Stacked Cross Refinement Network for Edge-Aware Salient Object Detection

Zhe Wu[1,2], Li Su*[1,2,3], and Qingming Huang[1,2,3,4]

[1]School of Computer Science and Technology, University of Chinese Academy of Sciences (UCAS), Beijing, China
[2]Key Lab of Big Data Mining and Knowledge Management, UCAS, Beijing, China
[3]Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, China
[4]Peng Cheng Laboratory, ShenZhen, China
wuzhe14@mails.ucas.ac.cn, {suli, qmhuang}@usas.ac.cn

Abstract

Salient object detection is a fundamental computer vision task. The majority of existing algorithms focus on aggregating multi-level features of pre-trained convolutional neural networks. Moreover, some researchers attempt to utilize edge information for auxiliary training. However, existing edge-aware models design unidirectional frameworks which only use edge features to improve the segmentation features. Motivated by the logical interrelations between binary segmentation and edge maps, we propose a novel Stacked Cross Refinement Network (SCRN) for salient object detection in this paper. Our framework aims to simultaneously refine multi-level features of salient object detection and edge detection by stacking Cross Refinement Unit (CRU). According to the logical interrelations, the CRU designs two direction-specific integration operations, and bidirectionally passes messages between the two tasks. Incorporating the refined edge-preserving features with the typical U-Net, our model detects salient objects accurately. Extensive experiments conducted on six benchmark datasets demonstrate that our method outperforms existing state-of-the-art algorithms in both accuracy and efficiency. Besides, the attribute-based performance on the SOC dataset show that the proposed model ranks first in the majority of challenging scenes. Code can be found at `https://github.com/wuzhe71/SCAN`.

1. Introduction

Salient object detection [1, 5, 10, 11, 17] aims to detect and segment the most attractive objects in images or videos. In the past decades, hundreds of traditional methods have been developed to address the task of salient object detection and widely applied as a pre-processing procedure in other computer vision tasks [2, 3].

Recently, convolutional neural networks (CNNs) greatly promote the research of computer vision. Early deep salient object detection models [14, 18, 19, 27, 38] utilize classification network to determine each region of an image is salient or not. These models generate better results than the traditional models along with expensive computation overhead. Then fully convolutional networks (FCNs) [23] based approaches [4, 15, 22, 24, 29, 30, 31, 35, 36, 37] further boost the development of salient object detection. These works have achieved state-of-the-art performance via a designing reasonable decoders to extract discriminative multi-level features and aggregate them together. Besides, researchers also attempt to leverage the complementary information between the two tasks of salient object detection and edge detection. Some strategies use edge labels to improve the training procedure of segmentation networks: adding auxiliary boundary loss at the end of the segmentation network [24], designing unidirectional frameworks [12, 39] which only use edge information to improve the representation ability of segmentation feature. Though the previous works have demonstrated that fusing edge features is propitious to generate more accurate segmentation maps, they may confront the problem of inaccurate edge features. And the edge information has not been fully exploited in existing edge-aware frameworks.

In this paper, we investigate the interrelations between binary segmentation and edge maps, and figure out that boundary region in the edge map is the proper subset of object region in the corresponding segmentation map. Inspired by this observation, we propose a novel edge-aware

*Li Su (suli@ucas.ac.cn) is the corresponding author.

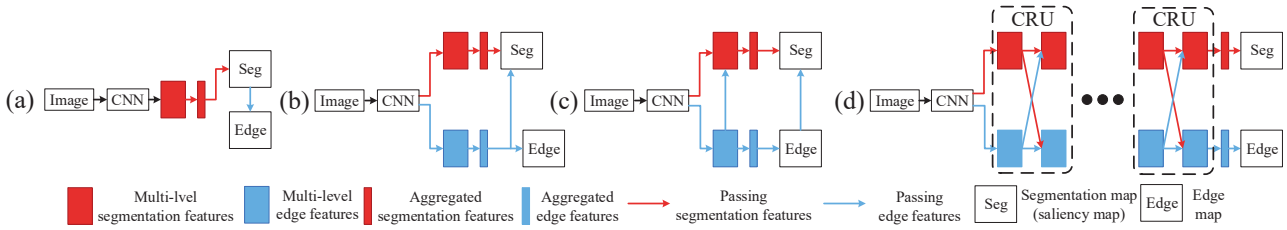


Figure 1: *(a) Auxiliary edge loss in [24], (b) Unidirectional fusing aggregated segmentation and edge features [39], (c) Unidirectional fusing multi-level edge features and predicted edge map [12], (d) The proposed framework. Our model consecutively stacks multiple cross refinement units in an end-to-end manner, which bidirectionally refine multi-level features of the two tasks by designing two direction-specific integration operations.*

salient object detection approach, named Stacked Cross Refinement Network (SCRN), which bidirectionally passes messages between the two tasks and simultaneously refines multi-level edge and segmentation features. We first extract two separate sets of multi-level deep features from a shared backbone network, which are utilized to construct two parallel decoders: one is for edge detection, and the other is for salient object detection. We extend the logical interrelations from binary map level to feature level, and propose a Cross Refinement Unit (CRU) which contains two different direction-specific integration operations. Through consecutively stacking multiple CRUs in an end-to-end manner, the multi-level features of the two task are gradually improved. In conjunction with two independent U-Net structures [26], our framework detects both the salient objects and edges and outperforms state-of-the-art algorithms in both accuracy and efficiency.

