Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

R^3Net: Recurrent Residual Refinement Network for Saliency Detection

Zijun Deng[1,*], **Xiaowei Hu**[2,*], **Lei Zhu**[3,2],
Xuemiao Xu[1,†], **Jing Qin**[3], **Guoqiang Han**[1], **and Pheng-Ann Heng**[2,4]
[1] South China University of Technology,
[2] The Chinese University of Hong Kong, [3] The Hong Kong Polytechnic University,
[4] Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
zijdeng@gmail.com, {xwhu, pheng}@cse.cuhk.edu.hk,
{xuemx, csgqhan}@scut.edu.cn, {henry.zhu, harry.qin}@polyu.edu.hk

Abstract

Saliency detection is a fundamental yet challenging task in computer vision, aiming at highlighting the most visually distinctive objects in an image. We propose a novel recurrent residual refinement network (R^3Net) equipped with residual refinement blocks (*RRBs*) to more accurately detect salient regions of an input image. Our *RRBs* learn the residual between the intermediate saliency prediction and the ground truth by alternatively leveraging the low-level integrated features and the high-level integrated features of a fully convolutional network (FCN). While the low-level integrated features are capable of capturing more saliency details, the high-level integrated features can reduce non-salient regions in the intermediate prediction. Furthermore, the *RRBs* can obtain complementary saliency information of the intermediate prediction, and add the residual into the intermediate prediction to refine the saliency maps. We evaluate the proposed R^3Net on five widely-used saliency detection benchmarks by comparing it with 16 state-of-the-art saliency detectors. Experimental results show that our network outperforms our competitors in all the benchmark datasets.

1 Introduction

Saliency detection seeks to highlight the most visually distinctive objects in an image. Inferring salient objects benefits many applications, such as weakly-supervised object detection [Baisheng Lai, 2017] and visual tracking [Hong *et al.*, 2015]. However, detecting salient objects requires the semantic understanding of the whole image as well as the detailed structures of the objects. Hence, saliency detection is a fundamental yet challenging problem in computer vision.

Traditional methods employed hand-crafted visual features (e.g, color, texture, and contrast) or heuristic priors [Liu *et al.*, 2011; Yang *et al.*, 2013; Zhu *et al.*, 2014; Cheng *et al.*, 2015] to detect salient objects from the input images. These hand-crafted features and priors are incapable of capturing the high-level semantic knowledge, making the saliency prediction unsatisfactory. To improve the detection accuracy, many deep saliency networks [Li and Yu, 2015; Zhao *et al.*, 2015; Li *et al.*, 2016] have been proposed by leveraging the deep semantic features of fully convolutional neural networks (FCNs) [Long *et al.*, 2015]. Although the deep features of FCN-based methods capture more high-level semantic information over hand-crafted features, their results suffer from neglecting many fine saliency details due to the coarseness of feature maps in the deep layers of FCNs [Kong *et al.*, 2016].

More recently, several works [Liu and Han, 2016; Li and Yu, 2016; Hou *et al.*, 2017] utilized short connections to merge multi-level features of FCNs in order to simultaneously exploit high-level semantic information and low-level detailed structures for saliency detection. Although they improve predictions by using complementary information of multi-level features, those methods conduct the prediction at one stage, making the results still unsatisfactory. Wang et al. [Wang *et al.*, 2017] further improve the performance by leveraging a stage-wise refinement network to process the saliency prediction in many stages, which can progressively refine the intermediate saliency maps by absorbing low-level detail information. However, such a refinement procedure tends to introduce non-salient regions as it mainly relies on the low-level features to refine the saliency maps. In addition, it is insufficient to learn salient objects for refinement using a plain network at each stage, because it produces the refined saliency prediction from the scratch (random noise) without preserving previous prediction. On the other hand, the residual learning has exhibited better performance in many vision tasks, such as image classification [He *et al.*, 2016] and face attribute manipulation [Shen and Liu, 2017]. We employ a

*Joint first authors
†Corresponding author (xuemx@scut.edu.cn)

residual network as the starting point of our refinement network.

