The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Recurrently Aggregating Deep Features for Salient Object Detection

Xiaowei Hu,[1,*] **Lei Zhu,**[2,*] **Jing Qin,**[2] **Chi-Wing Fu,**[1,3] **Pheng-Ann Heng**[1,3]
[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China
[2] Centre for Smart Health, School of Nursing, The Hong Kong Polytechnic University, Hong Kong, China
[3] Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
https://github.com/xw-hu/RADF

Abstract

Salient object detection is a fundamental yet challenging problem in computer vision, aiming to highlight the most visually distinctive objects or regions in an image. Recent works benefit from the development of fully convolutional neural networks (FCNs) and achieve great success by integrating features from multiple layers of FCNs. However, the integrated features tend to include non-salient regions (due to low level features of the FCN) or lost details of salient objects (due to high level features of the FCN) when producing the saliency maps. In this paper, we develop a novel deep saliency network equipped with recurrently aggregated deep features (RADF) to more accurately detect salient objects from an image by fully exploiting the complementary saliency information captured in different layers. The RADF utilizes the multi-level features integrated from different layers of a FCN to recurrently refine the features at each layer, suppressing the non-salient noise at low-level of the FCN and increasing more salient details into features at high layers. We perform experiments to evaluate the effectiveness of the proposed network on 5 famous saliency detection benchmarks and compare it with 15 state-of-the-art methods. Our method ranks first in 4 of the 5 datasets and second in the left dataset.

Introduction

Salient object detection aims to identify the most visually distinctive objects from an input image. By working as a preprocessing step of many computer vision tasks, detecting salient objects benefits lots of practical applications, such as image and video compression (Guo and Zhang 2010), content-aware image editing (Cheng et al. 2010), object recognition (Wei et al. 2017), scene classification (Siagian and Itti 2007), object re-targeting (Sun and Ling 2013), and visual tracking (Hong et al. 2015). However, salient object detection is a very challenging research problem as many mutually affected factors contribute to the definition of saliency regions, including image structure, object semantic meaning and context information.

Traditional saliency detection methods employed hand-crafted visual features (e.g, color, texture, and contrast) with heuristic priors (Cheng et al. 2015; Jiang et al. 2013; Liu et al. 2011) to distinguish salient objects from background. These hand-crafted features and priors are ineffective to capture the high level semantic knowledge, and are incapable of producing satisfactory predictions. To improve the detection accuracy, salient object detection algorithms (Li et al. 2016; Li and Yu 2015; Zhao et al. 2015) based on fully convolutional neural network (FCN) (Long, Shelhamer, and Darrell 2015) have been proposed, aiming at leveraging the deep features with more semantic information to generate high-quality saliency maps. However, due to the pooling operations, the outputs of these FCN-based methods, albeit containing richer high-level semantic information compared with the results of hand-crafted feature based methods, lose much significant location information and hence neglect many fine details. Hence, their results usually suffer from poor localization of salient object boundaries. More recently, several works (Liu and Han 2016; Li and Yu 2016; Hou et al. 2017) utilized short connections to combine multi-level features produced from deep convolutional neural networks in order to incorporate semantic information at high levels and detail structures at low levels. However, their salient results still tend to contain many non-salient objects and simultaneously lose some parts (details) of salient object when directly merging multiple level features in those methods; see the 1st and 3rd rows of Figure 1.

In this paper, in order to address these challenges, we propose a novel deep saliency network to recurrently aggregate deep features (RADF) for salient object detection by fully exploiting the complementary information encoded in features generated in different layers. Firstly, we combine the features at multiple layers of a FCN and compress these multi-level features into one (we call it multi-level integrated features (*MLIF*)). Then the compressed *MLIF* are merged with the features of each layer of the FCN using a convolutional operation, which is capable of automatically selecting the discriminative features and suppressing information of non-salient regions. This convolution enables the *MLIF* to refine the features at each layer, where non-salient regions of low levels are reduced, and salient details at high levels are enhanced. Moreover, we adopt a recurrent mechanism to integrate the features at individual layers and the *MLIF* iteratively for a progressive refinement of both of them. Specifically, we combine the refined features at each layer to generate a new MLIF, which are aggregated to each individual

[*]Both authors contributed equally.

