Disentangled High Quality Salient Object Detection

Lv Tang  Bo Li1*  Yijie Zhong  Shouhong Ding1  Mofei Song2,3
1Youtu Lab, Tencent, Shanghai, China
2The School of Computer Science and Engineering, Southeast University, Nanjing, China
3The Key Lab of Computer Network and Information Integration (Ministry of Education), Southeast University, Nanjing, China

Abstract

Aiming at discovering and locating most distinctive objects from visual scenes, salient object detection (SOD) plays an essential role in various computer vision systems. Coming to the era of high resolution, SOD methods are facing new challenges. The major limitation of previous methods is that they try to identify the salient regions and estimate the accurate objects boundaries simultaneously with a single regression task at low-resolution. This practice ignores the inherent difference between the two difficult problems, resulting in poor detection quality. In this paper, we propose a novel deep learning framework for high-resolution SOD task, which disentangles the task into a low-resolution saliency classification network (LRSCN) and a high-resolution refinement network (HRRN). As a pixel-wise classification task, LRSCN is designed to capture sufficient semantics at low-resolution to identify the definite salient, background and uncertain image regions. HRRN is a regression task, which aims at accurately refining the saliency value of pixels in the uncertain region to preserve a clear object boundary at high-resolution with limited GPU memory. It is worth noting that by introducing uncertainty into the training process, our HRRN can well address the high-resolution refinement task without using any high-resolution training data. Extensive experiments on high-resolution saliency datasets as well as some widely used saliency benchmarks show that the proposed method achieves superior performance compared to the state-of-the-art methods.

1. Introduction

Salient object detection (SOD) is derived with the goal of accurately detecting and segmenting the most distinctive objects from visual scenes. As a preliminary step, it plays an essential role in various visual systems, such as video object segmentation [43], light field image segmentation [39], image-sentence matching [18], person re-identification [23] and instance segmentation [64].

Recently, the rapid development of the commodity imaging and display device, has resulted in higher requirements for the producing and editing of high-resolution (e.g., 720p, 1080p and 4K) images. Salient object detection as well as many state-of-the-art computer vision tasks are facing various challenges when encountering high-resolution scenarios. A good high-resolution salient object detection method should not only accurately detect the whole salient object but also predict the precise boundaries of salient objects. Despite the conventional Deep Neural Networks (DNNS) based SOD models have achieved remarkable performance at low-resolution (e.g., typical size $224 \times 224$, $384 \times 384$), they often fail to generate high quality detection results for high-resolution images. The major reason for this drawback is that the most previous methods try to identify the salient regions and estimate the accurate objects boundaries simultaneously in one step, which are two difficult and inherently different problems for high-resolution salient object detection. To address the first problem, a network is required to capture sufficient semantics by maintaining a larger receptive field. However, since the memory usage increases dramatically along with the image resolution, it is impractical for these models to directly learn sufficient semantics for high-resolution images. One plausible way is introducing downsample operations, but the structure details are inevitably lost during the downsampling, which however is precisely the key to solving the second problem.

Unfortunately, most of the existing low-resolution SOD
methods [24, 57, 4, 47] try to address the aforementioned two problems with a single regression framework, which ignore the inherent difference between the two problems and result in blurry boundaries. As shown in Fig.1, if we take a deeper look at the saliency map generated by the representative existing methods LDF [47] and GateNet [62], we can observe that pixels can be divided into three different sets: (1) most pixels inside the salient object have the highest saliency value, and we call these pixels as definite salient pixels; (2) most pixels in the background regions have the lowest saliency value, which belong to definite background pixels; (3) saliency values of the pixels at blurry object boundaries fluctuate between 0 and 1, so we call these pixels as uncertain pixels. An ideal SOD method should effectively identify the definite salient and background regions in the image and accurately calculate the saliency value of pixels in the uncertain region to preserve a clear object boundary. From this perspective, there are essentially two tasks in SOD which demand quite different abilities to address the aforementioned two problems. The former task can be viewed as a classic classification task, while the later one is a typical regression task.

