Detect Globally, Refine Locally: A Novel Approach to Saliency Detection

Tiantian Wang[1], Lihe Zhang[1], Shuo Wang[1], Huchuan Lu[1], Gang Yang[2], Xiang Ruan[3], Ali Borji[4]
[1] Dalian University of Technology,[2] Northeastern University, China
[3] Tiwaki Co., Ltd [4] University of Central Florida, USA
Tiantianwang.ice@gmail.com, zhanglihe@dlut.edu.cn
lhchuan@dlut.edu.cn, aliborji@gmail.com

Abstract

Effective integration of contextual information is crucial for salient object detection. To achieve this, most existing methods based on 'skip' architecture mainly focus on how to integrate hierarchical features of Convolutional Neural Networks (CNNs). They simply apply concatenation or element-wise operation to incorporate high-level semantic cues and low-level detailed information. However, this can degrade the quality of predictions because cluttered and noisy information can also be passed through. To address this problem, we proposes a global Recurrent Localization Network (RLN) which exploits contextual information by the weighted response map in order to localize salient objects more accurately. Particularly, a recurrent module is employed to progressively refine the inner structure of the CNN over multiple time steps. Moreover, to effectively recover object boundaries, we propose a local Boundary Refinement Network (BRN) to adaptively learn the local contextual information for each spatial position. The learned propagation coefficients can be used to optimally capture relations between each pixel and its neighbors. Experiments on five challenging datasets show that our approach performs favorably against all existing methods in terms of the popular evaluation metrics.

1. Introduction

Visual saliency has gained a lot of interest in recent years. It has been shown effective in a wide range of applications including person identification [2], visual tracking [9], image captioning [7, 8], robot navigation [6] and visual question answering [21]. When it comes to the image-based salient object detection, two major problems need to be tackled: how to highlight salient objects against the cluttered background and how to preserve the boundaries of salient objects. However, in view of the fact that salient objects may share some similar visual attributes with the background distractors and sometimes multiple salient objects overlap partly or entirely with each other, saliency detection still remains challenging in computer vision tasks. The recent CNNs-based approaches [18, 22, 10, 33, 29] have been successful in mitigating the above issues, and have given rise to the proliferation of a significant variety of neural network structures. Usually, standard convolutional neural networks are composed of a cascade of repeated convolutional stages, followed by the spatial pooling. The deeper layers are encoded with richer semantic representation albeit at the expense of spatial resolution, while the shallower layers contain much finer structures. Existing saliency detection methods [18, 10, 33] attempt to combine hierarchical features to capture distinctive objectness and detailed information simultaneously. However, these approaches usually concentrate their analysis on how to combine features effectively in general. What is often overlooked is that directly applying concatenation or element-wise operation to different feature maps are suboptimal because some maps are too cluttered which can introduce misleading information when detecting and segmenting salient objects. The problem is illustrated in Figure 1.

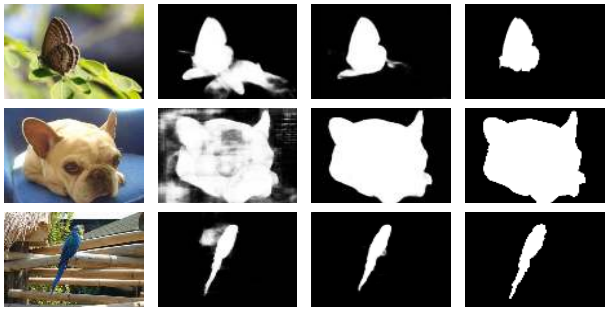


Figure 1. Comparison with the feature integration based method. (a) Input images. (b) Amulet [33]. (c) Our method. (d) Ground truth masks.

Therefore, from a global perspective, we propose a novel Recurrent Localization Network (RLN) which consists of two modules: an inception-like Contextual Weighting Module (CWM) and a Recurrent Module (RM). CWM aims to

predict a spatial response map to adaptively weight the features maps for each position, which can localize the most attentive parts for every given input. Specifically, CWM lies on top of the side output results of each convolutional block, which takes the output feature maps as input and learns a weight for each pixel based on the multi-scale contextual information. The weights are then employed to each feature map for producing a weighting spatial representation. CWM serves to filter out the distractive and cluttered background and make salient objects stand out. Moreover, a recurrent structure is proposed in order to gradually refine the predicted saliency map over 'time'. It establishes recurrent connections to propagate the outputs of certain blocks to its input so as to exploit the context cue in the training process of different layers.

Second, from a local perspective, we adopt a Boundary Refinement Network (BRN) to recover the detailed boundary information. The BRN takes both the initial RGB image and the saliency map as input. The saliency map serves as the prior map which can assist the learning process to generate more accurate predictions. BRN can predict a $n \times n$ propagation coefficient map for each pixel which indicates the relations between the center point and its $n \times n$ neighbors. For each pixel, the corresponding coefficients are position-aware and can adaptively learn the local contextual information for the $n \times n$ neighbors.

To summarize, our contributions are as follows:

- We propose a novel Localization-to-Refinement network where the former recurrently focuses on the spatial distribution of various scenarios to help better localize salient objects and the latter helps refine the saliency map by the relations between each pixel and their neighbors.
- In the Recurrent Localization Network, a contextual module is adopted for weighting features maps at each position. Also, a recurrent mechanism is proposed to gather contextual information for refining the convolutional features iteratively. In the Boundary Refinement Network, a refinement module is adopted to learn local context information by the propagation efficient.
- Compared with all state-of-the-art works, the proposed model achieves the best performance on ECSSD, THUR15K, DUT-OMRON, HKU-IS and DUTS benchmark datasets.