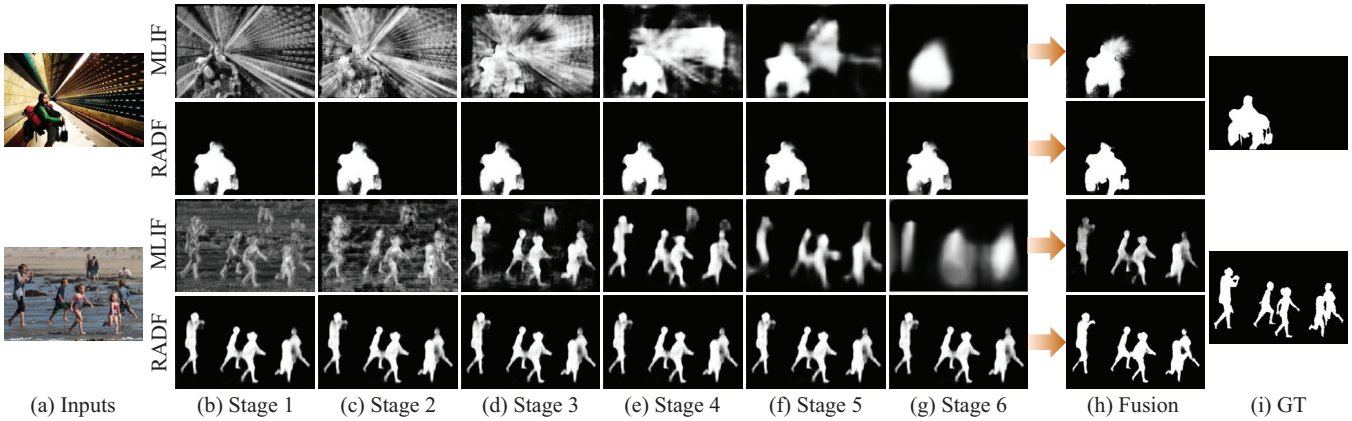


Figure 1: Visual comparisons of predicted saliency maps from different features. The 1st and 3rd rows in the middle show the predicted saliency maps using the multi-level integrated features (MLIF) and features at each layer, while the 2nd and 4th rows in the middle use our proposed method that recurrently aggregates deep features (RADF). (a) is the input images; (b) to (g) represent the results from stage 1 (lowest layer) to stage 6 (highest layer); (h) is the final fusion results predicted by MLIF and RADF; and (i) denotes the ground truths. From the results, we can observe that MLIF tends to detect extra non-salient regions (Figure 1 (h) in the 1st row) or over suppress salient details (Figure 1 (h) in the 3rd row). Contrarily, the salient results (Figure 1 (h) in the 2nd and 4th rows) with our aggregated features are much closer to the ground truths.

layer for next recurrent step. By harnessing RADF, we can produce high quality features with both semantic and detailed information of salient regions while effectively suppressing noise from non-salient regions and hence produce more accurate prediction maps. In addition, we impose the supervision signal to the network at each recurrent step, so that the network can generate more useful information towards to the salient regions. The whole network is trained in an end-to-end manner.

To verify the effectiveness of the proposed RADF model, we evaluate our network equipped with RADF on five famous salient object detection benchmarks, and compare our results against 15 state-of-the-art methods. The experiment results demonstrate that our model quantitatively and qualitatively outperforms others with respect to the accuracy of salient object detection; see Figure 1 and Figure 3 for visual comparisons. Overall, the contributions of this work can be summarized as follows:

- First, we find that simply integrating multi-level deep features are not enough for the salient object detection task and propose to more effectively leverage the complementary information encoded in the features generated in different layers.
- Second, we develop a FCN with a novel scheme to aggregate the multi-level deep features to features of each layer in a recurrent manner. We call it RADF. Such a scheme can produce more distinguishing features containing both semantic and detailed information of salient objects. In addition, the proposed RADF, as a general strategy to aggregate multiple level deep features, has potential to be used in other computer vision applications such as object detection and semantic segmentation.
- Third, we evaluate the proposed method on five famous benchmark datasets and compare it with 15 state-of-the-art methods. We consistently achieve better performance than all the 15 algorithms on 4 of the 5 datasets and rank second in the left dataset. Overall, we set a new state-of-the-art performance on salient object detection.

Related Work

In this section, we do not aim to be exhaustive, but will focus on salient object detection methods. Early methods are based on hand-crafted visual priors, including image contrast (Perazzi et al. 2012; Jiang et al. 2013), color (Borji and Itti 2012; Mahadevan and Vasconcelos 2013), texture (Yan et al. 2013; Yang et al. 2013) and other kinds of relevant visual cues (Harel, Koch, and Perona 2007). More comprehensive analysis of these hand-crafted feature based methods can be found in (Borji et al. 2015). However, these features have limited ability of feature representation, which is difficult to capture high-level semantic meaning of salient objects.

Recently, fully convolutional neural network (Long, Shelhamer, and Darrell 2015) (FCN) based algorithms (Zhao et al. 2015; Li et al. 2016; Wang et al. 2016; Luo et al. 2017; Zhang et al. 2017b) have achieved remarkable performance on salient object detection due to strong feature representation ability. For example, Zhao et al. (Zhao et al. 2015) used FCN to get the features of full image to model the global context and the features of a part of image to model the local context. Then they designed a multi-context framework to integrate the global and local context for salient object detection since these two kinds of context information are able to determine the salient objects from different views. Wang et al. (Wang et al. 2016) developed a recurrent fully convolutional network to predict saliency maps based on the prediction results of the last recurrent step and the original image. Hence, the saliency map can be stage-wisely refined by the deep network. Zhang et al. (Zhang et al. 2017b) learned deep uncertain convolutional features by a dropout technique that

