Weakly-Supervised Salient Object Detection with Saliency Bounding Boxes

Yuxuan Liu, Pengjie Wang, Ying Cao, Zijian Liang, and Rynson W.H. Lau

***Abstract*—In this paper, we propose a novel form of weak supervision for salient object detection (SOD) based on saliency bounding boxes, which are minimum rectangular boxes enclosing the salient objects. Based on this idea, we propose a novel weakly-supervised SOD method, by predicting pixel-level pseudo ground truth saliency maps from just saliency bounding boxes. Our method first takes advantage of the unsupervised SOD methods to generate initial saliency maps and addresses the over/under prediction problems, to obtain the initial pseudo ground truth saliency maps. We then iteratively refine the initial pseudo ground truth by learning a multi-task map refinement network with saliency bounding boxes. Finally, the final pseudo saliency maps are used to supervise the training of a salient object detector. Experimental results show that our method outperforms state-of-the-art weakly-supervised methods.**

***Index Terms*—Saliency bounding boxes, salient object detection, weak supervision.**

I. Introduction

The aim of salient object detection (SOD) is to detect the regions in an image that attract human attention. It has important applications in a variety of vision tasks, e.g., object recognition, image retrieval and object tracking. The recent advance in deep convolutional neural networks has significantly improved the performance of salient object detection. However, this progress is mostly driven by a large number of pixel-level labels. According to [1], annotating pixel-level labels requires considerable human efforts. Hence, it is essential to find a way to alleviate such annotation overheads.

Seeking for weak supervisory signals is an active and promising research direction, since it can substantially reduce human supervision and enable models to scale much easier than fully-supervised methods. Previous weakly-supervised SOD methods focus on using image-level category labels as weak supervision [2] [3]. However, this type of methods often suffer from inaccurate localization of saliency objects and missing of some salient object parts. This is mainly because high-level category labels can only guide models to focus on the most discriminative regions. They do not provide any location information about the salient objects. As shown in the first row of Figure 1(c), the saliency map obtained from the model supervised by category labels only highlights the dog head, because heads are already discriminative enough to classify the image as dog. Compared with image-level labels, bounding boxes are able to provide more accurate information of where the foreground objects are, which can help improve detection performance greatly, as shown in the first row of Figure 1(d).

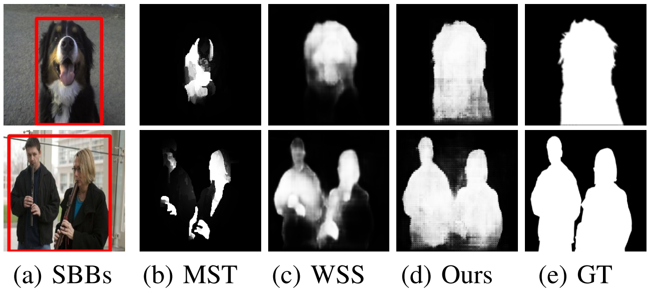


Fig. 1. (a) Images with saliency bounding boxes (SBBs) that are marked in red. (b) Saliency maps from an unsupervised method [2]. (c) Saliency maps from a model using image-level category labels as supervision [2]. (d) Saliency maps by our method using saliency bounding boxes as supervision. (e) Ground truth saliency maps.

In this paper, instead of using image-level category labels, we propose a new form of weak supervision using saliency bounding boxes (SBBs) for salient object detection. In particular, we first obtain the initial saliency maps by fusing the results from state-of-the-art unsupervised SOD methods, and adjust these initial maps by fixing over-coverage and under-coverage issues, guided by the saliency bounding boxes. We then use the adjusted maps and saliency bounding boxes to train a multi-task network to estimate both saliency maps and saliency bounding boxes, and the trained network to refine the saliency maps in an iterative manner. Finally, the refined saliency maps are used as pseudo ground truth for training a salient object detection model. To our knowledge, we are the first to explore using bounding boxes to supervise the learning of a SOD model. In our method, saliency bounding boxes not only reduce the cost of labeling, they also provide accurate location clues of objects compared with using category labels. Experimental results show that our method outperforms all weakly-supervised.

Our main contributions can be summarized as follows:

- We propose a novel form of supervision, saliency bounding boxes (SBBs), for weakly-supervised salient object detection. Our method is supervised only by saliency bounding boxes, without the need for pixel-level salient object labels.

Y. Liu is with the Department of Computer Science, Dalian Minzu University, China (e-mail: liuyuxuan_mail@foxmail.com)

P. Wang is with the Department of Computer Science, Dalian Minzu University, China (e-mail: pengjiewang@gmail.com). He is the corresponding author.

Y. Cao is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: caoying59@gmail.com)

Z. Liang is with the Department of Computer Science, Dalian Minzu University, China (e-mail: liang_zijian@foxmail.com)

R.W.H. Lau is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: Rynson.Lau@cityu.edu.hk).

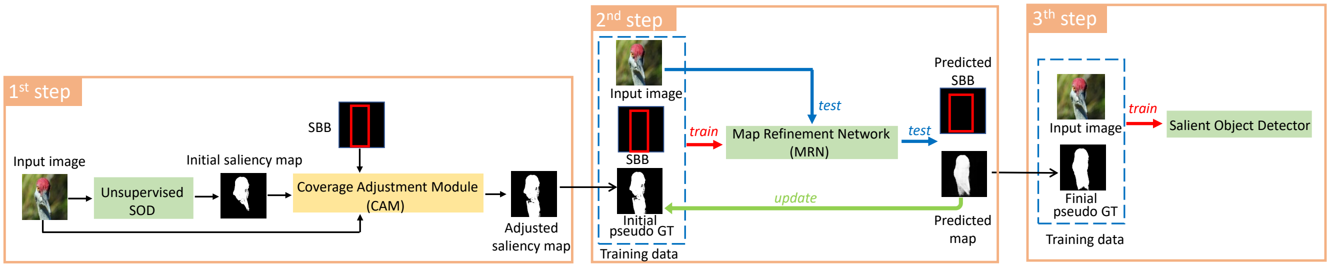


Fig. 2. The pipeline of the proposed method. Given initial saliency maps obtained from multiple unsupervised saliency object detection methods, we first address the over/under prediction problems with a coverage adjustment module guided by the saliency bounding boxes (SBBs). We then use the adjusted saliency maps as initial pseudo ground truth saliency maps and train a map refinement network to refine the saliency maps. Finally, the high-quality saliency maps from the previous step serve as pseudo ground truth maps to train a salient object detector.

- We propose a learning framework that, under the supervision of saliency bounding boxes, can produce high-quality pixel-level pseudo ground truth saliency maps for training a salient object detector.
- Experimental results demonstrate that our method outperforms state-of-the-art weakly-supervised methods.

II. Related Works

A. Unsupervised Salient Object Detection

Traditional unsupervised salient object detection methods are always driven by some hand-crafted features [4] [5] [6]. [7] proposed an efficient algorithm based on minimum barrier distance to detect saliency regions, and an extended version to further improve the performance based on color whitening. [8] proposed a model based on the minimum spanning tree, which enables real-time detection. Human annotations are not needed for these methods. [9] fused saliency maps derived from unsupervised methods within a deep learning framework. [10] proposed to update a saliency detection model through learning from multiple noisy labels generated by unsupervised saliency methods with handcrafted features. In [11], labels with noise are refined in a self-supervision manner, which are then used as pseudo labels for training a saliency detection network.

B. Fully-Supervised Salient Object Detection

Deep convolutional neural networks, especially fully convolutional neural networks, bring in the possibility of training salient object detection models end-to-end with pixel-level full supervision [12] [13] [14] [15] [16] [17]. [18] improved the boundary accuracy of detected salient objects by learning to solve contour and edge detection tasks jointly in salient object detection. Although these methods achieve impressive performances, they all depend on pixel-level labels, which are expensive to collect.

C. Weakly-Supervised Salient Object Detection

To reduce labeling cost, CNN-based weakly-supervised methods are very popular in recent years. These methods are often driven by higher-level labels that are much cheaper to collect, such as object categories. [2] designed a foreground inference network (FIN) and trained it on object category labels jointly with a fully convolutional network whose output saliency maps are used to fine-tune the FIN. [3] trained a multi-task fully convolutional network using saliency maps from unsupervised methods and category labels. They derived saliency annotations using the class activation maps from the network and unsupervised saliency maps. [19] proposed to leverage multiple sources of supervision for saliency object detection. They trained two networks for object recognition and image caption generation. Two attention losses are used to supervise the learning of the networks, which force the model to find salient regions to train a new salient object detector.

