DeepCO^3: Deep Instance Co-segmentation by Co-peak Search and Co-saliency Detection

Kuang-Jui Hsu[1,2] Yen-Yu Lin[1] Yung-Yu Chuang[1,2]
[1]Academia Sinica, Taiwan [2]National Taiwan University, Taiwan

给定图像有特定类的目标实例类别，实例共同分割目标是识别出所有的实例并分割出每一个实例
分成两个子任务：co-peak搜索和实例mask分割，前者用CNN得到，后者用一个rank函数输出结果

Abstract

In this paper, we address a new task called instance co-segmentation. *Given a set of images jointly covering object instances of a specific category, instance co-segmentation aims to identify all of these instances and segment each of them,* i.e. *generating one mask for each instance. This task is important since instance-level segmentation is preferable for humans and many vision applications. It is also challenging because no pixel-wise annotated training data are available and the number of instances in each image is unknown. We solve this task by dividing it into two sub-tasks,* co-peak search *and* instance mask segmentation. *In the former sub-task, we develop a CNN-based network to detect the co-peaks as well as co-saliency maps for a pair of images. A co-peak has two endpoints, one in each image, that are local maxima in the response maps and similar to each other. Thereby, the two endpoints are potentially covered by a pair of instances of the same category. In the latter sub-task, we design a ranking function that takes the detected co-peaks and co-saliency maps as inputs and can select the object proposals to produce the final results. Our method for instance co-segmentation and its variant for object co-localization are evaluated on four datasets, and achieve favorable performance against the state-of-the-art methods. The source codes and the collected datasets are available at https://github.com/KuangJuiHsu/DeepCO3/.*

1. Introduction

Object co-segmentation aims to segment the common objects repetitively appearing in a set of images. It is a fundamental and active research topic in computer vision. As an important component of image content understanding, it is essential to many vision applications, such as semantic segmentation [48], image matching [4, 19, 25, 52, 60, 61], object skeletonization [8, 27], and 3D reconstruction [42].

Object co-segmentation has recently gained significant progress owing to the fast development of *convolutional neural networks* (CNNs). The CNN-based methods [21, 33, 62] learn the representation of common objects in an end-to-end manner and can produce object-level results of high quality. However, they do not explore instance-aware information, *i.e.* one segmentation mask for each instance rather than each class, which is more consistent with human perception and offers better image understanding, such as the locations and shapes of individual instances.

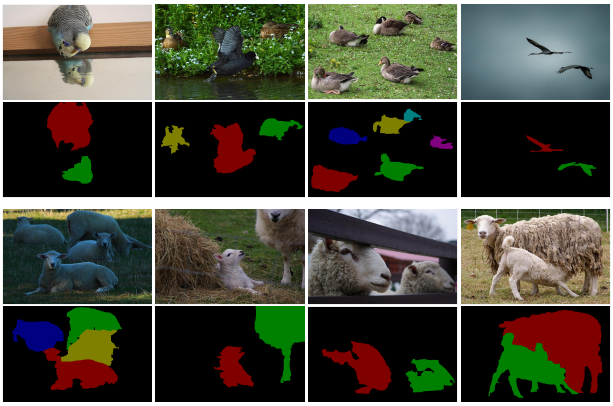

Figure 1. Two examples of instance co-segmentation on categories *bird* and *sheep*, respectively. An *instance* here refers to an object appearing in an image. In each example, the top row gives the input images while the bottom row shows the instances segmented by our method. The instance-specific coloring indicates that our method produces a segmentation mask for each instance.

In this work, we present a new and challenging task called *instance-aware object co-segmentation* (or *instance co-segmentation* for short). Two examples of this task are shown in Figure 1 for a quick start. Given a set of images of a specific object category with each image covering at least one instance of that category, instance co-segmentation aims to identify all of these instances and segment each of them out, namely one mask for each instance. Note that unlike semantic [18] or instance segmentation [65], no pixel-wise data annotations are collected for learning. The object category can be arbitrary and unknown, which means that no training images of that category are available in advance. Instance-level segments that can be obtained by solving this task are valuable to many vision applications, such as autonomous driving [2, 64], instance placement [31], image and sentence matching [26] or amodal segmentation [23].

Therefore, instance co-segmentation has a practical setting in input collection and better accomplishing it potentially advances the field of computer vision.

In this paper, we develop a CNN-based method for instance co-segmentation. Based on the problem setting, our method has no access to annotated instance masks for learning and cannot involve any pre-training process. Inspired by Zhou *et al*. [65]'s observation that object instances often cover the *peaks* in a response map of a classier, we design a novel *co-peak* loss to detect the common peaks (or co-peaks for short) in two images. The co-peak loss is built upon a 4D tensor that is learned to encode the inter-image similarity at every location. The co-peaks inferred from the learned 4D tensor correspond to two locations, one in each of the two images, where discriminative and similar features are present. Therefore, the two locations are potentially covered by two object instances. Using the co-peak loss alone may lead to unfavorable false positives and negatives. Thus, we develop the *affinity* loss and the *saliency* loss to complement the co-peak loss. The former carries out discriminative feature learning for the 4D tensor construction by separating the foreground and background features. The latter estimates the co-saliency maps to localize the co-salient objects in an image, and can make our model focus on co-peak search in co-salient regions. The three loss functions work jointly and can detect co-peaks of high quality. We design a ranking function taking the detected co-peaks and co-saliency maps as inputs and accomplish instance mask segmentation by selecting object proposals.

We make the following contributions in this work. First, we introduce a new and interesting task called instance co-segmentation. Its input is a set of images containing object instances of a specific category, and hence is easy to collect. Its output is instance-aware segments, which are desired in many vision applications. Thus, we believe instance co-segmentation worth exploring. Second, a simple and effective method is developed for instance co-segmentation. The proposed method learns a model based on the *fully convolutional network* (FCN) [40] by optimizing three losses, including the co-peak, affinity, and saliency losses. The learned model can reliably detect co-peaks and co-saliency maps for instance mask segmentation. Third, we collect four datasets for evaluating instance co-segmentation. The proposed method for instance co-segmentation and its variant for object co-localization [5,6,51,58,59] are extensively evaluated on the four datasets. Our method performs favorably against the state-of-the-art methods.

