






Feature Recalibration Network for Salient Object Detection

Zhenshan Tan  and Xiaodong Gu  

Department of Electronic Engineering, Fudan University, Shanghai 200433, China
{zstan19, xdgu}@fudan.edu.cn

Abstract. Learning-based models have demonstrated the superiority of extracting and aggregating saliency features. However, we observe that most off-the-shelf methods mainly focus on the calibration of decoder features while ignore the recalibration of vital encoder features. Moreover, the fusion between encoder features and decoder features, and the transfer between boundary features and saliency features deserve further study. To address the above issues, we propose a feature recalibration network (FRCNet) which consists of a consistency recalibration module (CRC) and a multi-source feature recalibration module (MSFRC). Specifically, intersection and union mechanisms in CRC are embedded after the decoder unit to recalibrate the consistency of encoder and decoder features. By the aid of the special designed mechanisms, CRC can suppress the useless external superfluous information and enhance the useful internal saliency information. MSFRC is designed to aggregate multi-source features and reduce parameter imbalance between saliency features and boundary features. Compared with previous methods, more layers are applied to generate boundary features, which sufficiently leverage the complementary features between edges and saliency. Besides, it is difficult to predict the pixels around the boundary because of the unbalanced distribution of edges. Consequently, we propose an edge recalibration loss (ERC) to further recalibrate the equivocal boundary features by paying more attention to salient edges. In addition, we also explore a compact network (cFRCNet) that improves the performance without extra parameters. Experimental results on five widely used datasets show that the FRCNet achieves consistently superior performances under various evaluation metrics. Furthermore, FRCNet runs at the speed of around 30 fps on a single GPU.

Keywords: Salient object detection · Feature recalibration · Edge recalibration loss

1 Introduction

Salient Object Detection (SOD) aims at locating the most attractive regions of images or videos and is used as the pre-processing procedure for downstream vision tasks [1, 2]. Earlier SOD algorithms mainly rely on hand-crafted features such as background prior, color contrast and contextual cue to extract salient

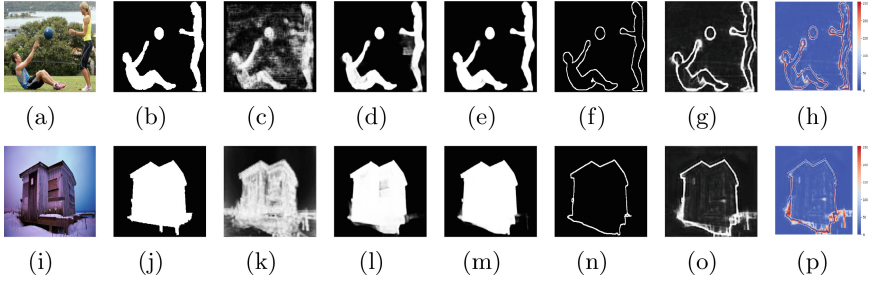


Fig. 1. Feature fusion and loss difference. (a) (i): images, (b) (j): ground truths, (c) (k): encoder features, (d) (l): decoder features, (e) (m): fusion features, (f) (n): edges directly from fusion features, (g) (o): generated edges from network, (h) (p): error maps of (f) (g) and (n) (o).

objects. However, these unsupervised stimuli-driven algorithms hardly capture high-level semantic features and are not robust to detect salient object in challenging complex background. In recent years, convolutional neural networks have been introduced to SOD and achieve best results on many benchmark datasets [5, 8, 10–13, 15, 18]. One representative network is U-net-like [9] structure, which significantly improves the resulting saliency maps. More recently, boundary features are integrated to convolutional features to predict more reliable saliency structure [19].

There are still three main challenges in SOD. Firstly, most of the previous works mainly focus on the construction of decoder features and various effective connection. However, the reconstruction of important encoder features and the fusion between encoder and decoder features remain scarce. As shown in Fig. 1 (c) (k), the raw encoder features are coarse and blurry, which may mislead the subsequent feature transfer. Secondly, the aggregation of boundary features and saliency features has not been comprehensively studied. In fact, the correlation and difference between edges and saliency directly determine the performance of the generated saliency maps. Moreover, because the boundary features are less than the saliency features, conventional feature fusion methods may lead to parameter imbalance [19]. Finally, at present, boundary features are applied to refine the saliency features. Most of the existing methods generate boundary features from several convolutional layers after saliency features [19, 21]. Nevertheless, the boundary features from the previous saliency convolutional layer may be changed during the process of feature transfer (see Fig. 1 (h) (p)). Therefore, when the saliency feature is supervised, the boundary features extracted from the saliency features should be supervised simultaneously, aiming at recalibrating the structure of saliency maps.