The above methods are supervised by image-level category labels or image captions, and usually suffer from inaccurate detection of salient objects. In contrast to these methods, our method is supervised by saliency bounding boxes, which provide more specific location cues of the salient objects and thus enable more accurate saliency localization.

D. Object Proposals for Salient Object Detection

There are also salient object detection works based on generating object proposals [20] [21]. [20] proposed a novel subset optimization framework to help generate noisy object proposals. [21] first predicted a saliency region, which is then used to derive the object proposal. Compared to these works, we have two key differences. First, the bounding boxes generated in these two works contain a lot of noise. In our paper, saliency bounding boxes are obtained through annotations and, therefore, are much more accurate. Second, in [20] [21], the tasks of predicting saliency maps and object proposals are solved in a sequential manner. Unlike these works, we propose a multi-task network (i.e., Map Refinement Network) to learn both tasks (i.e., salient bounding box and salient map predictions) simultaneously to achieve more accurate prediction.

E. Weakly-Supervised Segmentation with Bounding Boxes

There have been some interesting attempts to supervise semantic segmentation models with semantic bounding boxes (i.e., bounding boxes and category labels on objects of interest) [22] [23]. While sharing a similar high-level idea in leveraging bounding boxes as a weak supervision signal, our method differs from these methods in two aspects. First, our method is supervised by bounding boxes only, without using object

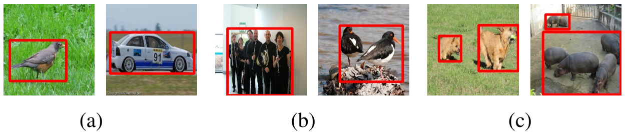


Fig. 3. Saliency bounding boxes (marked in red). The annotations of the dataset are classified into 3 cases: (a) a single box with a single object, (b) a single box with multiple objects, (c) multiple boxes.

category labels. Second, our saliency bounding boxes focus only on the salient objects, rather than all semantic objects in an image.

III. The Proposed Method

Figure 2 shows the pipeline of the proposed method, which includes three main steps. First, we fuse saliency prediction maps from multiple unsupervised methods to obtain the initial per-pixel saliency maps. We then improve the initial saliency maps by adjusting under/over coverage with a coverage adjustment module. Second, we train a multi-task learning network with supervision from the adjusted saliency maps and saliency bounding boxes as learning targets, and use them to iteratively refine the saliency maps. Finally, the refined per-pixel saliency maps are used to train a salient object detector. In this section, we first introduce the definition of saliency bounding boxes and then elaborate on each of the three main steps.

A. Saliency Bounding Boxes

We design a novel form of supervision for weakly supervision, namely saliency bounding boxes (SBBs). Each saliency bounding box covers a saliency region and is labeled based on the following rules:

- A saliency bounding box is a rectangular box that contains at least one salient object.
- When an image has multiple salient regions, there should be no overlap between any two bounding boxes. Two overlapped bounding boxes would be merged into one saliency bounding box that encloses these two boxes.

As shown in Figure 3, we divide the annotations of the dataset into 3 cases: (a) a single box with a single object, (b) a single box with multiple objects, (c) multiple boxes. Note that our annotations are different from those used in existing object detection datasets [24] [25] [26], where one bounding box is assigned to one specific object. According to our two rules stated above, one saliency bounding box may include multiple objects. Salient pixels only appear inside the saliency bounding boxes, while the pixels outside the saliency bounding boxes belong to the background.

B. Saliency Map Generation

Given a set of images, we first obtain the initial saliency maps from many unsupervised methods [27] [8] [7]. With the help of saliency bounding boxes, we can then improve the initial saliency maps by addressing any potential over-coverage and under-coverage problems.

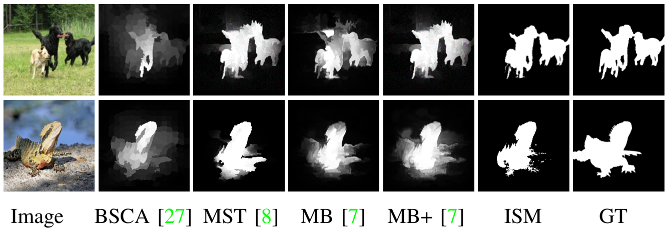


Fig. 4. Generating the initial saliency maps (ISMs) by combining the outputs of multiple unsupervised methods (BSCA [27], MST [8], MB [7], MB+ [7]).

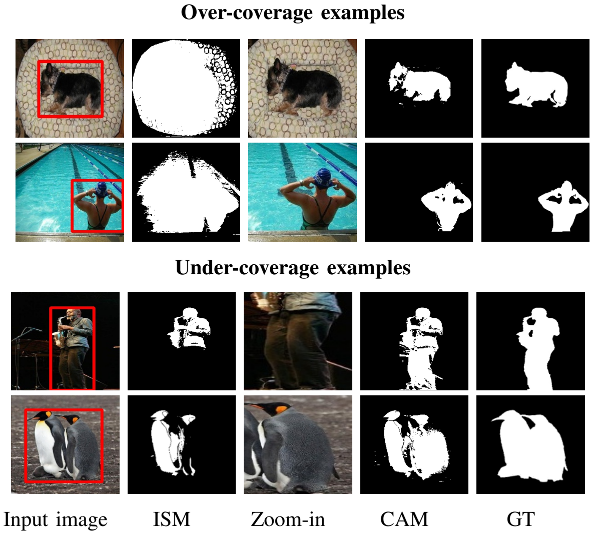


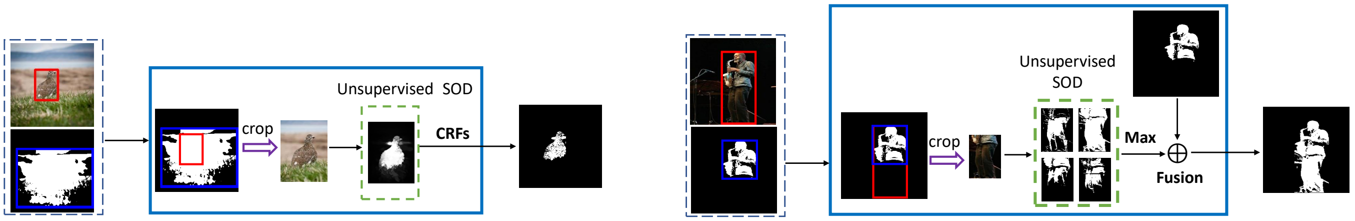
Fig. 5. Saliency map adjustment for over- and under-coverage cases. An initial saliency map (ISM) is obtained by combining the saliency maps from four unsupervised methods. The over-coverage examples indicate over-detection, i.e., over-detecting part of the backgrounds as the salient objects. The under-coverage examples indicate under-detection, i.e., missing parts of the salient objects. With the help from ground truth saliency bounding boxes, we eliminate non-related parts to obtain zoom-in images. Zoom-in shows the cropped regions that the coverage adjustment processing should attend to. CAM shows the results after coverage adjustment.

1) Initial Saliency Maps (ISMs): As shown in Figure 4, we use four state-of-the-art unsupervised methods, including BSCA [27], MST [8], MB [7], and MB+ [7] (an improved version of MB) to infer four saliency maps, denoted as M_1, M_2, M_3, M_4, respectively. We fuse the four maps from these four methods to obtain an initial saliency map I. The saliency value at pixel (m, n) in I is defined as:

$$I(m,n) = \begin{cases} 1, & c \geq 2, \\ \\ 0, & c < 2. \end{cases} \qquad c = \sum_{i=1}^{4} CRF(M_i(m,n)), \quad (1)$$

where $CRF()$ is the conditional random fields (CRFs) [28].

2) Over-coverage and Under-coverage: Unsupervised salient object detection methods typically suffer from over-coverage and under-coverage problems. As shown in Figure 5, we define over-coverage as over-detecting part of the background as belonging to the salient object, and this over-detected background fully surrounds the salient object. We define under-coverage as missing some parts of the salient object. To address these two problems, we propose a coverage adjustment module, which first determines if an input saliency map has over-coverage or under-coverage problems based on

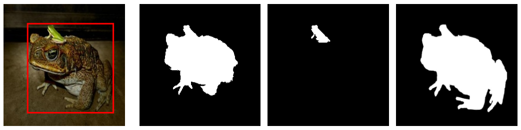


(a) Over-coverage sub-module (b) Under-coverage sub-module

Fig. 6. The coverage adjustment module (CAM) consists of an over-coverage sub-module and an under-coverage sub-module. For both sub-modules, the inputs are the input image together with a saliency bounding box (SBB, in red) and an initial saliency map together with a tight bounding box (in blue), and the output is an adjusted saliency map.

a coverage check, and then uses an over-coverage sub-module and an under-coverage sub-module to adjust the saliency map accordingly. Over-coverage often happens in images that contain small objects or low-contrast objects relative to the background. Under-coverage often happens in images where part of the salient object is mis-classified as the background.