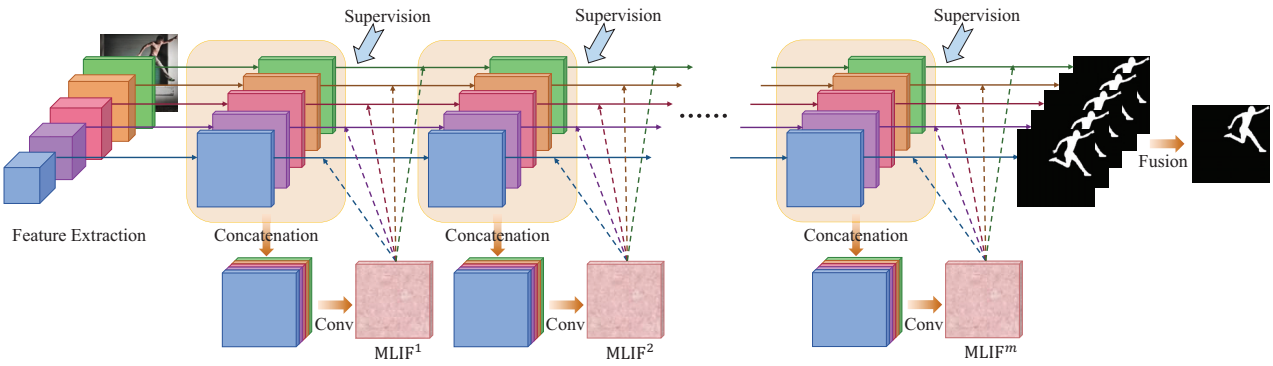


Figure 2: The schematic illustration of proposed RADF. An input image is sent to the convolutional neural network and a set of feature maps with multiple scales are obtained. The feature maps with different scales are upsampled to the size of input image and concatenated together as the multi-level integrated features (MLIF). Then MLIF is added to the features of each layer and merged by a convolutional operation. This step is performed m iterations to alternatively refine MLIF and the features at each layer. Moreover, the deep supervision mechanism is imposed at each step. Finally, output score maps at the last step are merged together and 1*1 conv is used to generate the fusion score map. Best viewed in color.

randomly drops units of deep network during the training process, aiming at incorporating uncertainties for improving generalization capability which is important for salient object detection. Unfortunately, these methods just produce saliency maps from the features at deep layers of FCN and are difficult to handle low-level details of saliency regions.

To remedy the above issue, several efforts (Li and Yu 2016; Hou et al. 2017; Zhang et al. 2017a) focus on finding an integration strategy of multi-level features to simultaneously encode high-level semantic information and low-level detail information for high-quality saliency maps. Li at al. (Li and Yu 2016) integrated the feature maps from multiple layers with different resolutions to capture the semantic properties and the visual contrast of salient objects. Hou et al (Hou et al. 2017) aimed to help low-level features have the knowledge of object localization information by integrating the prediction maps of deep layers and the features of shallow layers. Zhang et al (Zhang et al. 2017a) concatenated feature maps at multiple resolutions and predicted saliency maps from the integrated features. All of them are aimed to explore different combinations of features at multiple layers to integrate salient visual cues from multiple views. However, not all the multi-level features are useful for salient object detection. The rich multi-level features contain redundant information that will cover useful features and introduce undesirable information, so that the network is difficult to effectively leverage the deep features of FCN to detect salient objects accurately.

Method

The workflow of the proposed FCN equipped with RADF is illustrated in Figure 2. Our network takes the whole image as input and outputs the saliency map in an end-to-end manner. It begins by utilizing the convolutional neural network to generate a set of hierarchical features which encode the detail and semantic information with different scales in a pyramid. The features at different levels of this hierarchical pyramid represent the objects in the image and their contextual information from different views (Mahendran and Vedaldi 2015), which are proved useful for salient object detection (Hou et al. 2017). After getting the hierarchical feature maps, we enlarge them to the size of input image and concatenate them together followed by a convolution layer with 1*1 kernel to reduce channels of feature maps to a small number. We denote these new features as the multi-level integrated features (*MLIF*), as they encompass the information from features at multiple levels. Then, we aggregate the generated *MLIF* with each layer by leveraging a convolution operation in order to refine the features at each layer for better saliency representations. After that, we integrate the newly generated features at each layer again and form refined *MLIF* and add them back to each individual layer to further refine them. We alternatively refine the *MLIF* and features at each layer in a recurrent manner (e.g., m iterations in Figure 2). Meanwhile, the deep supervision mechanism is also imposed to the network at each recurrent step and at last we obtain the saliency prediction map by fusing outputs from multiple layers, as shown in Figure 2.

In the following subsections, we will elaborate how to effectively integrate the multi-level features and recurrently aggregate the *MLIF* and features at each layer in our RADF.

Integrating Multi-Level Features

One of the main advantage of convolutional neural network is that its hierarchical structure, once well-trained, is capable of producing well-organized features consisting of abundant semantic and fine information (Kong et al. 2016). Note that in our task salient objects in an image are determined by various factors such as multi-scale contextual information, the semantic meaning of the objects and their boundary details. In this regard, integrating multi-level features is able to enhance the discrimination capability for salient object detection, as while the deep layers can capture highly semantic features tending to describe the attributes of salient objects as a whole, the shallow layers are more effective to extract

subtly fine features to represent delicate structures. Both of them are essential for accurate salient object detection.