In summary, our contributions are concluded as follows:

- We propose an effective Cross Refinement Unit (CRU), which bidirectionally passes messages between the two tasks of salient object detection and edge detection. In the CRU, we design two direction-specific integration operations to simultaneously refine multi-level features of the two tasks.

- We propose a novel framework for salient object detection, named Stacked Cross Refinement Network (SCRN), which stacks multiple CRUs to gradually improve the two sets of multi-level features. Incorporated with the typical U-Net structures, our framework segments salient objects from images precisely.

- Extensive experiments conducted on six traditional benchmark datasets show that our model outperforms state-of-the-art models in all six metrics. In addition, we also demonstrate that our model ranks first in the majority of challenging scenes of the SOC dataset.

2. Related Work

In the past two decades, hundreds of hand-crafted feature based traditional approaches have been proposed for salient object detection. More details of them can be found in [2, 3]. Here we mainly discuss FCN-based deep aggregation models and edge-aware deep models.

Deep aggregation models. Based on the successful FCN [23] for semantic segmentation, a large amount of FCN-based salient object detection models have been developed for salient object detection. Hou *et al.* [15] introduce short connections to the skip-layer structures within the HED [32] architecture. Zhang *et al.* [36] integrate multi-level feature maps into multiple resolutions, predict the saliency map in each resolution and fuse them to generate the final saliency map. Deng *et al.* [7] 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 FCN. In [35], the work extracts context-aware multi-level features and then utilizes a bidirectional gated structure to pass messages between them. Liu *et al.* [22] leverage global and local pixel-wise contextual attention network to capture global and local context information. Zhang *et al.* propose a novel attention guided network which selectively integrates multi-level contextual information in a progressive manner. Chen *et al.* [4] propose a reverse attention network, which eventually explores the missing object parts and details by erasing the current predicted salient regions from side-output features.

Edge-Aware models. In addition to training the model only with segmentation labels, researchers also attempt to use the edge labels. In [24], the work uses an extra IOU-based edge loss to directly optimize edges of predicted saliency maps. In [39], the authors integrate multi-level convolutional features recurrently with the guidance of object boundary information. Guan *et al.* [12] use the fine-tuned HED to detect edges, which are then served as the complementary in-

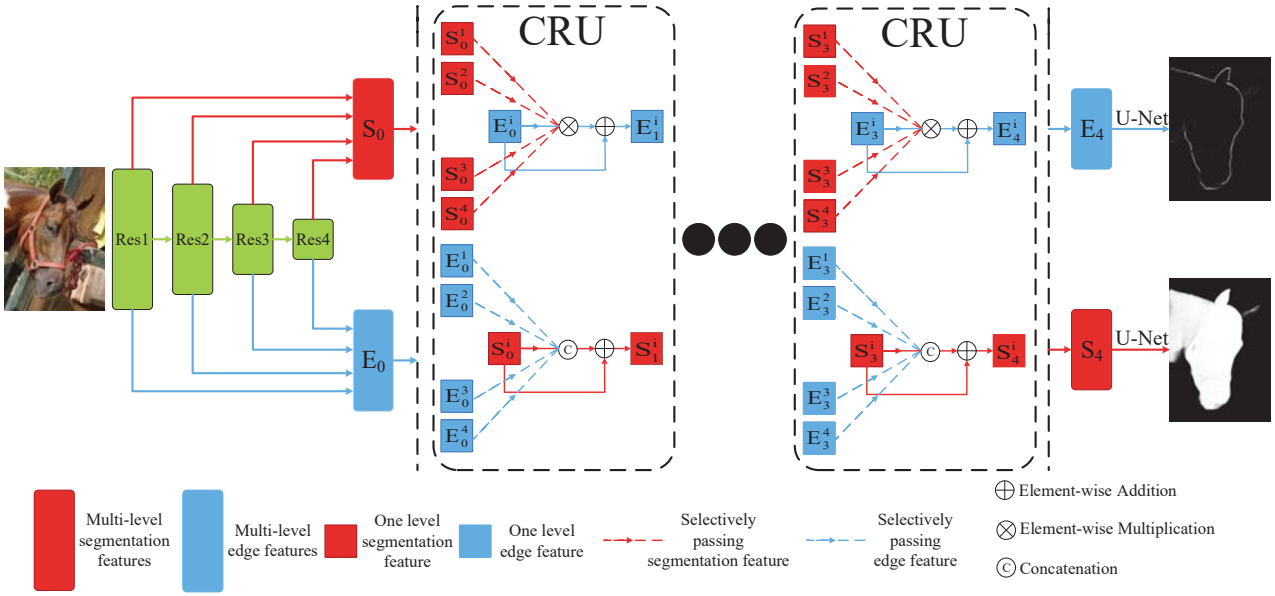


Figure 2: *The framework of the proposed SCRN model. We first extract two separate multi-level features for salient object detection and edge detection. Then we utilize stacked CRUs to refine these features by two different direction-specific operations. In each CRU, we use a selective mechanism. When refining one layer feature of one task, the lower-level features of the other task are ignored. In the proposed model, we stack four CRUs in an end-to-end manner. Incorporating with the typical U-Net structures, we finally generate the segmentation and edge maps at the same time.*

formation and integrated with the saliency detection stream to depict continuous boundary for salient objects. These approaches simply use edge information to improve the accuracy of saliency maps and have not paid enough attention on improving the edge features. In this paper, we investigate the logical interrelations between binary segmentation and edge maps, which are then promoted to bidirectionally refine multi-level features of the two tasks. Fig. 1 shows different frameworks of edge-aware deep salient object detection models.

