3SD: Self-Supervised Saliency Detection With No Labels

Rajeev Yasarla* [1,2], Renliang Weng[2], Wongun Choi[2], Vishal M. Patel[1†],
Amir Sadeghian[2]
[1] Johns Hopkins University, [2] AIBEE
https://github.com/rajeevyasarla/3SD

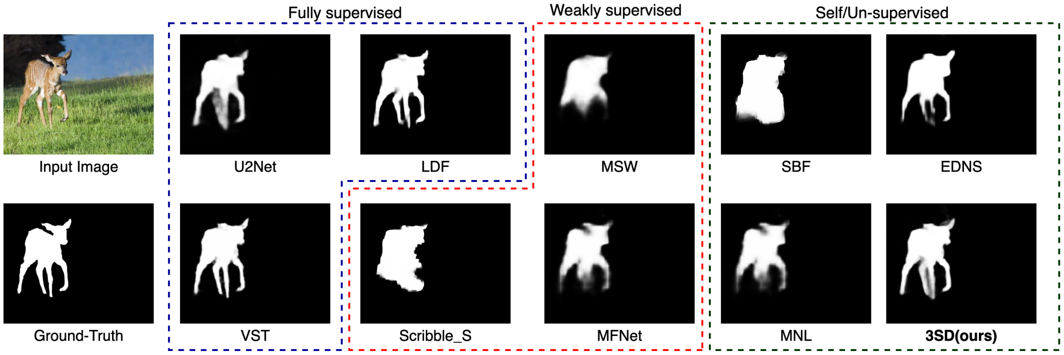

Figure 1. Comparisons of saliency maps. Self-supervised: **3SD (ours)**. Weakly supervised: MSW [50], Scribble [54], MFNet [32]. Unsupervised: SBF [51], MNL [55], EDNS [53]. Fully-supervised: U2Net [33], LDF [43], VST [25].

Abstract

*We present a conceptually simple self-supervised method for saliency detection. Our method generates and uses pseudo-ground truth labels for training. The generated pseudo-GT labels don't require any kind of human annotations (*e.g.*, pixel-wise labels or weak labels like scribbles). Recent works show that features extracted from classification tasks provide important saliency cues like structure and semantic information of salient objects in the image. Our method, called 3SD, exploits this idea by adding a branch for a self-supervised classification task in parallel with salient object detection, to obtain class activation maps (CAM maps). These CAM maps along with the edges of the input image are used to generate the pseudo-GT saliency maps to train our 3SD network. Specifically, we propose a contrastive learning-based training on multiple image patches for the classification task. We show the multi-patch classification with contrastive loss improves the quality of the CAM maps compared to naive classification on the entire image. Experiments on six benchmark datasets demonstrate that without any labels, our 3SD method outperforms all existing weakly supervised and unsupervised methods, and its performance is on par with the fully-supervised methods.*

1. Introduction

Salient object detection (SOD) task is defined as pixel-wise segmentation of interesting regions that capture human attention in an image. It is widely used as a prior to improve many computer vision tasks such as visual tracking, segmentation, *etc.* Early methods based on hand-crafted features like histograms [27], boundary connectivity [60], high-dimensional color transforms [20], may fail in producing high-quality saliency maps on cluttered images where the foreground object is similar to the background. In recent years, deep convolutional neural networks (CNNs), and in particular fully convolutional networks (FCN) [26] have provided excellent image segmentation and salient object detection performance.

In general, the CNN-based salient object detection methods can be classified into three groups: (i) fully-supervised methods (that require large-scale datasets with pixel-wise

[*] This work was done during an internship at AIBEE.
[†] Vishal M. Patel was supported by NSF CAREER award 2045489

annotations), (ii) weakly supervised, and (iii) unsupervised or self-supervised methods (that don't require actual pixel-wise annotations of salient object detection). The main drawback of the fully-supervised methods [25,33,34,43,45] is that they require a large amount of pixel-wise annotations of salient objects which is time-consuming and expensive. On the other hand, to minimize human efforts in labeling datasets, weakly-supervised approaches [21,50] have been proposed which address saliency detection either by using weak sparse labels such as image class labels [21] or image captions [50]. Alternatively, [54] present a weakly-supervised SOD method based on scribble annotations. [32] propose a learnable directive filter based method that extracts saliency cues using multiple labels from the attentions. Note, [32] and MSW [50] rely on the features or attention maps obtained from a classification task, and might fail to produce high-quality pseudo-GTs since the classification task is trained with global class label or image caption. For example, Fig. 1 shows that the outputs of [32, 50] are not sharp and miss fine details like legs and ears. Although these weakly-supervised methods reduce the amount of labeling required for SOD, they still require labeling resources to obtain image captions [50], image class labels [21, 32], or accurate scribble annotations [54]. On the other hand, unsupervised methods [6,29,30,35,41,42,53,55] devise a refinement procedure or generative based approach (noise-aware), that utilizes the hand crafted features, and/or noisy annotations. Note that performance of these unsupervised methods highly rely on the noisy annotation, and might struggle to produce high-quality saliency maps if they fail to recover the underlying semantics from the noisy annotations. For instance, we can observe in Fig. 1 saliency outputs of [30, 53, 55] miss parts like legs and ears.

In an attempt to overcome these issues, we propose our Self-Supervised Saliency Detection (3SD) method. Our framework follows the conventional encoder-decoder structure to generate saliency map. In terms of encoder, we present a novel encoder architecture which consists of a local encoder and global context encoder. The local encoder learns pixel-wise relationship among neighbourhood while the global encoder encodes the global context. The outputs of these encoders are concatenated and subsequently fed to the decoder stream. By fusing both local features and global context we are able to extract both fine-grain contour details as well as adhere to the underlying object structure.

For the decoder, we follow the literature [3] by adding an auxiliary classification task to capture important saliency cues like semantics and segmentation of the salient object in the image, which can be extracted in the form of class activation map (CAM map). However, performing a self-supervised classification with single global class label might result in low-quality class activation map (illustrated in Fig. 2f, where the CAM map is incomplete). To address this issue, we propose a contrastive learning [5] based patch-wise self-supervision for the classification task, where we perform patch-wise (32 × 32 pixels in our implementation) classification and train it with proposed self-supervised contrastive loss. Specifically, positive patches (patches similar to the salient object) and negative patches (patches dissimilar to the salient object) are identified. Positive patches are pulled together, and they are pushed away from the negative patches. In this way, the network strives to learn the attentions or semantic information that are responsible for classifying the salient object at a fine-grain patch level.