To address the above challenges, we propose a feature recalibration network (FRCNet) for accurate and fast SOD. In a specific, for the first issue, we propose a consistency recalibration module (CRC). CRC adopts an intersection and union mechanism, in which intersection mechanism is used to filter noise

information and union mechanism is used to enhance saliency information. With the help of CRC, the noises are removed and the coarse edges are refined. In addition, after CRC, top-down feature refinement is adopted to aggregate multi-level features. Different from previous top-down connection mechanism, we reduce the spatial resolution to decrease the memory computation. For the second issue, we introduce a multi-source feature recalibration module (MSFRC) to aggregate multiple features and reduce parameter imbalance between edges and saliency. Considering the complementarity between edge information and salient information, MSFRC is designed to progressively learn the correlation and recalibrate the difference between boundary features and saliency features. In addition, more boundary feature layers help balance the parameters between edges and saliency. For the final issue, we propose an edge recalibration loss (ERC) to further recalibrate the boundary features directly from the saliency features. Because the pixels around the edges are hard to predict and discriminate, paying more attention to these edge pixels can refine the resulting saliency maps. In ERC, the edge pixels are given the maximum attention, then the saliency maps, and finally the background. The specially designed loss can lead the network to focus on the edges and further enhance the detection of saliency structure.

Experimental results on five popular datasets show that the proposed FRC-Net achieves consistently superior performance in comparison with other state-of-the-arts. The visual assessment verifies the results. Besides, the ablation studies demonstrate the effectiveness of each proposed module. FRCNet runs at around 30 fps on a single NVIDIA 2080Ti GPU. The codes will be released. In a word, the main contributions can be highlighted as follows.

- We propose a consistency recalibration module to recalibrate the encoder features and the decoder features, which is able to refine the coarse encoder features and selectively aggregate encoder features and decoder features.
- We propose a multi-source feature recalibration module to aggregate multi-source features and reduce parameter imbalance, which progressively learns the correlation and recalibrates the difference between edges and saliency.
- A novel edge recalibration loss is designed to guide the network to focus on the edge information and refine the coarse edges.
- Visual and objective assessments on five datasets show that the proposed FRCNet can achieve consistently superior performance, which verifies the superiority of the proposed model.

2 Proposed Method

As illustrated in Fig. 2, we propose a consistency recalibration module to aggregate encoder features and decoder features, which can suppress the useless information and enhance useful information. To progressively couple the complementarity and reduce parameter imbalance, we propose a multi-source feature recalibration module to aggregate multi-source features. To recalibrate the saliency edges, we design a novel edge recalibration loss to focus on the refinement of the edges.

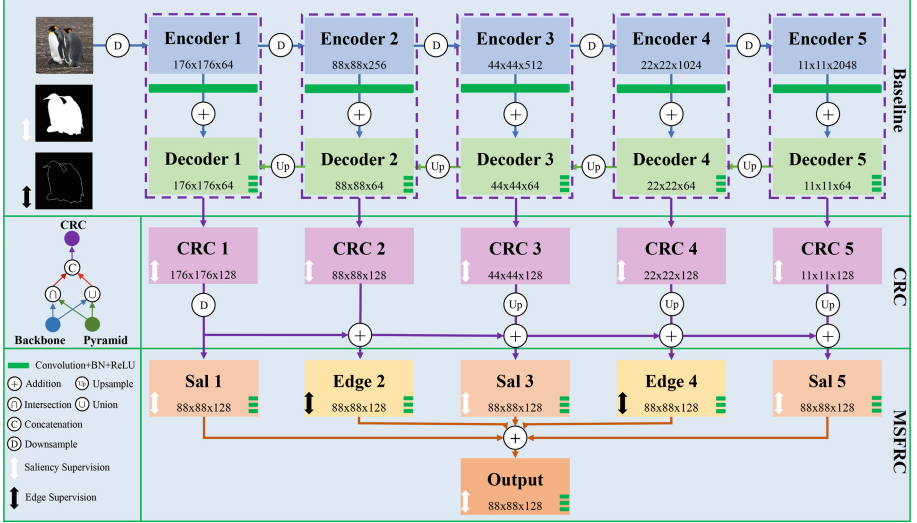


Fig. 2. Overall architecture. Baseline includes an encoder network and a decoder network. ResNet-50 is adopted as the encoder network and a U-net-like structure is used as the decoder network. CRC: Consistency recalibration module. MSFRC: Multi-source feature recalibration module.

