The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Co-Saliency Detection within a Single Image

Hongkai Yu,[2] **Kang Zheng,**[2] **Jianwu Fang,**[3, 4] **Hao Guo,**[2] **Wei Feng,**[1] **Song Wang**[1, 2,*]
[1] School of Computer Science and Technology, Tianjin University, Tianjin, China
[2] Department of Computer Science and Engineering, University of South Carolina, Columbia, SC
[3] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China
[4] School of Electronic and Control Engineering, Chang'an University, Xi'an, China

Abstract

Recently, saliency detection in a single image and co-saliency detection in multiple images have drawn extensive research interest in the vision community. In this paper, we investigate a new problem of co-saliency detection within a single image, i.e., detecting **within-image co-saliency**. By identifying common saliency within an image, e.g., highlighting multiple occurrences of an object class with similar appearance, this work can benefit many important applications, such as the detection of objects of interest, more robust object recognition, reduction of information redundancy, and animation synthesis. We propose a new bottom-up method to address this problem. Specifically, a large number of object proposals are first detected from the image. Then we develop an optimization algorithm to derive a set of **proposal groups**, each of which contains multiple proposals showing good common saliency in the original image. For each proposal group, we calculate a co-saliency map and then use a low-rank based algorithm to fuse the maps calculated from all the proposal groups for the final co-saliency map in the image. In the experiment, we collect a new dataset of 364 color images with within-image co-saliency. Experiment results show that the proposed method can better detect the within-image co-saliency than existing algorithms.

Introduction

Research on image-based saliency detection has drawn extensive interest in the vision community in the past decade. It started with saliency detection in a single image, i.e., *within-image saliency* detection, which aims at highlighting the visually standing-out regions/objects/structures from the surrounding background (Cheng et al. 2015; Mahadevan and Vasconcelos 2009; Zhao, Ouyang, and Wang 2013; Huang, Feng, and Sun 2015), as illustrated in Fig. 1(a). More recently, co-saliency detection in multiple images, e.g., *cross-image co-saliency* detection (Fu, Cao, and Tu 2013; Zhang et al. 2015a; Huang, Feng, and Sun 2017), has been attracting much attention with many successful applications (Meng et al. 2012; Yu, Xian, and Qi 2014; Joulin, Tang, and Fei-Fei 2014; Tang et al. 2014). As illustrated in Fig. 1(b), cross-image co-saliency detection aims to detect the common saliency, e.g., red-clothed soccer players, that are present in all three images. In this paper, we investigate a new problem of detecting co-saliency within a single image, i.e., *within-image co-saliency* detection, which aims to highlight the common saliency within an image. An example is shown in Fig. 1(c), where the two red-clothed players show good within-image co-saliency, but the white-clothed player does not because only one white-clothed player is present in the image.

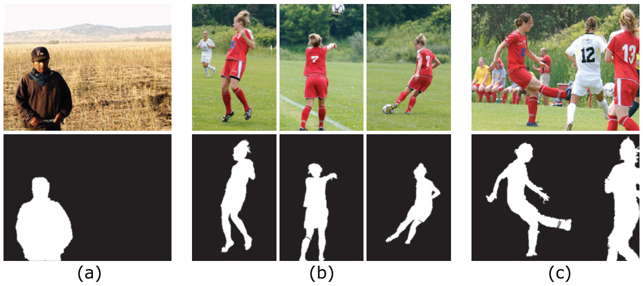

Figure 1: Illustrations of different saliency detection problems. (a) Within-image saliency detection. (b) Cross-image co-saliency detection, where co-saliency is detected across three images. (c) The proposed within-image co-saliency detection. First row: images. Second row: ground-truth saliency/co-saliency maps.

Within-image co-saliency detection can benefit many important applications in computer vision. For example, it can be used to help detect multiple instances of an object class in an image and help estimate the number of instances of the same object class (He and Gould 2014; Lin et al. 2014). By combining the features of the identified co-salient objects, we may obtain more accurate and more reliable object recognition and detection in the image. Within-image co-saliency detection can also help identify and reduce information redundancy within an image. For example, recent mobile plant-recognition systems (Kumar et al. 2012) usually require the user to take an plant image using his/her smart phone camera and then send the plant image to a remote server for large-scale plant-species classification. The proposed within-image co-saliency detection can identify multiple instances of the same plant part, e.g., leaf, and then crop out only one of them before sending it

[*] Corresponding author: songwang@cec.sc.edu

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

to the remote server. This may substantially reduce the data size and communication load. As in (Xu et al. 2008), repeated instances of an object class can be used to synthesize realistic animation from a still picture, which could be helped by within-image co-saliency detection.

However, within-image co-saliency detection is a nontrivial problem. As far as we know, there is **no existing work** that explicitly discusses and tackles this problem. On one hand, this problem cannot be well addressed by directly applying the existing methods on saliency detection. By using a within-image saliency detection method, we may also highlight objects that show good saliency but not co-saliency, e.g., the white-clothed player in Fig. 1(c). On the other hand, cross-image co-saliency methods are also not applicable here because we only have one input image. One naive solution might be making multiple copies of the input image and then applying a cross-image co-saliency detection method. However, this solution still does not work, because it will highlight all the salient objects in the image. For example, if we make two copies of the image shown in Fig. 1(c) and then apply a cross-image co-saliency detection algorithm, all three players including the white-clothed player will be highlighted. The white-clothed player may even be emphasized more because she is located at the center of both image copies.