2. Related Work

Various approaches have been proposed to solve the problem of saliency detection. Early research [23, 12, 31, 32, 11, 19, 14, 4, 25] focuses on low-level visual features, such as center bias, contrast prior and background prior. Recently, significant progress has been made by deep learning based methods [26, 36, 17, 28, 15, 16, 18, 22, 10, 29, 33, 3], which can be broadly categorized into region-based and Fully Convolutional Network (FCN)-based methods. In the following, we briefly review recent developments on these two categories.

2.1. Region-based Saliency

Region-based approaches leverage each image patch as the basic processing unit for making saliency prediction. In [17], Li *et al.* utilize multi-scale features extracted from a deep CNN via exploiting contextual information. A classifier network is employed to infer the saliency score of each image segment. In [36], Zhao *et al.* propose a multi-context deep learning structure for salient object detection. They attempt to model each superpixel by jointly optimizing both global and local context. In [26], a two-stage training strategy is proposed to combine both image patches and candidate objects. Local features and global cues are incorporated for generating a weighted sum of salient object regions. Lee *et al.* [16] utilize a two-stream framework with high-level feature descriptors extracted from the VGG-net and low-level heuristic features such as color histogram and Gabor responses. A neural network with fully-connected layers is proposed to evaluate the saliency of every region.

2.2. FCN-based Saliency

While region-based deep learning approaches improve the performance over the ones based on hand-crafted features by a large margin, they ignore important spatial information as they assign one saliency label to each image patch. Also, these methods are time-consuming since the whole networks are run for many times to get predictions of all patches in the image. To overcome this problem, one of the most popular CNNs adopted is the Fully Convolutional Network. Several existing works try to improve the saliency detection task mainly based on the following aspects.

Skip Connections. Skip connections aim to add deeper layers to lower ones and integrate saliency prediction at multiple resolutions. In [18], a multiscale FCN is proposed to capture effective semantic features and visual contrast information for saliency inference. Hou *et al.* in [10] introduce short connections by transforming high-level features to shallower side-output layers. The multi-scale feature maps at each layer can assist to locate salient regions and recover detailed structures at the same time. Zhang *et. al.* [33] learn to aggregate multi-level feature maps at each resolution and predict saliency maps in a recursive manner. In [29], Wang *et. al.* propose a stagewise refinement model and a pyramid pooling module to include both local and global context information for saliency prediction. In particular, the stagewise model is utilized to add lower level detailed features to the predicted map stage by stage. The aforementioned works attempt to utilize hierarchical features of CNNs to make prediction. However, messy and

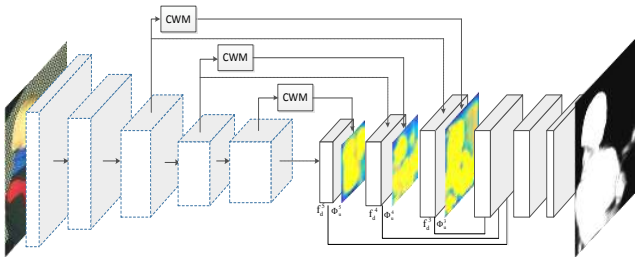


Figure 2. The overall structure of Recurrent Localization Network (RLN). The blue and black dotted lines denote the recurrent blocks and convolutional operation, respectively.

cluttered information are also included when low-level features are combined directly with high-level ones. To deal with it, we propose an inception-like contextual weighting module for purifying the convolutional features.

Recurrent Structure. Recurrent Structure can help reduce prediction errors by iteratively integrating contextual information. Kuen [15] firstly adopt a convolutional-deconvolutional network to produce a coarse saliency map. Then a spatial transformer and recurrent network units are used to iteratively search for the attentive image sub-regions for the saliency refinement. Liu and Han [22] propose an end-to-end method based on the fully convolutional network. A hierarchical recurrent CNN is adopted to progressively recover image details of saliency maps through integrating local context information. In [28], Wang *et. al.* utilize the predicted saliency map as the feedback signal, which serves as the saliency prior to automatically learn to refine the saliency prediction by correcting its previous errors. Different from those works, we propose a block-wise recurrent module which can combine the output and input features of certain convolutional block over multiple time steps thereby incorporating the contextual information.

3. The Proposed Method

In this section, we will elaborate on the proposed framework for saliency detection. We firstly describe the global Recurrent Localization Network (RLN) in Section 3.1, and then give a detailed depiction of the local Boundary Refinement Network (BRN) in Section 3.2. The overall architecture of the proposed network is illustrated in Figure 2.

3.1. Recurrent Localization Network

3.1.1 Base Network

We tackle the saliency detection problem based on the fully convolutional network. Our proposed method is based on the ResNet-50 network [24]. Specifically, we remove the original global average pooling, fully connected and softmax loss layers and retain the bottom convolutional blocks in ResNet-50 network. The base network is composed of repetitive residual building blocks with different output dimensions. For an input image I, the base network generates 5 feature maps $(\mathbf{f}^1, \mathbf{f}^2, ..., \mathbf{f}^5)$ with decreasing spatial resolution by stride 2. Each map is produced by one residual convolution block. The feature map $\mathbf{f}^5$ obtained from $Conv5$ has the smallest spatial dimension while $\mathbf{f}^1$ has the largest one. For efficient computation, we obtain the k-th feature map $\mathbf{f}_d^k (k \in \{3, 4, 5\})$ by applying a 3×3 convolutional layer with 128 channels behind the output feature map $\mathbf{f}^k$ of the k-th residual block to reduce the dimension. We upsample the feature maps $\mathbf{f}_d^k (k \in \{4, 5\})$ to the same size as $\mathbf{f}_d^3$. Then an element-wise multiplication layer is applied to all feature maps $\mathbf{f}_d^k$ followed by one 1×1 convolutional layer with 128 channels and one 1×1 convolutional layer with 2 channels to produce a prediction map $\mathbf{S}$. We set the number of output channels in the prediction map equal to the number of possible labels. Each channel of $\mathbf{S}$ corresponds to a confidence measure used in predicting each spatial position as one of the two classes. Finally, we directly upsample $\mathbf{S}$ using bilinear interpolation to match the input image size.