3. Methodology

In this section, we first explore the logical interrelations between the binary segmentation and edge maps. Then we promote the interrelations to integrate multi-level features of salient object detection and edge detection, and propose a novel Cross Refinement Unit (CRU). These features are gradually refined by stacking multiple CRUs in an end-to-end manner. Combining with the typical U-Net structures, we obtain accurate segmentation maps. The overview of the proposed model is shown in Fig. 2.

3.1. Interrelations of Edge and Segmentation

Salient object detection is a pixel-wise binary classification problem. We define a ground truth segmentation map $M_s = \{M_s^p, p \in (0,1), p = 1, ..., N\}$, where p indicates one pixel of an image and N is the number of pixels in the image. Then the corresponding edge map can be defined as M_e. For an image, M_s highlights the whole salient objects and M_e only highlights the edges of salient objects. Therefore, the edge region in M_e is the proper subset of the object region in M_s. This leads to that the logical interrelations can be represented by:

$$\begin{cases} M_s \quad \wedge \quad M_e = M_e \\ M_s \quad \vee \quad M_e = M_s, \end{cases} \tag{1}$$

where $\wedge$ is the Boolean AND operation and $\vee$ is the Boolean OR operation. In this paper, these logical interrelations are extended to refine the multi-level features of the two tasks.

3.2. Network Architecture

3.2.1 Feature Extraction

Referring to previous works [22, 30, 31], our model is based on the ResNet50 [13]. We obtain four level features from the four residual blocks of the backbone network, which are defined as $F = \{F^i, i = 1, 2, 3, 4\}$. Given an image with size of $H \times W$, the size of each feature is $\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times C$. C is the channel number of a feature and is equal to 2^{i+7}. For each level, we use two 1×1 convolutional layers to extract two features with 32 channel number for the two tasks. Then we use $S = \{S_n^i, i = 1, 2, 3, 4\}$ and $E = \{E_n^i, i = 1, 2, 3, 4\}$ to represent the multi-level features of

salient object detection and edge detection respectively. In the proposed model, we stack multiple CRUs and use n to indicate which CRU a feature belongs to. For the features which have not been refined, n is equal to 0.

3.2.2 Cross Refinement Unit

According to the logical interrelations between binary segmentation and edge maps, we propose the CRU to improve the multi-level features of the two tasks. Through stacking multiple CRUs in an end-to-end manner, the two sets of features are gradually refined. More specifically, the input of one CRU is equal to the output of the previous CRU. The features (S_n^i and E_n^i) in nth CRU and i level are calculated by integrating the features (S_{n-1}, E_{n-1}). Therefore, we design two direction-specific integration operations in a CRU. The general formulations of these two operations are defined as:

$$\begin{aligned} S_n^i &= S_{n-1}^i + f(S_{n-1}^i, E_{n-1}) \\ E_n^i &= E_{n-1}^i + g(E_{n-1}^i, S_{n-1}) \end{aligned} \tag{2}$$

where f and g are designed to refine S_{n-1}^i/E_{n-1}^i with E_{n-1}/S_{n-1} respectively. In addition, they are combined with the successful residual learning [13] to generate more discriminative features. The detailed forms of the two functions are designed according to the two different logical interrelations. Especially, two problems exist in designing the two functions. One problem is how to integrate features in each direction. The other problem is how many level features of one task should be selected to improve one level feature of the other task. To address these two problems, we progressively introduce three styles of CRUs in the following.

Point-to-Point style. For each level feature of one task, we can directly use the corresponding level feature of the other task to refine it, namely only using E_{n-1}^i and S_{n-1}^i to refine each other. This is called point-to-point style of the CRU. When using segmentation features to refine edge features, we use the feature level multiplication to approximate the Boolean AND operation. In this case, the function g in the point-to-point style is defined as:

$$g = Conv(E_{n-1}^i \otimes S_{n-1}^i), \tag{3}$$

where $\otimes$ is element-wise product and $Conv$ represents a 3×3 convolutional layer with 32 output channel.

In contrast, the Boolean OR operation cannot be directly implemented in feature level and it is non-differentiable as well. Hence we use an alternative strategy to enhance the segmentation features by integrating the edge features. The function f in the point-to-point style is formulated as:

$$f = Conv(Cat(S_{n-1}^i, E_{n-1}^i)), \tag{4}$$

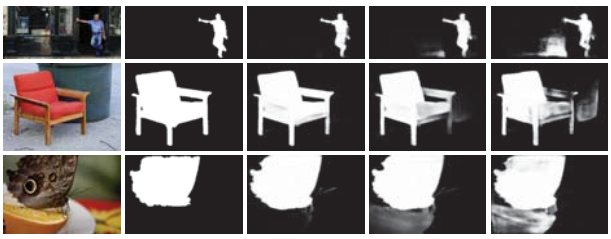


Figure 3: *Visual comparisons of different styles of the proposed cross refinement model. PPS: point-to-point style, SPS: set-to-point style, SSPS: selective set-to-point style.*

where Cat is the concatenation operation among channel axis, and $Conv$ also represents a 3×3 convolutional layer with 32 output channel like the $Conv$ of g. But the difference is that the input channel of the convolutional layer is 64. For all versions of f and g, we follow a rule that the channel number remains at 32 after applying each function.