In this paper, we propose a new bottom-up method for detecting within-image co-saliency. Given an image, we first detect a large number of object proposals (Zitnick and Dollár 2014). We then develop an optimization algorithm to derive a set of *proposal groups*, each of which consists of multiple selected proposals showing good common saliency in the original image. Three factors are considered in measuring the common saliency for a group of proposals: 1) the saliency of each proposal in the original image, 2) similar image appearance of the proposals, and 3) low spatial overlap of the proposals. For each derived proposal group, a co-saliency map is computed by a clustering-based algorithm. We then fuse the co-saliency maps computed from different proposal groups into a final co-saliency map using a low-rank based algorithm. Since most existing image datasets used for saliency detection do not consider the within-image co-saliency, we collect a new dataset of 364 images for performance evaluation. In the experiment, we test the proposed method and other comparison methods on the new dataset and quantitatively evaluate their performance based on the annotated ground truth.

The remainder of the paper is organized as follows. Section 2 overviews the related work. Section 3 introduces the proposed method on within-image co-saliency detection. Section 4 reports the image dataset and experimental results, followed by a brief conclusion in Section 5.

Related Work

As mentioned above, most previous work on image-based saliency detection is focused on two problems: saliency detection in a single image, i.e., within-image saliency detection, and co-saliency detection in multiple images, i.e., cross-image co-saliency detection.

Many within-image saliency detection models and methods have been developed in the past decades. Most traditional methods identify salient regions in an image based on visual contrasts (Cheng et al. 2015). Many hand-crafted rules, such as center bias (Fu, Cao, and Tu 2013), frequency (Achanta et al. 2009), and spectral residuals (Hou and Zhang 2007) have been incorporated to improve the saliency detection performance. Graph-based segmentation algorithms (Rother, Kolmogorov, and Blake 2004; Yu et al. 2015) could be applied to refine the resulting saliency maps (Cheng et al. 2015). In (Chang et al. 2011; Li et al. 2014b; Wang et al. 2013; Shen and Wu 2012), high-level knowledges such as objectness, fixation predictions, object boundary, and low rank consistency are integrated to achieve within-image saliency detection, besides the use of low-level features like color, texture and SIFT features. Recently, deep learning techniques have also been used for detecting saliency in an image by automatically learning the features. In particular, it has been shown that multi-scale deep learning (Li and Yu 2015) and deep contrast learning (Li and Yu 2016) using patch-level convolutional neural networks (CNN) or pixel-level fully convolutional networks (FCN) (Long, Shelhamer, and Darrell 2015) and recurrent fully convolutional networks (RFCN) (Wang et al. 2016) can detect the within-image saliency more accurately than many of the above-listed traditional methods.

Cross-image co-saliency detection has also been studied by many researchers recently. In (Fu, Cao, and Tu 2013; Ge et al. 2016), each pixel is ranked by using manually designed co-saliency cues such as inter-image saliency cue, intra-image saliency cue, and repeatedness cue. In (Cao et al. 2014; Li et al. 2014a; Tan et al. 2013), co-saliency maps produced by different methods are fused by further exploring the inter-image correspondence. Recently, machine learning based methods like weakly supervised learning (Cheng et al. 2014), multiple instance learning (Zhang et al. 2015b), and deep learning (Zhang et al. 2016; 2015a) are also used for cross-image co-saliency detection. Other problems related to cross-image co-saliency detection are co-localization (Joulin, Tang, and Fei-Fei 2014; Tang et al. 2014) and co-segmentation (Meng et al. 2012; Yu, Xian, and Qi 2014), which aim to localize or segment common objects that are present in multiple input images. However, all these within-image saliency detection and cross-image co-saliency detection methods cannot address the problem of within-image co-saliency detection, on which this paper is focused, because they could not de-emphasize salient objects that do not show within-image co-saliency, e.g., the white-clothed player in Fig. 1(c).

Other work related to our problem is the supervised object detection, a fundamental problem in computer vision. In (Kanan et al. 2009), top-down approaches considering object detection are developed for detecting within-image saliency – objects detected in the image are emphasized in the saliency map. Ideally, we may extend it to within-image co-saliency detection: run an object detector (Ren et al. 2015; Redmon et al. 2015) on the given image and then match the detected objects. If two or more detected objects show high-level of similarity and belong to the same object

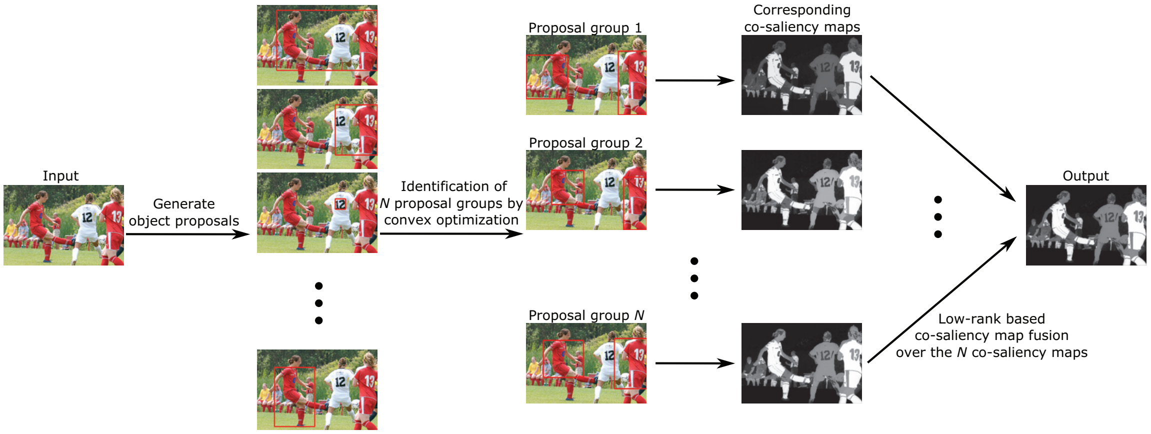

Figure 2: Diagram of the proposed method for detecting within-image co-saliency.

class, we highlight them in the resulting co-saliency map. If a detected object does not match to any other detected object in the image, we de-emphasize it in the resulting co-saliency map. However, object detector can only detect known object classes (Everingham et al. 2012) that are pre-trained using supervised learning, not to mention that highly-accurate large-scale object detection itself is still a challenging research problem. Just like most previous work on saliency detection, in this paper we detect within-image co-saliency without assuming any specific object class and recognizing any objects in the image.