2. Related Work

Object co-segmentation. This task [13, 28, 45, 46, 54, 56, 57] aims to segment the common objects in images. Its major difficulties lie in large intra-class variations and background clutter. Most methods rely on robust features, such as handcrafted and deep learning based features, for addressing these difficulties. In addition, saliency evidence, including single-image saliency [12, 20, 27, 28, 46, 53] or multi-image co-saliency [3, 54, 57], has been explored to localize the salient and common objects. Recently, CNN-based methods [21, 33, 62] achieve better performance by joint representation learning and co-segmentation.

Despite effectiveness, the aforementioned methods do not provide instance-level results. In this work, we go beyond object co-segmentation and investigate instance co-segmentation. Our method can determine the number, locations, and contours of common instances in each image, and offers instance-aware image understanding.

Object co-localization. This task [5,6,51,58,59] discovers the common instances in images. Different from object co-segmentation, it is instance-aware. It detects and outputs the bounding box of a single instance in each image even if multiple instances are present in the image. Compared with object co-localization, instance co-segmentation identifies all instances in an image in the form of instance segments.

Instance-aware segmentation. Instance-aware segmentation includes *class-aware* [1, 7, 15, 17, 65] and *class-agnostic* [11, 24, 32] methods. Given training data of predefined categories, class-aware instance segmentation, aka instance segmentation, learns a model to seek each object instance belonging to one of these categories. A widely used way for instance segmentation is to first detect instance bounding boxes and then segment the instances within the bounding boxes [7,15–17,35,38,43]. Another way is to directly segment each instance without bounding box detection [1,30,36,39,65]. While most methods for instance segmentation are supervised, Zhou *et al*. [65] present a weakly supervised one. All these methods for instance segmentation rely on training data to learn the models. Despite the effectiveness and efficiency in testing, their learned models are not applicable to unseen object categories.

In practice, it is difficult to enumerate all object categories of interest in advance and prepare class-specific training data, which limits the applicability of class-aware instance segmentation. Class-agnostic instance segmentation [11,24,32] aims at segmenting object instances of arbitrary categories, and has drawn recent attention. It is challenging because it involves both generic object detection and segmentation. Instance co-segmentation is highly related to class-agnostic instance segmentation in the sense that both of them can be applied to arbitrary and even unseen object categories. However, existing class-agnostic methods require annotated training data in the form of object contours. On the contrary, our method for instance co-segmentation explores the mutual information regarding the common instances in given images, and does not need any pre-training procedure on additional data annotations. Thus, our method has better generalization.

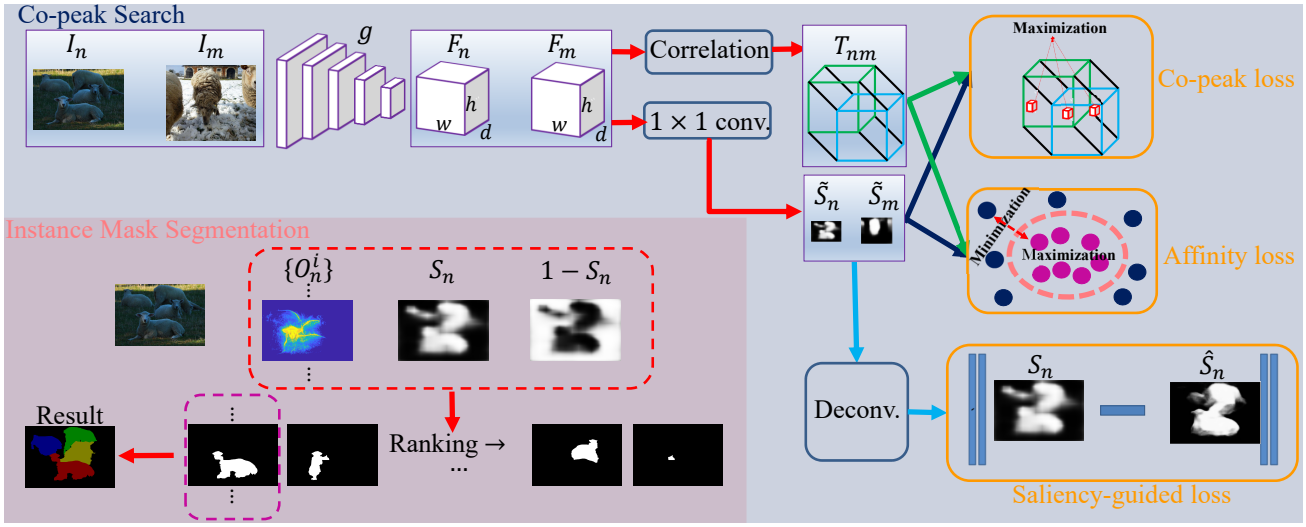

Figure 2. Overview of our method, which contains two stages, *co-peak search* within the blue-shaded background and *instance mask segmentation* within the red-shaded background. For searching co-peaks in a pair of images, our model extracts image features, estimates their co-saliency maps, and performs feature correlation for co-peak localization. The model is optimized by three losses, including the co-peak loss ℓ_t, the affinity loss ℓ_a, and the saliency loss ℓ_s. For instance mask segmentation, we design a ranking function taking the detected co-peaks, the co-saliency maps, and the object proposals as inputs, and select the top-ranked proposal for each detected instance.

3. Proposed Method

In this section, we give an overview of our method, describe its components, *co-peak search* and *instance mask segmentation*, and provide the implementation details.

3.1. Overview

Suppose that a set of images $D = \{I_n\}_{n=1}^N$ consisting of object instances of a particular category is given, where $I_n \in \mathbb{R}^{W \times H \times c}$ is the nth image while W, H, and c are the width, the height, and the number of channels of I_n, respectively. The goal of instance co-segmentation is to identify and segment each of all instances in D. Note that no training data with pixel-wise annotations are provided. In addition, both the object category and the number of instances in each image are unknown.

