Noname manuscript No.
(will be inserted by the editor)

Learning to Detect Instance-level Salient Objects using Complementary Image Labels

Xin Tian* · **Ke Xu*** · **Xin Yang†** · **Baocai Yin** · **Rynson W.H. Lau†**

Received: date / Accepted: date

Abstract Existing salient instance detection (SID) methods typically learn from pixel-level annotated datasets. In this paper, we present the first weakly-supervised approach to the SID problem. Although weak supervision has been considered in general saliency detection, it is mainly based on using class labels for object localization. However, it is non-trivial to use only class labels to learn instance-aware saliency information, as salient instances with high semantic affinities may not be easily separated by the labels. As the subitizing information provides an instant judgement on the number of salient items, it is naturally related to detecting salient instances and may help separate instances of the same class while grouping different parts of the same instance. Inspired by this observation, we propose to use class and subitizing labels as weak supervision for the SID problem. We propose a novel weakly-supervised network with three branches: a Saliency Detection Branch leveraging class consistency information to locate candidate objects; a Boundary Detection Branch exploiting class discrepancy information to delineate object boundaries; and a Centroid Detection Branch using subitizing information to detect salient instance centroids. This complementary information is then fused to produce a salient instance map. To facilitate the learning process, we further propose a progressive training scheme to reduce label noise and the corresponding noise learned by the model, via reciprocating the model with progressive salient instance prediction and model refreshing. Our extensive evaluations show that the proposed method plays favorably against carefully designed baseline methods adapted from related tasks.

Keywords SID · weak supervision · saliency detection · subitizing

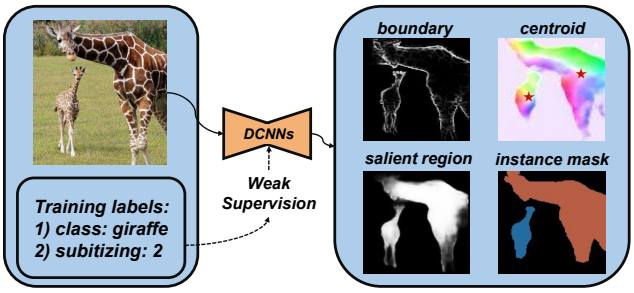


Fig. 1: Our key idea is to leverage complementary image-level labels (class and subitizing) to train a salient instance detection model in a weakly-supervised manner, via synergically learning to predict salient objects, detecting object boundaries and locating instance centroids.

1 Introduction

Salient Object Detection (SOD) is a long-standing vision task that aims to segment visually salient objects in a scene. It often serves as a core step for downstream vision tasks like video object segmentation [50], object proposal generation [4], and image cropping [49]. Recent deep learning-based SOD methods have achieved a significant performance progress [47,74,56,42,68,17,48], benefited from the

* Joint first authors, † joint corresponding authors. Rynson Lau leads this project.

Xin Tian*
Dalian University of Technology and City University of Hong Kong.

Ke Xu*
City University of Hong Kong, Hong Kong SAR, China.

Xin Yang†
Dalian University of Technology, Dalian, China.

Baocai Yin
Dalian University of Technology and Pengcheng Lab, China.

Rynson W.H. Lau†
City University of Hong Kong, Hong Kong SAR, China.

powerful representation learning capability of neural networks and large-scale pixel-level annotated training data. Since annotating pixel-level labels is extremely tedious, there are some works [47,60] that aim to explore cheaper image-level labels (*e.g.*, class labels) to train SOD models in a weakly-supervised manner.

Salient Instance Detection (SID) goes further from SOD as it aims to differentiate individual salient instances. This instance-level saliency information can benefit vision tasks that require fine-grained scene understanding, *e.g.*, object rank [41], image captioning [21], image editing [59] and semantic segmentation [43]. However, existing SID methods [11,25,62] still rely on large-scale annotated ground truth masks in order to learn how to segment salient instances with their boundaries delineated. Hence, it is worthwhile studying the SID problem from the weakly-supervised perceptive by using cheaper image-level labels.

A straightforward solution to the weakly-supervised SID problem is to use class labels for training, like the weakly-supervised SOD methods [47,60]. However, using just class labels to learn a SID model is non-trivial for two reasons. First, while class labels can help detect semantically predominant regions [70], there is no guarantee that the detected regions are visually salient. Second, objects of the same class may not be easily distinguished due to their high semantic affinity. We observe that subitizing refers to the number of certain objects and is therefore naturally related to saliency instance detection. By predicting the number of salient objects, we may use it as a global supervision to help separate instances of the same class while clustering different parts of an instance with diverse appearances into one. Inspired by this insight, we propose to learn a weakly-supervised SID network (denoted as WSID-Net) using class and subitizing labels.

Our WSID-Net consists of three synergic branches: a salient object detection branch is proposed to locate candidate salient objects while a boundary detection branch is proposed to delineate their boundaries, both by exploiting semantics from the class labels; and a centroid detection branch is proposed to detect the centroid of each salient instance, by leveraging saliency cues from the subitizing labels. This information is fused to obtain the salient instance map. To facilitate the learning process, we propose a Progressive Training Scheme (PTS) to reduce the noise generated in our salient object detection branch (*e.g.*, incomplete object proposals and cluttered background objects), by reciprocally updating the branch using generated pseudo labels and refreshing the branch in a self-supervised manner. To demonstrate the effectiveness of the proposed model, we compare it with a variety of baselines adapted from related tasks on the standard benchmark [25].

To summarize, this work has four main contributions:

- To the best of our knowledge, we propose the first weakly-supervised method for salient instance detection, which only requires image-level class and subitizing labels to obtain salient instance maps.
- We propose a novel network (WSID-Net), with a novel centroid-based subitizing loss to exploit salient instance number, a novel Boundary Enhancement module to learn instance boundaries, and a novel Cross-layer Attention module to enhance cross-layer context feature learning of centroids and the boundaries.
- We propose a novel Progressive Training Scheme, to facilitate the learning of the saliency detection branch by reducing the noise in a self-supervised manner.
- We conduct extensive experiments to analyze the proposed method, and verify its superiority against baselines adapted from related state-of-the-art approaches.

2 Related Work

2.1 Salient Instance Detection

Existing SID methods are fully-supervised. Zhang *et al.* [62] propose to detect salient instances with bounding boxes, and propose a MAP-based optimization framework to regress a large amount of pre-defined bounding boxes into a compact number of instance-level bounding boxes of high confidences. However, this method based on bounding boxes cannot detect salient instances with accurately delineated boundaries. Other works predict pixel-wise masks for the detected salient instances, and typically rely on large amount of manually annotated ground truth labels. Specifically, Li *et al.* [25] propose to first predict the saliency mask and instance-aware saliency contour, and then apply the Multi-scale Combinatorial Grouping (MCG) algorithm [5] to extract instance-level masks. Fan *et al.* [11] propose an end-to-end SID network based on the object detection model FPN [28], with a segmentation branch to segment the salient instances.

Unlike these existing SID methods, we propose in this paper to train a weakly-supervised network, which only requires image-level class and subitizing labels.

The work presented in this paper extends our BMVC oral paper [45] in three aspects. First, we provide a more comprehensive literature survey on the weakly supervised salient instance detection task and other relevant works. Second, we note that the earlier method [45] typically suffers from the salient instance incompleteness problem, due to the noise generated in both salient object detection and boundary detection branches. To address this problem, we propose a Cross-layer Attention module here to learn boundary and centroid features, and a self-supervised Progressive Training Scheme to reduce the noise in the salient object detection branch. Third, we perform more experiments to analyze

the properties of our method and show its effectiveness over existing state-of-the-art approaches.

2.2 Salient Object Detection

SOD methods aim at detecting salient objects in a scene without differentiating the detected instances. Liu *et al.* [30] formulate the SOD task as a binary segmentation problem for segmenting out the visually conspicuous objects of an image via color and contrast histogram based priors.

Traditional methods propose to leverage different handcrafted priors to detect salient objects, *e.g.*, image colors and luminance [1], global and local contrast priors [37,8], and background geometric distance prior [57]. Recently, deep learning based SOD methods achieve superior performances on the standard SOD benchmarks [57,27,39,19,47,8]. Among them, several methods explore boundary information for salient object detection. Xu *et al.* [56] propose a CRF-based architecture to refine boundaries of both deep features and saliency maps in a coarse-to-fine manner. Some methods [74,71,53,42] propose to formulate saliency and edge detection with two network branches as multi-task learning. Feature fusion strategies have also been widely explored in salient object detection. DSS [17], DGRL [48], and MINet [35] integrate multi-level features in the top-down direction. In GBMPM [66] and PAGE-Net [51], multi-level saliency features are fused in both top-down and bottom-up directions, to detect salient objects of varying scales. F3Net [52] and PFPN [46] propose to fuse features progressively, to enrich saliency features with recurrent feedback information. Attention mechanism has also been exploited to reweigh multi-scale features in order to suppress noise and enhance context learning, via dynamic weight decay scheme [13], mutual relation learning of object parts [6], gate-based interference control [69], and spatial-/channel-wise attentions on different features [68]. In particular, He *et al.*[16] propose to leverage numerical representation of subitizing to enrich spatial representations of salient objects. These methods are typically benefited from the powerful learning ability of deep neural networks as well as large-scale annotated ground truth data.

To alleviate the data annotation efforts, many weakly-supervised SOD methods are proposed, by investigating different approaches of generating pseudo saliency labels. A method leverage subitizing information alone for refining saliency prediction Some methods [61,65,31] propose to use traditional SOD methods to generate pseudo labels for training deep saliency models. Some other methods [47,60] propose to train weakly-supervised deep models using object class labels and class activation maps (CAMs) [70]. There are also some methods [26,64] that propose to combine pre-trained contour networks with segment proposals [26] or scribbles [64] to generate pseudo labels for training saliency detection networks.

However, existing weakly-supervised SOD methods cannot be directly applied to our problem, as class labels along do not provide instance-level information. In this paper, we propose to use class and subitizing labels to train our SID model.

