SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

Special Topic: Deep Learning for Computer Vision

November 2023, Vol. 66 210101:1–210101:18
https://doi.org/10.1007/s11432-022-3686-1

An end-to-end network for co-saliency detection in one single image

Yuanhao YUE[1,2], Qin ZOU[1,2*], Hongkai YU[3], Qian WANG[1],
Zhongyuan WANG[2] & Song WANG[4]

[1] *School of Cyber Science and Engineering, Wuhan University, Wuhan* 430072*, China;*
[2] *School of Computer Science, Wuhan University, Wuhan* 430072*, China;*
[3] *Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland* OH 44115*, USA;*
[4] *Department of Computer Science and Engineering, University of South Carolina, Columbia* SC 29200*, USA*

Received 16 July 2022/Revised 24 November 2022/Accepted 4 December 2022/Published online 17 May 2023

Abstract Co-saliency detection within a single image is a common vision problem that has not yet been well addressed. Existing methods often used a bottom-up strategy to infer co-saliency in an image in which salient regions are firstly detected using visual primitives such as color and shape and then grouped and merged into a co-saliency map. However, co-saliency is intrinsically perceived complexly with bottom-up and top-down strategies combined in human vision. To address this problem, this study proposes a novel end-to-end trainable network comprising a backbone net and two branch nets. The backbone net uses ground-truth masks as top-down guidance for saliency prediction, whereas the two branch nets construct triplet proposals for regional feature mapping and clustering, which drives the network to be bottom-up sensitive to co-salient regions. We construct a new dataset of 2019 natural images with co-saliency in each image to evaluate the proposed method. Experimental results show that the proposed method achieves state-of-the-art accuracy with a running speed of 28 fps.

Keywords saliency detection, convolutional neural network, regional feature mapping, co-saliency detection, deep learning

Citation Yue Y H, Zou Q, Yu H K, et al. An end-to-end network for co-saliency detection in one single image. Sci China Inf Sci, 2023, 66(11): 210101, https://doi.org/10.1007/s11432-022-3686-1

1 Introduction

In the past two decades, saliency detection [1–5] has attracted a lot of attention from the visual media community, e.g., image/video content analysis [6–8], background extraction [9, 10], and light-field imaging [11]. Co-saliency is among the various types of saliency that refer to the highlighted saliency shared by multiple similar objects. However, most existing methods define co-saliency detection as a problem of cross-image co-saliency detection [12–15], which highlights the objects co-occurring in multiple images, e.g., a pair of images or a sequence of video frames [8, 16, 17]. While cross-image co-saliency detection has received considerable attention, another co-saliency detection problem—within-image co-saliency detection—has not been well addressed yet.

The within-image co-saliency detection aims at highlighting multiple occurrences of the same object class with a similar appearance in a single image. In human vision, this is a common visual ability that has been frequently used in our daily life, such as spotting the same team's players on the sports field, counting the red apples on a tree, identifying the sunflowers in the farmland [18], and spotting texts in nature scenes [19–21]. However, in computer vision, asking the algorithm itself to find proposal groups that include co-occurring saliency regions in a single image remains a challenge. Until now, only a few researchers have given this problem a serious and direct consideration [22].

While humans can detect co-saliency by noticing the multiple occurrences of the same object class, computers may have difficulty because they cannot recognize all the objects in the world. A computer

[*] Corresponding author (email: qzou@whu.edu.cn)

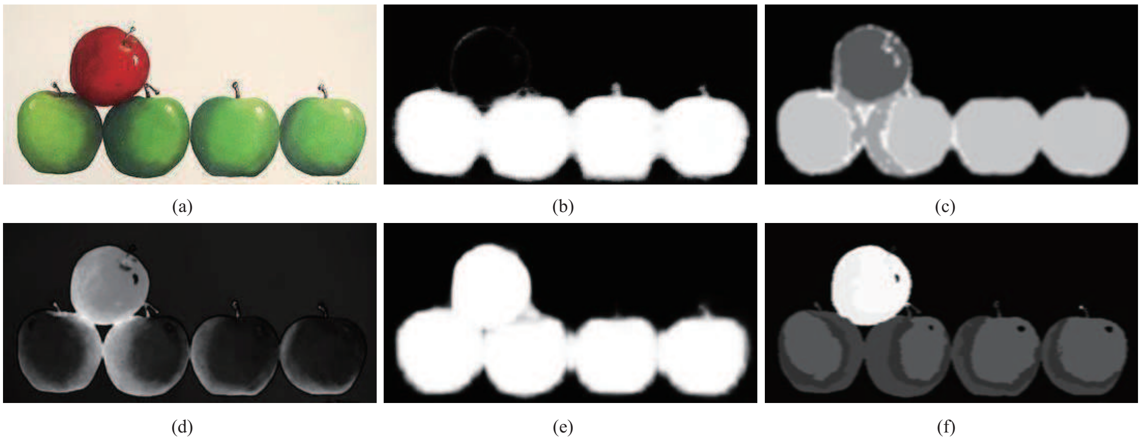

Figure 1 (Color online) Illustration of within-image co-saliency detection. (a) An input image containing four green apples and one red apple; (b) the co-saliency map obtained by the proposed method; (c) the co-saliency map produced by [22]; (d–f) the results obtained by three general saliency detection methods without considering the co-occurrence of the same object class.

has no idea what an object is. Some researchers have proposed developing a universal model to recognize all objects. However, in the current stage of deep learning, a universal object model remains unsolved. The problem then becomes how to build an effective detector for co-saliency detection within an image without a universal object model. Yu et al. pioneered the research in [22], by developing a two-stage method. The first stage generates object proposals using the classical EdgeBox [23], and the second stage computes co-saliency by deriving proposal groups with good common saliency, based on mutual similarity scores. This typical bottom-up strategy often suffers from high computation costs in processing low-level features and low integrity in constructing a unified optimization model for co-saliency learning; i.e., it cannot work in an end-to-end manner.

Other typical studies, such as RFCN [24], DCL [25], and CPFE [26], learn high-level features for salient object detection using deep neural networks, which directly use human-annotated saliency masks to guide the training in a top-down manner. However, co-saliency is intrinsically perceived in a complex manner that combines bottom-up and top-down strategies. Solely using one of these strategies will not yield satisfactory results. Figure 1 shows an example in which Figures 1(d)–(f) are obtained using the general saliency-detection methods.

It can also be noticed that co-saliency is a complex problem involving texture, shape, color, etc. It is very difficult to build an accurate co-saliency model that applies to all scenarios. For example, when judging a co-saliency, it is difficult to determine which is more important: texture, shape, or color.

Based on the discussion above, we practically simplify the problem by considering the color as the primary visual information of co-saliency in our research. We define co-saliency as salient objects with similar textures, shapes, and the same color. That is to say, salient objects that have similar textures and shapes but different colors will not be considered co-saliency. We can construct the datasets without controversy and perform a uniform evaluation under this constraint. Then, the top-down and bottom-up strategies are combined to propose an end-to-end trainable deep neural network for within-image co-saliency detection. When using high-level features for common saliency object identification, the key point is how to make high-level layers sensitive to the co-saliency regions. We used an encoder-decoder architecture as the backbone net for co-saliency map prediction, and two branch nets to guide the training process and make the learned model more sensitive to co-salient regions. One branch net is a region proposal network [27] (RPN), which generates triplet proposals. The other branch net is a regional feature mapping (RFM), which works in a bottom-up manner to drive the backbone net to be sensitive to co-salient regions. The training loss was built using the similarity of the triplet features [28, 29].

The entire network is trained in a simple data-driven manner, avoiding complex fusion strategies, increasing speed, and simplifying training. After training is done, the backbone net is used to predict end-to-end co-saliency. The main contributions of this work are three-fold.

- First, a unified end-to-end network for co-saliency detection in a single image is proposed. It combines the top-down and bottom-up approaches: a backbone net, i.e., encoder-decoder net, is used for co-saliency map prediction, and two branch nets, i.e., RPN and RFM, are used to drive the network to be sensitive to co-salient regions.

- Second, an online training sample selection strategy is presented. It enhances the proposed method by assisting it in achieving significantly higher accuracy than the offline selection strategy with random scaling and offset.

- Third, a new dataset for within-image saliency detection was constructed. It contains 2019 nature images from over 300 object classes, each with instance-level annotations. The dataset, as well as the codes and trained models, will be made available to the public, serving as a benchmark and promoting research in this field.

2 Related work

In this section, we introduce the literature review on the related research topics, including saliency detection, co-saliency detection, and within-image co-saliency detection.

2.1 Saliency detection

Saliency detection is a fundamental problem in computer vision [30,31]. It highlights regions in a single image that attracts human visual attention. It is usually divided into two categories: eye fixation prediction [32,33] and salient object detection (SOD) [34,35]. The purpose of SOD is to accurately highlight and segment the salient object regions in the image. The earliest saliency detection models were heavily influenced by human visual attention mechanisms. Itti et al. [36] proposed a saliency detection method based on center-surround mechanisms. In order to predict the salient regions, some early studies focused on local contrast [32,36,37] and global contrast [4] to separate the salient object from the image background. SOD was considered as an image segmentation problem, and Liu et al. [37] used the conditional random field to effectively combine low-level features. In [4], a regional contrast algorithm was proposed during the evaluation of the global contrast differences and the spatially weighted coherence scores. To obtain more accurate object boundaries, some methods started to introduce the prior knowledge in the model, such as background priors [9,38], and high-level priors [3,39,40]. In [38], boundary and connectivity priors were used to provide more cues for SOD. Similarly, Zhu et al. [9] proposed a principled optimization framework for integrating multiple low-level cues to obtain clean and uniform saliency maps. Goferman et al. [3] presented a context-aware saliency detection algorithm based on four principles observed in the psychological study. In [41], bottom-up strategies were proposed for saliency detection using low-level features based on the amplitude spectrum, edges, gradients, etc.

