applied sciences

MDPI

Article

Self-Improved Learning for Salient Object Detection

Songyuan Li, Hao Zeng, Huanyu Wang and Xi Li *

College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China; leizungjyun@zju.edu.cn (S.L.); zenghao_97@zju.edu.cn (H.Z.); huanyuhello@zju.edu.cn (H.W.)
* Correspondence: xilizju@zju.edu.cn

Abstract: Salient Object Detection (SOD) aims at identifying the most visually distinctive objects in a scene. However, learning a mapping directly from a raw image to its corresponding saliency map is still challenging. First, the binary annotations of SOD impede the model from learning the mapping smoothly. Second, the annotator's preference introduces noisy labeling in the SOD datasets. Motivated by these, we propose a novel learning framework which consists of the Self-Improvement Training (SIT) strategy and the Augmentation-based Consistent Learning (ACL) scheme. SIT aims at reducing the learning difficulty, which provides smooth labels and improves the SOD model in a momentum-updating manner. Meanwhile, ACL focuses on improving the robustness of models by regularizing the consistency between raw images and their corresponding augmented images. Extensive experiments on five challenging benchmark datasets demonstrate that the proposed framework can play a plug-and-play role in various existing state-of-the-art SOD methods and improve their performances on multiple benchmarks without any architecture modification.

Keywords: salient object detection; self-improved strategy

check for updates

Citation: Li, S.; Zeng, H.; Wang, H.; Li, X. Self-Improved Learning for Salient Object Detection. *Appl. Sci.* **2023**, *13*, 12966. https://doi.org/10.3390/app132312966

Academic Editors: Gary KL Tam, Frederick W. B. Li, Xianghua Xie, Avishek Siris and Jianbo Jiao

Received: 22 October 2023
Revised: 28 November 2023
Accepted: 30 November 2023
Published: 4 December 2023

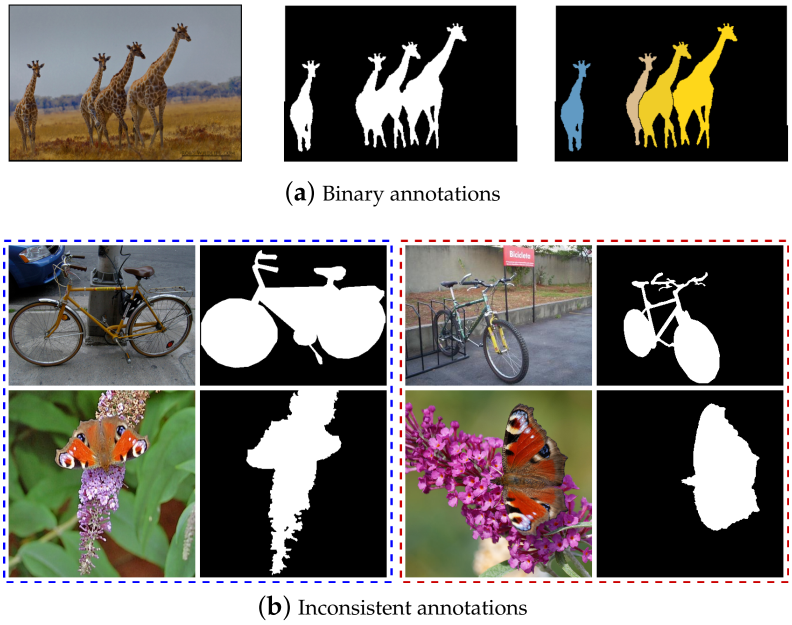
1. Introduction

Salient Object Detection (SOD) refers to the problem of identifying the most visually distinctive objects. SOD methods typically learn a mapping from an image to a saliency mask, which benefits various downstream tasks such as object tracking [1], image captioning [2], video analysis [3], person re-identification [4], image retrieval [5], semantic segmentation [6], and object detection [7].

Despite the efforts long devoted to it, SOD remains a challenging task. One difficulty lies in the fact that the annotations are **binary**, i.e., each pixel is labeled either salient or non-salient, which does not fully capture the nuanced nature of saliency. For example, the four giraffes in Figure 1a are labeled as equally salient regardless of their different sizes, locations, and depths in the image. Furthermore, the binary annotations pose challenges in learning the mapping from images to saliency masks. The abrupt transition between salient and non-salient regions can make the learning process less smooth and hinder the ability of SOD models to generalize well to unseen data.

In this regard, we propose a novel **S**elf-**I**mprovement **T**raining (SIT) strategy to reduce the learning difficulty incurred by the binary annotations. We design a progressively updated module to generate smooth labels, providing flexible degrees of saliency for each object to represent the difference of saliency between objects. The smooth labels provide more nuanced information about the saliency levels within an image, allowing the model to better understand the varying degrees of importance assigned to different objects. The use of smooth labels together with the binary annotations smoothens the learning process. However, it is worth noting that introducing smooth labels can also introduce noise into the training process. To this end, we mitigate the noise brought by the smooth labels by adapting the weights of smooth labels in the optimization phase.

Another difficulty is that the annotations are **subjective**, i.e., the saliency of an object is influenced by the annotator's personal preference. As shown in Figure 1b, the hollow part of a bicycle is labeled salient in the left but *not* salient in the right. Moreover, both images with butterflies in Figure 1b share similar layouts (butterfly, flowers, and green background) but the flowers are marked as salient in the left image and not salient in the right. Therefore, SOD requires the model to be of high robustness to noisy labeling.



(**a**) Binary annotations

(**b**) Inconsistent annotations

Figure 1. (**a**) The four giraffes are labeled as equally salient regardless of their different sizes and location in the image. (**b**) The bike on the left is from PASCAL-S [8] and the right is from DUT-OMRON [9]. The butterfly on the left is from ECSSD [10] and the right is from HKU-IS [11].

To improve the robustness, we propose a novel **A**ugmentation-based **C**onsistent **L**earning (ACL) scheme. By simulating a cross-dataset validation scenario, ACL introduces salient-related data augmentation to images, aiming to enhance the consistency of predictions between raw and augmented images. In this way, ACL makes the prediction of a raw image be consistent to that of the augmented image. As a result, the SOD model is capable of accurately locating the most salient objects while disregarding irrelevant variations caused by augmentation, thus improving the robustness of the model.

Our main contributions are summarized as follows.

- We propose a novel Self-Improved Training strategy for SOD, which reduces the learning difficulty and effectively improves the performance.
- We present an Augmentation-based Consistent Learning scheme that regularizes the consistency between raw images and their corresponding augmented images at training time and improves the robustness of the models.
- The proposed method is model independent, which can be applied to existing prevalent methods without modifying the architecture to gain considerable improvements.

2. Related Work

2.1. Network Designs for SOD

The network designs for Salient Object Detection can be summarized into three categories: architecture-based, aggregation-based, and contour-based methods.

First, architecture-based methods explore various feature extractors for SOD. Early methods such as [11–17] obtain handcrafted or deep features from image subunits and use MLP classifiers to predict saliency scores based on the features. For example, Super-CNN [16] and Multi-Context [13] use super-pixels or patches as image subunits. These methods fail to model the global context and cannot be trained end to end. Inspired by FCNs [18], CNN-based methods [19–25] formulate the salient object detection as a pixel-level prediction task and generate the saliency map in an end-to-end manner. Recently, Transformer has also been applied to SOD [26–30]. They combine both Transformer and CNN to process multi-level features. For instance, VST [30] considers the SOD task from the sequence-to-sequence perspective and designs a pure transformer model to handle the tokenized input. In addition, Transformer is also utilized to model the cross-modal information from the RGB and depth data [26,27].

Second, aggregation-based methods focus on feature aggregation or feature fusion to combine features from different layers. These methods capture both spatial details and global semantics for an accurate saliency map [13,24,31–36]. A simple yet effective way is to skip connect the features from the encoder to the features with the same spatial size from the decoder. This U-shape-based structure has been employed in many works [13,35,36]. Furthermore, GateNet [34] introduces gated mechanisms to control the information fusion flow. Also, channel-wise attention and spatial-wise attention have been applied to weigh the importance during the feature aggregation process [31–33]. MENet [37] designs a multi-scale feature enhancement module to gradually aggregate and refine global or detailed features by changing the size order of the input feature sequence.

To achieve high-quality saliency prediction with fine edge details, contour-based methods attempt to explicitly regularize the contours of salient objects. This line of work involves two aspects: network design and loss function design. Contour-aware networks attempt to model contours explicitly [32,33,38,39]. Specifically, EGNet [38] proposes to make the model predict saliency and contours simultaneously. Based on this, C2SNet [39] adopts an alternating converting method between contour maps and saliency maps. Unlike existing two-branch methods [33,38] that use one branch for the boundary and the other one for the entire object, MENet [37] uses two parallel branches that learn internal regional features and global features, respectively, so as to reduce the interference of inaccurate boundary information with global features. Some other works design contour-aware loss functions. The structural similarity index (SSIM) [40] and weighted BCE loss [41] are introduced for more edge details in the saliency maps.