In this paper, we propose a novel deep refinement network to more accurately detect salient objects from the input images by leveraging advantages of the residual learning and saliency information encoded in multiple layers of an FCN. To achieve this, we first design a residual refinement block (*RRB*), which takes the deep features concatenated with the previous predicted saliency map as inputs to learn the difference (residual) between the ground truth and the previous saliency map. Then, we embed a sequence of *RRBs* in an FCN to construct our recurrent residual refinement network (R^3Net), which progressively refines the saliency maps at each recurrent step by alternatively leveraging the high-level semantic features and the low-level detailed features as the inputs, which can enhance the saliency details and suppress non-salient regions of intermediate saliency maps simultaneously. Finally, we take the saliency prediction at the last recurrent step as the final output of our network. The whole network is trained in an end-to-end manner.

To verify the effectiveness of the presented R^3Net, we evaluate it on five famous salient object detection benchmarks, and compare our results with those of 16 state-of-the-art methods. The experiments demonstrate that our model quantitatively and qualitatively outperforms other saliency detectors. Overall, we can summary the contributions of this work as follows.

- First, we design a novel residual refinement block (*RRB*) to learn the residual between the ground truth and the saliency map at each recurrent step. This learning strategy can make the network easy to train and help to learn the complementary saliency information of previous prediction for the refinement.
- We propose a recurrent residual refinement network (R^3Net) to progressively refine the saliency maps by building a sequence of *RRBs* to alternatively use the low-level features and high-level features. Such a general refinement strategy has potential to be used in other tasks such as semantic segmentation and object detection.
- Third, we evaluate our network on five famous benchmark datasets and compare it with 16 state-of-the-art saliency detection methods. Overall, our method consistently has the best performance on all the five datasets.

2 Methodology

We illustrate the architecture of our recurrent residual refinement network (R^3Net) equipped with residual refinement blocks (*RRBs*) in Figure 1. It begins by utilizing a feature extraction network to produce a set of feature maps, which contain low-level details and high-level semantic information with different scales. Then, the feature maps at shallow layers are integrated to generate the low-level integrated features (denoted as L), and the feature maps at deep layers are merged together to form the high-level integrated features (denoted as H). After that, we generate an initial saliency map from H, and then develop a set of residual refinement blocks (*RRBs*) to progressively refine the intermediate saliency maps by harnessing L and H alternatively. Meanwhile, at each recurrent step, we impose the supervision signal [Xie and Tu, 2015] to compute the loss between the ground truth and the predicted saliency map during the training process. Finally, we take the saliency map at the last recurrent step as the final output of our network. In following subsections, we first elaborate how to build the *RRBs*, and then introduce the proposed R^3Net for saliency detection.

2.1 Residual Refinement Block (RRB)

We develop an RRB at each recurrent step to correct prediction errors in the previous saliency map for refinements. The *RRB* alternatively takes the low-level integrated features (L) or the high-level integrated features (H) with the saliency map predicted at the previous step as inputs, and outputs a refined saliency map by adding the previous saliency map with a learned residual (difference between the ground truth and the previous saliency map); see orange dash boxes of Figure 1 for the first and second *RRBs*.

Formally, an *RRB* is defined as:

$$\begin{aligned} \text{residual}_j &= \mathbf{\Phi}_j(Cat(S_{j-1}, F)), \\ S_j &= S_{j-1} \oplus \text{residual}_j, \end{aligned} \tag{1}$$

where we first obtain the residual (residual_j) at j-th recurrent step by sending the concatenation (Cat) of the predicted saliency map S_{j-1} at $(j-1)$-th recurrent step and the feature maps F, to the function $\mathbf{\Phi}_j$, which consists of three convolutional layers (see Figure 1). Then, the residual_j is added with S_{j-1} in an element-wise way to compute the output S_j of our *RRB*. Note that the feature map F is alternatively set as L or H in our recurrent network (see Section 2.2 for details), and we find the unshared parameters $\mathbf{\Phi}$ of our *RRBs* at different recurrent steps show a superior performance than the shared parameters; see Table 1 for detailed comparisons.

