

# Test Time Adaptation with Regularized Loss for Weakly Supervised Salient Object Detection

Olga Veksler  
 University of Waterloo, Canada  
 oveksler@uwaterloo.ca

## Abstract

*It is well known that CNNs tend to overfit to the training data. Test-time adaptation is an extreme approach to deal with overfitting: given a test image, the aim is to adapt the trained model to that image. Indeed nothing can be closer to the test data than the test image itself. The main difficulty of test-time adaptation is that the ground truth is not available. Thus test-time adaptation, while intriguing, applies to only a few scenarios where one can design an effective loss function that does not require ground truth. We propose the first approach for test-time Salient Object Detection (SOD) in the context of weak supervision. Our approach is based on a so called regularized loss function, which can be used for training CNN when pixel precise ground truth is unavailable. Regularized loss tends to have lower values for the more likely object segments, and thus it can be used to fine-tune an already trained CNN to a given test image, adapting to images unseen during training. We develop a regularized loss function particularly suitable for test-time adaptation and show that our approach significantly outperforms prior work for weakly supervised SOD.*

## 1. Introduction

A well known problem with CNNs is that they tend to overfit to the training data. One approach to deal with overfitting is test-time adaptation [30]. A model is trained on the training data, and then fine-tuned for a few epochs during test time on a given test image. The main difficulty of test-time adaptation is that there is no ground truth for the test image. Thus test-time adaptation has been previously used for only a few applications [11, 16, 22, 30, 45] where a suitable loss function can be designed without ground truth.

We propose the first approach for test-time Salient Object Detection (SOD). The goal of SOD is to find image regions that attract human attention. Convolutional Neural Networks (CNNs) [14, 15] brought a significant breakthrough for SOD [18, 28, 46, 47]. Traditional training of

CNNs for SOD requires pixel precise ground truth, but obtaining such annotations is a substantial effort. Therefore, there is an increased interest in semi-supervised [20, 23] and weakly supervised SOD [17, 25, 33, 40, 41, 44]. Weak supervision requires less annotation effort, compared to semi-supervision. In this paper, we assume image level supervision, the weakest supervision type, where one provides only images containing salient objects, with no other annotation.

Most image level weakly supervised SOD approaches [17, 25, 40, 41, 44] are based on noisy pseudo labels, constructed from unsupervised SOD methods. These approaches are hard to modify for test-time adaptation, as at test time, one has only one image to create noisy pseudo-ground truth samples, but combating noise requires diverse samples.

Our test-time adaptation is based on the approach in [33]. Different from the other image level supervised SOD methods, the approach in [33] does not require pseudo labels. They design a *regularized loss*<sup>1</sup> to use for CNN training without pixel precise annotations. Regularized loss models class-independent properties of object shapes and is likely to have lower values for segmentations that separate the object from the background. This makes regularized loss approach particularly suitable for test time adaptation.

We design a regularized loss function tailored for test time adaptation. The main problem with regularized loss in [33] is that it may result in a trivial empty solution if hyperparameters of the loss function are not chosen correctly. When training on a large dataset, an occasional empty result is not a big problem, but when training on a single test image, an empty result is disastrous, catastrophically worsening the performance. Thus we must develop a hyperparameter setting method that avoids empty solutions. However, in the absence of ground truth, setting hyperparameter weights is not trivial. We propose a method for setting hyperparameters specific for each test image such that an empty solution is avoided. This method is crucial for obtaining good results

<sup>1</sup>In the context of CNNs, regularization is a term often used to refer to the norm regularization on the network weights [8]. Here, regularized loss refers to the loss function on the output of CNN.

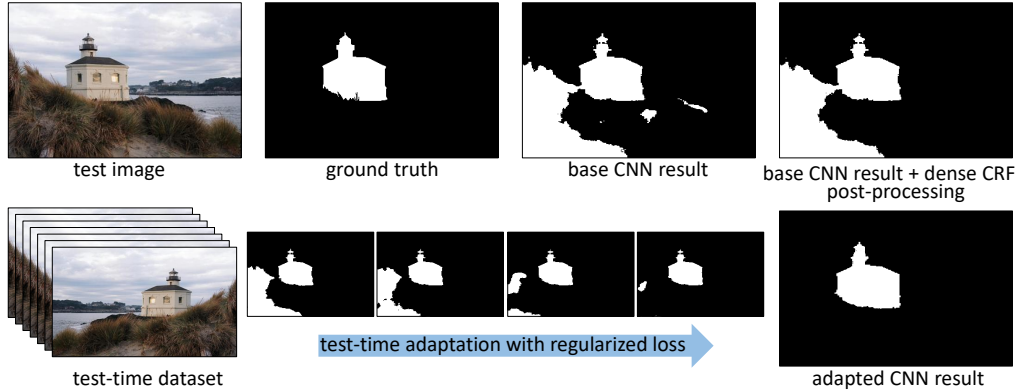


Figure 1. Overview of our approach. Top: test image, ground truth, output of base CNN on the test image, and the result of dense CRF [13] post-processing of the base CNN result. Bottom: test-time dataset, CNN output on the test image during several test time epochs, and the final result of the adapted CNN.

during test-time adaptation.

Fig. 1 is an overview of our approach. First we train CNN for SOD in weakly supervised setting with image level labels using [33]. The result is called the *base* CNN. The top row of Fig. 1 shows a test image, ground truth, and the result of the base CNN. Given a test image, we create a small training dataset by augmentation, see Fig. 1, bottom, left. Then we fine tune the base CNN on the small dataset, using our version of regularized loss, Fig. 1, bottom, middle. The resulting CNN is called the *adapted* CNN.