With the novel designs of encoder and decoder, our 3SD is able to generate high-quality CAM maps (see Fig. 2). While the generated CAM maps provide salient object information, they might not have proper boundary corresponding to the salient object. To deal with this issue, we fuse CAM map with a gated edge map of the input image to generate the pseudo-GT salient map (Fig. 2d).

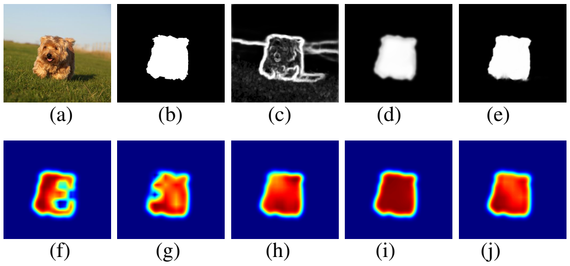

Figure 2. The CAM map comparisons with different patch-wise auxiliary classification tasks: (a) input image, (b) ground-truth, (c) edge map of the input image, (d) pseudo-GT computed using our 3SD method, (e) saliency map computed using 3SD where we use 32 × 32 pixel-wise classification, (f), (g), (h), (i), (j) are CAM-maps computed when auxiliary classification task is global class label, patchwise classification with patch size 16 × 16, 24 × 24, 32 × 32, 48 × 48 respectively.

Fig. 1 compares sample results of the proposed 3SD with the existing SOTA weakly, unsupervised [50,51,53–55] and SOTA fully-supervised [43] methods. One can clearly observe that [50, 51, 53–55] fail to produce sharp edges and proper saliency maps. In contrast, our method is able to provide sharper and better results. To summarize, the main contributions of our paper are as follows:

- We propose a self-supervised 3SD method that requires no human annotations for training an SOD model. Our 3SD method is trained using high-quality pseudo-GT saliency maps generated from CAM maps using a novel self-supervised classification task.

- We present a patch-wise self-supervised contrastive learning paradigm, which substantially improves the quality of pseudo-GT saliency maps and boosts 3SD performance.

- We construct a novel encoder architecture for SOD that attends features locally (pixel-level understanding), and globally (patch-level understanding).
- Extensive experiments on five benchmark datasets show that the proposed 3SD method outperforms the SOTA weakly/unsupervised methods.

2. Related Work

Classical image processing methods address SOD using histograms [27], boundary connectivity [60], high-dimensional color transforms [20], hand-crafted features like foreground consistency [52], and similarity in superpixels [57]. In recent years, various supervised CNN-based methods have been proposed for SOD [12, 24, 28, 33, 34, 40, 43–45, 56, 58] which extensively study architectural changes, attention mechanisms, multi-scale contextual information extraction, boundary-aware designs, label decoupling, *etc*. In contrast, our 3SD method is a self-supervised method trained using pseudo-GT data. In what follows, we will review recent weakly/unsupervised SOD methods as well as the self-supervised methods.

Weakly supervised SOD methods: To reduce human efforts and expenses in pixel-level labeling and annotations, various weakly methods have been proposed. These methods use high-level labels such as image class labels and image captions [21, 50], and scribble annotations [54]. [7], [19] follow a bounding-box label approach to solve weakly-supervised segmentation task. [39] extract cues using image-level labels for foreground salient objects. [18] propose a category-based saliency map generator using image-level labels. [4, 21, 31] propose a CRF-based method for weakly supervised SOD. [50] train a network with multiple source labels like category labels, and captions of the images to perform saliency detection. [54] introduce scribble annotations for SOD. Unlike these methods, the proposed 3SD method doesn't require any kind of human annotations, noisy labels, scribble annotations, or hand-crafted features to perform SOD.

Unsupervised SOD methods: [51] devise a fusion process that employs unsupervised saliency models to generate supervision. [30] propose an incrementally refinement technique that employs noisy pseudo labels generated from different handcrafted methods. [55] and [53] propose a saliency prediction network and noise modeling module that jointly learn from the noisy labels generated from multiple "weak" and "noisy" unsupervised handcrafted saliency methods. Unlike these unsupervised methods that highly rely on the noisy annotations, we propose a novel pseudo-GT generation technique using patchwise contrastive learning based self-supervised classification task.

Self-supervised/Contrastive learning methods: Several approaches explore discriminative approaches for self-supervised instance classification [9, 46]. These methods treat each image as a different class. The main limitation of these approaches is that they require comparing features from images for discrimination which can be complex in a large-scale dataset. To address this challenge, [13] introduce metric learning, called BOYL, where better representations of the features are learned by matching the outputs of momentum encoders. Subsequently, [3] propose the DINO method that is based on mean Teacher [37] self-distillation without labels. Recently, [5, 15, 16, 38] propose self-supervised contrastive learning based methods where the losses are inspired by noise-contrastive estimation [14], triplet loss [17], and N-pair loss [36]. Motivated by [1, 5], we perform patch-wise contrastive learning within the self-supervised classification framework to obtain high-quality CAM maps, leading to good-quality pseudo-GTs.

3. Proposed Method

The proposed 3SD method is a fully self-supervised approach that doesn't require any human annotations or noisy labels. As shown in Fig. 3, our method 3SD consists of a base network (BN) and a pseudo label generator. The base network outputs the saliency map along with the class labels which are used to generate the CAM map. BN utilizes self-supervised classification task to extract the semantic information for CAM map. The pseudo label generator fuses gated edge of the input image with CAM map to compute the pseudo-GT for training. In this section, we will discuss: (i) construction of the base network (BN), and (ii) pseudo-label generator.