3.1.2 Network Architecture

Most of the existing saliency detection methods typically involve a combination of multi-scale convolutional features, which is driven by the notion that different layers of CNNs usually carry rich representation varying from low-level visual characteristics to high-level discriminative information. However, as mentioned earlier, there exist limitations among the integrated features if certain "bad" features are adopted because simple incorporation of convolutional features can make the noise in "bad" feature maps unrestrainedly pass to the prediction layer.

Motivated by the above observation, we propose a contextual weighting mechanism based on the inception architecture to modulate the features being passed. In particular, a recurrent structure is adopted for learning context-aware features, which can connect the output of each block to the input of the same block in a feedback fashion.

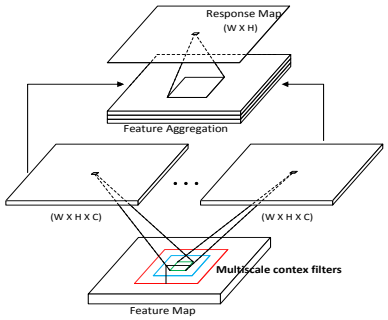


Figure 3. Details of Inception-like Module.

Inception-like Contextual Weighting Module. Our module is inspired by the success of contextual reweighting network [13] in image geo-localization. In order to obtain

the spatial response map for each position, we first connect a downsampling layer behind the feature map $\mathbf{f}^k$ which is generated by the k-th residual block. Then a convolutional layer with kernel size m is applied for sliding a $m \times m$ spatial window on the local feature, which is shown in Figure 3. Thus the context information can be included in the hidden context filter.

To obtain multi-scale contextual information, we adopt an inception-like module by using three context filters with different kernel sizes (3×3, 5×5, 7×7). Each filter produces an activation map with the size $W \times H \times C$, followed by a L_2 normalization layer. Then we concatenate these activation maps to form features $\mathbf{f}^k_{cat}$.

To compute the contextual weighting response map $\mathbf{M}^k$, we utilize a convolutional layer with one output channel behind $\mathbf{f}^k_{cat}$, which is formulated as

$$\mathbf{M}^k = \mathbf{W} * \mathbf{f}^k_{cat} + \mathbf{b}, \tag{1}$$

where $\mathbf{W}$ represents the kernel and $\mathbf{b}$ denotes the bias parameter. The resulting weighting response map is of size $W \times H$ where each value in this map determines the importance of each spatial position.

Then the Softmax operation is applied to $\mathbf{M}^k$ spatially to get the final weighting response map,

$$\mathbf{\Phi}^k(x, y) = \frac{\exp(\mathbf{M}^k(x, y))}{\sum_{(x',y')} \exp(\mathbf{M}^k(x', y'))}, \tag{2}$$

where $\mathbf{\Phi}^k(x, y)$ represents the normalized response value at (x, y) and k is the index of the residual block. Intuitively, if pixel i is salient at position (x, y), the pixel in the response map related to it should be assigned a higher value. Finally, the weighting map is upsampled to get $\mathbf{\Phi}^k_u$ and applied to the feature $\mathbf{f}^k_d$,

$$\mathbf{F}^k(c) = \mathbf{\Phi_u}^k \circ \mathbf{f}^k_d(c), \tag{3}$$

where c denotes the c-th feature channel. We use $\circ$ to represent the element-wise product operation. Note that $\mathbf{\Phi}^k_u$ is shared across all the channels of $\mathbf{f}^k_d$.

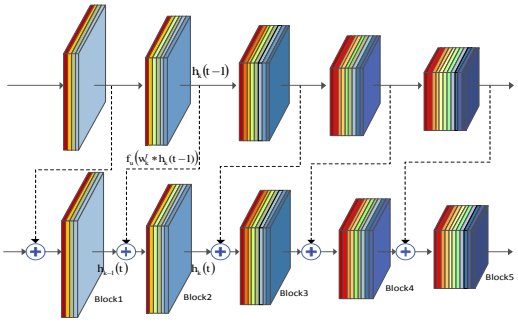


Figure 4. Illustration of the Recurrent Module (RM). The dotted lines represent the convolution and upsample operations. The symbol $\oplus$ denotes element-wise addition.