In the proposed method, we decompose instance co-segmentation into two stages, *i.e. co-peak search* and *instance mask segmentation*. The overview of our method is shown in Figure 2, where the two stages are highlighted with the blue-shaded area and the red-shaded backgrounds, respectively.

At the stage of co-peak search, we aim to seek co-peaks in the response maps of two images, where a co-peak corresponds two discriminative and similar points, one in each image, so that each point is potentially within an object instance. We design a network model for co-peak detection. The front part of our model is a fully convolutional network (FCN) g, which extracts the feature maps of input images. After feature extraction, our model is split into two streams. One stream correlates the feature maps of two images for co-peak localization. The other estimates the co-saliency maps of input images, which in turn enforces FCN g to generate more discriminative feature maps. Our model is optimized by three novel losses, including the co-peak loss ℓ_t, the affinity loss ℓ_a, and the saliency loss ℓ_s. After optimization, co-peaks are detected and co-saliency maps are estimated. At the stage of instance mask segmentation, we design a ranking function that takes the detected co-peaks, the estimated co-saliency maps, and the instance proposals into account, and yield one mask for each detected instance.

3.2. Co-peak search

As shown in Figure 2, our model takes a pair of images, I_n and I_m, from D as input at a time. It first extracts the feature maps $F_n \in \mathbb{R}^{w \times h \times d}$ for I_n, where w, h, and d are the width, the height, and the number of channels, respectively. Similarly, feature maps $F_m \in \mathbb{R}^{w \times h \times d}$ are yielded for I_m. Our model is then divided into two streams. One stream performs correlation between F_n and F_m, and yields a 4D correlation tensor $T_{nm} \in \mathbb{R}^{w \times h \times w \times h}$. Each element $T_{nm}(i, j, s, t) = T_{nm}(\mathbf{p}, \mathbf{q})$ records the normalized inner product between the feature vectors stored at two spatial locations, *i.e.* $\mathbf{p} = [i, j]$ in F_n and $\mathbf{q} = [s, t]$ in F_m. The other stream employs a 1×1 convolutional layer to estimate the co-saliency map $\tilde{S}_k \in \mathbb{R}^{w \times h}$ of I_k, and adopts deconvolution layers to generate a high-resolution co-saliency map $S_k \in \mathbb{R}^{W \times H}$, for $k \in \{n, m\}$. We design three loss functions, including the co-peak loss ℓ_t, the affinity loss ℓ_a, and the saliency loss ℓ_s, to derive the network, leading to the

following object function

$$\mathcal{L}(\mathbf{w}) = \lambda_t \sum_{n=1}^{N} \sum_{m \neq n} \ell_t(I_n, I_m; \mathbf{w}) + \lambda_a \sum_{n=1}^{N} \sum_{m \neq n} \ell_a(I_n, I_m; \mathbf{w}) + \sum_{n=1}^{N} \ell_s(I_n; \mathbf{w}), \quad (1)$$

where $\mathbf{w}$ is the set of learnable parameters of the network. Nonnegative weights λ_t and λ_a control the relative importance among the three losses. They are fixed to 0.5 and 0.1 in this work, respectively. The co-peak loss ℓ_t stimulates co-peak detection. The affinity loss ℓ_a refers to the co-saliency maps and enables discriminative feature learning. The saliency loss ℓ_s working with the other two losses carries out co-saliency detection and hence facilitates instance co-segmentation. The three losses are elaborated in the following.

3.2.1 Co-peak loss ℓ_t

This loss aims to stimulate co-peak detection. A co-peak consists of to two points, one in each of I_n and I_m. Since a co-peak covered by a pair of instances of the same object category is desired, the two points of the co-peak must be inside the object and similar to each other. Therefore, both *intra-image saliency* and *inter-image correlation* are taken into account in this loss.

As shown in Figure 2, our two-stream network produces the intra-image saliency maps $\tilde{S}_n$ and $\tilde{S}_m$ in one stream and inter-image correlation map T_{nm} in the other stream. To jointly consider the two types of information, a saliency-guided correlation tensor $T^s_{nm} \in \mathbb{R}^{w \times h \times w \times h}$ is constructed with its elements defined below

$$T^s_{nm}(\mathbf{p}, \mathbf{q}) = \tilde{S}_n(\mathbf{p}) \tilde{S}_m(\mathbf{q}) T_{nm}(\mathbf{p}, \mathbf{q}), \quad (2)$$

where $\mathbf{p} \in \mathcal{P}$, $\mathbf{q} \in \mathcal{P}$, and $\mathcal{P}$ is the set of all spatial coordinates of the feature maps. In Eq. (2), $\tilde{S}_n(\mathbf{p})$ is the saliency value of $\tilde{S}_n$ at point $\mathbf{p}$, and $\tilde{S}_m(\mathbf{q})$ is similarly defined.

To have more reliable keypoints to reveal object instances, we define a co-peak as a local maximum in T^s_{nm} within a 4D local window of size $3 \times 3 \times 3 \times 3$. Suppose that $(\mathbf{p}, \mathbf{q})$ is a peak in T^s_{nm}. Both point $\mathbf{p}$ in F_n and point $\mathbf{q}$ in F_m are salient, and they are the most similar to each other in a local region. The former property implies that the two points probably reside in two salient object instances. The latter one reveals that the two instances are likely of the same class, since they have similar parts. Based on above discussion, the co-peak loss used to stimulate reliable co-peaks is defined by

$$\ell_t(I_n, I_m) = -log\left(\frac{1}{|\mathcal{M}_{nm}|} \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{M}_{nm}} T^s_{nm}(\mathbf{p}, \mathbf{q})\right), \quad (3)$$

where $\mathcal{M}_{nm}$ is the set of co-peaks.