Since 2015, many deep learning-based SOD methods have been proposed. Early deep learning-based SOD [42–45] mainly used multi-layer perceptron classifiers to predict the pixel-level saliency scores. Zhao et al. [42] proposed a method for extracting local and global contexts from two super-pixel-centered windows of different sizes using two pathways. Zhang et al. [43] used a convolutional neural network (CNN)-based model to generate a set of proposal bounding boxes and selected an optimized compact subset of bounding boxes for multiple salient objects. Recently, fully convolutional networks have received widespread applications in SOD [46–49]. Cheng et al. [46] proposed to introduce short connections to the skip-layer structures within the HED architecture [50], which fuses multilevel and contrast features in a top-down manner. In [47], global and local pixel-wise contextual attention modules were embedded in the U-Net [51] structure, which learns to selectively draw attention to informative context locations. In [52], image sequences assisted with optical flow information were processed by convolutional neural networks for saliency inference. With the acquisition technology development, more comprehensive information, such as depth cue, interimage correspondence, or temporal relationship, is available to extend image saliency detection to RGBD saliency detection, co-saliency detection, or video saliency detection. Cong et al. [53] reviewed various types of saliency detection algorithms and summarized these important issues. Zhang et al. [54] proposed global context-aware attention structures for optical remote sensing images (RSIs). Li et al. [55] proposed a two-stream pyramid network to extract a set of complementary information in RSIs. Chen et al. [56] proposed a mode fusion algorithm for RGBD saliency detection based on image and depth information. Cong et al. [57] proposed a depth-guided transformation algorithm from RGB saliency to RGBD saliency.

2.2 Co-saliency detection

Generally, co-saliency detection highlights common salient objects from multiple images. It has a wide range of applications in many computer vision tasks, including video object segmentation [58, 59]. Traditional co-saliency methods [12, 60, 61] rely primarily on low-level features and saliency cues such as interimage saliency cue, intraimage saliency cue, and repetitive cue. The co-saliency was modeled as a linear combination of the single-image saliency map and the multi-image saliency map in [61]. The task of co-saliency was extended to image groups with more than two related images in [12]. Recently, following the successful application of CNNs in saliency detection, some researchers have attempted to directly learn the patterns of the co-salient objects from a given image group. Unlike most traditional co-saliency detection methods that are based on saliency cues, CNN-based methods [16, 62–66] learn co-saliency patterns through data-driven supervision, thereby avoiding the limitations caused by handicraft features. Zhang et al. [62] proposed a self-paced multi-instance learning framework for detecting effective co-saliency patterns from numerous ambiguous image regions. Wei et al. [16] proposed an end-to-end group-wise deep co-saliency detection method to adaptively learn group-wise interaction information for a group of images. Zhang et al. [63] used the Gromov-Wasserstein distance to measure the similarity of a group of pictures and match their characteristics, which could avoid the noise from different picture styles, colors, and contrast. Zhang et al. [64] proposed a consensus-aware dynamic convolution model to successfully summarize the consensus features and search for corresponding objects in each image. Tang et al. [65] contributed a new dataset and proposed a simple and effective benchmark framework. Ren et al. [66] invented a scale-aware loss to help the model in capturing the scale of different groups and discriminatively process the groups during the training phase.

2.3 Within-image co-saliency detection

Within-image co-saliency can be considered a special case of co-saliency. The within-image co-saliency detection highlights the common salient object regions within an image. However, all previous studies, including saliency detection and co-saliency detection, have failed to solve this problem. Yu et al. [22] first raised this issue and proposed a bottom-up method to address it. Specifically, they used an optimization algorithm to derive a set of proposal groups and calculated a co-saliency map, then used a low-rank-based algorithm to fuse the maps calculated from all the proposal groups and generate the final co-saliency map. However, in this framework, most knowledge about proposal groups or co-salient regions heavily relies on the manually designed metrics inferred.

Generally, data-driven methods have the potential to capture better patterns of co-salient objects than human-designed metrics-based methods. However, there has been little research into end-to-end trainable CNN-based methods for within-image co-saliency detection. Yu et al. [67] detected co-saliency in multiple images using within-image co-saliency algorithms in [22]. An improved version of [22] proposed a multi-scale multiple instance learning model for co-saliency detection, which uses an easy-to-hard learning strategy for data fusion [68]. However, this framework also requires sample superpixel proposals at first and the strategy of multimethod fusion, which may result in unacceptable performance degradation.

Within-image co-saliency detection can help segment multiple instances of an object class in an image, estimating the number of instances of the same object class, which can be applied to industrial scenarios such as counting the objects on a production line and detecting anomalous objects from the targets.

Another work related to ours is supervised semantic segmentation—a fundamental problem in computer vision. In an ideal condition, we can do within-image co-saliency detection based on semantic segmentation. That is, we perform semantic segmentation [51] on a given image and then conduct matching across the segmented objects. If two or more detected objects show a high level of similarity and belong to the same object class, we highlight them in the co-saliency map. Nevertheless, in most cases, semantic segmentation models can detect only known object classes that have been pretrained through supervised learning [69]. We detected within-image co-saliency without assuming any specific object class or recognizing any objects in the image, as in most previous studies on saliency detection [34, 35].

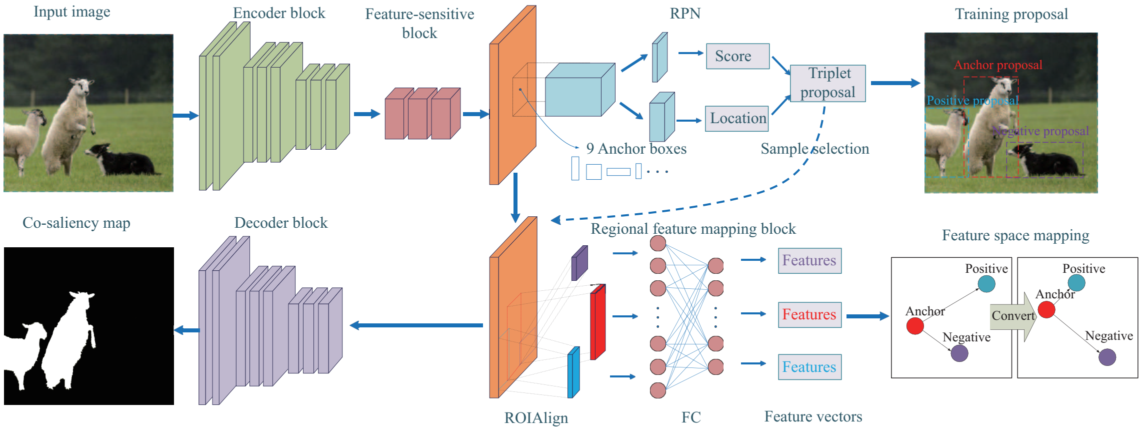

Figure 2 (Color online) The schematic illustration of our network. The encoder and decoder blocks use the skip-layer structures within the U-Net architecture. The feature-sensitive block is composed of three trainable 3 × 3 convolution layers. The feature-sensitive block is supervised by two kinds of supervisory signals during the training phase: region proposal network (RPN) and regional feature mapping (RFM). RPN regresses saliency region confidence and locations and provides training samples for the RFM. The RFM network transforms the corresponding region features to feature vectors of a fixed length.

3 Proposed approach

3.1 Problem formulation

The goal of within-image co-saliency detection is to find common and salient objects with similar appearances in a single image. If two or more detected objects have a high level of similarity and belong to the same category, the corresponding regions in the resulting co-saliency map will be highlighted. The experimental scenario was simplified by removing images that did not contain objects from the same category. Given a training data set containing N images as $S = \{(X^n, Y^n), n = 1, \ldots, N\}$, where $X^n = \{x_i^{(n)}, i = 1, \ldots, I\}$ denotes an input image, $Y^n = \{y_i^{(n)}, i = 1, \ldots, I, y_i^{(n)} \in \{0,1\}\}$ denotes the ground-truth binary map corresponding to the input image containing saliency objects of the same class, I denotes the number of pixels in the image, the goal of the co-saliency detection model $F(\cdot)$ is to train the network to produce prediction saliency maps $P = \{p_i^{(n)}, n = 1, \ldots, N\}$ approaching to the ground truth annotated by experts:

$$P = F(X; \Theta), \tag{1}$$

where Θ represents the optimized parameters of the model in this task.

3.2 Methodology

The model is divided into four major parts, as shown in Figure 2. First, the original image is fed into the pretrained encoder block, which generates high-level semantic feature maps. The feature maps are then fed into the feature-sensitive block to extract the feature map that is sensitive to the image's co-saliency regions. In the training phase, the feature-sensitive block is supervised by two kinds of supervisory signals: one from the decoder block and the other from the RFM. The decoder block generates the corresponding final saliency map, while the RFM network generates the high-level feature coding and position regression of the salient object region corresponding to the original input image. In the evaluation phase, co-saliency regions are predicted only through the encoder-decoder backbone net and the feature-sensitive block.

