captures not only to what extent the images seem to agree in appearance, but also the uncertainty resulting from their shared foreground model. We formulate the task in a learning-to-rank objective, where successful pairs are constrained to rank higher than those that cosegment poorly together.

Our approach offers a novel way to automatically "partner" images for cosegmentation. Existing methods assume that the "what to cosegment?" question is already answered by some external oracle [1, 2, 9, 10, 3, 5, 4, 7, 6, 8], or else use image similarity alone to gauge compatibility [8, 11]. In contrast, we show how to explicitly *learn* how well image pairs are likely to cosegment together. We demonstrate our approach on two challenging datasets, and show there is great potential to focus joint segmentation only on images where it is most valuable.

2 Related Work

Methods to jointly segment images vary foremost in what they assume about the input images. At one end of the spectrum are methods that assume strong agreement in the inputs' foregrounds, i.e., that the two images contain the same exact object against differing backgrounds [1]. This setting continues to be developed, e.g., for greater efficiency [12] and multi-image collections with interactive user input [13]. In the middle of the spectrum is the *weakly supervised* scenario, where the input images are assumed to contain instances of the same object category [2, 9, 10, 3–8], and the goal is to extract the foreground per image (or possibly multiple foreground objects [6, 7]). At the other end of the spectrum are unsupervised methods, which permit the input images to come from multiple categories. These methods attempt to simultaneously discover the object region boundaries and the category groupings [14, 11, 15]. We apply our method in the middle scenario, where we have a pool of candidate partners that are likely to contain the same object, but they may vary significantly in appearance.

Prior methods assume that all the input images are amenable to cosegment together. In the strict same-object cosegmentation setting, this is assured by manually selecting the input pair (or set). For example, a designer may supply a set of images to be rotoscoped [1], or an analyst may gather aligned brain images from which to segment pathologies [12], or a consumer may group a burst of photos at an event (e.g., a soccer game) into a mini-album [13]. In the weakly supervised setting, the related images often originate from Internet search for an object's name. In this case, the majority of methods assume that all images are mutually amenable to a joint segmentation [2, 9, 10, 3, 5, 4, 7, 6]. In contrast, we propose to automatically determine which among the plausible candidates would serve as the most effective partners for cosegmentation.

To our knowledge, the only prior work that specifically avoids jointly segmenting all input images does so on the basis of a manually defined (i.e., non-learned) image similarity metric [11, 8]. In [11], regions are clustered using a context-based descriptor, and a fixed number of the top clusters are used for joint graph-cut segment refinement. In [8], the joint segmentation is restricted

to an image and a fixed number of its K nearest neighbors using global descriptor (GIST) similarity. In that work, the motivation for paring down the neighbors happens to be computational cost—not accuracy—since it uses inter-image dense correspondences, which are prohibitive to perform on all pairs of examples for large datasets. In both existing methods, the assumption is that image similarity alone is sufficient to predict cosegmentation success. In contrast, our approach learns the behavior of the cosegmentation algorithm, and thus can predict its success for a novel input pair.

There is limited prior work on predicting the quality of a segmentation, and all of it targets the single-image segmentation problem [16–20]. Given the output of a bottom-up segmentation, various methods attempt to classify or rank the regions by their "object-like" quality, having learned the properties of true object segmentations [19, 17, 18]. The method of [16] aims to predict the segmentation accuracy of an algorithm on a novel image based on its global descriptor, while the interactive approach of [20] estimates how much user input is required to sufficiently segment a novel input. Unlike any of the above, our method predicts the extent to which a *joint* segmentation will succeed based on the paired relationship of two candidate images.

3 Approach

As input, our approach takes a **"query"** image I^q and a pool of candidate partner images $\mathcal{P} = \{I^1, \ldots, I^N\}$. Among those N candidates, our method selects the best partner image for I^q, that is, the image that when paired with I^q for cosegmentation is expected to produce the most accurate result. Then, as output, our method returns the result of cosegmenting I^q with its selected partner, namely, a foreground mask for I^q. In the following, we refer to a candidate partner image as a **"source"** image, denoted $I^s \in \mathcal{P}$.

In our implementation, we study the weakly supervised setting, where images in $\mathcal{P}$ contain the same object category as I^q. This forces our method to perform fine-grained analysis to select among all the possibly relevant partners. Even with weak supervision, not all images are satisfactory cosegmentation partners, since they contain objects exhibiting complex appearance and viewpoint variations, as discussed above.

In the following, we first define a basic single-image segmentation algorithm (Sec. 3.1). We then expand that basic engine to handle cosegmentation of a pair of images (Sec. 3.2). Finally, we introduce our ranking approach to predict the compatibility of two images for cosegmentation (Sec. 3.3).