Proposed Method

The basic idea of the proposed method is to first generate many object proposals (in the form of rectangular bounding boxes) in the image, and then compute co-saliency by deriving proposal groups with good common saliency in the image. The diagram of the proposed method is illustrated in Fig. 2. For object proposals, as mentioned above, we do not consider any prior information on the object classes and they are detected only based on general objectness. Considering the possibility that many detected object proposals do not well cover a real object, as shown in the second column of Fig. 2, we identify different proposal groups where each group of proposals show good common saliency. We compute such common saliency for each proposal group in the form of a co-saliency map in the original image and finally fuse the co-saliency maps computed from different proposal groups for the desired within-image co-saliency.

In this paper, we use the classical bottom-up EdgeBox method (Zitnick and Dollár 2014) to generate object proposals in the image. More specifically, we first use EdgeBox to generate a large pool of proposals with different objectness scores. After pruning overly small proposals (with size $< 1\%$ of the image size), we select M object proposals with the highest objectness scores from the pool and denote them as $P_i, i = 1, 2, \ldots, M$. Based on these M detected proposals, we elaborate on the other three main components of the proposed methods, i.e., identification of proposal groups, computing co-saliency map for a proposal group, and fusion of multiple co-saliency maps, in this section.

Identification of Proposal Groups

Given M object proposals $P_i, i = 1, 2, \ldots, M$ in the image, we identify N different proposal groups, each of which consists of a subset of proposals with good common saliency. In this section, we identify these N proposal groups iteratively: After identifying the first proposal group with highest common saliency, we exclude the identified proposals and apply the same algorithm to identify another proposal group. This process is repeated N times to obtain N proposal groups. For simplicity, we fix the number of proposals in each group to be $K > 1$, which is a pre-set constant. In this paper, we consider three main factors in measuring the common saliency of K proposals in a group: 1) saliency of each of these K proposals, 2) high appearance similarity of these K proposals, and 3) low spatial overlap of these K proposals.

A proposal group can be denoted by a vector $\mathbf{z} = (z_1, z_2, \ldots, z_M)^T$, where $z_i \in \{0, 1\}$, with 1 indicating that proposal i is included in the group and 0 otherwise. First, we can use any within-image saliency detection algorithm (Li and Yu 2016; Zhao et al. 2015; Cheng et al. 2015; Fu, Cao, and Tu 2013) to compute an initial saliency map $h(X)$, where X represents all the pixels in the input image and $h(\mathbf{x})$ is the saliency value at pixel $\mathbf{x} \in X$. The saliency of each proposal P_i can then be estimated as $h_i = \frac{1}{|P_i|}\sum_{\mathbf{x}\in P_i} h(\mathbf{x})$. The saliency of all M proposals can be summarized into a column vector $\mathbf{h} = (h_1, h_2, \ldots, h_M)^T$. Following (Tang et al. 2014; Rubinstein et al. 2013), we define a *saliency energy term* to reflect the total saliency of a proposal group $\mathbf{z}$ in the original image by

$$E_1(\mathbf{z}) = -\mathbf{z}^T \log(\mathbf{h}). \tag{1}$$

The smaller this energy term, the larger the saliency of this proposal group in the original image.

To consider the high appearance similarity and low spatial overlap of the proposals in a group $\mathbf{z}$, we first define a pairwise similarity between two proposals, say P_i and P_j, as

$$w_{ij} = \frac{1}{d_{ij}^2 + o_{ij}^2}, \tag{2}$$

where d_{ij} is the L_2 distance between the appearance features of P_i and P_j, and o_{ij} reflects the spatial overlap of P_i

and P_j. Specifically, we compute the appearance feature of a proposal by using the normalized RGB color histogram (256×3 bins) of all the pixels in the proposal. We define o_{ij} as $\frac{|P_i \cap P_j|}{\min(|P_i|,|P_j|)}$.

Based on the pairwise similarity w_{ij}, we construct a similarity matrix $\mathbf{W} = (w_{ij})_{M\times M}$. $\mathbf{W}$ is a symmetric matrix and we set all diagonal element w_{ii} to be 0. The normalized Laplacian matrix can then be computed by $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{I}$ is an $M \times M$ identity matrix, $\mathbf{D}$ is the degree matrix, i.e., a diagonal matrix, whose i-th diagonal element takes the value of $\sum_{j=1}^{M} w_{ij}$. Using $\mathbf{L}$, we can define a *similarity energy term* for a proposal group $\mathbf{z}$ that encourages high appearance similarity and low spatial overlap as

$$E_2(\mathbf{z}) = \mathbf{z}^T\mathbf{L}\mathbf{z}. \tag{3}$$

Combining the two energy terms shown in Eqs. (1) and (3), we define the following constrained optimization problem for identifying a proposal group:

$$\begin{aligned} \min_{\mathbf{z}} \quad & \mathbf{z}^T\mathbf{L}\mathbf{z} + \lambda(-\mathbf{z}^T \log(\mathbf{h})) \\ \text{s.t.} \quad & z_i \in \{0,1\}, i = 1,2,\ldots,M \\ & \sum_{i=1}^{M} z_i = K, \end{aligned} \tag{4}$$