Despite the demand for effective high-resolution SOD methods, this line of work is rarely studied. In this paper, motivated by the new observation that SOD should be disentangled into two tasks, we propose a novel deep learning framework for high-resolution salient object detection. Specifically, we decouple the high-resolution salient object detection into a low-resolution saliency classification network (LRSCN) and a high-resolution refinement network (HRRN). LRSCN is designed to capture sufficient semantics at low-resolution and classify the pixels into three different sets for later process. HRRN aims at accurately refining the saliency value of pixels in the uncertain region to preserve a clear object boundary at high-resolution with limited GPU memory. As discussed above, HRRN requires structure details in high-resolution image. However, widely used low-resolution saliency datasets generally have some problems in annotation quality [52], making it almost impossible to directly obtain enough object boundary details from these defective datasets to train the high-resolution network. In the very recent work, Zeng et al. [52] proposed to train their SOD network by using high-resolution images with accurate annotation. However, such high-quality image annotation requires heavy labor costs. In our paper, we argue that it is unnecessary to use such accurately annotated high-resolution images in network training. By introducing uncertainty [19] in the training process, our HRRN can well address the high-resolution refinement task only using the low-resolution training datasets with poor annotation.

Our major contributions can be summarized as:

- We provide a new perspective that high-resolution salient object detection should be disentangled into two tasks, and demonstrate that the disentanglement of the two tasks is essential for improving the performance of DNN based SOD models.
- Motivated by the principle of disentanglement, we propose a novel framework for high-resolution salient object detection, which uses LRSCN to capture sufficient semantics at low-resolution and HRRN for accurate boundary refinement at high-resolution.
- We make the earliest efforts to introduce the uncertainty into SOD network training, which empowers HRRN to well address the high-resolution refinement task without any high-resolution training datasets.
- We perform extensive experiments to demonstrate the proposed method refreshes the SOTA performance on high-resolution saliency datasets as well as some widely used saliency benchmarks by a large margin.

2. Related Work

Over the past decades, a large amount of SOD algorithms have been developed. Traditional models [17, 6, 36, 40, 20] detect salient objects by utilizing various heuristic saliency priors with hand-crafted features. More details about the traditional methods can be found in the survey [1]. Recently, with the development of deep learning, the performance of saliency detection has archived great improvement [24, 9, 26, 34, 65, 38, 59]. Here we mainly focus on deep learning based saliency detection models.

Recently, some DNN-based models use various feature enhancement strategies to improve the ability of localization and awareness of salient objects [16, 48, 35, 5, 46, 28, 62, 12], or take advantage of edge features to restore the structural details of salient objects [44, 60, 49, 63]. For example, Pang et al. [28] applied the transformation-interaction-fusion strategy on multi-level and multi-scale features to learn discriminant feature representation. Zhao et al. [62] designed a gated dual branch structure to build the cooperation among different levels of features and improve the discriminability of the whole network. In [60],
edge features from edge feature branch was fused with salient features as complementary information to enhance the structural details for accurate saliency detection. Zhou et al. [63] used two individual branches for representing saliency and contour stream respectively, and a novel feature fusion module for their correlation combination.

Different from the above methods, some methods consider leveraging predict-refine architecture to generate fine salient objects. For example, Wang et al. [41] proposed to localize salient objects globally and then refine them by a local boundary refinement module. Qin et al. [31] was composed of an Encoder-Decoder network and a residual refinement module, which were respectively in charge of saliency prediction and saliency map refinement.

However, all these methods cannot handle high-resolution salient object detection problem well since such simple regression framework cannot identify the salient regions and estimate the accurate objects boundaries simultaneously and their architectures are not optimized for high-resolution SOD. Zeng et al. [52] tried to alleviate this problem by leveraging both global semantic information and local high-resolution details to accurately detect salient objects in high-resolution images. However, Zeng et al. [52] relies on high-resolution training images with accurate annotation, which requires heavy labor costs. Different from the above methods, we disentangle high-resolution SOD into two tasks at different resolutions: identifying the salient regions at low-resolution and estimating the accurate objects boundaries at high-resolution. Moreover, unlike Zeng et al. [52], we introduce novel uncertainty loss, which empowers our HRRN to well address the high-resolution refinement task without using any high-resolution training datasets. Recently, Wei et al. [47] and Zhang et al. [55] also leverage disentanglement in their SOD methods. However, they still try to address the SOD task under a single regression framework but with decoupled supervisions. Unlike our proposed methods, their disentanglement frameworks barely touch the very nature of the SOD, which essentially contains two different tasks. For more information about the DNN-based methods, please refer to survey [42, 14].