After applying the operations defined in Eq. 3 and Eq. 4, the features of two tasks will become clearer and more discriminative. On the one hand, the segmentation features contain complete edge information and can be used to improve the edge features by the multiplication operation. On the other hand, the distractors in the segmentation features can be suppressed by concatenating the edge features.

Set-to-Point Style. CNNs extract multi-level features from an input image, which represent different information. More specifically, the high-level features always represent semantic information (*e.g.* face), and low-level features focus on class-agnostic spatial information (*e.g.* edge, texture). In order to encode more information in feature refinement, we further propose a set-to-point style, which refines each layer feature of one task by integrating all-level features of the other task. For example, E_{n-1}^i is refined by four-level segmentation features $\{S_{n-1}^k, k = 1, ..., 4\}$. In this case, the function g is defined as:

$$g = Conv(E_{n-1}^i \otimes \prod_{k=1}^{4} CU(S_{n-1}^k)), \tag{5}$$

where CU is a scale transformation operation along with a 1×1 convolutional layer with 32 output channel number. CU aims to ensure the spatial size consistence between segmentation and edge features. When $k > i$, CU uses bilinear upsampling operation with scale factor 2^{k-i}. When $k < i$, CU uses the bilinear downsampling operation with scale factor 2^{i-k}. When $k = i$, CU uses the identity function. Besides, $\otimes$ is element-wise multiplication, and $\prod$ means element-wise multiplying all-level segmentation features.

Correspondingly, the function f of this style is defined as:

$$f = Conv(Cat(S^i_{n-1}, Cat^4_{k=1}[CU(E^k_{n-1})])), \quad (6)$$

where $Cat[*]$ means concatenating all-level edge features of the $(n-1)$th CRU. In this concatenation fashion, $Conv$ has 160 input channel number. Compared to the point-to-point style, the segmentation and edge features are further improved via fusing more information.

Selective Set-to-Point Style. When CNNs extract multi-level features from an input image, the distractors in features are gradually suppressed as CNNs go deeper. The lower-level features contain many spatial details of background and the higher-level features focus more on discriminative regions. Since there are more distractors in lower-level features, we improve the original set-to-point style to a selective version, and the function g is updated to:

$$g = Conv(E^i_{n-1} \otimes \prod_{k=i}^{4} CU(S^k_{n-1})) \quad (7)$$

And the function f is defined as:

$$f = Conv(Cat(S^i_{n-1}, Cat^4_{k=i}[CU(E^k_{n-1})])) \quad (8)$$

In this selective version, for one level feature of one task, the lower-level features of the other task are ignored in feature refinement. For example, the lowest-level edge feature E^1_{n-1} is still refined by four-level segmentation features $\{S^k_{n-1}, k = 1, ..., 4\}$, but the top-most edge feature E^4_{n-1} is only refined by S^4_{n-1}. Furthermore, the selective style costs less computation overhead than the original set-to-point style. Moreover, the performance increases because less distractors have been introduced in feature integration. Some visual examples of the three different styles of CRUs are shown in Fig. 3.

With stacked multiple CRUs, we obtain the improved multi-level features of the two tasks. Then we use two typical U-Net structures to fuse them respectively and generate two $\frac{H}{4} \times \frac{W}{4} \times C$ features by upsampling and concatenating features from high-level to low-level. Each upsampling and concatenation operation is followed by a convolutional layer as the CRU. With two additional 1×1 convolutional layers, two upsampling operations (scale factor 4) and the sigmoid function, we get the predicted segmentation and edge maps (P_s, P_e). Given the ground truth segmentation map GT_s, we infer the edge label GT_e as [24]. Then the loss of the proposed framework is formulated as:

$$L = L_{ce}(P_s, GT_s) + L_{ce}(P_e, GT_e), \quad (9)$$

where L_{ce} is the standard cross entropy loss:

$$L_{ce} = -\sum_{j=1}^{N} \sum_{c \in \{0,1\}} \delta(GT^j = c) \log p(P^j = c|\theta), \quad (10)$$

where δ is the indicator function, $\theta \in \{\theta_s, \theta_e\}$ are parameters corresponding to the maps $P \in \{P_s, P_e\}$.

4. Experiments

4.1. Implementation Details

The proposed model is implemented on the public Pytorch toolbox, and we run it on a PC with 3.6Ghz CPU, 16GB RAM and a GTX Titan X GPU. We train the proposed model on DUTS [28] training set as the previous works [22, 30, 31]. For the convolutional layers in decoders, their weights are initialized by normal distribution with 0.01 standard deviation and zero mean value. And each convolutional layer is followed by a batch norm layer [16] except the last two 1×1 convolutional layers. For data augmentation, we use multi scale input images with sizes of $[0.75, 1, 1.25]$. We use the stochastic gradient descent to train the network with momentum of 0.9 and weight decay of 0.0005. The batch size is set as 8 and the input image is resized to 352×352. It takes 30 epochs for the whole training procedure. The learning rate is set as 0.002 and decreased by 10% at 20 epochs. We will make the code available in the future.