where $\lambda > 0$ is a balance factor for the two energy terms, and the last constraint indicates that we seek a group of K proposals with high common saliency. Since the optimization variables in $\mathbf{z}$ are binary, this is not a convex optimization problem. To make it convex, we relax the first constraint in Eq. (4) to

$$0 \le z_i \le 1, i = 1,2,\ldots,M.$$

This way, the optimization problem becomes a standard quadratic programming under linear constraints, since the saliency energy term in Eq. (1) is linear and the similarity energy term in Eq. (3) is quadratic. We can solve this problem efficiently using the primal-dual interior-point method by the CVX convex optimization toolbox (Grant, Boyd, and Ye 2008). After we get the optimal solution $\mathbf{z}$, we simply select the K proposals with the highest values in $\mathbf{z}$ to form a proposal group. As mentioned above, we iterate this optimization algorithm N times to construct N proposal groups. Figure 3 shows the proposal groups identified from a sample image.

Co-saliency Detection in a Proposal Group

Without loss of generality, let $\mathcal{P} = \{P_1, P_2, \ldots, P_K\}$ be an identified proposal group. In this section, we detect the common saliency in this proposal group and summarize this common saliency into a co-saliency map in the original image. Starting from the initial saliency map $h(X)$, we first threshold this saliency map by a threshold (0.2 in our experiments) to obtain salient region X_T. Inspired by previous work on cross-image co-saliency detection (Fu, Cao, and Tu 2013), we apply the Kmeans algorithm to cluster all the pixels X in the input image into Z clusters $C_1, C_2, \ldots, C_Z$ based on these pixels' RGB color values. If a cluster shows good spatial overlap with the considered proposal group $\mathcal{P}$, the pixels in this cluster tends to show higher within-image co-saliency in the original image.

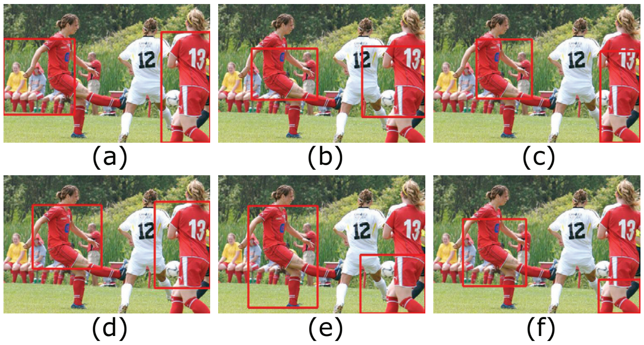

Figure 3: Six proposal groups identified from a sample image. (a-f) Proposal groups identified from iteration 1 to iteration 6, respectively. Here we set $K = 2$.

More specifically, for each pixel $\mathbf{x} \in C_z$, we define its unnormalized common-saliency map value triggered by proposal group $\mathcal{P}$, which consists of proposals $P_1, P_2, \ldots, P_K$, as

$$\hbar'_{\mathcal{P}}(\mathbf{x}) = \frac{|(\cup_{k=1}^{K} P_k) \cap X_T \cap C_z|}{|(\cup_{k=1}^{K} P_k) \cap X_T|}, \tag{5}$$

where the denominator is the number of salient pixels that are located in the proposal group $\mathcal{P}$ and the numerator is the number of salient pixels in cluster C_z that are located in the proposal group $\mathcal{P}$. We then normalize the map $\hbar'_{\mathcal{P}}(X)$ to a standard Gaussian distribution and denote the normalized common-saliency map triggered by proposal group $\mathcal{P}$ as $\hat{\hbar}_{\mathcal{P}}(X)$. To reduce the effect of clustering errors, we further combine the initial saliency map $h(X)$ and the common-saliency map $\hat{\hbar}_{\mathcal{P}}(X)$ by pixel-wise multiplication to construct a co-saliency map $\hbar_{\mathcal{P}}(X)$ as

$$\hbar_{\mathcal{P}}(\mathbf{x}) = \hat{\hbar}_{\mathcal{P}}(\mathbf{x}) \cdot h(\mathbf{x}), \mathbf{x} \in X,$$

followed by thresholding (0.2 in our experiments), holes filling and average filtering. In Fig. 4, we use a sample image to illustrate the process of this co-saliency detection.

Co-Saliency Map Fusion

Based on N identified proposal groups, we can use each of them as the trigger to compute a co-saliency map. In this way, we obtain N co-saliency maps, which we denote as $\{\hbar_1(X), \hbar_2(X), \ldots, \hbar_N(X)\}$. In this section, we study how to fuse these N co-saliency maps into a unified co-saliency map.

After simple thresholding, we find that the co-salient regions in the N co-saliency maps display color-feature consistency when mapped back to the original color image, where the color-feature consistency could be thought as a low rank constraint. Meanwhile other salient objects but not showing within-image co-saliency and the background are treated as sparse noises. In this paper, we adapt the method in (Cao et al. 2014) for fusing the N co-saliency maps. First,