The result of base CNN (Fig. 1, top, third image) has many erroneous regions. Sometimes dense CRF [13] is used for post processing to improve performance. We apply dense CRF to the output of base CNN (Fig. 1, top, right). Dense CRF removes small spurious regions but is unable to remove the large erroneous regions as these are salient with high confidence according to the base CNN.

In contrast, our approach is able to produce much better results, Fig. 1, bottom, right. This is because the base CNN has a high value of regularized loss for this test image. As the fine tuning proceeds, CNN is forced to find alternative segmentations with better loss values, resulting in increasingly better segmentations. Unlike CRF post processing, our adapted CNN is able to remove the large erroneous regions, as their high confidence values according to the base CNN are ignored, and these regions give higher values of regularized loss. Both dense CRF and our approach use CRF models. However, as opposed to post-processing, we use CRF for CNN supervision, enabling CNN to learn a better segmentation.

Our experiments on the standard benchmarks show that test time adaptation significantly improves performance, achieving the new state-of-art in image level weakly supervised SOD.

This paper is organized as follows: Sec. 2 is related

work, Sec. 3 explains the approach in [33], Sec. 4 describes our approach, and Sec. 5 presents experiments.

## 2. Related Work

### Weakly Supervised Salient Object Detection

Most prior work on image level weak supervision consists of two stages. In the first stage, pseudo labels for SOD task are generated, and in the second stage, CNN is trained on the pseudo labels.

One of the first methods to use image level weak supervision for SOD is in [34]. Their method generates pseudo labels by training CNN for object category prediction. In their setting, image level supervision consists of a large set of 200 object classes categories from Image net. After training for object class prediction, they extract foreground heat maps capturing potential object regions. These generalize to unseen categories, and provide initial pseudo labels. Next, they have a self-training stage which alternates between estimating pseudo labels and training CNN on them. For more accurate pseudo-labels, the predicted pseudo-labels are refined with dense CRF [13].

Later image level SOD approaches [17, 25, 27, 40, 41, 44] use one or more conventional weak (i.e. unsupervised) saliency methods [4, 9, 10, 19] for pseudo labels. In [40], they use multiple weak saliency methods and fuse their results in pseudo-ground truth. During the fusion process, based on the difficulty of training data, they estimate the corresponding confidence measure and utilize it for training with their pseudo ground truth to better handle label noise.

In [44] they develop a noise modeling module, which enables them to deal with noisy saliency maps obtained from multiple weak saliency methods in a probabilistic way. They later extend their method to rely only on a single weak saliency method [41].

In [17], they approach the noise in pseudo labels by exploiting dense CRF [13]. Initial pseudo labels are generated by a weak saliency method. Then they alternate applying dense CRF and CNN training. Dense CRF corrects the noisy labels through spatial consistency and structure preservation, while CNN is trained on the current version of pseudo labels to update and improve the current model.

In [25], they develop a curriculum framework to incrementally refine the pseudo-labels. Instead of directly using the saliency maps produced by weak saliency methods for training, they first train CNNs to generate pseudo labels for each weak saliency method. This improves the pseudo labels as CNNs are forced to learn representations across a broad set of images. Then pseudo labels are further refined via an iterative self-supervision technique.

In [27] they utilize multiple pseudo labels to achieve robustness to noise. They design a multi-filter directive CNN which uses multiple directive filters and a multi-guidance loss to integrate multiple pseudo labels.

In [42] they propose a noise-aware encoder-decoder to deal with noisy pseudo labels. Their method has a saliency predictor that maps input images to clean saliency maps, and a noise generator. The model that represents noisy labels is a sum of these two models. They train their model to simultaneously infer the corresponding latent vector of each noisy label and the saliency predictor.

All approaches above develop elaborate techniques to deal with noisy pseudo labels. In contrast to the above approaches, in [33] they develop a method for image level supervised SOD which does not rely on pseudo labels from conventional weak saliency methods. The main tool in [33] is regularized loss, based on sparse CRF [2]. The approach in [33] uses off-the-shelf Unet [29] architecture, consists of a single stage, has an intuitive and easy to interpret loss function, and outperforms other image level supervised SOD methods. We review [33] in detail in Sec 3.

In addition to image level weakly supervised SOD methods, there are approaches that use additional sources of supervision. They do not necessarily rely on conventional weak saliency methods. In [39], they use additional weak sources, such as captions, etc. In [38,43] they use scribbles, and in [21] they use boxes, both much stronger forms of weak supervision. We achieve comparable or better results without additional weak sources.

### Regularized Loss

Regularized loss for weak supervision was first used in [31], for semantic segmentation supervised with scribbles. It was subsequently generalized in [24,32].

### Test Time Adaptation

The idea of test time adaptation has been used before. For example, in the context of traditional computer vision, it was used in [30] for online updates in visual tracking and in [11] for adapting detectors from images to videos. In the

context of deep learning, it was used in [45] for person re-identification, in [16] for image inpainting, and in [22] for enforcing consistency constraints in depth estimation. In all these methods, the loss to perform test time adaptation is application specific and is unrelated to our approach.

## 3. Regularized Loss for SOD

In this section, we review the approach in [33], which is designed for image level weakly supervised segmentation for datasets with a single object class. It naturally applies to SOD, since there is only one class, the salient object. The main idea is to design a regularized loss which incorporates the likely properties of generic objects and to use this loss to train CNN instead of cross entropy on ground truth.

Regularized loss is applied to the output of CNN, which is the same size as the input image. Let  $\mathbf{x}$  denote CNN output, and  $x_p$  denote the output for pixel  $p$ . The last layer of CNN is sigmoid, so that  $x_p \in [0, 1]$ . Background corresponds to 0 and salient object to 1.