2.3 Noise Reduction

Noise commonly exists in a weakly-supervised setting, typically when the task is a pixel-level prediction and the supervision is provided at the image-level. Existing methods typically rely on auxiliary full-annotated labels (referred to as clean labels) or pre-trained models to regularize the noise. Hu *et al.* [18] formulate it as a multi-task learning problem, in which the networks trained on a small set of clean labels can help regularize the noise in the networks trained on a large set of weak labels. Zhang *et al.* [63] propose a noise-aware method for learning a disentangled clean saliency detector from noisy labels. Lu *et al.* [32] propose a sparse learning model to learn the noise statistics from over-segmented superpixels, while Zhu *et al.* [73] propose to filter out noisy segment proposals with low matching scores. Both methods rely on additional pre-trained models for noise reduction.

Unlike the above methods, we do not leverage clean labels or pre-trained proposals as assistances. We achieve this goal by training our salient object detection branch in a progressive manner, via model refreshing and pseudo label regeneration.

3 Methodology

Class labels are widely explored in weakly-supervised SOD methods for learning to localize candidate objects, based on the pixel-level semantic affinity derived from the network responses to the class labels. However, class labels lack instance-level information, causing over- and under-detection when salient instances are from the same category. We note that subitizing, which is a cheap image-level label that denotes the number of salient instances of a scene, can serve as a complementary supervision to the class labels to provide instance-related information. Hence, we propose to use both class and subitizing labels to address our weakly-supervised SID problem. To this end, we propose a weakly-supervised SID network (WSID-Net), as shown in Figure 2.

The proposed WSID-Net has three branches:

1. A *Saliency Detection Branch* for locating candidate salient objects. This saliency detection branch is based on Deeplab [7] by modifying its last layer for binary

prediction. We propose a novel Progressive Training Scheme (PTS) to self-correct the noise coming from the weak labels and the corresponding noise learned by this saliency branch.
2. A *Centroid Detection Branch* for detecting the centroids of salient instances, where subitizing knowledge is utilized in a novel loss function to provide regularization on the global number of instance centroids.
3. A *Boundary Detection Branch* for delineating salient instance boundaries, where a novel Boundary Enhancement (BE) module is introduced to resolve the discontinuity problem of detected boundaries.

Finally, we propose a novel Cross-layer Attention (CA) module for the Centroid Detection Branch and Boundary Detection Branch to learn the context information for detecting centroids and boundaries, respectively.

3.1 Centroid Detection Branch

Detecting object centroids is crucial to separating object instances in a weakly-supervised scheme. Unlike existing semantic (instance) segmentation methods [2,34,72,9,73,24] that detect the centroids based on network responses to the class labels, we propose to introduce subitizing information to explicitly supervise the salient centroid detection process.

3.1.1 Centroid-based Subitizing Loss

It has been shown that penalizing the centroid loss [2,34] helps cluster local pixels with high semantic affinities. However, this typically fails when salient instances from the same object category have varying shapes and appearances. The reason is that the clustering process of local pixels lacks global saliency supervision. Hence, we introduce the centroid-based subitizing loss $\mathcal{L}_{\mathcal{SU}}$ to resolve this problem. We use subitizing to explicitly supervise the number of predicted centroids, which is implicitly related to the learned offset vectors as $\mathcal{L}_{su}$ is back-propagated to the centroid detection branch during training, to guide the centroid-aware pixel clustering process. The detailed formulation is discussed below.

The Centroid Detection Branch predicts an offset vector map $\mathcal{V} \in \mathbb{R}^{W \times H \times 2}$, where $W \times H$ denotes the spatial size of the map, and each 2D vector $v_i \in \mathcal{V}$ indicates the vertical and horizontal distances of the i^{th} pixel from its associated instance centroid. We follow [2] to iteratively derive $\mathcal{V}$ as:

$$v_i^{m+1} = v_i^m + v_{v_i^m + p_i}, \tag{1}$$

where p_i is the coordinates of the i^{th} pixel, m is the iteration number, and $v_i^m + p_i$ indexes the the current centroid that the offset of the i^{th} pixel points to. Ideally, Eq. 1 would converge within a few iterations when $v_i^{m+1} = v_i^m$ and the offset of the centroid is zero, and yields a set of centroids that represent the instances. Pixel i can then be assigned to its corresponding centroid c_n by measuring its distance from the centroid, as described by:

$$c_{i \to n} = \arg\min_n \ \|v_i + p_i - p_{c_n}\|. \tag{2}$$

We then use the saliency map $\mathcal{S}$ of the saliency detection branch to filter out non-salient instances by computing their $IoU = (\mathcal{SI}_n \cap \mathcal{S})/\mathcal{SI}_n > \theta$, to obtain a set of saliency instances $\mathcal{SI}^* = \{\mathcal{SI}_1, \mathcal{SI}_2, ..., \mathcal{SI}_{T^*}\}$, where T^* represents the number of predicted salient instances. Finally, we use MSE to measure $\mathcal{L}_{su}$ as:

$$\mathcal{L}_{su} = MSE(T^*, T), \tag{3}$$

where T is the subitizing ground truth, and T^* denotes the number of predicted centroids extracted from the offset vectors of the pixels in the salient region. Note that the loss $\mathcal{L}_{su}$ is back-propagated to update the offset vectors only in the salient region, which avoids the learning process of instance centroid detection being distracted by the non-salient background. The gradient δ of $\mathcal{L}_{su}$ is calculated as:

$$\delta = \frac{1}{K} \cdot \frac{\partial \mathcal{L}_{su}}{\partial \mathcal{V}^*}, \tag{4}$$

where $\mathcal{V}^*$ are the offset vectors in the salient region, and K is the total number of offset vectors in $\mathcal{V}^*$.

Figure 3 visualizes the results from centroid detection and the corresponding instance segmentation, with and without using the centroid-based subitizing $\mathcal{L}_{\mathcal{SU}}$ loss function. We can see that the network groups the two dogs into one when not using $\mathcal{L}_{\mathcal{SU}}$, as these two dogs have similar appearances and lie next to each other (Figure 3(b,e)). By introducing $\mathcal{L}_{\mathcal{SU}}$, the network is able to predict a correct number of centroids, and generate reasonable salient instance masks compared with the ground truth (Figure 3(c,f)).

3.1.2 Network Structure

We adopt the image-to-image translation scheme, where our network outputs a 2D centroid map, in which the value of each pixel location indicates the offset vector to its instance centroid. The bottom part of Figure 2 shows the network structure of our centroid detection branch. Given an input image, we first extract multi-scale backbone features f_1 to f_5 and feed them to the Cross-layer Attention (CA) modules with boundary-aware features for joint refinement (to be discussed in Section 3.3). We then fuse the high-level features to obtain f_h: $f_h = Conv(Concat({f_4}^*, {f_5}^*))$, which is further fused with the low-level features to produce the centroid map $\mathcal{V}$: $\mathcal{V} = \sigma(Conv(Conv(Concat(f_h, {f_1}^*, {f_2}^*, {f_3}^*))))$.

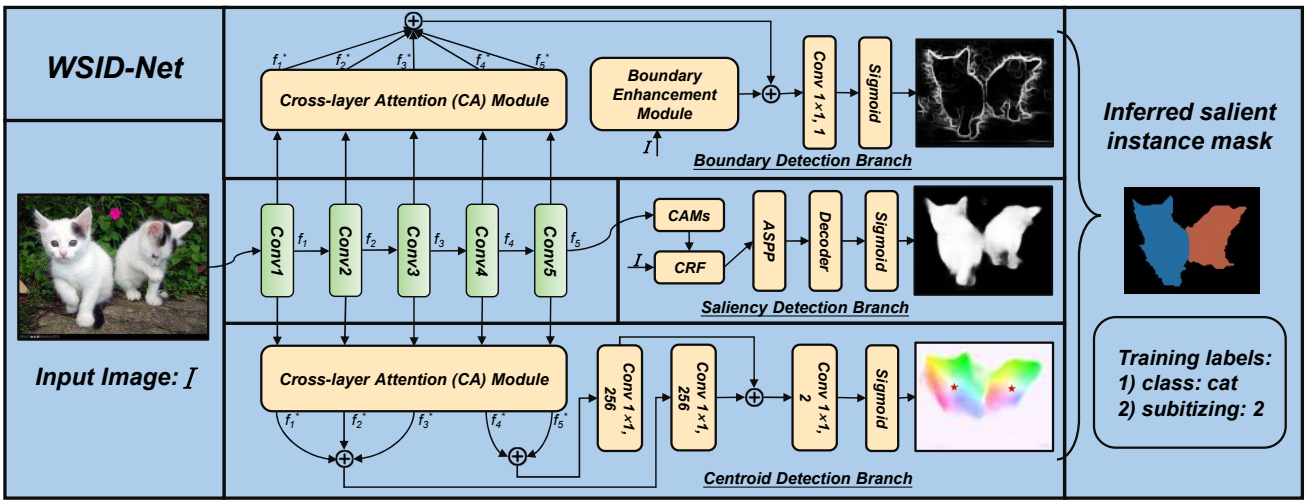


Fig. 2: Pipeline overview. Our SID model is trained using only image-level class and subitizing labels. It has three synergic branches: (1) a Boundary Detection Branch for detecting object boundaries using class discrepancy information; (2) a Saliency Detection Branch for detecting objects using class consistency information; and (3) a Centroid Detection Branch for detecting salient instance centroids using subitizing information. A random walk method is further applied to fuse these information to obtain a final salient instance mask.

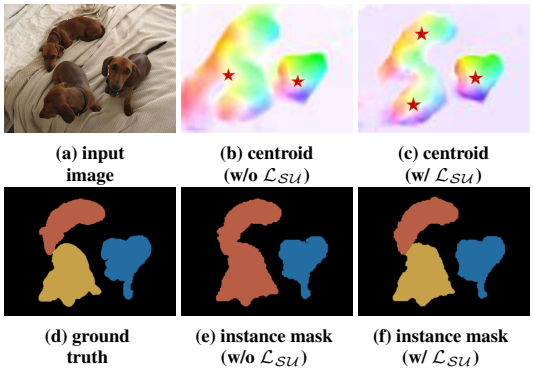


Fig. 3: Visualization of the centroid detection branch with and without $\mathcal{L}_{SU}$. Using class labels alone fails to train the network to detect instance centroids (denoted by red stars) if they have similar appearances (b), resulting in wrong segmentations (e). In contrast, our proposed subitizing loss can segment these salient objects in instance-level (f), by learning to identify the correct number of salient instances (c).