3.1 Single-image segmentation engine

We first describe an approach to perform *single-image* segmentation. In addition to serving as a baseline for the cosegmentation methods, we will also use the output of the single-image segmentation when we predict cosegmentation

compatibility (cf. Sec. 3.3). The method below produces good foreground initializations, though alternative single-image methods could also be plugged into our framework.

Given an image I^i, the goal is to estimate a label matrix L^i of the same dimensions, where $L^i(p) = y_p^i$ denotes the binary label for the pixel p, and $y_p^i \in \{0, 1\}$. The label 0 denotes background (bg) and 1 denotes foreground (fg). We use a standard Markov Random Field (MRF) approach, where each pixel p is a node connected to its spatial neighbors.

We define the MRF's unary potentials using saliency and a foreground color model, as follows. Since this is a single-image segmentation, there is no external knowledge about where the foreground is. Thus, we rely on a generic saliency metric to estimate the plausible foreground region, then boostrap an approximate foreground color model from those pixels. Specifically, for image I^i we first compute its pixel-wise saliency map S^i using a state-of-the-art algorithm [21]. We threshold that real-valued map by its average, yielding an initial estimate for the foreground mask. Then, we use the pixels inside (outside) that mask to learn a Gaussian mixture model (GMM) for the foreground (background) in RGB space. Let G^i_{fg} and G^i_{bg} denote those two mixture models.

The single-image MRF energy function uses these color models and the saliency map:

$$E_{sing}(L^i) = \sum_p A^i_p(y^i_p) \; + \sum_p X^i_p(y^i_p) \; + \sum_{p,p' \in \mathcal{N}} T^i_{p,p'}\big(y^i_p, y^i_{p'}\big), \tag{1}$$

where A^i_p and X^i_p are unary terms, $T^i_{p,p'}$ is a pairwise term, and $\mathcal{N}$ consists of all 4-connected neighborhoods. We define the *appearance likelihood* term as:

$$A^i_p(y^i_p) = -\log P(F^i(p)|G^i_{y^i_p}), \tag{2}$$

where $F^i(p)$ denotes the RGB color for pixel p in image I^i. This term reflects the cost of assigning a pixel as fg (bg) according to the GMM models. We define the *saliency prior* unary term as:

$$X^i_p(y^i_p = 1) = -\log P(S^i(p)), \tag{3}$$

where $S^i(p)$ denotes the saliency value for pixel p. This term reflects the cost of assigning a pixel as fg, where more salient pixels are assumed more likely to be foreground. For the background label, we have the corresponding term, $X^i_p(y^i_p = 0) = -\log(1 - P(S^i(p)))$. Finally, the pairwise term,

$$T^i_{p,p'}(y^i_p, y^i_{p'}) = \delta(y^i_p \neq y^i_{p'}) \exp(-\beta \|F^i(p) - F^i(p')\|), \tag{4}$$

is a standard smoothness prior that penalizes assigning different labels to neighboring pixels that are similar in color, where β is a scaling parameter.

We employ graph cuts to efficiently minimize Eqn. 1 and apply five rounds of iterative refinement (as in GrabCut [22]), alternating between learning the likelihood functions and obtaining the label estimates. The result is a label matrix $L^{i^*}_{sing} = \arg\min_{L^i} E_{sing}(L^i)$.

3.2 Paired-image cosegmentation engine

Next we define the cosegmentation engine we use in our implementation, which expands on the single-image approach above. During training, our method targets a given cosegmentation algorithm, as we will see in the next section. Any existing cosegmentation algorithm could be plugged in; the role of our method is to improve its results by focusing on the most compatible image partners.

Given a query and source image pair, I^q and $I^s \in \mathcal{P}$, we define an energy function over their joint labeling. This model is initialized using GMM appearance models learned from $L^{q^*}_{sing}$ and $L^{s^*}_{sing}$, the single-image results for the two inputs obtained by optimizing Eqn. (1). Specifically, we pool the foreground (background) pixels from both label masks to learn the joint GMM G^{qs}_{fg} (G^{qs}_{bg}) in RGB space. Here and below, the superscript qs denotes a joint term that is a function of both the query and source images.

Let L^{qs} be shorthand for the two label matrices output by the cosegmentation, $L^{qs} = (L^q, L^s)$. Our joint energy function takes the following form:

$$E_{coseg}(L^{qs}) = E_{sing}(L^q) \; + E_{sing}(L^s) \; + \Theta^{qs}_{app}(L^{qs}) \; + \Theta^{qs}_{match}(L^{qs}), \quad (5)$$

where the first two terms refer to the single-image energy for either output, as defined in Eqn. (1), and Θ^{qs}_{app} and Θ^{qs}_{match} capture the energy of a joint label assignment based on appearance and matching terms, respectively (and will be defined next). Note that even though the energy function contains terms for individual label matrices, they are optimized *jointly* to minimize Eqn. (5).