For obtaining the initial saliency map, we perform unsupervised methods on the whole image, not inside the salient bounding box. This is because directly applying the unsupervised methods to the saliency bounding box may lead to an under-coverage problem, as shown in Figure 7(c), given unsupervised methods usually rely on the contrast between foreground and background regions to extract the salient foreground. Therefore, we chose to perform unsupervised methods on the whole image, to avoid this problem, as shown in Figure 7(b).



(a) SBB (b) Whole (c) Region (d) GT

Fig. 7. Results of applying the unsupervised methods to the whole image and a saliency bounding box. (a) An image with saliency bounding boxes (SBBs) that are marked in red. (b) Performing the unsupervised methods on the whole image. (c) Performing the unsupervised methods inside the salient bounding box. (d) Ground truth saliency map.

Coverage checking: We first introduce a tight bounding box (i.e., the blue box in Figure 6) that tightly encloses the foreground pixels in the initial saliency map. This tight bounding box is used to compare with the ground truth bounding box to determine whether the initial saliency map is over-coverage or under-coverage. We represent the ground truth saliency bounding box by (x, y, w, h), where (x, y), w and h are its upper-left corner coordinates, width and height, respectively. The tight bounding box is similarly represented by $(\hat{x}, \hat{y}, \hat{w}, \hat{h})$. We compute the distances between the corresponding boundaries of the two boxes as:

$$S = (x - \hat{x}, y - \hat{y}, \hat{x} + \hat{w} - x - w, \hat{y} + \hat{h} - y - h), \qquad (2)$$

where S has four components representing four boundary distances (left, top, right and bottom), respectively. Here, we denote S_i as the i-th component in S, $i \in$ {1,2,3,4}.

An initial saliency map is regarded as over-coverage if the ground truth saliency bounding box is completely enclosed by the tight bounding box and the pixels outside the ground truth bounding box are falsely predicted as foreground, i.e., $S_i > 0, \forall i \in \{1,2,3,4\}$ and $\frac{FP}{TP} > k$. FP represents false positive, while TP represents true positive, according to the initial saliency map and saliency bounding box. (k is set to 1.5 in our implementation.) The initial saliency map is regarded as under-coverage along the i-th boundary if $S_i < d$, where d is set to -30, and under-coverage is detected if $S_i < d, \exists i \in \{1,2,3,4\}$. Since these two problems do not happen at the same time, two sub-modules are used in parallel. Note that if an initial saliency map is considered as neither over-coverage nor under-coverage, we would simply skip this step.

Over-coverage sub-module: As shown in Figure 6(a), the inputs to this sub-module include an image together with a saliency bounding box and an initial saliency map with its tight bounding box around the salient region. To address over-coverage, we propose to "Zoom in" what surrounds the ground truth salient object according to the saliency bounding box, as shown in the Zoom-in column of Figure 5. That is, we crop a new region around the ground truth saliency bounding box and feed it to the unsupervised method for re-prediction. Specifically, we construct the new region by extending the width and height of the ground truth saliency bounding box by a factor of 1.8. The expansion is designed to make the salient object stand out from its local context for better prediction. We then re-predict the saliency map of the new region through the unsupervised method. For this, we only use one unsupervised method MB+ because both BSCA and MST are not suitable for small size images. We then apply CRFs to the new prediction and replace the original initial saliency map with it.

Under-coverage sub-module: As shown in Figure 6(b), similar to the over-coverage sub-module, this module receives an image with its saliency bounding box and an initial saliency map with a tight bounding box around the salient region. To address under-coverage, we propose to "Zoom in" to the missing part according to the saliency bounding box, as shown in the Zoom-in column of Figure 5. Based on the saliency bounding box and tight bounding box, we define a set of complementary bounding boxes that cover the saliency regions missed in the initial saliency map. Specifically, if an initial saliency map is under-coverage along the i-th boundary, we construct a complementary box for it. The four possible

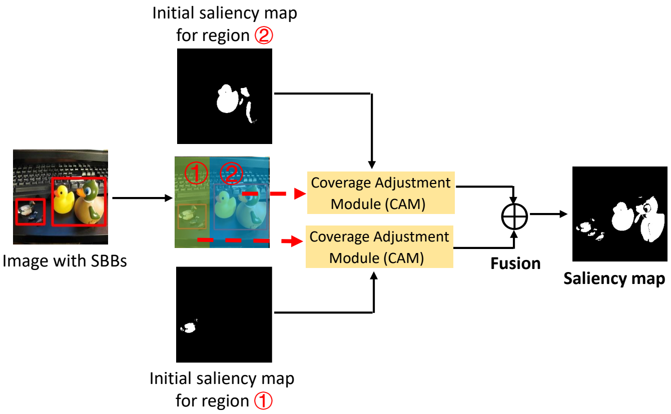


Fig. 8. An example image of having multiple saliency bounding boxes. Regions ① and ② are sent into the coverage adjustment module along with their initial saliency maps. Each region is processed independently. The final saliency map is a fusion of the adjusted saliency maps for the two regions.

complementary boxes (left, top, right, bottom) are defined as:

$$\begin{aligned} B_{\mathrm{l}} &= (x-\beta, y, |S_1|+2\beta, h), \\ B_{\mathrm{t}} &= (x, y-\beta, w, |S_2|+2\beta), \\ B_{\mathrm{r}} &= (\hat{x}+\hat{w}-\beta, y, |S_3|+2\beta, h), \\ B_{\mathrm{b}} &= (x, \hat{y}+\hat{h}-\beta, w, |S_4|+2\beta). \end{aligned} \tag{3}$$

In Eq. 3, we expand each bounding box by a parameter β, which is set to 32. When an initial saliency map is under-coverage along multiple boundaries, we will end up with multiple complementary boxes. As shown in Figure 6, we crop the region under each complementary box, send it to four unsupervised methods (BSCA, MST, MB and MB+) and refine the results with CRFs. In this way, we obtain four saliency region maps. We then select the one with the largest number of salient pixels and combine it with the original initial saliency map to obtain an adjusted saliency map.

Multiple saliency bounding boxes: If an input image has multiple saliency bounding boxes, we consider each of them separately. As shown in Figure 8, we can divide the image into multiple regions. Each region is sent to the coverage adjustment module along with its initial saliency map. The adjusted saliency maps of all the regions are combined together to obtain the complete saliency map.

As the last step of the processing, we designate all the pixels of the saliency map that fall outside the saliency bounding boxes as the background, to eliminate potential false positives in the regions that are not covered by the saliency bounding boxes. Figure 9 shows two examples.

C. Saliency Map Refinement

Based on the pseudo ground truth saliency maps that we have obtained together with the saliency bounding boxes, we propose a map refinement network to jointly estimate the saliency maps and saliency bounding boxes to improve the pseudo ground truth, with an assumption that the two related tasks can benefit each other. As shown in Figure 10, the map refinement network consists of an attention-based feature fusion module and a two-task prediction module. It is guided by saliency bounding boxes. We first elaborate on each of the modules and then introduce our iterative saliency map updating method with the map refinement network.

1) Attention-based Feature Fusion: As shown in Figure 10, the feature extractor of our network is based on ResNet101 [29] by removing the average pooling and fully connected layers. In the last two blocks, Res3 and Res4, we use two dilated convolution layers (with dilation factors of 2 and 5, respectively), instead of the original convolution layers, to obtain larger receptive fields.

We introduce transition layers [30] and attention modules for learning a strong feature representation. A transition layer includes a convolutional layer with a kernel size of 1×1, a BN layer [31], and ReLU nonlinearity. We add three transition layers with compression factors [30] (the number of output channels / the number of input channels) of 1, 0.5 and 0.5. Attention mechanisms can help improve the representation power of CNN networks [32]. To obtain a spatial attention map A_s, we apply max pooling along the channel dimension to the output of the first transition layer T_1. To obtain a channel attention map A_c, we apply channel-wise average pooling to the output of Res3 F_3, as:

$$\begin{aligned} A_s &= sigmoid(MaxPool(T_1)), \\ A_c &= softmax(AvgPool(F_3)). \end{aligned} \tag{4}$$

We then multiply the output of the third transition layer T_3 with A_s element-wise to obtain a new T_3, and multiply the output of the second transition layer T_2 with A_c element-wise

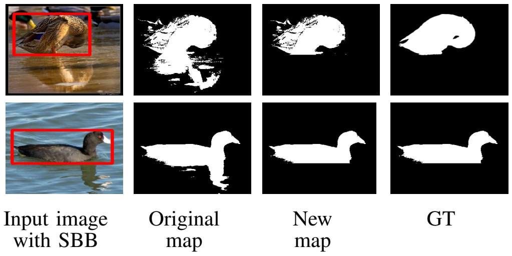


Fig. 9. We obtain a new saliency map by designating all pixels falling outside the saliency bounding box as background.