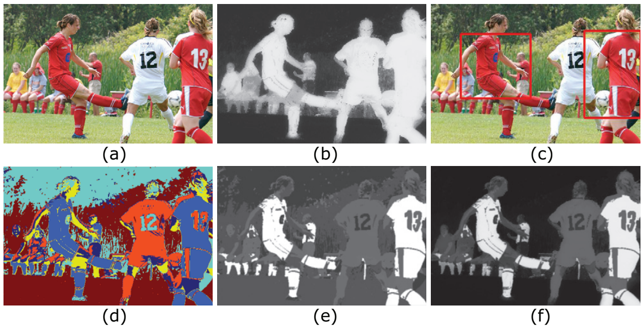

Figure 4: An example of co-saliency detection triggered by a proposal group. (a) Original image, (b) initial saliency map $h(X)$ (Li and Yu 2016), (c) a proposal group $\mathcal{P}$ with two proposals, (d) Kmeans clustering results, where each color indicates a cluster, in total six clusters, (e) normalized common-saliency map $\hat{\hbar}_{\mathcal{P}}(X)$, and (f) co-saliency map $\hbar_{\mathcal{P}}(X)$.

for each co-saliency map, say $\hbar_i(X)$, we apply a simple thresholding as that in (Cao et al. 2014) to get pixels with high co-saliency. We then compute the RGB color histogram (1,000 bins) of all the identified pixels with high co-saliency by mapping back to the original color image and denote this histogram as a column vector $\mathbf{f}_i$. Combining the N histograms computed from N co-saliency maps, respectively, we obtain a feature matrix $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N)$. We then seek to recover a low-rank matrix $\mathbf{R}$ from $\mathbf{F}$, i.e.,

$$\begin{aligned}(\mathbf{R}^*, \mathbf{E}^*) &= \arg\min_{\mathbf{R},\mathbf{E}} \; (rank(\mathbf{R}) + \beta \left\|\mathbf{E}\right\|_0) \\ \text{s.t.} \;\; \mathbf{F} &= \mathbf{R} + \mathbf{E},\end{aligned} \tag{6}$$

where $\beta > 0$ is a balance factor between the rank of $\mathbf{R}$ and the L_0 norm of the sparse noise $\mathbf{E}$. By using nuclear norm to approximate $rank(\mathbf{R})$ and L_1 norm to approximate $\left\|\mathbf{E}\right\|_0$, this low-rank matrix recovery problem becomes convex and can be solved by robust PCA (Wright et al. 2009).

Following (Cao et al. 2014), the final fused co-saliency map can be written as a weighted linear combination of the N co-saliency maps, i.e.,

$$\hbar(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \cdot \hbar_i(\mathbf{x}), \mathbf{x} \in X, \tag{7}$$

where the weight α_i can be calculated by

$$\alpha_i = \frac{\exp(-\left\|\mathbf{e}_i^*\right\|_2)}{\sum_{i=1}^{N} \exp(-\left\|\mathbf{e}_i^*\right\|_2)}. \tag{8}$$

In Eq. (8), $\mathbf{e}_i^*$ is the i-th column of $\mathbf{E}^*$ resulting from Eq. (6). Less sparse noise $\mathbf{e}_i^*$ indicates that the i-th co-saliency map $\hbar_i(X)$ is more credible and it should be weighted more in computing the final co-saliency map $\hbar(X)$. The entire proposed method for detecting co-saliency within a single image is summarized in Algorithm 1.

Algorithm 1 Co-saliency detection within a single image.

```
Input: A color image
Use EdgeBox (Zitnick and Dollár 2014) to generate
$M$ object proposals.
Compute the initial saliency map $h(X)$.
Generate $Z$ clusters by Kmeans algorithm.
FOR $i = 1 : N$
    Identify $i$-th proposal group by solving Eq. (4).
    Compute co-saliency map $\hbar_i(X)$.
    Exclude proposals in the $i$-th proposal group.
END FOR
Fuse the $N$ co-saliency maps $\hbar_i(X), i = 1, 2, \ldots, N$
for the final co-saliency map $\hbar(X)$.
```

Experiments

Existing publicized image datasets for evaluating saliency detection such as MSRA (Liu et al. 2011), PASCAL-S (Li et al. 2014b), HKU-IS (Li and Yu 2015), iCoseg (Batra et al. 2010) are mainly collected for testing within-image saliency detection or cross-image co-saliency detection methods. In most cases, each image only contains one salient object, which is annotated as the ground truth. In this paper, we have a different goal of detecting within-image co-saliency, which is not shown in most images in the publicized datasets. Therefore, we collect a new image dataset, consisting of 364 color images. Each image shows certain level of within-image co-saliency, e.g., the presence of multiple instances of the same object class with very similar appearance. In this new dataset, 65 *challenging* images also contain salient objects without showing any within-image co-saliency, while other 299 *easy* images do not. Sample *easy* and *challenging* images with corresponding ground truths in the collected dataset are shown in Fig. 5. In this new dataset, about 100 images are selected from the iCoseg (Batra et al. 2010), MSRA (Liu et al. 2011), HKU-IS (Li and Yu 2015) datasets and the remaining images are collected from the Internet. Co-salient objects within each image are manually labeled as the ground truth (a binary mask) for performance evaluation. To avoid unreasonable labeling, the ground truths are double checked by five different researchers in computer vision area. The image size ranges from 150 × 150 to 808 × 1078 pixels.

In our experiment, we generate $M = 100$ object proposals. The number of proposal groups is set to $N = 10$. The number of proposals in each group is set to $K = 2$. We set the balance factors $\lambda = 0.01$ in Eq. (4) and $\beta = 0.05$ in Eq. (6). The number of clusters is set to $Z = 6$ in the Kmeans algorithm. The initial within-image saliency map $h(X)$ is computed using the algorithm developed in (Li and Yu 2016). Seven state-of-the-art within-image saliency detection methods are chosen as the comparison methods: CWS (Fu, Cao, and Tu 2013), LRK (Shen and Wu 2012), SR (Hou and Zhang 2007), FT (Achanta et al. 2009), RC (Cheng et al. 2015), DCL (Li and Yu 2016), and RFCN (Wang et al. 2016). The first five are traditional feature-based methods and the last two are based on deep learning.