Recurrent Module. Contextual information [22, 36, 29] has been proved effective in saliency detection. Larger context usually captures global spatial relations among objects while smaller context focuses on the local appearance, both contributing to the saliency detection. In this paper, we propose a novel recurrent module which offers the advantage that increasing time steps enable the whole network to integrate contextual knowledge in a larger neighborhood as time evolves and serves as a refinement mechanism by combining the semantic cues and detailed information in the inner blocks of Resnet-50.We treat each block in ResNet-50 as the basic recurrent unit, which shares the same parameters of weight layers in our structure over time. The state of the current block is determined by the current feed-forward input and the previous state of the same block. Specifically, the state of block $\mathbf{h}_k$ at time step t is calculated by taking the output feature maps from the previous prediction $\mathbf{h}_k(t-1)$ at time step $t-1$ of the same block and the current output $\mathbf{h}_{k-1}(t)$ at time step t of its previous block $k-1$ as the input,

$$\mathbf{h}_k(t) = \begin{cases} f_k(\mathbf{w}^f_k * \mathbf{h}_{k-1}(t) + \mathbf{b}_k), & t = 0 \\ f_k(\mathbf{w}^f_k * (\mathbf{h}_{k-1}(t) + \mathbf{f}_u(\mathbf{w}^r_k * \mathbf{h}_k(t-1))) + \mathbf{b}_k), & t > 0 \end{cases} \tag{4}$$

where the symbol $*$ denotes the convolution operation. $f_k(\cdot)$ is a composite of multiple specific functions including the BatchNorm and ReLU activation function. $f_u(\cdot)$ denotes the upsampling operation. $\mathbf{w}^f_k$ and $\mathbf{w}^r_k$ are feed-forward and recurrent weights for block k. $\mathbf{b}_k$ represents the bias for block k. Note that $\mathbf{w}^f_k$ is shared by the same block, which is used multiple times at each block to reduce memory consumption. $\mathbf{w}^r_k$ is learned independently across the same block at different time steps in order to learn specific transformations for incorporating context information from the current block at time step $t-1$.

Figure 4 illustrates the overall recurrent structure in the process of forward- and backward- propagation following depth and time dimensions (here we set $t = 1$). There are several advantages with the proposed recurrent structure. First, by adopting the recurrent connection of the same block at different time steps, the recurrent structure is able to absorb the contextual and structural information with the hidden convolution units. Second, by sharing weights for multiple times at each layer, the new architecture can increase the depth of traditional CNNs without significantly increasing the total number of parameters.

3.2. Boundary Refinement Network

The RLN can aggregate useful features by filtering out noisy parts and progressively refining the predictions by integrating dependent information. However, some detailed structures along the boundaries of salient objects are still missing. In order to recover continuous details for obtaining spatial precision, we adopt a local Boundary Refinement Network (BRN) [35] to adaptively rectify the prediction.

The details of BRN is illustrated in Figure 5. The saliency map generated by the RLN and the original RGB image

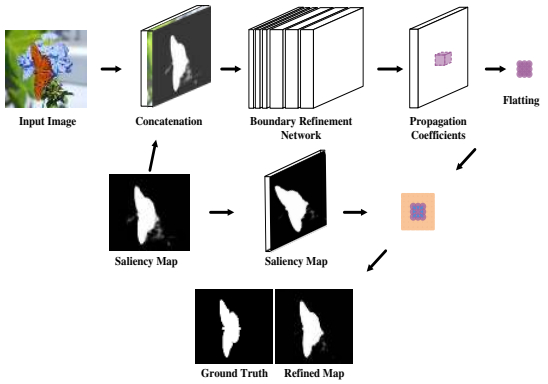


Figure 5. The structure of Boundary Refinement Network (BRN).

are concatenated to serve as the input of BRN. For each position, BRN aims to learn a $n \times n$ propagation coefficient map with which local context information can be aggregated to the center pixel.

For position i, BRN will first output a propagation coefficient vector, which will then be flattened to a $n \times n$ square. The refinement map at position i will be generated by a multiplied sum of the propagation map and the saliency map in the neighborhood of i.

$$\mathbf{s}'_i = \sum_{d=1}^{n \times n} \mathbf{v}_i^d \cdot \mathbf{s}_i^d, d \in 1, 2, ..., n \times n, \tag{5}$$

where $\mathbf{v}_i^d$ is the coefficient vector at position i of the d-th neighbor and $n \times n$ represents the size of local neighbors. $\mathbf{s}_i^d$ and $\mathbf{s}'_i$ denotes the prediction vector at location i before and after the refinement operation, respectively. Each position in BRN is position-adaptive with a different propagation coefficient, which can be automatically learned via back-propagation without explicit supervision.

Implementation details. As shown in Table 2, BRN is composed of 7 convolutional layers, each with the kernel size of 3×3. The ReLU nonlinearity operations are performed between two convolutional layers. We do not utilize pooling layers and large strides in convolutional layers in order to keep the same resolution between input and output feature maps.

Layer	Channel	Kernel size	Bias size
1	64	$(K+3) \times 64 \times 3 \times 3$	64
2	64	$64 \times 64 \times 3 \times 3$	64
3	64	$64 \times 64 \times 3 \times 3$	64
4	128	$64 \times 128 \times 3 \times 3$	128
5	128	$128 \times 128 \times 3 \times 3$	128
6	128	$128 \times 128 \times 3 \times 3$	128
7	$n \times n$	$128 \times (n \times n) \times 3 \times 3$	$n \times n$

Table 2. The parameters of the BRN, where $K = 1$ represents the one-channel saliency map.

The propagation matrices can model spatial relations among neighbors to help refine the predicted map generated by the RLN. Compared to the initial saliency map, the refined one should not change too much in terms of the visual appearance. To achieve this, we adopt the following initialization in BRN:

$$\begin{cases} \mathbf{k}_l(z,c) = \delta, \\ \mathbf{b}(c) = \begin{cases} 1 & l = L, c = (n \times n + 1)/2 \\ 0 & others \end{cases} \end{cases} \tag{6}$$

where $l \in \{1, 2, ..., L\}$ denotes the l-th convolutional layer of BRN. $\mathbf{k}_l$ is the convolutional kernel initialized by the Gaussican distribution $\delta \sim \mathcal{N}(\mu, \sigma^2)$, where $\mu = 0, \sigma = 0.1$. z is the position in each kernel and c represents the index of the channel. We set all bias parameters in the l-th layer $(l < L)$ to 0. For the L-th layer, biases are set to 0 except that the value at the center position of $n \times n$ neighbors is set to 1. Following this initialization, saliency prediction of a certain pixel will be primarily influenced by the central coefficient of the propagation map and also be be affected by the other coefficients.