3.2.1 *Encoder-decoder block*

The encoder and decoder blocks in our network are based on U-Net [51]. The skip-layer structures are used in the encoder and decoder blocks. The encoder block uses the convolution part of the pretrained VGG-16 network [70] targeted on the classification task on the ImageNet dataset [71]. Its parameters are frozen, and thus, it will not participate in the subsequent training. This will result in better generalization and more stable feature high-level feature abstractions while avoiding the influence of specific categories. The decoder and encoder blocks are almost completely symmetrical, except that the downsampling operation is replaced by an upsampling of the feature map. In the decoder block, the decoder layer fuses the feature map of the corresponding encoder layer to assemble a more precise output. Therefore, the underlying features of the encoder block can be transmitted more easily to the decoder and hence retain more accurate information about object boundaries.

3.2.2 *Feature-sensitive block*

High-level semantic guidance, as well as interimage interaction at the semantic level, is important for co-saliency detection and is inspired by the mechanism of human visual co-saliency. This module aims to optimize the within-image co-saliency task in a simple data-driven manner, thereby avoiding complex fusion strategies, improving the speed, and making training simple and effective.

The structure is composed of three trainable 3×3 convolution layers. Our goal is to learn the mapping and train it to be sensitive to similar feature regions such that

$$R = F_{\text{sensitive}}(H, \theta), \tag{2}$$

where H is the semantic feature retained after feature extraction by encoder block, $F_{\text{sensitive}}$ is a mapping function that takes the feature H as input, and outputs the mapped features into the decoder block by learning a set of hidden layers parameters θ.

We try to use the RFM network added in the training phase to supervise the feature-sensitive module and make it sensitive to common saliency object regions. The high-level semantic information is passed to guide the learning of the decoder network, thereby achieving better saliency segmentation results.

3.2.3 *Regional feature mapping block*

Considering two similar salient target object regions A and B, in the form of the rectangular bounding boxes, in the image X, corresponding to the regions i and j in the feature map H produced by the encoder block, is mapped to R, as marked in orange in Figure 2, through the feature-sensitive blocks R_i and R_j, respectively, and then the RFM network transforms the region features in the R into the corresponding feature vectors, as formulated by

$$V_i = F_{\text{RFM}}(R_i, \theta), \tag{3}$$

where F_{RFM} is a mapping function that takes the region feature map R_i as the input, and generates the corresponding fixed length feature vector V_i by learning a set of hidden-layer parameters θ.

Therefore, the RFM module's optimization objective is to minimize the Euclidean distance between any two similar salient object feature vectors. However, there are two challenges: the first is determining how to map feature regions R_i and R_j of different scales to the vectors of the same length, and the second is determining how to obtain the same feature representation for two identical objects under the influence of scales, shooting angles, and illumination. Therefore, it is difficult to measure the characteristic representation of two identical objects by Euclidean distance.

For the first problem, inspired by previous studies [27, 72], ROIAlign [72] was used to map regional features of different sizes to feature vectors of the same length, which is more accurate on pixel-align task compared with ROIPool [27]. The feature regions of different scales are subdivided into spatial ROI bins, such as 7×7 bins, which use bilinear interpolation to compute the exact values of the input features at four regularly sampled locations in each ROI bin and aggregate the results in terms of a maximum or averaging operation. However, because of this structure, the resulting feature vector will introduce more noise. Hence, we try to construct a special feature space in which the features of similar salient object regions have a close distance in the clustering. Specifically, the Euclidean distance of the feature vector corresponding to the similar object region is less than that of the corresponding dissimilar object region.

Figure 3 shows the three different selected saliency proposals in the form of rectangular bounding boxes, namely the anchor region (A), the positive region (P), and the negative region (N) in the image I. The regions of A and P include the same salient object classes. The regions of N include a different salient region or a nonsalient object region.

The feature vector corresponding to the triplet regions V_A, V_P, and V_N should satisfy the following relationship in this feature space:

$$\| V_A - V_P \|_2^2 < \| V_A - V_N \|_2^2 . \tag{4}$$

https://engine.scichina.com/doi/10.1007/s11432-022-3686-1

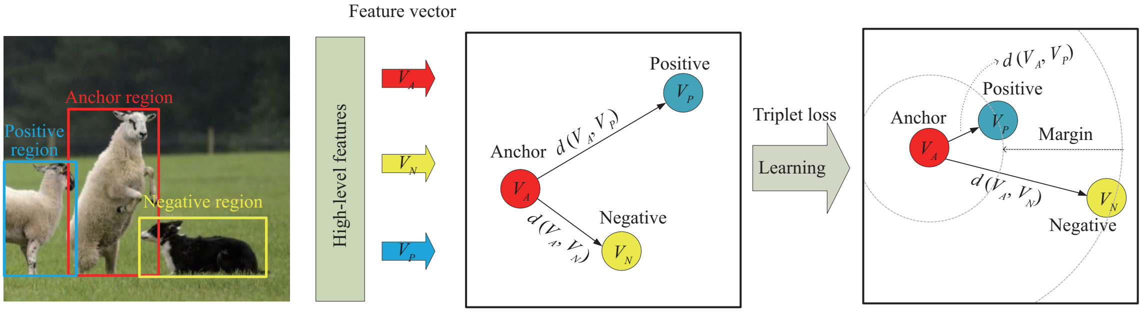

Figure 3 (Color online) An illustration of the learning process using triplet loss. The positive pair consists of an anchor region and a positive region, which include two similar salient target object regions. The negative pair consists of an anchor region and a negative region. The feature vector mapping by the corresponding saliency regions in Euclidean feature space satisfies the clustering relationship during the training phase.

3.3 Training

(1) Sample selection strategy. An effective sample selection strategy is vital to get more robust feature space and avoid overfitting in case of insufficient sample selection.

In the selection of triplet samples, the positive sample pair needs to be augmented. The positive pair is screened by generating a Jaccard overlap between the sample regions and the ground-truth boxes. Specifically, during the training phase, the selection of a positive pair must satisfy that each generated region contains only one salient object and has a matching box to the ground truth with a Jaccard overlap higher than a threshold (0.5). It is worth noting that the two generated sample regions can be the same salient target. The negative sample could be a dissimilar saliency proposal with a Jaccard overlap of less than a threshold of 0.1 or the background regions. For example, the algorithm selects a triplet sample from an image that contains three salient objects with similar appearances and a salient object with a different appearance. First, it randomly selects one of the three salient objects as an anchor region according to the ground truth. Next, it selects a positive region. The positive sample region can be selected from the other two samples or the anchor object itself as long as it meets the above requirements. Finally, it selects a negative region. The negative sample region can be selected from the interference object, the nonsaliency background region, or the area whose salient objects have a Jaccard overlap lower than a threshold of 0.1.

We propose using an offline sample selection strategy and an online sample selection strategy to generate positive and negative samples. The offline strategy is to use ground-truth boxes to add random scaling and offset within a certain range, thereby satisfying the sample generation conditions stated above. The length-width ratio of the generated region ranges from 0.5 to 2, and its area ranges from 128^2 to 512^2. For each training image, 32 positive samples and 96 negative samples are generated, and 128 triplets are randomly constructed.

The other strategy is to generate training samples online. We hope to find more difficult positive and negative samples to satisfy the generation conditions using this strategy. To achieve online triplet mining, we should evaluate the selected sample regions. We use the idea of online hard example mining [73], which was designed to improve the accuracy of Fast RCNN. Specifically, we add a region proposal network (RPN) during the training process. In our model, RPN is considered as a sample region generator, which evaluates the confidence of the sample regions based on their loss during the training process. The hard examples were selected by sorting the regions based on the training loss and using the standard nonmaximum suppression to extract a highly overlapping region. Finally, the hard positive and hard negative samples were selected, which satisfy the generation conditions of triplet sample construction.

In the experiment, we will compare the performance of the online and offline sample selection strategies.

(2) Loss function. As shown in Figure 2, our network has three outputs during the training phase. The total loss function is a weighted sum of the following three parts:

$$L_{\text{total}} = \alpha L_{\text{decoder}} + \beta L_{\text{RPN}} + \gamma L_{\text{RFM}}, \tag{5}$$

where α, β, γ are all set to 1.

The decoder block produced the corresponding saliency map [74]. The loss function of the decoder part is computed by a pixel-wise cross-entropy loss with sigmoid between predicted saliency map P and the ground truth Y:

$$L_{\text{decoder}} = -\frac{1}{N}\sum_{i=1}^{N}\Big(Y_i\log(P_i) + (1-Y_i)\log(1-P_i)\Big), \tag{6}$$

where N is the number of pixels in the ground truth. (P_i, Y_i) denotes the pixel values corresponding to the predicted saliency map and the ground truth.

For training the RPN network, we selected anchor boxes on 3 scales, with 3 boxes on each scale. The areas of the anchor boxes are 128^2, 256^2, and 512^2 on the three scales, respectively, and the edge ratios of the three boxes on the same scale are 1:1, 1:2, and 2:1. We select the anchor box that holds the highest Jaccard overlap with the ground-truth box as a positive sample. As a data augmentation strategy, we selected all the anchors that have a Jaccard overlap higher than (0.5) with the ground-truth box as positive samples. The label t is 1 if the anchor is positive; otherwise, it is 0 if the anchor is negative. The loss function is similar to that of Faster RCNN [27]. We regress the anchor offsets for the center point (x, y), the length values for width (w) and height (h), and the confidence for the saliency object (c). The loss function is a weighted sum of the localization part and the confidence part:

$$L_{\text{RPN}} = \frac{1}{N}\big(L_{\text{conf}}(t, c) + \alpha L_{\text{loc}}(t, l, g)\big), \tag{7}$$

where

$$L_{\text{loc}}(t, l, g) = \sum_{i\in\text{Pos}}^{N} \text{smooth}_{L1}(l_i - \hat{g}_i), \tag{8}$$

and N is the number of matched anchor boxes. The weight coefficient α is set to 1 by cross-validation. (l, g) contains the predicted box information including (x, y, w, h) and the label. The localization loss is the smooth L1 loss [75]. Let τ be the corresponding anchor box information for g. Then we have

$$\hat{g_i^x} = (g_i^x - \tau_i^x)/\tau_i^w,$$

$$\hat{g_i^y} = (g_i^y - \tau_i^y)/\tau_i^h,$$

$$\hat{g_i^w} = \log\left(\frac{g_i^w}{\tau_i^w}\right),$$

$$\hat{g_i^h} = \log\left(\frac{g_i^h}{\tau_i^h}\right),$$

where $\hat{g}_i$ represents the anchor offset between the ground-truth box and the corresponding anchor box.