Unlike [Wang *et al.*, 2017], which directly learns the desired underlying saliency mapping for the refinement in a plain network, our *RRB* explicitly learns to fit a residual that reflects the difference between the ground truth and the previous saliency map, since learning residual is much easier, and usually outperforms learning from plain networks in various tasks, as suggested in [He *et al.*, 2016; Shen and Liu, 2017; Xie *et al.*, 2017]. Figure 2 shows the training loss of our R^3Net (see Section 2.2 for details) with/without the residual learning technique in our *RRB*. Obviously, our residual learning can ease the optimization task with a faster convergence at early stages, and reduce the training error over directly learning underlying saliency mapping; see Table 1 in Sec. 3.3 for the quantitative comparisons of these two learning ways.

2.2 Recurrent Residual Refinement Network

To learn the salient regions in a refinement mechanism, we develop a novel deep network with a sequence of *RRBs* to gradually refine the saliency predictions. As illustrated in Figure 1, our network first applies ResNeXt [Xie *et al.*, 2017] as the feature extraction network to produce a set of feature maps with different scales. The feature maps at deep layers have large scales and are able to capture the high-level semantic information of salient objects, while the feature maps at shallow layers have small scales and can extract the fine

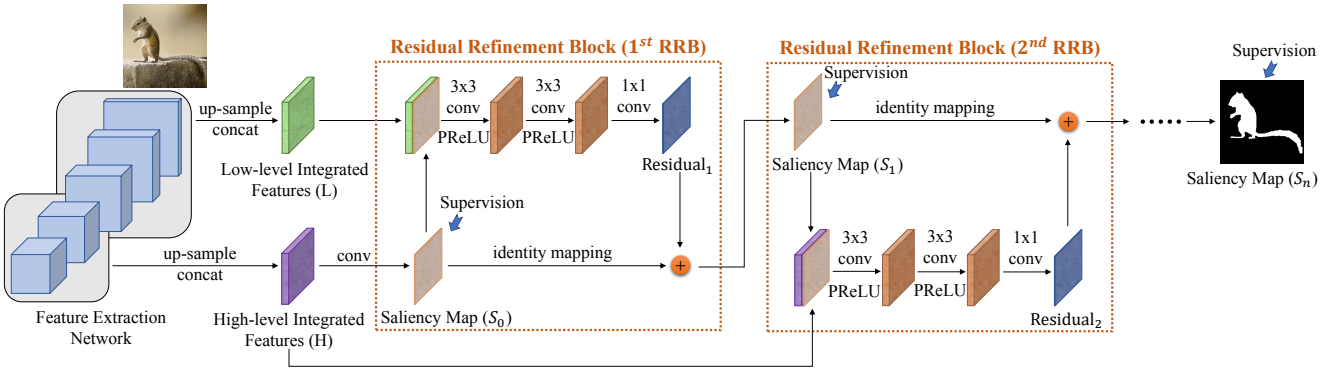


Figure 1: The schematic illustration of our R^{3}Net. We produce a set of feature maps at multiple scales for the input image. The feature maps at the first three layers are concatenated to generate the low-level integrated features (denoted as L) while features at the last two layers are concatenated to construct the high-level integrated features (denoted as H). Then, we generate the initial saliency map using H, which is recurrently refined by a sequence of residual refinement blocks (*RRBs*). Meanwhile, the supervision signal is imposed at each recurrent step.

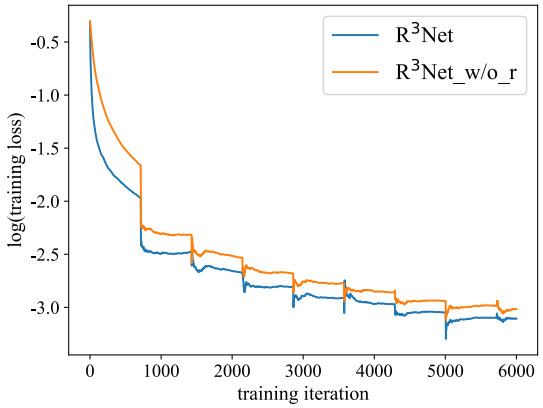


Figure 2: Training on the MSRA10K dataset. The blue line denotes R^{3}Net with residual learning while the orange line denotes R^{3}Net without residual learning, which learns the saliency map directly.