3.1. Base Network

We construct our framework with two encoders (local encoder E_L and global encoder E_G) and two decoders (saliency decoder De_S and classification decoder De_C) as shown in Fig. 3. Both local features and global context are vital for SOD task. To learn the pixel-wise relationship with local features, the local encoder E_L is constructed using similar structure as U2Net [33]. Specifically, E_L has nested two-level U-structure network with ResUBlock to capture contextual information at different scales. Even though E_L learns the inter-dependency between neighboring pixels in the receptive field size (approximately 96×96), it fails to capture the global context for high-resolution images. To remedy this issue, we introduce a transformer based encoder E_G to model long-range relationship across patches (inspired by ViT [8]). By combining the outputs of both encoders, our 3SD is able to capture the local fine-grain details and reason globally. This combined encoded features are fed to saliency decoder De_S to obtain saliency map, and classification decoder De_C to obtain class label map as output. We use the similar architecture proposed in [33] for saliency decoder De_S. Our major contribution is on

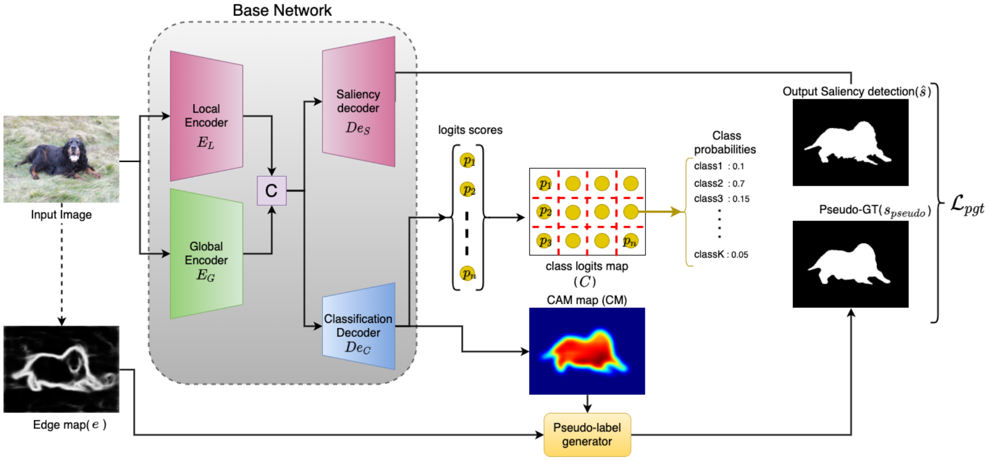

Figure 3. Overview of the 3SD method in computing the pseudo-GT for a given input image.

the classification decoder De_C. In contrast to conventional single-image-single-label classification design, our decoder performs self-supervised learning in patch wise. To enhance the feature representation and fully exploit the semantics, patches belonging to the object are encouraged to be differentiated from background patches using our novel contrastive learning paradigm. More details about E_L, E_G, De_S, and De_C are provided in the supplementary document.

As shown in Fig. 3, given an input image, BN predicts the salient object $\hat{s}$, and patch-wise class logits map C. It is trained in a fully self-supervised way using our proposed pseudo-label generator, which is described in the next subsection.

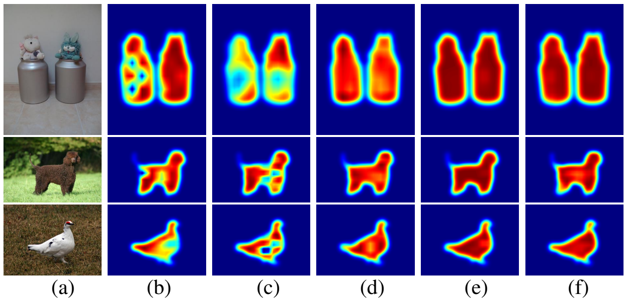

Figure 4. The CAM map comparison with different patch-wise auxiliary classification task. (a) input image, (b) CAM-map computed when auxiliary classification is one global class label, patch-wise classification with patch size (c) 16×16, (d) 24×24, (e) 32×32, (f) 48×48.

3.2. Pseudo-label generator

The main goal of 3SD is to train an SOD network without any GT-labels of salient objects. To achieve this, 3SD should be able to extract structural or semantic information of the object in the image. From the earlier works [21, 32] it is evident that attention maps from classification task provide important cues for salient object detection. In contrast to [21, 32] which require image class and/or caption labels, we train our BN with student-teacher based knowledge-distillation technique, and found the features from encoders of BN contain structural or semantic information of the salient object in the image. These semantics are the key to generate high-quality CAM maps [59]. But as shown in Fig. 4, the quality of CAM maps obtained from the self-supervised image-wise classification task when trained with single global class label, might not be high enough to produce pseudo-GT. This is due to the fact that single label classification task does not need to attend to all object regions. Instead, classification task drives the model to focus on the discriminative object parts. To address this issue, we propose a contrastive learning based patch-wise classification on the image patches as shown in Fig. 5. Patch-wise learning drives 3SD to capture local structures that constitute the object. Fig. 4 (b)-(e) show CAM map comparison between different classification tasks. Finally, by guiding the CAM map with the edge information of the input image, we obtain high-quality pseudo-GT (s_{pseudo}). This pseudo-GT is taken as the training labels to update the parameters of the 3SD network.

3.2.1 Self-supervised classification.

Student-teacher based knowledge distillation is a well-known learning paradigm that is commonly used for self-supervised classification tasks. As shown in Fig. 4, self-supervised classification task as done in [3] fails to produce high quality pseudo-GTs. This is because image-wise self-supervised classification with one global class only requires a few important activations in the salient object, leading to incomplete saliency map. To overcome this hurdle, our proposed self-supervised classification contains (i) global level self-supervision using $\mathcal{L}_{st}$ loss, and (ii) patch-wise contrastive learning based self-supervision using $\mathcal{L}_{\rho}$ loss.

In our 3SD, student and teacher networks share the same

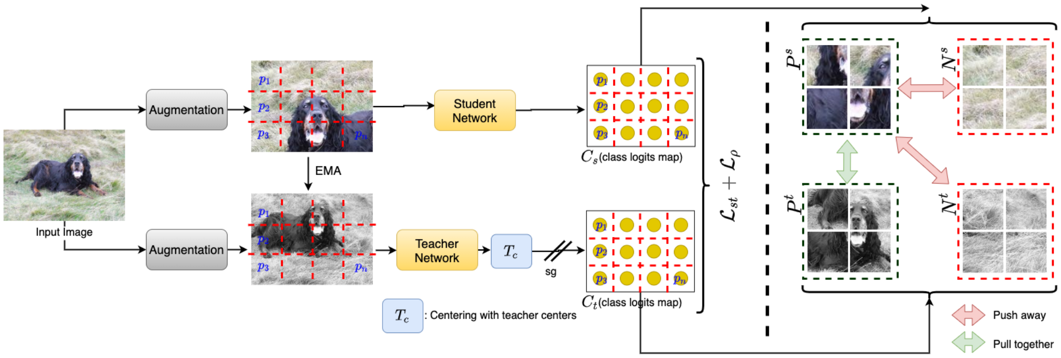

Figure 5. Self-distillation based training for classification. Here, "ema" means EMA update rule used to update the teacher network's parameters, and "sg" means stop gradient. T_c represents teacher centers to avoid trivial solution as explanied in [3].