As in many previous works (Achanta et al. 2009; Fu, Cao, and Tu 2013; Cheng et al. 2015; Li and Yu 2016), we evaluate the performance using precision-recall (PR) curve, max-

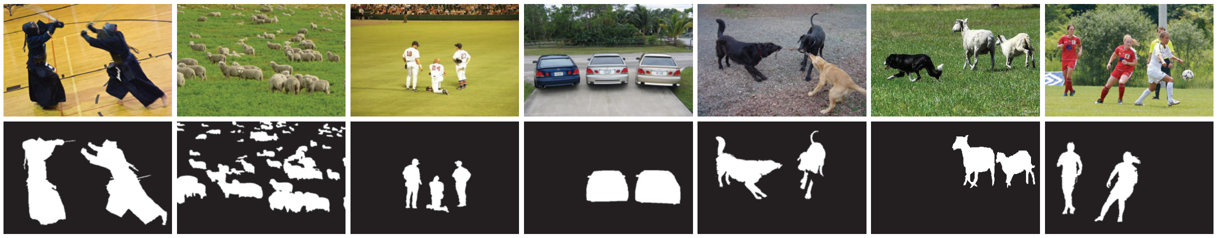

Figure 5: Sample images and their corresponding within-image co-saliency ground truth in the new dataset (first three: *easy*, last four: *challenging*).

imum F-measure (maxF), MAE error and also report the average precision, recall and F-measure using an adaptive threshold. The resulting saliency map can be converted to a binary mask with a threshold, and the precision and recall are computed by comparing the binary mask and the binary ground truth. Varying the threshold continuously in the range of $[0, 1]$ leads to a PR curve, which is averaged over all the images in the dataset in this paper. As in (Li and Yu 2016), we can calculate the maximum F-measure (maxF) from the PR curve and the MAE error as the average absolute per-pixel difference between the resulting saliency map and the ground truth. As in (Achanta et al. 2009; Li and Yu 2016), we also use an adaptive threshold, i.e., twice the mean value of the saliency map, to convert the saliency map into a binary mask. Comparing the binary mask with the binary ground truth, we can compute the precision and recall, based on which we can compute F-measure as $F_\gamma = \frac{(1+\gamma^2)\times Precision\times Recall}{\gamma^2\times Pecision+Recall}$, where γ^2 is set to 0.3 as defined in (Achanta et al. 2009; Fu, Cao, and Tu 2013; Li and Yu 2016).

Results

Figure 6 shows the PR curves of the proposed method and seven comparison methods that were developed for within-image saliency detection. We can see that, in general, the proposed method performs better than all these seven comparison methods in detecting the within-image co-saliency, in terms of the PR curve. The main reason lies on that these seven comparison methods detect saliency in the image, including the salient object without showing any within-image co-saliency. We can also see that the two deep learning based methods (DCL (Li and Yu 2016), RFCN (Wang et al. 2016)) can detect better within-image co-saliency than the five traditional saliency detection methods. Among the five traditional methods, CWS (Fu, Cao, and Tu 2013) and RC (Cheng et al. 2015) show relatively better performance in detecting within-image co-saliency. Table 1 compares the maxF and MAE error of the proposed method against these seven comparison methods. From this table, we can also see that the proposed method achieves the best performance in detecting within-image co-saliency.

The average precision, recall and F-measure using adaptive thresholds (Achanta et al. 2009; Li and Yu 2016) are shown as a bar chart in Fig. 7. We can see that, using adaptive thresholds, the proposed method achieves the best average precision, recall and F-measure against the seven comparison methods in detecting within-image co-saliency. Specifically, the average precision, recall and F-measure using adaptive thresholds are [**0.882**, **0.907**, **0.881**] when using the proposed method, while the second best is achieved similarly by DCL (Li and Yu 2016) ($[0.856, 0.904, 0.859]$) and RFCN (Wang et al. 2016) ($[0.872, 0.856, 0.859]$). Among the seven comparison methods, DCL, RFCN, RC and CWS show relatively better performance than the others. Using adaptive thresholds, F-measure of the proposed method on the 299 *easy* images is 0.905 and the second best F-measure is 0.897 by RFCN. With adaptive thresholds, F-measure of the proposed method on the 65 *challenging* images is 0.771 and the second best F-measure is 0.688 by RFCN.

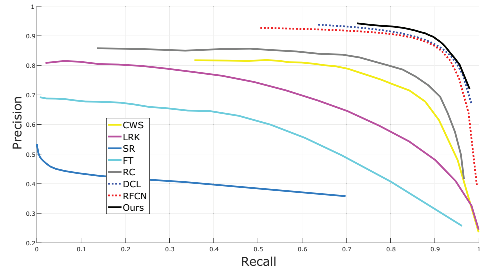

Figure 6: PR curves of the proposed method ('Ours') and the seven saliency detection methods, averaged over all the 364 images in the collected dataset.

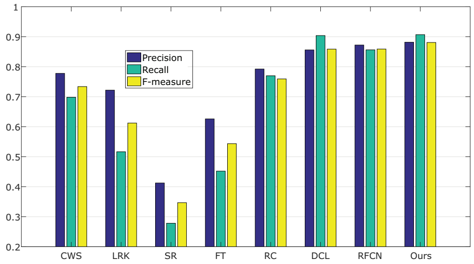

Figure 7: Average precision, recall and F-measure using adaptive thresholds. Our F-measure: 88.1%; second best F-measure: 85.9%.