delicate structures of salient regions. Then, we produce the low-level integrated features (denoted as L) by upsampling the feature maps from the first three layers to a quarter of the size of the input image, concatenating them together, and applying a convolution operation to merge those features and reduce the feature dimensions:

$$L = f_{conv}(Cat(F_1, F_2, F_3)) \,, \tag{2}$$

where F_i is the upsampled feature maps at the i-th layer; Cat operation is to concatenate feature maps at the first three layers; f_{conv} is known as the feature fusing network, consisting of 3 convolution layers, followed by 3 PReLU activation functions [He *et al.*, 2015]. Similarly, we produce the high-level integrated features (denoted as H) by using the feature maps (F_4 and F_5) at the last two layers:

$$H = f_{conv}(Cat(F_4, F_5)) \,. \tag{3}$$

Our network first predicts an initial saliency map (denoted as S_0) from the high-level integrated features (H), which tend to capture the locations of the salient objects but neglect a lot of saliency details. Then, taking S_0 as a starting point, we develop a sequence of *RRBs* to progressively refine the saliency predictions. Specifically, since the low-level integrated features L is capable of discovering many saliency details of input images, we build the first *RRB* by setting F in Eq. 1 as the L to refine the initial saliency map (S_0) and obtain the saliency map (S_1) with more fine details. However, the low-level integrated features (L) also contain a large number of non-saliency cues, and thus simultaneously introduce non-salient regions into the S_1. Hence, we build the second *RRB* by using H to replace F in Eq. 1 to remove non-salient regions introduced by L. Since the high-level features H focus on semantic cues of the salient objects, such an operation can eliminate non-saliency details that are not located in semantic salient regions. To further improve the saliency prediction, we construct a sequence of *RRBs* by alternatively incorporating L and H several times.

In addition, we apply deep supervision mechanism [Xie and Tu, 2015] to impose the supervision signal on the predicted saliency map at each recurrent step during the training process. By adding auxiliary supervisions connected to the intermediate steps, each *RRB* is capable to learn the residual from the ground truth directly, which makes the network optimization easier [Xie and Tu, 2015]. Finally, we take the saliency map at the last recurrent step to compute the final output of our R^{3}Net network.

3 Experiments

In this section, we describe the training and testing strategies of our R^{3}Net, introduce the benchmark datasets and evaluation metrics, and report the experimental results.

3.1 Training and Testing Strategies

Loss function. As shown in Figure 1, our network can output several saliency maps, including the initial saliency map S_0, and a sequence of refined saliency maps (S_1, ..., S_n) after applying *RRB* n times. During the training process, we apply deep supervision mechanism [Xie and Tu, 2015] into impose a supervision signal (ground truth) for each saliency output, and thus we can compute the cross-entropy loss between each predicted saliency map and the ground truth (supervision). The total loss Θ of our network is defined as the

Method	ECSSD		HKU-IS		PASCAL-S		SOD		DUT-OMRON	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
R^{3}Net-0	0.918	0.049	0.900	0.044	0.831	0.101	0.816	0.128	0.769	0.079
R^{3}Net-1	0.926	0.044	0.910	0.038	0.841	0.100	0.833	0.125	0.783	0.073
R^{3}Net-2	0.931	0.043	0.911	0.038	0.844	0.104	0.836	0.127	0.787	0.073
R^{3}Net-3	0.934	0.041	0.915	**0.036**	0.847	0.100	0.841	0.123	0.794	0.066
R^{3}Net-4	0.932	0.042	0.912	0.038	0.843	0.102	0.841	0.125	0.782	0.073
R^{3}Net-5	0.933	0.042	0.913	0.037	0.845	0.100	0.841	0.122	0.791	0.069
R^{3}Net-6	**0.935**	**0.040**	**0.916**	**0.036**	0.845	0.100	**0.847**	0.124	**0.805**	**0.063**
R^{3}Net-7	0.934	**0.040**	0.914	**0.036**	**0.848**	**0.096**	0.842	**0.121**	0.804	**0.063**
R^{3}Net_w/o_r	0.931	0.042	0.910	0.039	0.839	0.103	0.839	**0.121**	0.782	0.077
R^{3}Net_w_s	0.933	0.041	0.914	0.037	0.841	0.102	0.842	0.122	0.794	0.070
R^{3}Net_LL	0.932	0.041	0.910	0.038	0.844	0.100	0.839	0.125	0.778	0.080
R^{3}Net_HH	0.926	0.046	0.902	0.042	0.836	0.101	0.819	0.128	0.786	0.071
R^{3}Net-D	0.928	0.046	0.907	0.042	0.829	0.112	**0.847**	0.127	0.793	0.067
R^{3}Net-V	0.913	0.049	0.891	0.047	0.814	0.105	0.818	**0.121**	0.746	0.089