architecture as the BN, and we denote student and teacher networks as f_{θ_s} and f_{θ_t}, respectively with θ_s and θ_t as their corresponding parameters. Given an input image, we compute class logits maps C_s and C_t for both networks, as well as their corresponding softmax probability output $P(C_s)$ and $P(C_t)$. Meanwhile, we obtain image wise class logits for student model: $c_s = \sum_{\{p_i\}} C_s(p_i)$, where p_i is i^{th} patch in the image. Similar expression holds for the teacher's logits c_t. Given a fixed teacher network f_{θ_t}, we match the image-level class probability distribution $P(c_s)$ and $P(c_t)$ by minimizing the cross-entropy loss to update the parameters of the student network f_{θ_s}:

$$\mathcal{L}_{st} = Q(P(c_s), P(c_t)), \tag{1}$$

where $Q(a, b) = -a \log b$. Note that $\mathcal{L}_{st}$ is used to update the student network's parameters θ_s with stochastic gradient descent. The teacher weights are updated using exponential moving average (EMA) update rule as in [3, 13], *i.e.*, $\theta_t \leftarrow \lambda\theta_t + (1 - \lambda)\theta_s$, with λ following a cosine schedule from 0.996 to 1 during training. Additionally, in order to improve quality of CAM maps we perform patch-wise contrastive learning for the classification task, where we identify the top M_ρ positive patches (patches similar to salient object) and M_ρ negative patches (patches dissimilar to salient object) from C_s and C_t. Using these positive and negative pairs we construct the following contrastive learning [5] based loss,

$$\mathcal{L}_\rho = \tfrac{1}{M_\rho^2} \textstyle\sum_{\{p_i^s\} \in P^s} \sum_{\{p_j^t\} \in P^t} - \log \frac{\exp(\text{sim}\left(C_s(p_i^s), C_t(p_j^t)\right)/\tau)}{\sum_{n_k \in N^t, N^s} \exp(\text{sim}\left(C_s(p_i^s), C_t(n_k)\right)/\tau)} \tag{2}$$

where P^s and N^s are the set of M_ρ positive and negative patches from C_s respectively, and P^t and N^t are the set of M_ρ positive and negative patches from C_t respectively. Concretely, P^s and N^s are extracted from C_s by comparing $C_s(p_i)$ with c_s using normalized dot product. Similarly, P^t and N^t are obtained from C_t. From Fig. 4, we can observe that addition of loss $\mathcal{L}_\rho$ drives the student network to extract class-sensitive local features across the whole object region, which results in better quality CAM maps.

In summary, the following steps are performed in self-supervised classification task to learn student network weights: (i) Augment the image with two different augmentations and randomly crop the images to obtain x''^l (local crop or smaller resolution) and x''^g (global crop or smaller resolution) as shown in Fig. 5. (ii) Fix the teacher network's parameters. (iii) Pass x''^l as input to the student network and x''^g as input to the teacher network, to obtain output class logits maps C'_s and C''_t, respectively. (iv) Match the outputs by minimizing the loss $\mathcal{L}_{st}$ and update the student network's parameters θ_s. (v) Additionally, we extract positive and negative patches from C_s and C_t, and update the student network's parameters θ_s using $\mathcal{L}_\rho$. (vi) Finally, employ the EMA update rule to update the teacher network's parameters θ_t. Fig. 5 shows the overview diagram of these steps in updating the student network's parameters.

3.2.2 Pseudo-GT

When 3SD is trained with self-supervised classification, features from classification decoder explicitly contain the structural or semantic information of the salient object in the image. Hence, we compute CAM (using technique from [59]) to obtain this semantic information of the salient object. We can clearly observe in Fig. 6 (c) and (d), that the obtained CAM map doesn't depict a proper or sharp boundary. To overcome this problem, we compute edge map using the Canny edge detection. Then we dilate the thresholded CAM map and gate the edge map to obtain the edges of the salient objects. This design effectively removes the background edges. Pseudo-GT (s_{pseudo}) is then defined as the union of the gated edge map and the CAM map, *i.e.*, $s_{pseudo} = CM \cup g_e$, where CM is the CAM map and g_e is the gated edge. Fig. 6 (g) and (h) show the pseudo-GT (s_{pseudo}) and the thresholded version of the pseudo-GT (s_{pseudo}) (with threshold 0.5). We compute cross-entropy between the pseudo-GT (s_{pseudo}) and the salient object $\hat{s}$ prediction output of 3SD,

$$\mathcal{L}_{pgt} = -\textstyle\sum_{(j,k)}^{(H,W)} \left[s_{pseudo}(j,k) \log \hat{s}(j,k) \; + (1 - s_{pseudo}(j,k)) \log \left(1 - \hat{s}(j,k)\right)\right)\right], \tag{3}$$

where (j, k) are pixel coordinates, and (H, W) are height and width of the image. We minimize $\mathcal{L}_p$ and update the

3SD parameters to perform SOD.

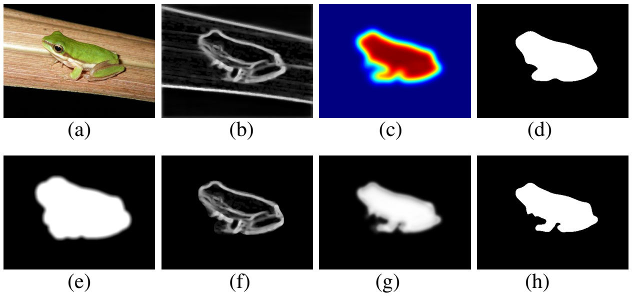

Figure 6. (a) input image, (b) edge map (c) CAM-maps computed by 3SD, (d) thresholded cam map, (e) dialated cam map, (f) gated edge map, (g) pseudo-GT, (h) thresholded pseudo-GT.

3.3. Loss

To improve the boundary estimation for SOD, we further introduce the gated structure-aware loss ($\mathcal{L}_{gs}$) proposed by [54]. Gated structure-aware ($\mathcal{L}_{gs}$) is defined as follows,

$$\mathcal{L}_{gs} = \sum_{(i,j)} \sum_{d \in (\overrightarrow{h}, \overrightarrow{v})} \Psi\left(|\partial_d s(i,j)| exp\left(-0.5|\partial_d (g \cdot x(i,j))|\right)\right), \tag{4}$$

where $\Psi(s) = \sqrt{s^2 + e^{-6}}$, (i,j) are pixel coordinates, $(\overrightarrow{h}, \overrightarrow{v})$ are horizontal and vertical directions, ∂ is derivative operation, and g is the gate for the structure loss (see Fig. 6(e)).