3.2 Boundary Detection Branch

Boundaries provide strong cues for separating salient instances. Unlike fully-supervised SID methods that learn boundary-aware information based on pixel-level ground truth masks, we propose the Boundary Enhancement module to leverage the Canny prior [20] to delineate continuous instance boundaries.

3.2.1 Boundary Enhancement (BE) Module

We apply a random walk algorithm to search a salient instance from a centroid to its boundary. However, it may fail when part of the boundary is discontinuous as the random walk algorithm will also search the region outside the boundary. Hence, we propose the BE module to incorporate the edge prior for learning continuous instance boundaries, as shown in Figure 4. Specifically, we first extract low-level features along the horizontal and vertical directions from the input image, by two 1×7 and 7×1 convolution layers. These low-level features are then fed into three Residual Blocks [15] for feature refinement, which are further concatenated with enriched edges computed from the Canny operator [20]. To compute the final enriched boundary features, another 1×1 convolution layer is applied.

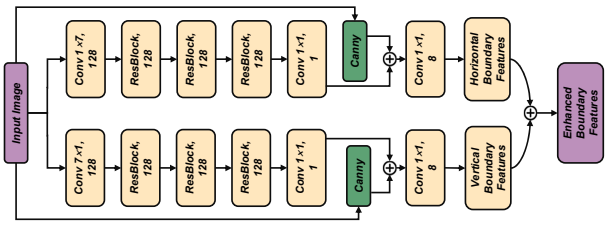


Fig. 4: Boundary Enhancement (BE) module.

Figure 5 visualizes two examples of boundary detection and the corresponding salient instance detection with and without the BE module. We can see that our BE module helps detect the boundaries between objects, which is crucial to salient instance segmentation.

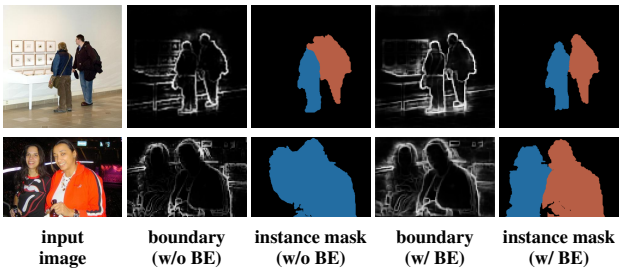


Fig. 5: Visualization of the boundary detection branch with and without the BE module, which shows the effectiveness of our proposed BE module in mining continuous boundary information for separating salient objects of same classes.

3.2.2 Network Structure

The top part of Figure 2 shows the architecture of the boundary detection branch. Given an input image $\mathcal{I}$, the backbone network produces multi-scale features (f_1 to f_5), each of which is enhanced by a CA module (to be discussed in Section 3.3) before they are concatenated and computed to predict the boundary map. We also feed the input image into the BE module to obtain enhanced edge features f_b. The output boundary map $\mathcal{B}$ is then computed as: $\mathcal{B} = \sigma(Conv(Concat({f_1}^*, ..., {f_5}^*, f_{b1}, f_{b2})))$, where σ is the sigmoid activation function.

3.3 Cross-layer Attention (CA) Module

Detecting instance centroids and boundaries are two highly coupled sub-tasks, *i.e.*, they can influence each other and further affect the SID performance. To effectively learn these two sub-tasks, we propose the Cross-layer Attention (CA) module for refining backbone features before they are used for these two sub-tasks. Its design is based on two observations. First, low-level features contain high-resolution but noisy information for delineating salient instance boundaries, while high-level features have low-resolution but robust information for salient instance localization. Second, since salient instances may have various shapes and they may correspond to different class labels, we need to model both long-range spatial and cross-channel contextual information. Unlike existing dual attention mechanisms [54, 12] that only enhance the feature representation capacity of one fixed layer, our CA module first incorporates a Cross-layer Feature Mixing (CFM) unit to enhance the communication across different levels of backbone features and then uses multiple Mutual Attention (MA) units to learn hierarchical channel-wise and spatial-wise attentive features for each sub-task. The CFM unit shares its parameters to allow information exchanges across the boundary and centroid branches. Figure 6 shows the module structure.

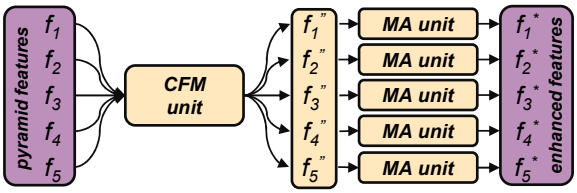


Fig. 6: Cross-layer Attention (CA) module.

3.3.1 Structure of CFM Unit

Figure 7 shows the structure of the CFM unit. f_1 to f_5 are the features from the pyramid layers of the ResNet backbone. We first upsample f_2, f_3, f_4, and f_5 such that the given feature maps have the same resolution, and apply 1×1 convolution on the five feature maps such that they have the same channel depth (256). We get features ${f_1}'$ to ${f_5}'$ that have the same shape for the following operation. We then apply CFM on the pyramid features to generate cross-layer features. CFM is implemented via a concatenation-split-concatenation operation on the feature channels. We concatenate features ${f_1}'$ to ${f_5}'$ as f_c, with 1280 (256×5) channels. The split-concatenation operation could be considered as a reshape-transpose-reshape process. We reshape channel dimension of f_c to 2 dimensions (*i.e.*, [5, 256]), transpose it to [256, 5], and then flatten it to 1280. Finally, we concatenate the features before/after CFM, and feed these concatenated features to 1×1 convolutional filters to generate the final enhanced features (${f_1}''$ to ${f_5}''$).

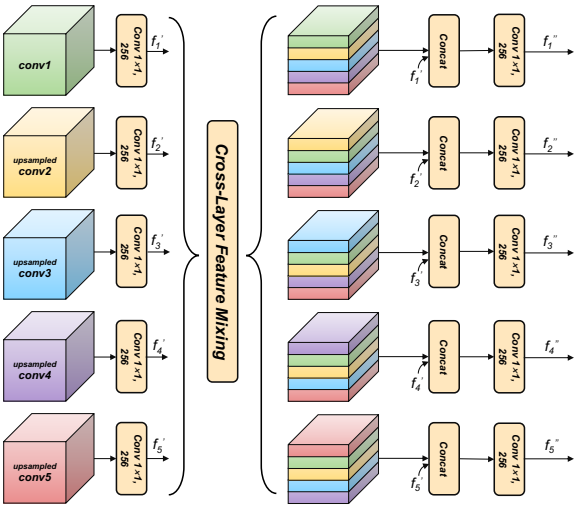


Fig. 7: Cross-layer Feature Mixing (CFM) unit.

3.3.2 Structure of MA Unit

Figure 8 shows the structure of our MA unit. The top and bottom branches are channel-wise and spatial-wise attention blocks, respectively. Specifically, given the input fea-

tures $f_n{}''$, we compute the channel-wise attention features $\mathcal{F}_c$ as:

$$\mathcal{F}_c = \sigma(MLP(AvgPool_c(f_n{}'')) + MLP(MaxPool_c(f_n{}''))), \tag{5}$$

where $MaxPool_c$ and $AvgPool_c$ denote two channel-wise pooling operations, and MLP is the multi-layer perception with one hidden layer to generate the attention features. We also compute the spatial-wise attention features $\mathcal{F}_s$ as:

$$\mathcal{F}_s = \sigma(Conv_{7\times7}([AvgPool_s(f_n{}''); MaxPool_s(f_n{}'')])), \tag{6}$$

where $Conv_{7\times7}$ is a convolutional layer with kernel size 7. The final attention features $f_n{}^*$ are then computed as:

$$f_n{}^* = f_n{}'' \times \mathcal{F}_c + f_n{}'' \times \mathcal{F}_s, \tag{7}$$

where $\times$ denotes the dot product operation, and $+$ is the element-wise summation operation.

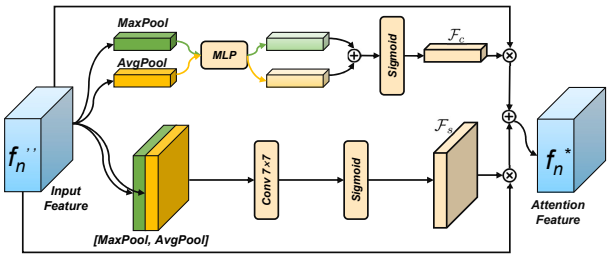


Fig. 8: Mutual Attention (MA) unit.

Figure 9 shows the effectiveness of the proposed CA module in enhancing the boundary and centroid detection performances.

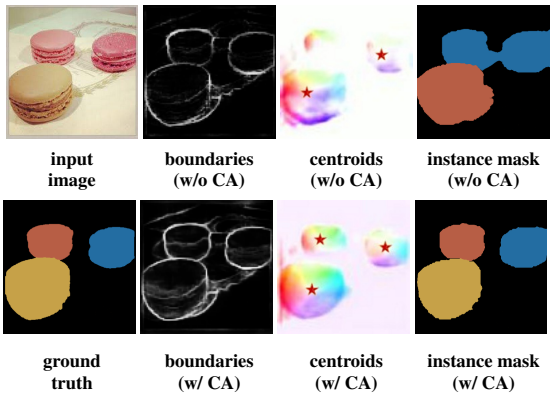


Fig. 9: Illustration on how the CA module benefits both boundary and centroid tasks. We can see that the CA module influence both the continuity of the detected boundaries and the accuracy of detected centroids.

3.4 Progressive Training Scheme (PTS)

Weak annotated labels would inevitably introduce noise into the learning process. To reduce the noise, previous works propose to use temporal ensemble learning [23,44] in a semi-supervised setting, where latent knowledge learned from labeled data can be applied to noisy unlabeled data. We extend this idea in our weakly-supervised setting. Since we do not have fully-annotated data, we explore this ensemble learning strategy in a progressive self-supervised manner, *i.e.*, by reciprocally training the salient object detection branch using newly predicted salient instance maps. It mainly contains two iterative steps: pseudo label generation, and model refreshing. Figure 10 shows the overview of the proposed Progressive Training Scheme, and Algorithm 1 summarizes the main steps.