Regularized loss in [33] consists of a weighted combination of several components. The most important component is sparse CRF loss [2].

$$L_{\text{crf}}(\mathbf{x}) = \frac{1}{|\mathcal{P}|} \sum_{(p,q) \in \mathcal{N}} e^{-\frac{\|C_p - C_q\|^2}{2\sigma^2}} \cdot |x_p - x_q|, \quad (1)$$

where  $\mathcal{P}$  is the set of all pixels in the image,  $\mathcal{N}$  is the set of neighboring pixel pairs on a 4-connected grid, and  $C_p \in \mathbb{R}^3$  is the color of  $p$ . Parameter  $\sigma$  controls the edge strength and is set to  $\sigma = 0.15$  in all experiments.

Usually CNN produces a sharp distribution, i.e. most  $x_p$  are close either to 0 or 1. Thus if two neighboring pixels are not assigned to the same class, there is a penalty which depends on the strength of the edge between them. Sparse CRF loss is low when segment boundary aligns with image intensity edges. Salient objects are likely to cause image intensity edges, and thus sparse CRF is the main driving tool for salient object discovery. The lowest value of sparse-CRF is zero, achieved at trivial solutions: empty, everything classified as object, or full, everything classified as background. Thus one cannot train with sparse CRF alone.

Another component of regularized loss is batch volumetric loss. It encourages the average object size in a batch of images to be half of the image size. This loss is useful to prevent both empty and full trivial solutions. Averaging over a batch makes the loss less strict: some batch objects can be significantly smaller or larger than half of the image.

The next component of regularized loss is minimum volume loss. It encourages the object to be of at least a certain size. It is more likely to be valid in practice, compared to volumetric batch loss, as objects can be expected to be at least a certain minimum size. The last two components of regularized loss are border and center losses. The border

loss encourage the image border to be the background, and the center loss encourages image center be the object. This is a realistic assumption for many, but not all, images containing a salient object.

Training CNN with regularized loss is difficult and tends to get stuck in a bad local minimum. In [33] they devise a two-stage strategy for training. For our approach, we start with base CNN trained with the method developed in [33].

In our test-time adaptation, we only use sparse CRF loss in Eq. (1), minimum volume loss, and border loss. We now describe minimum volume and border loss in detail. Let  $\bar{x} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} x_p$ , i.e. the normalized object size. Minimum volume loss  $L_m$  penalizes segmentations if the normalized object size is less than  $\text{obj}_{min}$

$$L_m(\mathbf{x}) = \text{ReLU}^2(\text{obj}_{min} - \bar{x}). \quad (2)$$

Following [33], we set  $\text{obj}_{min} = 0.15$ .

Let  $\mathcal{B}$  be the set of pixels on the image border of width  $w = 3$ .<sup>2</sup> The border loss  $L_b(\mathbf{x})$  is

$$L_b(\mathbf{x}) = \left( \frac{1}{|\mathcal{B}|} \sum_{p \in \mathcal{B}} x_p \right)^2. \quad (3)$$

## 4. Test Time Adaptation

We now describe our approach. We start with an overview in Sec. 4.1. We describe our regularized loss in Sec. 4.2, and our hyperparameter setting method in Sec. 4.3.

### 4.1. Overview

The overview of our approach is in Fig. 1. Given a test image  $I$ , we first construct a small training dataset  $D(I)$  from  $I$  by data augmentation. There are various forms of augmentation. However, since we are training without pixel precise ground truth, we need to avoid augmentations which make the salient object more difficult to detect in an image. For example, large crops can remove the salient object. Similarly, adding random image noise changes the distribution of intensity edges, and sparse CRF in Eq. (1) may become less useful for object discovery. We use small random vertical and horizontal image shifts, and random additive shift to color channels (while keeping the shift the same for all image pixels). We found random vertical and horizontal shifts to be the most effective.

Next, we take the base CNN trained for SOD task in image level weakly supervised setting, according to the method proposed in [33], and train it on  $D(I)$  for a small number of epochs using the regularized loss approach described in Sec. 4.3. The resulting CNN is called the adapted CNN. Training for a small number of iterations ensures that the adapted CNN retains most of the information useful for

<sup>2</sup>The width of the border can be adjusted relative to the image size, but we train on fixed size images,  $256 \times 256$ .

SOD learned from the large training dataset, as well as ensuring efficiency. In practice, we train for 10 epochs.

### 4.2. Regularized Loss

We now describe our loss function. For test-time adaptation, we only use some of the regularized loss components from [33]. We do not use batch volumetric loss as it encourages the salient segment to be half of the image size and is far from realistic for most test images. We also do not use center loss. It assumes that the image center belongs to the salient object and is likely not true for many specific test images we wish to adapt CNN for.

In our regularized loss, we use sparse CRF, minimum volume, and border losses, Eq. (1, 2, 3). Sparse CRF is the most important loss component for discovering salient objects, as it favours segments that align with image intensity edges. Minimum volume loss is necessary to prevent collapse to an empty solution favoured by sparse CRF. Border loss encourages image border to be assigned to the background, and is a realistic assumption for most test images. The complete regularized loss for test time adaptation is

$$L_{\text{reg}}(\mathbf{x}) = \lambda_{\text{crf}} L_{\text{crf}}(\mathbf{x}) + \lambda_m L_m(\mathbf{x}) + \lambda_b L_b(\mathbf{x}). \quad (4)$$

### 4.3. Hyperparameter Setting Method

Consider the loss components in Eq. (4). To get the best performance during test-time adaptation, it is important to chose a good setting of their relative weights. Each test image has its own best setting, however, we cannot find this setting as the ground truth is not available. Thus we must develop a hyperparameter setting method that does not rely on ground truth.