Our method aims to smoothen the learning process and improve the robustness of SOD. We do not design a specific architecture or model contours for SOD explicitly. Instead, our learning strategy is capable of working with all these methods without modifying their networks.

2.2. Non-Fully Supervised Learning for SOD

To alleviate the burden of human efforts on pixel-level annotations, semi-/weakly-supervised methods for SOD have been proposed. These methods require image-level [42–46], scribble-level [47,48], or point-level annotations [49,50]. For example, FIN [42] leverages the category labels and is jointly optimized with an FCN to capture potentially salient regions. WS3A [47] uses scribbles annotation to do local supervision while utilizing auxiliary edges to provide details. PSOD [49] uses point annotation to provide the locations of salient objects to obtain pseudo labels to gain the first round of supervision, and then again uses point annotation to suppress non-salient objects to obtain optimized pseudo labels for the second round.

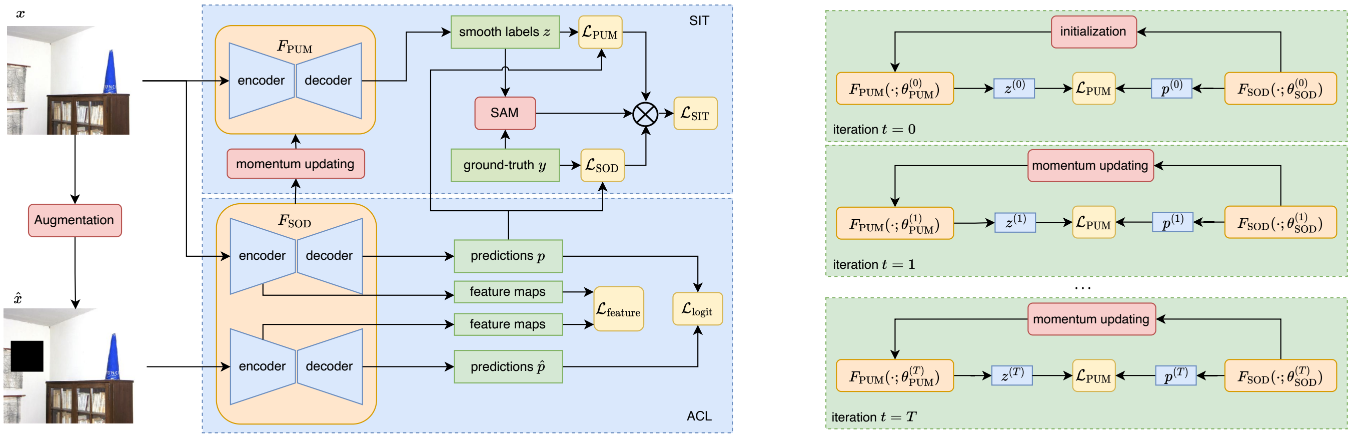
Recent years also witness the adoption of self-supervised learning for SOD, which requires no human annotations. A2S [51] and A2Sv2 [52] employ large-scale unsupervised pre-trained networks. They utilize a two-step approach where the first stage involves constructing reliable pseudo labels and the second stage trains the final segmenter after intermediate label processing. UNSS [53] introduces a top–down context guidance strategy that extracts detailed signals for both global and local segmentation learning.

In this paper, the training of our method consists of two parts: the Self-Improvement Training strategy (SIT) and Augmentation-based Consistent Learning. They are motivated by self-supervised learning, but both of them utilize pixel-level annotations. Specifically, SIT utilizes the SOD model, which uses pixel-level annotations, to generate smooth labels to alleviate the training difficulty of itself. ACL adopts the CutOut augmentation to remove part of the image and forces the model to make the results of the augmented image consistent with the results of the original image. The removed part is determined using the ground truth, and thus, ACL cannot be treated as unsupervised. In essence, the aim of ACL is to make the model robust instead of reducing human efforts on annotations.

3. Method

Salient object detection learns a function from an image to saliency predictions. Generally, an SOD network $\mathcal{F}_{\text{SOD}}$ consists of an encoder E and a decoder D. Given an input RGB image $x \in \mathbb{R}^{H\times W\times 3}$ and its ground truth $y \in \{0,1\}^{H\times W}$ with the height H and width W, encoder E extracts features and decoder D generates a saliency prediction $p \in \mathbb{R}^{H\times W}$ by $p = \mathcal{F}_{\text{SOD}}(x) = D(E(x))$.

Next, we present our framework as illustrated in Figure 2a. We design the proposed Self-Improvement Training strategy (SIT) in Section 3.1 and Augmentation-based Consistent Learning scheme (ACL) in Section 3.2. Finally, we summarize the whole training procedure in Section 3.3.


x
Augmentation
$\hat{x}$
SIT
F_{PUM}
encoder
decoder
momentum updating
smooth labels z
$\mathcal{L}_{\text{PUM}}$
SAM
$\mathcal{L}_{\text{SIT}}$
ground-truth y
$\mathcal{L}_{\text{SOD}}$
F_{SOD}
encoder
decoder
encoder
decoder
predictions p
feature maps
feature maps
predictions $\hat{p}$
$\mathcal{L}_{\text{feature}}$
$\mathcal{L}_{\text{logit}}$
ACL
initialization
$F_{\text{PUM}}(\cdot;\theta^{(0)}_{\text{PUM}})$
$z^{(0)}$
$\mathcal{L}_{\text{PUM}}$
$p^{(0)}$
$F_{\text{SOD}}(\cdot;\theta^{(0)}_{\text{SOD}})$
iteration $t=0$
momentum updating
$F_{\text{PUM}}(\cdot;\theta^{(1)}_{\text{PUM}})$
$z^{(1)}$
$\mathcal{L}_{\text{PUM}}$
$p^{(1)}$
$F_{\text{SOD}}(\cdot;\theta^{(1)}_{\text{SOD}})$
iteration $t=1$
...
momentum updating
$F_{\text{PUM}}(\cdot;\theta^{(T)}_{\text{PUM}})$
$z^{(T)}$
$\mathcal{L}_{\text{PUM}}$
$p^{(T)}$
$F_{\text{SOD}}(\cdot;\theta^{(T)}_{\text{SOD}})$
iteration $t=T$

(**a**) Our Framework (**b**) Progressively Updated Module (PUM)

Figure 2. The overview of our proposed framework. Our framework consists of two components, the Self-Improvement Training strategy (SIT) and Augmentation-based Consistent Learning scheme (ACL). The PUM Module of SIT generates smooth labels at training time, progressively improving the SOD model in a momentum-updating manner. SIT relaxes the binary annotations with smooth labels, reducing the learning difficulty. ACL enforces the consistency between raw images and augmented images, regularizing the model at both feature level and prediction level.

3.1. Self-Improvement Training Strategy

In this work, we propose a Self-Improvement Training (SIT) strategy to reduce the learning difficulty by relaxing the binary annotations with smooth labels. We decompose SIT into two steps. First, we generate smooth labels with a proposed Progressively Updated Module (PUM). Second, we design a Sample Adaptive Module (SAM) to weigh the balance between binary and smooth labels.

3.1.1. PUM: Progressively Updated Module

Formally, we design a progressively updated module (PUM), denoted by $\mathcal{F}_{\text{PUM}}$, to generate smooth labels. $\mathcal{F}_{\text{PUM}}$ shares the same network structure with the SOD network $\mathcal{F}_{\text{SOD}}$. Specifically, let the parameters of $\mathcal{F}_{\text{SOD}}$ be θ_{SOD}, and the parameters of $\mathcal{F}_{\text{PUM}}$ be θ_{PUM}. First, we randomly initialize θ_{SOD}. Then, we initialize θ_{PUM} by

$$t = 0: \qquad \theta^{(0)}_{\text{PUM}} = \theta^{(0)}_{\text{SOD}}, \tag{1}$$

where t is the iteration of the training (Iteration t refers to a mini-batch training iteration instead of an epoch). At this point, F_{PUM} is identical to F_{SOD}. They can only generate a random salient output before training. To progressively improve the quality of the smooth labels, we update θ_{PUM} at a momentum coefficient η from θ_{SOD} as follows

$$\begin{aligned} t = 1: &\qquad \theta^{(1)}_{\text{PUM}} = \eta \cdot \theta^{(0)}_{\text{PUM}} + (1-\eta) \cdot \theta^{(1)}_{\text{SOD}}, \\ t = 2: &\qquad \theta^{(2)}_{\text{PUM}} = \eta \cdot \theta^{(1)}_{\text{PUM}} + (1-\eta) \cdot \theta^{(2)}_{\text{SOD}}, \\ \vdots &\qquad\qquad \vdots \\ t = T: &\qquad \theta^{(T)}_{\text{PUM}} = \eta \cdot \theta^{(T-1)}_{\text{PUM}} + (1-\eta) \cdot \theta^{(T)}_{\text{SOD}}. \end{aligned} \tag{2}$$

It is worth noting that only parameters θ_{SOD} are updated via backpropagation, whereas parameters θ_{PUM} are updated progressively by θ_{SOD} at the training time of $\mathcal{F}_{\text{SOD}}$. The momentum update in Equation (2) makes θ_{PUM} evolve more smoothly than θ_{SOD}. As a result, the generated smooth labels are improved smoothly with the training of $\mathcal{F}_{\text{SOD}}$. In this way, the PUM progressively integrates the parameter from the SOD network and generates more informative supervisions for the output of the SOD network during training, as illustrated in Figure 2b.