3.4.1 Pseudo Label Generation

Since we do not have any fully-annotated labels to learn a noise-free representation, we propose to use our WSID-Net as a pseudo-label generator and refine its salient object detection branch in a self-supervised manner. This is because the output of our WSID-Net has more accurate boundaries with the help of other two branches, compared with its salient object detection branch. On the other hand, the retrained saliency detection branch can further boost the performance of our WSID-Net due to the improved salient object localization. To this end, we first forward the WSID-Net to generate salient instance masks, and use them as the initial pseudo training labels to update the network parameters of the salient object detection branch. In the next training iteration, we update the pseudo training labels using the retrained WSID-Net.

3.4.2 Model Refreshing

Before we update the pseudo training labels in the next iteration, we need to refresh our salient object detection branch. Note that we do not have any clean data (fully-annotated labels in our case) that can be used for learning noise-free features. It is possible that our model may overfit the noise in the pseudo labels and converge to a local minimum. To avoid these problems, we adopt the Exponential Moving Average method to update the model parameters with a weighted sum of model parameters in the current and former iterations. We define ω_r as the model weight in iteration r, and the model refreshing is formulated as:

$$\omega_{r+1} = \alpha\omega_{r-1} + (1-\alpha)\omega_r, \tag{8}$$

where α is the smoothing hyper-parameter that balances the contributions of model parameters from different iterations.

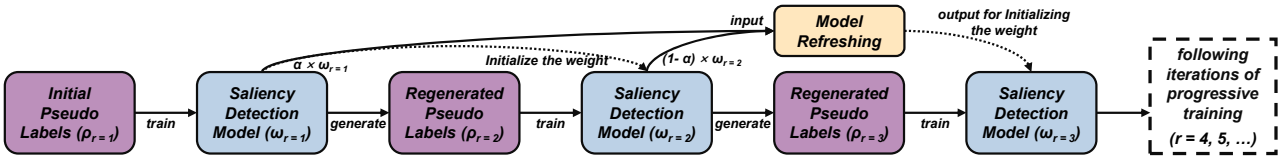
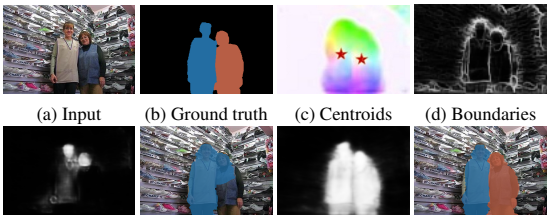


Fig. 10: Progressive Training Scheme for our saliency detection branch. Each training iteration in the dashed box contains two steps, model refreshing and pseudo label regeneration, leading the model to correct the noise in a self-supervised manner.

Algorithm 1 : Progressive Training Scheme (PTS).

Input: Initial pseudo label ρ_1, initial model weight ω_1, training iterations R, and training epochs E per iteration
Output: Final trained model weight ω_f

```
for r = 1 to R do
    for e = 1 to E do
        update model weight ω_r via backpropagating gradients and learning from pseudo labels ρ_r
    end for
    if r > 2 then
        update model weight ω_r using Eq. 8
    end if
    generate pseudo label ρ_{r+1} using WSID-Net, where the saliency branch uses model weight ω_r, and further uses CRF to refine the boundaries of the pseudo labels
end for
ω_f = ω_R
```



(a) Input (b) Ground truth (c) Centroids (d) Boundaries

(e) Saliency map (w/o PTS) (f) Instance mask (w/o PTS) (g) Saliency map (w/ PTS) (h) Instance mask (W/ PTS)

Fig. 11: Illustration of how the proposed PTS benefits our task progressively in a self-supervised manner. While WSID-Net initially fails due to inaccurate detection (e), the output of WSID-Net (f) is still better than (e) with the help of the centroid (c) and boundary (d) detections. By using (f) to further refine the saliency detection branch (g), our WSID-Net could also enjoy the performance boost (h).

Figure 11 shows one example of how the proposed PTS benefits our task in a self-supervised manner: given a challenging input image of two persons leaning on each other (a), our saliency detection branch fails to detect the majority of the salient regions due to the diverse foreground appearance and cluttered background (e), which further causes the failure of our WSID-Net (f). However, since (f) is still better than (e) due to the help of the centroid and boundary detections ((c) and (d)), after we have applied the proposed PTS training strategy, we can see that our saliency detection branch enjoys a better performance in detecting the saliency region as a whole (g), leading to an accurate salient instance detection result (h).

4 Experiments

4.1 Implementation Details

We implement WSID-Net on the Pytorch framework [36]. Both training and testing are performed on a PC with an i7 4GHz CPU and a GTX 1080Ti GPU. CRF is used to generate or refine pseudo labels. The hyper parameters of CRF are set as $w_1 = 4.0$, $w_2 = 3.0$, $\sigma_\alpha = 49.0$, $\sigma_\beta = 5.0$, and $\sigma_\gamma = 3.0$. We choose ResNet50 as the backbone for all three branches in WSID-Net. The backbone are initialized as in [40]. Input images are resized to 512×512 resolution. To minimize the loss function, we use the SGD optimizer with batch size 6 and initial learning rate 0.01. The learning rate decreases following poly policy ($lr_{itr} = lr_{init}(1 - \frac{itr}{max_itr})^\gamma$). We train our WSID-Net for 5 epoches. The proposed PTS training is further applied to refine the saliency detection branch for another 6 iterations, of which each iteration contains 8 epoches (R and E in Algorithm 1). α in Eq. 8 is set to $\frac{r}{r+1}$, where r is the current iteration index. The learning rate begins with 0.0001, and the decay follows the aforementioned poly policy.

4.2 Training and Evaluation Details

Datasets. Our WSID-Net is trained on two kinds of image-level labels, class and subitizing. We use the class labels from the PASCAL VOC 2012 [10] dataset (which is originally proposed for semantic and instance segmentation) to train our network. For subitizing labels, we count the numbers of salient instances from the ILSO [25] dataset, and train our network on this training set. For the proposed progressive training scheme, we augment the training data by using the unlabeled training data from C2SNet [26]. For testing, following the existing fully-supervised SID method [11], we perform SID evaluations on the test set of ILSO [25].

Training and Inference. We train the boundary and centroid branches together with different losses, and train the

Table 1: Quantitative evaluation of our method against six baseline methods and state-of-the-art fully-supervised SID methods. For the compared methods, we show their original tasks, supervision types, training labels and auxiliary pre-trained models in the $2nd$ to $5th$ columns. SID, SOD, OD, and SIS represent salient instance detection, salient object detection, object detection and semantic instance segmentation, respectively. FS and WS denote Fully-Supervised and Weakly-Supervised. Best performances among the weakly-supervised methods are marked in **red**.

Methods	Original task	Supervision types	Training labels	Auxiliary models	mAP @0.5↑	mAP @0.7↑
MSRNet [25]	SID	FS	object-level and instance-level pixel masks	MAP [62], MCG [5]	65.3%	52.3%
MAP [62]	SID	FS	instance-level bounding boxes	N/A	56.6%	24.8%
S4Net [11]	SID	FS	instance-level pixel masks	N/A	82.2%	59.6%
C2SNet [26]	SOD	WS	unlabeled images	CEDN [58], MAP [62], MCG [5]	41.1%	25.4%
NLDF [33]	SOD	WS	object-level pixel masks	MAP [62], MCG [5]	45.5%	24.5%
DeepMask [38]	OD	WS	instance-level bounding boxes	N/A	37.1%	20.5%
PRM+D [9]	SIS	WS	class, subitizing labels	MCG [5]	49.6%	31.2%
IRN [2]	SIS	WS	class labels	N/A	57.1%	37.4%
Ours	SID	WS	class, subitizing labels	N/A	**68.3%**	**51.7%**

saliency branch independently. We train the centroid detection branch using the proposed centroid-based subitizing loss together with the centroid loss introduced in [2,34]. We train the boundary detection branch using the boundary loss introduced in [3,2]. To train the saliency detection branch, we follow existing weakly-supervised SOD methods to use pseudo masks derived from class labels. Specifically, we first compute class activation maps via [70]. We then feed these maps together with the input image to a Conditional Random Field (CRF) [22] to generate pseudo object maps, and use these pixel-level pseudo labels to train the saliency detection branch. We further utilize the proposed PTS with model refreshing and self-generated pseudo labels to retrain the saliency detection branch.

During inference, given an input image, WSID-Net first computes the centroids, boundaries, and saliency maps. We first obtain the initial saliency instance map $\mathcal{SI}^*$ via the saliency map and centroid map, as discussed in Section 3.1.1. We then use the boundary map to refine the initial saliency instance map with the random walk algorithm. The transition probability matrix $\mathcal{M}$ is defined as:

$$\mathcal{M} = \mathcal{D}^{-1}\mathcal{H}^{\chi}, \tag{9}$$

where $\mathcal{H}$ is the affinity matrix of the learned boundary map $\mathcal{B}$, and $\mathcal{D}$ is a diagonal matrix relating to $\mathcal{H}$. The element in $\mathcal{H}$ is defined as: $h_k = 1 - \max_{k \in \Pi_{ij}} \mathcal{B}(x_k)$, where Π_{ij} is a set of pixels on the line between boundary pixels x_i and x_j. In addition, $\mathcal{H}^{\chi}$ is the self production of $\mathcal{H}$ with power χ for affinity distillation, and $\mathcal{D}$'s diagonal element $\mathcal{D}_{ii}$ equals to $\sum h_{ij}^{\chi}$ for summarizing values of $\mathcal{H}^{\chi}$ by row. The random walk algorithm for instance-wise saliency value propagation is conducted as:

$$vec(\overline{\mathcal{SI}}_n^*) = \mathcal{M}^i vec(\mathcal{SI}_n^*(1-\mathcal{B})), \tag{10}$$

where $vec()$ refers to the vectorization of the matrix, and $\overline{\mathcal{SI}}_n^*$ is our final saliency instance map.

Evaluation Metrics. We use the mean Average Precision (mAP) metric [14] to evaluate the SID performance. The IoU is set to 0.5 and 0.7 for this metric.