We use a weighted sum of triplet samples to train the RFM network: anchor(a), positive(p), and negative(n). The weight is calculated by

$$\beta_i = I(\text{anchor}, \text{gt})I(\text{positive}, \text{gt}), \tag{9}$$

where $I(\text{anchor}, \text{gt})$ is the Jaccard overlap between the anchor box and the corresponding ground-truth box, and $I(\text{positive}, \text{gt})$ is the Jaccard overlap between the positive box and the corresponding ground-truth box.

As shown by Figure 3, the loss of a triplet sample in the distance embedding space d is defined as

$$L_{\text{RFM}} = \max\big(d(V_A, V_P) - d(V_A, V_N) + \text{margin}, 0\big), \tag{10}$$

where V_A, V_P, and V_N are the feature vectors corresponding to regions in the triplet samples, and the margin is a margin placed between the positive and negative pairs. When minimizing this loss, it pushes $d(V_A, V_P)$ to be 0 and $d(V_A, V_N)$ to be greater than '$d(V_A, V_P)$ + margin'. As for simple examples, this loss becomes zero.

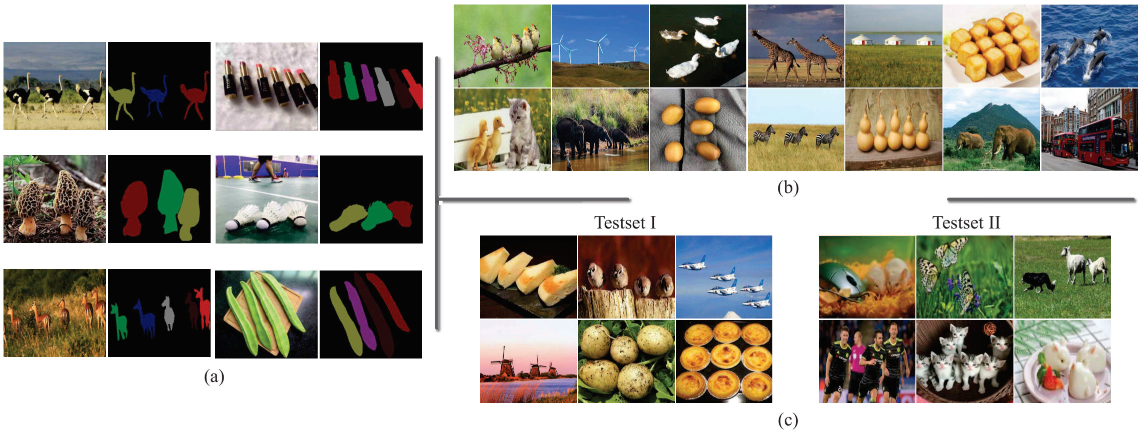

Figure 4 (Color online) Sample images of the WhuCoS dataset. (a) Annotated visual examples; (b) examples from the training set; (c) examples from the testing set. Testset II is a subset of Testset I, which contains 100 challenge images. Each challenge image contains multiple salient objects with the same appearance and at least one salient object that looks different.

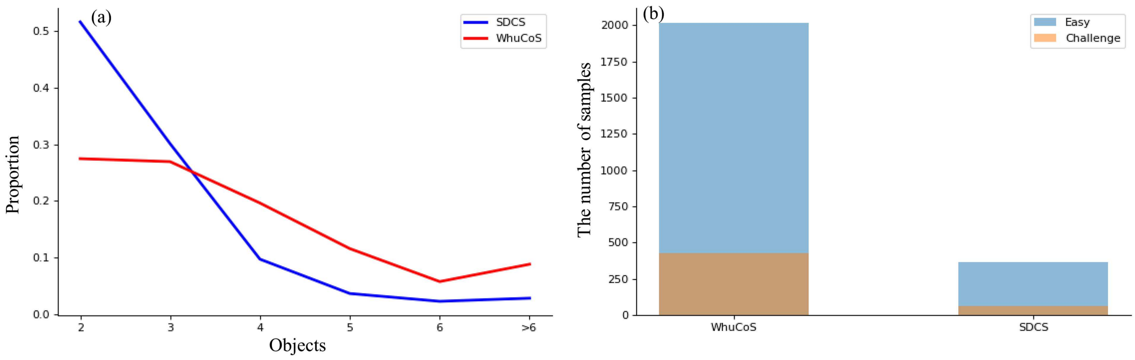

Figure 5 (Color online) Statistic data of the WhuCoS and SDCS datasets. (a) The proportion of the number of salient objects in one single image; (b) the number of easy and challenging samples in the dataset.

4 Experiments

4.1 Datasets

The general standard dataset, such as iCoseg [76], MSRA [37], and HKU-IS [44], are all collected for the evaluation of within-image saliency detection or cross-image co-saliency detection. These datasets generally include just one salient object or multiple objects of different categories. The SDCS dataset [22] is targeted for within-image co-saliency detection; however, it only contains two to three co-salient objects in a single image. Hence, we collect a larger dataset named WhuCoS for within-image co-saliency detection. In this work, we use the WhuCoS dataset and the SDCS dataset for evaluation.

WhuCoS dataset[1]. This dataset is instance-level annotated for within-image co-saliency detection. It contains 2019 natural images, over 300 categories of daily necessities, and 7000 salient object instances. To simplify our experiment, in our data collection, we define the objects from the same category and with the highest occurrence as the co-salient objects. Figure 4 shows some sample images. The WhuCoS dataset contains an average of 3.52 objects per image. Each image contains at least two or more identical salient objects. We label its position and contour for each object. The image sizes range from 450×256 to 808×1078 pixels. Additionally, about one-fifth of the images contain interfering objects, as shown by the images in Figure 4(c). For the test, we construct two testsets for WhuCoS. The statistics of the WhuCoS and SDCS datasets is shown in Figure 5.

- Testset I. It contained 524 images for the test. These images are randomly selected from WhuCoS. Exactly, WhuCoS contains both simple and difficult samples. For a simple sample, it only contains multiple salient objects with the same appearance.

[1] Codes and data are available at https://github.com/qinnzou/co-saliency-detection.

https://engine.scichina.com/doi/10.1007/s11432-022-3686-1

- Testset II. It contained 100 images that were selected from Testset I. These 100 images are hard samples that are more challenging to handle. Each image contains multiple salient objects with the same appearance and at least one salient object with a different appearance. In Testset II, there were 436 salient objects, including 342 co-saliency objects and 94 other salient objects. Samples are shown in Figure 4(c).

SDCS dataset. It consists of 364 color images, including 65 challenging images and 299 easy images. About 100 images were selected from the iCoseg, MSRA, and HKU-IS datasets. Each image contains 2.61 objects on average. For this dataset, 300 images are used for training and the remaining 64 for the test.

4.2 Evaluation metrics

We examine the performance in the experiments by computing pixel-wise errors between the prediction saliency map and the mask of ground truth. The metrics we used include the mean absolute error (MAE), the precision and recall (PR) [34,77], and the F-measure [25,78]. The precision and recall are formulated as follows:

$$\text{precision } = \frac{|P \cap G|}{|P|}, \text{ recall } = \frac{|P \cap G|}{|G|}, \tag{11}$$

where P denotes the ratio of detected salient pixels in the predicted co-saliency map, and G denotes the ratio of detected salient pixels in the ground-truth map, according to an adaptive threshold. Generally, high precision and recall are both required, which can be represented by a fused metric F-measure,

$$F_\beta = \frac{\left(1+\beta^2\right) \times \text{precision} \times \text{recall}}{\beta^2 \text{precision} + \text{recall}}, \tag{12}$$

where β^2 is set as 0.3 to give the precision a higher weight than recall as suggested in [78]. MAE score can be computed as

$$\text{MAE} = \frac{1}{W \times H} \sum_{x=1}^{H} \sum_{y=1}^{W} \|P(x,y) - G(x,y)\|, \tag{13}$$

where H and W denote the height and width of a saliency map, respectively.

Besides the above four metrics, two recently-proposed metrics, S-measure [34], and E-measure [79], are also included in the evaluation.

4.3 Implementation details

In this study, the weights of the 16-layer VGG pretrained on ImageNet are used to initialize and fix the encoder block. Each time the model is trained, only one image is used. To fully train the network, various data-augmentation operations are applied before the image input, including multiscale scaling, random object clipping, and random occlusion. The experimental results in Table 1 have shown that data enhancement is very effective in reducing the overfitting of triplet loss and improving the generalization ability of the model. We choose SGD as the optimizer in our experiment, and the learning rate and the weight decay are set to 1×10^{-5} and 5×10^{-4}, respectively. Finally, the learning rate decreases to 1×10^{-6} after 30000 training iterations. It requires about 80000 training iterations for convergence. The proposed network is trained on two NVIDIA GTX 1080Ti 11G GPUs, where the GPU memory consumption is about 18660 Mb. The training time on the WhuCoS dataset is approximately 13 h. We conducted the test using a single GPU. In the test, with a single image as the input, the running speed of our method was 28 fps.