With PUM, we can obtain smooth labels z by feeding an input image x into $\mathcal{F}_{\text{PUM}}$ as

$$z^{(t)} = \mathcal{F}_{\text{PUM}}(x; \theta^{(t)}_{\text{PUM}}), \tag{3}$$

where predictions z are of the same size as ground truth y. While y is $\in \{0,1\}^{H\times W}$, z is $\in (0,1]^{H\times W}$

With smooth labels z, we regularize $\mathcal{F}_{\text{SOD}}$ by

$$\mathcal{L}_{\text{PUM}} = \sum_{i=1}^{H}\sum_{j=1}^{W} (p^{(t)}_{i,j} - z^{(t)}_{i,j})^2, \tag{4}$$

where $p \in (0,1]^{H\times W}$ are the predictions of $\mathcal{F}_{\text{SOD}}$, and $z \in (0,1]^{H\times W}$ are the obtained smooth labels by $\mathcal{F}_{\text{PUM}}$.

3.1.2. SAM: Sample Adaptive Module

Although smooth labels z aim at providing more informative supervisions for the output of the SOD network, they are noisy and unreliable at the beginning of the training. Figure 3 shows that the quality of the smooth labels is getting better and better as the training goes. To alleviate the noise incurred by smooth labels z, we introduce an adaptive weight λ defined by

$$\lambda = \exp(-\zeta \cdot (y \cdot \log z^{(t)} + (1-y) \cdot \log(1 - z^{(t)}))), \tag{5}$$

where y is the ground truth, $z^{(t)}$ are the smooth labels at training iteration t, and ζ is a hyperparameter. The definition of λ implies that λ is always positive, and that when z is significantly different from y, which means the quality of the smooth labels is bad, λ will be close to zero. In this way, instead of directly enforcing $\mathcal{F}_{\text{SOD}}$ to mimic the output from $\mathcal{F}_{\text{PUM}}$, our Sample Adaptive Module (SAM) measures the quality of generated smooth labels and strikes a balance between ground truth and the smooth labels. As a result, the overall loss function of SIT is

$$\mathcal{L}_{\text{SIT}} = \mathcal{L}_{\text{SOD}} + \lambda \cdot \mathcal{L}_{\text{PUM}}, \tag{6}$$

where $\mathcal{L}_{\text{PUM}}$ is defined in Equation (4), and $\mathcal{L}_{\text{SOD}}$ is the commonly used SOD loss function defined by

$$\mathcal{L}_{\text{SOD}} = \sum_{i=1}^{H}\sum_{j=1}^{W} (y_{i,j} \cdot \log p_{i,j} + (1 - y_{i,j}) \cdot \log(1 - p_{i,j})). \tag{7}$$

At the beginning of training, z is noisy, which leads to a very small λ to punish the impact of $\mathcal{L}_{\text{PUM}}$. As the training goes, the quality of z will be better and better and λ will be larger, which makes the impact of $\mathcal{L}_{\text{PUM}}$ more significant. In other words, as the quality of smooth labels becomes better, F_{SOD} will learn more from F_{PUM}.

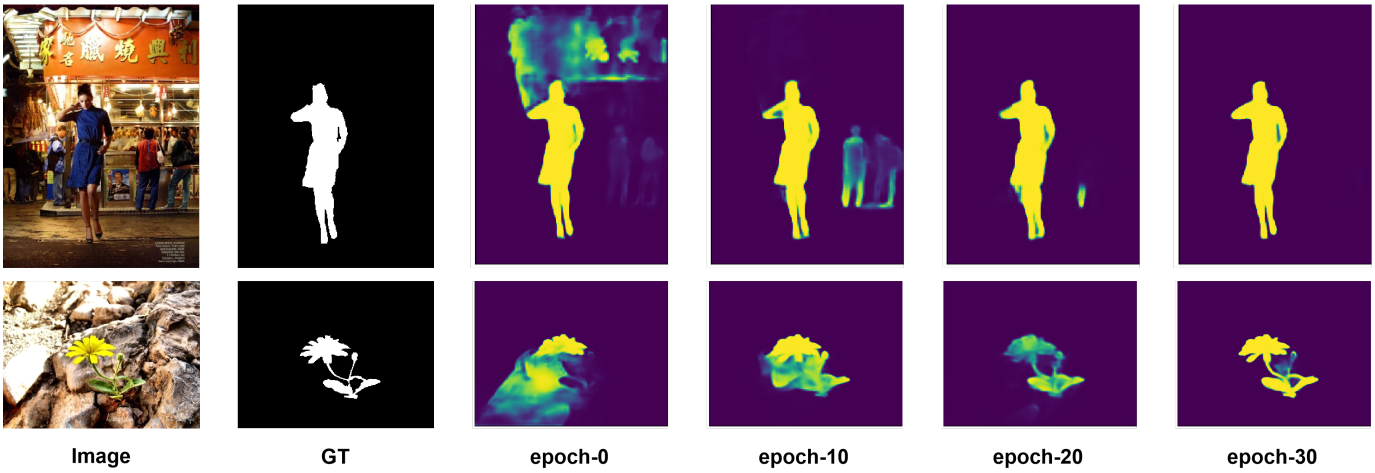


Figure 3. The visualization of the smooth labels generated via the Progressively Updated Module (PUM) at different training epochs. GT: Ground truth.

3.2. Augmentation-Based Consistent Learning

We design an augmentation-based consistent learning (ACL) scheme to keep the model's robustness for the salient-related data augmentation. By regularizing the model at both the prediction level and feature level, ACL helps the SOD model to capture more discriminative contextual information and improve the robustness.

3.2.1. Regularizing with Prediction Consistency

We first augment the raw image x with task-specific augmentation methods (detailed in Section 4.1.3) and obtain its augmentations $\hat{x}$ and its corresponding saliency prediction $\hat{p}$. Next, we regularize the prediction consistency between the raw image and its augmentations by

$$\mathcal{L}_{\text{logit}} = \sum_{i=1}^{H} \sum_{j=1}^{W} (p_{i,j}^{(t)} - \hat{p}_{i,j}^{(t)})^2, \tag{8}$$

where p is the prediction of x and $\hat{p}$ is the prediction of $\hat{x}$.

3.2.2. Regularizing with Feature Consistency

Moreover, we introduce a multiscale feature consistency loss $\mathcal{L}_{\text{feature}}$ to regularize the feature consistency for each layer of encoder E_i between the raw image x and augmented image $\hat{x}$ by

$$\mathcal{L}_{\text{feature}} = \sum_{i=1}^{N} \left\| E_i(x; \theta_{\text{SOD}}^{(t)}) - E_i(\hat{x}; \theta_{\text{SOD}}^{(t)}) \right\|_F^2, \tag{9}$$

where N is the number of feature levels in encoder E, $E_i(x)$ are the features of x at the i-th layer of E, $E_i(\hat{x})$ are the features of $\hat{x}$ at the i-th layer, and $\| \cdot \|_F$ is the Frobenius norm of the matrix.

3.3. The Training Procedure

Finally, we put all things together and train $\mathcal{F}_{\text{SOD}}$ in an end-to-end manner. The total loss $\mathcal{L}_{\text{total}}$ is defined as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{SIT}} + \mathcal{L}_{\text{logit}} + \mathcal{L}_{\text{feature}}. \tag{10}$$

The optimization procedure of our proposed method is summarized in Algorithm 1. It is worth noting that our method is orthogonal to the network architectures; thus, it can work as a plug-and-play module to boost the existing methods.