4.3 Comparing to the State-of-the-art Methods

As we are the first to propose a weakly-supervised SID method, we compare our method to 2 existing fully-supervised state-of-the-art SID methods: S4Net [11] and MSRNet [25]. We also prepare the following baselines from related tasks for evaluation. We choose 6 state-of-the-art weakly-supervised methods, with two from the SOD task C2SNet [26] and NLDF [33]; one from the SID task MAP [62]; one from the object detection (OD) task, DeepMask [38]; and two from the Semantic Instance Segmentation task, PRM+D [9] and IRN [2]. We adapt them by adding different post-processing strategies to these methods for deriving instance-level saliency maps from their original outputs, or modifying their networks and retrain them using our training data. Details are summarized as follows:

– We choose "MCG [5] + MAP [62]" as the post-processing strategy for the weakly-supervised SOD methods (*i.e.*, C2SNet [26] and NLDF [33]), inspired by the fully-supervised SID method MSRNet [25]. It has been shown in [25] that MCG [5] can be used to produce

segments given the contour maps as input and assign these segments with confidence scores. Segments with low confidences can then be filtered out by MAP [62]. Considering that both C2SNet [26] and NLDF [33] learn to produce contour maps, we find this post-process strategy suitable for weakly-supervised SOD methods with contour predictions.
- We select CRF [22] as the post-processing strategy for fully-supervised bounding-box-based SID method MAP [62], due to the fact that CRF is a popular graphical model used as post-processing for boosting segmentation performance. Considering that MAP [62] can generate instance-level bounding boxes, CRF can be used to obtain instance maps by refining the boundaries, which gives a performance boost of 3%.
- We choose a weakly-supervised SOD method as post-processing for filtering out non-salient segments produced by DeepMask [38], as DeepMask [38] is a class-agnostic object detection method that is not aware of saliency information. We choose WSS [47] as the weakly-supervised SOD method for a fair comparison, as it performs closely to our Saliency Detection Branch in our preliminary experiment. This strategy improves the performance by 5%.
- IRN [2] produces class-specific instance segmentation maps, which do not have saliency information. To adapt its results from class-specific to class-agnostic, we remove the CAM in their method and directly use their centroid and boundary maps to obtain instance maps. We then utilize WSS [47] to select salient instances.
- PRM+D [9] is trained with class and per-class subitizing labels to predict semantic instance maps. However, this method can only response to the instances with predefined class labels. To adapt it to class-agnostic, we merge its per-class outputs (originally 20 output maps for 20 classes) into one class-agnostic map by adding an additional convolutional layer, and then retrain it using our training data.

Quantitative Comparisons. We quantitatively evaluate our method in Table 1[†]. It is worth noting that our method achieves a significantly better performance of about 20% over the second-place weakly-supervised baseline, on the mAP@0.7 metric (which is very challenging as it requires the IoU value to be over 70%). These results show that our method achieves the best performance using just two image-level labels.

[†] As of today, the codes for MSRNet [25] are still not available. Following [11], we directly copy the numbers reported in [25] to our submission for a quantitative comparison.

Qualitative Comparisons. We further qualitatively evaluate our method with fully-supervised methods and baselines in Figure 12. The visual results verify that our method is able to delineate the instance boundaries clearly, and output accurate numbers of segmented salient instances directly for different scenes, *i.e.* scenes with single instances, small instances, (non-)adjacent instances, similar/varied instances, and cluttered contents. In contrast, the compared methods exhibit different limitations as follows:

- PRM+D and IRN fail to detect integral instances with inferior detected boundaries (*e.g.*, rows 1, 13).
- C2SNet and NLDF tend to recognize texture boundaries, causing fragmented instances (*e.g.*, rows 10, 12, 14).
- DeepMask and S4Net suffer from the over-detection problem, as they fail to distinguish instance proposals belonging to the same instance (*e.g.*, rows 2, 3, 13).
- MAP is a bounding-box based method. It fails to get clear instance boundaries even post-processed by CRF (*e.g.*, rows 1, 2, 7).

Overall, our method outperforms all these baselines, as a result of the centroid-based subitizing loss, the carefully designed BE and CA modules, and the progress training scheme.

4.4 Internal Analysis and Discussions

4.4.1 Ablation Study of Network Design

We begin by investigating the effectiveness of the proposed network design, including the proposed Boundary Enhancement module, Cross-layer Attention module, Progressive Training Scheme, and $\mathcal{L}_{\mathcal{SU}}$ loss. Table 2 shows the results. We can see that the SID performance would drop if we remove any of the components from the network. This shows that these components can help boost the performances of the saliency, centroid and boundary detection sub-tasks, which play a vital role in detecting salient instances. Figures 3, 5, and 9 provide additional visual comparisons to demonstrate the effectiveness of these components.

Table 2: Ablation study of network design.

method	**mAP@0.5↑**	**mAP@0.7↑**
Ours (w/o CA, BE, PTS, $\mathcal{L}_{\mathcal{SU}}$)	57.1%	37.4%
Ours (w/o CA)	64.3%	48.4%
Ours (w/o BE)	65.2%	48.9%
Ours (w/o $\mathcal{L}_{\mathcal{SU}}$)	62.1%	46.9%
Ours (w/o PTS)	63.9.%	47.2%
Ours	**68.3%**	**51.7%**

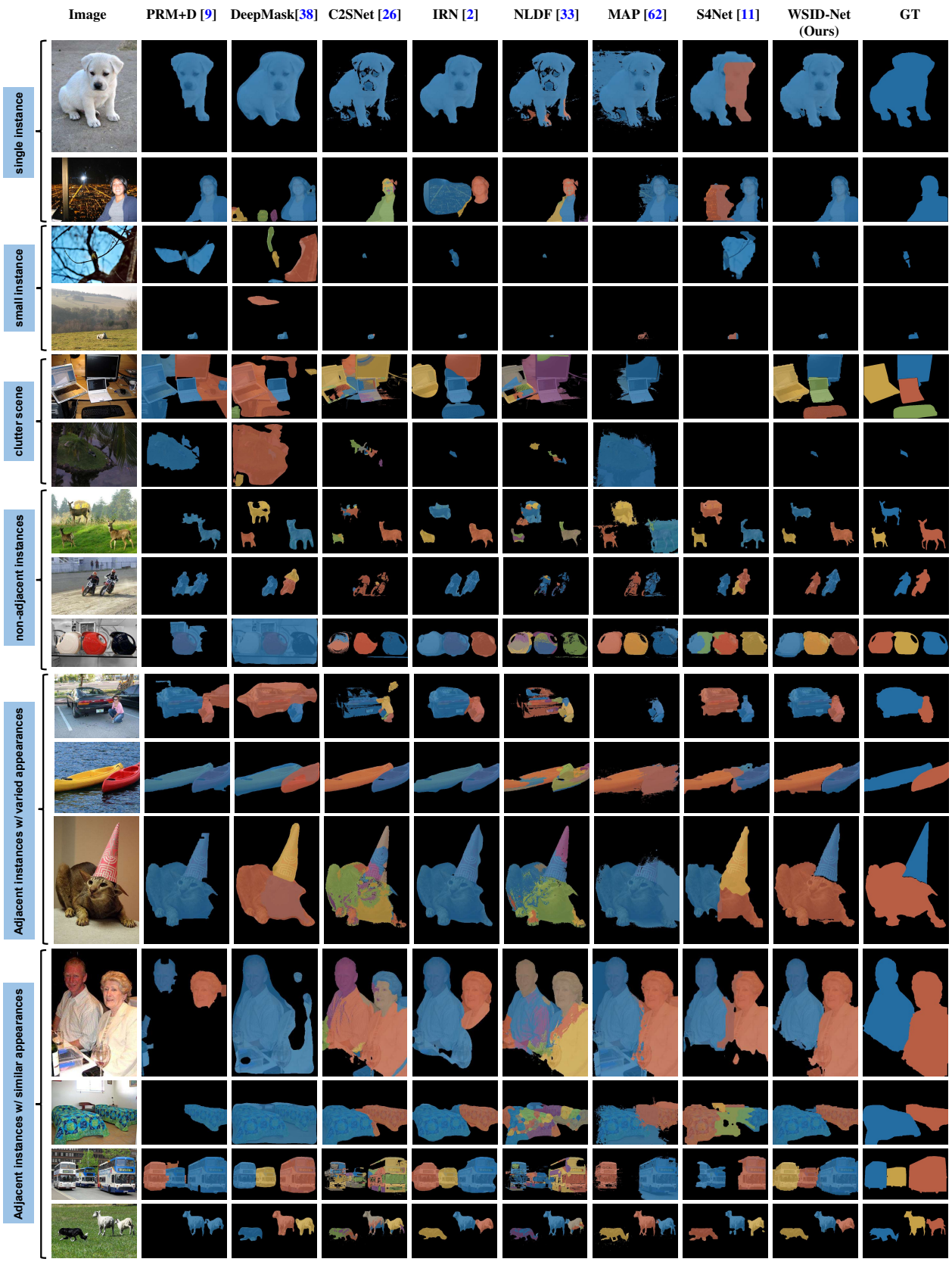


Fig. 12: Qualitative results of our method, compared with existing fully-supervised methods (S4Net[11] and MAP [62]) and modified baselines (PRM+D [9], DeepMask [38], C2SNet [26], NLDF [33], and IRN [2]). Refer to Section 4.3 and Table 1 on how we modify and train these baselines, in order to carry out a fair comparison.

4.4.2 Evaluation of the CA Module

We then investigate our CA module on its design choices and intermediate feature visualization.

Design Choices: we examine our CA module against its variants, as reported in Table 3. First, we study the benefits of the CFM and MA units. Row 1 shows that removing the CFM unit leads to a performance drop, due to the reduction in communication enhancement among the pyramid feature layers. Row 2 also shows a performance drop without the MA unit, as we are not able to learn long-range dependencies. Second, we study the connection styles (parallel and cascade) of the attention mechanisms in the MA unit. Rows 3, 4 and 5 show that parallel connection of the spatial- and channel-wise attentions performs better than the cascade one. This may be because cascade connection may lose the learned context information of the former attention mechanism.