The *joint appearance likelihood* term is defined as

$$\Theta^{qs}_{app}(L^{qs}) = \sum_{p \in I^q} A^{qs}_p(y^q_p) \; + \sum_{r \in I^s} A^{qs}_r(y^s_r), \quad (6)$$

and it captures the extent to which the two output masks deviate from the expected foreground/background appearance discovered with saliency. As before, each A^{qs}_p term is defined as the negative log likelihood over the GMM probabilities; however, here it uses the joint GMM appearance models G^{qs}_{fg} and G^{qs}_{bg} obtained by pooling pixels from the two images' initial foreground estimates.

The *matching likelihood* term $\Theta^{qs}_{match}(L^{qs})$ leverages a dense pixel-level correspondence to establish pairwise links between the two input images. Let $\mathcal{F}_{qs}(p)$ denote the 2D flow vector from pixel p in image I^q to its match in image I^s. We introduce an edge in the cosegmentation MRF connecting each pixel $p \in I^q$ to its matching pixel $r \in I^s$, where $r = p + \mathcal{F}_{qs}(p)$. Using these correspondences, the matching likelihood is a contrast-sensitive smoothness potential over linked (matched) pixels in the two images:

$$\Theta^{qs}_{match}(L^{qs}) = \sum_{p \in I^q, r \in I^s} \delta(y^q_p \neq y^s_r) \exp(-\beta \| D^q(p) - D^s(r) \|), \quad (7)$$

where $D^i(p)$ is a local image descriptor computed at pixel p (we use dense SIFT), and β is a scaling constant. This energy term encourages similar-looking *matched* pixels between the query and source to take the same fg/bg label.

The matching in Eqn. (7) helps cosegmentation robustness. We compute $\mathcal{F}_{qs}$ using the Deformable Spatial Pyramid (DSP) matching algorithm [23], an efficient method that regularizes match consistency across a pyramid of spatial regions and permits cross-scale matches. By linking $p \in I^q$ to $r \in I^s$—rather than naively linking $p \in I^q$ to $p \in I^s$—we gain robustness to the translation and scale of the foreground object in the two input images. This is valuable when the inputs do share a similar-looking object, but its global placement or size varies. Notably, this flexibility is lacking in a strictly image-based global comparison approach (like GIST and the scale-sensitive SIFT Flow as used in [8]). It thus enables mutual discovery of the object between the two images.

To optimize Eqn. (5), we again employ graph cuts with iterative updates. This yields the cosegmented output image pair, $(L^{q^*}_{coseg}, L^{s^*}_{coseg}) = \arg\min_{L^{qs}} E_{coseg}(L^{qs})$.

3.3 Learning cosegmentation compatibility to predict partners

Having defined the underlying single-image and paired-image segmentation algorithms, we can now present our approach to predict which partner image is best suited for cosegmentation with a novel query image. There are two main components: 1) extracting features that are suggestive of cosegmentation success, and 2) training a ranking function to prioritize successful partners.

We are given a training set $\mathcal{T} = \{(T^1, L^1), \ldots, (I^M, L^M)\}$ of M images labeled with their ground truth foreground masks, where T^i denotes an image and L^i denotes its mask. This set is not only disjoint from the candidate partner set $\mathcal{P}$ defined above, it also does *not* contain images of the same object category as what appears in $\mathcal{P}$ or the eventual novel queries. This is important, since it means our approach is required to learn generic cues indicative of cosegmentation compatibility, as opposed to object-specific cues. While object-specific cues are presumably easier to exploit, it may be impractical to train a model for every new object class of interest. Instead, all learning is done on data and classes disjoint from the weakly supervised image set $\mathcal{P}$.

Training a ranker for cosegmentation compatibility First, we apply the cosegmentation algorithm (Sec. 3.2) to every pair of images in $\mathcal{T}$. Each image in the training set acts as a "query" in turn, while the remaining images act as its candidate source images. Let (T^i_q, T^j_s) denote one such query-source pair comprised of training images T^i and T^j. For each pairing, we record the cosegmentation quality that results for T^i_q, that is, the intersection-over-union overlap score between the ground truth L^i and the cosegmentation estimate $L^{i^*}_{coseg}$ that results from optimizing Eqn. (5) with T^i as the query and T^j as the source. After computing these scores for all training pairs $(i, j) \in \{1, \ldots, M\}$, we have a set of training tuples $\langle T^i, T^j, o_{ij} \rangle$, where o_{ij} denotes the overlap score for pair i, j. The scores will vary across pairs depending on their compatibility.