3.2.2 Affinity loss ℓ_a

The co-peak loss refers to the feature maps of the images, so discriminative features that can separate instances from background are preferable. Besides, the co-peak loss is applied to the locations of co-peaks, and features on other locations are ignored. The affinity loss is introduced to address the two issues. It aims to derive the features with which pixels in the salient regions are similar to each other while being distinct from those in the background. For a pair of images I_n and I_m, a loss $\tilde{\ell}_a(I_n, I_m)$ is defined by

$$\tilde{\ell}_a(I_n, I_m) = \sum_{\mathbf{p} \in \mathcal{P}} \sum_{\mathbf{q} \in \mathcal{P}} \tilde{S}_n(\mathbf{p}) \tilde{S}_n(\mathbf{q}) \left(1 - T_{nm}(\mathbf{p}, \mathbf{q})\right) + \alpha (\tilde{S}_n(\mathbf{p}) - \tilde{S}_n(\mathbf{q}))^2 T_{nm}(\mathbf{p}, \mathbf{q}), \quad (4)$$

where constant α is empirically set to 4. In Eq. (4), the first term penalizes the case of low similarity between two salient pixels, while the second term prevents high similarity between a salient pixel and a non-salient pixel. The proposed affinity loss generalizes $\tilde{\ell}_a$ in Eq. (4) to consider both inter-image and intra-image affinities and is defined by

$$\ell_a(I_n, I_m) = \tilde{\ell}_a(I_n, I_m) + \tilde{\ell}_a(I_n, I_n) + \tilde{\ell}_a(I_m, I_m). \quad (5)$$

3.2.3 Saliency loss ℓ_s

This term aims to identify the salient regions and can guide the training of our model. Following the studies of object co-segmentation [27, 28, 46, 53], we utilize an off-the-shelf method for saliency detection. The resultant saliency maps can serve as the object prior. In this work, we adopt the unsupervised method, SVFSal [63], which produces the saliency map $\hat{S}_n$ for image I_n. Note that the resolutions of $\hat{S}_n$ and I_n are the same. Thus, the deconvolutional layers are employed to increase the resolution. Following [22], the saliency loss ℓ_s applied to image I_n is defined by

$$\ell_s(I_n) = \sum_{\mathbf{p} \in I_n} \rho_n(\mathbf{p}) \| S_n(\mathbf{p}) - \hat{S}_n(\mathbf{p}) \|_2^2, \quad (6)$$

where $\mathbf{p}$ indexes the pixels of I_n, $\rho_n(\mathbf{p})$ is a weight representing the importance of pixel $\mathbf{p}$, and S_n is the predicted saliency map for I_n by our model. The weight $\rho_n(\mathbf{p})$ deals with the imbalance between the salient and non-salient areas. It is set to $1 - \varepsilon$ if pixel $\mathbf{p}$ resides in the salient region, and ε otherwise, where ε is the ratio of the salient area to the whole image. The mean value of $\hat{S}_n$ is used as the threshold to divide $\hat{S}_n$ into the salient and non-salient regions. In this way, the salient and non-salient regions contribute equally in Eq. (6). As shown in Figure 2, except for the deconvolutional layers, our model used to produce maps $\{S_n\}$ is derived by the three losses jointly. Thus, $\{S_n\}$ derived with both intra- and inter-image cues are called co-saliency maps. This prior term is helpful as it compensates for the lack of supervisory signals in instance co-segmentation.

3.3. Instance mask segmentation

After optimizing Eq. (1), we simply use the detected peaks on the estimated co-saliency maps as the final co-peaks, because detecting the co-peaks on all possible image pairs is complicated. Thus, the peaks $\{p_n^i\}_{i=1}^M$ of each image I_n are collected, where M is the number of the peaks. We adopt the method called *peak back-propagation* [65] to infer an instance-aware heat map O_n^i for each peak p_n^i. The map O_n^i is supposed to highlight the instance covering p_n^i. An example is given in Figure 2.

For instance mask generation, we utilize an unsupervised method, called *multi-scale combinatorial grouping* (MCG) [44], to produce a set of instance proposals for image I_n. With the heat maps $\{O_n^i\}_{i=1}^M$ and the co-saliency map S_n, we extend the proposal ranking function in [65] by further taking the co-saliency cues into account, and select the top-ranked proposal as the mask for each detected peak. Specifically, given the maps O_n^i and S_n, the ranking function R applied to an instance proposal P is defined by

$$R(P) = \beta(O_n^i * S_n) * P + (O_n^i * S_n) * \hat{P} - \gamma(1 - S_n) * P, \quad (7)$$

where $\hat{P}$ is the contour of the proposal P and operator $*$ is the Frobenius inner product between two matrices. The coefficients β and γ are set to 0.8 and 10^{-5}, respectively. In Eq. (7), three terms, *i.e*. the instance-aware, contour-preserving, and object-irrelevant terms, are included. The instance-aware term prefers the proposals that cover the regions with high responses in O_n^i and high saliency in S_n. The contour-preserving term focuses on the fine-detailed boundary information. The background map, $1 - S_n$, is used in the object-irrelevant term to suppress background regions. Compared with the ranking function in [65], ours further exploits the properties of instance co-segmentation, *i.e*. the high co-saliency values in object instances, and can select more accurate proposals. Following a standard protocol of instance segmentation, we perform *non-maximum suppression* (NMS) to remove the redundancies.

3.4. Implementation details

We implement the proposed method using *MatConvNet* [55]. VGG-16 [49] is adopted as the feature extractor g. It is pre-trained on the ImageNet [47] dataset, and is updated during optimizing Eq. (1). The same network architecture is used in all experiments. Note that the objective in Eq. (1) involves all image pairs. Direct optimization is not feasible due to the limited memory size. Thereby, we adopt the *piecewise training* scheme [50]. Namely, only a subset of images is considered in each epoch, and the subset size is set to 6 in this work. The learning rate, weight decay, and momentum are set to 10^{-6}, 0.0005, and 0.9, respectively. The optimization procedure stops after 40 epochs. We choose ADAM [29] as the optimization solver. All images are resized to the resolution 448×448 in advance. We resize the instance co-segmentation results back to the original image resolution for performance evaluation.

dataset	(a)	(b)	(c)	(d)	(e)
COCO-VOC	12	1281	3151	106.8	2.5
COCO-NONVOC	32	3130	8303	91.8	2.7
VOC12	18	891	2214	178.2	2.5
SOC	5	522	835	29.0	1.6

Table 1. Some statistics of the four collected datasets, including (a) the number of classes, (b) the number of images, (c) the number of instances, (d) the average number of images per class, and (e) the average number of instances per image.