Table 3: Evaluation of different designs of the CA module. s→c represents using spatial-wise attention before channel-wise attention, while c→s represents the reverse connection. The best performance among different designs is marked in **bold**.

method	mAP@0.5↑	mAP@0.7↑
w/o CL, w/ parallel MA	66.1%	49.5%
w/ CL, w/o parallel MA	66.8%	50.2%
w/ CL, w/ cascade (s→c) MA	67.4%	51.4%
w/ CL, w/ cascade (c→s) MA	67.2%	51.0%
Ours (w/ CL, w/ parallel MA)	**68.3%**	**51.7%**

Feature Visualization: we visualize multi-level intermediate features learned by our CA module in Figure 13. Given multi-level backbone features $f_1 \sim f_5$ (1^{st} row), the CFM unit in the CA module first generates multi-level mixed context features $f_1^* \sim f_5^*$ (2^{nd} row), which are then used for learning boundary features $f_{1\Rightarrow\mathcal{B}}^* \sim f_{5\Rightarrow\mathcal{B}}^*$ (3^{rd} row) and centroid features $f_{1\Rightarrow\mathcal{V}}^* \sim f_{5\Rightarrow\mathcal{V}}^*$ (4^{th} row), respectively. First, we can see that our CFM unit is able to highlight the salient objects via aggregating multi-level backbone features, as shown in row 2. Second, the boundary-aware feature maps have high responses in different regions as shown in row 3, which suggests that determining the instance boundaries also require multi-level features. Third, in row 4, the visualization of centroid-aware features generally corresponds to the centroid map $\mathcal{V}$, where the instance boundaries are generally highlighted, and pixel values of the centroid locations are close to zero. Overall, our proposed CA module is able to help adapt the backbone features into different task-specific features.

In addition, our CA module differs from CAM in two aspects. First, CAM is conditioned on the class label input, but our CA module learns class-agnostic attentions from pseudo labels. Second, CAM is unable to delineate clear boundary and instance information, while our CA module can learn this information to complement the CAM for detecting salient instances, as shown in Figure 14.

4.4.3 Evaluation of the BE Module

We conduct ablation studies to investigate the effect of the Canny filter in the Boundary Enhancement (BE) module. We compare our BE module to two ablated versions: removing the Canny filter from the BE module (denoted as BE Module w/o Canny), and using Canny filter only (denoted as Canny Only). Results are shown in Table 4. We can see that our method outperforms both ablated versions. The Canny filter is used to enrich the high-level boundary features with low-level edge information. Without the Canny filter, the BE module may not detect small boundaries accurately. However, relying only on the Canny filter cannot obtain high-level boundary information, which typically leads to over-segmentation of instances.

Table 4: Evaluation on the Canny filter. Best performances are marked in **bold**.

method	mAP@0.5↑	mAP@0.7↑
BE Module w/o Canny	65.0%	49.3%
Canny Only	63.7%	48.0%
Ours	**68.3%**	**51.7%**

4.4.4 Evaluation of Parameter settings for the Canny Filter

We empirically set the thresholds (*i.e.*, θ_{up} and θ_{low} for controlling the connectivity and density of the detected edges) in the Canny operator to be automatically determined by the channel median of the gray-scale image. We find that this works well in our experiments.

To further investigate how these two thresholds affect the performance, we provide both qualitative and quantitative comparisons between our threshold choice with two manual choices described below:

- Large range: we manually set θ_{low} and θ_{up} to 30 and 200, respectively, so that the Canny operator is sensitive to textures and can detect more edges.
- Small range with large values: we manually set θ_{low} and θ_{up} to 230 and 260, respectively, so that only structural edges of objects can be detected.

Table 5 shows that both manual strategies would degrade the performance. Figures 15 and 16 show two scenes that these manual strategies fail. In row 2 of Figure 15, if the Canny edge map provides insufficient object structure information and the learned boundaries are partially weak, our

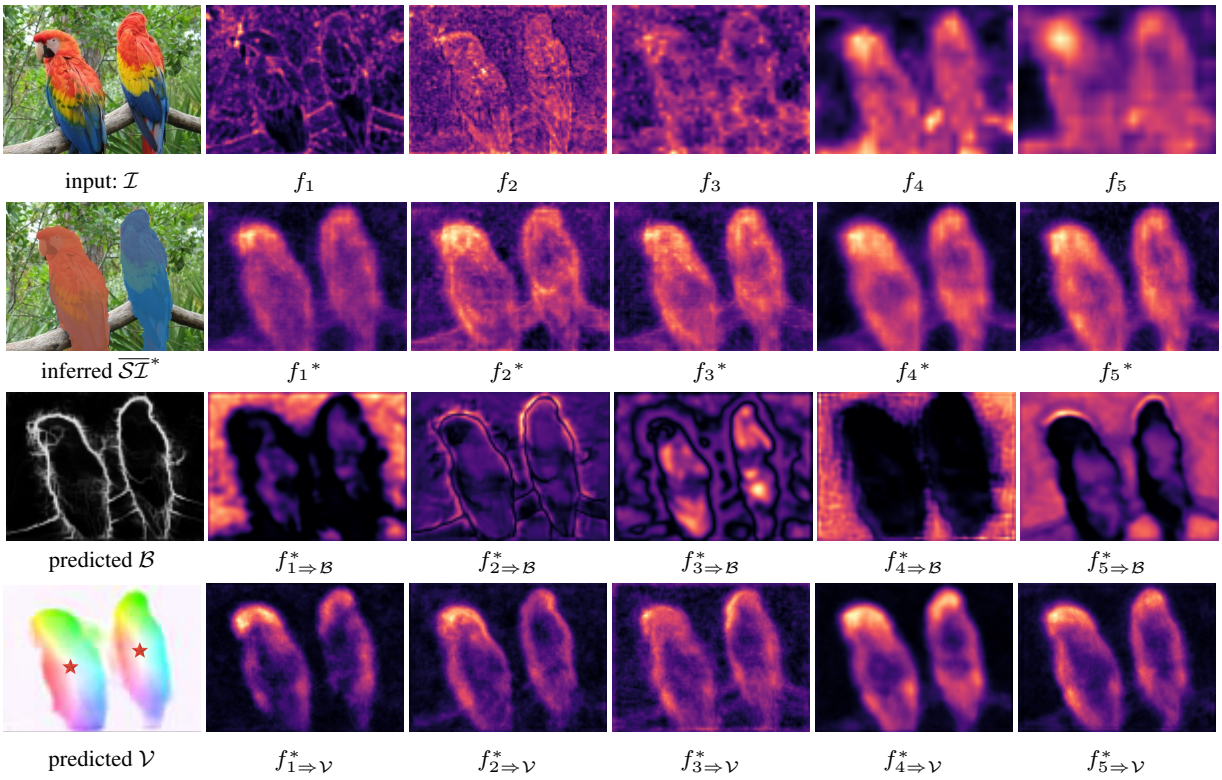


Fig. 13: Visualization of the multi-level intermediate features learned by our CA module.

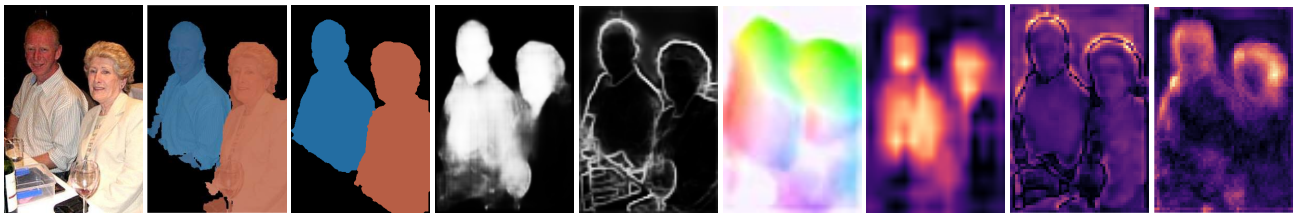


Fig. 14: Visual comparison between CAM and CA features. As shown in column 7, CAM itself cannot delineate the boundaries between two persons and locate their centroids. Hence, we only use CAM in the saliency detection branch. Our CA module successfully learns this information for complementing CAM.

Table 5: Evaluation on different parameter settings for the Canny operator in the BE module. Best performances are marked in **bold**.

settings of θ_{low} and θ_{up}	**mAP@0.5↑**	**mAP@0.7↑**
$\theta_{low} = 30$, and $\theta_{up} = 200$	67.4%	50.8%
$\theta_{low} = 230$, and $\theta_{up} = 260$	66.9%	50.2%
Ours	**68.3%**	**51.7%**

method fails to separate nearby instances. In row 1 of Figure 16, the Canny edge contains extensive non-structure textures that affect the learned boundaries, making it difficult for the saliency values to propagate to the target boundaries from the centroid for determining the instance. In contrast, our choice successfully detects accurate instances since we can obtain high-quality boundaries, as shown in rows 1 and 3 in Figure 15, and rows 2 and 3 in Figure 16. This visually verifies that the Canny edge generated under our automatic setting is more stable to provide pleasant instance boundaries.

4.4.5 Evaluation of PTS

We study how PTS helps improve the saliency detection performance iteration by iteration. Figure 17 and 18 show the progressively improving results over six training iterations.

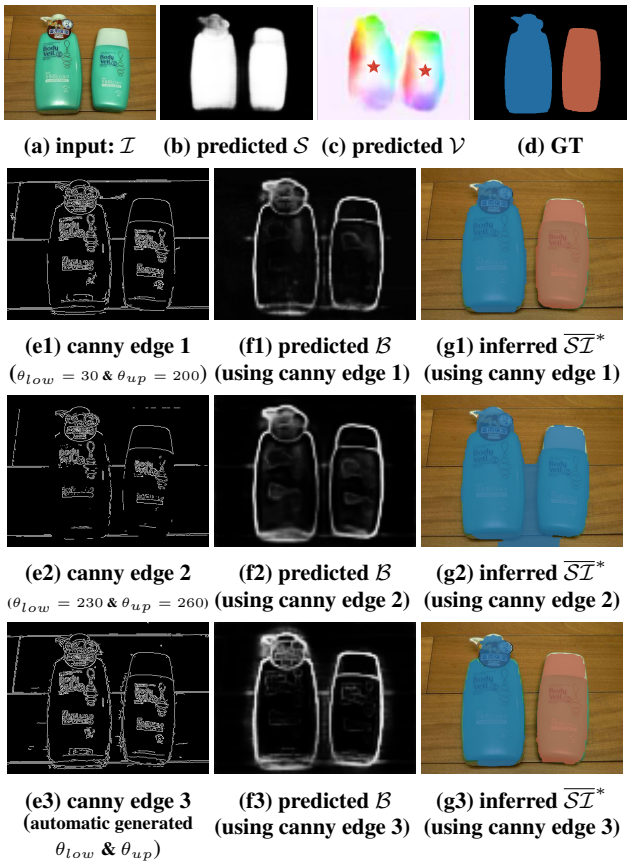


Fig. 15: Visual comparison between results using different parameters of the Canny operator.