Next, we generate a ranked list of source images for each training example. We use these M-length ranked lists to train a ranking function. As input, the learned ranking function f takes features computed on an image pair $\phi(I^q, I^s)$ (to be

defined below), and it returns as output a score predicting their cosegmentation compatibility. For simplicity we train a linear ranking function:

$$f(\phi(I^q, I^s)) = \boldsymbol{w}^T \phi(I^q, I^s), \tag{8}$$

where $\boldsymbol{w}$ is a vector of the same dimensionality as the feature space. To learn $\boldsymbol{w}$ from the training tuples, we want to constrain it to return higher scores for more compatible pairs. Let $\mathcal{O}$ be the set of *pairs* of all training tuples $\{(i,j),(i,k)\}$ for which $o_{ij} > o_{ik}$, for all $i = 1, \ldots, M$. Using the SVM Rank formulation of [24], we seek the projection of the data that preserves these training set orders, with a regularizer that favors a large margin between nearest-projected pairs:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||\boldsymbol{w}||_2^2 + C \sum \xi_{ijk}^2 \\ \text{s.t.} \quad & \boldsymbol{w}^T \phi(T^i, T^j) \geq \boldsymbol{w}^T \phi(T^i, T^k) + 1 - \xi_{ijk} \\ & \forall (i,j,k) \in \mathcal{O}, \end{aligned} \tag{9}$$

where the constant C balances the regularizer and constraints. In other words, the model should score a training pair with greater overlap higher than one with lower overlap.[2]

Defining features indicative of compatibility Next we define the features $\phi(I^q, I^s)$. Their purpose is to expose the images' compatibility for cosegmentation. We define features of two types: 1) *source image features* meant to capture the quality of the source in general, and 2) *inter-image features* meant to capture the likelihood of success in coupling a particular source and query. The former makes use of the single-image segmentation mask $L_{sing}^{s^*}$ from Sec. 3.1; the latter makes use of the cosegmentation estimates $L_{coseg}^{q^*}$ and $L_{coseg}^{s^*}$ from Sec. 3.2.

Source image features Ideally, we would like to cosegment with a source image that is easy to segment on its own, since then it has better ability to guide the foreground (when the query is compatible). Thus, our three source features aim to expose the predicted quality of the source's single-image segmentation:

- *Foreground-background separability*: We use $L_{sing}^{s^*}$ to compute separate color histograms for the (estimated) fg and bg regions, then record the χ^2 distance between the two histograms as a feature. More distinctive foregrounds will yield higher χ^2 distances.
- *Graph cuts uncertainty*: We use dynamic graph cuts [25] to measure each pixel's graph cut uncertainty. We bin these uncertainties from the foreground pixels of $L_{sing}^{s^*}$ into 5 bins and record this distribution as the feature. It captures how uncertain the single image segmentation is.
- *Number of connected components*: We record the number of connected components in $L_{sing}^{s^*}$ as a measure of how coherent the source's single-image segmentation is.

[2] Alternatively, one could use regression. However, ranking has the advantage of giving us more control over which training tuples are enforced, and it places emphasis only on the relative scores (not absolute values), which is what we care about for deciding which partner is best.

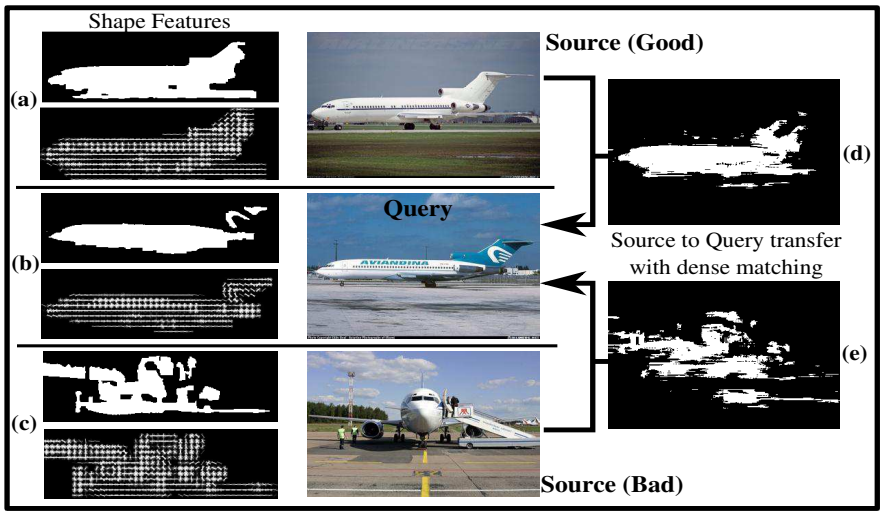

Fig. 2: Feature illustration. **Center:** an example query and two candidate source images. **(a-c)**: Cropped single-image segmentation masks (top) and corresponding HOGs (bottom). These features are good indicators of foreground shape similarity, as we see by comparing the query (b) to its good and bad source partners (a) and (c), respectively. **(d-e)**: Results of mask transfer with dense matching from the source image to the query image. The success of this transfer clearly depends on the compatibility between the query and source (i.e., it succeeds in (d) but fails in (e)).