Figure 8 shows sample results of within-image co-saliency detection from the proposed method and the comparison methods including seven within-image saliency detection methods. We can see that the proposed method is capable of highlighting the regions that show within-image co-saliency and de-emphasizing the salient regions that do not show within-image co-saliency. However, the comparison methods might highlight all the salient regions or ig-

Metric	CWS	LRK	SR	FT	RC	DCL	RFCN	Ours
maxF (%)	76.7	67.4	40.3	58.2	80.2	88.8	88.3	**90.3**
MAE error	0.165	0.241	0.246	0.244	0.142	0.059	0.083	**0.050**

Table 1: The maximum F-measure (maxF) and MAE error of the proposed method ('Ours') and the seven within-image saliency detection methods. Larger maxF and smaller MAE error indicate better performance.

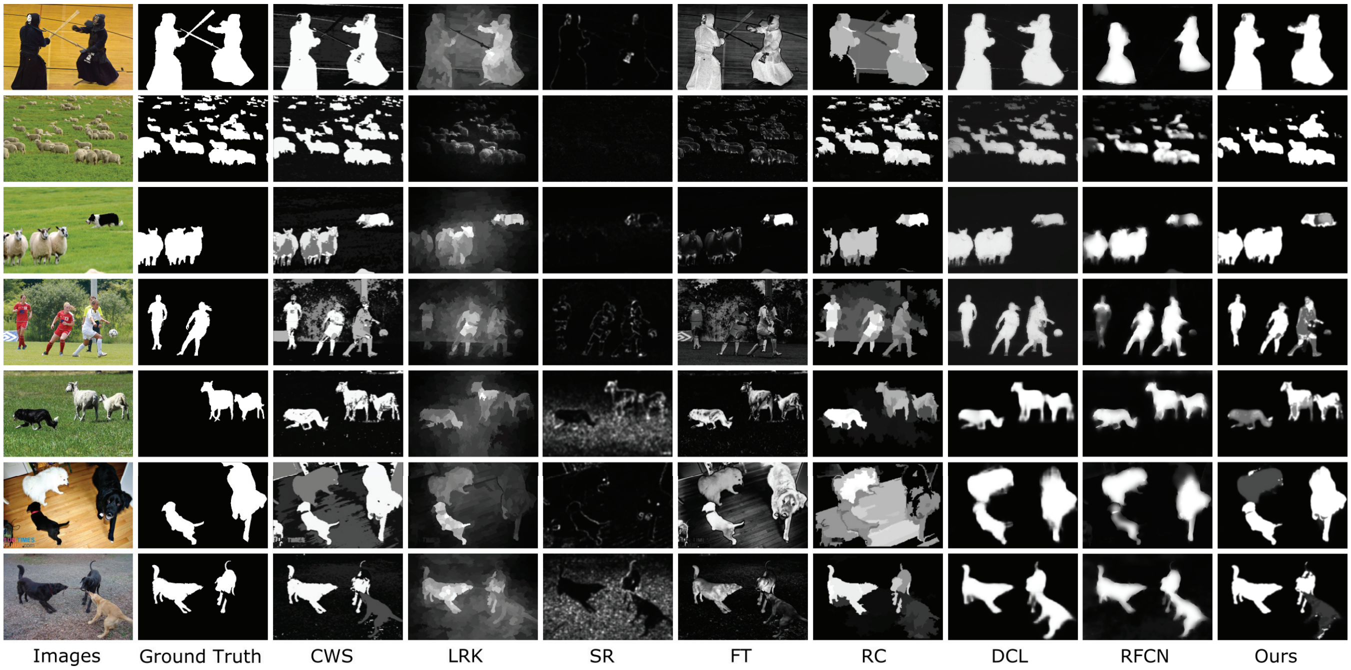

Figure 8: Within-image co-saliency detection results on seven sample images.

nore to emphasize the regions that show within-image co-saliency.

Conclusions

In this paper, we raised a new problem of detecting co-saliency in a single image, i.e., detecting within-image co-saliency. We developed a new bottom-up method to solve this problem. This method starts with detecting a large number of object proposals in the image, without using any prior information on the object classes. We then developed an optimization model to identify a set of proposal groups, each of which consists of multiple proposals with good common saliency in the original image. Co-saliency is then detected in each proposal group and fused for the final within-image co-saliency map. We collected a new set of 364 images with good within-image co-saliency, and then used them to test the proposed method. Experimental results showed that the proposed method outperforms the recent state-of-the-art saliency detection methods in detecting within-image co-saliency.

Acknowledgment This work was supported in part by NSF-1658987 and NSFC-61671325, 61672376, 61603057.

References

Achanta, R.; Hemami, S.; Estrada, F.; and Susstrunk, S. 2009. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1597–1604.

Batra, D.; Kowdle, A.; Parikh, D.; Luo, J.; and Chen, T. 2010. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3169–3176.

Cao, X.; Tao, Z.; Zhang, B.; Fu, H.; and Feng, W. 2014. Self-adaptively weighted co-saliency detection via rank constraint. *IEEE Transactions on Image Processing* 23(9):4175–4186.

Chang, K.-Y.; Liu, T.-L.; Chen, H.-T.; and Lai, S.-H. 2011. Fusing generic objectness and visual saliency for salient object detection. In *IEEE International Conference on Computer Vision*, 914–921.

Cheng, M.-M.; Mitra, N. J.; Huang, X.; and Hu, S.-M. 2014. Salientshape: Group saliency in image collections. *The Visual Computer* 30(4):443–453.

Cheng, M.-M.; Mitra, N.; Huang, X.; Torr, P.; and Hu, S.-M. 2015. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3):569–582.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2012. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Fu, H.; Cao, X.; and Tu, Z. 2013. Cluster-based co-saliency detection. *IEEE Transactions on Image Processing* 22(10):3766–3778.