Algorithm 1 SIT and ACL training strategy at the *t* iteration

```
Input: $x$, an input images; $y$, its ground-truth
Networks: $\mathcal{F}_{\text{SOD}}(\cdot;\theta_{\text{SOD}})$, an SOD network with its parameters; $\mathcal{F}_{\text{PUM}}(\cdot;\theta_{\text{PUM}})$, a PUM network with its parameters
Output: $\theta_{\text{SOD}}$, $\theta_{\text{PUM}}$.
Initialization: $\theta^{(0)}_{\text{PUM}} \leftarrow \theta^{(0)}_{\text{SOD}}$
for $t = 1$ to $T$ do
    // SIT Self-Improvement Training Strategy
    $p = \mathcal{F}_{\text{SOD}}(x;\theta^{(t-1)}_{\text{SOD}})$ // Obtain saliency prediction
    $z = \mathcal{F}_{\text{PUM}}(x;\theta^{(t-1)}_{\text{PUM}})$ // Obtain smooth label
    Compute the SIT loss $\mathcal{L}_{\text{SIT}}$ defined in Equation (6)

    // ACL Augmentation-based Consistent Learning
    $\hat{x} = \text{Augment}(x)$ // Get augmentation
    $\hat{p} = \mathcal{F}_{\text{SOD}}(\hat{x};\theta^{(t-1)}_{\text{SOD}})$ // Obtain saliency prediction
    Compute $\mathcal{L}_{\text{logit}}$ and $\mathcal{L}_{\text{feature}}$ // defined in Equations (8) and (9)

    // Network Optimization
    Compute the total loss $\mathcal{L}_{\text{total}}$ defined in Equation (10)
    $\theta^{(t)}_{\text{SOD}} \leftarrow \text{Adam}(\mathcal{L}_{\text{total}}, \theta^{(t-1)}_{\text{SOD}})$ // Optimize SOD network
    $\theta^{(t)}_{\text{PUM}} \leftarrow \eta \cdot \theta^{(t-1)}_{\text{PUM}} + (1-\eta) \cdot \theta^{(t)}_{\text{SOD}}$ // Update PUM
end for
return $\theta_{\text{SOD}}$, $\theta_{\text{PUM}}$
```

4. Experiment

In this section, we evaluate our method. First, we describe the experiment setup in Section 4.1. Next, we conduct a series of ablation studies to demonstrate the impact of each component in our proposed framework and compare our method with others in Section 4.2. Furthermore, we carry out a detailed analysis of different design choices of our proposed SIT and ACL in Section 4.3. Finally, we present a visual comparison in Section 4.4.

4.1. Experiment Setup

4.1.1. Datasets

We evaluate our method on five popular datasets as shown in Table 1, including ECSSD [10] with 1000 images, PASCAL-S [8] with 850 images, DUT-OMRON [9] with 5168 images, HKU-IS [11] with 4447 images, and DUTS [42] with 15,572 images. All datasets are human-labeled with pixel-wise ground truth for quantitative evaluations. DUTS is currently the largest SOD dataset, which is divided into 10,553 training images (DUTS-TR) and 5019 testing images (DUTS-TE). We follow [40,54] to use DUTS-TR as the training dataset and the others as the testing datasets.

Table 1. The datasets we use. Only part of DUTS is used at training time and the rest are used at testing.

Datasets	Training Images	Testing Images
DUTS [42]	10,533	5019
ECSSD [10]	–	1000
DUT-OMRON [9]	–	5168
HKU-IS [11]	–	4447
PASCAL-S [8]	–	850

4.1.2. Evaluation Metrics

For all the experiments, we use four evaluation metrics to measure the performance, including the Mean Absolute Error (MAE), Mean F-measure (mF), structural similarity measure ($S_\alpha, \alpha = 0.5$) [55], and E-measure (E_ξ) [56].

First, MAE is a metric that computes the average absolute difference between the predicted saliency map p and its corresponding ground-truth map y pixel-by-pixel as follows

$$MAE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |p_{i,j} - y_{i,j}|, \tag{11}$$

where H and W are the height and width of p correspondingly. Second, the F-measure, termed mF, is a metric that evaluates both precision and recall comprehensively, and we provide the mean F-measure using varying fixed (0–255) thresholds. Next, the S-measure, termed S_α, is a metric that consists of the region-aware (S_r) and object-aware (S_o) structural similarity:

$$S_\alpha = \alpha \cdot S_o + (1 - \alpha) \cdot S_r, \tag{12}$$

where α is set to 0.5 to weigh the balance. Finally, the E-measure, termed E_ξ, is a metric that takes both the image-level mean value and local pixel matching information into consideration to evaluate the similarity between the prediction and the ground truth.

4.1.3. Implementation Details

Our framework is implemented via PyTorch. As our goal is to design a general model independent framework, we choose three state-of-the-art SOD algorithms: F^{3}Net [57], MINet [58], and GateNet [34] as baselines and integrate our methods into these different baselines. ResNet-50 [59], pre-trained on ImageNet, is used as the backbone network for all the three baselines. For training the hyperparameters' setting, we follow different baseline's implementation details according to their paper. For the data augmentation method in ACL, we use CutOut [60]. Specifically, we will conduct Cutout on the image area where there are no salient objects.

4.2. Ablation Study

First, we carry out ablation experiments to validate the effectiveness of our proposed SIT strategy and ACL scheme. Then, we integrate the proposed framework into different SOD baseline models. Since our method is orthogonal to the network architectures, we conduct all ablation studies on the basis of F^{3}Net due to its flexibility and effectiveness without a loss of generality.

4.2.1. Effectiveness of SIT

The integration of SIT (the second row in Table 2) gives performance gains of 0.7%, 0.3%, and 0.2%, in terms of mF, S_α, and E_ξ on the DUTS-TE dataset over the F^{3}Net baseline, respectively. Such results validate that the SIT strategy serves as a useful way to reduce the learning difficulty and improve the performance.

Table 2. Ablation study for SIT and ACL on the F^{3}Net [57] baseline on the DUTS-TE dataset. SIT: Self-Improvement Training. ACL: Augmentation-based Consistent Learning. ↑ and ↓ indicate that the larger and smaller scores are better, respectively. The best results are highlighted in bold.

SIT	**ACL**	**MAE** ↓	mF ↑	S_α ↑	E_ξ ↑
—	—	0.035	0.840	0.888	0.902
✓	—	0.035	**0.847**	**0.891**	0.904
—	✓	0.035	0.845	0.888	**0.905**
✓	✓	**0.034**	**0.846**	**0.891**	**0.904**

4.2.2. Effectiveness of ACL

As shown in the third row of Table 2, simply embedding the ACL scheme into the F^{3}Net baseline also helps improve the performance on both mF and E_ξ on the DUTS-TE dataset by 0.5% and 0.3%, respectively. It reveals that the robustness of the model is improved by enforcing consistency regularization for augmented images in ACL. The visual effects of ACL are illustrated in Figure 4. We can see that ACL helps effectively suppress the distracting background and accurately locate the salient objects because the richer contextual information can be captured via the consistency learning process.

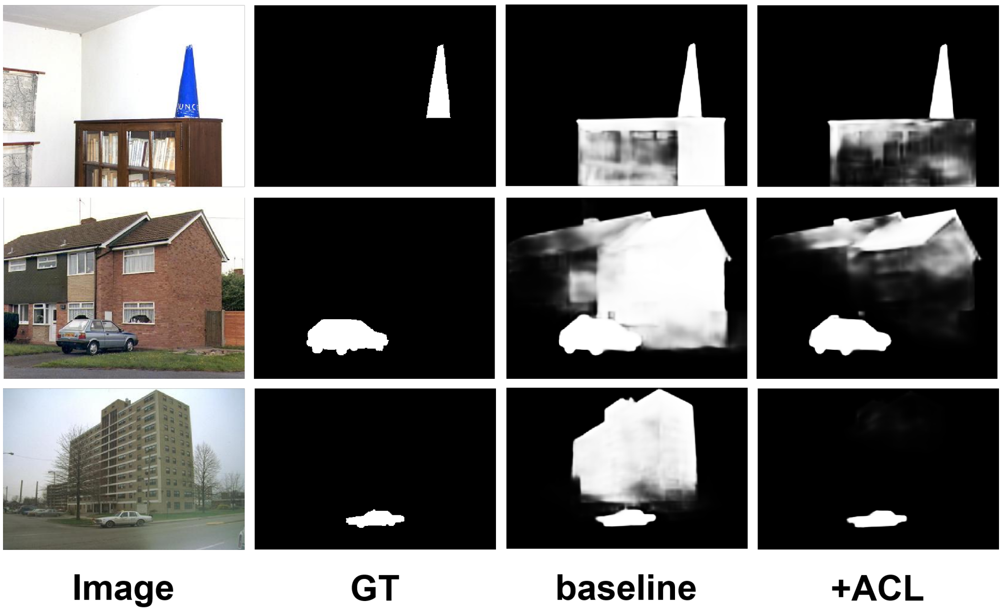


Figure 4. Visual comparisons for showing the benefits of the proposed methods. GT: Ground truth; ACL: Augmentation-based Consistent Learning scheme.