The overall loss function used for training the 3SD network is defined as follows,

$$\mathcal{L}_{total} = \mathcal{L}_{st} + \mathcal{L}_{\rho} + \mathcal{L}_{pgt} + \beta_1 \mathcal{L}_{gs}(\hat{s}, x). \tag{5}$$

We set $\beta_1 = 0.3$ in all our experiments.

4. Experiments and results

In this section, we present the details of various experiments conducted on six benchmarks to demonstrate the effectiveness of the proposed 3SD method. Additionally, we perform extensive ablation studies which shows the factors and parameters that influence the performance of the proposed 3SD method.

4.1. Datasets

Training dataset: We use images from the DUTS-TR dataset which is the training dataset of DUTS [39]. Note that we only use the images to train 3SD, *i.e.*, we don't use the corresponding salient object annotations. The DUTS-TR dataset contains 10,553 images, and it is the largest and most commonly used dataset for training SOD.

Evaluation datasets: We evaluate the performance of 3SD on five benchmark test datasets: (i) The DUTS-TE testing dataset (5,019 images) [39], (ii) DUT-OMRON (5,168 test images) [49], (iii) HKU-IS [22] (4,447 test images), (iv) PASCAL-S [23] (850 test images), and (v) ECSSD [48] (1,000 images).

4.2. Comparison methods and metrics

We compare the performance of 3SD with 11 SOTA weakly/unsupervised methods [21,42,47,50,51,53–55], and six SOTA fully-supervised methods [24, 25, 33, 34, 43, 44]. We use the following five evaluation metrics for comparison: structure measure [10] (S_m), mean F-score (F_β), mean E-measure [11] (E_η), and mean absolute error (MAE). Additionally, we plot precision vs. recall curves to show the effectiveness of our 3SD method.

4.3. Implementation Details

Given a training image, we randomly crop 320×320 pixels and use it for training 3SD. As explained in the earlier sections, we perform patch-wise contrastive learning based self-supervised classification without labels. Specifically, we learn the teacher centers T_c (K dimension vector) to sharpen the teacher's output before computing $\mathcal{L}_{st}$. Here, we sharpen the teachers output class logits map with T_c as shown in Fig 5 (*i.e.* $C_t(p_i) = C_t(p_i) + T_c$). In all of our experiments, we perform 32×32 patch-wise self-supervised classification in our 3SD method. For augmentations, we use the augmentations of BYOL [13] (Gaussian blur solarization, and color jittering) and multi-crop [2] with bi-cubic interpolation to resize the crop. In our 3SD method, the student and teacher networks share the same architecture as BN but initialized with different random weights. Initially, we train the student and teacher networks for self-supervised classification as explained in section 3.2 for 30 epochs. Later, we perform patch-wise self-supervised classification and pseudo-GT based salient object detection in an iterative manner as explained in the algorithm provided in the supplementary document. We set M_ρ as 10 for all our experiments. We train 3SD for 150 epochs using a stochastic gradient descent optimizer with batch-size 8 and learning rate 0.001. Following DUTS [39], we set $K = 200$ classes in all of our experiments. The student network (f_{θ_s}) in 3SD is used for all evaluations during inference.

4.4. Comparisons

Quantitative results: As shown in Table 1, our 3SD method consistently outperforms the SOTA weakly/unsupervised methods on all datasets and in all metrics. This shows training with our pseudo-GT generation is superior to the existing weakly/unsupervised SOD techniques. Moreover, although 3SD is not trained with any human annotations or weak labels (*e.g.* image captions, handcrafted features, or scribble annotations), its performance is on par with fully-supervised methods. In some cases the performance of 3SD is even better than some fully-supervised methods like [24, 28, 58]. Similar

Table 1. Comparison with SOTA methods on five benchmark datasets (DUTS-TE, DUT-OMRON, HKU-IS, PASCAL, and ECSSD) using the metrics S_m, B_μ, F_β, E_η, and MAE where ↑ & ↓ denote larger and smaller is better, respectively.

Dataset	DUTS-TE				DUT-OMRON				HKU-IS				PASCAL				ECSSD			
Metric	$S_m \uparrow$	$F_\beta \uparrow$	$E_\eta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$F_\beta \uparrow$	$E_\eta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$F_\beta \uparrow$	$E_\eta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$F_\beta \uparrow$	$E_\eta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$F_\beta \uparrow$	$E_\eta \uparrow$	$MAE \downarrow$
Fully Supervised SOD methods																				
PiCANet [24](CVPR'18)	0.851	0.757	0.853	0.062	0.826	0.710	0.823	0.072	0.906	0.854	0.909	0.047	0.848	0.799	0.804	0.129	0.867	0.871	0.909	0.054
MSNet [44](CVPR'19)	0.851	0.792	0.883	0.050	0.809	0.709	0.830	0.064	0.907	0.878	0.930	0.039	0.844	0.671	0.813	0.822	0.905	0.885	0.922	0.048
BASNet [34](CVPR'19)	0.866	0.823	0.896	0.048	0.836	0.767	0.865	0.057	0.909	0.903	0.943	0.032	0.838	0.821	0.821	0.122	0.910	0.913	0.938	0.040
U2Net [33](PR'20)	0.861	0.804	0.897	0.044	0.842	0.757	0.867	0.054	0.916	0.890	0.945	0.031	0.844	0.797	0.831	0.074	0.918	0.910	0.936	0.033
LDF [43](CVPR'20)	0.881	0.855	0.910	0.034	0.847	0.773	0.873	0.051	0.919	0.914	0.954	0.027	0.851	0.848	0.865	0.060	0.912	0.930	0.925	0.034
VST [25](ICCV'21)	0.885	0.870	0.939	0.037	0.839	0.800	0.883	0.058	0.919	0.922	0.962	0.030	0.863	0.829	0.865	0.067	0.917	0.929	0.945	0.034
BN(ours)	0.883	0.829	0.913	0.036	0.843	0.777	0.869	0.049	0.918	0.918	0.959	0.024	0.856	0.832	0.849	0.068	0.924	0.933	0.940	0.032
Weakly supervised SOD methods																				
WSS [39](CVPR'17)	0.748	0.633	0.806	0.100	0.730	0.590	0.729	0.110	0.822	0.773	0.819	0.079	-	0.698	0.690	0.184	0.808	0.767	0.796	0.108
WSI [21](AAAI'18)	0.697	0.569	0.690	0.116	0.759	0.641	0.761	0.100	0.808	0.763	0.800	0.089	-	0.653	0.647	0.206	0.805	0.762	0.792	0.068
MSW [50](CVPR'19)	0.759	0.648	0.742	0.091	0.756	0.597	0.728	0.109	0.818	0.734	0.786	0.084	0.697	0.685	0.693	0.178	0.825	0.761	0.787	0.098
Scribble_S [54](CVPR'20)	0.793	0.746	0.865	0.062	0.771	0.702	0.835	0.068	0.855	0.857	0.923	0.047	0.742	0.788	0.798	0.140	0.854	0.865	0.908	0.061
MFNet [32](ICCV'21)	0.775	0.770	0.062	0.076	0.742	0.646	0.803	0.087	0.846	0.851	0.921	0.059	0.770	0.751	0.817	0.115	0.834	0.854	0.885	0.084
Unsupervised SOD methods																				
SBF [51](ICCV'17)	0.739	0.622	0.763	0.107	0.731	0.612	0.763	0.108	0.812	0.783	0.855	0.075	0.712	0.735	0.746	0.167	0.813	0.782	0.835	0.096
MNL [55](CVPR'18)	0.813	0.725	0.853	0.075	0.733	0.597	0.712	0.103	0.860	0.820	0.858	0.065	0.728	0.748	0.741	0.158	0.845	0.810	0.836	0.090
EDNS [53](ECCV'20)	0.828	0.747	0.859	0.060	0.791	0.701	0.816	0.070	0.890	0.878	0.919	0.043	0.750	0.759	0.794	0.142	0.860	0.852	0.883	0.071
DeepUSPS [30](Neurips'19)	0.787	0.734	0.848	0.068	0.795	0.713	0.848	0.063	0.876	0.864	0.930	0.041	0.757	0.768	0.792	0.151	0.861	0.870	0.903	0.063
Yan *et al.* [47] (AAAI'22)	0.840	0.758	0.859	0.052	0.801	0.711	0.841	0.066	0.890	0.873	0.931	0.047	0.759	0.770	0.792	0.158	0.862	0.876	0.888	0.068
UMNet [42](CVPR'22)	0.802	0.749	0.863	0.067	0.804	0.727	0.859	**0.063**	0.886	0.872	0.939	0.041	0.762	0.775	0.800	0.144	0.867	0.872	0.902	0.064
3SD(ours)	**0.846**	**0.772**	**0.877**	**0.043**	**0.806**	**0.738**	**0.863**	0.064	**0.908**	**0.898**	**0.947**	**0.039**	**0.774**	**0.786**	**0.817**	**0.137**	**0.888**	**0.894**	**0.928**	**0.049**