Table 1: The F-measure and MAE of different settings on five saliency detection datasets for ablation analysis.

summation of the loss on all predicted saliency maps:

$$\Theta = w_0\mathcal{Y}_0 + \sum_{i=1}^{n} w_i\mathcal{Y}_i, \tag{4}$$

where w_0 and $\mathcal{Y}_0$ are the weight and loss in our initial saliency prediction; w_i and $\mathcal{Y}_i$ denote the weight and loss of the prediction at i-th recurrent step; n is the number of recurrent steps employed in our method. In our experiment, we empirically set all the weights (including w_0 and w_i) as 1, and set the hyper-parameter n as 6 by balancing the time performance and the detection accuracy (see Section 3.3 for details).

Training parameters. In order to accelerate the training process and reduce the over-fitting issue, we use the well-trained ResNeXt network on ImageNet [Xie *et al.*, 2017] to initialize parameters of feature extraction network (see Figure 1), while other layers are randomly initialized from a Gaussian distribution. We use the stochastic gradient descent (SGD) to train the network with the momentum of 0.9 and the weight decay of 0.0005, set the basic leaning rate as 0.001, adjust the learning rate by the "poly" policy [Liu *et al.*, 2015] with the power of 0.9, and stop the training procedure after 6k iterations. The R^{3}Net is trained on the MSRA10K dataset [Cheng *et al.*, 2015], which is widely used for training the saliency models [Lee *et al.*, 2016; Zhang *et al.*, 2017a]. Images in this dataset are randomly rotated, cropped and horizontally flipped for data augmentation. Our R^{3}Net is trained on a single GPU with a mini-batch size of 14, and it takes only 80 minutes to train the network.

Inference. In the testing stage, for each input image, our R^{3}Net can predict a saliency map at each recurrent step. Our final result is obtained by upsampling the prediction at the last recurrent step to the size of the input image, and then applying the fully connected conditional random field (CRF) [Krähenbühl and Koltun, 2011] to enhance the spatial coherence of the saliency maps.

3.2 Datasets and Evaluation Metrics

After training the R^{3}Net on the MSRA10K dataset [Cheng *et al.*, 2015] (containing 10,000 images), we perform various experiments to evaluate the proposed network on five widely-used saliency benchmark datasets, including ECSSD [Yan *et al.*, 2013] with $1,000$ images, HKU-IS [Li and Yu, 2015] with $4,447$ images, PASCAL-S [Zhang *et al.*, 2017a] with 850 images, SOD [Hou *et al.*, 2017] with 300 images, and DUT-OMRON [Yang *et al.*, 2013] with $5,168$ images. Please refer to the [Hou *et al.*, 2017] for the detail descriptions of these saliency benchmark datasets.

We use two metrics to quantitatively compare our method with our rivals: F-measure (F_β) and mean absolute error (MAE) (see [Hou *et al.*, 2017] for their definitions). A better saliency detector shall have a larger F-measure and a smaller MAE. To do fair comparisons, we apply the implementations of [Hou *et al.*, 2017] to compute the F-measure and MAE for all the compared methods.

3.3 Ablation Analysis

We first perform ablation experiments on the 5 benchmarks to evaluate the effectiveness of our R^{3}Net. First, we show the results of our R^{3}Net with different recurrent steps. Second, we perform a comparison with "R^{3}Net_w/o_r", which has a similar structure with our R^{3}Net but refines the saliency map (six times) without residual learning. Third, we modify our R^{3}Net by using the shared parameters among all the $RRBs$ and denote the new one as "R^{3}Net_w_s" for a comparison. Then, we also compare with "R^{3}Net-LL" using only L as the features F of Eq. 1 in all the recurrent steps, and "R^{3}Net-HH" using only H as the features F of Eq. 1 in all the recurrent steps to verify the effectiveness of alternatively using L and H during the refinement procedure. Moreover, we compare another two models (denoted as "R^{3}Net-D" and "R^{3}Net-V"), which use the DenseNet [Huang *et al.*, 2017] (161 layers) and the VGG-Net [Simonyan and Zisserman, 2015] (16 layers) respectively, instead of the our ResNeXt.