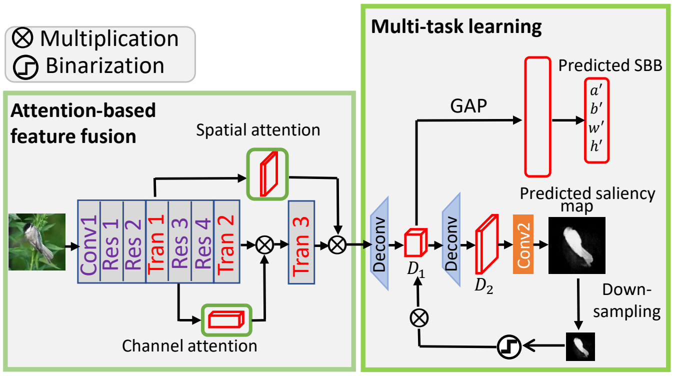


Fig. 10. Map refinement network. Attention-based feature fusion involves two types of attention, spatial attention and channel attention. Multi-task learning predicts a saliency bounding box and a saliency map jointly.

to obtain a new T_2, as:

$$\begin{aligned} T_3 &= A_s \times T_3, \\ T_2 &= A_c \times T_2. \end{aligned} \tag{5}$$

2) Multi-task Learning: The map refinement network performs two tasks, saliency map prediction and saliency bounding box prediction. They share the same feature extractor and thus force the network to learn the features that are useful to the two tasks. In this way, the implicit guidance from the saliency bounding boxes can help improve the predicted saliency maps.

As shown in Figure 10, the feature map from the attention-based feature fusion goes through two deconvolutional layers of kernel size 2×2 and a stride of 2. Each deconvolutional layer is followed by a ReLU activation function and a BN layer. The outputs of the two deconvolution layers are D_1 and D_2. D_2 then goes through a convolutional layer with a kernel size of 1×1 to obtain a predicted saliency map y'_{map}.

We use y'_{map} to activate potential foreground features and suppress potential background features, by multiplying binarized y'_{map} with D_1. The result then goes through a global average pooling layer and a fully connected layer to obtain the predicted saliency bounding box as:

$$y'_{box} = FC(GAP(D_1 \times Bin(Down(y'_{map})))), \tag{6}$$

where $Down()$ represents down-sampling, $Bin()$ represents binarization, $GAP()$ represents global average pooling and $FC()$ represents a fully connected layer with 4 neurons. The first two numbers in y'_{box} represent the upper-left corner coordinates of the predicted bounding box, and the last two numbers are its width and height. In the above process, the predicted saliency map has to conform to the predicted saliency bounding box, which implies that saliency map prediction can be guided by saliency bounding box prediction.

The final loss is the sum of the losses for the two tasks weighted by α, which is set to 0.1 in our experiment:

$$Loss = \alpha \times L1(y'_{box}, y_{box}) + BCE(y'_{map}, y_{map}), \tag{7}$$

where $L1()$ is the smooth L1 loss. $BCE()$ is the binary cross entropy loss. y_{box} and y_{map} are the ground truth saliency bounding box and map, respectively.

In the above saliency bounding box prediction, we only describe the case with only one saliency region in an input image. For images with more than one saliency region, the prediction of saliency bounding boxes is slightly different. In particular, to predict the saliency bounding box for a salient region, we first take the predicted saliency map and mask out all positions that fall within other saliency bounding boxes. We then use the masked saliency map as y'_{map} for the salient region, and use Eq. 6 to predict the corresponding saliency bounding box. The loss of predicting saliency bounding boxes is the average of the losses for all the boxes.

3) Iterative Saliency Map Updating: We use the map refinement network to iteratively update the initial pseudo ground truth saliency maps to obtain the final pseudo ground truth labels. Iterative updating is performed by training the map refinement network and updating the saliency maps over training iterations. More specifically, at each iteration, for each saliency map, we use the weights of the network obtained from the previous iteration to predict a new saliency map y'_{map}. We then check if y'_{map} is qualified to be the final pseudo ground truth. For this, we first eliminate false predictions outside the saliency bounding box. y'_{map} is then binarized to generate a new tight bounding box $\hat{y}_{box}$. We further compute boundary distances S between y_{box} and $\hat{y}_{box}$ according to Eq. 2. When the absolute values of the four distances are less than threshold d (i.e., $|S_i| < 10, \forall \text{ i} \in \{1, 2, 3, 4\}$), y'_{map} is regarded as *updated* and assigned to the final pseudo ground truth y_{map}. Note that saliency bounding boxes are directly involved in the updating process of the pseudo ground truth saliency maps, to provide explicit guidance to the updating process.

To determine when to terminate iterative updating, we define *update rate* as the number of updated images divided by the number of samples in the training dataset. We stop the training when the $update\ rate$ exceeds 0.8. For the pseudo ground truth that is not updated, we pick the best predicted saliency map from different epochs of the training to update it. The best map is selected by minimizing the difference between its tight bounding box and the corresponding saliency bounding box, measured by $\sum_{i=1}^{4} |S_i|$, where S_i is defined in Eq. 2.

D. Training a Salient Object Detector

Once we have obtained high-quality pseudo ground truth saliency maps, we use them to train a salient object detector. In particular, we treat the map refinement network as our salient object detector and train it on the pseudo ground truth labels for the salient object detection task while keeping the weights of the saliency bounding box prediction branch fixed.

IV. Experiments

A. Experimental Setup

1) Training Details: Our training dataset is DUTS-TR [2], which is widely used for weakly-supervised SOD [2] [19]. It includes about 10,550 salient images, and we create saliency bounding boxes for them, resulting in a total of 11,003 saliency bounding boxes. Our saliency bounding boxes are generated from the ground truth saliency maps of the images according to the rules defined in Section III-A. In particular, for each salient region in an image, we take its tight bounding box as the saliency bounding box. If two saliency bounding boxes overlap, they are merged into a single saliency bounding box. Our training has two steps. First, we train the map refinement network with both saliency bounding boxes and initial pseudo ground truth saliency maps (i.e., the second step in Figure 2). Second, we train it on the final pseudo ground truth only (i.e., the third step in Figure 2). The optimizer is Adam [43] with a learning rate of 0.0001, and the batch size is set to 4. All training images are resized to 256 × 256. At the first few epochs of the first training step, we do not update the saliency maps since the prediction maps are inaccurate at the early stage. At the second training step, we flip the images horizontally to double the size of the training dataset. We train our network from scratch by initializing its weights randomly.

TABLE I
QUANTITATIVE COMPARISON OF OUR METHOD WITH FULLY-SUPERVISED METHODS (MARKED WITH †), UNSUPERVISED METHODS, AND WEAKLY-SUPERVISED METHODS (MARKED WITH *).

Method	Supervision	ECSSD		DUTS-TE		HKU-IS		DUT-OMRON	
		$F_\beta \uparrow$	MAE ↓	$F_\beta \uparrow$	MAE ↓	$F_\beta \uparrow$	MAE ↓	$F_\beta \uparrow$	MAE ↓
UCF † [33]	fully-supervised	0.841	0.080	0.629	0.117	0.808	0.074	0.613	0.132
Amulet † [34]	fully-supervised	0.868	0.059	0.736	0.085	0.842	0.052	0.647	0.098
PAGR † [35]	fully-supervised	0.891	0.064	0.788	0.055	0.886	0.048	0.711	0.072
DGRL † [36]	fully-supervised	0.903	0.045	0.768	0.051	0.882	0.037	0.709	0.063
PAGE † [37]	fully-supervised	0.924	0.042	0.815	0.051	0.918	0.037	0.770	0.066
TBIN † [38]	fully-supervised	0.931	0.032	0.840	0.040	0.920	0.030	0.781	0.056
CSF † [39]	fully-supervised	0.947	0.036	0.893	0.037	0.936	0.030	0.833	0.055
MR [40]	unsupervised	0.690	0.186	0.510	0.189	0.655	0.174	0.577	0.194
wCtr [41]	unsupervised	0.676	0.179	0.506	0.163	0.677	0.149	0.536	0.145
HS [42]	unsupervised	0.627	0.229	0.460	0.258	0.623	0.223	0.507	0.237
MB+ [7]	unsupervised	0.697	0.174	0.528	0.179	0.678	0.151	0.531	0.167
BSCA [27]	unsupervised	0.707	0.185	0.500	0.197	0.654	0.175	0.509	0.190
MST [8]	unsupervised	0.693	0.151	0.540	0.156	0.680	0.131	0.542	0.149
ASMO* [3]	category	0.810	0.114	0.625	0.123	0.821	0.091	0.633	0.100
WSS* [2]	category	0.828	0.105	0.657	0.106	0.821	0.081	0.611	0.111
NWS* [19]	category and caption	0.846	0.096	0.704	0.097	0.823	0.086	0.619	0.109
Ours*	saliency bounding boxes	**0.860**	**0.072**	**0.736**	**0.079**	**0.853**	**0.058**	**0.686**	**0.081**

TABLE II
QUANTITATIVE COMPARISON OF OUR METHOD WITH UNSUPERVISED METHODS, AND WEAKLY-SUPERVISED METHODS (MARKED WITH *) IN E-MEASURE (E_m), S-MEASURE (S_m) AND WEIGHTED F-MEASURE (F_β^w).