4. Experiments

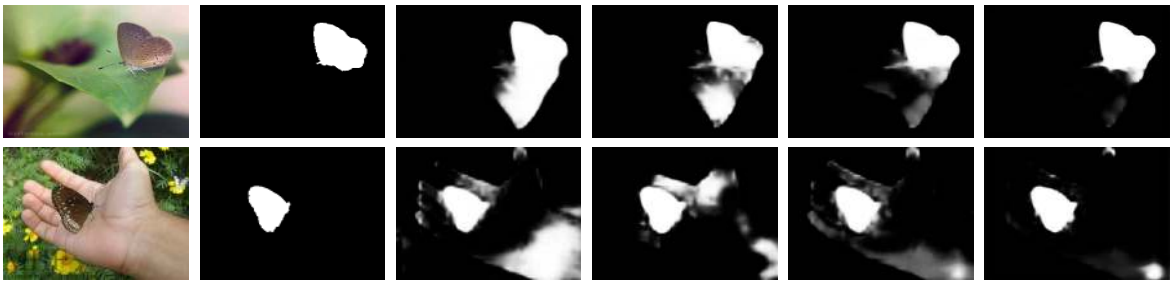
4.1. Experimental Setup

Evaluation Datasets. We evaluate the proposed framework on five popular datasets: ECSSD [31], DUT-OMRON [32], THUR15K [5], HKU-IS [17], and DUT-S [27]. **ECSSD** contains 1,000 natural and complex images with pixel-accurate ground truth annotations. The images are manually selected from the Internet. **DUT-OMRON** has more challenging images with 5,168 images. All images are resized so as to the maximal dimension is 400 pixels long. **THUR15K** includes 6,232 categorized images with 'butterfly', 'coffee', 'dog', 'giraffe' and 'plane'. **HKU-IS** has 4,447 images which are selected by meeting at least one of the following three criteria: multiple salient objects with overlapping, objects touching the image boundary and low color contrast. **DUTS** is the latest released dataset containing 10,553 images for training and 5,019 images for testing. Both training and test sets contain very complex scenarios with high content variety.

Evaluation Criteria. We utilize three evaluation metrics to evaluate the performance of our method with other salient object detection methods, including Precision-Recall (PR) curve, F-measure score and mean absolute error (MAE). Given a saliency map with continuous values normalized to the range of 0 and 255, we compute the corresponding binary maps by using every possible fixed integer threshold. Then we compute the precision/recall pairs of all binary maps to plot the PR curve by a mean value over all saliency maps in a given dataset. Also, we utilize the F-measure score to evaluate the quality of a saliency map, which is formulated by a weighted combination of Precision and Recall.

*	ECSSD [31]		THUR15K [5]		HKU-IS [17]		DUTS [27]		DUT-OMRON [32]	
	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE
Ours	0.903	0.045	0.716	0.077	0.882	0.037	0.768	0.051	0.709	0.063
SRM [29]	0.892	0.056	0.708	0.077	0.874	0.046	0.757	0.059	0.707	0.069
Amulet [33]	0.869	0.061	0.670	0.094	0.839	0.052	0.676	0.085	0.647	0.098
UCF [34]	0.841	0.080	0.645	0.112	0.808	0.074	0.629	0.117	0.613	0.132
KSR [30]	0.782	0.135	0.604	0.123	0.747	0.120	0.602	0.121	0.591	0.131
RFCN [28]	0.834	0.109	0.627	0.100	0.835	0.089	0.712	0.090	0.627	0.111
DS [20]	0.821	0.124	0.626	0.116	0.785	0.078	0.632	0.091	0.603	0.120
DCL [18]	0.827	0.151	0.676	0.161	0.853	0.136	0.714	0.149	0.684	0.157
DHS [22]	0.871	0.063	0.673	0.082	0.852	0.054	0.724	0.067	-	-
LEGS [26]	0.785	0.119	0.607	0.125	0.732	0.119	0.585	0.138	0.592	0.133
MCDL [36]	0.796	0.102	0.620	0.103	0.757	0.092	0.594	0.105	0.625	0.089
MDF [17]	0.805	0.108	0.636	0.109	-	-	0.673	0.100	0.644	0.092
BL [25]	0.684	0.217	0.532	0.219	0.660	0.207	0.490	0.238	0.499	0.239
DRFI [12]	0.733	0.166	0.576	0.150	0.722	0.145	0.541	0.175	0.550	0.138

Table 1. Quantitative evaluation in terms of F-measure and MAE scores. The best two scores are shown in red and blue colors, respectively.



(a) Image (b) GT (c) Baseline (d) CWM (e) RM (f) BRN

Figure 6. Visual examples of the proposed modules.

$$F_\gamma = \frac{(1+\gamma^2)Precision \times Recall}{\gamma^2 Precision + Recall}. \quad (7)$$

γ is set to be 0.3 to emphasize more on precision over recall as suggested in [1].