Intermediate results in both figures verity that our PTS does not only penalize the background distraction but also recover the integral objects. Overall, our PTS is able to reduce noise and improve the performance in a self-supervised manner.

4.4.6 Evaluation of the Saliency Detection Branch

Since our WSID-Net relies on the performance of the saliency detection branch in detecting salient objects, we are particularly interested in the question of to what extent that the quality of the saliency object detection maps may affect the SID performance. To answer this question, we replace the outputs of our weakly-supervised salient object detection branch with five state-of-the-art full-supervised SOD methods (*i.e.*, DSS [48], PiCANet [29], EGNet [67], ITSD [71], and SCRN [55]), as well as the ground truth saliency maps, to generate the salient instance maps. Results are reported in Table 6. We can see that the performance generally increases when inferring salient instance masks using the fully-supervised SOD results. This is because the fully-supervised methods are more robust to background distractions and able to delineate full object masks. However, we can still observe that the performance would not be saturated

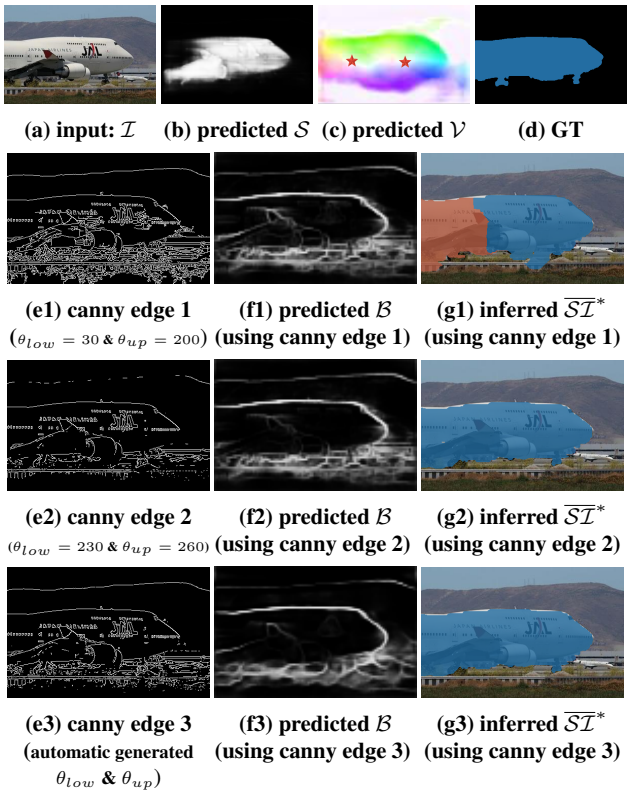


Fig. 16: Visual comparison between results using different parameters of the Canny operator.

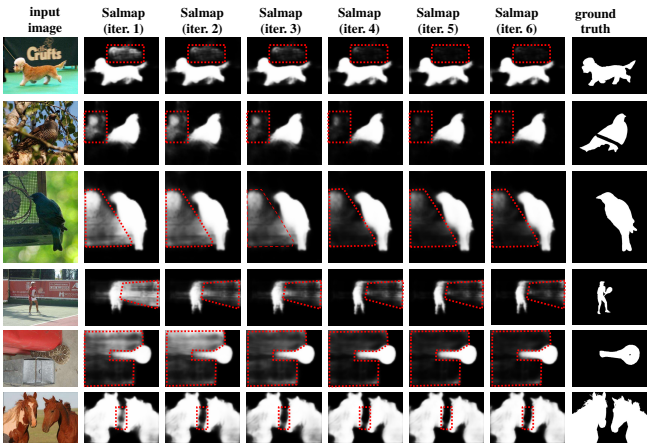


Fig. 17: Visualization of intermediate results over six training rounds. The dashed red regions denote background distraction noise. We can see that the noise is progressively suppressed over the iterations.

even if we feed the ground truth saliency maps to generate the SID maps. This is because the instance boundaries are still very difficult to detect, especially when these salient instances overlap each other. This suggests that developing an effective method for detecting salient instance boundaries in a weakly-supervised setting would be a promising solution.

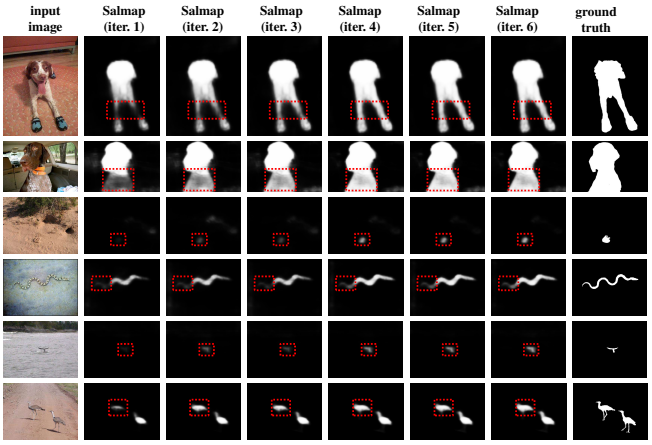


Fig. 18: Visualization of intermediate results over six training rounds. The dashed red regions denote missed salient parts that are progressively recovered over the iterations. We can see that the detected objects are becoming more and more complete.

Table 6: Investigation on how the SID performance is affected by the quality of the SOD maps. We show the performances by replacing the saliency maps (denoted Salmap) predicted by our saliency detection branch with saliency maps computed by different fully-supervised state-of-the-art SOD methods.

method	mAP@0.5↑	mAP@0.7↑
Salmap → GT	72.1%	58.3%
Salmap → DSS [48]	67.2%	54.3%
Salmap → PiCANet [29]	67.9%	53.8%
Salmap → EGNet [67]	69.3%	54.9%
Salmap → ITSD [71]	70.0%	56.4%
Salmap → SCRN [55]	69.2%	55.9%
Ours	68.3%	51.7%

4.4.7 Evaluation of CRF in the Saliency Branch

CRF is used to produce pseudo ground truth saliency maps given the coarse CAM activation maps, so that the saliency detection branch can learn more accurate boundary information. Figure 19 shows that CRF helps produce more accurate pseudo ground truth saliency maps. We also provide quantitative results in Table 7, from which we can see that the performance drops without CRF refinement.

Table 7: Evaluation of the effect of CRF to the saliency branch. Best performances are marked in **bold**.

method	mAP@0.5↑	mAP@0.7↑
w/ CRF	62.4%	43.1%
w/o CRF (Ours)	**68.3%**	**51.7%**

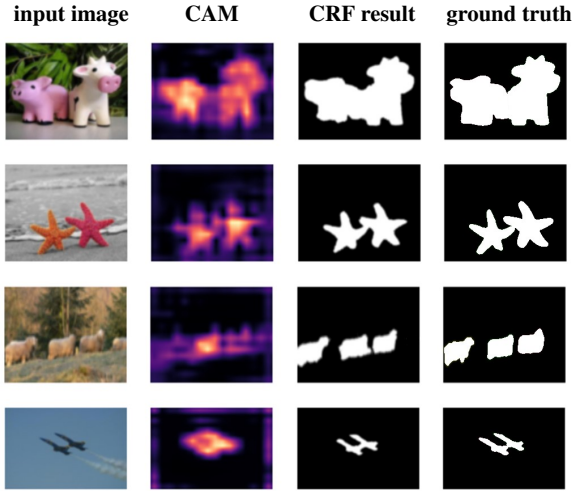


Fig. 19: Visualization of the effect of CRF on refining the coarse CAM.

5 Conclusion and Future Work

In this paper, we have proposed the first weakly-supervised SID method, called WSID-Net, which is trained on class and subitizing labels. Our WSID-Net learns to predict object boundaries, instance centroids, and salient regions. By using the proposed Boundary Enhancement module, Cross-layer Attention module, Progressive Training Scheme, and centroid-based subitizing loss, our method can identify and segment salient instances effectively. Both quantitative and qualitative experiments demonstrate the effectiveness of the proposed method compared with baseline methods.

Our method does have its limitation. It may fail when the images are taken with improper exposures. Therefore, our method cannot detect salient objects/instances with low contrast to their surroundings. As a future work, we are currently exploring the use of a discriminative network of generative adversarial learning to overcome this limitation. We would also like to extend this work for videos.

Acknowledgements This work was partly supported by NNSFC Grants 91748104, 61972067, 61632006, U1811463, U1908214, 61751203; the National Key Research and Development Program of China, Grant 2018AAA0102003; a General Research Fund from RGC of Hong Kong (RGC Ref.: 11205620); and a Strategic Research Grant from City University of Hong Kong (Ref.: 7005674).