Out of three components in Eq. (4), the border loss is the least important as it contributes the least to the salient object discovery. Indeed, if our only knowledge about a segmentation is that the image border is assigned to the background, then this gives us almost no information to judge whether this segmentation corresponds to the salient object. Therefore, we set  $\lambda_b$  to a low value of 1. Out of the two remaining components, sparse CRF is more important for object discovery. Since objects tend to cause intensity edges in an image, a segment that aligns to image edges is more likely to correspond to a salient object, compared to a segment that just obeys the minimum size constraints. Thus sparse CRF weight  $\lambda_{\text{crf}}$  should be much higher compared to the minimum volume weight  $\lambda_m$ . At the same time, the minimum volume weight should be sufficiently large to prevent the collapse to a trivial empty solution. We can compute this sufficiently large weight from the solution obtained from our base CNN.

We now explain how to set  $\lambda_m$  relative to  $\lambda_{\text{crf}}$  so that a trivial solution is avoided. Let  $\mathbf{x}^i$  be the output of the base

CNN on input image  $I$ . Assuming that  $\mathbf{x}^i$  is not empty, we set the weight of minimum volume loss large enough to make the loss on  $\mathbf{x}^i$  smaller than the loss of an empty solution. Then switching to an empty solution from the initial  $\mathbf{x}^i$  is too costly.

The results produced by the base CNN tend to be sharp, that is most pixels have values close to either 0 or 1. Therefore we will assume that the trivial empty solution corresponds to all pixels having value 0 in the output. Let  $\mathbf{x}^0$  denote such a trivial empty solution, i.e.  $x_p^0 = 0$  for all pixels  $p$ . Suppose we wish to set sparse-CRF weight to  $\lambda_{\text{crf}}$ . We need to set  $\lambda_m$  to a value that makes collapse to an empty solution too costly. For this, we need  $\lambda_m$  that satisfies

$$L_{\text{reg}}(\mathbf{x}^i) = L_{\text{reg}}(\mathbf{x}^0) + \epsilon, \quad (5)$$

where  $\epsilon > 0$  is a small constant, set in practice to 0.1.

Plugging  $\mathbf{x}^0$  in Eqs.(1, 2, 3) we obtain

$$L_b(\mathbf{x}^0) = 0, \quad L_{\text{crf}}(\mathbf{x}^0) = 0, \quad L_m(\mathbf{x}^0) = \text{obj}_{\text{min}}^2. \quad (6)$$

Plugging Eq. (6) into Eq. (5) and solving for  $\lambda_m$ , we get our formula for setting  $\lambda_m$

$$\lambda_m = \frac{\lambda_{\text{crf}} L_{\text{crf}}(\mathbf{x}^i) + L_b(\mathbf{x}^i) - \epsilon}{\text{obj}_{\text{min}}^2 - L_m(\mathbf{x}^i)}. \quad (7)$$

Eq. (7) applies when the output  $\mathbf{x}^i$  of the base CNN is not a trivial empty solution. If  $\mathbf{x}^i$  is an empty solution, it means that the base CNN fails to extract a salient object. In this case, we set both CRF and minimum volume loss to be of equal weight,  $\lambda_{\text{crf}} = \lambda_m$ . This reflects the fact that extracting a salient object from the current image is difficult and the penalty for not satisfying the minim object size is increased. We do not set  $\lambda_m$  higher than  $\lambda_{\text{crf}}$  as some images may not contain a salient object, in which case an empty solution is appropriate. In addition, while setting  $\lambda_m$  high enough would result in some non-empty solution, most likely it would be an erroneous solution, as minimum volume loss is less useful for extracting object segments and, therefore, should not have a weight higher than that of sparse CRF. In practice, we set  $\lambda_{\text{crf}} = 10^3$  and compute  $\lambda_m$  for each test image using Eq. (7).

In Table 1 we experimentally evaluate the effectiveness of our method for setting  $\lambda_m$ . We randomly select 500 images from DUTO [37] dataset. We test our adaptive  $\lambda_m$  for each image, and the same fixed setting of  $\lambda_m$  for all test images, in a range from 1 to  $10^3$ , see Table 1. For this experiment, we use a fixed  $\lambda_{\text{crf}} = 10^3$ . The performance metric is  $F_\beta$ . The adaptive  $\lambda_m$  performs better than any fixed choice of  $\lambda_m$ . As expected, the performance degrades significantly for the larger values of  $\lambda_m$  as minimum volume loss starts to play a bigger role in the regularized loss function, however, it is less discriminating of salient objects. For smaller  $\lambda_m$ , the performance suffers as the results on some images collapse to an empty solution.

## 5. Experimental Results

Our implementation is in Pytorch using RTX 2080 GPU. For our base CNN, we use image level weakly supervised SOD approach [33]. We use Unet [29] architecture with ResNeXt [35] fixed features pretrained on Imagenet [5] and train on  $256 \times 256$  images. Unlike [33], we train on larger images and on a larger training dataset, thus getting better results for the base CNN than those reported in [33].

We train the base CNN on the training set of DUTS [34], which contains 10,553 images. We then evaluate our test-time CNN adaptation approach on the test set of DUTS (5019 images), DUT-OMRON [37] (5168 images), ECSSD [36] (1000 images), THUR [3] (6233 images), PascalS [19] (850 images), and HKU-IS [18] (4447 images).