3. Proposed Method

In this section, we first describe the overall architecture of the proposed disentangled high quality salient object detection network, then elaborate our main contributions, which are corresponding to LRSCN and HRRN.

3.1. Network Overview

The architecture of the proposed approach is illustrated in Fig.2. As can be seen, the disentanglement includes two decoupled tasks at two different resolutions. LRSCN aims at capturing sufficient semantics at low-resolution and classifying the pixels into three different sets, which also can save the memory usage. While estimating the accurate objects boundaries needs more local details at high-resolution. So, we design HRRN to regress the saliency value of pixels and preserve a clear object boundary at high-resolution.

LRSCN has a simple U-Net like Encoder-Decoder architecture [32]. VGG-16 [33] is used as backbone. Following [16, 60], we connect another side path to the last pooling layer in VGG-16. Hence, we obtain six side features Conv1-2, Conv2-2, Conv3-3, Conv4-3, Conv5-3 and Conv6-3 from backbone network. Because Conv1-2 and Conv2-2 are too close to the input and their receptive fields are too small, following [60, 48], we only use the last four levels features for the following process. Conv6-3 is denoted as $\{F_i|h = 6\}$, the other three levels features are denoted as $\{F_i|i = 3, 4, 5\}$. Multi-scale feature extraction and Cross-level feature fusion (MECF) module is added between encoder and decoder to help improve the discriminability of feature representations. Decoder focuses the output features from MECF and the upsampled features from the previous stage in a bottom-up manner. The output of each decoder is defined as $\{D_i|i = 3, 4, 5, 6\}$. Finally, SGA module is built upon $D_3$ for accurate trimap $T$ generation.

As described, LRSCN is classification task and aims at
capturing sufficient semantics at low-resolution. To regress a clear object boundary value, the input of HRRN is a high-resolution image under guidance of the trimap provided by LRSCN. HRRN has a basic Encoder-Decoder architecture and with the help of uncertainty loss, the network can be more robust to noisy data and predict a high-resolution saliency map with clear boundary.

3.2. Architecture of LRSCN

To capture sufficient semantics at low-resolution, learning discriminant feature representations is essential. The network should not only consider scale and location variations of different salient objects, but also distinguish the appearance difference between the salient object and the non-salient regions. To achieve the first goal, we develop a multi-scale feature extraction module (ME) based on Global Convolutional Network (GCN) [29] to enlarge the feature receptive field and obtain multi-scale information. To achieve the second goal, we utilize cross-level feature fusion module (CF) to leverage the advantages of features at different levels. Moreover, in designing the network architecture, inspired by [50], we use split-transform-merge strategy to further enlarge feature receptive fields and hence results in more discriminative feature representations. Specifically, we uniformly split the input $F$ into two portions $\{F^1, F^2\}$ by channel dimension, then $F^1$ is sent into multi-scale feature extraction pathway and $F^2$ is sent into cross-level feature fusion pathway. The outputs of these two pathways are concatenated together as the final output. We call this bridge module as MECF module, which is shown in Fig.3. More details about MECF module can be found in section 6 of supplementary materials.

SGA Module. As illustrated in Fig.2, each decoder fuses features from MECF module and previous decoder stage, then uses $3 \times 3$ convolutional layer for final prediction. To maintain consistency between trimap and saliency map and ensure the uncertain regions of the trimap can accurately cover the boundary of saliency map, we design a saliency guide attention module (SGA) on $D_3$. Specifically, we first use a $3 \times 3$ convolution and sigmoid function to compute a saliency map. Then, the saliency map is treated as spatial weight map which can help refine feature and generate an accurate trimap. Finally, the output trimap $T$ is 3-channel classification logits. The whole SGA module guarantees the alignment of trimap and saliency map.

3.3. Architecture of HRRN

Following the principle of disentanglement, HRRN aims at accurately refining the saliency value of pixels in the uncertain region to preserve a clear object boundary at high-resolution under the guidance of the trimap provided by LRSCN. The architecture of HRRN is shown in Fig.2. HRRN has a simple U-NET like architecture. For better prediction at high-resolution, we then do some non-trivial modifications. First, lower-level features contain rich spatial and detail information which play a crucial role in restoring a clear object boundary, so decoder combines encoder features before each upsampling block instead of after each upsampling block. Moreover, we use a two layers short cut block to align channels of encoder features for feature fusion. Second, to let network pay more attention to detail information, we directly feed the original input to the last convolutional layer through a short cut block to generate better results. Finally, learning from image generation tasks [3, 53], we use the spectral normalization [27] to each convolutional layer to add a constraint on Lipschitz constant of the network and stable the training.