References

1. Achanta, R., Hemami, S., Estrada, F., Süsstrunk, S.: Frequency-tuned salient region detection. In: CVPR (2009)
2. Ahn, J., Cho, S., Kwak, S.: Weakly supervised learning of instance segmentation with inter-pixel relations. In: CVPR (2019)

3. Ahn, J., Kwak, S.: Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In: CVPR (2018)
4. Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. IEEE PAMI (2012)
5. Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR (2014)
6. Chen, C., Sun, X., Hua, Y., Dong, J., Xv, H.: Learning deep relations to promote saliency detection. In: AAAI (2020)
7. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587 (2017)
8. Cheng, M.M., Mitra, N.J., Huang, X., Torr, P.H., Hu, S.M.: Global contrast based salient region detection. IEEE TPAMI (2014)
9. Cholakkal, H., Sun, G., Khan, F.S., Shao, L.: Object counting and instance segmentation with image-level supervision. In: CVPR (2019)
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)
11. Fan, R., Cheng, M.M., Hou, Q., Mu, T.J., Wang, J., Hu, S.M.: S4net: Single stage salient-instance segmentation. In: CVPR (2019)
12. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: CVPR (2019)
13. Gao, S.H., Tan, Y.Q., Cheng, M.M., Lu, C., Chen, Y., Yan, S.: Highly efficient salient object detection with 100k parameters. In: ECCV (2020)
14. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: ECCV (2014)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
16. He, S., Jiao, J., Zhang, X., Han, G., Lau, R.W.: Delving into salient object subitizing and detection. In: ICCV (2017)
17. Hou, Q., Cheng, M.M., Hu, X., Borji, A., Tu, Z., Torr, P.H.: Deeply supervised salient object detection with short connections. In: CVPR (2017)
18. Hu, M., Han, H., Shan, S., Chen, X.: Weakly supervised image classification through noise regularization. In: CVPR (2019)
19. Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., Li, S.: Salient object detection: A discriminative regional feature integration approach. In: CVPR (2013)
20. John, C.: A computational approach to edge detection. IEEE TPAMI (1986)
21. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015)
22. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: NeurIPS (2011)
23. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242 (2016)
24. Laradji, I.H., Vazquez, D., Schmidt, M.: Where are the masks: Instance segmentation with image-level supervision. In: BMVC (2019)
25. Li, G., Xie, Y., Lin, L., Yu, Y.: Instance-level salient object segmentation. In: CVPR (2017)
26. Li, X., Yang, F., Cheng, H., Liu, W., Shen, D.: Contour knowledge transfer for salient object detection. In: ECCV (2018)
27. Li, Y., Hou, X., Koch, C., Rehg, J.M., Yuille, A.L.: The secrets of salient object segmentation. In: CVPR (2014)
28. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
29. Liu, N., Han, J., Yang, M.H.: Picanet: Learning pixel-wise contextual attention for saliency detection. In: CVPR (2018)
30. Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., Shum, H.Y.: Learning to detect a salient object. In: CVPR (2007)
31. Liu, Y., Wang, P., Cao, Y., Liang, Z., Lau, R.W.: Weakly-supervised salient object detection with saliency bounding boxes. TIP (2021)
32. Lu, Z., Fu, Z., Xiang, T., Han, P., Wang, L., Gao, X.: Learning from weak and noisy labels for semantic segmentation. IEEE TPAMI (2016)
33. Luo, Z., Mishra, A., Achkar, A., Eichel, J., Li, S., Jodoin, P.M.: Non-local deep features for salient object detection. In: CVPR (2017)
34. Neven, D., Brabandere, B.D., Proesmans, M., Gool, L.V.: Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In: CVPR (2019)
35. Pang, Y., Zhao, X., Zhang, L., Lu, H.: Multi-scale interactive network for salient object detection. In: CVPR (2020)
36. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019)
37. Perazzi, F., Krähenbühl, P., Pritch, Y., Hornung, A.: Saliency filters: Contrast based filtering for salient region detection. In: CVPR (2012)
38. Pinheiro, P.O., Collobert, R., Dollár, P.: Learning to segment object candidates. In: NeurIPS (2015)
39. Shi, J., Yan, Q., Xu, L., Jia, J.: Hierarchical image saliency detection on extended cssd. IEEE TPAMI (2015)
40. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
41. Siris, A., Jiao, J., Tam, G.K., Xie, X., Lau, R.W.: Inferring attention shift ranks of objects for image saliency. In: CVPR (2020)
42. Su, J., Li, J., Zhang, Y., Xia, C., Tian, Y.: Selectivity or invariance: Boundary-aware salient object detection. In: ICCV (2019)
43. Tan, X., Xu, K., Ying, C., Yiheng, Z., Ma, L., Rynson, L.: Nighttime scene parsing with a large real dataset. IEEE TIP (2021)
44. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: NeurIPS (2017)
45. Tian, X., Xu, K., Yang, X., Yin, B., Lau, R.W.: Weakly-supervised salient instance detection. In: BMVC (2020)
46. Wang, B., Chen, Q., Zhou, M., Zhang, Z., Jin, X., Gai, K.: Progressive feature polishing network for salient object detection. In: AAAI (2020)
47. Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., Ruan, X.: Learning to detect salient objects with image-level supervision. In: CVPR (2017)
48. Wang, T., Zhang, L., Wang, S., Lu, H., Yang, G., Ruan, X., Borji, A.: Detect globally, refine locally: A novel approach to saliency detection. In: CVPR (2018)
49. Wang, W., Shen, J.: Deep cropping via attention box prediction and aesthetics assessment. In: ICCV (2017)
50. Wang, W., Shen, J., Porikli, F.: Saliency-aware geodesic video object segmentation. In: CVPR (2015)
51. Wang, W., Zhao, S., Shen, J., Hoi, S.C., Borji, A.: Salient object detection with pyramid attention and salient edges. In: CVPR (2019)
52. Wei, J., Wang, S., Huang, Q.: F^3net: Fusion, feedback and focus for salient object detection. In: AAAI (2020)
53. Wei, J., Wang, S., Wu, Z., Su, C., Huang, Q., Tian, Q.: Label decoupling framework for salient object detection. In: CVPR (2020)
54. Woo, S., Park, J., Lee, J.Y., So Kweon, I.: Cbam: Convolutional block attention module. In: ECCV (2018)
55. Wu, Z., Su, L., Huang, Q.: Stacked cross refinement network for edge-aware salient object detection. In: ICCV (2019)
56. Xu, Y., Xu, D., Hong, X., Ouyang, W., Ji, R., Xu, M., Zhao, G.: Structured modeling of joint deep feature and prediction refinement for salient object detection. In: ICCV (2019)
57. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: CVPR (2013)

58. Yang, J., Price, B., Cohen, S., Lee, H., Yang, M.H.: Object contour detection with a fully convolutional encoder-decoder network. In: CVPR (2016)
59. Yang, X., Xu, K., Chen, S., He, S., Yin, B.Y., Lau, R.: Active matting. In: NeurIPS (2018)
60. Zeng, Y., Zhuge, Y., Lu, H., Zhang, L., Qian, M., Yu, Y.: Multi-source weak supervision for saliency detection. In: CVPR (2019)
61. Zhang, D., Han, J., Zhang, Y.: Supervision by fusion: Towards unsupervised learning of deep salient object detector. In: ICCV (2017)
62. Zhang, J., Sclaroff, S., Lin, Z., Shen, X., Price, B., Mech, R.: Unconstrained salient object detection via proposal subset optimization. In: CVPR (2016)
63. Zhang, J., Xie, J., Barnes, N.: Learning noise-aware encoder-decoder from noisy labels by alternating back-propagation for saliency detection. In: ECCV (2020)
64. Zhang, J., Yu, X., Li, A., Song, P., Liu, B., Dai, Y.: Weakly-supervised salient object detection via scribble annotations. In: CVPR (2020)
65. Zhang, J., Zhang, T., Dai, Y., Harandi, M., Hartley, R.: Deep unsupervised saliency detection: A multiple noisy labeling perspective. In: CVPR (2018)
66. Zhang, L., Dai, J., Lu, H., He, Y., Wang, G.: A bi-directional message passing model for salient object detection. In: CVPR (2018)
67. Zhao, J.X., Liu, J., Fan, D.P., Cao, Y., Yang, J., Cheng, M.M.: Egnet: Edge guidance network for salient object detection. In: ICCV (2019)
68. Zhao, T., Wu, X.: Pyramid feature attention network for saliency detection. In: CVPR (2019)
69. Zhao, X., Pang, Y., Zhang, L., Lu, H., Zhang, L.: Suppress and balance: A simple gated network for salient object detection. In: ECCV (2020)
70. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR (2016)
71. Zhou, H., Xie, X., Lai, J.H., Chen, Z., Yang, L.: Interactive two-stream decoder for accurate and fast saliency detection. In: CVPR (2020)
72. Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q., Jiao, J.: Weakly supervised instance segmentation using class peak response. In: CVPR (2018)
73. Zhu, Y., Zhou, Y., Xu, H., Ye, Q., Doermann, D., Jiao, J.: Learning instance activation maps for weakly supervised instance segmentation. In: CVPR, pp. 3116–3125 (2019)
74. Zhuge, Y., Yang, G., Zhang, P., Lu, H.: Boundary-guided feature aggregation network for salient object detection. IEEE Signal Processing Letters (2018)

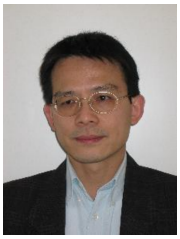
Author Biographies

Xin Tian is a PhD student in the Department of Computer Science at Dalian University of Technology and City University of HongKong. His research interests include salient object detection and image restoration.

Ke Xu is currently with the Department of Computer Science at City University of Hong Kong. He obtains the dual Ph.D. degrees from Dalian University of Technology and City University of Hong Kong. His research interests include deep learning, object detection, and image enhancement and editing.

Xin Yang is a professor in the Department of Computer Science at Dalian University of Technology, China. Xin received his B.S. degree in Computer Science from Jilin University in 2007. From 2007 to June 2012, he was a joint Ph.D. student in Zhejiang University and UC Davis for Graphics, and received his Ph.D. degree in July 2012. His research interests include computer graphics and robotic vision.

Baocai Yin is a professor of computer science department at the Dalian University of Technology and the dean of Faculty of Electronic Information and Electrical Engineer. His research concentrates on digital multimedia and computer vision. He received his B.S. degree and Ph.D. degree in computer science, both from Dalian University of Technology.



Rynson W.H. Lau received his Ph.D. degree from University of Cambridge. He was on the faculty of Durham University and is now with City University of Hong Kong.
Rynson serves on the Editorial Board of the International Journal of Computer Vision (IJCV) and Computer Graphics Forum. He has served as the Guest Editor of a number of journal special issues, including ACM Trans. on Internet Technology, IEEE Trans. on Multimedia, IEEE Trans. on Visualization and Computer Graphics, and IEEE Computer Graphics & Applications. He has also served in the committee of a number of conferences, including Program Co-chair of ACM VRST 2004, ACM MTDL 2009, IEEE U-Media 2010, and Conference Co-chair of CASA 2005, ACM VRST 2005, ACM MDI 2009, ACM VRST 2014. Rynson's research interests include computer graphics and computer vision.