For test time adaptation, given a test image  $I$ , we construct a test time adaptation dataset  $D(I)$  that has 16 images. These images are constructed from  $I$  by shifting to the left or right with probability 0.5, and, in addition, shifting up or down with probability 0.5. Each shift is chosen uniformly from the range  $[0, 15]$ . This creates a dataset of images similar to  $I$ , and it is unlikely that the salient object is removed out of the sample by these small shifts. We use Adam optimizer [12], a fixed learning rate 0.001, and batch size equal to the size of the test time adaptation dataset, set to 16 images. We train for 10 epoch. The images are scaled to resolution  $256 \times 256$ . Test time adaptation takes 2.8 seconds per image.

We use the standard metrics widely adopted for SOD. Our first metric is  $F_\beta$  score [1], with  $\beta^2 = 0.3$ . We also use *mae* [26], defined as the average absolute per pixel difference between the predicted saliency and ground truth. In addition, we use two newer metrics, namely  $S_\alpha$  [6], and  $E_s$  [7].  $S_\alpha$  evaluates the structural information for saliency map and region-aware and object-aware structural similarity between saliency maps and ground truth.  $E_s$  is based on unification of global and local information.

### 5.1. SOD results

In Table 2, we report the results of our approach and compare to prior image level weakly supervised SOD methods. Our results with fixed hyperparameter settings in Eq. (4) are in column ‘ours’ (with  $\lambda_m = 20$ , which is the best fixed setting according to Table 1). The results with our method for hyperparameter setting for each image individually, as described in Sec. 4.3, are in column ‘ours+h’. Except one case, all metrics for our approach with hyperparameter setting method are improved, often significantly. This shows that our hyperparameter setting method is crucial for obtaining a significantly improved performance.

Dense CRF [13] is often used to post-process CNN results. Therefore, we also post-process base CNN results with dense CRF for comparison. The performance of the base CNN is in column [33], and its performance after post-

$\lambda_m$	1	5	10	20	50	100	200	500	1000	adaptive
$F_\beta$	.741	.738	.752	.755	.742	.732	.728	.718	.703	.765

Table 1. Performance in terms of  $F_\beta$  score of fine tuning with a fixed value of  $\lambda_m$  vs. our adaptive  $\lambda_m$  computation according to Eq. (7). For all experiments,  $\lambda_{\text{crf}} = 10^3$ .

	Metric	[34]	[17]	[39]	[44]	[42]	[27]	[33]	[33]+dCRF	ours	ours+h
DUTS	$F_\beta \uparrow$	.654	.614	.684	.725	.747	.710	.753	.763	<u>.771</u>	<b>.795</b>
	$mae \downarrow$	.100	.116	.091	.075	.060	.076	.056	.055	<u>.054</u>	<b>.052</b>
	$E_s \uparrow$	.795	.772	.814	.853	<u>.859</u>	.839	.833	.842	.855	<b>.863</b>
	$S_\alpha \uparrow$	.748	.697	.759	.813	<b>.827</b>	.775	.791	.798	.800	<u>.817</u>
ECSSD	$F_\beta \uparrow$	.823	.797	.609	.810	.852	.854	.868	.873	<u>.882</u>	<b>.911</b>
	$mae \downarrow$	.104	.110	.109	.092	.071	.084	.063	.060	<u>.057</u>	<b>.046</b>
	$E_s \uparrow$	.869	.853	.673	.836	.883	.885	.884	<u>.886</u>	.885	<b>.909</b>
	$S_\alpha \uparrow$	.811	.802	.756	.846	<u>.860</u>	.834	.845	.855	.859	<b>.883</b>
DUTO	$F_\beta \uparrow$	.603	.622	.609	.597	.701	.646	.697	.710	<u>.721</u>	<b>.752</b>
	$mae \downarrow$	.109	.101	.109	.103	.070	.087	.074	.069	<u>.066</u>	<b>.057</b>
	$E_s \uparrow$	.768	.776	.673	.712	.816	.803	.821	.827	<u>.833</u>	<b>.842</b>
	$S_\alpha \uparrow$	.725	.752	.756	.733	<u>.791</u>	.742	.763	.771	.782	<b>.798</b>
PascalS	$F_\beta \uparrow$	.715	.693	.713	-	-	.751	.796	<u>.801</u>	<u>.801</u>	<b>.807</b>
	$mae \downarrow$	.139	.149	.135	-	-	.115	<u>.083</u>	<b>.082</b>	<b>.082</b>	.084
	$E_s \uparrow$	.791	.772	.790	-	-	.817	<u>.827</u>	<u>.827</u>	.826	<b>.828</b>
	$S_\alpha \uparrow$	.744	.717	.768	-	-	.770	.791	<u>.793</u>	.792	<b>.796</b>
HKUIS	$F_\beta \uparrow$	-	-	.814	.820	.878	.851	.866	.871	<u>.883</u>	<b>.899</b>
	$mae \downarrow$	-	-	.084	.065	.043	.059	.044	.040	<u>.039</u>	<b>.037</b>
	$E_s \uparrow$	-	-	.895	.858	.919	.921	.912	.925	<u>.926</u>	<b>.931</b>
	$S_\alpha \uparrow$	-	-	.818	.860	<b>.890</b>	.846	.857	.866	.871	<u>.879</u>
THUR	$F_\beta \uparrow$	-	-	-	.694	.719	-	.731	.749	<u>.753</u>	<b>.760</b>
	$mae \downarrow$	-	-	-	.086	.070	-	.073	.069	<u>.067</u>	<b>.064</b>
	$E_s \uparrow$	-	-	-	.807	.838	-	.840	.842	<u>.843</u>	<b>.844</b>
	$S_\alpha \uparrow$	-	-	-	.804	<u>.810</u>	-	.794	.798	.806	<b>.819</b>

Table 2. Comparison to SOD methods that also use image level weak supervision using metrics  $F_\beta$ ,  $mae$ ,  $E_s$ , and  $S_\alpha$ . We use  $\uparrow$  to indicate that the higher score is better, and  $\downarrow$  to indicate the lower score is better. The best result is in bold, and the second best is underlined.

processing with dense CRF is in column [33]+dCRF. Our approach is significantly better than [33] with dense CRF post-processing. This illustrates that it is important to include CRF during training in the loss function, rather than use CRF for just post-processing.