3.4. Loss Function of LRSCN

To supervise LRSCN, we should generate trimap groundtruth $T^{gt}$, which can represent the definite salient, definite background and uncertain regions. As described, uncertain regions exist mainly at the boundaries of the objects. So we erase and dilate binary groundtruth maps at the object boundaries with a random pixel number $(5,7,9,11,13)$ to generate the GT uncertain regions. The remaining foreground and background regions represent definite salient and background regions. $T^{gt}$ is defined as:

$$T^{gt}(x, y) = \begin{cases} 
2, & T^{gt}(x, y) \in \text{definite salient} \\
0, & T^{gt}(x, y) \in \text{definite background} \\
1, & T^{gt}(x, y) \in \text{uncertain region} 
\end{cases}$$

(1)
where \((x, y)\) stands for each pixel location on the image. Some examples can be seen in Fig.4.

For trimap supervision, we use Softmax cross-entropy loss, which is defined as:

\[
L_{trimap} = \frac{1}{N} \sum_{i} -\log(\frac{e^{T_i}}{\sum_{j} e^{T_j}}). \tag{2}
\]

To guarantee the accuracy of trimap, we add extra saliency supervision \(L_{saliency}\) as the supplement of trimap supervision. Similar to BASNet [31], we use pixel-level, region-level and object-level supervision strategy on multi-levels to better keep the uniformity and wholeness of the salient objects. Specifically, binary cross-entropy (BCE) [7], SSIM [45] and F-measure loss [61] are denoted as pixel-level, region-level and object-level loss. Note that all parts of LRSCN are trained jointly, so the overall loss function is given as:

\[
L_{LRSCN} = L_{saliency} + L_{trimap}. \tag{3}
\]

We do not use uncertainty loss because the main goal of LRSCN is to capture sufficient semantics, not accurate boundary. More details about \(L_{saliency}\) can be found in section 5 of supplementary materials.

### 3.5. Loss Function of HRRN

We perform a \(L_1\) loss and novel uncertainty loss to restore the fine structures and boundaries of salient objects. For an input high-resolution image \(I\), let \(G^H\) denote its groundtruth, and predicted saliency map is \(S^H\).

We leverage the \(L_1\) loss to compare an absolute difference between predicted saliency map and groundtruth over the definite salient and background regions:

\[
L_1 = \frac{1}{E} \sum_{i \in E} |S^H_i - G^H_i|, \tag{4}
\]

where \(E\) indicates the number of pixels which are labeled as definite salient or background in the trimap, \(S^H_i\) and \(G^H_i\) denote the predicted and groundtruth value at position \(i\).

We cannot directly compute \(L_1\) loss between predicted saliency map and groundtruth over the uncertain regions because widely used saliency training datasets have some problems in annotation quality [52]. We show these low quality annotations in section 4 of supplementary materials. It is almost impossible to directly obtain enough object boundary details from these defective datasets to train the high-resolution network. To address this problem, we design uncertainty loss, which empowers our HRRN to well address the high-resolution refinement task only using these defective low-resolution training datasets. It is worth noting that there are some previous works [54, 58] involving “uncertainty” in their titles, which seem relevant to our method.

To show how proposed uncertainty loss makes the network attenuate the effect from erroneous labels during training, we visualize the impact of \(L_1\) loss and uncertainty loss.

![Figure 5. The impact of the losses. Best viewed by zooming in.](image-url)

However, in [54], “uncertainty” means the human perceptual uncertainty modeled by CVAE. While in [58], “uncertainty” indicates the saliency prediction system uncertainty modeled by R-dropout. Obviously, their usages of uncertainty are different from ours.