4. Experimental Results

In this section, our method for instance co-segmentation and its variant for co-localization are evaluated. First, the adopted datasets and evaluation metrics are described. Then, the competing methods are introduced. Finally, the comparison results are reported and analyzed.

4.1. Dataset collection

As instance co-segmentation is a new task, no public benchmarks exist. Therefore, we establish four datasets with pixel-wise instance annotations by collecting images from three public benchmarks, including the MS COCO [37], PASCAL VOC 2012 [9, 14], and SOC [10] datasets. The following pre-processing is applied to each dataset. First, we remove the images where objects of more than one category are present. Second, we discard the categories that contain less than 10 images. The details of collecting images from each dataset are described below.

MS COCO dataset. We collect images from the training and validation sets of the MS COCO 2017 object detection task. As MS CCCO is a large-scale dataset, we further remove the images that do not contain at least two instances. Total 44 categories remain. Some competing methods are pre-trained on PASCAL VOC 2012 dataset. For the ease of comparison, we divide the 44 categories into two disjoint sets, *COCO-VOC* and *COCO-NONVOC*. The former contains 12 categories covered by the PASCAL VOC 2012 dataset, while the latter contains the rest.

PASCAL VOC 2012 dataset. Because few pixel-wise instance annotations are available in the PASCAL VOC 2012 dataset, we adopt the augmented VOC12 dataset [14], which has 18 object categories after dataset preprocessing.

SOC dataset. SOC [10] is a newly collected dataset for saliency detection. It provides image-level labels and instance-aware annotations. After preprocessing, only five object categories remain because many images contain object instances of multiple categories and some categories have less than 10 images.

method	year	trained	COCO-VOC		COCO-NONVOC		VOC12		SOC	
			$mAP^r_{0.25}$	$mAP^r_{0.5}$	$mAP^r_{0.25}$	$mAP^r_{0.5}$	$mAP^r_{0.25}$	$mAP^r_{0.5}$	$mAP^r_{0.25}$	$mAP^r_{0.5}$
CLRW [51]	CVPR 2014	×	33.3	13.7	24.6	10.7	29.2	10.5	34.9	15.6
UODL [5]	CVPR 2015	×	9.6	2.2	8.5	1.8	9.4	2.0	11.0	2.7
DDT [58]	IJCAI 2017	×	31.4	10.1	25.7	9.7	30.7	8.8	43.0	25.7
DDT+ [59]	arXiv 2017	×	31.7	10.6	26.0	10.1	33.6	9.4	39.6	22.4
DFF [6]	ECCV 2018	×	30.8	11.6	22.6	7.3	27.7	13.7	42.3	17.0
NLDF [41]	CVPR 2017	√	39.1	18.2	23.9	8.5	34.3	12.7	49.5	21.6
C2S-Net [34]	ECCV 2018	√	39.6	13.4	25.1	7.6	30.1	10.7	37.0	12.5
PRM [65]	CVPR 2018	√	44.9	14.6	-	-	45.3	14.8	-	-
Ours	-	×	52.6	21.1	35.3	12.3	45.6	16.7	54.2	26.0

Table 2. Performance of instance co-segmentation on the four collected datasets. The numbers in red and green show the best and the second best results, respectively. The column "trained" indicates whether additional training data are used.

The statistics and the abbreviations of the four collected datasets are given in Table 1. Note that our method can work on images containing one or multiple instances of the common object category. The SOC dataset helps test this issue. As shown in Table 1, the average number of instances in SOC is 1.6, less than 2. It shows that there exist many images in this dataset with only one object instance. Please refer to the supplementary material for more details and some image samples of the four collected datasets.

4.2. Evaluation metrics

For instance co-segmentation, mean average precision (mAP) [15] is adopted as the performance measure. Following [65], we report mAP using the IoU thresholds at 0.25 and 0.5, denoted as $mAP^r_{0.25}$ and $mAP^r_{0.5}$, respectively.

For object co-localization, the performance measure CorLoc [5,6,51,58,59] is used as the evaluation metric. The measure CorLoc is designed for evaluating the results in the form of object bounding boxes. For comparing with methods whose output is object or instance segments, we extend CorLoc to $CorLoc^r$ to evaluate the results in the form of object segments.

4.3. Competing methods

As instance co-segmentation is a new task, there are no existing methods for performance comparison. We adopt two strategies for comparing our method with existing ones. First, we consider competing methods of three categories, including *object co-localization*, *class-agnostic saliency segmentation*, and *weakly supervised instance segmentation*. For methods of the three categories, we convert their predictions into the results in the form of instance co-segmentation, namely one segment mask for each detected instance. In this way, our method can be compared with these methods on the task of instance co-segmentation.

Second, we compare our method with methods of all the aforementioned three categories on the task of object co-localization. To this end, we need to convert the output of each compared method into the results in the form of object co-localization, namely the object bounding box with the highest confidence in each image.

In the two strategies of method comparison, two types of prediction conversion are required, including converting a bounding box to an instance segment and its inverse direction. Unless further specified, we adopt the following way to convert a bounding box prediction to an instance segment. Given a bounding box in an image, we apply MCG [44] to that image to generate a set of instance proposals, and retrieve the proposal with the highest IoU with the bounding box to represent it. On the other hand, it is easy to concert a given instance segment to a bounding box. We simply use the bounding box of that instance segment to represent it. In the following, the selected competing methods from each of the three categories are specified.

Object co-localization. We choose the state-of-the-art methods of this category for comparison, including CLRW [51], UODL [5], DDT [58], DDT+ [59], and DFF [6]. The first two methods, CLRW and UODL, output all bounding boxes with their scores, but cannot determine the number of instances in each image. Thus, we pick the top-scored bounding boxes as many as the instances detected by our method, and similarly apply NMS to remove redundancies. The last three methods, DDT, DDT+, and DFF, first produce the heat maps to highlight objects, then convert the heat maps into the binary masks by using their proposed mechanisms, and finally take the bounding boxes of the connected components on the binary masks.