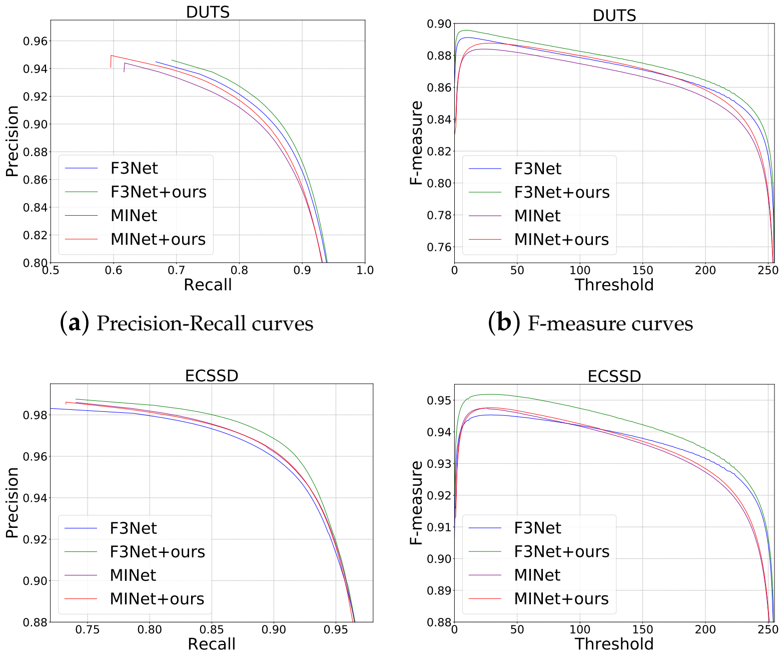
4.2.3. The Whole Framework

To demonstrate the effectiveness of the proposed methods, we compare it against 13 state-of-the-art SOD algorithms, including AFNet [33], BASNet [40], CPD-R [54], BMPM [24], R^{3}Net [61], PiCA-R [23], DGRL [22], TDBU [62], PoolNet [63], PAGE [32], RAS [64], C2SNet [39], and F^{3}Net [57]. As shown in Table 3, the performances of these baselines are boosted considerably by being integrated with our proposed framework.

Compared with the baselines, the mF results of our method on the DUTS-TE dataset are improved by 0.6%, 0.6%, and 0.2% based on three different baselines, respectively. In addition, Figure 5 shows the standard PR curves and the F-measure curves of the aforementioned baselines on the DUTS dataset, which can evaluate the performance of the models comprehensively. From these curves, we can observe that the models trained with our framework consistently outperform all corresponding baselines under different thresholds, which means that our framework has an excellent capability to detect salient regions and generate accurate saliency maps. All of these reveal that the proposed framework is effective on different SOTA methods and achieves a clearly better-averaged performance gain without modifying any architecture.

Table 3. Performance comparison with 12 state-of-the-art methods over five datasets. MAE (smaller is better), mean F-measure (mF, larger is better), S-measure (S_α, larger is better), and E-measure (E_ξ, larger is better) are utilized to evaluate the model's performance. ↑ and ↓ indicate that the larger and smaller scores are better, respectively. The best results are highlighted in red. Our model ranks first on most datasets and metrics.

Algorithm	**ECSSD**				**PASCAL-S**				**DUTS-TE**				**HKU-IS**				**DUT-OMRON**			
	MAE ↓	mF ↑	S_α ↑	E_ξ ↑	**MAE** ↓	mF ↑	S_α ↑	E_ξ ↑	**MAE** ↓	mF ↑	S_α ↑	E_ξ ↑	**MAE** ↓	mF ↑	S_α ↑	E_ξ ↑	**MAE** ↓	mF ↑	S_α ↑	E_ξ ↑
C2SNet	0.059	0.853	0.882	0.906	0.086	0.761	0.822	0.835	0.066	0.710	0.817	0.841	0.051	0.839	0.873	0.919	0.079	0.664	0.780	0.817
RAS	0.055	0.890	0.894	0.916	0.102	0.782	0.792	0.832	0.060	0.750	0.838	0.861	0.045	0.874	0.888	0.931	0.063	0.711	0.812	0.843
R^{3}Net	0.051	0.883	0.910	0.914	0.101	0.775	0.809	0.824	0.067	0.716	0.837	0.827	0.047	0.853	0.894	0.921	0.073	0.690	0.819	0.814
PiCA-R	0.046	0.886	0.917	0.913	0.075	0.798	0.849	0.833	0.051	0.759	0.869	0.862	0.043	0.870	0.904	0.936	0.065	0.717	0.832	0.841
BMPM	0.044	0.894	0.911	0.914	0.073	0.803	0.840	0.838	0.049	0.762	0.861	0.859	0.039	0.875	0.906	0.937	0.063	0.698	0.809	0.839
DGRL	0.043	0.903	0.906	0.917	0.074	0.807	0.834	0.836	0.051	0.764	0.846	0.863	0.037	0.881	0.896	0.941	0.063	0.709	0.810	0.843
PAGE	0.042	0.906	0.912	0.920	0.077	0.810	0.835	0.841	0.052	0.777	0.854	0.869	0.037	0.882	0.903	0.940	0.062	0.736	0.824	0.853
AFNet	0.042	0.908	0.913	0.918	0.070	0.821	0.844	0.846	0.046	0.792	0.867	0.879	0.036	0.888	0.905	0.942	0.057	0.738	0.826	0.853
TDBU	0.041	0.880	0.918	0.922	0.071	0.779	0.844	0.852	0.048	0.767	0.865	0.879	0.038	0.878	0.907	0.942	0.061	0.739	0.837	0.854
PoolNet	0.039	0.915	0.921	0.924	0.074	0.822	0.845	0.850	0.040	0.809	0.883	0.889	0.032	0.899	0.916	0.949	0.055	0.747	0.835	0.863
BASNet	0.037	0.880	0.916	0.921	0.076	0.775	0.832	0.847	0.048	0.791	0.866	0.884	0.032	0.895	0.909	0.946	0.056	0.756	0.836	0.869
CPD-R	0.037	0.917	0.918	0.925	0.072	0.824	0.842	0.849	0.043	0.805	0.869	0.886	0.034	0.891	0.905	0.944	0.056	0.747	0.825	0.866
PoolNet	0.039	0.915	0.921	0.924	0.075	0.810	0.836	0.847	0.040	0.809	0.883	0.889	0.033	0.893	0.913	0.946	0.056	0.747	0.836	0.863
PoolNet+ours	0.039	0.918	0.922	0.924	0.074	0.815	0.841	0.848	0.039	0.814	0.884	0.892	0.031	0.899	0.915	0.950	0.052	0.752	0.838	0.870
F^{3}Net	0.033	0.925	0.924	0.927	0.062	0.840	0.855	0.859	0.035	0.840	0.888	0.902	0.028	0.910	0.917	0.953	0.053	0.766	0.838	0.870
F^{3}Net+ours	0.032	0.930	0.927	0.929	0.061	0.837	0.855	0.861	0.034	0.846	0.891	0.904	0.027	0.916	0.921	0.955	0.052	0.769	0.842	0.870
MINet	0.033	0.924	0.925	0.927	0.063	0.829	0.850	0.851	0.037	0.828	0.884	0.898	0.029	0.909	0.919	0.953	0.055	0.755	0.833	0.865
MINet+ours	0.033	0.926	0.925	0.927	0.059	0.835	0.857	0.859	0.036	0.834	0.887	0.901	0.029	0.910	0.920	0.954	0.054	0.756	0.834	0.869
GateNet	0.040	0.916	0.920	0.924	0.067	0.819	0.851	0.851	0.040	0.807	0.885	0.889	0.033	0.899	0.915	0.949	0.055	0.746	0.838	0.861
GateNet+ours	0.038	0.915	0.924	0.924	0.065	0.820	0.856	0.857	0.039	0.808	0.888	0.891	0.031	0.901	0.921	0.951	0.055	0.746	0.839	0.862



(**a**) Precision-Recall curves (**b**) F-measure curves

(**c**) Precision-Recall curves (**d**) F-measure curves

Figure 5. Performance comparison with baseline models and our method on the DUTS dataset. The **first column** shows comparison of precision–recall curves. The **second column** shows comparison of F-measure curves over different thresholds. As a result, our method improve the performance of different baseline models.