Method	ECSSD			DUTS-TE			HKU-IS			DUT-OMRON		
	$S_m\uparrow$	$E_m \uparrow$	$F_\beta^w \uparrow$	$S_m\uparrow$	$E_m \uparrow$	$F_\beta^w \uparrow$	$S_m\uparrow$	$E_m \uparrow$	$F_\beta^w \uparrow$	$S_m \uparrow$	$E_m\uparrow$	$F_\beta^w \uparrow$
MR [40]	0.662	0.610	0.465	0.585	0.569	0.314	0.639	0.599	0.456	0.607	0.591	0.358
wCtr [41]	0.673	0.643	0.502	0.635	0.634	0.385	0.696	0.669	0.488	0.678	0.668	0.435
HS [42]	0.685	0.630	0.468	0.590	0.571	0.315	0.673	0.652	0.445	0.620	0.599	0.352
MB+ [7]	0.723	0.720	0.565	0.655	0.667	0.413	0.711	0.705	0.562	0.671	0.679	0.432
BSCA [27]	0.727	0.671	0.517	0.632	0.607	0.347	0.699	0.656	0.508	0.652	0.629	0.372
MST [8]	0.708	0.768	0.623	0.645	0.709	0.482	0.710	0.780	0.618	0.657	0.711	0.487
ASMO* [3]	0.802	0.807	0.705	0.696	0.690	0.49	0.784	0.788	0.675	0.752	0.765	0.563
WSS* [2]	0.811	0.806	0.721	0.749	0.746	0.57	0.822	0.820	0.713	0.730	0.731	0.530
NWS* [19]	0.827	0.791	0.725	0.759	0.743	0.593	0.814	0.809	0.717	0.756	0.729	0.536
Ours*	**0.858**	**0.889**	**0.82**	**0.789**	**0.831**	**0.680**	**0.852**	**0.897**	**0.805**	**0.776**	**0.810**	**0.650**

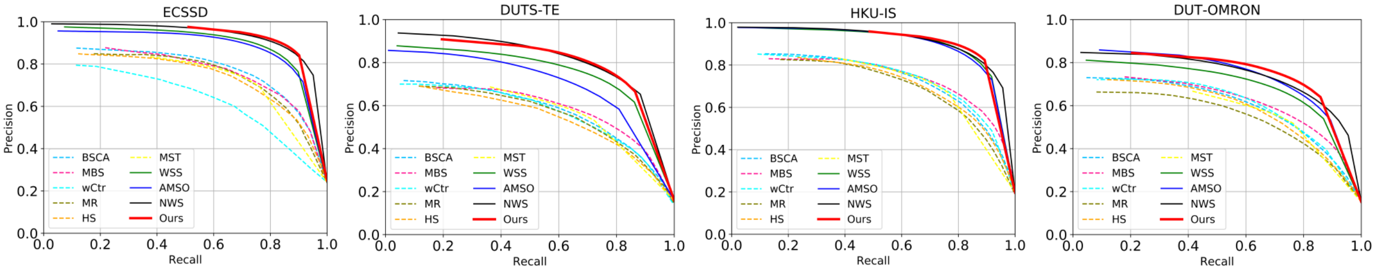


Fig. 11. Precision-recall curves of our method compared against 3 weakly-supervised methods (solid lines) and 6 unsupervised methods (dashed lines).

2) Evaluation Metrics: We evaluate our method on four benchmark datasets: ECSSD [42], DUTS-TE [2], HKU-IS [44], and DUT-OMRON [40].

We adopt popular SOD evaluation metrics, including F-measure (F_β) and MAE, to quantitatively evaluate the performance of our method. F-measure is defined as:

$$F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}, \tag{8}$$

where β^2 is set to 0.3 as in [45]. MAE measures the average pixel-wise absolute difference between a predicted saliency map S and the ground truth G as:

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - G(x,y)|. \tag{9}$$

To further evaluate our method, we also include additional evaluation metrics, including S-measure [46], E-measure [47] and weighted F-measure [48].

S-measure evaluates spatial structure similarity based on region-aware structural similarity S_r and object-aware structural similarity S_o as:

$$S_m = \alpha \times S_o + (1-\alpha) \times S_r, \tag{10}$$

where α is set to 0.5 in our experiment.

E-measure is an enhanced alignment measure to jointly capture image-level statistics and local pixel matching information with an alignment matrix ϕFM as:

$$E_m = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} \phi FM(x, y), \tag{11}$$

where H and W are the height and width of the image, respectively.

B. Comparison with State-of-the-arts

We compare our method with state-of-the-art methods, including six unsupervised methods (MR [40], wCtr [41], HS [42], BSCA [27], MB+ [7] and MST [8]), three weakly-supervised methods (AMSO [3], WSS [2] and NWS [19]) and seven fully-supervised methods (UCF [33], Amulet [34], PAGR [35], DGRL [36], PAGE [37], TBIN [38] and CSF [39]).

1) Quantitative Results: As shown in Table I, our method outperforms previous weakly-supervised methods by a large margin. It outperforms the best-performing existing weakly-supervised method by more than 0.03 in F-measure and more than 0.02 in MAE on the four datasets. In addition, we can see that our method outperforms these existing methods more on complex datasets (e.g., DUT-OMRON) than on simple datasets (e.g., ECSSD). This shows that our method can produce high-quality saliency maps even for some complex scenes with cluttered background. Specifically, our method outperforms the best-performing existing weakly-supervised method by 0.053 on DUT-OMRON and by 0.014 on ECSSD in F-measure. The performance gain comes from the benefit of using saliency bounding boxes as supervision, which provide more accurate location and size information of the salient regions. In addition, although lacking the ground truth saliency maps, we can obtain high-quality pseudo ground truth, by leveraging multi-task learning, to effectively extract saliency features under the supervision of saliency bounding boxes. Table II shows a further quantitative comparison on S-measure [46], E-measure [47] and weighted F-measure [48].

Figure 11 shows the precision-recall curves of our method compared with three weakly-supervised methods and six unsupervised methods on the four datasets. The precision-recall curves of our method are closer to the coordinates (1,1), which means that our method can detect more ground truth foreground pixels with high accuracy.

2) Qualitative Results: Figure 12 shows some qualitative results. We can see that our results are significantly better than those from the existing methods on various types of images. For example, in the first two rows, our method has almost no false positives on the background, since saliency bounding boxes can provide accurate localization of the saliency regions. Our method also performs well on small objects, e.g., the doll in the third row and the person in the fourth row. This may be partly because our method can fix the over-coverage problem well via the coverage adjustment module. Compared with the existing weakly-supervised methods, our method can detect the salient objects more precisely, without missing some salient parts, e.g., the two boys in the fifth row and the three rabbits in the sixth row. Our performance advantage may be due to three reasons. First, with under-coverage adjustment, we can obtain high-quality saliency maps that cover the whole objects. Second, saliency bounding boxes provide more accurate location information than using only image-level labels. They can guide the detector to roughly locate the salient regions. Third, each saliency region individually goes through the map refinement network to obtain the saliency bounding box just for that region. This enables our method to locate the center of the object and sharpen its edge.

TABLE III
EFFECT OF EACH COMPONENT (IN TERMS OF F-MEASURE) ON THE ECSSD DATASET. EACH ROW REPRESENTS A VARIANT AND ✓ INDICATES THAT THE COMPONENT IS INCLUDED IN THE VARIANT.