The hyperparameter margin in (10) is set by evaluating the model's training results on the verification set. A larger margin leads to a greater distance between the positive and negative samples. However, a too large margin may make the model difficult to converge. We set the margin to 0.1, 0.3, and 1.0 in the training. We find that the model training process is difficult to converge when margin = 1.0, and the converge speed has almost no difference between 0.1 and 0.3. Therefore, we set the default value of the margin to 0.3.

Table 1 An ablation analysis of various design choices for the proposed method

Module	With/without							
U-Net (baseline)	✓	✓	✓	✓	✓	✓	✓	✓
+ Data augumentation		✓		✓		✓		✓
+ Feature-sensitive block			✓	✓	✓	✓	✓	✓
+ RFM (offline training)					✓	✓		
+ RFM (online training)							✓	✓
F-measure (%)	82.4	83.2	82.3	83.6	88.1	89.2	89.7	90.4

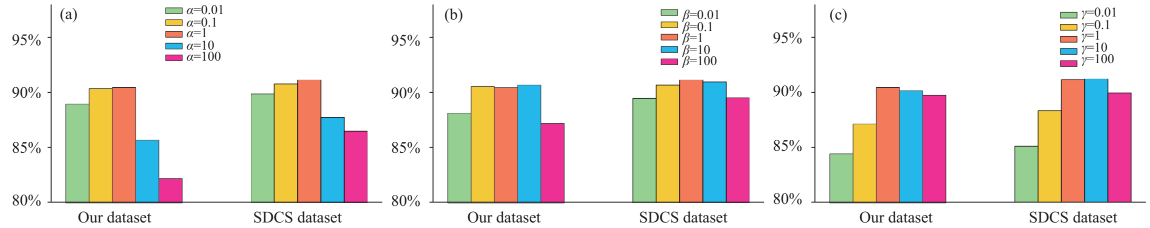

Figure 6 (Color online) An ablation analysis of different loss weights for the proposed method on both the SDCS dataset and our WhuCoS dataset. α, β, γ are the three loss weights in (5). (a) α is the loss weight of the decoder block. (b) β is the loss weight of RPN. (c) γ is the loss weight of RFM. The vertical axis represents the F-measure value. While one of the weights is tuned, the other two weights are set to 1.

4.4 Ablation analysis

To better analyze the role of each module, we conducted an ablation study using numerous horizontal comparison experiments. The results are shown in Table 1 and Figure 6. Table 1 shows that the first column in the right part corresponds to the results using only the encoding and decoding network of the U-Net structure, which achieves an accuracy of 82.4%. In the third column, we can see that the performance slightly decreases when adding some additional convolution layers as feature-sensitive layers on U-Net, i.e., a decrease of 0.1%. It indicates that only using a top-down strategy with only the masks as guidance is insufficient to fully train the network to identify within-image co-saliency.

We can also see that training U-Net with data augmentation improves its accuracy by 0.8%, which is not a significant improvement. However, when the RFM layer is added to the training, both online and offline, the performance of co-saliency detection improves significantly. This is because the RFM provides a bottom-up way to guide the network training and make it sensitive to co-salient regions.

Meanwhile, Table 1 shows that the results of the online version RFM are better than those of the offline version, which validates the effectiveness of the proposed online training sample selection strategy. Specifically, we speculate that there may be two reasons. First, the introduction of additional supervisory signals enhances the supervisory effect on the feature-sensitive layer. Second, the feature vectors generated by the sample region recommended by the RPN network introduce less noise than the randomly generated sample region. This helps to better learn the characteristics of the feature space.

Additionally, comparing the results in the last second and the last third columns, we can see that data augmentation improves performance significantly. This is because data augmentation can help train the RFM module more effectively.

It can be seen from Figure 6(a) that too large or too small loss weights will result in poor performance. When larger weights ($\alpha > 1$) are set to the decoder block, lower F-measure can be observed on the SDCS dataset and our dataset, which indicates that allocating too much weight for the decoder blocks will reduce the effect of RFM. The reason is that when the decoder blocks are given more weight, wrong pixel prediction will receive a heavier punishment. As a result, optimizing the model toward the RFM target is difficult. The loss weight β on the RPN loss has little effect on the model performance, as shown in Figure 6(b). This is because the loss of RPN only counts a small part of the total loss, and RPN can quickly converge during the training and maintain a low loss value. As can be seen from Figure 6(c), the model performance significantly improves when the weight γ on the RFM loss increases from 0.01 to 1. When γ is larger than 1, the performance gradually decreases.

Compared with the ordinary convolution layer, the feature-sensitive layer is more like an attention layer with a supervised signal, which can learn the response to similar appearance object regions through subsequent modules. Figure 7 shows the visualization of the feature heatmaps produced by the baseline

https://engine.scichina.com/doi/10.1007/s11432-022-3686-1

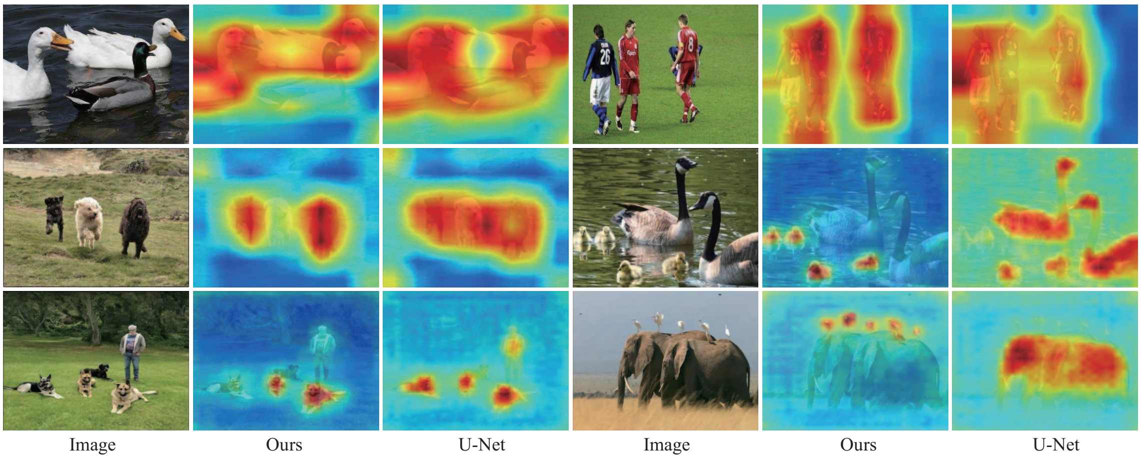

Figure 7 (Color online) The feature heatmaps obtained by the proposed method and the baseline (U-Net) in the same feature scale. The feature-sensitive block of the proposed method can cause the network to pay more attention to salient objects with a similar appearance.

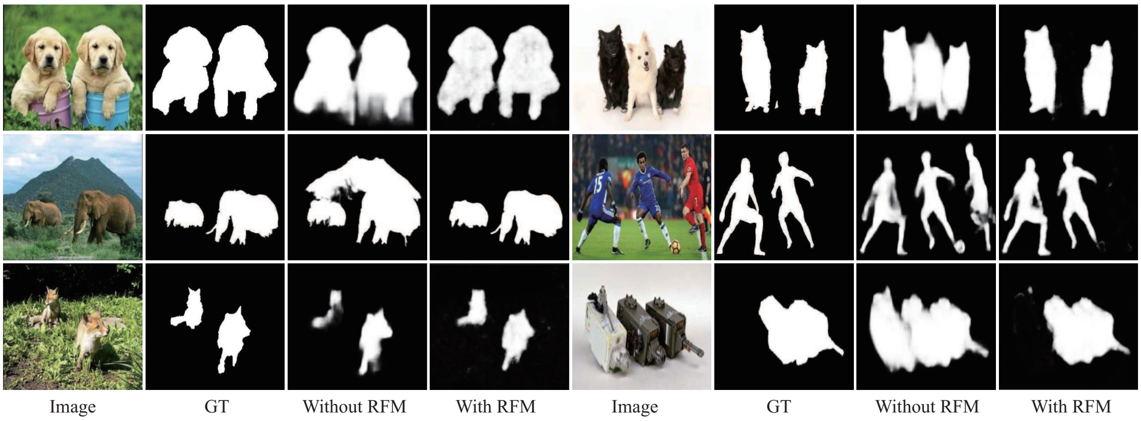

Figure 8 (Color online) Comparison of co-saliency maps generated by our methods with and without RFM on three sample images from two benchmark datasets.

model and our model in the same scale feature layer. The feature-sensitive layer can be seen to emphasize the target areas with similar appearances and suppresses the target areas with different appearances. However, the features of the baseline model respond similarly to different appearance saliency objects.

Figure 8 shows some visual examples of co-saliency maps for ablation studies. We can see that, without the RFM module, the results for the co-salient objects may be distracted or incomplete, and the method cannot distinguish whether the salient objects are co-salient or not. However, when incorporating the RFM module into the framework, the common objects are more highlighted, and the backgrounds are more suppressed. Overall, by merging the RFM module, we can improve the performance both in quantitative and qualitative results.