We aim to fully exploit the complementary information encoded in multi-level deep features for better salient object detection. Bearing this idea in mind, we propose to integrate features at all stages (a stage includes several layers of feature maps with the same resolution and here we harness the feature map of the last layer in the stage as the feature map of the whole stage) and enlarge them to the size of the input image. Specifically, we firstly apply 1*1 convolution to reduce the dimensions (channels) of feature maps at each stage and up-sample these feature maps to the size of input image. After that, we concatenate all the enlarged feature maps followed by a convolution operation to merge the features from different stages and reduce the feature dimensions (as shown in Figure 2). The *MLIF* is defined as:

$$MLIF = \sigma(W * Cat(F_1, F_2, ..., F_n) + b), \qquad (1)$$

where the F_i is the enlarged feature map at i-th stage and n is the total number of stages; Cat is the concatenation operation across channels; $*$ represents convolution operation; W and b are the weights and bias of the convolution, which can be learned from the training data; σ is the activation function and we use ReLU (Krizhevsky, Sutskever, and Hinton 2012) in our implementation.

Recurrently Aggregating Deep Features

Although multi-level integrated features (*MLIF* in Eq. 1) have encoded lots of fruitful saliency cues from different levels of FCN, directly using *MLIF* to predict salient objects cannot guarantee satisfactory results. The predicated saliency maps may still include many non-salient regions and lose parts of saliency regions, as illustrated in Figure 1. This is because the generated *MLIF*, albeit including features extracted from different levels, also incorporates a lot of non-salient details from shallow layers and some wrong semantic information irrelevant to saliency regions from deeper layers. Not only would this information provide wrong guidance for saliency map generation, but also it will weaken the useful information originally containing in individual layers. In this regard, some researchers propose to further post-process the *MLIF* to refine the prediction results (Li et al. 2016; Wang et al. 2016).

In this paper, we propose a novel method to leverage the complementary advantages of the *MLIF* and features in individual layers to achieve better prediction results than existing methods only depending on the *MLIF*. To achieve this, we propose a deep saliency network to recurrently aggregate the *MLIF* to each individual layer. In our recurrent aggregation process, the *MLIF* can work as a fruitful feature pool to refine the features of individual layers and then the refined features in individual layers are integrated together again to generate refined *MLIF* in order to progressively push more saliency information back and reduce non-salient cues in the feature pool. Specifically, as the features at shallow layers are responsible for discovering the fine detail information but lack of semantic information of salient regions, the *MLIF* can be used as a guidance to help them gradually suppress details that are not located in the semantic saliency regions while capturing more details in semantic saliency regions based on semantic information integrated in the *MLIF*. On the other hand, as features at deep layers are responsible for capturing cues of the whole salient objects and somehow lack of salient details due to the relatively larger receptive fields than shallow layers, the *MLIF* can serve as an enhancer to help them complement more salient boundary details based on the meticulous features integrated in the *MLIF*. By refining the *MLIF* and features of individual layers recurrently, our network can learn to select more discriminative multi-level features targeting for salient object detection and progressively refine the results.

Specifically, after getting the *MLIF*, we aggregate it to the feature maps of all stages. Then a convolutional operation is used to select the useful multi-level information with respect to the features of each individual layer as well as reduce the number of feature channels to the original number before aggregation. At the j-th recurrent step of our network, we compute features (denoted as F_i^j) for each stage i as:

$$F_i^j = \sigma(W_i^j * Cat(F_i^{(j-1)}, MLIF^j) + b_i^j), \qquad (2)$$

where $F_i^{(j-1)}$ represents the features for each stage i at (j-1)-th step; the initial value F_i^0 is the initial features at i-th stage generated by the FCN network. The $MLIF^j$ (see Equation 1 for its definition) is the obtained multi-level integrated features at the j-th step by taking $F_i^{(j-1)}$ as input, and we compute the $MLIF^1$ (*MLIF* at the first iteration) using Equation 1 with F_i^0 as the input features at i-th stage. The $*$ denotes the convolution operation; note that the convolutional weights W_i^j and bias b_i^j can be automatically optimized to achieve distinguishing features of i-th stage at j-th recurrent step during the end-to-end training procedures. Meanwhile, reducing the feature channels to a unified number promotes the consistent feature aggregation, making recurrently aggregating multi-level information possible.

In addition, we apply deeply supervised mechanism (Xie and Tu 2015) to impose the supervision signal to the last layer of each stage at each recurrent step in order to enhance the capability of the proposed network to find salient features during the feature aggregation process. With the help of deeply supervision, we are able to get multiple prediction results for better prediction. Let P_i^j denote the predicted score map at the i-th stage during the j-th step. Then, the fused score map is generated (denoted as P_f) by adding a convolution layer on score maps predicted from n layers, and the definition of P_f is given by:

$$P_f = \sigma(W_f * Cat(P_1^m, P_2^m, ..., P_n^m) + b_f), \qquad (3)$$

where m is last recurrent step; $Cat()$ is used to concatenate the score maps; W_f and b_f are the weight and bias of the convolution layer on the concatenated score maps to learn the relationship among these score maps, respectively.

Experiments

In this section, we describe the training and testing strategies of our RADF, introduce the benchmark datasets and evaluation metrics used by the research society on salient object detection, and conduct experiments on salient datasets

Table 1: The F-measure and MAE of different settings on five saliency detection datasets. "RADF-D" denotes the network using DenseNet to replace VGG part of our model. "RADF-i" is the network only using features of each individual layer. "RADF-m" denotes the network only using multi-level features. "RADF1" and "RADF2" are our networks aggregating multi-level features and features at individual layers once and twice respectively. "RADF2-s" denotes weights shared in recurrent steps.