Inspired by [19], Gaussian likelihood is used to model the uncertainty. Let \(x\) and \(f(x)\) be the input and output of HRRN, and Gaussian likelihood is defined as:

\[
p(y|f(x)) = \mathcal{N}(f(x), \sigma^2), \tag{5}
\]

where \(\sigma\) measures uncertainty of the estimation, \(y\) is the label of output. In maximum likelihood inference, we maximize the log likelihood of the model, which is written as:

\[
\log p(y|f(x)) \propto -\frac{||y - f(x)||^2}{2\sigma^2} - \frac{1}{2} \log \sigma^2, \tag{6}
\]

so the proposed uncertainty loss is defined as:

\[
L_{uncertainty} = \frac{||y - f(x)||^2}{2\sigma^2} + \frac{1}{2} \log \sigma^2. \tag{7}
\]

We only care about the pixels in uncertain regions, so \(L_{uncertainty}\) is written as:

\[
L_{uncertainty} = \frac{1}{U} \sum_{i \in U} \frac{||S^H_i - G^H_i||^2}{2\sigma_i^2} + \frac{1}{2} \log \sigma_i^2, \tag{8}
\]

where \(U\) is the total number of pixels in uncertain region, \(\sigma_i\) is the uncertainty of each pixel and is generated from HRRN. Different from directly learning from noisy data, uncertainty loss can allow the network to learn how to attenuate the effect from erroneous labels. Specifically, pixels for which the network learned to predict high uncertainty will have a smaller value of the first term of Eq.8, so have little effect on the loss. Meanwhile, large uncertainty increases the contribution of the second term of Eq.8, and in turn penalizes the model and lets the model make a better prediction that has low uncertainty. Note that all parts of HRRN are trained jointly, so the overall loss function is given as:

\[
L_{HRRN} = L_{uncertainty} + L_1. \tag{9}
\]
at the same training iteration in Fig 5. We show the impact of both these losses in the same image. The images shown in Fig 5 are these which have problems in annotation quality. Compared with column 4 and column 5, if we only use $L_1$ loss, the weight of the loss in the uncertainty region will be large, which leads the network hard to converge. While uncertainty loss will make the weight of the loss in the uncertainty region be small and let the network ignore effects from noisy data as much as possible. Column 6 shows the uncertainty value of pixels in uncertain regions. It can be seen that pixels in uncertain regions usually have higher uncertainty value. In general, compared to $L_1$ loss, uncertainty loss reduces the weight of the loss in the uncertainty region, thus mitigating the impact of noisy data on the network. But due to the uncertainty value, it will allow the network to learn how to predict a better prediction with a low uncertainty value, instead of ignoring the learning of the uncertainty region completely. These visual comparisons show how uncertainty loss makes network more robust to noisy data.

4. Experiment
4.1. Experimental Settings

Implementation Details. Following the works [31, 60, 28, 47], we train our proposed network on DUTS-TR. We use Pytorch\(^1\) to implement our model. A GTX 1080Ti GPU is used for acceleration. VGG-16 [33] is used as the backbone network of LRSCN, and the whole network is trained end-to-end by stochastic gradient descent (SGD). For a more comprehensive demonstration, we also trained our network with ResNet-50 [15] backbone. Maximum learning rate is set to 0.001 for backbone and 0.01 for other parts. Warm-up and linear decay strategies are used to adjust the learning rate. Momentum and weight decay are set to 0.9 and 0.0005 respectively. Batchsize is set to 32 and maximum epoch is set to 100. Horizontal flip and multi-scale input images are utilized for data augmentation as done in [31, 60, 46]. During testing, the input of LRSCN is about 352 × 352 resolution.

The learning rate of HRRN is initialized to 0.0005. Warmup and cosine decay are applied to the learning rate. The network HRRN is trained for 10000 iterations with a batch size of 20. During training, the resolution of input image and trimap is 512 × 512. During testing, we first resize the image and trimap to 1024 × 1024, then we split the image and trimap into four sub-images and sub-trimaps with 512 × 512 resolution, as shown in Fig 2. Finally, we send each sub-image and sub-trimap together to HRRN to generate sub-prediction result, and use 4 sub-predictions stitched together to make one high-resolution saliency result.