Figure 9 [80] shows some visual examples of co-saliency maps for ambiguous images. We test some images with ambiguous interpretations in simple scenes to better analyze and understand the role of the RFM module between comparison blocks. The images in the first row of Figure 8 both have two interfering objects with the same color and the same shape. In this situation, the RFM module can select the objects with the highest co-occurrence to be the co-salient objects. The images in the second row have a similar scene, but the interfering objects have different colors and quantities. The RFM module selects the objects with higher co-occurrence and the same color as the co-salient objects. There are many groups of objects with the same number and the same shape and color in the third row. Consequently, there are no domain objects. It is clear that the RFM module is looking for something with the same color or texture, not shape. The reason is that the selected salient region cannot cover the whole object because the positive sample pair augmented the strategy used during the training process. As a result, the model will be unable to take the shape of the object as a factor in the optimization direction.

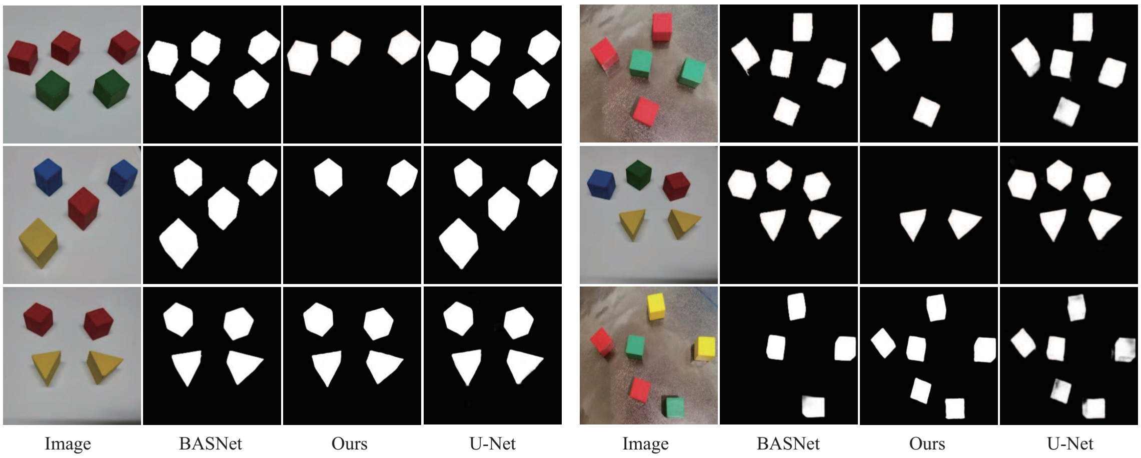

Figure 9 (Color online) Comparison of co-saliency maps generated by U-Net [70], BASNet [80], and our method on ambiguity images.

Table 2 The performance of the proposed method and the comparison methods on two datasets$^{a)}$

Method	WhuCoS Testset I						SDCS dataset					
	F-Measure	Recall	Precision	MAE	S-measure	E-measure	F-Measure	Recall	Precision	MAE	S-measure	E-measure
FT [78]	0.536	0.640	0.461	0.258	0.088	0.722	0.58	0.530	0.6	0.244	0.097	0.692
CWS [12]	0.707	0.764	0.658	0.166	0.132	0.743	0.767	0.7	0.79	0.165	0.16	0.705
U-Net [51]	0.824	0.805	0.846	0.096	0.806	0.934	0.856	0.762	0.889	0.107	0.772	0.851
RFCN [24]	0.852	0.858	0.847	0.083	0.819	0.911	0.883	0.848	0.895	0.083	0.819	0.911
PiCANet [47]	0.856	0.883	0.83	0.086	0.791	0.923	0.854	0.841	0.869	0.088	0.806	0.904
DCL [25]	0.887	0.860	0.916	0.059	0.813	0.876	0.888	0.844	0.902	0.059	0.803	0.907
LFM [22]	0.893	0.881	0.907	0.051	0.765	0.899	0.903	0.85	0.912	0.05	0.79	0.875
GCPANet [81]	0.878	0.856	0.902	0.058	0.826	0.898	0.882	0.842	0.925	0.055	0.834	0.915
MINet [82]	0.89	0.870	0.911	0.052	0.823	0.928	0.892	0.854	**0.933**	0.053	0.852	0.92
BASNet [80]	0.896	0.879	0.914	0.049	0.862	0.936	0.891	0.851	**0.937**	0.051	0.857	0.923
CADC [64]	0.896	**0.909**	0.883	0.048	0.884	0.921	0.896	**0.887**	0.905	**0.047**	**0.884**	**0.935**
GCoNet [83]	**0.901**	0.905	0.897	**0.047**	**0.892**	0.936	0.886	0.861	0.912	0.049	**0.872**	0.926
DCFM [84]	0.899	0.881	**0.918**	**0.047**	**0.895**	**0.943**	0.894	**0.881**	0.908	0.048	0.868	0.928
Ours (offline)	0.892	0.870	**0.917**	0.050	0.839	0.934	**0.907**	0.855	0.924	0.048	0.821	0.932
Ours (online)	**0.904**	**0.917**	0.892	**0.047**	0.851	**0.945**	**0.911**	0.865	0.925	**0.044**	0.848	**0.941**

a) The top three scores are marked in bold with underline, bold, and underlined, respectively.

4.5 Comparison with state-of-the-art methods

The results of our method were compared to those of other representative saliency detectors. Among them, DCL [25], RFCN [24], LFM [22], PiCANet [47], U-Net [70], GCPANet [81], MINet [82], BASNet [80], CADC [64], GCoNet [83] and DCFM [84] are deep learning-based methods, and CWS [12] and FT [78] are traditional saliency detection methods. Among the deep learning-based methods, U-Net is a semantic segmentation method; DCL, RFCN, LFM, PiCANet, GCPANet, MINet, BASNet, CADC, GCoNet, and DCFM are saliency detection methods.

The comparison methods were trained on our dataset by using the recommended parameter settings to ensure a fair comparison. We compute the average performance for these models with respect to F-measure, S-measure, E-measure, MAE, recall, and precision. The results are shown in Table 2.

As shown in Table 2, deep learning-based methods obtain better saliency results than traditional methods. The baseline model (U-Net) obtains worse saliency results than the other deep learning-based methods. Our method, which builds an evaluation part on the skip-layer connection structure of U-Net, achieves higher average recall and better average precision than other saliency methods such as DCL, RFCN, PiCANet, GCPANet, and MINet. This is because these saliency detection methods cannot accurately judge co-saliency objects even though trained with the correct co-saliency masks. The accuracy of boundary pixels is slightly lower when compared to the SOTA method, which focuses on boundary discrimination, such as BASNet, but the overall performance is better. Furthermore, as an end-to-end

https://engine.scichina.com/doi/10.1007/s11432-022-3686-1

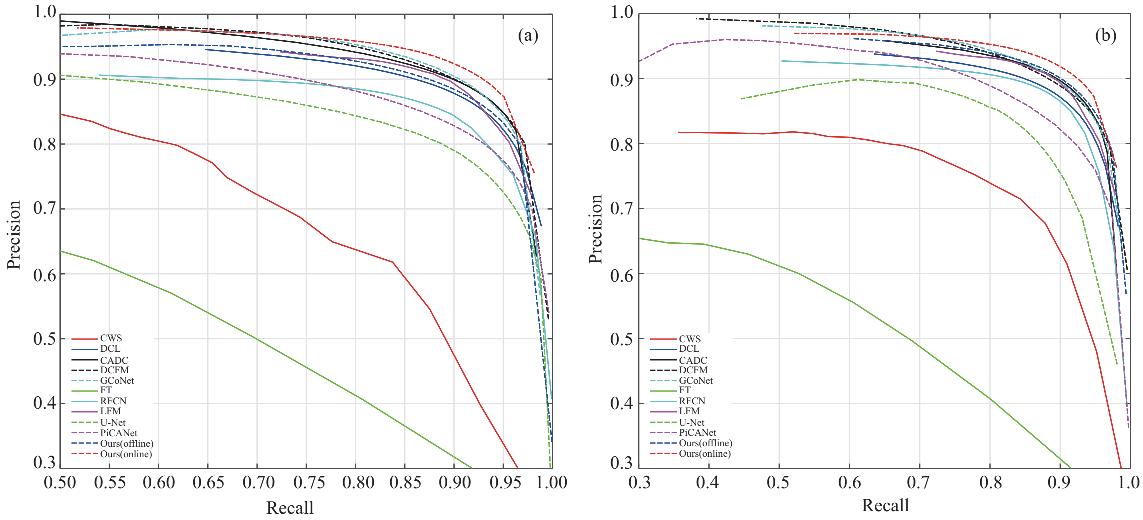

Figure 10 (Color online) Precision-recall curves obtained by the proposed method and the comparison methods on (a) the WhuCoS Testset I and (b) the SDCS dataset.