Method	ECSSD		HKU-IS		PASCAL-S		SOD		DUT-OMRON	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
RADF-D	0.886	0.068	0.865	0.060	0.781	0.126	0.788	0.148	0.729	0.086
RADF-i	0.913	0.054	0.908	0.041	0.826	0.105	0.827	0.129	0.763	0.072
RADF-m	0.917	0.053	0.907	0.046	0.828	0.107	0.829	0.131	0.772	0.066
RADF1	0.923	0.050	0.913	0.041	0.828	0.106	0.831	0.129	0.787	0.063
RADF2	**0.924**	**0.049**	**0.914**	**0.039**	**0.832**	**0.102**	**0.835**	**0.125**	0.789	**0.060**
RADF2-s	0.923	**0.049**	0.911	0.043	0.830	0.105	0.829	0.131	**0.793**	0.062

to evaluate the effectiveness of proposed network equipped with RADF.

Training and Testing Strategies

We choose the VGG (Simonyan and Zisserman 2014) network to produce the feature extraction layers, and we use conv1_2, conv2_2, conv3_3, conv4_3, conv5_3 and pool5 of the VGG network to represent the features of each individual layer. Moreover, we add two more convolutional layers to further enhance the discrimination capability of feature maps at each stage. Other implementation details can be found in our public codes.

Training During the training process, cross-entropy loss is used for each output of this network. The total loss L_t is defined as the loss summation of all predicted score maps:

$$L_t = \sum_{i=1}^{n}\sum_{j=1}^{m} w_i^j L_i^j + w_f L_f, \tag{4}$$

where w_i^j and L_i^j represent the weight and loss of i-th stage at j-th step; n and m denote the total stages of the network and the maximum steps, respectively; and w_f and L_f are the weight and loss for the fusion layer, respectively. In our experiment, we empirically set all the weights to 1.

In order to accelerate the training process and reduce the over-fitting problem, the parameters of the feature extraction layers (as shown in Figure 2) are initialized from the well-trained VGG network (Simonyan and Zisserman 2014). Other layers are initialized by random noise. Stochastic gradient descent (SGD) is used to optimize the whole network with the momentum of 0.9 and the weight decay of 0.0005. We set the learning rate as 1e-8 and it reduces by a factor of 0.1 at 7k iterations. Learning stops after 10k iterations. Our network equipped with RADF is trained on the MSRA10K dataset (Cheng et al. 2015) which is widely used for training the salient object detection models (Lee, Tai, and Kim 2016; Zhang et al. 2017a). In addition, images of this dataset are randomly rotated, resized and horizontally flipped for data argumentation, and our model is trained on 4 GPUs with a mini-batch size of 4.

Inference In testing, for each input image, our network produces several output score maps, since we add a supervision signal to each stage and each recurrent steps. The final prediction map P_{final} is obtained by averaging all the score maps (P_i^j denotes the score map at i-th stage and j-th step) as well as the fusion score map P_f:

$$P_{final} = Mean(P_1^1, ..., P_n^m, P_f). \tag{5}$$

After getting the final prediction map, we apply the fully connected conditional field (CRF) (Krähenbühl and Koltun 2011) to improve the spatial coherence of the prediction map by considering the relationships of neighborhood pixels.

Datasets and Evaluation Metrics

We employ a benchmark dataset to train the proposed model and perform experiments to validate its effectiveness on five widely-used datasets.

Benchmark Datasets Our training dataset (MSRA10K dataset (Cheng et al. 2015)) has $10,000$ images with high-quality pixel-wise annotations, and most of them contain only one salient object.

ECSSD (Yan et al. 2013) This dataset consists of $1,000$ natural images with many semantically meaningful and complex structures.

HKU-IS (Li and Yu 2015) It includes $4,447$ images. Most of the images have low contrast, or more than one salient object.

PASCAL-S (Li et al. 2014) This dataset has 850 challenging images with several objects, which are carefully selected from the PASCAL VOC dataset (Everingham et al. 2010).

SOD (Martin et al. 2001; Movahedi and Elder 2010) It is composed of 300 images selected from the BSDS dataset. This dataset is challenging, since most of images possess multiple objects either with low contrast or touching the image boundary.

DUT-OMRON (Yang et al. 2013) This dataset contains $5,168$ high-quality images. Each image has one or more salient objects, and thus saliency detection on this dataset is very difficult and challenging. Since images of DUT-OMRON has controversial saliency annotations among different human observers, none of existing methods achieves a high accuracy of detecting salient objects on this dataset.

Evaluation Metrics To quantitatively evaluate the performance of different saliency models, two widely-used metrics are employed: F-measure and mean absolute error (MAE). A better performance has a larger F-measure

Table 2: Comparison with the state-of-the-arts. The top three results are highlighted in red, green, and blue, respectively.