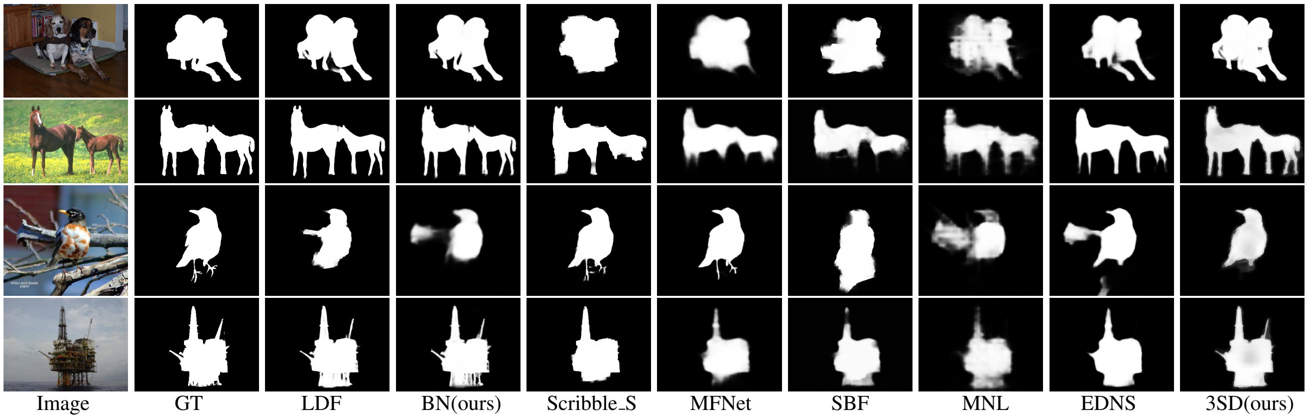

Figure 7. Qualitative comparison of 3SD method against SOTA fully-supervised, and SOTA weakly/unsupervised methods.

behaviors can be observed in the precision vs. recall curves shown in Fig. 8. Curves corresponding to our method are close to the SOTA fully-supervised methods, while the curves corresponding to the SOTA weakly/unsupervised methods are far below our 3SD method.

Qualitative results: Fig. 7 illustrates the qualitative comparisons of our 3SD method with SOTA methods on 4 sample images from DUTS-TE, DUT-OMRON, HKU-IS, and ECSSD. It can be seen that the outputs of [50, 51, 53, 54] are blurred or incomplete, and include parts of non-salient objects. In contrast, 3SD outputs are accurate, clear, and sharp. For example, output saliency maps of [50,51,53,54] in the second row of Fig. 7 miss parts of legs for the horses. And, output saliency maps of [50,51,53,54] in the third row of Fig. 7 contains artifacts or parts of non-salient objects. In contrast, saliency maps of our 3SD method delineate legs for the horses properly, and are free of artifacts. More qualitative comparisons are provided in the supplementary material.

4.5. Ablation study

The goal of these ablation experiments is to analyze the components in the pseudo-GT generation that effect the performance of 3SD. We perform seven experiments, patch-wise contrastive learning based self-supervised classification experiments (PCL16: 16×16, PCL24: 24×24, PCL32: 32×32, PCL48: 48×48 pixels), self-supervised classification with one global class label (GCL), PCM: generating pseudo-GT without using the edge map e (using CM as s_{pseudo}), PGE: generating pseudo-GT without using the CAM map CM (using g_e as s_{pseudo}), CMG: training 3SD with the pseudo-GT ($s_{pseudo} = CM \cup g_e$) and without $\mathcal{L}_{gs}$.