Table 3 Comparison of the performance on WhuCoS Testset II$^{a)}$

Method	F-Measure	Recall	Precision	MAE	S-measure	E-measure
FT [78] (CVPR'09)	0.418	0.547	0.339	0.361	0.063	0.672
CWS [12] (TIP'13)	0.664	0.709	0.626	0.213	0.106	0.691
U-Net [51] (MICCAI'15)	0.788	0.733	0.851	0.129	0.742	0.879
RFCN [24] (ECCV'16)	0.827	0.781	0.879	0.097	0.815	0.902
PiCANet [47] (CVPR'18)	0.822	0.80	0.845	0.092	0.793	0.884
DCL [25] (CVPR'16)	0.841	0.816	0.87	0.08	0.82	0.869
LFM [22] (AAAI'18)	0.892	0.878	0.906	0.052	0.786	0.913
BASNet [80] (CVPR'19)	0.874	0.835	**0.916**	0.062	0.844	0.926
GCPANet [81] (AAAI'20)	0.855	0.826	0.886	0.076	0.824	0.891
MINet [82] (CVPR'20)	0.860	0.821	0.903	0.074	0.832	0.908
CADC [64] (ICCV'21)	0.876	0.861	0.892	0.057	0.866	0.924
GCoNet [83] (CVPR'21)	0.879	0.855	0.905	0.055	**0.874**	0.921
DCFM [84] (CVPR'22)	0.896	0.869	**0.924**	0.054	**0.872**	**0.937**
Ours (offline)	**0.903**	**0.894**	0.912	**0.048**	0.839	0.927
Ours (online)	**0.908**	**0.915**	0.901	**0.046**	0.856	**0.931**

a) The top three scores are marked in bold with underline, bold, and underlined, respectively.

framework, our method has a faster speed than the LFM and a higher average precision. Compared with the SOTA methods in co-saliency detection, e.g., DCFM, CADC, GCoNet, our model achieved higher performance in most metrics in the within-image co-saliency detection task. The possible reason is that the proposed network can make better use of the consensus between objects in a single image.

For quantitative comparison, the precision-recall curves obtained on the SDCS dataset are plotted in Figure 10. It can be observed that the proposed method has the best performance. Then, the online version of the proposed method, which uses an online training strategy, outperforms the offline version, which is consistent with the results in Table 2. We also evaluate the performance of these methods on Testset II of the WhuCoS dataset. Table 3 shows the results. We can see that the proposed method obtained a higher performance over the comparison methods that is more significant than that on Testset I. As Testset II is more challenging than Testset I, it simply indicates that the proposed method is better at detecting the co-saliency of challenge samples.

4.6 Visual comparison

We also visually compare the results obtained by our method and five comparison methods on several sample images, which are shown in Figure 11. Some images are simple, while others are challenging. According to the results, the U-Net structure can save many salient details, which are helpful in producing

https://engine.scichina.com/doi/10.1007/s11432-022-3686-1

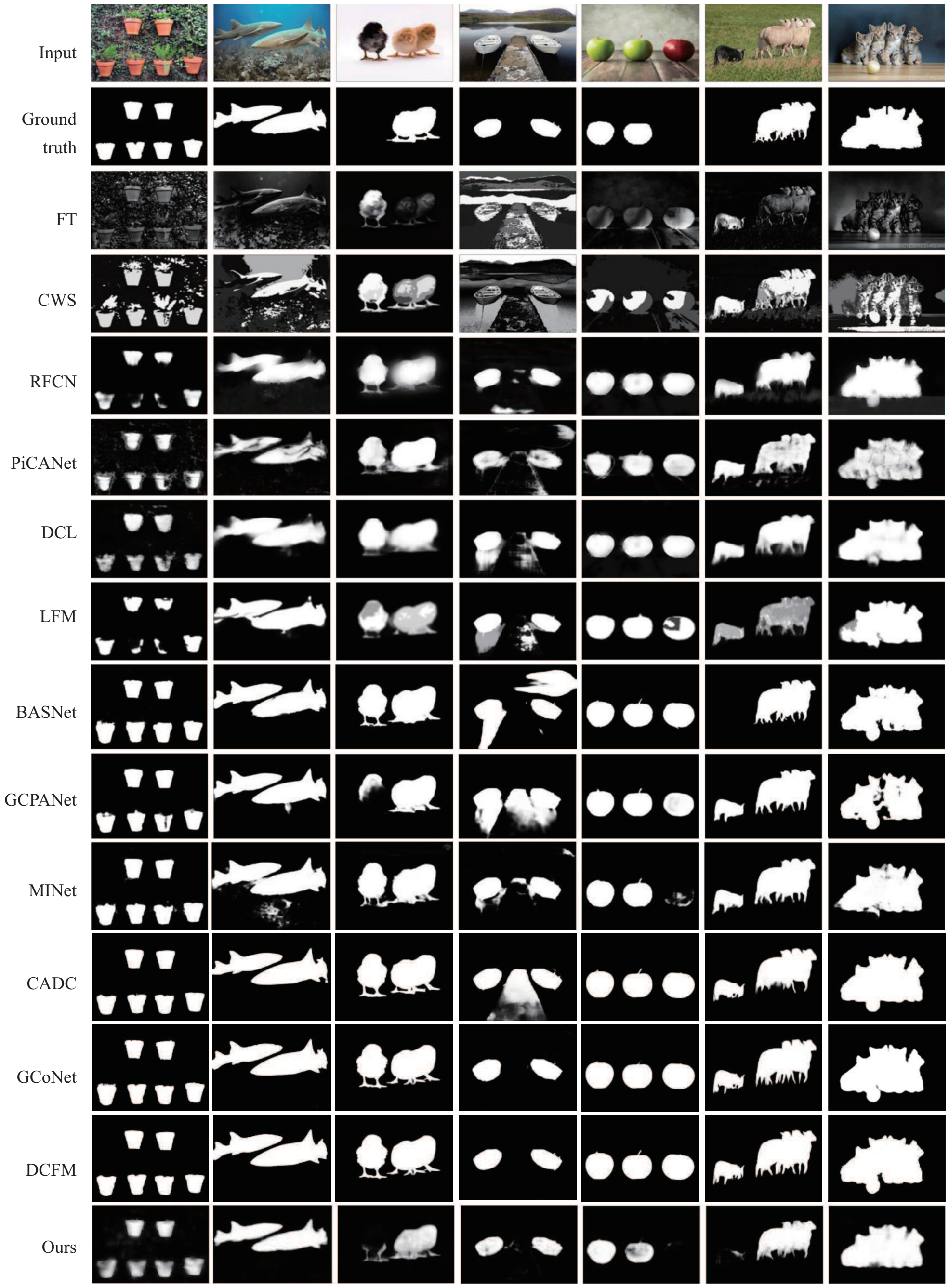

Figure 11 (Color online) Visual comparison of the results obtained by different methods.

more accurate edges in the segmentation task. When compared with other methods in hard images, general saliency detectors based on deep learning cannot judge co-saliency objects using only masks as the supervision information. The high-level hidden layers were trained to learn the mapping of the pure saliency sensitivity to co-saliency objects sensitivity, which leads to better results. Figure 8 visually shows the role of RFM in our method.

5 Conclusion

In this study, a unified end-to-end network for within-image co-saliency detection was proposed. It combined both top-down and bottom-up strategies. Particularly, an encoder-decoder net was used for co-saliency map prediction in a top-down manner, and an RPN and an RFM were used to guide the model to be sensitive to co-salient regions in a bottom-up manner. An online training sample selection algorithm was presented to enhance the performance of the proposed network. A new dataset WhuCoS was constructed for within-image co-saliency detection. It contains 2019 natural images, over 300 categories of daily necessities, and 7000 salient object instances. The experimental results on two target datasets showed that the proposed method achieved state-of-the-art accuracy while running at a speed of 28 fps, and the ablation study validated the effectiveness of the RPN and RFM modules.

Acknowledgements This work was supported by Key Research and Development Program of Hubei Province (Grant No. 2020BAB018), National Natural Science Foundation of China (Grant No. 62171324), and National Key R&D Program of China (Grant No. 2022YFF0901902).

References

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

2 Huang T J, Tian Y H, Li J, et al. Salient region detection and segmentation for general object recognition and image understanding. Sci China Inf Sci, 2011, 54: 2461–2470

3 Goferman S, Zelnik-Manor L, Tal A. Context-aware saliency detection. IEEE Trans Pattern Anal Mach Intell, 2011, 34: 1915–1926

4 Cheng M M, Mitra N J, Huang X, et al. Global contrast based salient region detection. IEEE Trans Pattern Anal Mach Intell, 2014, 37: 569–582

5 Li Z Q, Fang T, Huo H. A saliency model based on wavelet transform and visual attention. Sci China Inf Sci, 2010, 53: 738–751

6 Huang Z Y, He F Z, Cai X T, et al. Efficient random saliency map detection. Sci China Inf Sci, 2011, 54: 1207–1217

7 Li Q N, Li Y D, Lang C Y. Salient object detection with side information. Sci China Inf Sci, 2020, 63: 189202

8 Wang W, Shen J, Shao L. Video salient object detection via fully convolutional networks. IEEE Trans Image Process, 2018, 27: 38–49

9 Zhu W, Liang S, Wei Y, et al. Saliency optimization from robust background detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014. 2814–2821

10 Liu Y, Li X Q, Wang L, et al. Interpolation-tuned salient region detection. Sci China Inf Sci, 2014, 57: 012104

11 Piao Y, Jiang Y, Zhang M, et al. PANet: patch-aware network for light field salient object detection. IEEE Trans Cybern, 2023, 53: 379–391

12 Fu H, Cao X, Tu Z. Cluster-based co-saliency detection. IEEE Trans Image Process, 2013, 22: 3766–3778

13 Cao X, Tao Z, Zhang B, et al. Self-adaptively weighted co-saliency detection via rank constraint. IEEE Trans Image Processing, 2014, 23: 4175–4186

14 Huang R, Feng W, Sun J. Color feature reinforcement for cosaliency detection without single saliency residuals. IEEE Signal Process Lett, 2017, 24: 569–573