Our test time adaptation improves the performance of [33] across all metrics, except one case. For easier datasets (ECSSD, HKUIS), the performance gain is not as large as the base CNN already performs well, but for harder to fit datasets, the performance gain is significant.

Comparing to other SOD methods, our method is better in almost all cases except  $S_\alpha$  metric for *HKUIS* and *DUTS*. For most datasets, our approach significantly improves  $F_\beta$  metric over the second best method, notably by roughly 7% for DUTO and DUTS datasets.

In Table 3, we compare our method to weakly supervised SOD methods that use stronger form of weak supervision. In [21], they use bounding box supervision, and in [38, 43],

they use scribbles. Our method is better than the bounding box method [43] in all cases. Our method is almost always better than [43], and better in about half of the cases compared to [38]. If Fig. 2, we show qualitative comparisons of our method to the scribble supervised methods of [43] and [38]. For this illustration, we chose examples where all methods work relatively well, in terms of the standard metrics. However, our results have better detail preservation qualities, especially when the salient object has sharp edges on the boundary. Observe the detail preservation in the chair, in the last row. This is because CRF based loss function is adapted to the particular test image, driving the segmentation boundary to align with image edges.

## 5.2. Ablation Experiments

### Hyperparameters in Regularized Loss

We first discuss and perform ablation for hyperparameters of our regularized loss function, Eq. (4). The most im-

		DUTS	ECSSD	DUTO	PascalS	HKUIS	THUR
boxes [21]	$F_\beta \uparrow$	.736	.860	.686	-	.853	-
	$mae \downarrow$	.079	.072	.081	-	.058	-
	$E_s \uparrow$	.831	.889	.810	-	.897	-
	$S_\alpha \uparrow$	.789	.858	.776	-	.852	-
scribbles [43]	$F_\beta \uparrow$	.747	.865	.715	.788	.858	.718
	$mae \downarrow$	.062	.061	.068	.140	.047	.077
	$E_s \uparrow$	<u>.865</u>	.908	.835	.798	.923	.873
	$S_\alpha \uparrow$	-	-	-	-	-	-
scribbles [38]	$F_\beta \uparrow$	<b>.823</b>	<u>.900</u>	<b>.758</b>	<b>.823</b>	<u>.896</u>	<u>.755</u>
	$mae \downarrow$	<b>.049</b>	<u>.049</u>	<u>.060</u>	<b>.078</b>	<u>.038</u>	<u>.069</u>
	$E_s \uparrow$	<b>.890</b>	<u>.908</u>	<b>.862</b>	<b>.847</b>	<b>.938</b>	<u>.843</u>
	$S_\alpha \uparrow$	-	-	-	-	-	-
ours	$F_\beta \uparrow$	<u>.795</u>	<b>.911</b>	<u>.752</u>	<u>.807</u>	<b>.899</b>	<b>.760</b>
	$mae \downarrow$	<u>.052</u>	<b>.046</b>	<b>.057</b>	<u>.082</u>	<b>.037</b>	<b>.064</b>
	$E_s \uparrow$	.863	<b>.909</b>	<u>.842</u>	<u>.828</u>	<u>.931</u>	<b>.844</b>
	$S_\alpha \uparrow$	<b>.817</b>	<b>.883</b>	<b>.798</b>	<b>.796</b>	<b>.879</b>	<b>.819</b>

Table 3. Comparison to prior SOD methods that use stronger forms of weak supervision using metrics  $F_\beta$ ,  $mae$ ,  $E_s$ , and  $S_\alpha$ . In [21], they use bounding box supervision, and in [38, 43], they use scribbles. We use  $\uparrow$  to indicate that the higher score is better, and  $\downarrow$  to indicate the lower score is better. The best result is in bold, and second best is underlined.



Figure 2. Qualitative comparison. From left to right: input image, ground truth, results of [43], [38], and our results, respectively. Note that [43], [38] both use scribble supervision, a much stronger supervision form compared to what we use.

important parameter to set is  $\lambda_m$  in relation to  $\lambda_{\text{crf}}$ , as the correct relative settings of these two parameters ensures there is no collapse to a trivial solution. For this, we have de-

	with border loss	without border loss
DUTS	.795	.789
ECSSD	.911	.910
DUTO	.752	.748
PascalS	.807	.800
HKUIS	.899	.892
THUR	.760	.755

Table 4. Performance of our approach with border loss and without border loss. Performance metric is  $F_\beta$  score.

veloped an effective automatic approach, Sec. 4.3. Since  $\lambda_m$  is computed in relation to  $\lambda_{\text{crf}}$ , the setting of  $\lambda_{\text{crf}}$  can be arbitrary (up to the precision that can be handled by the software),  $\lambda_m$  is adjusted accordingly: for larger  $\lambda_{\text{crf}}$ ,  $\lambda_m$  will be automatically set larger according to Eq. (7).