4.2. Datasets and Evaluation Metrics

We evaluate the proposed approach on six traditional benchmark datasets: ECSSD [33], PASCAL-S [21], DUTS [28], HKU-IS [19], DUT-OMRON [34] and THUR15K [6]. In addition, we evaluate attribute-based performance on the challenging SOC dataset [8]. Six metrics are mainly used to evaluate our model and existing state-of-the-art algorithms. The first two metrics are mean absolute error (MAE), maximum F-measure (maxF) (see [3] for their definitions), and both are widely adopted in previous models [4, 15, 20, 22, 35]. Then weighted F-measure (F^ω_β) [25] and structural similarity measure ($S_\alpha, \alpha = 0.5$) [9] are also adopted to evaluate saliency maps. Besides, we also draw the precision-recall (PR) and F-measure curves.

4.3. Ablation Analysis

In this section, we carefully analyze the variants of our model. We set a baseline which does not use the proposed CRU and contains two separate branches for the two tasks. We select two benchmark datasets (DUT-OMRON [34], and DUTS-TEST [28]) and two metrics (F^ω_β and S_α) for ablation analysis.

The Number of CRUs. We find that only using one CRU does not obviously improve the performance. This may be because one CRU has limited effect on enlarging receptive field. Therefore, we test the proposed model with even stacking number $(2, 4, 6, 8)$, and the results are shown in Table 1, which exhibits that the model with

Table 1: *Comparisons of the proposed model with different number of CRUs. Each variant is named as SCRN$_k$, $k = 2, 4, 6, 8$. Specially, baseline ($k = 0$) means two separate branches without using the CRU.*

Model	FPS	DUT-OMRON		DUTS-TEST	
		F_β^ω	S_α	F_β^ω	S_α
Baseline	**125**	0.667	0.810	0.752	0.861
SCRN$_2$	78	0.699	0.827	0.786	0.879
SCRN$_4$	52	**0.720**	**0.837**	0.803	0.885
SCRN$_6$	41	0.716	0.832	**0.807**	0.885
SCRN$_8$	34	0.714	0.831	**0.807**	**0.887**

Table 2: *Comparisons of the proposed bidirectional model with its two unidirectional variants.*

Method	Direction	DUT-OMRON		DUTS-TEST	
		F_β^ω	S_α	F_β^ω	S_α
Baseline	-	0.667	0.810	0.752	0.861
SCRN$_4$	$S \rightarrow E$	0.688	0.819	0.773	0.868
	$S \leftarrow E$	0.683	0.814	0.772	0.866
	$S \leftrightarrow E$	**0.720**	**0.837**	**0.803**	**0.885**

two CRUs (SCRN$_2$) obviously outperforms the baseline. When the number of CRUs is bigger than 4, the performance grows slowly in DUTS-TEST dataset and decreases in DUT-OMRON dataset. This is because adding too many CRUs lead to over-fitting due to introducing too many parameters. In conclusion, we consider both the performance and efficiency synthetically, and select the version of four CRUs (SCRN$_4$) as the final model.

Bidirectional model VS Unidirectional variants. In the proposed bidirectional model (masked as $S \leftrightarrow E$ here), messages are passed between the multi-level features of the two tasks. We compare it with two unidirectional variants: only using edge features to refine segmentation features ($S \leftarrow E$), and only using segmentation features to refine edge features ($S \rightarrow E$). Similar to the proposed bidirectional model, the two variants also have four unidirectional refinement units. The results are shown in Table 2. We can find that both the two unidirectional variants outperform the baseline. Although the segmentation features have not been directly refined by edge features in the direction $S \rightarrow E$, the gradient of the edge branch still propagates to the segmentation branch, which leads to indirect refinement. As for the other direction $S \leftarrow E$, its performance is worse than the final bidirectional model. This is because that the edge features have not been improved in this direction, which leads to limited enhancement of the segmentation features. The proposed bidirectional model significantly outperforms the two unidirectional variants. This indicates that the two proposed direction-specific integration operations collaboratively work well.

Table 3: *We test the proposed model with different integration operations in each direction. Cat means concatenation operation, and Mul means multiplication operation.*

Method	$S \rightarrow E$	$S \leftarrow E$	DUT-OMRON		DUTS-TEST	
			F_β^ω	S_α	F_β^ω	S_α
SCRN$_4$	Cat	Cat	0.679	0.814	0.764	0.866
	Mul	Mul	0.708	0.834	0.790	0.881
	Mul	Cat	**0.720**	**0.837**	**0.803**	**0.885**

Table 4: *The performances of different styles of CRUs and the effect of the residual learning.*

Model	Residual	Style	DUT-OMRON		DUTS-TEST	
			F_β^ω	S_α	F_β^ω	S_α
SCRN$_4$	✓	PPS	0.699	0.827	0.785	0.876
	✓	SPS	0.707	0.833	0.787	0.880
	✓	SSPS	**0.720**	**0.837**	**0.803**	**0.885**
	✗	SSPS	0.719	0.835	0.802	**0.885**

The effect of the interrelations. Motivated by the logical operations between binary segmentation and edge maps, we use two different integrating strategies in different directions: multiplication operation for the direction $S \rightarrow E$, and concatenation operation for the other direction $S \leftarrow E$. Here we test the effect of these operations and the results are shown in Table 3. It is obviously that using multiplication operation in both directions is better than using concatenation operation. This is because the edge features are easier to be affected by segmentation features owing to the fact that the segmentation features contain much more information than the edge features.