Impact of patch-wise contrastive learning based self-supervised classification: As can be seen from Table 2, PG (global class wise classification) fails to produce proper CAM maps (see Fig. 4) which results in lower performance when compared to patch-wise contrastive learning based self-supervised classification with patch-size PCL16, PCL24, PCL32, PCL40 (16×16, 24×24, 32×32, 48×48 re-

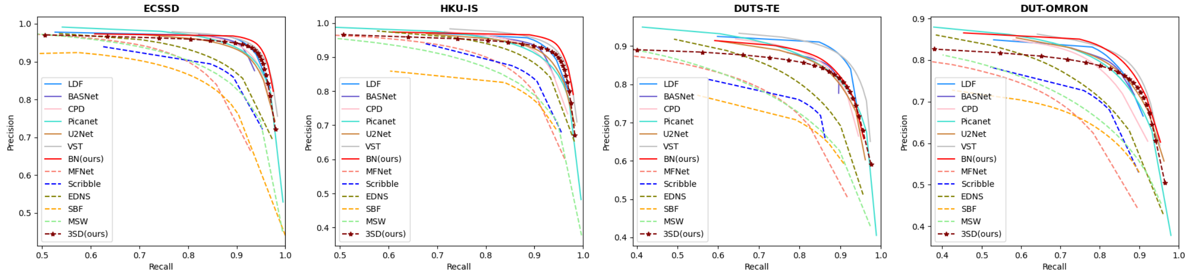

Figure 8. Precision vs. recall curve comparison of our method with SOTA methods on the ECSSD, HKU-IS, PASCAL, DUTS-TE, and DUT-OMRON datasets. In the precision vs. recall graphs, we represent all supervised methods with thick lines and weakly/unsupervised methods with dotted lines.

Table 2. Ablation study experiment on the DUTS-TE, HKU-IS, and ECSSD datasets. ↑ & ↓ denote larger and smaller is better, respectively.

Dataset	DUTS-TE				DUT-OMRON				ECSSD				HKU-IS			
Metrics	$S_m \uparrow$	$F_\beta \uparrow$	$E_\eta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$F_\beta \uparrow$	$E_\eta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$F_\beta \uparrow$	$E_\eta \uparrow$	$MAE \downarrow$	$S_m \uparrow$	$F_\beta \uparrow$	$E_\eta \uparrow$	$MAE \downarrow$
Self or Unsupervised																
PCL16	0.832	0.759	0.866	0.047	0.792	0.726	0.832	0.062	0.878	0.881	0.921	0.050	0.862	0.864	0.920	0.042
PCL24	0.841	0.762	0.874	0.044	0.796	0.738	0.838	0.062	0.881	0.891	0.926	0.049	0.904	0.885	0.935	0.039
PCL32	0.846	0.772	0.877	0.043	0.806	0.738	0.863	0.063	0.888	0.894	0.928	0.049	0.908	0.898	0.947	0.039
PCL48	0.842	0.767	0.869	0.045	0.795	0.732	0.849	0.062	0.879	0.894	0.930	0.048	0.896	0.883	0.941	0.040
GCL	0.804	0.748	0.844	0.058	0.782	0.715	0.826	0.070	0.869	0.873	0.904	0.052	0.863	0.859	0.916	0.048
PCM	0.801	0.741	0.837	0.072	0.770	0.697	0.814	0.077	0.864	0.848	0.869	0.077	0.844	0.852	0.903	0.055
PGE	0.720	0.654	0.737	0.120	0.677	0.624	0.738	0.128	0.780	0.759	0.860	0.127	0.752	0.739	0.826	0.123
CMG	0.828	0.756	0.858	0.052	0.783	0.718	0.817	0.063	0.876	0.892	0.922	0.049	0.892	0.878	0.931	0.040
Fully supervised Base Network(BN)																
B0	0.836	0.798	0.884	0.052	0.828	0.747	0.854	0.060	0.895	0.891	0.929	0.047	0.890	0.893	0.941	0.035
B1	0.861	0.807	0.897	0.046	0.832	0.756	0.859	0.054	0.905	0.907	0.934	0.043	0.902	0.906	0.946	0.031
B2	0.883	0.829	0.913	0.036	0.843	0.777	0.869	0.049	0.924	0.933	0.940	0.032	0.918	0.918	0.959	0.024

spectively). From Table 2 and Fig. 4, it is evident that when we increase the patch size in patch-wise self-supervised classification from 16 × 16 to 32 × 32, we obtain better quality CAM maps which in turn results in better pseudo-GT and as a result better SOD performance. We obtained the best performance using setting of 32 × 32. Note that, larger patch doesn't improve the performance of 3SD method.

Impact of Pseudo-GT: As explained in the section 3.2, pseudo-GT is defined as $s_{pseudo} = CM \cup g_e$. Here we perform experiments to validate the important role played by CAM (CM) map and gated edge (g_e) in the construction of the pseudo-GT. From Table 2 columns PCM and PGE, we can clearly observe a huge improvement in performance when we use CM as the pseudo-GT instead of g_e. This shows that CM obtained from patch-wise self-supervised classification contains more consistent semantic information than gated edge (g_e). Furthermore in the CMG column of Table 2, the combination of both CM and g_e brings in further increase of the performance. The PCL32 column in Table 2 corresponds to the case where we train 3SD with the pseudo-GT ($s_{pseudo} = CM \cup g_e$), with additional boundary aware loss ($\mathcal{L}_{gs}$). In this case, we found a small improvement in the performance.

BaseNetwork (BN): We perform three experiments to evaluate the effectiveness of the constructed Base Network (BN). We use the following definitions in this experiment: 1) B0: BN with one encoder (local encoder E_L) and one decoder(saliency decoder De_S). 2) B1: BN with two encoders(E_L and E_G), and one decoder(De_S) (adding global encoder (E_G) to BN). 3) B2: Using two encoders and two decoders (adding classification decoder to BN), as shown in Fig. 3. For this experiment, B0, B1, B2 methods are trained in supervised fashion using the actual ground-truth labels with cross-entropy loss. As can be seen from Table 2, we obtain an improvement when we add transformer based global encoder (E_G) to BN (B1 in Table 2). This implies that E_G is efficient in capturing patch-wise relations to obtain better saliency maps. Furthermore, adding a classification decoder improves BN's performance (B2 in Table 2).

5. Conclusion

We presented a Self-Supervised Saliency Detection method (3SD), which doesn't require any labels, *i.e.*, neither human annotations nor weak labels like image captions, handcrafted features or scribble annotations. The cornerstone of successful self-supervised SOD approach is generation of high-quality of pseudo-GTs. Our novel patch-wise contrastive learning paradigm effectively captures the semantics of salient objects. And this is the key of our superior performance verified on five benchmarks.