4.3. Analysis on Design Choices

We analyze the design choices of our method from three aspects. First, we explore the updating strategy of PUM. Second, we conduct an ablation study of the selection of momentum coefficient. Finally, we show the effectiveness of the proposed SAM.

4.3.1. Updating Strategy of PUM

To obtained smooth labels in SIT, we need to update the PUM module from the SOD network. A straightforward updating strategy is to update the PUM module only once per epoch, which is similar to the Π model [65]. In this paper, we adopt a momentum-updating manner to integrate models of different steps to the PUM module progressively. From Table 4, we can see that all the momentum-updating strategy settings outperform the epoch-based updating strategy setting. With the momentum coefficient η in the range of 0.9 to 0.9999, the momentum-updating strategy could achieve a stable performance improvement, exceeding the epoch-based updating strategy on average by 1.1% on mF and 0.8% in S_α.

Table 4. The effect of different updating strategies of PUM in our proposed SIT on the DUTS-TE dataset. 'Epoch-based' denotes updating θ_{PUM} only once per epoch. 'Iteration-based' denotes updating θ_{PUM} by Equation (2). η denotes the momentum coefficient. ↑ and ↓ indicate that the larger and smaller scores are better, respectively. The best results are highlighted in bold.

Updating Strategy	η	**MAE** ↓	mF ↑	S_α ↑	E_ξ ↑
Epoch Based	—	0.038	0.833	0.882	0.895
Iteration Based	0.9	0.035	0.843	**0.891**	0.901
Iteration Based	0.99	**0.034**	**0.846**	**0.891**	**0.904**
Iteration Based	0.999	0.035	**0.846**	0.889	0.903
Iteration Based	0.9999	0.035	0.841	0.888	0.902

4.3.2. The Selection of Momentum Coefficient η

As described in Section 3.1, the momentum coefficient η determines the update speed of θ_{PUM}. Our model's sensitivity to η is shown in Table 4. When η is set to 0.99, the model can achieve the best performance on all metrics. We have revised this part to present our results more clearly.

4.3.3. Effect of Sample Adaptive Module

In this Sample Adaptive Module (SAM), we introduce an adaptive weight to dynamically tune the loss contribution between ground truth and smooth labels. To demonstrate the effectiveness of our SAM, we compared our dynamic adaptive weight strategy with a fixed weight strategy. We choose three different fixed weights, which are 0.3, 0.5, and 0.7, respectively. From Table 5, we can observe that our SAM method exceeds all the fixed weight settings. Compared with the best fixed weight setting (λ = 0.3), the mF of our SAM method exceeds it by 0.5%, which illustrates the effectiveness of SAM.

Table 5. The effect of the fixed λ compared with the adaptive λ in SAM on the DUTS-TE dataset. ↑ and ↓ indicate that the larger and smaller scores are better, respectively. The best results are highlighted in bold.

$\boldsymbol{\lambda}$	MAE ↓	$\boldsymbol{mF}$ ↑	$\boldsymbol{S_\alpha}$ ↑	$\boldsymbol{E_\xi}$ ↑
0.3	0.035	0.841	0.889	0.901
0.5	0.036	0.838	0.885	0.899
0.7	0.040	0.829	0.883	0.891
SAM	**0.034**	**0.846**	**0.891**	**0.904**

The hyperparameter ζ in Equation (5) is used to modulate the loss contribution of the PUM module. Thus, we conduct a series of experiments to evaluate the effect of hyperparameter ζ. As shown in Table 6, setting ζ to 70 makes the model learn well from the PUM module and obtain the best performance with 0.846 on mF.

Table 6. The effect of the scale factor ζ in the SAM module on the DUTS-TE dataset. ↑ and ↓ indicate that the larger and smaller scores are better, respectively. The best results are highlighted in bold.

$\boldsymbol{\zeta}$	MAE ↓	$\boldsymbol{mF}$ ↑	$\boldsymbol{S_\alpha}$ ↑	$\boldsymbol{E_\xi}$ ↑
30	0.035	0.843	0.889	0.901
70	**0.034**	**0.846**	**0.891**	**0.904**
110	0.037	0.839	0.885	0.898

4.4. Visualization Analysis

4.4.1. Visualization of Smooth Labels

In this part, we show the smooth labels generated via the PUM at different training iterations. As shown in Figure 3, we can observe that the PUM gradually improves the quality of the smooth labels. As the learning process progresses, the smooth labels not only obtain a sharper boundary but also suppress the distractors in the background.

4.4.2. Visualization of Feature Attention Maps

To further demonstrate the effectiveness of our framework, we visualize some attention maps in Figure 6. It can be observed that even if the squirrel has very low contrast with the ground (see the second column), by using our framework, the high contrast between the object region and the background is always maintained, thereby making the salient objects be effectively distinguished. In addition, our framework can effectively suppress distraction objects (see the fifth column). The people in the corner of the image are suppressed well in the feature attention map.

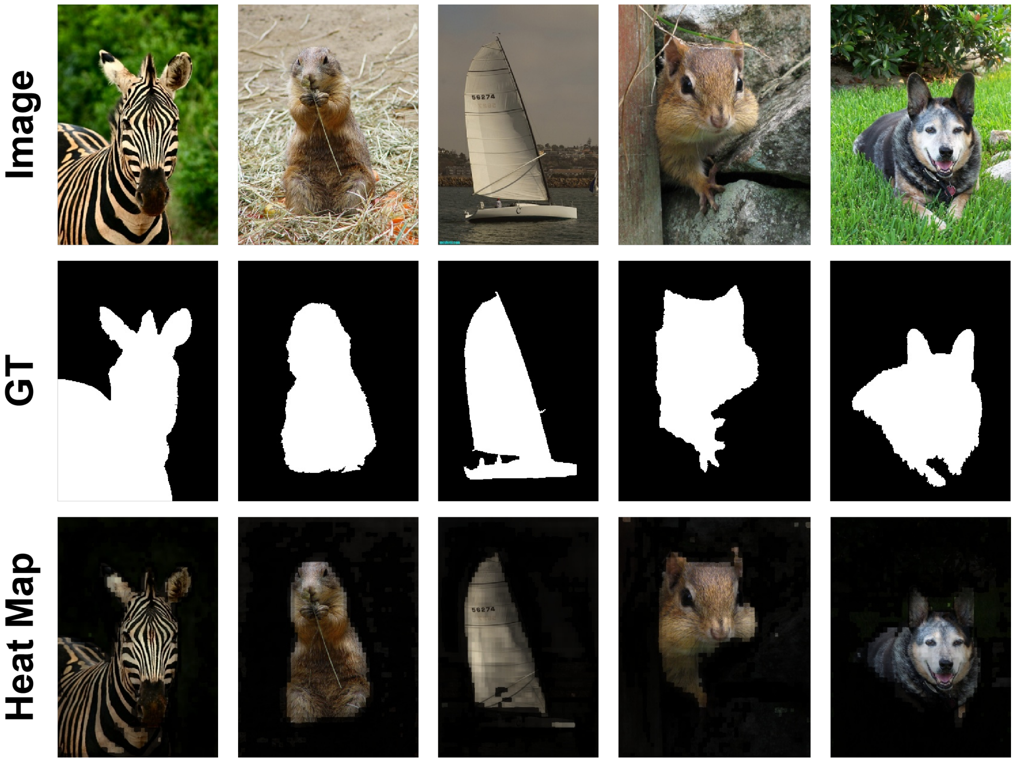


Figure 6. Visualization of attention of feature maps. The last row represents attention map for intermediate feature. Best viewed in colors.

4.4.3. Visual Comparison with Baseline Methods

To evaluate the robustness and effectiveness of our framework, we visualize some saliency maps and exhibit some typical images from the public test dataset of saliency detection in Figure 7.

In the first, second, and fifth rows, the complex backgrounds. such as seats and windows in the first row, stacked boxes in the second row, and books in the fifth row, are suppressed in ours but highlighted in all other baselines. Especially for the first and second rows, the backgrounds are suppressed well. Presumably, it is because the color of these backgrounds' objects are similar. Thus, our method can accurately locate and suppress all of them. In the third row, the baseline methods are severely affected by the gray box. Our framework can not only better pick out the salient objects accurately, but it also well suppresses these distractors. In the fourth and sixth row, the baseline methods fail to capture complete salient objects, such as the person on the left in the fourth row and the elephant on the right in the sixth row. However, our framework can accurately locate these salient objects.

In a word, our proposed framework can possess good robustness and effectiveness in detecting various salient objects.

4.5. Discussion

In this part, we will discuss the strengths and weaknesses of our method. Let us review the visual comparison in Figure 6. The third column shows a sailing boat. Our method misses its hull. The fourth column is a squirrel. Our method misses its paw. The fifth column is a dog. Our method misses its forelegs. These errors have a common point: the method misses their spatial details.