Evaluation Datasets. Following work [52], we evaluate our method on two high-resolution saliency detection datasets, including HRSOD-TE and DAVIS-S, which contain 400 and 92 images. DAVIS-S dataset is collected from DAVIS [30]. Images in these two datasets are precisely annotated and have very high resolutions (i.e., 1920 × 1080). We also evaluate our method on three low-resolution datasets, including DUT-OMRON [51], DUTS-TE [37] and HKU-IS [22], which contain 5168, 5019 and 4447 images. Our results are available at https://luckybird1994/HQSOD.

Evaluation Metrics. Six metrics are used to evaluate the performance of our method. The first is Mean Absolute Error (MAE), which characterize the average 1-norm distance between ground truth maps and predictions. The second is F-measure ($F_{\beta}$ and $F_{\beta}^{\max}$), a weighted mean of average precision and average recall, calculated by $F_{\beta} = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}$. We set $\beta^2$ to be 0.3 as suggested in [2]. The third is Structure Measure ($S_m$), a metric to evaluate the spatial structure similarities of saliency maps based on both region-aware structural similarity $S_r$ and object-aware structural similarity $S_o$, defined as $S_m = \alpha \times S_r + (1-\alpha) \times S_o$, where $\alpha = 0.5$ [10]. In addition, precision-recall (PR) curve is used to show the whole performance. To further evaluate the boundary quality, Following [52] and [56], we use Boundary Displacement Error (BDE) [11] and $B_o$ metrics. More details about BDE and $B_o$ can be found in section 7 of supplementary materials. The last two metrics are only used in two high-resolution datasets, because their boundaries annotation is accurate and evaluating results are reliable.

4.2. Comparisons with the State-of-the-Arts

We compare our approach with 16 SOTA methods, including Amulet [57], R3Net [8], DGRL [41], DSS [16], BASNet [31], CPD [48], EGNet [60], PFPN [35], GCPA [5], F3N [46], MNNet [28], ITSD [63], LDF [47], GateNet [62], CSF [12] and HRNet [52]. For a fair comparison, we use either the implementations with recommended parameter settings or the saliency maps provided by the authors. The evaluation toolbox used in this paper is same as F3N [46].

Quantitative Evaluation. From Table 1, when we train our network only using DUTS (Ours), our method can already improve the $F_{\beta}^{\max}, F_{\beta}, S_m$ and MAE achieved by the best-performing existing algorithms, especially two high-resolution test datasets. It is worth noting that for boundary accuracy, our method is far better than other methods on two high-resolution. These results demonstrate the efficiency of the proposed disentangled SOD framework in both identifying the salient regions and estimating the accurate objects boundaries. Other than numerical results, we also show the PR curves on two high-resolution datasets

\(^1\)https://pytorch.org/
Table 1. Quantitative comparison with SOTA on two high-resolution and three low-resolution datasets. The best three results are in red, green and blue fonts. "\textsuperscript{+}" means the results are post-processed by dense conditional random field(CRF) [21]. "\textsuperscript{*}" means using ResNeXt-101 [50] backbone. "\textsuperscript{a}" means using ResNet-101 backbone. "\textsuperscript{b}" means using Res2Net50 [13] backbone. MK: MSRA10K [6], DUTS: DUTS-TR [37], MB: MSRA-B [25], HR: HRSOD-Training [52], HR-L: HRSOD-Training resized in low-resolution. Smaller MAE, BDE and $B_m$ larger $F_p$ \textsuperscript{+}, $F_p$ and $S_m$ correspond to better performance.

![Figure 6. Comparison of PR curves across two high-resolution and three low-resolution datasets.](image)

![Figure 7. Visual comparison between our method and other SOTA methods. Each sample occupies two rows. Best viewed by zooming in.](image)

and three low-resolution datasets in Fig.6. As can be seen, the PR curves by our method (red ones) are especially outstanding compared to all other previous methods. Besides, shorter PR curves imply that our saliency maps are usually more assertive with sharper boundaries than the results of other methods. An interesting observation is that when we add HRSOD-training datasets (resized to low-resolution like $352 \times 352$) in LRSCN, the performance in two high-resolution datasets HRSOD-TE and DAVIS-S can be further improved. However, this practice seems to be of little help in improving the performance in other three low-resolution datasets. A similar phenomenon can also be found in the performance of HRNet [52]. We think there could be some image selection or data annotation biases between the high-resolution datasets and the low-resolution datasets, which cause this phenomenon.