Inter-image features To detect good partner candidates, the quality of the source image alone is insufficient; we also want to look explicitly at the compatibility of the particular input pair. Thus, our three inter-image features aim to reveal the predicted success of the pair's cosegmentation:

- *Foreground similarity*: We compute the foreground similarity between the source and query using their estimated foregrounds from single-image segmentation. Specifically, we record two χ^2 distances: one between their color histograms, and one between their SIFT bag-of-words histograms. By excluding background from this feature, we leave open the possibility to discover compatible partners with varying backgrounds.
- *Shape similarity*: We resize the cropped foreground region from $L^{s^*}_{sing}$ to the size of the cropped foreground region from $L^{q^*}_{sing}$. To gauge shape similarity, we record both the overlap between those masks as well as the L_2 distance on the HOG features computed on the original images at those masked positions (see Figure 2 (a-c)).
- *Dense matching quality*: We warp $L^{s^*}_{sing}$ to the query using the dense matching flow field $\mathcal{F}_{qs}$ from DSP [23]. To capture the matching quality, we record the overlap score between the transferred source mask and $L^{q^*}_{sing}$ (see Figure 2 (d-e)). Here the saliency-driven foreground masks and dense matching serve as two independent signals of alignment. If the two images permit an

accurate dense match that agrees with the saliency-based foreground, there is evidence that they are closely related. This compatibility cue offers some tolerance to foreground translation and scale variation in the two inputs.
– *GIST similarity*: To capture global layout similarity of the image pair, we record the L_2 distance between their GIST [26] descriptors.

Altogether, we have 7 and 6 feature dimensions for the source and inter-image features, respectively. We concatenate them to form the 13-dimensional $\phi(I^q, I^s)$ feature. These descriptors are used in training (Eqn. (9)). Analyzing the learned weights, we find that the dense matching quality, shape similarity, GIST similarity, and fg-bg separability are the most useful features for our task.

Predicting the partner for a novel image At test time, we are given a novel image I^q and the partner candidate set $\mathcal{P}$. We compute its descriptor $\phi(I^q, I^s)$ for every $I^s \in \mathcal{P}$, apply the learned ranking function, and select as its partner the one that maximizes the predicted cosegmentation compatibility:

$$I^{p^*} = \arg\max_{I^s \in \mathcal{P}} f(\phi(I^q, I^s)). \tag{10}$$

Finally, we return the foreground segmentation for I^q that results from cosegmenting the pair (I^q, I^{p^*}) using the algorithm in Sec. 3.2.

4 Results

Datasets: We evaluate our approach on two challenging publicly available datasets. The first is **MIT Object Discovery** (MIT), a dataset recently introduced for evaluating object foreground discovery through cosegmentation [8].[3] It consists of Internet images of objects from three classes: Airplane, Car, and Horse. The images within a class contain significant appearance and viewpoint variation. We use the 100-image per class subset designated by the authors to enable comparisons with multiple other existing methods. The second dataset is the **Caltech-28**, a subset of 28 of the Caltech-101[4] classes designated by [3] for study in weakly supervised joint segmentation. The 30 images per class originate from Internet search and cover an array of different objects.

Methods compared: We compare to results reported by a number of state-of-the-art cosegmentation techniques, namely [5, 7, 6, 8] on MIT and [3, 22, 9, 27] on Caltech-28. In addition, we implement several baseline techniques:

– **Single-Seg:** the saliency-based single-image approach defined in Sec. 3.1. This baseline reveals to what extent a query benefits at all from cosegmentation.

[3] `http://people.csail.mit.edu/mrub/ObjectDiscovery/`
[4] `http://www.vision.caltech.edu/ImageDatasets/Caltech101/`

	Single-Seg	Rand-Coseg	GIST-Coseg	Ours	Ours-Best k	Upper bound
Airplane	39.14	42.22	42.34	**45.81**	46.26	57.39
Car	46.76	52.47	50.95	**53.63**	54.31	61.81
Horse	49.82	51.69	**52.73**	50.18	52.86	63.52

Table 1: Overlap accuracy on the MIT Object Discovery dataset.