Given the saliency map S and ground truth mask G, the MAE score can be calculated by the element-wise difference between S and G,

$$MAE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} |S(i,j) - G(i,j)|, \quad (8)$$

where $S(i,j)$ represents the saliency score at position (i,j) and W and H are width and height.

Implementation Details. We have implemented our network on a single Nvidia GTX 1080 GPU. Pre-trained ResNet-50 is used to initialize the convolutional layers in the RLN network (i.e. the $conv1$ to $conv5$ block). Other convolutional parameters are randomly assigned. We train our model on the training set of DUTS and test on its testing set and other datasets. All training and test images are resized to 384×384 as the input to the RLN and 480× 480 to the BRN. We do not use validation set and train the model until its training loss converges. We use the SGD method to train our network. A fixed learning rate is set to 1e-10 for training the RLN and 1e-8 for the BRN with the weight decay 0.0005. We use the softmax entropy loss to train both networks. For the recurrent structure, the time step t is set to 2 and we employ three top supervisions between the ground truth and prediction maps.

4.2. Performance Comparison

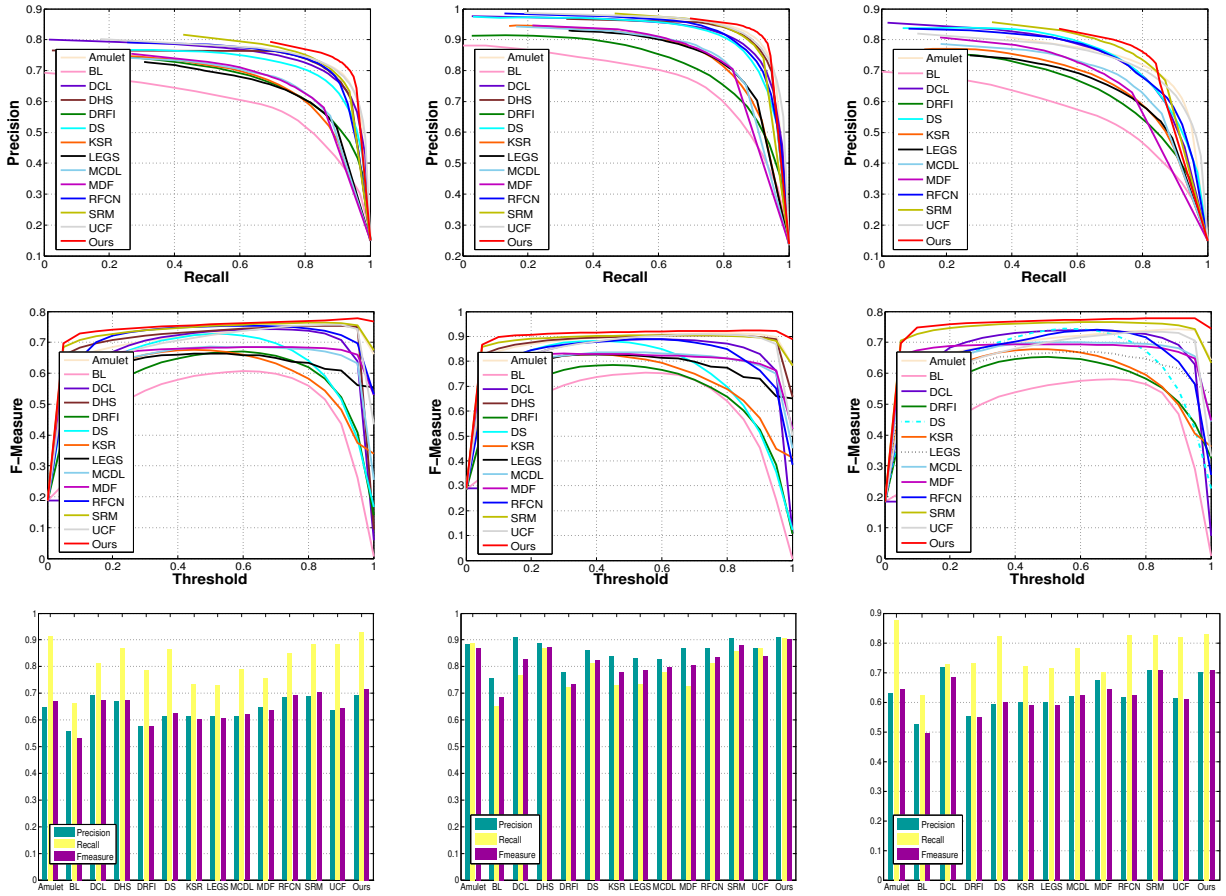
We compare the proposed algorithm against 13 state-of-the-art algorithms, including the deep learning based methods as well as other non-deep competitors, DRFI [12], BL [25], LEGS [26], MDF [17], MCDL [36], DS [20], DCL [18], DHS [22], RFCN [28], KSR [30], UCF [34], Amulet [33] and SRM [29].

Quantitative Evaluation. First, we compare the proposed method with the others in terms of PR curves, F-measure curves and F-measure scores, which are shown in Figure 7. Among all datasets and evaluation metrics, the proposed method performs favorably against other counterparts. Also, we show F-measure and MAE scores in Table 1. As we can see, our approach generates the best score across all datasets. More results can be found in the supplementary material.