Method	ECSSD		HKU-IS		PASCAL-S		SOD		DUT-OMRON	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
MR (Yang et al. 2013)	0.736	0.189	0.715	0.174	0.666	0.223	0.619	0.273	0.610	0.187
wCtr* (Zhu et al. 2014)	0.716	0.171	0.726	0.141	0.659	0.201	0.632	0.245	0.630	0.144
BSCA (Qin et al. 2015)	0.758	0.183	0.723	0.174	0.666	0.224	0.634	0.266	0.616	0.191
MC (Zhao et al. 2015)	0.822	0.106	0.798	0.102	0.740	0.145	0.688	0.197	0.703	0.088
LEGS (Wang et al. 2015)	0.827	0.118	0.770	0.118	0.756	0.157	0.707	0.215	0.669	0.133
MDF (Li and Yu 2015)	0.831	0.108	0.860	0.129	0.759	0.142	0.785	0.155	0.694	0.092
ELD (Lee, Tai, and Kim 2016)	0.867	0.080	0.844	0.071	0.771	0.121	0.760	0.154	0.719	0.091
DS (Li et al. 2016)	0.882	0.123	-	-	0.758	0.162	0.781	0.150	0.745	0.120
FPN (Lin et al. 2016)	0.895	0.062	0.896	0.044	0.793	0.114	0.808	0.126	0.730	0.084
DeepLab (Chen et al. 2016)	0.904	0.053	0.890	0.041	0.812	0.108	0.810	0.128	0.765	0.068
RFCN (Wang et al. 2016)	0.898	0.097	0.895	0.079	0.827	0.118	0.805	0.161	0.747	0.095
DCL (Li and Yu 2016)	0.898	0.071	0.904	0.049	0.822	0.108	0.832	0.126	0.757	0.080
DHSNet (Liu and Han 2016)	0.907	0.059	0.892	0.052	0.827	0.096	0.823	0.127	-	-
NLDF (Luo et al. 2017)	0.905	0.063	0.902	0.048	0.831	0.099	0.810	0.143	0.753	0.080
UCF (Zhang et al. 2017b)	0.910	0.078	0.886	0.073	0.821	0.120	0.800	0.164	0.735	0.131
DSS (Hou et al. 2017)	0.916	0.053	0.911	0.040	0.829	0.102	0.842	0.118	0.771	0.066
Amulet (Zhang et al. 2017a)	0.913	0.059	0.887	0.053	0.828	0.095	0.801	0.146	0.737	0.083
RADF (ours)	0.924	0.049	0.914	0.039	0.832	0.102	0.835	0.125	0.789	0.060

value and a smaller MAE value; see (Achanta et al. 2009; Hou et al. 2017) for the detailed definitions. To have fair comparisons, we apply the implementations of (Hou et al. 2017) to compute these two metrics.

Ablation Analysis

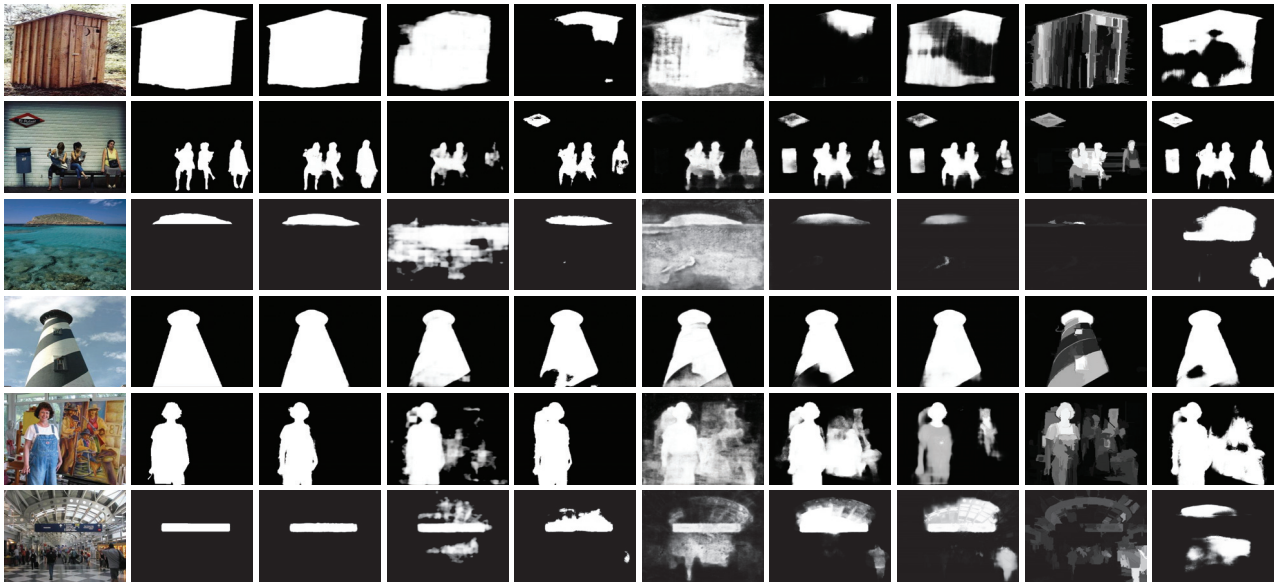
We perform ablation experiments to evaluate the effectiveness of the proposed RADF. These ablation experiments are implemented on the five datasets mentioned above. Here, we set two *baselines*. These two baselines have similar structures with our RADF, but one (RADF-i) just predicts the saliency map based on the features of each layer without employing the *MLIF* and another one (RADF-m) uses the multi-level integrated features to predict the saliency maps directly. For our RADF, we set different number of steps to aggregate the deep features between *MLIF* and features of each layer, which is used to verify the importance of recurrent aggregation. Moreover, we compare our model with shared weights in the two recurrent steps (RADF2-s), and another with a DenseNet (Huang et al. 2017) (161 layers) to replace the VGG part of our model (RADF-D) by re-training it for saliency detection.