The setting of  $\lambda_b$ , as long as it is much smaller than  $\lambda_{\text{crf}}$ , is not important. Parameter  $\lambda_b$  controls the border loss that encourages border pixels to be assigned to the background. While this loss is helpful, it is the least important regularized loss component. To show that this loss is the least important, we perform an experiment where we remove border loss from regularized loss ( $\lambda_b = 0$ ). The results are summarized in Table 4. Without border loss, the results worsen only a little. On the other hand, omitting either CRF or minimum volume losses from our loss function in Eq. (4) results in a catastrophic performance collapse.

In Eq. (1), we set  $\sigma = 0.15$  as in [33]. Since the training images are normalized, this setting works well across all datasets. We also experimented with other values of  $\sigma$ , see

	$\sigma = 0.1$	$\sigma = 0.15$	$\sigma = 0.20$
DUTS	.789	.795	.797
ECSSD	.894	.911	.915
DUTO	.751	.752	.743
PascalS	.803	.807	.803
HKUIS	.902	.899	.895
THUR	.763	.760	.757

Table 5. Performance of our approach for different  $\sigma$  values. Performance metric is  $F_\beta$  score.

$ D(I) $	2	4	8	16	32	64
$F_\beta \uparrow$	.879	.887	.890	.910	.915	.913
$mae \downarrow$	.058	.055	.049	.046	.045	.046

Table 6. Different choice of test time dataset size for ECSSD dataset.

# epoch	2	5	10	20	30	50
$F_\beta \uparrow$	.873	.883	.910	.915	.915	.912
$mae \downarrow$	.062	.053	.047	.045	.046	.047

Table 7. Different number of test time adaptation iterations for ECSSD dataset.

Table 5. The performance is stable for a range of  $\sigma$  values.

### 5.3. Other Hyperparameters

We now perform ablation experiments for various hyperparameter settings (other than those in the regularized loss function) of our approach. We perform experiments on ECSSD dataset, and evaluate the sensitivity of our approach to the size of the test time adaptation dataset  $D(I)$ , the number of epoch performed for test time adaptation, and different data augmentation strategies. For all the experiments, the performance metric is  $F_\beta$  and  $mae$ .

In Table 6, we show how performance depends on varying the size of the test time dataset  $D(I)$ . Our standard setting for the size of  $D(I)$  is 16. Decreasing it to 8 slightly degrades the performance, increasing it to 64 results in almost no change, but the running time is twice longer.

In Table 7, we show how performance depends on increasing the number of epoch for test time adaptation. Our standard setting for the number of epoch is 10. Decreasing it to 2 noticeably degrades the performance, increasing it to 50 results in only a slight improvement but significantly increases the running time.

In Table 8, we show the effect of different augmentation strategies. Our main experiments use small random shifts in the vertical and horizontal directions. We also tested adding random Gaussian noise to images, a standard data augmentation technique, but it significantly degrades the performance. Adding Gaussian noise decreases the effectiveness of our sparse CRF loss as neighboring pixels become less

type	shifts	Gauss noise	color perturbation	shifts+color
$F_\beta \uparrow$	.911	.854	.897	.909
$mae \downarrow$	.046	.060	.050	.049

Table 8. Different augmentation strategies when fine tuning on ECSSD dataset.

	DUTS	ECSSD	DUTO	PascalS	HKUIS	THUR
% improved	79.3	83.7	85.3	81.2	76.7	75.0
mean $F_\beta$ increase	.048	.053	.064	.17	.037	.036
mean $F_\beta$ decrease	.021	.009	.011	.08	.019	.015

Table 9. Percentage of images for which performance improves as the result of test-time adaptation. For the improved (and degraded) images, we also show the mean increase (and the decrease) in  $F_\beta$  score.

strongly connected. We also tested random color perturbation where all pixels change the value of a color channel by the same random value. This works much better than Gaussian noise, but not as well as random shifts. Shifts and color perturbation work as well as shifts alone.

### 5.4. Additional Experiments

During test-time adaptation, the majority, but not all images, improve. In Table 9, we show the percentage of images that improve, in terms of their  $F_\beta$  score, for each of the test datasets. In addition, we show the mean improvement of  $F_\beta$  score, averaged over the images that improve, and the mean decline of  $F_\beta$  score, averaged over the images that decline in performance. As can be seen from the table, for all dataset, over 75% of images improve during test-time adaptation. Also, the average improvement is significantly larger than the average decline of the results. Thus overall, test-time adaptation significantly improves the performance. In the supplementary materials, we show the most improved and the most degraded images during test time adaptation.

## 6. Conclusions

We propose the first approach to test time adaptation for SOD with image level weak supervision. We develop a regularized loss function particularly effective for test time adaptation, and a method to find an appropriate setting of the hyperparameters in our loss function. Our approach uses a standard CNN architecture and a loss function that is intuitive and simple to interpret. We achieve a new state of the art in image level supervised SOD, and a better or competitive result with the prior work that uses stronger forms of weak supervision. The drawback of our approach is that the running time for testing is more expensive, since the base CNN needs to be fine-tuned for each test image.