ISMs	CAM	Initial pseudo GT	SBB	$F_\beta \uparrow$
✓				0.746
✓	✓			0.777
✓	✓	✓		0.823
✓	✓	✓	✓	0.860

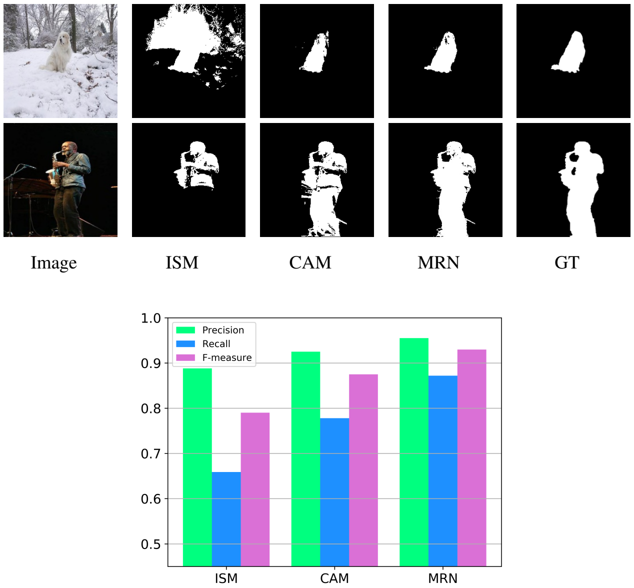


Fig. 13. Quality change of saliency maps. Top: visualization of the saliency maps from different steps of our method. Bottom: Precision, recall and F-measure of the saliency maps, compared against the ground truth from DUTS-TE. ISM is the initial saliency map by combining the outputs of unsupervised salient object detection methods. CAM is the saliency map after the coverage adjustment module. MRN is the final saliency map output by the map refinement network.

C. Ablation Study

In this section, we use the SOD model trained with only the initial saliency maps (ISMs) as a baseline, and analyze the contributions of the critical components in our method: coverage adjustment module (CAM), training the map refinement network with initial pseudo GT and with saliency bounding box (SBB). The results are shown in Table III, which demonstrates that all the components in our model are important to its superior performances.

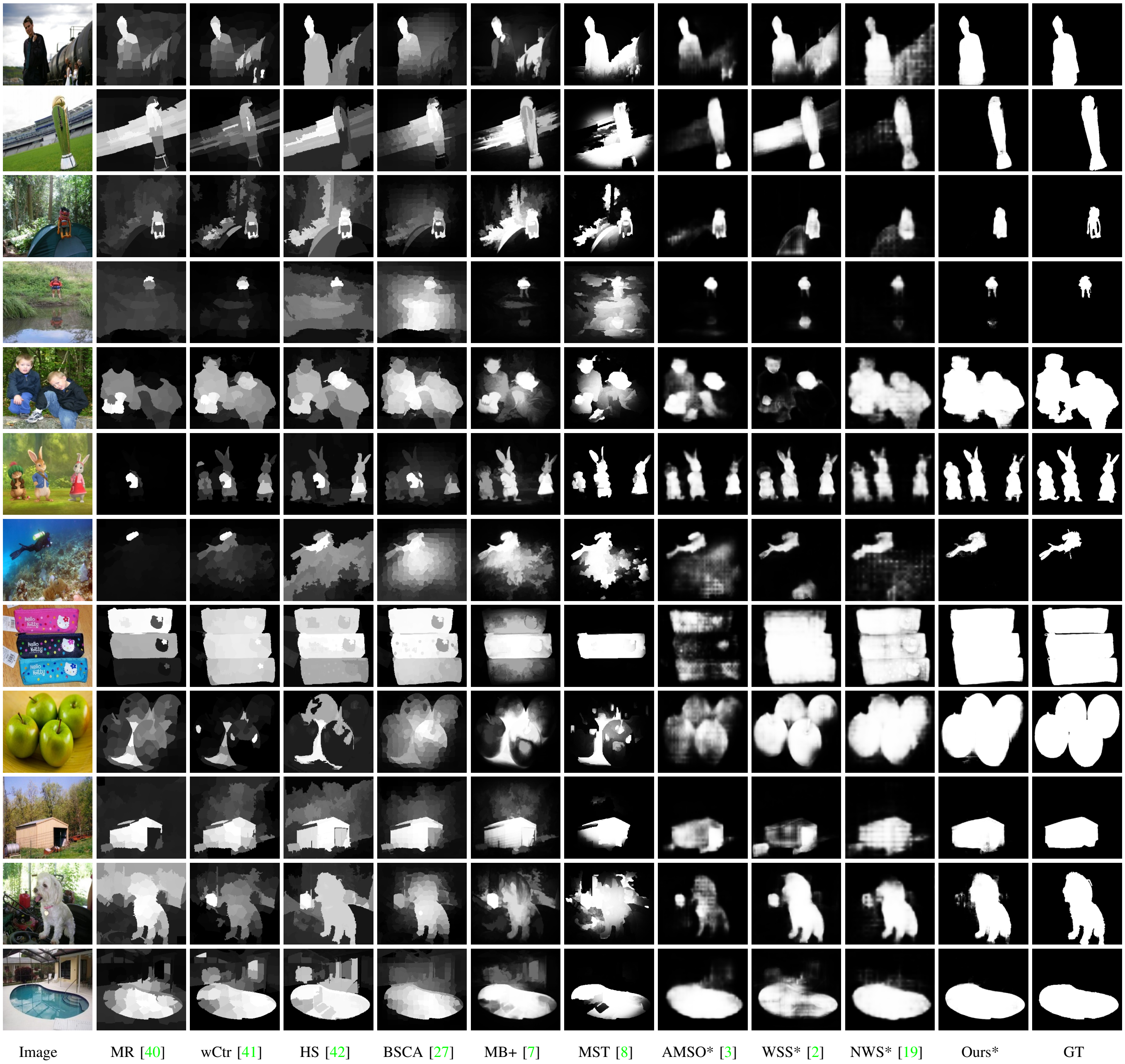


Fig. 12. Visual comparison of our method with unsupervised methods and weakly-supervised methods (marked with *). AMSO is trained on MSCOCO [49], MSRA-B [50] and HKU-IS [44] (training set), WSS is trained on ImageNet [26], NWS is trained on MSCOCO [49] and DUTS-TE [2]. These images are from four benchmark datasets: ECSSD [42], DUTS-TE [2], HKU-IS [44], DUT-OMRON [40].

TABLE IV
QUANTITATIVE IMPROVEMENT OF SALIENCY MAPS FOR THE WHOLE DATASET EVERY 0.05K ITERATIONS, MEASURED IN F_β.

Iteration	0.05k	0.1k	0.15k	0.2k	0.25k	0.3k	0.3k	0.4k
F_β	0.853	0.859	0.872	0.887	0.901	0.911	0.923	0.929

D. Quality Improvement of the Saliency Maps

Figure 13(top) visualizes the saliency maps of two examples through each step of our method. We can see that the quality of the saliency maps is gradually improved after each step, and finally becomes comparable to that of the ground truth. For quantitative evaluation, we compare the saliency maps from different steps of our method with the ground truth. We report precision, recall and F-measure on DUTS-TR [2] in Figure 13(bottom). To test the effectiveness of the iterative saliency map updating step discussed in Section III-C3, we show some saliency maps at different iterations in Figure 14. We can observe that the saliency maps are continuously improved over the iterations. In Table IV, we quantitatively demonstrate the improvement of the saliency maps for the entire dataset over the iterations.

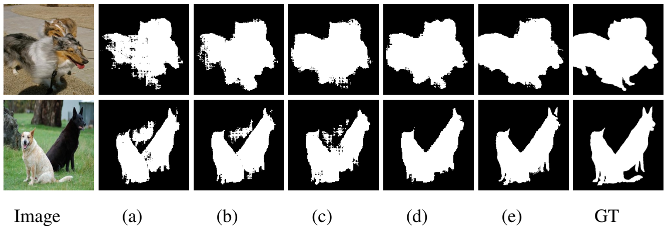


Fig. 14. Saliency maps at different iterations of the iterative updating process in Section III-C3 (a-d). (e) shows the final results obtained by applying CRFs to (d), which are considered as the final pseudo ground truth.

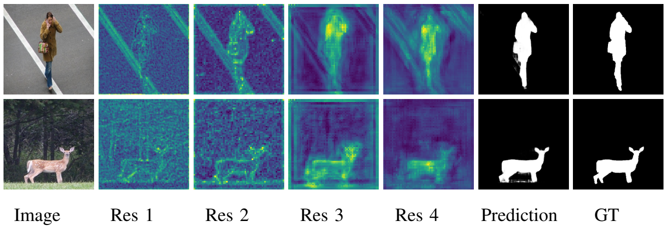


Fig. 15. Visualization of the feature maps from our salient object detector.