**Qualitative Evaluation.** To exhibit the superiority of the proposed approach, Fig.7 shows representative examples of saliency maps generated by our approach and other state-of-the-art algorithms. As can be seen, with the help of LRSCN, our method can not only keep the wholeness of the salient object (row 3), but also accurately locate salient objects and suppress non-salient regions (row 5), compared to other methods. HRRN can help the model to restore accurate and complete boundaries of salient objects, which are more consistent with the GT boundaries. It can be clearly
observed that our method achieves impressive performance in all these cases, which indicates the effectiveness of disentangled framework and uncertainty loss. More comparison experiments can be found in section 2 of supplementary materials.

4.3. Ablation Studies

To validate the effectiveness of the proposed components of our method, we conduct a series of experiments on two high-resolution datasets with different settings under VGG-16 backbone. Specifically, we first verify the effectiveness of MECF and SGA module in LRSCN. Then we validate the effectiveness of uncertainty loss in HRRN and the superiority of the proposed disentangled architecture.

Ablation Studies of LRSCN. To prove the effectiveness of MECF and SGA module, we report the quantitative comparison results of LRSCN with different architectures in Table 2. Baseline denotes that we conduct a experiment over $L_{\text{LRSN}}$ with a pure U-Net architecture. We can see that only using ME or CF can already heavily improve the performance. A better performance has been achieved through the combination of these two architectures. Finally, performance can be further improved by SGA module, especially BDE and $B_s$, which means that SGA can help generate accurate trimap. While $L_{\text{sal}}$ is not our core innovation, more ablation studies about $L_{\text{sal}}$ can be found in section 5 of supplementary materials.

Ablation Studies of HRRN. In HRRN, uncertainty loss play a key role to estimate the accurate objects boundaries, so we first investigate the effectiveness of our proposed uncertainty loss. From Table 3, we can see that without uncertainty loss (Ours($L_1$)), the performance decreased a lot. Besides, when we add high-resolution HRSOD-Training datasets in HRRN (Ours-DH($L_1 + L_{\text{uncertainty}}$)), the performance has no obvious improvement, which demonstrates that our network is not reliant on accurately annotated high-resolution images during training. To further demonstrate the effectiveness of our disentangled framework, we compare our HRRN with CRF [21], a widely used post-processing for saliency detection. Results in Table 3 show that our proposed method (Ours (LRSCN + HRRN)) outperforms CRF (Ours (LRSCN + CRF)) by a large margin. The same phenomenon can be found in the refinement of EGNet, CD and BASNet (Their trimaps are generated with the corresponding saliency maps following Eq.1). Moreover, compared to RRM module proposed in BASNet, our HRRN can better improve performance. This ablation study demonstrates the superiority of HRRN within our novel disentangled framework. More analyses of the proposed disentangled framework can be found in section 3 of supplementary materials.

Analysis of Uncertainty Loss To further demonstrate that uncertainty loss can make network more robust to noisy data, we erase and dilate binary groundtruth maps of DUT-16 dataset at the object boundaries with a random pixel number (3,4,5,6,7,9,11,13) to generate noisy training data. Then we train the network on these noisy data, BDE and $B_s$ results on HRSOD-TE are reported in Fig.8(a) and Fig.8(b). When the erosion or dilation kernel ranges from 3 to 7, the network trained with uncertainty loss has a rather stable performance. With the increase in erosion or dilation kernels, even though the performance is dropped, training with uncertainty loss still yields a better model than training without uncertainty loss. This experiment further validates the effectiveness of uncertainty loss.

5. Conclusions

In this work, we argue that there are two difficult and inherently different problems in high-resolution SOD. From this perspective, we propose a novel deep learning framework to disentangle the high-resolution SOD into two tasks: LRSCN and HRRN. LRSCN can identify the definite salient, background and uncertain regions at low-resolution with sufficient semantics. While HRRN can accurately refining the saliency value of pixels in the uncertain region to preserve a clear object boundary at high-resolution with limited GPU memory. We also make the earliest efforts to introduce the uncertainty into SOD network training, which empower HRRN to learn rich details without using any high-resolution training datasets. Extensive evaluations on high-resolution datasets and popular benchmark datasets not only verify the superiority of our method but also demonstrate the importance of disentanglement for SOD. We believe our novel disentanglement view in this work can contribute to other high-resolution computer vision tasks in the future.
References