Visual Comparison. To qualitatively evaluate the proposed method, we visualize some example saliency maps of our method with respect to the above-mentioned approaches in Figure 8. The examples are shown in various scenarios, including multiple salient objects (row 1-2), the small ob-

*	ECSSD		THUR15K		HKU-IS		DUTS		DUT-OMRON	
	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE
Baseline	0.861	0.058	0.659	0.099	0.838	0.050	0.696	0.073	0.643	0.092
CWM	0.867	0.054	0.667	0.084	0.840	0.047	0.716	0.060	0.661	0.075
RM	0.893	0.048	0.702	0.080	0.875	0.041	0.760	0.054	**0.712**	0.066
BRN	**0.903**	**0.045**	**0.716**	**0.077**	**0.882**	**0.037**	**0.768**	**0.051**	0.709	**0.063**

Table 3. Performance of the proposed modules.



(a) THUR15K dataset (c) ECSSD dataset (d) DUT-OMRON dataset

Figure 7. The first row shows the performance of the proposed method with other state-of-the-art methods in terms of PR curves. The second shows F-measure curves. The last show the precision, recall, and F-measure scores across four datasets. For all metrics, the proposed method achieves better performance than others on all datasets.

ject (row 3), the object touching the image boundary (row 4) and salient objects sharing similar color appearance with the background (row 5-7). From this picture, we can see that our method can produce more accurate saliency maps which are much closer to the ground truth masks.

4.3. Ablation Study

In this section, we provide the results about the contribution of each component in the proposed network.

Performance of the RLN and BRN. To investigate the efficacy of the proposed Recurrent Localization Network (RLN) and the Boundary Refinement Network (BRN), we conduct ablation experiments across all five datasets. We utilize the Base Network described in Section 3.1.1 as our baseline model. The overall results in terms of F-measure and MAE scores are shown in Table 3. Based on the baseline network, we analyze the performance of each proposed component, i.e., the inception-like Contextual Weighting Module (CWM), Recurrent Module (RM), and BRN.

We first evaluate the CWM and the overall performance can be improved for F-measure and MAE scores, respectively. The increased performance benefits from the role that CWM plays in filtering out the noise and cluttered

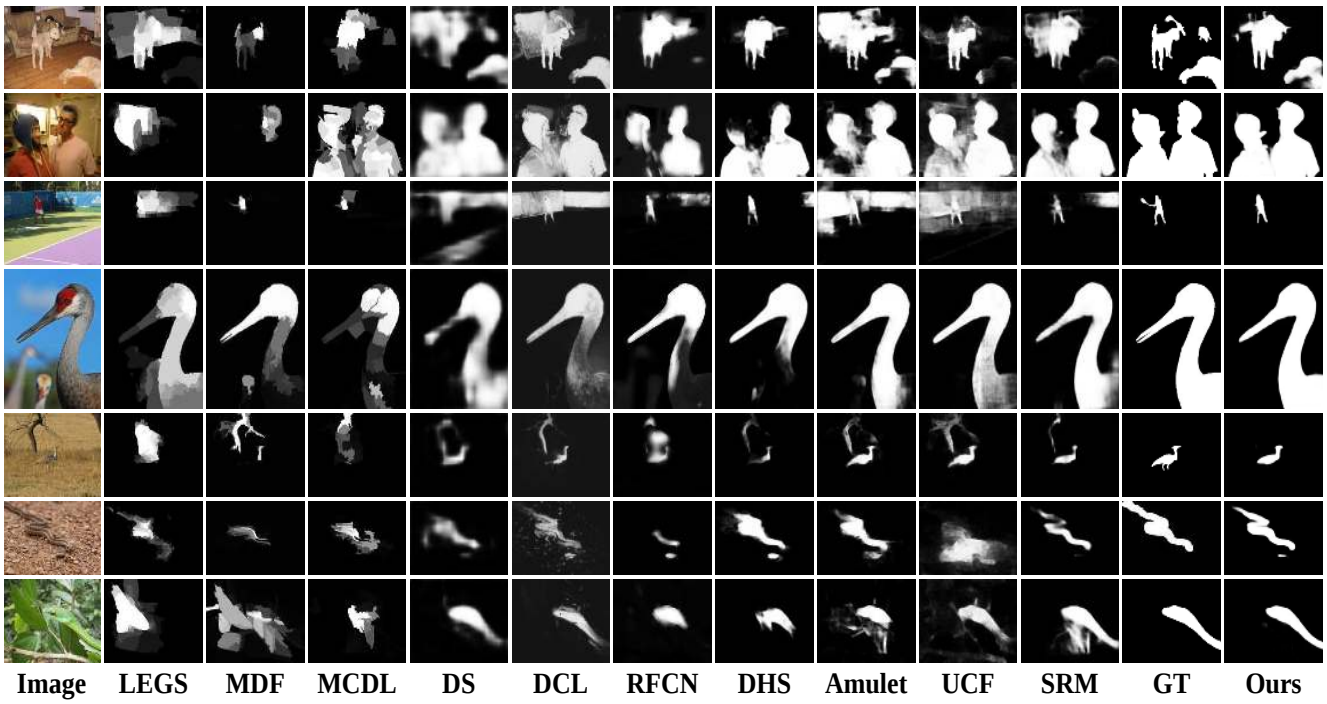


Figure 8. Example results of the proposed method with state-of-the-art.

background information. Besides, through RM, the saliency map can capture contextual dependencies to distinguish confusing local pixels, so the mistakes can be corrected by the network. Both modules can help the network localize salient objects more accurately and remove distractors in background. The final BRN can also show the improvement, deriving from the learned propagation to help adaptively refine the boundaries of predicted map generated by the RLN.

We also provide examples of the RLN and BRN. As shown in Figure 6, with the connection of CWM, RM and BRN, the proposed method can generate more accurate results.