As shown in Table 1, the two baselines obtain comparable performances while our RADF achieves an obvious improvement compared with these two baselines, demonstrating that by taking the complementary advantaged of *MLIF* and the features in each individual layer, the proposed RADF can effectively beef up the discrimination capability of the saliency detection network. Moreover, the "RADF2" outperforms "RADF1" on all datasets, corroborating both the *MLIF* and the features in each individual layer can be gradually refined under the proposed recurrent aggregation scheme. And the "RADF2" with separated weights has slightly better results. By comparing "RADF-D" and "RADF2", we can observe that our results are better than that of RADF-D for all the 5 benchmark datasets of saliency detection. The reason is that the feature map resolutions of DenseNet are smaller than VGG network with the limited memory, resulting in losing detail information.

Comparison with the State-of-the-arts

We further extensively compare the results of our method with 15 state-of-the-art methods for salient object detection (see the first column of Table 2 for compared methods), a semantic segmentation algorithm (DeepLab (Chen et al. 2016)), and an object detector (FPN (Lin et al. 2016)). Among these salient object detection algorithms, MR (Yang et al. 2013), wCtr* (Zhu et al. 2014) and BSCA (Qin et al. 2015) are based on hand-crafted features while others are deep learning based methods. For DeepLab (Chen et al. 2016) and FPN (Lin et al. 2016), we re-train their models to detect salient objects. For a fair comparison, we obtain the saliency results of our competitors by using either the saliency maps provided by the authors, or the implementations with recommended parameter setting.

Quantitative Comparison Table 2 reports the results of F-measure and MAE of different methods. It is observed that our method consistently outperforms others on almost all the five datasets in terms of both two metrics, indicating the advantages of our methods over existing approaches. Looking into the quantitative results in Table 2, we make the following observations. (1) Our RADF achieves a great improvement on both metrics (F-measure and MAE) on two relatively larger datasets (ECSSD and DUT-OMRON) with complex salient structures and/or controversial salient regions. It proves that our method is capable of generating more distinguishing features to tackle these challenging regions that previous methods cannot well dealt with. Follow-



(a) Image (b) GT (c) RADF (d) Amulet (e) DSS (f) UCF (g) NLDF (h) DCL (i) MDF (j) FPN

Figure 3: Visual comparison of saliency maps. Note that "GT" stands for "Ground truth". Apparently, our method (RADF) can produce more accurate saliency maps than others. More comparisons can be found at this paper's website.

ing visual comparisons further demonstrate this point. (2) While a lot of previous methods, including DCL, DHSNet, DSS and Amulet, aimed to use the multi-level features to improve the detection accuracy, they neglect that the integrated multi-level features may contain many non-salient regions from shallow layers and lose some salient details when integrating the semantic information from deeper layers. The proposed RADF alleviate these shortcomings by recurrently aggregating *MLIF* and the features of each layer, hence we achieve superior performances in both metrics than these methods. (3) Although our method is not the best on the SOD datasets, it is still very competitive with a 2nd rank. Note that this dataset is relative small compared to other datasets (Zhang et al. 2017a), with just 300 images. (4) Our RADF is just trained on MSRA10k dataset, but it still outperforms other methods (e.g., RFCN and MDF) that are pretrained on PASCAL-S or HKU-IS, indicating a good generalization capability of the proposed RADF, which is essential for saliency detection models. (5) Our method also achieves superior performance than the re-trained semantic segmentation algorithm (DeepLab) and object detector (FPN) for saliency detection.

Visual Comparison We further provide some typical saliency maps of different methods to intuitively demonstrate advantages of the proposed RADF over other methods, as shown in Figure 3. From these results, we can observe that our RADF is more effective to detect the saliency regions accurately, and obtain more clean backgrounds (less false positive) for input images. In most of these examples, especially those challenging ones, our method achieves much better results than others, demonstrating the effectiveness and robustness of the proposed RADF.

Conclusion

In this paper, we propose a novel FCN with recurrently aggregated deep features for salient object detection. In order to take full advantages of the complementary information encoded in the features captured from different layers of the FCN, we employ multi-level features to progressively refine the features of each layer. During the recurrent aggregation procedure, non-salient noise in low layer features are gradually reduced and the saliency details in high layer features are continuously enhanced. As a result, we can generate more discriminative features for more accurate salient object detection. In addition, the supervision signal is imposed into the layers of each stage in each recurrent step. Extensive experiments corroborate the effectiveness of the proposed network with RADF. The proposed feature aggregation scheme is general enough and has great potential to be used in other applications such as instance detection and semantic segmentation.

Acknowledgments

This work was supported by National Basic Program of China, 973 Program (Project No. 2015CB351706), the grant from the Research Grants Council of the Hong Kong Special Administrative Region (Projet No. CUHK 14225616), the grant from the Hong Kong Polytechnic University (Project no. 1-ZE8J), the CUHK strategic recruitment fund and direct grant (4055061), and the grant from the Shenzhen Science and Technology Program (JCYJ20170413162617606).

References

Achanta, R.; Hemami, S.; Estrada, F.; and Susstrunk, S. 2009. Frequency-tuned salient region detection. In *CVPR*.

Borji, A., and Itti, L. 2012. Exploiting local and global patch rarities for saliency detection. In *CVPR*.