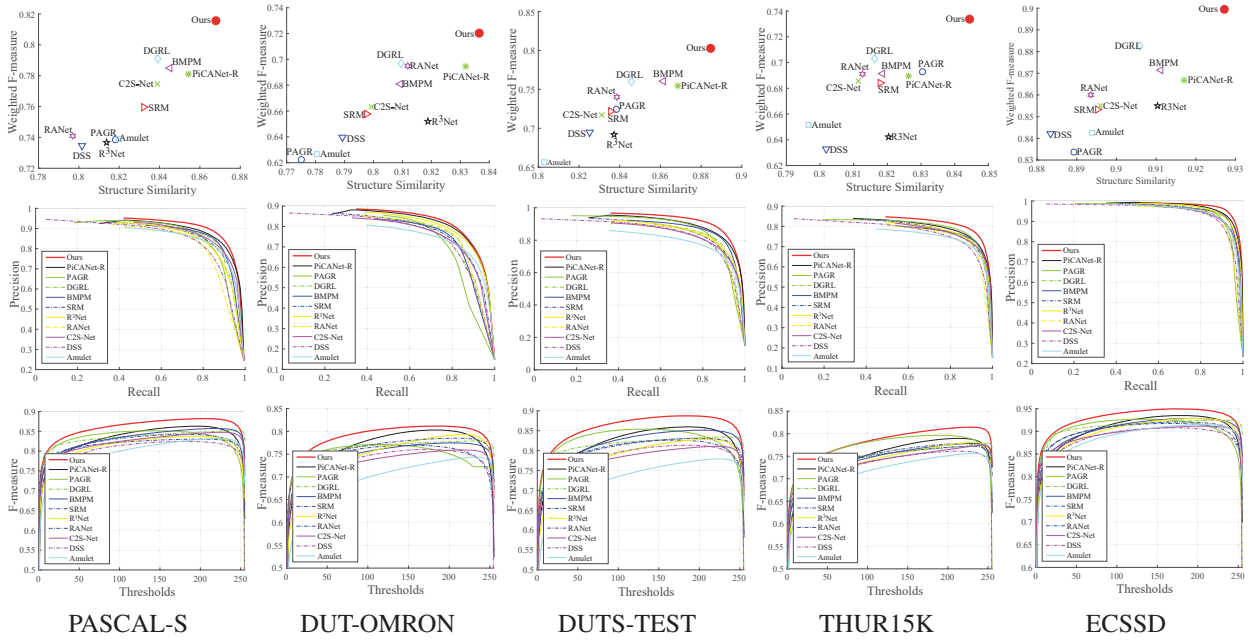
The effect of different styles of CRUs. Table 4 shows the performances of different styles of the CRUs. SPS outperforms PPS because it encodes more information in feature refinement. Furthermore, SSPS obtains the highest performance for it neglecting parts of lower-level features which contain more distractors than the higher-level features. In addition, we test the effect of the residual learning. It is found that residual learning improves the performance while hardly increasing the computation overhead.

4.4. Comparison with State-of-the-arts

We compare the proposed approach with 10 FCN-based SOD algorithms: Amulet [36], SRM [30], DSS [15], PAGR [37], RANet [4], R^{3}Net [7], C2S-Net [20], DGRL [31], BMPM [35] and PiCANet-R [22]. For fair comparison, we use the saliency maps provided by the authors or running the available source codes. Besides, some algorithms are trained on MSRA-B or MSRA10K (DSS, Amulet, RANet, R^{3}Net). So we re-train the proposed model on the MSRA-B

Table 5: *The maximum F-measure (maxF), mean absolute error (MAE) and frame per second (FPS) of the proposed model and* 10 *state-of-the-art algorithms. Top three scores are shown in red, green and blue.*

Method	FPS	ECSSD		HKU-IS		PASCAL-S		DUT-OMRON		DUTS-TEST		THUR15K	
		maxF↑	MAE↓	maxF↑	MAE↓	maxF↑	MAE↓	maxF↑	MAE↓	maxF↑	MAE↓	maxF↑	MAE↓
DSS [15]	23	0.908	0.063	0.898	0.051	0.826	0.102	0.764	0.072	0.813	0.065	0.761	0.083
Amulet [36]	20	0.913	0.059	0.887	0.053	0.828	0.095	0.737	0.083	0.779	0.085	0.756	0.093
RANet [4]	45	0.918	0.059	0.913	0.045	0.834	0.104	0.786	0.062	0.831	0.060	0.772	0.075
C2S-Net [20]	30	0.910	0.054	0.896	0.048	0.846	0.081	0.757	0.071	0.811	0.062	0.775	0.083
R^3Net [7]	29	0.929	0.051	0.910	0.047	0.837	0.101	0.793	0.073	0.828	0.067	0.781	0.078
SRM [30]	37	0.917	0.056	0.906	0.046	0.844	0.087	0.769	0.069	0.826	0.059	0.778	0.077
PAGR [37]	-	0.927	0.061	0.918	0.048	0.851	0.092	0.771	0.071	0.855	0.056	0.796	0.070
BMPM [35]	28	0.928	0.044	0.920	0.038	0.862	0.074	0.775	0.063	0.850	0.049	0.779	0.079
DGRL [31]	6	0.925	0.043	0.914	0.037	0.853	0.074	0.779	0.063	0.834	0.051	0.779	0.077
PiCANet-R [22]	5	0.935	0.047	0.919	0.043	0.863	0.075	0.803	0.065	0.860	0.051	0.790	0.081
Ours	52	0.950	0.038	0.934	0.034	0.882	0.064	0.812	0.056	0.888	0.040	0.814	0.066