15 Cong R, Lei J, Fu H, et al. An iterative co-saliency framework for RGBD images. IEEE Trans Cybern, 2017, 49: 233–246

16 Wei L, Zhao S, Bourahla O E F, et al. Group-wise deep co-saliency detection. 2017. ArXiv:1707.07381

17 Guo F, Wang W, Shen J, et al. Video saliency detection using object proposals. IEEE Trans Cybern, 2017, 48: 3159–3170

18 Zou Q, Ni L, Wang Q, et al. Local pattern collocations using regional co-occurrence factorization. IEEE Trans Multimedia, 2017, 19: 492–505

19 Shi B, Bai X, Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans Pattern Anal Mach Intell, 2016, 39: 2298–2304

20 Liao M, Shi B, Bai X. TextBoxes++: a single-shot oriented scene text detector. IEEE Trans Image Process, 2018, 27: 3676–3690

21 Huang T T, Xu Y C, Bai S, et al. Feature context learning for human parsing. Sci China Inf Sci, 2019, 62: 220101

22 Yu H, Zheng K, Fang J, et al. Co-saliency detection within a single image. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2018

23 Zitnick C L, Dollár P. Edge boxes: locating object proposals from edges. In: Proceedings of the 13th European Conference on Computer Vision, Zurich, 2014. 391–405

24 Wang L, Wang L, Lu H, et al. Saliency detection with recurrent fully convolutional networks. In: Proceedings of the 14th European Conference on Computer Vision, Amsterdam, 2016. 825–841

25 Li G, Yu Y. Deep contrast learning for salient object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 478–487

26 Zhao T, Wu X. Pyramid feature attention network for saliency detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 3085–3094

27 Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of Advances in Neural Information Processing Systems, 2015. 28

28 Schroff F, Kalenichenko D, Philbin J. FaceNet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 815–823

29 Bell S, Bala K. Learning visual similarity for product design with convolutional neural networks. ACM Trans Graph, 2015, 34: 1–10

30 Cong R, Zhang Y, Fang L, et al. RRNet: relational reasoning network with parallel multiscale attention for salient object detection in optical remote sensing images. IEEE Trans Geosci Remote Sens, 2022, 60: 1–11

31 Cong R, Yang N, Li C, et al. Global-and-local collaborative learning for co-salient object detection. IEEE Trans Cybern, 2023, 53: 1920–1931

32 Han J, Zhang D, Wen S, et al. Two-stage learning to predict human eye fixations via SDAEs. IEEE Trans Cybern, 2015, 46: 487–498

33 Bylinskii Z, Recasens A, Borji A, et al. Where should saliency models look next? In: Proceedings of the 14th European Conference on Computer Vision, Amsterdam, 2016. 809–824

34 Borji A, Cheng M M, Jiang H, et al. Salient object detection: a benchmark. IEEE Trans Image Process, 2015, 24: 5706–5722

35 Zhou Y, Huo S, Xiang W, et al. Semi-supervised salient object detection using a linear feedback control system model. IEEE Trans Cybern, 2018, 49: 1173–1185

36 Itti L, Koch C, Niebur E. A model of saliency-based visual attention for rapid scene analysis. IEEE Trans Pattern Anal Machine Intell, 1998, 20: 1254–1259

37 Liu T, Yuan Z J, Sun J, et al. Learning to detect a salient object. IEEE Trans Pattern Anal Mach Intell, 2010, 33: 353–367

38 Wei Y, Wen F, Zhu W, et al. Geodesic saliency using background priors. In: Proceedings of the 12th European Conference on Computer Vision, Florence, 2012. 29–42

39 Borji A. Boosting bottom-up and top-down visual features for saliency estimation. In: Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012. 438–445

40 Li J, Pan Z, Liu Q, et al. Complementarity-aware attention network for salient object detection. IEEE Trans Cybern, 2020, 52: 873–886

41 Fang Y, Lin W, Lee B S, et al. Bottom-up saliency detection model based on human visual sensitivity and amplitude spectrum. IEEE Trans Multimedia, 2011, 14: 187–198

42 Zhao R, Ouyang W, Li H, et al. Saliency detection by multi-context deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 1265–1274

43 Zhang J, Sclaroff S, Lin Z, et al. Unconstrained salient object detection via proposal subset optimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 5733–5742

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

45 Kim J, Pavlovic V. A shape-based approach for salient object detection using deep learning. In: Proceedings of the 14th European Conference on Computer Vision, Amsterdam, 2016. 455–470

46 Hou Q, Cheng M M, Hu X, et al. Deeply supervised salient object detection with short connections. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 3203–3212

47 Liu N, Han J, Yang M H. PiCANet: learning pixel-wise contextual attention for saliency detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 3089–3098

48 Chen S, Tan X, Wang B, et al. Reverse attention for salient object detection. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 234–250

49 Luo Z, Mishra A, Achkar A, et al. Non-local deep features for salient object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 6609–6617

50 Xie S, Tu Z. Holistically-nested edge detection. In: Proceedings of the IEEE International Conference on Computer Vision, 2015. 1395–1403

51 Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, 2015. 234–241

52 Ye L, Liu Z, Li L, et al. Salient object segmentation via effective integration of saliency and objectness. IEEE Trans Multimedia, 2017, 19: 1742–1756

53 Cong R, Lei J, Fu H, et al. Review of visual saliency detection with comprehensive information. IEEE Trans Circuits Syst Video Technol, 2018, 29: 2941–2959

54 Zhang Q, Cong R, Li C, et al. Dense attention fluid network for salient object detection in optical remote sensing images. IEEE Trans Image Process, 2020, 30: 1305–1317

55 Li C, Cong R, Hou J, et al. Nested network with two-stream pyramid for salient object detection in optical remote sensing images. IEEE Trans Geosci Remote Sens, 2019, 57: 9156–9166

56 Chen Z, Cong R, Xu Q, et al. DPANet: depth potentiality-aware gated attention network for RGB-D salient object detection. IEEE Trans Image Process, 2020, 30: 7012–7024

57 Cong R, Lei J, Fu H, et al. Going from RGB to RGBD saliency: a depth-guided transformation model. IEEE Trans Cybern, 2019, 50: 3627–3639

58 Fang Y, Wang Z, Lin W, et al. Video saliency incorporating spatiotemporal cues and uncertainty weighting. IEEE Trans Image Process, 2014, 23: 3910–3921

59 Li Y, Sheng B, Ma L, et al. Temporally coherent video saliency using regional dynamic contrast. IEEE Trans Circuits Syst Video Technol, 2013, 23: 2067–2076

60 Chen H T. Preattentive co-saliency detection. In: Proceedings of 2010 IEEE International Conference on Image Processing, 2010. 1117–1120

61 Li H, Ngan K N. A co-saliency model of image pairs. IEEE Trans Image Process, 2011, 20: 3365–3375

62 Zhang D, Meng D, Han J. Co-saliency detection via a self-paced multiple-instance learning framework. IEEE Trans Pattern Anal Mach Intell, 2016, 39: 865–878

63 Zhang K, Dong M, Liu B, et al. DeepACG: co-saliency detection via semantic-aware contrast Gromov-Wasserstein distance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 13703–13712

64 Zhang N, Han J, Liu N, et al. Summarize and search: learning consensus-aware dynamic convolution for co-saliency detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 4167–4176

65 Tang L, Li B, Kuang S, et al. Re-thinking the relations in co-saliency detection. IEEE Trans Circuits Syst Video Technol, 2022, 32: 5453–5466

66 Ren G, Dai T, Stathaki T. Adaptive intra-group aggregation for co-saliency detection. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022. 2520–2524

67 Yu H, Zheng K, Fang J, et al. A new method and benchmark for detecting co-saliency within a single image. IEEE Trans Multimedia, 2020, 22: 3051–3063

68 Song S, Yu H, Miao Z, et al. An easy-to-hard learning strategy for within-image co-saliency detection. Neurocomputing, 2019, 358: 166–176

69 Guo Y, Liu Y, Georgiou T, et al. A review of semantic segmentation using deep neural networks. Int J Multimed Info Retr, 2018, 7: 87–93

70 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv:1409.1556

71 Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. In: Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009. 248–255

72 He K, Gkioxari G, Dollár P, et al. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 2961–2969

73 Shrivastava A, Gupta A, Girshick R. Training region-based object detectors with online hard example mining. In: Proceedings

of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 761–769

74 Zou Q, Zhang Z, Li Q, et al. DeepCrack: learning hierarchical convolutional features for crack detection. IEEE Trans Image Process, 2018, 28: 1498–1512

75 Girshick R. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 2015. 1440–1448

76 Batra D, Kowdle A, Parikh D, et al. iCoseg: interactive co-segmentation with intelligent scribble guidance. In: Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010. 3169–3176

77 Li Q Q, Zou Q, Ma D, et al. Dating ancient paintings of Mogao Grottoes using deeply learnt visual codes. Sci China Inf Sci, 2018, 61: 092105

78 Achanta R, Hemami S, Estrada F, et al. Frequency-tuned salient region detection. In: Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009. 1597–1604

79 Fan D P, Gong C, Cao Y, et al. Enhanced-alignment measure for binary foreground map evaluation. 2018. ArXiv:1805.10421

80 Qin X, Zhang Z, Huang C, et al. BASNet: boundary-aware salient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 7479–7489

81 Chen Z, Xu Q, Cong R, et al. Global context-aware progressive aggregation network for salient object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 10599–10606

82 Pang Y, Zhao X, Zhang L, et al. Multi-scale interactive network for salient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 9413–9422

83 Fan Q, Fan D P, Fu H, et al. Group collaborative learning for co-salient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 12288–12298

84 Yu S, Xiao J, Zhang B, et al. Democracy does matter: comprehensive feature mining for co-salient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 979–988