E. Visualization of the Feature Maps

We visualize the feature maps of our salient object detector, which uses ResNet101 as the backbone. We take the outputs of different residual blocks, and apply average pooling to each output along the channel dimension. Figure 15 shows the resulting feature maps. We can see that the feature maps at lower layers respond to low-level features of the salient objects, such as corners and edges, while the feature maps from higher layers respond strongly to salient objects and their parts.

F. Comparison with Weakly-Supervised Semantic Segmentation / Interactive Segmentation Methods

One may question whether existing weakly-supervised semantic segmentation methods can already tackle salient object detection. To investigate this, we compare with a recent weakly-supervised semantic segmentation method based on bounding boxes [23]. Note that our saliency bounding boxes are conceptually different from the generic bounding boxes [49] used in weakly-supervised semantic segmentation, which are defined on all the objects in each image without considering saliency information. Thus, for fair comparison, we train their model on our saliency dataset with saliency bounding box annotations. As shown in Table V, our results are better than those of SDI [23].

TABLE V
COMPARISON OF OUR METHOD WITH A WEAKLY-SUPERVISED SEMANTIC SEGMENTATION METHOD SDI [23] AND AN INTERATIVE SEGMENTATION METHOD GRABCUT [51].

Method	ECSSD		DUTS-TE	
	$F_\beta \uparrow$	MAE $\downarrow$	$F_\beta \uparrow$	MAE $\downarrow$
SDI [23]	0.844	0.074	0.672	0.110
GrabCut [23]	0.829	0.102	0.674	0.098
Ours	**0.860**	**0.072**	**0.736**	**0.079**

We further compare our method with an interactive segmentation method, GrabCut [51]. We first apply GrabCut to the saliency bounding boxes to obtain pseudo ground truth saliency maps. We then train our saliency object detector using the resulting saliency maps. For fair comparison, we only use saliency bounding boxes for pseudo ground truth map generation without additional user interaction. As shown in Table V, our method performs better than GrabCut, mainly due to two reasons. First, we design the coverage adjustment module to refine the unsupervised saliency maps. Second, we propose the multi-task network to jointly learn two tasks, which can extract more useful features and thus obtain higher quality saliency maps compared to the unsupervised method.

TABLE VII
AVERAGE ANNOTATION TIMES FOR ONE IMAGE USING DIFFERENT FORMS OF SUPERVISION.

Supervision	Annotation Time (sec)
Category label [49]	80
Bounding box [52]	78.5
Saliency bounding box (Ours)	10.6

G. Annotation Time

To evaluate the efficiency of acquiring saliency bounding boxes, we compare the annotation times for different forms of supervision in Table VII. It can be seen that our annotation time is much less than those of the other supervision forms. Collecting image-level labels takes about 1 second per class on average [53]. Thus, annotating an image with 80 object classes in MSCOCO dataset [49] would takes about 80 seconds. According to [52], labeling one bounding box takes about 10.2 seconds. MSCOCO contains 7.7 object instances per image [49]. Hence, labeling one image takes about 78.5 seconds. In this paper, we annotate the DUTS-TR dataset [2], and label 1.04 saliency bounding boxes per image on average. Therefore, labeling an image takes about 10.6 seconds (1.04 boxes × 10.2 seconds per box).

H. Effect of Hyperparameters

We study the effects of the main hyperparameters on the quality of the final saliency maps. These hyperparameters include k in Section III-B (set to 1.5 in our implementation), d in coverage checking (set to -30 in our implementation), β in the under-coverage sub-module (set to 32 in our implementation), and the update rate in saliency map refinement (set to 0.8 in our implementation). For each of these hyperparameters, we set it to different values to observe its effect on the performance of the final pseudo ground truth maps, measured by F_β. The results are shown in Table VIII, where we adjust each of the hyperparameters within a reasonable range. For the update rate, 0.8 gives the best performance and the performance gradually degrades as the value further increases. This is because there are always some cases that are difficult for our weakly-supervised method to produce good enough final pseudo ground truth maps.

TABLE VI
COMPARISON WITH A FULLY SUPERVISED METHOD, CSF [39], TRAINED ON OUR GENERATED PSEUDO LABELS (OURS) AND TRUE LABELS (TRUE).

Label	ECSSD				DUTS-TE				HKU-IS				DUT-OMRON			
	$F_\beta\uparrow$	MAE $\downarrow$	$S_m\uparrow$	$E_m\uparrow$	$F_\beta\uparrow$	MAE $\downarrow$	$S_m\uparrow$	$E_m\uparrow$	$F_\beta\uparrow$	MAE $\downarrow$	$S_m\uparrow$	$E_m\uparrow$	$F_\beta\uparrow$	MAE $\downarrow$	$S_m\uparrow$	$E_m\uparrow$
Ours	0.919	0.035	0.929	0.947	0.839	0.039	0.889	0.913	0.897	0.033	0.916	0.913	0.764	0.054	0.844	0.866
True	0.922	0.033	0.930	0.948	0.843	0.038	0.890	0.911	0.902	0.031	0.921	0.946	0.755	0.055	0.838	0.854

TABLE VIII
EFFECT OF SOME KEY HYPERPARAMETERS ON THE QUALITY OF THE FINAL SALIENCY MAPS, MEASURED VIA F_β. THE BEST RESULTS ARE SHOWN IN BOLD.

k	1.0	1.2	1.5	1.8	2.1
F_β	0.920	0.923	0.932	**0.933**	0.904

d	-46	-38	-30	-22	-14
F_β	0.878	0.912	0.932	**0.934**	0.931

β	16	24	32	40	48
F_β	0.899	0.921	**0.932**	**0.932**	0.930

update rate	0.70	0.75	0.80	0.85	0.90
F_β	0.886	0.910	**0.932**	0.930	0.924

I. Application to Fully-Supervised Methods

In Section III-D, while our method aims to be weakly-supervised, we investigate if our framework can also benefit fully-supervised methods by training them on our generated pseudo ground truth labels. To this end, we consider a state-of-the-art fully-supervised SOD method, CSF [39]. We train two variants of CSF on each dataset, one with pseudo ground truth labels generated by our framework and one with true labels provided by the dataset, and compare the performances of the two variants. We show results in Table VI. The model trained with our pseudo ground truth labels achieves comparable performances with its counterpart trained on the true labels. However, our pseudo labels are significantly cheaper to obtain than the true labels. For example, generating pseudo labels for all the images in DUTS-TR with our framework takes around 42 hours (32 hours for annotating saliency bounding boxes plus 10 hours for generating pseudo labels), as opposed to around 230 hours of manual labeling to obtain the true labels. This suggests that our framework can benefit fully-supervised methods by dramatically reducing the amount of labeling efforts.

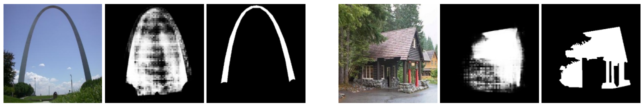


Fig. 16. Typical failure cases of our method. Our method has a poor performance on images with complex shapes.

J. Failure Cases

Our method may have poor performances on salient objects with complex shapes, as shown in Figure 16. This is because saliency bounding boxes can only provide rough information about salient object size and location, lacking information about object shape details. One possible solution for future work is to add some local edge annotation to help localize salient object edges, and combine this information with saliency bounding boxes.

V. CONCLUSION

In this paper, we propose a novel weakly-supervised method for salient object detection by leveraging supervision with bounding boxes. Given the saliency bounding boxes, we have proposed a framework to generate pseudo ground truth saliency maps to supervise the learning of a robust salient object detector. To obtain high-quality pseudo ground truth labels, we first design a coverage adjustment module to address under/over predictions from existing unsupervised methods. We then propose a multi-task map refinement network to iteratively refine the pseudo saliency maps. Experimental results demonstrate that our method greatly outperforms all state-of-the-art weakly-supervised methods on multiple popular benchmark datasets.