That being said, our method can predict results with spatial details as shown in the first row of Figure 7. In this case, all of our models (F^3Net+ours, MINet+ours, and GateNet+ours) can detect the contour of the person's feet. However, by looking at the pictures more carefully, one can find that the detection of the foot's contour should be attributed to the baseline models (F^3Net, MINet, and GateNet). F^3Net+ours has a similar foot's contour as F^{3}Net. Similar results can be found in MINet vs. MINet+ours and GateNet vs. GateNet+ours.

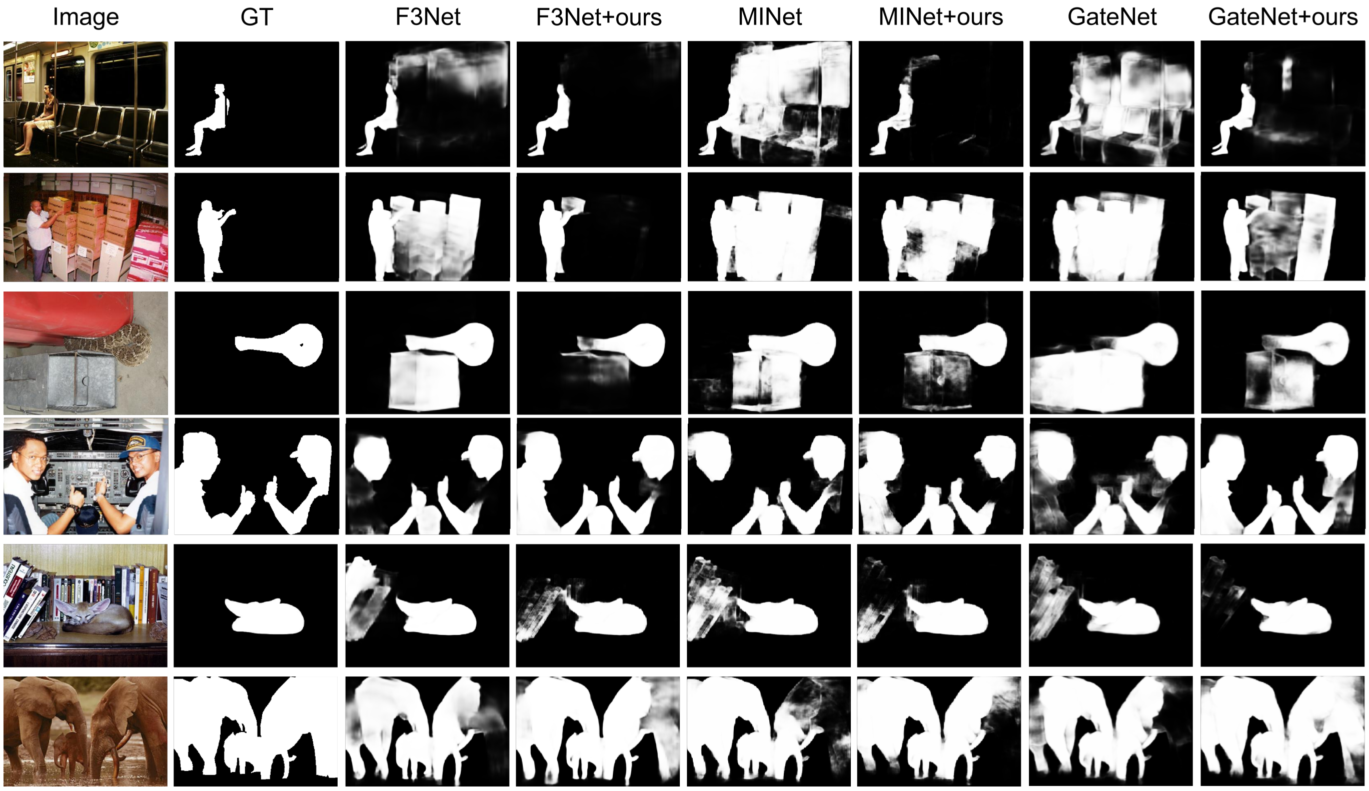


Figure 7. Salient object detection examples on several popular datasets. F^{3}Net+ours, MINet+ours, and GateNet+ours indicate the original architectures trained with our proposed SIT and ACL. SIT and ACL provide more reasonable smooth labels for the model and reduce the effect of distractors.

Our method aims to alleviate the learning difficulty and improve the robustness of the SOD methods. We do not design specific contour-based techniques to improve the performance, but these techniques are orthogonal and complementary to our work.

5. Conclusions

In this paper, we present a learning framework to reduce the learning difficulty for the SOD task. A novel self-improvement training (SIT) strategy is designed to generate smooth labels, which alleviates the learning difficulty. Moreover, by regularizing the prediction consistency and multi-level feature consistency in augmentation-based consistent learning (ACL), the robustness of the model can be further improved. Comprehensive benchmarks on several popular datasets illustrate the advantage of the proposed framework. A further ablation study shows the effectiveness of each method of our framework. Briefly, our framework can play a plug-and-play role to be easily embedded in the existing SOD networks to achieve a promising performance gain, without any modification of the network architecture.

Author Contributions: Conceptualization, S.L. and H.W.; methodology, S.L. and H.W.; software, H.Z.; writing—original draft, S.L. and H.Z.; writing—review and editing S.L.; supervision, X.L.; project administration, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The DUTS dataset: http://saliencydetection.net/duts/. The ECSSD dataset: https://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/dataset.html. The DUT-OMRON dataset: http://saliencydetection.net/dut-omron/. The HKU-IS dataset: https://i.cs.hku.hk/~yzyu/research/deep_saliency.html. The PASCAL-S dataset: http://cbi.gatech.edu/salobj/ (accessed on 29 November 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mahadevan, V.; Vasconcelos, N. Saliency-based discriminant tracking. In Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR), Miami Beach, FL, USA, 20–25 June 2009; pp. 1007–1013.
2. Fang, H.; Gupta, S.; Iandola, F.N.; Srivastava, R.K.; Deng, L.; Dollár, P.; Gao, J.; He, X.; Mitchell, M.; Platt, J.C.; et al. From captions to visual concepts and back. In Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1473–1482.
3. Zhang, W.; Liu, H. Study of Saliency in Objective Video Quality Assessment. *IEEE Trans. Image Process.* **2017**, *26*, 1275–1288. [CrossRef] [PubMed]
4. Zhao, R.; Ouyang, W.; Wang, X. Unsupervised Salience Learning for Person Re-identification. In Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 25–27 June 2013; pp. 3586–3593.
5. Liu, G.; Fan, D. A Model of Visual Attention for Natural Image Retrieval Computing Companion. In Proceedings of the International Conference on Information Science and Cloud, Guangzhou, China, 7–8 December 2013; pp. 728–733.
6. Hou, Q.; Jiang, P.; Wei, Y.; Cheng, M. Self-Erasing Network for Integral Object Attention. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 547–557.
7. Zhang, D.; Han, J.; Zhao, L.; Meng, D. Leveraging Prior-Knowledge for Weakly Supervised Object Detection under a Collaborative Self-Paced Curriculum Learning Framework. *Int. J. Comput. Vis.* **2019**, *127*, 363–380. [CrossRef]
8. Li, Y.; Hou, X.; Koch, C.; Rehg, J.M.; Yuille, A.L. The Secrets of Salient Object Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 280–287.
9. Yang, C.; Zhang, L.; Lu, H.; Ruan, X.; Yang, M.H. Saliency Detection via Graph-Based Manifold Ranking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 25–27 June 2013; pp. 3166–3173.
10. Yan, Q.; Xu, L.; Shi, J.; Jia, J. Hierarchical Saliency Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 25–27 June 2013; pp. 1155–1162.
11. Li, G.; Yu, Y. Visual Saliency Based on Multiscale Deep Features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5455–5463.
12. Wang, L.; Lu, H.; Ruan, X.; Yang, M. Deep networks for saliency detection via local estimation and global search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3183–3192.
13. Zhao, R.; Ouyang, W.; Li, H.; Wang, X. Saliency detection by multi-context deep learning. In Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1265–1274.
14. Zhang, J.; Sclaroff, S.; Lin, Z.; Shen, X.; Price, B.L.; Mech, R. Unconstrained Salient Object Detection via Proposal Subset Optimization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5733–5742.
15. Lee, G.; Tai, Y.; Kim, J. Deep Saliency with Encoded Low Level Distance Map and High Level Features. In Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 660–668.
16. He, S.; Lau, R.W.H.; Liu, W.; Huang, Z.; Yang, Q. SuperCNN: A Superpixelwise Convolutional Neural Network for Salient Object Detection. *Int. J. Comput. Vis.* **2015**, *115*, 330–344. [CrossRef]
17. Kim, J.; Pavlovic, V. A Shape-Based Approach for Salient Object Detection Using Deep Learning. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 455–470.
18. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
19. Fan, D.P.; Cheng, M.M.; Liu, J.J.; Gao, S.H.; Hou, Q.; Borji, A. Salient Objects in Clutter: Bringing Salient Object Detection to the Foreground. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
20. Li, G.; Xie, Y.; Lin, L.; Yu, Y. Instance-Level Salient Object Segmentation. In Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
21. Zhang, P.; Wang, D.; Lu, H.; Wang, H.; Yin, B. Learning Uncertain Convolutional Features for Accurate Saliency Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
22. 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 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
23. Liu, N.; Han, J.; Yang, M.H. PiCANet: Learning Pixel-Wise Contextual Attention for Saliency Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.