## References

- [1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1597–1604. IEEE, 2009. 5
- [2] Yuri Boykov, Olga Veksler, and Ramin Zabih. A variable window approach to early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1283–1294, December 1998. An earlier version of this work appeared in *CVPR '97*. 3
- [3] Ming-Ming Cheng, NiloyJ. Mitra, Xiaolei Huang, and Shi-Min Hu. Salienshape: group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014. 5
- [4] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):569–582, 2014. 2
- [5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 5
- [6] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE international conference on computer vision*, pages 4548–4557, 2017. 5
- [7] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 698–704, 2018. 5
- [8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT Press, 2016. 1
- [9] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2007. 2
- [10] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998. 2
- [11] Vidit Jain and Erik Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR 2011*, pages 577–584. IEEE, 2011. 1, 3
- [12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [13] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Neural Information Processing Systems*, pages 109–117, 2011. 2, 3, 5
- [14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, 2012. 1
- [15] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, December 1989. 1
- [16] Eunhye Lee, Jeongmu Kim, Jisu Kim, and Tae Hyun Kim. Restore from restored: Single-image inpainting. *arXiv preprint arXiv:2102.08078*, 2021. 1, 3
- [17] Guanbin Li, Yuan Xie, and Liang Lin. Weakly supervised salient object detection using image labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1, 2, 3, 6
- [18] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 478–487, June 2016. 1, 5
- [19] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287, 2014. 2, 5
- [20] Jiawei Liu, Jing Zhang, and Nick Barnes. Semi-supervised salient object detection with effective confidence estimation. *arXiv preprint arXiv:2112.14019*, 2021. 1
- [21] Yuxuan Liu, Pengjie Wang, Ying Cao, Zijian Liang, and Rynson WH Lau. Weakly-supervised salient object detection with saliency bounding boxes. *IEEE Transactions on Image Processing*, 30:4423–4435, 2021. 3, 6, 7
- [22] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020. 1, 3
- [23] Yunqiu Lv, Bowen Liu, Jing Zhang, Yuchao Dai, Aixuan Li, and Tong Zhang. Semi-supervised active salient object detection. *Pattern Recognition*, 123:108364, 2022. 1
- [24] Dmitrii Marin and Yuri Boykov. Robust trust region for weakly supervised segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6608–6618, October 2021. 3
- [25] Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mum-madi, Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox. Deepusps: Deep robust unsupervised saliency prediction via self-supervision. In *Advances in Neural Information Processing Systems*, pages 204–214, 2019. 1, 2, 3
- [26] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *Conference on Computer Vision and Pattern Recognition*, pages 733–740. IEEE, 2012. 5
- [27] Yongri Piao, Jian Wang, Miao Zhang, and Huchuan Lu. Mfnet: Multi-filter directive network for weakly supervised salient object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4136–4145, 2021. 2, 3, 6
- [28] 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
- [29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 3, 5
- [30] David A Ross, Jongwoo Lim, Rwei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual track-

- ing. *International journal of computer vision*, 77(1):125–141, 2008. [1](#), [3](#)
- [31] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised CNN segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 1818–1827, 2018. [3](#)
- [32] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised CNN segmentation. In *European Conference on Computer Vision*, pages 524–540, 2018. [3](#)
- [33] Olga Veksler. Regularized loss for weakly supervised single class semantic segmentation. In *European Conference on Computer Vision*, pages 348–365. Springer, 2020. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#)
- [34] 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*, 2017. [2](#), [5](#), [6](#)
- [35] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Conference on Computer Vision and Pattern Recognition*, pages 5987–5995, 2017. [5](#)
- [36] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Conference on Computer Vision and Pattern Recognition*, pages 1155–1162. IEEE, 2013. [5](#)
- [37] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Conference on Computer Vision and Pattern Recognition*, pages 3166–3173. IEEE, 2013. [5](#)
- [38] Siyue Yu, Bingfeng Zhang, Jimin Xiao, and Eng Gee Lim. Structure-consistent weakly supervised salient object detection with local saliency coherence. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 3234–3242. AAAI Press, 2021. [3](#), [6](#), [7](#)
- [39] Yu Zeng, Yunzhi Zhuge, Huchuan Lu, Lihe Zhang, Mingyang Qian, and Yizhou Yu. Multi-source weak supervision for saliency detection. In *Conference on Computer Vision and Pattern Recognition*, pages 6074–6083, 2019. [3](#), [6](#)
- [40] Dingwen Zhang, Junwei Han, and Yu Zhang. Supervision by fusion: Towards unsupervised learning of deep salient object detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4048–4056, 2017. [1](#), [2](#)
- [41] Jing Zhang, Yuchao Dai, Tong Zhang, Mehrtash Harandi, Nick Barnes, and Richard Hartley. Learning saliency from single noisy labelling: A robust model fitting perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2866–2873, 2021. [1](#), [2](#)
- [42] Jing Zhang, Jianwen Xie, and Nick Barnes. Learning noise-aware encoder-decoder from noisy labels by alternating back-propagation for saliency detection. In *European conference on computer vision*, pages 349–366. Springer, 2020. [3](#), [6](#)
- [43] Jing Zhang, Xin Yu, Aixuan Li, Peipei Song, Bowen Liu, and Yuchao Dai. Weakly-supervised salient object detection via scribble annotations. In *Conference on Computer Vision and Pattern Recognition*, pages 12546–12555, 2020. [3](#), [6](#), [7](#)
- [44] Jing Zhang, Tong Zhang, Yuchao Dai, Mehrtash Harandi, and Richard Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9029–9038, 2018. [1](#), [2](#), [6](#)
- [45] Shun Zhang, Jia-Bin Huang, Jongwoo Lim, Yihong Gong, Jinjun Wang, Narendra Ahuja, and Ming-Hsuan Yang. Tracking persons-of-interest via unsupervised representation adaptation. *International Journal of Computer Vision*, 128(1):96–120, 2020. [1](#), [3](#)
- [46] Ting Zhao and Xiangqian Wu. Pyramid feature attention network for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3085–3094, 2019. [1](#)
- [47] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. 2020. [1](#)