PASCAL-S DUT-OMRON DUTS-TEST THUR15K ECSSD

Figure 4: *Quantitative comparisons of the proposed model with* 10 *state-of-the-art algorithms. The first row shows the weighted F-measure and structure similarity scores. The second and third rows are PR and F-measure curves, respectively.*

dataset and present the results in supplementary material.

Table 5 shows the maxF and MAE scores of the proposed model and 10 state-of-the-arts algorithms on six traditional benchmark datasets. We can see that the proposed model outperforms existing algorithms in all cases. In the first row of Fig. 4, we present F_β^ω (Y-axis) and S_α (X-axis) scores of the proposed model and the compared algorithms. This demonstrate that we generate more precise maps when evaluating them in different aspects. In the second and third rows of Fig. 4, we present the precision-recall curves and F-measure curves. And our curves are obviously higher than other curves. Fig. 5 shows the visual comparisons, which demonstrate that the proposed model can handle various challenging cases: complex scene (rows $4, 5$), low contrast (rows $1, 6$), small object (rows $2, 3$), large object (row 1) and multiple objects (rows $2, 4$). More visual comparison results can be found in the supplementary material.

Attributes-based performance on SOC. In the challenging SOC dataset [8], each salient image is accompanied by attributes that reflect common challenges in real-world scenes. These annotations are helpful to investigate the pros and cons of salient object detection models. Table 6 shows the structure similarity scores of the proposed model and 10 state-of-the-art algorithms. We can see that our model ranks

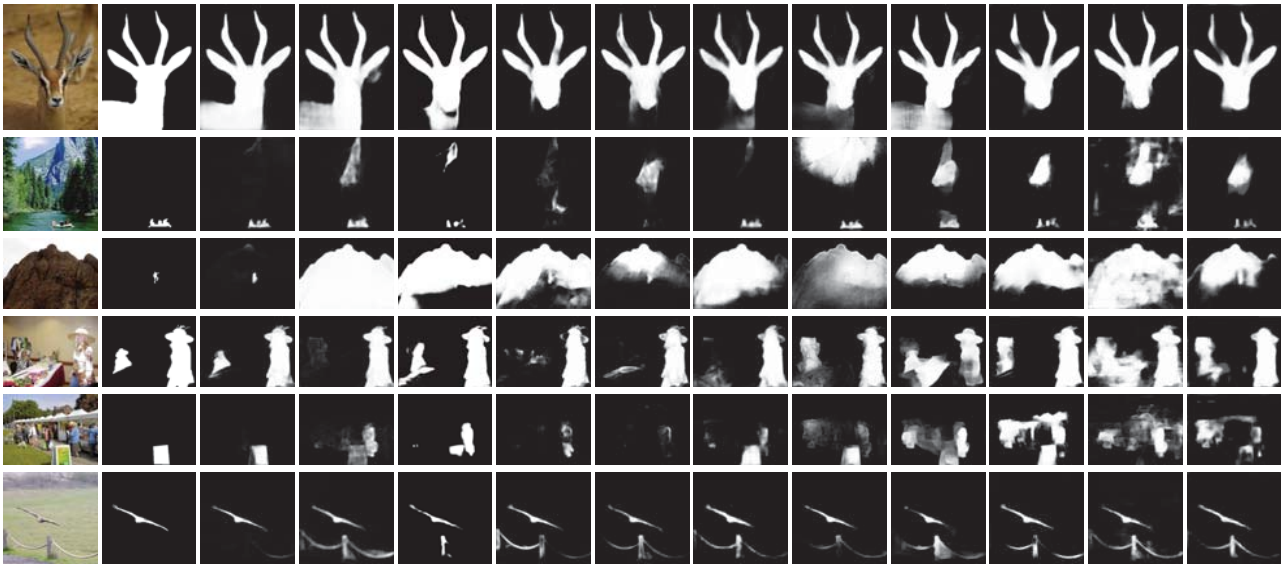


Figure 5: *Visual comparisons with the existing methods in some challenging cases: complex scene, low contrast, small object, large object, multiple objects.*

Table 6: *Attributes-based performance on the challenging SOC dataset [8]. We report the average structure similarity score over all datasets with that specific attribute as [8]. The average salient-object performance is presented in the last row. Top three scores are shown in red, green and blue.*