Table 1 summaries the quantitative results of these different settings. From the results, we has the following observations: (1) R^{3}Net-0 to R^{3}Net-7 in Table 1 are the initial prediction

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 *et al.*, 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
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
SRM [Wang *et al.*, 2017]	0.917	0.056	0.906	0.046	0.844	0.087	0.843	0.126	0.769	0.069
NLDF+	0.920	0.063	0.907	0.055	0.837	0.108	0.830	0.138	0.779	0.094
Amulet+	0.925	0.055	0.900	0.053	0.832	0.109	0.823	0.136	0.783	0.082
DSS+	0.928	0.044	0.909	0.038	0.843	0.101	0.839	0.125	0.779	0.071
SRM+	0.927	0.052	0.907	0.048	0.844	0.089	0.838	0.125	0.786	0.071
R^{3}Net (ours)	0.935	0.040	0.916	0.036	0.845	0.100	0.847	0.124	0.805	0.063

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

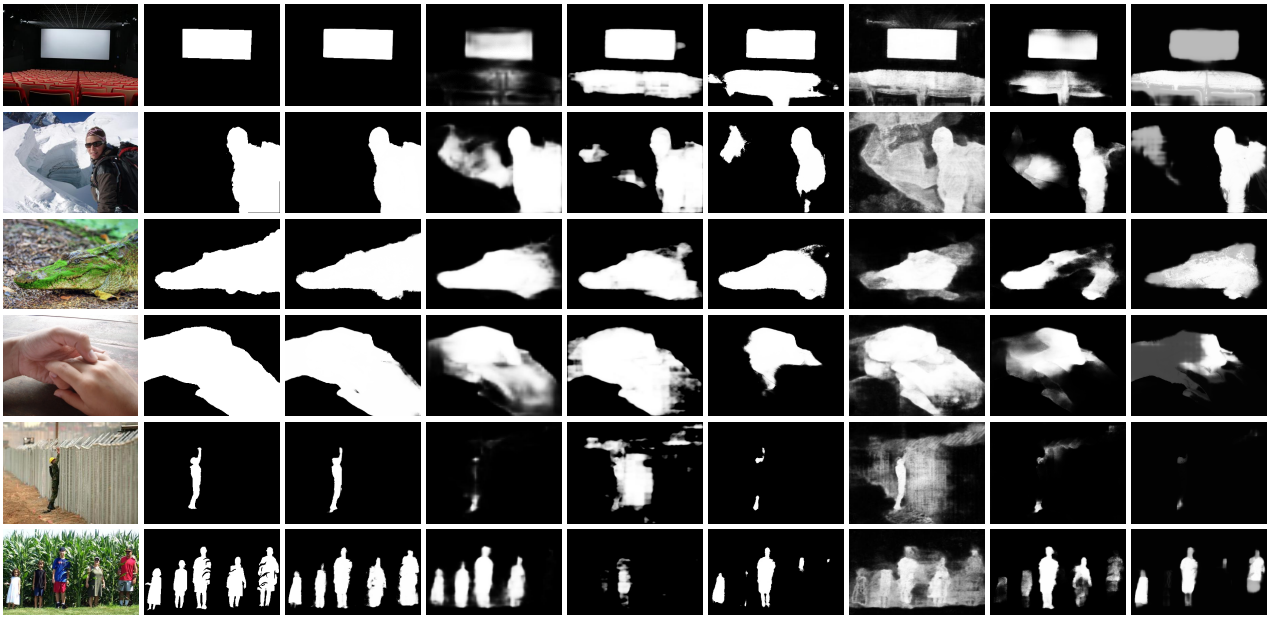
and our refined results from the first to seventh recurrent step. It is observed that our recurrent mechanism significantly outperforms the initial prediction (denoted as "R^{3}Net-0"), and the saliency detection accuracy consistently increases in the first four iterations, and then becomes stable from the fifth to seventh iteration. To the end, we empirically set the total recurrent step as six by balancing the performance and time complexity. (2) Comparing our "R^{3}Net" with "R^{3}Net_w/o_r", we can find that our model with residual learning is superior to the model without residual learning, demonstrating the effectiveness of the proposed RRB. (3) The comparison between our "R^{3}Net" and "R^{3}Net_w_s" demonstrates that our method with separated parameters in different *RRBs* has a superior performance. (4) Our model has a better performance over "R^{3}Net-LL", demonstrating that H can help to suppress non-saliency regions caused by L. Moreover, the superior performance of our method over the "R^{3}Net-HH" indicates that L can complement more saliency details lacked in H. (5) Comparing "R^{3}Net" with "R^{3}Net-D" and "R^{3}Net-V", we can find that our model equipped with ResNeXt has a better performance, showing that the ResNeXt [Xie *et al.*, 2017] extracts more powerful features than VGG-Net and DenseNet.