- **Rand-Coseg:** the cosegmentation approach defined in Sec. 3.2 applied with a random image *from the same object category* as the partner source image, averaged over 20 trials. This baseline helps illustrate the need to actively choose a cosegmentation partner among a weakly labeled dataset.
- **GIST-Coseg:** the same cosegmentation approach is applied using the source image that looks most similar to the query, in terms of GIST descriptors. This baseline highlights how image similarity alone—used in existing work [11, 8]—can be insufficient to determine good partners for cosegmentation.
- **Ours-Best k:** we apply our method, but instead of choosing the single maximally ranked image for cosegmentation, we refer to ground truth to pick the best partner from among the $k = 5$ source images our method ranks most highly.
- **Upper bound:** the upper bound for cosegmentation accuracy. We use ground truth to select the partner leading to the maximum overlap score for each query. This reveals the best accuracy any method could possibly attain for the cosegmentation partner selection problem.

As discussed above, we consider the weakly supervised setting. All baselines reference the exact same candidate set $\mathcal{P}$ as our method. Our method's training set $\mathcal{T}$ is always disjoint from $\mathcal{P}$, and furthermore $\mathcal{P}$ and $\mathcal{T}$ never overlap in object class. For example, when applying our method to Cars in the MIT data, we train it using only images of Airplanes and Horses.

To quantify segmentation accuracy, we use the standard intersection-over-union **overlap** accuracy score (Jaccard index), unless otherwise noted.

Implementation details: The color model GMMs consist of 5 mixture components. The scale parameters β are set automatically as the inverse of the mean of all individual distances. We use 50 visual words for the SIFT bag-of-words used in the inter-image foreground similarity, and 11 bins per color channel in all color histograms. The approximate run time per pair is between 10-12 seconds, which is dominated by the SIFT extraction step.

4.1 Results on MIT Object Discovery dataset

Table 1 shows our results against the baselines on all 3 classes in the MIT dataset. We observe several things from this result. First, the large gap between Single-Seg and the Upper bound underscores the fact that cosegmentation can indeed exceed the accuracy of single-image segmentation on challenging images—*if* suitable partners are used. Despite the images' diversity within a single class, the shared appearance in the optimally chosen partner is beneficial. Second, we see

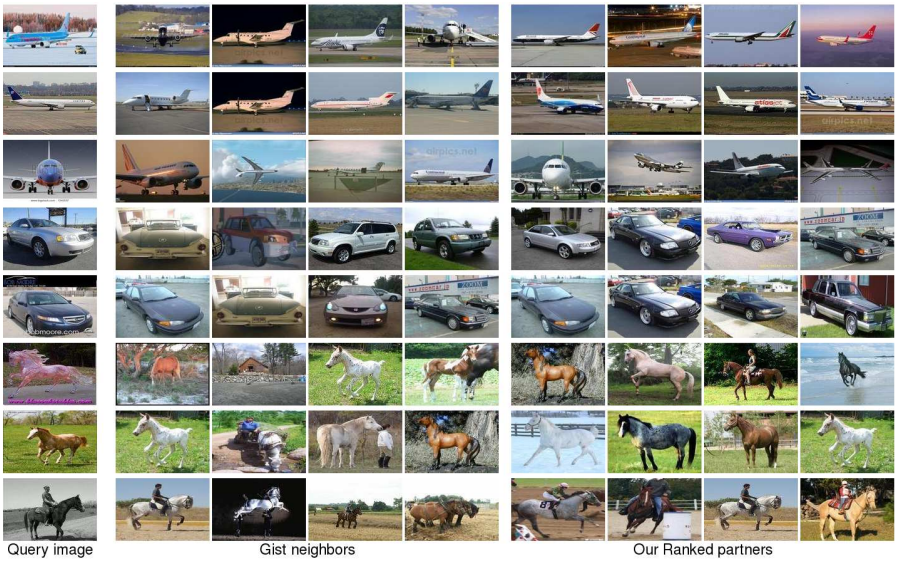

Fig. 3: Examples of the four top-ranked neighbors for a novel query, using either the GIST nearest neighbors (center block) or our learned ranking function (right block). Best viewed in color. While both methods can identify similar-looking source images among their top-ranked set, our method identifies partners that are more closely aligned in viewpoint or appearance and thus amenable to cosegmentation.

	Joulin et al. [5]	Joulin et al. [6]	Kim et al. [7]	Ours	Rub. et al. [8]
Airplane	15.26	11.72	7.9	45.81	**55.81**
Car	37.15	35.15	0.04	53.63	**64.42**
Horse	30.16	29.53	6.43	50.18	**51.65**

Table 2: Comparison to state-of-the-art cosegmentation methods on the MIT Object Discovery dataset, in terms of average overlap.

that our approach outperforms the baselines in nearly every case. This supports our key claim: it is valuable to actively choose an appropriate cosegmentation partner by learning the cues for success/failure. In two of three classes we outperform the GIST-Coseg baseline, showing that off-the-shelf image similarity is inferior to our learning approach for this problem. The Horse class is an exception, where we underperform the GIST-Coseg baseline. This is likely due to weak saliency priors in some of the more cluttered Horse images. Third, the fact that the Rand-Coseg approach does as well as it does (in fact, nearly as good as the GIST-Coseg method for Airplanes) indicates that many images of the same class offer *some* degree of help with cosegmentation. Hence, our method's gain is due to its fine-grained analysis of the candidate source images. Finally, the bump in accuracy we achieve if considering the k top-ranked source images (Ours-Best k) indicates that future refinements of our method should consider ways to exploit the ranked partners beyond the top-ranked example.