Ge, C.; Fu, K.; Liu, F.; Bai, L.; and Yang, J. 2016. Co-saliency detection via inter and intra saliency propagation. *Signal Processing: Image Communication* 44:69–83.

Grant, M.; Boyd, S.; and Ye, Y. 2008. Cvx: Matlab software for disciplined convex programming.

He, X., and Gould, S. 2014. An exemplar-based crf for multi-instance object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 296–303.

Hou, X., and Zhang, L. 2007. Saliency detection: A spectral residual approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

Huang, R.; Feng, W.; and Sun, J. 2015. Saliency and co-saliency detection by low-rank multiscale fusion. In *IEEE International Conference on Multimedia and Expo*, 1–6.

Huang, R.; Feng, W.; and Sun, J. 2017. Color feature reinforcement for cosaliency detection without single saliency residuals. *IEEE Signal Processing Letters* 24(5):569–573.

Joulin, A.; Tang, K.; and Fei-Fei, L. 2014. Efficient image and video co-localization with frank-wolfe algorithm. In *European Conference on Computer Vision*, 253–268.

Kanan, C.; Tong, M. H.; Zhang, L.; and Cottrell, G. W. 2009. Sun: Top-down saliency using natural statistics. *Visual Cognition* 17(6-7):979–1003.

Kumar, N.; Belhumeur, P. N.; Biswas, A.; Jacobs, D. W.; Kress, W. J.; Lopez, I. C.; and Soares, J. V. 2012. Leafsnap: A computer vision system for automatic plant species identification. In *European Conference on Computer Vision*. 502–516.

Li, G., and Yu, Y. 2015. Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 5455–5463.

Li, G., and Yu, Y. 2016. Deep contrast learning for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Li, H.; Meng, F.; Luo, B.; and Zhu, S. 2014a. Repairing bad co-segmentation using its quality evaluation and segment propagation. *IEEE Transactions on Image Processing* 23(8):3545–3559.

Li, Y.; Hou, X.; Koch, C.; Rehg, J. M.; and Yuille, A. L. 2014b. The secrets of salient object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 280–287.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 740–755.

Liu, T.; Yuan, Z.; Sun, J.; Wang, J.; Zheng, N.; Tang, X.; and Shum, H.-Y. 2011. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence* 33(2):353–367.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Mahadevan, V., and Vasconcelos, N. 2009. Saliency-based discriminant tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1007–1013. IEEE.

Meng, F.; Li, H.; Liu, G.; and Ngan, K. 2012. Object co-segmentation based on shortest path algorithm and saliency model. *IEEE Transactions on Multimedia* 14(5):1429–1441.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2015. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics* 23(3):309–314.

Rubinstein, M.; Joulin, A.; Kopf, J.; and Liu, C. 2013. Unsupervised joint object discovery and segmentation in internet images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 1939–1946.

Shen, X., and Wu, Y. 2012. A unified approach to salient object detection via low rank matrix recovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, 853–860.

Tan, Z.; Wan, L.; Feng, W.; and Pun, C.-M. 2013. Image co-saliency detection by propagating superpixel affinities. In *IEEE International Conference onAcoustics, Speech and Signal Processing*, 2114–2118.

Tang, K.; Joulin, A.; Li, L.-J.; and Fei-Fei, L. 2014. Co-localization in real-world images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1464–1471.

Wang, Q.; Yuan, Y.; Yan, P.; and Li, X. 2013. Saliency detection by multiple-instance learning. *IEEE transactions on cybernetics* 43(2):660–672.

Wang, L.; Wang, L.; Lu, H.; Zhang, P.; and Ruan, X. 2016. Saliency detection with recurrent fully convolutional networks. In *European Conference on Computer Vision*, 825–841.

Wright, J.; Ganesh, A.; Rao, S.; Peng, Y.; and Ma, Y. 2009. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, 2080–2088.

Xu, X.; Wan, L.; Liu, X.; Wong, T.-T.; Wang, L.; and Leung, C.-S. 2008. Animating animal motion from still. In *ACM Transactions on Graphics*, volume 27, 117.

Yu, H.; Zhou, Y.; Qian, H.; Xian, M.; Lin, Y.; Guo, D.; Zheng, K.; Abdelfatah, K.; and Wang, S. 2015. Loosecut: Interactive image segmentation with loosely bounded boxes. *arXiv preprint arXiv:1507.03060*.

Yu, H.; Xian, M.; and Qi, X. 2014. Unsupervised co-segmentation based on a new global gmm constraint in mrf. In *IEEE International Conference on Image Processing*, 4412–4416.

Zhang, D.; Han, J.; Li, C.; and Wang, J. 2015a. Co-saliency detection via looking deep and wide. In *IEEE conference on Computer Vision and Pattern Recognition*, 2994–3002.

Zhang, D.; Meng, D.; Li, C.; Jiang, L.; Zhao, Q.; and Han, J. 2015b. A self-paced multiple-instance learning framework for co-saliency detection. In *IEEE International Conference on Computer Vision*, 594–602.

Zhang, D.; Han, J.; Han, J.; and Shao, L. 2016. Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining. *IEEE transactions on neural networks and learning systems* 27(6):1163–1176.

Zhao, R.; Ouyang, W.; Li, H.; and Wang, X. 2015. Saliency detection by multi-context deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1265–1274.

Zhao, R.; Ouyang, W.; and Wang, X. 2013. Unsupervised salience learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3586–3593.

Zitnick, C., and Dollár, P. 2014. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*. 391–405.