Attr	DSS	Amulet	RANet	C2S-Net	R3Net	SRM	DGRL	BMPM	PiCANet-R	Ours
AC	0.744	0.756	0.694	0.771	0.703	0.794	0.791	0.775	0.796	0.824
BO	0.587	0.653	0.475	0.703	0.451	0.691	0.728	0.675	0.728	0.709
CL	0.689	0.718	0.619	0.744	0.680	0.747	0.756	0.737	0.771	0.790
HO	0.753	0.764	0.692	0.772	0.715	0.794	0.800	0.784	0.805	0.827
MB	0.758	0.756	0.691	0.806	0.696	0.817	0.827	0.813	0.860	0.870
OC	0.703	0.714	0.616	0.745	0.643	0.734	0.748	0.744	0.763	0.779
OV	0.702	0.744	0.622	0.755	0.639	0.775	0.778	0.769	0.807	0.803
SC	0.752	0.748	0.697	0.760	0.703	0.774	0.779	0.783	0.784	0.817
SO	0.707	0.675	0.678	0.705	0.686	0.727	0.727	0.729	0.738	0.766
Avg	0.719	0.715	0.664	0.738	0.683	0.757	0.759	0.756	0.774	0.793

first among seven attributes of the nine attributes. Besides, our model also ranks first in average. These results indicate that the proposed model outperforms existing algorithms in the majority of challenging cases. Although we obtain smaller scores than DGRL and PiCANet in two attributes, our model runs nearly 10 times faster than them (Table 5).

5. Conclusion

In this paper, we propose a novel framework for salient object detection, called Stacked Cross Refinement Network (SCRN). Motivated by the logical interrelations between binary segmentation and edge maps, we propose a Cross Refinement Unit (CRU) in which two direction-specific integration operations are designed to improve the multi-level features of the two tasks. Incorporating stacked CRUs with the typical U-Net structures, the proposed model detects salient objects accurately and quickly. Experiments show that the proposed model significantly outperforms existing state-of-the art algorithms on six benchmark datasets and ranks first in majority of scenes of the SOC dataset.

Acknowledgement. This work was supported by the University of Chinese Academy of Sciences, in part of National Natural Science Foundation of China: 61472389, 61620106009, U1636214 and 61836002, in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013.

References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009.

[2] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *arXiv preprint arXiv:1411.5878*, 2014.

[3] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *IEEE TIP*, 24(12):5706–5722, 2015.

[4] Shuhan Chen, Xiuli Tan, Ben Wang, and Xuelong Hu. Reverse attention for salient object detection. In *ECCV*, pages 236–252, 2018.

[5] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015.

[6] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, and Shi-Min Hu. Salientshape: Group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014.

[7] Zijun Deng, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Jing Qin, Guoqiang Han, and Pheng-Ann Heng. R^3net: Recurrent residual refinement network for saliency detection. In *IJCAI*, pages 684–690, 2018.

[8] Deng-Ping Fan, Ming-Ming Cheng, Jiangjiang Liu, Shanghua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In *ECCV*, pages 196–212, 2018.

[9] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, pages 4558–4567, 2017.

[10] Deng-Ping Fan, Zheng Lin, Jia-Xing Zhao, Yun Liu, Zhao Zhang, Qibin Hou, Menglong Zhu, and Ming-Ming Cheng. Rethinking rgb-d salient object detection: Models, datasets, and large-scale benchmarks. *arXiv preprint arXiv:1907.06781*, 2019.

[11] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, June 2019.

[12] Wenlong Guan, Tiantian Wang, Jinqing Qi, Lihe Zhang, and Huchuan Lu. Edge-aware convolution neural network based salient object detection. *IEEE Signal Process. Lett.*, 26(1):114–118, 2019.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[14] Shengfeng He, Rynson WH Lau, Wenxi Liu, Zhe Huang, and Qingxiong Yang. Supercnn: A superpixelwise convolutional neural network for salient object detection. *IJCV*, 115(3):330–344, 2015.

[15] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. *IEEE TPAMI*, 41(4):815–828, 2019.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[17] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, 20(11):1254–1259, 1998.

[18] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. Deep saliency with encoded low level distance map and high level features. In *CVPR*, pages 660–668, 2016.

[19] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *CVPR*, pages 5455–5463, 2015.

[20] Xin Li, Fan Yang, Hong Cheng, Wei Liu, and Dinggang Shen. Contour knowledge transfer for salient object detection. In *ECCV*, pages 370–385, 2018.

[21] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014.

[22] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *CVPR*, pages 3089–3098, 2018.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[24] Zhiming Luo, Akshaya Kumar Mishra, Andrew Achkar, Justin A Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. In *CVPR*, pages 6593–6601, 2017.

[25] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps. In *CVPR*, pages 248–255, 2014.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[27] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, pages 3183–3192, 2015.

[28] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, pages 136–145, 2017.

[29] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, pages 825–841, 2016.

[30] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, pages 4019–4028, 2017.

[31] Tiantian Wang, Lihe Zhang, Shuo Wang, Huchuan Lu, Gang Yang, Xiang Ruan, and Ali Borji. Detect globally, refine locally: A novel approach to saliency detection. In *CVPR*, pages 3127–3135, 2018.

[32] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *CVPR*, pages 1395–1403, 2015.

[33] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013.

[34] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013.

[35] Lu Zhang, Ju Dai, Huchuan Lu, You He, and Gang Wang. A bi-directional message passing model for salient object detection. In *CVPR*, pages 1741–1750, 2018.

[36] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, pages 202–211, 2017.

[37] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *CVPR*, pages 714–722, 2018.

[38] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *CVPR*, pages 1265–1274, 2015.

[39] Yun-Zhi Zhuge, Gang Yang, Pingping Zhang, and Huchuan Lu. Boundary-guided feature aggregation network for salient object detection. *IEEE Signal Process. Lett.*, 25(12):1800–1804, 2018.