Figure 3 shows examples of the top-ranked partner images produced by the GIST-Coseg baseline and our approach, for a variety of query images in the MIT dataset. We see how our method's learning strategy pays off: it focuses on

		Single-Seg	Rand-Coseg	GIST-Coseg	Ours	Ours-Best k	Upper bound
Best	brain	73.31	72.43	72.54	**75.73**	76.09	76.22
	ferry	54.99	55.87	55.23	**57.64**	57.71	58.02
	dalmatian	39.58	39.13	38.15	**40.23**	40.94	41.59
	ewer	63.87	62.58	63.87	**65.86**	66.18	66.53
	joshua tree	53.04	54.05	54.45	**56.21**	57.12	57.52
	cougar face	58.19	57.39	56.51	**58.25**	58.53	59.05
	sunflower	70.48	70.10	69.77	**71.29**	72.07	73.48
	motorbike	**57.38**	55.86	55.79	57.21	58.12	58.59
	euphonium	57.72	57.25	58.32	**59.45**	60.27	60.28
	kangaroo	59.79	59.26	59.13	**60.24**	60.57	61.81
Worst	lotus	76.71	75.98	**78.38**	77.59	79.51	80.16
	grand piano	67.21	67.28	**67.93**	66.58	67.01	68.33
	crab	61.86	62.25	**62.11**	61.23	62.3	62.46
	watch	55.00	56.4	**57.72**	56.11	56.16	58.30

Table 3: Accuracy on the Caltech-28 dataset, in terms of average overlap. We show the 10 best and 4 worst performing classes (see Supp. for all classes).

source images that have more fine-grained compatability with the query image. The GIST neighbors are globally similar, but can be too distinct in viewpoint or appearance to assist in cosegmenting the query. In contrast, the partner source images retrieved by our ranking algorithm are better equipped to share a foreground model due to their viewpoint, appearance, and/or individual saliency.

Table 2 compares our result to several state-of-the-art cosegmentation methods.[5] Our method outperforms all the existing methods by a large margin, except the method of [8]. Our disadvantage in that case may be due to the fact that [8] operates over a joint graph of all images in the class at once, whereas we consider pairs of images for cosegmentation. This suggests future work to extend our algorithm, e.g., by using our compatibility predictions as weights within a multi-image cosegmentation graph.

4.2 Results on Caltech-28 dataset

Table 3 shows the results for the Caltech-28 dataset, in the same format as Table 1 above. Due to space constraints, we show just a sample of the 28 classes. Specifically, we display the top 10 cases where we most outperform GIST-Coseg and the bottom 4 cases where we most underperform GIST-Coseg. See the Supp. file for all classes.

The analysis is fairly similar to our MIT dataset results. We again see good support for actively selecting a cosegmentation partner: our method outperforms the Rand-Coseg and GIST-Coseg baselines in most cases. Overall, we outperform GIST-Coseg in 23 of the 28 classes, and Single-Seg in 20 of the 28 classes. Our method is also quite close to the Upper bound on this dataset, only 1.5 points away on average.

[5] These are the overlap accuracies reported in [8], where the authors applied the public source code to generate results for [5–7].

Method	Average Precision
Spatial Topic Model-Coseg [9]	67
Single-Seg	82.71
GrabCut-Coseg (see [3])	81.5
ClassCut-Coseg [3]	83.6
BPLR-Coseg [27]	85.6
Ours	**85.81**

Table 4: Comparison to state-of-the-art cosegmentation algorithms on the Caltech-28 dataset.

However, for the Caltech data, the gap between Single-Seg and the Upper bound—while still noticeably wider than the gap between our method and the Upper bound—is also narrowed considerably compared to the MIT data. This indicates that the Caltech images have greater regularity within a class and/or more salient foregrounds (both of which we find true upon visual inspection). In fact, Single-Seg can even outperform the cosegmentation methods in some cases (e.g., see motorbike). This finding agrees with previous reports in [8, 4]; while one hopes to see gains from the "more supervised" cosegmentation task, single-image segmentation can be competitive either when the intra-class variation is too high or the foreground is particularly salient.