3.4 Comparison with the State-of-the-arts

We further compare the results of our method with those of 16 state-of-the-art saliency detectors (see the first column of Table 2 for compared saliency detectors). Among them, MR [Yang *et al.*, 2013], wCtr* [Zhu *et al.*, 2014] and BSCA [Qin *et al.*, 2015] use hand-crafted features to differentiate the salient objects from background while others are based on the deep learning framework to learn the features for saliency detection. In order to conduct a fair comparison, we obtain saliency detection results of compared methods by using either the saliency maps provided by the authors, or their implementations with recommended parameter settings.

Quantitative comparison. Table 2 summaries the comparison results in terms of F-measure and MAE. Our method (R^{3}Net) consistently outperforms others on almost all the five datasets with respect to both two metrics, demonstrating the superior performance of our method on detecting salient objects. Especially, our method outperforms the previous state-of-the-arts by a significant margin on the "DUT-OMRON" dataset, where the images are with complicated salient regions. It shows that our method is more powerful to deal with the challenging images, which is further manifested in the following visual comparisons (see Figure 3). Note that our network is trained on the MSRA10K dataset, but it still has a superior performance over others (e.g. RFCN and MDF) that are directly trained on PASCAL-S or HKU-IS, demonstrating a good generalization capability of our R^{3}Net, which is vital for the saliency detection.

For methods based on the deep neural networks, the training set and the feature extract network are important, because a good training set will provide a large number of representative training data and provide more knowledge to the deep neural network, and a strong feature extract network will obtain more powerful features for discovering the salient regions. However, the recent saliency detectors based on the deep neural networks use different kinds of training sets and feature extract networks. For a fair comparison, we retrain the models (NLDF [Luo *et al.*, 2017], DSS [Hou *et al.*, 2017], Amulet [Zhang *et al.*, 2017b], and SRM [Wang *et al.*, 2017]) by using the same basic network (ResNeXt) and the same training set (MSRA10K) as our method. Table 2 also reports the comparison results, where these retrained models are de-



(a) Inputs (b) GT (c) Ours (d) SRM (e) Amulet (f) DSS (g) UCF (h) NLDF (i) DCL

Figure 3: Visual comparison of saliency maps produced by different saliency detectors. Apparently, our method produces more accurate saliency maps than others, and our results are more consistent with the ground truths (denoted as 'GT').

noted as "XX+". By using the powerful basic network and the MSRA10K dataset, these models show better performances than the original ones, but our method still outperforms these new models by a significant margin.

Visual comparison. We also visually compare our method with our rivals on different input images; see Figure 3. From these results, we can observe that other methods tend to include non-saliency backgrounds (see (d)-(i) of the first two rows) or lost many saliency details (see (d)-(i) of the third and fourth rows), while our R^{3}Net produces more accurate results of detecting salient objects, and our results are more consistent with the ground truths (see (b) and (c)). In addition, for those challenging images with small salient objects (the fifth row in Figure 3) and multiple objects (the last row in Figure 3), our method also predicts more precise saliency maps than others, which further indicates the effectiveness and robustness of the presented R^{3}Net. The code, trained model and more results are publicly available at `https://github.com/zijundeng/R3Net`.