Class-agnostic instance segmentation (CAIS). We select two powerful methods, NLDF [41] and C2S-Net [34], of this category as the competing methods. The algorithm proposed in [32] is used to convert the saliency contours generated by NLDF and C2S-Net into the results in the form of instance co-segmentation.

Weakly supervised instance segmentation (WSIS). The WSIS method, PRM [65], is trained on the PASCAL VOC 2012 dataset, and it cannot be applied to the images whose categories are not covered by the PASCAL VOC 2012

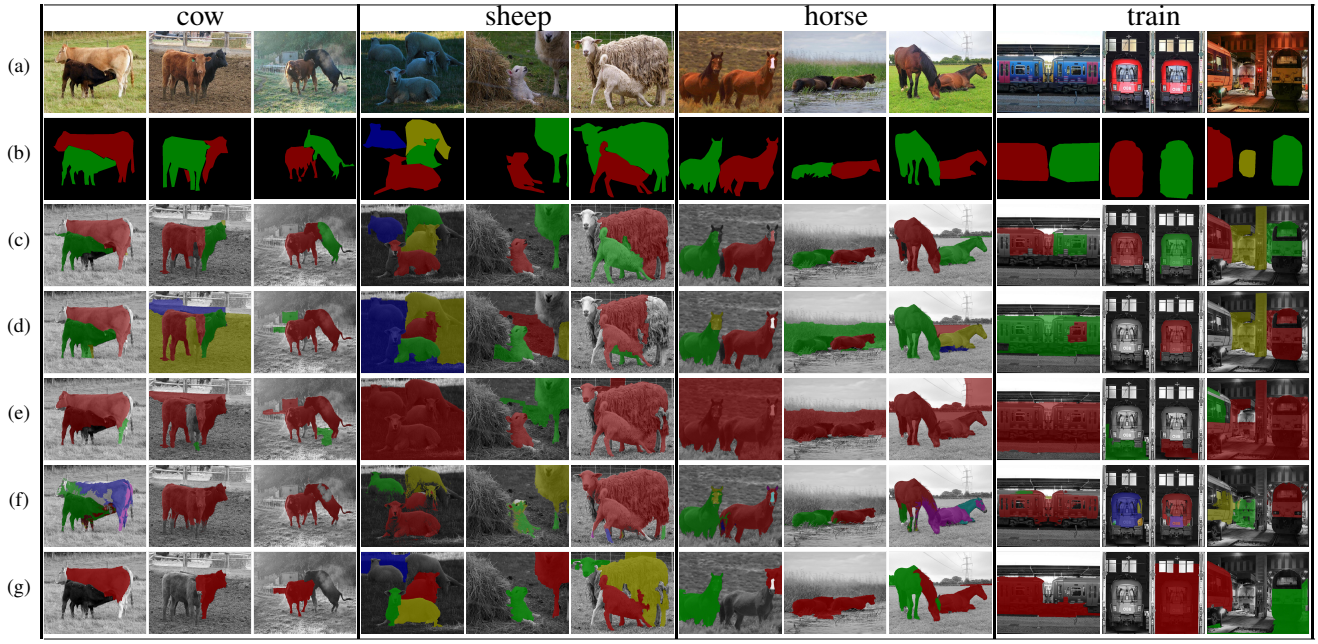

Figure 3. Results of instance co-segmentation on four object categories, *i.e.cow*, *sheep*, *horse*, and *train*, of the COCO-VOC dataset. (a) Input images. (b) Ground truth. (c) ∼ (g) Results with instance-specific coloring generated by different methods including (c) our method, (d) CLRW [51], (e) DFF [6], (f) NLDF [41], and (g) PRM [65], respectively.

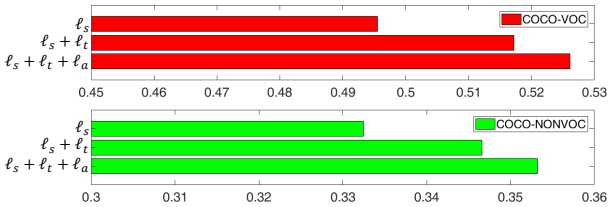

Figure 4. Performance in $\text{mAP}^r_{0.25}$ with different loss function combinations on the COCO-VOC and COCO-NONVOC datasets.

	COCO-VOC		COCO-NONVOC	
	$\text{mAP}^r_{0.25}$	$\text{mAP}^r_{0.5}$	$\text{mAP}^r_{0.25}$	$\text{mAP}^r_{0.5}$
w/o co-saliency map	33.5	12.4	25.3	8.3
w co-saliency map	52.6	21.1	35.3	12.3

Table 3. Performance of our method working with the proposal ranking function without or with the co-saliency information on the COCO-VOC and COCO-NONVOC datasets.

dataset. Therefore, PRM is compared with our method only on the COCO-VOC and VOC12 datasets.

4.4. Instance co-segmentation

For the ease of performance analysis, we divide the evaluated methods into two groups, *i.e.* `trained` and `non-trained`. The group `trained` includes NLDF [41], C2S-Net [34] and PRM [65]. Methods of this group require additional training data other than the input to instance co-segmentation. The other group `non-trained` contains our method and the rest of the competing methods. Methods of group `non-trained` have access to only the input to instance co-segmentation.

Our method and all competing methods are evaluated on the four collected datasets. Their performance is reported in Table 2. The proposed method outperforms the competing methods of group `non-trained` by large margins even though all of them access the same data. We attribute the performance gain yielded by our method to feature learning enabled CNNs. The competing methods of group `non-trained` adopt pre-defined features, and cannot well deal with complex and diverse intra-class variations and background clutters. On the contrary, our method leverages CNNs to carry out feature learning and instance co-segmentation simultaneously, leading to much better performance. Although the methods of group `trained` have access to additional training data, ours still reaches more favorable results. The main reason is that our method explores co-occurrent patterns via co-peak detection when images for instance co-segmentation are available, while the methods of group `trained` fix their models after training on additional data and cannot adapt themselves to newly given images for instance co-segmentation.