References

[1] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019. 3

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 6

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021. 2, 3, 4, 5

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2017. 3

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 2, 3, 5

[6] Subhabrata Choudhury, Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Guess what moves: Unsupervised video and image segmentation by anticipating motion. *arXiv preprint arXiv:2205.07844*, 2022. 2

[7] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, pages 1635–1643, 2015. 3

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[9] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 38(9):1734–1747, 2015. 3

[10] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, pages 4548–4557, 2017. 6

[11] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. *arXiv preprint arXiv:1805.10421*, 2018. 6

[12] Mengyang Feng, Huchuan Lu, and Errui Ding. Attentive feedback network for boundary-aware salient object detection. In *CVPR*, pages 1623–1632, 2019. 3

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *NIPS*, 2020. 3, 5, 6

[14] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304, 2010. 3

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 3

[16] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *ICML*, pages 4182–4192. PMLR, 2020. 3

[17] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 3

[18] Kuang-Jui Hsu12, Yen-Yu Lin, and Yung-Yu Chuang. Weakly supervised saliency detection with a category-driven map generator. *BMVC*, 2017. 3

[19] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, pages 876–885, 2017. 3

[20] Jiwhan Kim, Dongyoon Han, Yu-Wing Tai, and Junmo Kim. Salient region detection via high-dimensional color transform. In *CVPR*, pages 883–890, 2014. 1, 3

[21] Guanbin Li, Yuan Xie, and Liang Lin. Weakly supervised salient object detection using image labels. In *AAAI*, pages 7024–7031, 2018. 2, 3, 4, 6, 7

[22] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *CVPR*, pages 5455–5463, 2015. 6

[23] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014. 6

[24] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *CVPR*, pages 3089–3098, 2018. 3, 6, 7

[25] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *CVPR*, pages 4722–4732, 2021. 1, 2, 6, 7

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1

[27] Shijian Lu, Cheston Tan, and Joo-Hwee Lim. Robust and efficient saliency modeling from image co-occurrence histograms. *TPAMI*, 36(1):195–201, 2013. 1, 3

[28] Zhiming Luo, Akshaya Mishra, Andrew Achkar, Justin Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. In *CVPR*, pages 6609–6617, 2017. 3, 6

[29] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8364–8375, 2022. 2

[30] Duc Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox. Deepusps: Deep robust unsupervised saliency prediction with self-supervision. *arXiv preprint arXiv:1909.13055*, 2019. 2, 3, 7

[31] Anton Obukhov, Stamatios Georgoulis, Dengxin Dai, and Luc Van Gool. Gated crf loss for weakly supervised semantic image segmentation. *arXiv preprint arXiv:1906.04651*, 2019. 3

[32] Yongri Piao, Jian Wang, Miao Zhang, and Huchuan Lu. Mfnet: Multi-filter directive network for weakly supervised salient object detection. In *ICCV*, pages 4136–4145, October 2021. 1, 2, 4, 7

[33] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, 2020. 1, 2, 3, 6, 7

[34] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *CVPR*, pages 7479–7489, June 2019. 2, 3, 6, 7

[35] Gyungin Shin, Samuel Albanie, and Weidi Xie. Unsupervised salient object detection with spectral cluster voting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3971–3980, 2022. 2

[36] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *NIPS*, 29, 2016. 3

[37] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017. 3

[38] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, pages 776–794. Springer, 2020. 3

[39] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, pages 136–145, 2017. 3, 6, 7

[40] Tiantian Wang, Lihe Zhang, Shuo Wang, Huchuan Lu, Gang Yang, Xiang Ruan, and Ali Borji. Detect globally, refine locally: A novel approach to saliency detection. In *CVPR*, pages 3127–3135, 2018. 3

[41] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14543–14553, 2022. 2

[42] Yifan Wang, Wenbo Zhang, Lijun Wang, Ting Liu, and Huchuan Lu. Multi-source uncertainty mining for deep unsupervised saliency detection. In *CVPR*, pages 11727–11736, 2022. 2, 6, 7

[43] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *CVPR*, pages 13022–13031, June 2020. 1, 2, 3, 6, 7

[44] Runmin Wu, Mengyang Feng, Wenlong Guan, Dong Wang, Huchuan Lu, and Errui Ding. A mutual learning method for salient object detection with intertwined multi-supervision. In *CVPR*, pages 8150–8159, 2019. 3, 6, 7

[45] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. In *CVPR*, pages 3907–3916, 2019. 2, 3

[46] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 3

[47] Pengxiang Yan, Ziyi Wu, Mengmeng Liu, Kun Zeng, Liang Lin, and Guanbin Li. Unsupervised domain adaptive salient object detection through uncertainty-aware pseudo-label learning. *arXiv preprint arXiv:2202.13170*, 2022. 6, 7

[48] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013. 6

[49] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013. 6

[50] Yu Zeng, Yunzhi Zhuge, Huchuan Lu, Lihe Zhang, Mingyang Qian, and Yizhou Yu. Multi-source weak supervision for saliency detection. In *CVPR*, pages 6074–6083, 2019. 1, 2, 3, 6, 7

[51] Dingwen Zhang, Junwei Han, and Yu Zhang. Supervision by fusion: Towards unsupervised learning of deep salient object detector. In *ICCV*, pages 4048–4056, 2017. 1, 2, 3, 6, 7

[52] Jinxia Zhang, Krista A Ehinger, Haikun Wei, Kanjian Zhang, and Jingyu Yang. A novel graph-based optimization framework for salient object detection. *Pattern Recognition*, 64:39–50, 2017. 3

[53] Jing Zhang, Jianwen Xie, and Nick Barnes. Learning noise-aware encoder-decoder from noisy labels by alternating back-propagation for saliency detection. In *ECCV*, pages 349–366. Springer, 2020. 1, 2, 3, 6, 7

[54] Jing Zhang, Xin Yu, Aixuan Li, Peipei Song, Bowen Liu, and Yuchao Dai. Weakly-supervised salient object detection via scribble annotations. In *CVPR*, pages 12546–12555, 2020. 1, 2, 3, 6, 7

[55] Jing Zhang, Tong Zhang, Yuchao Dai, Mehrtash Harandi, and Richard Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *CVPR*, pages 9029–9038, 2018. 1, 2, 3, 6, 7

[56] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency detection. In *ICCV*, pages 212–221, 2017. 3

[57] Qiang Zhang, Zhen Huo, Yi Liu, Yunhui Pan, Caifeng Shan, and Jungong Han. Salient object detection employing a local tree-structured low-rank representation and foreground consistency. *Pattern Recognition*, 92:119–134, 2019. 3

[58] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *CVPR*, pages 714–722, 2018. 3, 6

[59] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 4, 5

[60] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *CVPR*, pages 2814–2821, 2014. 1, 3