4 Conclusion

This paper presents a novel refinement network equipped with a sequence of residual refinement blocks (*RRBs*) for single-image saliency detection. Our key idea is leveraging *RRBs* to recurrently learn the difference (called "residual" in this work) between the coarse saliency map and the ground truth by alternatively harnessing the low-level features (L) and high-level features (H) of an FCN. Learning the residual counterpart in our *RRB* module can make the network easy to be optimized, and obtain the complementary saliency information of the intermediate results to refine the saliency prediction maps. Furthermore, our *RRBs* employ the low-level features and the high-level features to alternatively refine the saliency maps by compensating saliency details and suppressing non-salient regions. We test our R^{3}Net on 5 benchmark datasets, and the experimental results show that our method consistently outperforms the other 16 state-of-the-art methods. The proposed refinement scheme is general enough to be applied to other important computer vision tasks, such as object detection and instance-aware saliency detection.

Acknowledgments

The work is supported by the Shenzhen Science and Technology Program (JCYJ20170413162256793), the Research Grants Council of the Hong Kong Special Administrative Region (Project no. CUHK 14225616), the Hong Kong Innovation and Technology Commission (Project no. ITS/304/16), NSFC (Grant No. 61772206, U1611461, 61472145), Special Fund of Science and Technology Research and Development on Application from Guangdong Province (Grant No. 2016B010124011), Guangdong High-level Personnel of Special Support Program (Grant No. 2016TQ03X319), and the Guangdong Natural Science Foundation (Grant No. 2017A030311027). Xiaowei Hu is funded by the Hong Kong Ph.D. Fellowship.

References

[Baisheng Lai, 2017] Xiaojin Gong Baisheng Lai. Saliency guided end-to-end learning for weakly supervised object detection. In *IJCAI*, 2017.

[Cheng *et al.*, 2015] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global

contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *CVPR*, 2015.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Hong *et al.*, 2015] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*, 2015.

[Hou *et al.*, 2017] Qibin Hou, Ming-Ming Cheng, Xiao-Wei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. In *CVPR*, 2017.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, K.Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017.

[Kong *et al.*, 2016] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *CVPR*, 2016.

[Krähenbühl and Koltun, 2011] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011.

[Lee *et al.*, 2016] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. Deep saliency with encoded low level distance map and high level features. In *CVPR*, 2016.

[Li and Yu, 2015] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *CVPR*, 2015.

[Li and Yu, 2016] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *CVPR*, 2016.

[Li *et al.*, 2016] Xi Li, Liming Zhao, Lina Wei, Ming-Hsuan Yang, Fei Wu, Yueting Zhuang, Haibin Ling, and Jingdong Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE Transactions on Image Processing*, 25(8):3919–3930, 2016.

[Liu and Han, 2016] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, 2016.

[Liu *et al.*, 2011] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):353–367, 2011.

[Liu *et al.*, 2015] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[Long *et al.*, 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[Luo *et al.*, 2017] Zhiming Luo, Akshaya Mishra, Andrew Achkar, Justin Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. In *CVPR*, 2017.

[Qin *et al.*, 2015] Yao Qin, Huchuan Lu, Yiqun Xu, and He Wang. Saliency detection via cellular automata. In *CVPR*, 2015.

[Shen and Liu, 2017] Wei Shen and Rujie Liu. Learning residual images for face attribute manipulation. In *CVPR*, 2017.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[Wang *et al.*, 2015] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, 2015.

[Wang *et al.*, 2016] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, 2016.

[Wang *et al.*, 2017] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *CVPR*, 2017.

[Xie and Tu, 2015] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, 2015.

[Xie *et al.*, 2017] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[Yan *et al.*, 2013] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, 2013.

[Yang *et al.*, 2013] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013.

[Zhang *et al.*, 2017a] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, 2017.

[Zhang *et al.*, 2017b] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency detection. In *ICCV*, 2017.

[Zhao *et al.*, 2015] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *CVPR*, 2015.

[Zhu *et al.*, 2014] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *CVPR*, 2014.