REFERENCES

[1] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, “Whats the point: Semantic segmentation with point supervision,” in ECCV, 2016, pp. 549–565. 1

[2] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, “Learning to detect salient objects with image-level supervision,” in CVPR, 2017, pp. 136–145. 1, 2, 6, 7, 8, 9, 10

[3] G. Li, Y. Xie, and L. Lin, “Weakly supervised salient object detection using image labels,” in AAAI, 2018. 1, 2, 7, 8, 9

[4] Y. Zhai and M. Shah, “Visual attention detection in video sequences using spatiotemporal cues,” in ACM Multimedia, 2006, pp. 815–824. 2

[5] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, “Salient region detection and segmentation,” in ICCV, 2008, pp. 66–75. 2

[6] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, “Global contrast based salient region detection,” TPAMI, vol. 37, no. 3, pp. 569–582, 2014. 2

[7] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, “Minimum barrier salient object detection at 80 fps,” in ICCV, 2015, pp. 1404–1412. 2, 3, 7, 8, 9

[8] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien, “Real-time salient object detection with a minimum spanning tree,” in CVPR, 2016, pp. 2334–2342. 2, 3, 7, 8, 9

[9] D. Zhang, J. Han, and Y. Zhang, “Supervision by fusion: Towards unsupervised learning of deep salient object detector,” in ICCV, 2017, pp. 4048–4056. 2

[10] J. Zhang, T. Zhang, Y. Dai, M. Harandi, and R. Hartley, “Deep unsupervised saliency detection: A multiple noisy labeling perspective,” in CVPR, 2018, pp. 9029–9038. 2

[11] T. Nguyen, M. Dax, C. K. Mummadi, N. Ngo, T. H. P. Nguyen, Z. Lou, and T. Brox, “Deepusps: Deep robust unsupervised saliency prediction via self-supervision,” in NIPS, 2019, pp. 204–214. 2

[12] G. Li and Y. Yu, “Contrast-oriented deep neural networks for salient object detection,” IEEE transactions on neural networks and learning systems, vol. 29, no. 12, pp. 6038–6051, 2018. 2

[13] L. Huang, G. Li, Y. Li, and L. Lin, “Lightweight adversarial network for salient object detection,” Neurocomputing, vol. 381, pp. 130–140, 2020. 2

[14] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, “Deep networks for saliency detection via local estimation and global search,” in CVPR, 2015, pp. 3183–3192. 2

[15] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, “Saliency detection with recurrent fully convolutional networks,” in ECCV. Springer, 2016, pp. 825–841. 2

[16] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. Torr, “Deeply supervised salient object detection with short connections,” in CVPR, 2017, pp. 3203–3212. 2

[17] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, “A bi-directional message passing model for salient object detection,” in CVPR, 2018, pp. 1741–1750. 2

[18] R. Wu, M. Feng, W. Guan, D. Wang, H. Lu, and E. Ding, “A mutual learning method for salient object detection with intertwined multi-supervision,” in CVPR, 2019, pp. 8150–8159. 2

[19] Y. Zeng, Y. Zhuge, H. Lu, L. Zhang, M. Qian, and Y. Yu, “Multi-source weak supervision for saliency detection,” in CVPR, 2019. 2, 6, 7, 8, 9

[20] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, “Unconstrained salient object detection via proposal subset optimization,” in CVPR, 2016, pp. 5733–5742. 2

[21] P. Siva, C. Russell, T. Xiang, and L. Agapito, “Looking beyond the image: Unsupervised learning for object saliency and detection,” in CVPR, 2013, pp. 3238–3245. 2

[22] J. Dai, K. He, and J. Sun, “Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation,” in ICCV, 2015, pp. 1635–1643. 2

[23] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, “Simple does it: Weakly supervised instance and semantic segmentation,” in CVPR, 2017, pp. 876–885. 2, 10

[24] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” IJCV, vol. 88, no. 2, pp. 303–338, 2010. 3

[25] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” IJCV, vol. 111, no. 1, pp. 98–136, 2015. 3

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., “Imagenet large scale visual recognition challenge,” IJCV, vol. 115, no. 3, pp. 211–252, 2015. 3, 9

[27] Y. Qin, H. Lu, Y. Xu, and H. Wang, “Saliency detection via cellular automata,” in CVPR, 2015, pp. 110–119. 3, 7, 8, 9

[28] C. Sutton, A. McCallum et al., “An introduction to conditional random fields,” Machine Learning, vol. 4, no. 4, pp. 267–373, 2012. 3

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in CVPR, 2016, pp. 770–778. 5

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in CVPR, 2017, pp. 4700–4708. 5

[31] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” arXiv:1502.03167, 2015. 5

[32] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, “Cbam: Convolutional block attention module,” in ECCV, 2018, pp. 3–19. 5

[33] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, “Learning uncertain convolutional features for accurate saliency detection,” in ICCV, 2017, pp. 212–221. 7, 8

[34] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, “Amulet: Aggregating multi-level convolutional features for salient object detection,” in ICCV, 2017, pp. 202–211. 7, 8

[35] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, “Progressive attention guided recurrent network for salient object detection,” in CVPR, 2018, pp. 714–722. 7, 8

[36] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji, “Detect globally, refine locally: A novel approach to saliency detection,” in CVPR, 2018, pp. 3127–3135. 7, 8

[37] W. Wang, S. Zhao, J. Shen, S. C. Hoi, and A. Borji, “Salient object detection with pyramid attention and salient edges,” in CVPR, 2019, pp. 1448–1457. 7, 8

[38] W. Wang, J. Shen, M.-M. Cheng, and L. Shao, “An iterative and cooperative top-down and bottom-up inference network for salient object detection,” in CVPR, 2019, pp. 5968–5977. 7, 8

[39] S.-H. Gao, Y.-Q. Tan, M.-M. Cheng, C. Lu, Y. Chen, and S. Yan, “Highly efficient salient object detection with 100k parameters,” in ECCV, 2020. 7, 8, 11

[40] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, “Saliency detection via graph-based manifold ranking,” in CVPR, 2013, pp. 3166–3173. 7, 8, 9

[41] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, “Saliency detection via dense and sparse reconstruction,” in ICCV, 2013, pp. 2976–2983. 7, 8, 9

[42] Q. Yan, L. Xu, J. Shi, and J. Jia, “Hierarchical saliency detection,” in CVPR, 2013, pp. 1155–1162. 7, 8, 9

[43] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv:1412.6980, 2014. 6

[44] G. Li and Y. Yu, “Visual saliency detection based on multiscale deep cnn features,” IEEE transactions on image processing, vol. 25, no. 11, pp. 5012–5024, 2016. 7, 9

[45] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, “Frequency-tuned salient region detection,” in CVPR, 2009, pp. 1597–1604. 7

[46] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, “Structure-measure: A new way to evaluate foreground maps,” in ICCV, 2017, pp. 4548–4557. 7, 8

[47] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, “Enhanced-alignment measure for binary foreground map evaluation,” arXiv preprint arXiv:1805.10421, 2018. 7, 8

[48] R. Margolin, L. Zelnik-Manor, and A. Tal, “How to evaluate foreground maps?” in CVPR, 2014, pp. 248–255. 7, 8

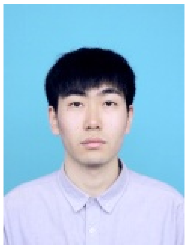
[49] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in ECCV, 2014, pp. 740–755. 9, 10

[50] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, “Learning to detect a salient object,” IEEE TPAMI, vol. 33, no. 2, pp. 353–367, 2010. 9

[51] C. Rother, “Grabcut : Interactive foreground extraction using iterated graph cuts,” vol. 23, 2004, pp. 309–314. 10

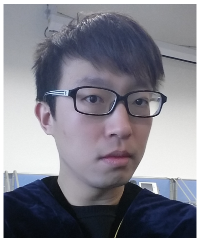
[52] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, “Whats the point: Semantic segmentation with point supervision,” in ECCV, 2016, pp. 549–565. 10

[53] D. P. Papadopoulos, A. D. Clarke, F. Keller, and V. Ferrari, “Training object class detectors from eye tracking data,” in ECCV, 2014, pp. 361–376. 10



Yuxuan Liu is currently a postgraduate student with School of Computer Science, Dalian Minzu University, Dalian, China. His research interests include computer graphics and computer vision.

Pengjie Wang is currently a professor with School of Computer Science, Dalian Minzu University, Dalian, China. His research interests include computer vision, computer graphics and data compression.



Ying Cao received the Ph.D. degree in computer science from City University of Hong Kong, and the M.Sc. and B.Eng. degrees in software engineering from Northeastern University, China. His general research interests lie in computer graphics and computer vision, with particular interest in data-driven graphic design.

Zijian Liang is currently a postgraduate student with School of Computer Science, Dalian Minzu University, Dalian, China. His research interests include computer vision and computer graphics.

Rynson W.H. Lau received the Ph.D. degree from the University of Cambridge. He was on the faculty of Durham University. He is currently with the City University of Hong Kong. His research interests include computer graphics and computer vision. He serves on the Editorial Board for Computer Graphics Forum, and Computer Animation and Virtual Worlds. He has served as the Guest Editor for the number of journal special issues, including the ACM Trans. on Internet Technology, the IEEE Trans. on Multimedia, the IEEE Trans. on Visualization and Computer Graphics, and the IEEE Computer Graphics & Applications.