Finally, we compare our method to state-of-the-art cosegmentation methods using their published numbers on the Caltech-28. Table 4 shows the results, in terms of average precision (the metric reported in the prior work). Our method is more accurate than all the previous results. Notably, all the prior cosegmentation results ([9, 3, 27] and the multi-image GrabCut [22] extension defined in [3]) indiscriminately use all the input images for joint segmentation, whereas our method selects the single most effective partner per query. This result is more evidence for the advantage of doing so.

5 Conclusions

Cosegmentation injects valuable implicit top-down information for segmentation, based on commonalities between related input images. Rather than assume that useful partners for cosegmentation will be known in advance, we propose to predict which pairs will work well together. Our results on two challenging datasets are encouraging evidence that it is worthwhile to actively focus cosegmentation on relevant pairs.

While so far we have focused on the weakly supervised setting—in which it is arguably harder to see impact, due to the potential relevance of *any* candidate partner—the approach is also applicable to the fully unsupervised setting, as we will explore in future work. We also plan to extend the algorithm from pairs to the multi-image joint segmentation scenario.

Acknowledgements This research is supported in part by ONR award N00014-12-1-0068.

References

1. Rother, C., Minka, T., Blake, A., Kolmogorov, V.: Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In: CVPR. (2006)
2. Winn, J., Jojic, N.: LOCUS: Learning Object Classes with Unsupervised Segmentation. In: ICCV. (2005)
3. Alexe, B., Deselaers, T., Ferrari, V.: Classcut for unsupervised class segmentation. In: ECCV. (2010)
4. Vicente, S., Rother, C., Kolmogorov, V.: Object cosegmentation. In: CVPR. (2011)
5. Joulin, A., Bach, F., Ponce, J.: Discriminative clustering for image co-segmentation. In: CVPR. (2010)
6. Joulin, A., Bach, F., Ponce, J.: Multi-class cosegmentation. In: CVPR. (2012)
7. Kim, G., Xing, E., Fei Fei, L., Kanade, T.: Distributed cosegmentation via submodular optimization on anisotropic diffusion. In: ICCV. (2011)
8. Rubinstein, M., Joulin, A., Kopf, J., Liu, C.: Unsupervised joint object discovery and segmentation in internet images. In: CVPR. (2013)
9. Cao, L., Fei-Fei, L.: Spatially coherent latent topic model for concurrent segmentation and classication of objects and scenes. In: ICCV. (2007)
10. Todorovic, S., Ahuja, N.: Unsupervised category modeling, recognition, and segmentation in images. PAMI **30** (2008) 2158–2174
11. Lee, Y.J., Grauman, K.: Collect-Cut: Segmentation with top-down cues discovered in multi-object images. In: CVPR. (2010)
12. Hochbaum, D., Singh, V.: An efficient algorithm for co-segmentation. In: ICCV. (2009)
13. Batra, D., Kowdle, A., Parikh, D., Luo, J., Chen, T.: iCoseg: Interactive Co-segmentation with Intelligen Scribble Guidance. In: CVPR. (2010)
14. Russell, B., Efros, A., Sivic, J., Freeman, W., Zisserman, A.: Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In: CVPR. (2006)
15. Faktor, A., Irani, M.: Clustering by composition: Unsupervised discovery of image categories. In: ECCV. (2012)
16. Liu, D., Xiong, Y., Pulli, K., Shapiro, L.: Estimating image segmentation difculty. In: Machine learning and data mining in pattern recognition. (2011)
17. Carreira, J., Sminchisescu, C.: CPMC: Automatic object segmentation using constrained parametric min-cuts. PAMI **34** (2012) 1312–1328
18. Endres, I., Hoiem, D.: Category independent object proposals. In: ECCV. (2010)
19. Ren, X., Malik, J.: Learning a classication model for segmentation. In: ICCV. (2003)
20. Jain, S., Grauman, K.: Predicting sufcient annotation strength for interactive foreground segmentation. In: ICCV. (2013)
21. Jiang, B., Zhang, L., Lu, H., Yang, C., Yang, M.H.: Saliency detection via absorbing markov chain. In: ICCV. (2013)
22. Rother, C., Kolmogorov, V., Blake, A.: Grabcut -interactive foreground extraction using iterated graph cuts. In: SIGGRAPH. (2004)
23. Kim, J., Liu, C., Sha, F., Grauman, K.: Deformable spatial pyramid matching for fast dense correspondences. In: CVPR. (2013)
24. Joachims, T.: Optimizing search engines with clickthrough data. In: KDD. (2002)
25. Kohli, P., Torr, P.H.S.: Measuring uncertainty in graph cut solutions. CVIU **112** (2008) 30–38
26. Torralba, A. Contextual priming for object detection **53** (2003) 169–191
27. Kim, J., Grauman, K.: Boundary preserving dense local regions. In: CVPR. (2011)