Borji, A.; Cheng, M.-M.; Jiang, H.; and Li, J. 2015. Salient object detection: A benchmark. *IEEE TIP* 24(12):5706–5722.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*.

Cheng, M.-M.; Zhang, F.-L.; Mitra, N. J.; Huang, X.; and Hu, S.-M. 2010. Repfinder: Finding approximately repeated scene elements for image editing. *ACM Transactions on Graphics (SIGGRAPH)* 29(4):83:1–83:8.

Cheng, M.-M.; Mitra, N. J.; Huang, X.; Torr, P. H.; and Hu, S.-M. 2015. Global contrast based salient region detection. *IEEE TPAMI* 37(3):569–582.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *IJCV* 88(2):303–338.

Guo, C., and Zhang, L. 2010. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE TIP* 19(1):185–198.

Harel, J.; Koch, C.; and Perona, P. 2007. Graph-based visual saliency. In *NIPS*.

Hong, S.; You, T.; Kwak, S.; and Han, B. 2015. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*.

Hou, Q.; Cheng, M.-M.; Hu, X.-W.; Borji, A.; Tu, Z.; and Torr, P. 2017. Deeply supervised salient object detection with short connections. In *CVPR*.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *CVPR*.

Jiang, H.; Wang, J.; Yuan, Z.; Wu, Y.; Zheng, N.; and Li, S. 2013. Salient object detection: A discriminative regional feature integration approach. In *CVPR*.

Kong, T.; Yao, A.; Chen, Y.; and Sun, F. 2016. Hypernet: Towards accurate region proposal generation and joint object detection. In *CVPR*.

Krähenbühl, P., and Koltun, V. 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Lee, G.; Tai, Y.-W.; and Kim, J. 2016. Deep saliency with encoded low level distance map and high level features. In *CVPR*.

Li, G., and Yu, Y. 2015. Visual saliency based on multiscale deep features. In *CVPR*.

Li, G., and Yu, Y. 2016. Deep contrast learning for salient object detection. In *CVPR*.

Li, Y.; Hou, X.; Koch, C.; Rehg, J. M.; and Yuille, A. L. 2014. The secrets of salient object segmentation. In *CVPR*.

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

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2016. Feature pyramid networks for object detection. *arXiv preprint arXiv:1612.03144*.

Liu, N., and Han, J. 2016. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*.

Liu, T.; Yuan, Z.; Sun, J.; Wang, J.; Zheng, N.; Tang, X.; and Shum, H.-Y. 2011. Learning to detect a salient object. *IEEE TPAMI* 33(2):353–367.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Luo, Z.; Mishra, A.; Achkar, A.; Eichel, J.; Li, S.; and Jodoin, P.-M. 2017. Non-local deep features for salient object detection. In *CVPR*.

Mahadevan, V., and Vasconcelos, N. 2013. Biologically inspired object tracking using center-surround saliency mechanisms. *IEEE TPAMI* 35(3):541–554.

Mahendran, A., and Vedaldi, A. 2015. Understanding deep image representations by inverting them. In *CVPR*.

Martin, D.; Fowlkes, C.; Tal, D.; and Malik, J. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*.

Movahedi, V., and Elder, J. H. 2010. Design and perceptual validation of performance measures for salient object segmentation. In *CVPRW*.

Perazzi, F.; Krähenbühl, P.; Pritch, Y.; and Hornung, A. 2012. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*.

Qin, Y.; Lu, H.; Xu, Y.; and Wang, H. 2015. Saliency detection via cellular automata. In *CVPR*.

Siagian, C., and Itti, L. 2007. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE TPAMI* 29(2):300–312.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sun, J., and Ling, H. 2013. Scale and object aware image thumbnailing. *IJCV* 104(2):135–153.

Wang, L.; Lu, H.; Ruan, X.; and Yang, M.-H. 2015. Deep networks for saliency detection via local estimation and global search. In *CVPR*.

Wang, L.; Wang, L.; Lu, H.; Zhang, P.; and Ruan, X. 2016. Saliency detection with recurrent fully convolutional networks. In *ECCV*.

Wei, Y.; Liang, X.; Chen, Y.; Shen, X.; Cheng, M.-M.; Feng, J.; Zhao, Y.; and Yan, S. 2017. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE TPAMI* 39(11):2314–2320.

Xie, S., and Tu, Z. 2015. Holistically-nested edge detection. In *ICCV*.

Yan, Q.; Xu, L.; Shi, J.; and Jia, J. 2013. Hierarchical saliency detection. In *CVPR*.

Yang, C.; Zhang, L.; Lu, H.; Ruan, X.; and Yang, M.-H. 2013. Saliency detection via graph-based manifold ranking. In *CVPR*.

Zhang, P.; Wang, D.; Lu, H.; Wang, H.; and Ruan, X. 2017a. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*.

Zhang, P.; Wang, D.; Lu, H.; Wang, H.; and Yin, B. 2017b. Learning uncertain convolutional features for accurate saliency detection. In *ICCV*.

Zhao, R.; Ouyang, W.; Li, H.; and Wang, X. 2015. Saliency detection by multi-context deep learning. In *CVPR*.

Zhu, W.; Liang, S.; Wei, Y.; and Sun, J. 2014. Saliency optimization from robust background detection. In *CVPR*.