To gain the insight into the quantitative results, Figure 3 visualizes the qualitative results generated by our method, CLRW [51], DFF [6], NLDF [41], and PRM [65]. The major difficulties of instance segmentation lie in instance mutual occlusions, intra-class variations, and clut-

Figure 5. Seven examples, one in each row, of the co-localization results by our method on the COCO-NONVOC dataset.

tered scene. As shown in Figure 3(c), our method still works well when instance mutual occlusions occur on categories *cow*, *sheep*, and *horse* and large intra-class variations and cluttered scene are present on category *train*. In Figure 3(d), CLRW yields some false alarms in the background while has false negatives on category *train*. In Figure 3(e), DFF cannot well address instance mutual occlusions due to computing connected components for instance identification. In Figure 3(f) and Figure 3(g), NLDF and CRP perform favorably against other competing methods, but still suffer from over-segmentation and misses, respectively.

Ablation studies. We analyze the proposed objective consisting of three loss functions in Eq. (1) on the COCO-VOC and COCO-NONVOC datasets, and report the results in Figure 4. Except loss ℓ_s, the other two losses, ℓ_t and ℓ_a, are added one by one. When ℓ_t is included, the performance gains are significant on both datasets. It implies that ℓ_t for reliable co-peak search is important in our method. Once ℓ_a is added, the performance is moderately enhanced, which means that discriminative feature learning is helpful for instance co-segmentation. In addition to the objective, the effect of referring to co-saliency maps in proposal ranking is analyzed in Table 3. The results clearly point out that information from co-saliency detection is crucial to proposal ranking. It is not surprised. Since co-peaks identify the keypoints within instances, we still need the evidence from co-saliency maps to reveal the corresponding instances.

method	year	trained	COCO-VOC	COCO-NONVOC	VOC12	SOC
CLRW [51]	CVPR 2014	×	33.4	31.6	29.9	30.9
UODL [5]	CVPR 2015	×	12.3	12.7	9.5	10.3
DDT [58]	IJCAI 2017	×	30.0	27.4	25.0	16.7
DDT+ [59]	PR 2019	×	29.5	25.8	23.7	18.4
DFF [6]	ECCV 2018	×	32.3	30.5	28.7	22.9
NLDF [41]	CVPR 2017	√	51.2	31.0	39.2	42.0
C2S-Net [34]	ECCV 2018	√	39.0	28.4	31.1	32.9
PRM [65]	CVPR 2018	√	18.1	-	23.3	-
Ours	-	×	49.6	34.3	39.2	43.1

Table 4. Performance of object co-localization on the four datasets. The numbers in red and green indicate the best and the second best results, respectively. The column "trained" indicates whether additional training data are used.

4.5. Object co-localization

We evaluate our method and the competing methods for object co-localization in the four datasets we collected. For our method, we pick the top-ranked proposal in each image when evaluating the performance in CorLoc^r. Table 4 reports the performance of all the compared methods. Our method achieves the comparable or even better performance, even though it is not originally designed for object co-localization. Seven examples of object co-localization by our method are shown in Figure 5, where accurate instance masks and the corresponding bounding boxes are discovered by our method.

5. Conclusions

In this paper, we present an interesting and challenging task called instance co-segmentation, and propose a CNN-based method to effectively solve it without using additional training data. We decompose this task into two sub-tasks, including co-peak search and instance mask segmentation. In the former sub-task, we design three novel losses, co-peak, affinity, and saliency losses, for joint co-peak and co-saliency map detection. In the latter sub-task, we develop an effective proposal ranking algorithm, and can retrieve high-quality proposals to accomplish instance co-segmentation. Our method for instance co-segmentation and its variant for object co-localization are extensively evaluated on the four collected datasets. Both quantitative and qualitative results show that our method and its variant perform favorably against the state-of-the-arts. In the future, we plane to integrate the proposed method into more high-level tasks, such as autonomous driving, visual question answering, image and sentence matching where instance-aware annotations are valuable.

Acknowledgments. This work was supported in part by Ministry of Science and Technology (MOST) under grants 107-2628-E-001-005-MY3 and 108-2634-F-007-009, and MOST Joint Research Center for AI Technology and All Vista Healthcare under grant 108-2634-F-002-004.

References

[1] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017.

[2] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation for autonomous driving. In *CVPR Workshop*, 2017.

[3] Kai-Yueh Chang, Tyng-Luh Liu, and Shang-Hong Lai. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *CVPR*, 2011.

[4] Hsin-I Chen, Yen-Yu Lin, and Bing-Yu Chen. Co-segmentation guided hough transform for robust feature matching. *TPAMI*, 2015.

[5] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *CVPR*, 2015.

[6] Edo Collins, Radhakrishna Achanta, and Sabine Süsstrunk. Deep feature factorization for concept discovery. In *ECCV*, 2018.

[7] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016.

[8] Jifeng Dai, Ying Nian Wu, Jie Zhou, and Song-Chun Zhu. Cosegmentation and cosketch by unsupervised learning. In *ICCV*, 2013.

[9] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 2010.

[10] Deng-Ping Fan, Ming-Ming Cheng, Jiang-Jiang Liu, Shang-Hua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In *ECCV*, 2018.

[11] Ruochen Fan, Qibin Hou, Ming-Ming Cheng, Tai-Jiang Mu, and Shi-Min Hu. S^4Net: Single stage salient-instance segmentation. In *CVPR*, 2019.

[12] H. Fu, D. Xu, B. Zhang, S. Lin, and R. Ward. Object-based multiple foreground video co-segmentation via multi-state selection graph. *TIP*, 2015.

[13] Junwei Han, Rong Quan, Dingwen Zhang, and Feiping Nie. Robust object co-segmentation using background prior. *TIP*, 2018.

[14] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[15] Bharath Hariharan, Pablo Arbelaez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

[16] Zeeshan Hayder, Xuming He, and Mathieu Salzmann. Boundary-aware instance segmentation. In *CVPR*, 2017.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[18] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Augmented multiple instance regression for inferring object contours in bounding boxes. *TIP*, 2014.

[19] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Robust image alignment with multiple feature descriptors and matching-guided neighborhoods. In *CVPR*, 2015.

[20] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Weakly supervised saliency detection with a category-driven map generator. In *BMVC*, 2017.

[21] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Co-attention CNNs for unsupervised object co-segmentation. In *IJCAI*, 2018.