24. Zhang, L.; Dai, J.; Lu, H.; He, Y.; Wang, G. A Bi-Directional Message Passing Model for Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
25. Ji, W.; Li, X.; Wei, L.; Wu, F.; Zhuang, Y. Context-Aware Graph Label Propagation Network for Saliency Detection. *IEEE Trans. Image Process.* **2020**, *29*, 8177–8186. [CrossRef] [PubMed]
26. Liu, Z.; Wang, Y.; Tu, Z.; Xiao, Y.; Tang, B. TriTransNet: RGB-D Salient Object Detection with a Triplet Transformer Embedding Network. In Proceedings of the ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021; pp. 4481–4490.
27. Wang, X.; Jiang, B.; Wang, X.; Luo, B. MTFNet: Mutual-Transformer Fusion Network for RGB-D Salient Object Detection. *arXiv* **2021**, arXiv:2112.01177.
28. Mao, Y.; Zhang, J.; Wan, Z.; Dai, Y.; Li, A.; Lv, Y.; Tian, X.; Fan, D.; Barnes, N. Transformer Transforms Salient Object Detection and Camouflaged Object Detection. *arXiv* **2021**, arXiv:2104.10127.
29. Qiu, Y.; Liu, Y.; Zhang, L.; Xu, J. Boosting Salient Object Detection with Transformer-based Asymmetric Bilateral U-Net. *arXiv* **2021**, arXiv:2108.07851.
30. Liu, N.; Zhang, N.; Wan, K.; Shao, L.; Han, J. Visual Saliency Transformer. In Proceedings of the International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 4722–4732.
31. Zhang, X.; Wang, T.; Qi, J.; Lu, H.; Wang, G. Progressive Attention Guided Recurrent Network for Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 714–722.
32. Wang, W.; Zhao, S.; Shen, J.; Hoi, S.C.H.; Borji, A. Salient Object Detection With Pyramid Attention and Salient Edges. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1448–1457.
33. Feng, M.; Lu, H.; Ding, E. Attentive Feedback Network for Boundary-Aware Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
34. Zhao, X.; Pang, Y.; Zhang, L.; Lu, H.; Zhang, L. Suppress and Balance: A Simple Gated Network for Salient Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.
35. Hou, Q.; Cheng, M.; Hu, X.; Borji, A.; Tu, Z.; Torr, P.H.S. Deeply Supervised Salient Object Detection with Short Connections. *TPAMI* **2019**, *41*, 815–828. [CrossRef] [PubMed]
36. Zhang, P.; Wang, D.; Lu, H.; Wang, H.; Ruan, X. Amulet: Aggregating Multi-level Convolutional Features for Salient Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 202–211.
37. Wang, Y.; Wang, R.; Fan, X.; Wang, T.; He, X. Pixels, Regions, and Objects: Multiple Enhancement for Salient Object Detection. In Proceedings of the IEEE Conference Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada, 18–22 June 2023; pp. 10031–10040.
38. Zhao, J.; Liu, J.; Fan, D.; Cao, Y.; Yang, J.; Cheng, M. EGNet: Edge Guidance Network for Salient Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8778–8787.
39. Li, X.; Yang, F.; Cheng, H.; Liu, W.; Shen, D. Contour Knowledge Transfer for Salient Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 355–370.
40. Qin, X.; Zhang, Z.; Huang, C.; Gao, C.; Dehghan, M.; Jagersand, M. BASNet: Boundary-Aware Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
41. Chen, Z.; Zhou, H.; Lai, J.; Yang, L.; Xie, X. Contour-Aware Loss: Boundary-Aware Learning for Salient Object Segmentation. *IEEE Trans. Image Process.* **2021**, *30*, 431–443. [CrossRef] [PubMed]
42. Wang, L.; Lu, H.; Wang, Y.; Feng, M.; Wang, D.; Yin, B.; Ruan, X. Learning to Detect Salient Objects with Image-Level Supervision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 2017; pp. 136–145.
43. Li, G.; Xie, Y.; Lin, L. Weakly Supervised Salient Object Detection Using Image Labels. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
44. Zeng, Y.; Zhuge, Y.; Lu, H.; Zhang, L.; Qian, M.; Yu, Y. Multi-source weak supervision for saliency detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 6074–6083.
45. Zhang, J.; Xie, J.; Barnes, N. Learning Noise-Aware Encoder-Decoder from Noisy Labels by Alternating back-propagation for saliency detection. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 349–366.
46. Piao, Y.; Wang, J.; Zhang, M.; Lu, H. MFNet: Multi-filter directive network for weakly supervised salient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 4136–4145.
47. Zhang, J.; Yu, X.; Li, A.; Song, P.; Liu, B.; Dai, Y. Weakly-Supervised Salient Object Detection via Scribble Annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 2020; pp. 12546–12555.
48. Yu, S.; Zhang, B.; Xiao, J.; Lim, E.G. Structure-consistent weakly supervised salient object detection with local saliency coherence. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 4–7 February 2021; pp. 3234–3242.

49. Gao, S.; Zhang, W.; Wang, Y.; Guo, Q.; Zhang, C.; He, Y.; Zhang, W. Weakly-Supervised Salient Object Detection Using Point Supervision. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; pp. 670–678.
50. Wu, Z.; Wang, L.; Wang, W.; Xia, Q.; Chen, C.; Hao, A.; Li, S. Pixel is All You Need: Adversarial Trajectory-Ensemble Active Learning for Salient Object Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 2883–2891.
51. Zhou, H.; Chen, P.; Yang, L.; Xie, X.; Lai, J. Activation to saliency: Forming high-quality labels for unsupervised salient object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *33*, 743–755. [CrossRef]
52. Zhou, H.; Qiao, B.; Yang, L.; Lai, J.; Xie, X. Texture-Guided Saliency Distilling for Unsupervised Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2023; pp. 7257–7267.
53. Song, Y.; Gao, S.; Xing, H.; Cheng, Y.; Wang, Y.; Zhang, W. Towards End-to-End Unsupervised Saliency Detection with Self-Supervised Top-Down Context. In Proceedings of the ACM International Conference on Multimedia, Ottawa, Canada, 29 October–3 November 2023; pp. 5532–5541.
54. Wu, Z.; Su, L.; Huang, Q. Cascaded Partial Decoder for Fast and Accurate Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
55. Fan, D.; Cheng, M.; Liu, Y.; Li, T.; Borji, A. Structure-Measure: A New Way to Evaluate Foreground Maps. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4558–4567.
56. Fan, D.; Gong, C.; Cao, Y.; Ren, B.; Cheng, M.; Borji, A. Enhanced-alignment Measure for Binary Foreground Map Evaluation. In Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 698–704.
57. Wei, J.; Wang, S.; Huang, Q. F3Net: Fusion, Feedback and Focus for Salient Object Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
58. Pang, Y.; Zhao, X.; Zhang, L.; Lu, H. Multi-Scale Interactive Network for Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
59. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
60. DeVries, T.; Taylor, G.W. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv* **2017**, arXiv:1708.04552.
61. Deng, Z.; Hu, X.; Zhu, L.; Xu, X.; Qin, J.; Han, G.; Heng, P. R^{3}Net: Recurrent Residual Refinement Network for Saliency Detection. In Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 684–690.
62. Wang, W.; Shen, J.; Cheng, M.; Shao, L. An Iterative and Cooperative Top-Down and Bottom-Up Inference Network for Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5968–5977.
63. Liu, J.; Hou, Q.; Cheng, M.; Feng, J.; Jiang, J. A Simple Pooling-Based Design for Real-Time Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3917–3926.
64. Chen, S.; Tan, X.; Wang, B.; Hu, X. Reverse Attention for Salient Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 236–252.
65. Laine, S.; Aila, T. Temporal Ensembling for Semi-Supervised Learning. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.