Performance of the controlled experiments. We compare our proposed RLN with different variants on DUTS dataset, as shown in Figure 9. 'RM'-k denotes there are k recurrent modules in our experiment. 'RM-1*' represents no parameters are shared between $t = 0$ and $t = 1$. 'RM-1**' represents that we train the RLN with only one loss at $t = 1$. It can be seen that the performance increases with more time steps. Also, top supervision of each time step and recurrent mechanism are important for the whole network.

5. Conclusion

In this paper, we propose a novel Localization-to-Refinement network for salient object detection from the global and local view. The Recurrent Localization Network (RLN) can learn to better localize salient objects by the weighted response map and a novel recurrent structure is proposed for iteratively refining each convolutional block over time. The Boundary Refinement Network (BRN) can refine the prediction map by the spatial relationship of each pixel and the neighbors. This is achieved via the propagation coefficient map learned by a small deep network. Experimental evaluation verify that the proposed model can consistently improve the state-of-the-art performance on all five benchmark datasets and all popular evaluation metrics.

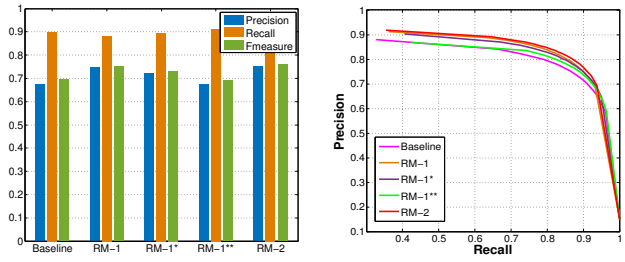


Figure 9. The F-measure scores and PR curves of the controlled experiments on the DUTS dataset.

6. Acknowledge

L. Zhang and H. Lu were supported by the National Natural Science Foundation of China (#61371157, #61472060 and #61528101) and the Fundamental Research Funds for the Central Universities (#DUT2017TB04 and #DUT17TD03)

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009.

[2] S. Bi, G. Li, and Y. Yu. Person re-identification using multiple experts with random subspaces. *Journal of Image and Graphics*, 2(2), 2014.

[3] X. Chen, A. Zheng, J. Li, and F. Lu. Look, perceive and segment: Finding the salient objects in images via two-stream fixation-semantic cnns. In *ICCV*, pages 1050–1058, 2017.

[4] M. Cheng, G. Zhang, N. Mitra, X. Huang, and S. Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015.

[5] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu. Salientshape: Group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014.

[6] C. Craye, D. Filliat, and J.-F. Goudou. Environment exploration for object-based visual saliency learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2303–2309, 2016.

[7] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? *arXiv preprint arXiv:1606.03556*, 2016.

[8] H. Fang, S. Gupta, F. Iandola, and R. K. Srivastava. From captions to visual concepts and back. In *CVPR*, pages 1473–1482, 2015.

[9] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*, pages 597–606, 2015.

[10] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. In *CVPR*, pages 5300–5309, 2017.

[11] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang. Saliency detection via absorbing markov chain. In *ICCV*, pages 1665–1672, 2013.

[12] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, pages 2083–2090, 2013.

[13] H. J. Kim, E. Dunn, and J.-M. Frahm. Learned contextual feature reweighting for image geo-localization. In *CVPR*, pages 3251–3260, 2017.

[14] J. Kim, D. Han, Y.-W. Tai, and J. Kim. Salient region detection via high-dimensional color transform. In *CVPR*, pages 883–890, 2014.

[15] J. Kuen, Z. Wang, and G. Wang. Recurrent attentional networks for saliency detection. In *CVPR*, pages 3668–3677, 2016.

[16] G. Lee, Y. W. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. In *CVPR*, pages 660–668, 2016.

[17] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *CVPR*, pages 5455–5463, 2015.

[18] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *CVPR*, pages 478–487, 2016.

[19] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang. Saliency detection via dense and sparse reconstruction. In *ICCV*, pages 2976–2983, 2013.

[20] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE TIP*, 25(8):3919–3930, 2016.

[21] Y. Lin, Z. Pang, D. Wang, and Y. Zhuang. Task-driven visual saliency and attention-based visual question answering. *arXiv preprint arXiv:1606.03556*, 2017.

[22] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, pages 678–686, 2016.

[23] X. Shen and Y. Wu. A unified approach to salient object detection via low rank matrix recovery. In *CVPR*, pages 853–860, 2012.

[24] M. Simon, E. Rodner, and J. Denzler. Imagenet pre-trained models with batch normalization. *arXiv preprint arXiv:1612.01452*, 2016.

[25] N. Tong, H. Lu, X. Ruan, and M.-H. Yang. Salient object detection via bootstrap learning. In *CVPR*, pages 1884–1892, 2015.

[26] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, pages 3183–3192, 2015.

[27] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, pages 136–145, 2017.

[28] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, pages 825–841, 2016.

[29] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, pages 4039–4048, 2017.

[30] T. Wang, L. Zhang, H. Lu, C. Sun, and J. Qi. Kernelized subspace ranking for saliency detection. In *ECCV*, pages 450–466, 2016.

[31] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013.

[32] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013.

[33] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, pages 202–211, 2017.

[34] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin. Learning uncertain convolutional features for accurate saliency detection. In *ICCV*, pages 212–221, 2017.

[35] R. Zhang, S. Tang, M. Lin, J. Li, and S. Yan. Global-residual and localboundary refinement networks for rectifying scene parsing predictions. In *IJCAI*, pages 3427–3433, 2017.

[36] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *CVPR*, pages 1265–1274, 2015.