[22] Kuang-Jui Hsu, Chung-Chi Tsai, Yen-Yu Lin, Xiaoning Qian, and Yung-Yu Chuang. Unsupervised CNN-based co-saliency detection with graphical optimization. In *ECCV*, 2018.

[23] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander Schwing. SAIL-VOS: Semantic amodal instance level video object segmentation - a synthetic dataset and baselines. In *CVPR*, 2019.

[24] Yuan-Ting Hu, Jia-Bin Huang, and Alexander Schwing. MaskRNN: Instance level video object segmentation. In *NIPS*, 2017.

[25] Yuan-Ting Hu and Yen-Yu Lin. Progressive feature matching with alternate descriptor selection and correspondence enrichment. In *CVPR*, 2016.

[26] Yan Huang, Wei Wang, and Liang Wang. Instance-aware image and sentence matching with selective multimodal LSTM. In *CVPR*, 2017.

[27] Koteswar Rao Jerripothula, Jianfei Cai, Jiangbo Lu, and Junsong Yuan. Object co-skeletonization with co-segmentation. In *CVPR*, 2017.

[28] Koteswar Rao Jerripothula, Jianfei Cai, and Junsong Yuan. Image co-segmentation via saliency co-fusion. *TMM*, 2016.

[29] Diederik Kingma and Jimmy Ba. ADAM: A method for stochastic optimization. In *ICLR*, 2014.

[30] Shu Kong and Charless Fowlkes. Recurrent pixel embedding for instance grouping. In *CVPR*, 2018.

[31] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. In *NIPS*, 2018.

[32] Guanbin Li, Yuan Xie, Liang Lin, and Yizhou Yu. Instance-level salient object segmentation. In *CVPR*, 2017.

[33] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *ACCV*, 2018.

[34] Xin Li, Fan Yang, Hong Cheng, Wei Liu, and Dinggang Shen. Contour knowledge transfer for salient object detection. In *ECCV*, 2018.

[35] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017.

[36] Xiaodan Liang, Yunchao Wei, Xiaohui Shen, Jianchao Yang, Liang Lin, and Shuicheng Yan. Proposal-free network for instance-level object segmentation. *TPAMI*, 2018.

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[38] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018.

[39] Yiding Liu, Siyu Yang, Bin Li, Wengang Zhou, Jizheng Xu, Houqiang Li, and Yan Lu. Affinity derivation and graph merge for instance segmentation. In *ECCV*, 2018.

[40] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional models for semantic segmentation. In *CVPR*, 2015.

[41] Zhiming Luo, Akshaya Mishra, Andrew Achkar, Justin Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. In *CVPR*, 2017.

[42] Armin Mustafa and Adrian Hilton. Semantically coherent co-segmentation and reconstruction of dynamic scenes. In *CVPR*, 2017.

[43] David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Semi-convolutional operators for instance segmentation. In *ECCV*, 2018.

[44] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T. Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *TPAMI*, 2017.

[45] Rong Quan, Junwei Han, Dingwen Zhang, and Feiping Nie. Object co-segmentation via graph optimized-flexible manifold ranking. In *CVPR*, 2016.

[46] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013.

[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Fei-Fei Li. ImageNet large scale visual recognition challenge. *IJCV*, 2015.

[48] Tong Shen, Guosheng Lin, Lingqiao Liu, Chunhua Shen, and Ian Reid. Weakly supervised semantic segmentation based on co-segmentation. In *BMVC*, 2017.

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[50] Charles Sutton and Andrew McCallum. Piecewise training for structured prediction. *ML*, 2009.

[51] Kevin Tang, Armand Joulin, Li-Jia Li, and Fei-Fei Li. Co-localization in real-world images. In *CVPR*, 2014.

[52] Tatsunori Taniai, Sudipta N. Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *CVPR*, 2016.

[53] Zhiqiang Tao, Hongfu Liu, Huazhu Fu, and Yun Fu. Image cosegmentation via saliency-guided constrained clustering with cosine similarity. In *AAAI*, 2017.

[54] Chung-Chi Tsai, Weizhi Li, Kuang-Jui Hsu, Xiaoning Qian, and Yen-Yu Lin. Image co-saliency detection and co-segmentation via progressive joint optimization. *TIP*, 2018.

[55] Andrea Vedaldi and Karel Lenc. MatConvNet – Convolutional neural networks for MATLAB. In *ACMMM*, 2015.

[56] Chuan Ping Wang, Hua Zhang, Liang Yang, Xiaochun Cao, and Hongkai Xiong. Multiple semantic matching on augmented n-partite graph for object co-segmentation. *TIP*, 2017.

[57] Wenguan Wang, Jianbing Shen, Hanqiu Sun, and Ling Shao. Video co-saliency guided co-segmentation. *TCSVT*, 2018.

[58] Xiu-Shen Wei, Chen-Lin Zhang, Yao Li, Chen-Wei Xie, Jianxin Wu, Chunhua Shen, and Zhi-Hua Zhou. Deep descriptor transforming for image co-localization. In *IJCAI*, 2017.

[59] Xiu-Shen Wei, Chen-Lin Zhang, Jianxin Wu, Chunhua Shen, and Zhi-Hua Zhou. Unsupervised object discovery and co-localization by deep descriptor transforming. *PR*, 2019.

[60] Tsun-Yi Yang, Jo-Han Hsu, Yen-Yu Lin, and Yung-Yu Chuang. DeepCD: Learning deep complementary descriptors for patch representations. In *ICCV*, 2017.

[61] Tsun-Yi Yang, Yen-Yu Lin, and Yung-Yu Chuang. Accumulated stability voting: A robust descriptor from descriptors of multiple scales. In *CVPR*, 2016.

[62] Zehuan Yuan, Tong Lu, and Yirui Wu. Deep-dense conditional random fields for object co-segmentation. In *IJCAI*, 2017.

[63] Dingwen Zhang, Junwei Han, and Yu Zhang. Supervision by fusion: Towards unsupervised learning of deep salient object detector. In *ICCV*, 2017.

[64] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *CVPR*, 2016.

[65] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *CVPR*, 2018.