2.1 Consistency Recalibration Module

At first, inspired by U-net-like [9] structure, a typical encoder-decoder network is adopted in this paper to generate the baseline features. Then, we propose consistency recalibration module (CRC) to refine both encoder features $f_e \in R^{H \times W \times C}$ and decoder features $f_d \in R^{H \times W \times C}$. As shown in Fig. 1, f_e contains lots of details such as textures and edges, while lacks enough semantic information. On the contrary, f_d extracts rich semantic features with coarse edges. The possible reason is that f_e is close to the input and the receptive fields are small. While f_d has relatively large receptive fields but goes through too many downpoolings and upsamplings, which leads to the loss or the extra of information.

Therefore, intersection and union mechanisms are applied in CRC to filter the noises and enhance the semantic features. Intersection mechanism picks up the most confident pixels and suppresses the noise pixels. However, some salient pixels especially around the edges may be corroded as well. Consequently, union mechanism is designed to enhance the internal saliency features. The union mechanism enhances the confident pixels in both encoder and decoder features, meanwhile, the indefinite pixels are enhanced as well. Based on the intersection and union mechanisms, the background noises are suppressed and the foreground information are strengthened. Therefore, the intersection and union are designed to refine the features instead of conventional encoder features and decoder features.

Specifically, as illustrated in Fig. 2, CRC contains three steps to obtain clear and complete saliency features. Firstly, the two features f_e and f_d are applied to generate the intersection features f_I . Then, the same two features f_e and f_d are used for generating the union features f_U . Here, f_e and f_d are reused twice because CRC optimizes these two features from suppression and enhancement. Finally, f_I and f_U are concatenated to preserve their own features. The whole progress can be concluded as follows.

$$f_{CRC_{in}}^i = \text{concat}(f_e^i \cap f_d^i, f_e^i \cup f_d^i), \quad (1)$$

where, $f_{CRC_{in}}^i$ is the input features of CRC^i , $i \in \{1, 2, 3, 4, 5\}$ is the layer number, and $\text{concat}(\cdot)$ is the operation of concatenation. As shown in Eq. (1), CRC has no convolution calculation, therefore, CRC refines the results without any parameter increasement. In addition, top-down mechanism is adopted to refine the saliency maps. In consideration of the different receptive fields in different layers, cross layer connection can directly transfer the semantic features to other layers, enriching and integrating the salient features of each layer. Therefore, the final CRC features can be denoted as follows.

$$f_{CRC_{out}}^i = \begin{cases} \text{down}(f_{CRC_{in}}^i), & i = 1 \\ \text{down}(f_{CRC_{in}}^1) + f_{CRC_{in}}^i, & i = 2 \\ \text{down}(f_{CRC_{in}}^1) + \text{up}(f_{CRC_{in}}^i), & i = \text{other} \end{cases} \quad (2)$$

where, $f_{CRC_{out}}^i$ is the output features of CRC^i , $\text{down}(\cdot)$ denotes the bilinear downsampling operation and $\text{up}(\cdot)$ denotes the bilinear upsampling operation. Finally, $f_{CRC_{out}}^i$ is supervised by saliency ground truth after a 1×1 channel adjustment layer.

2.2 Multi-source Feature Recalibration Module

Multi-source feature recalibration (MSFRC) is divided into two steps. The first step is to progressively learn the correlation and recalibrate the difference between edges and saliency, and the second step is to integrate boundary features and saliency features. As illustrated in Fig. 2, there are three feature changes in the first step of MSFRC, which brings more correlation information and difference information. In addition, previous methods [19] introduce boundary features as auxiliary supervision, however, they only use one of the convolutional blocks to generate boundary features, which may lead to parameter imbalance. The larger the saliency parameter is, the more attention the network pays to saliency, while the boundary features with less parameters are easily ignored. Note that the boundary features should not be larger than the saliency features, which may lead to put the cart before the horse.

For the first step, the 1-th block, the 3-th block and the 5-th block are supervised by saliency, which generate the saliency features. The 2-th block and the 4-th block are supervised by edges, which generate the boundary features. Because of back propagation, each layer has potential feature transfer

even though there is no direct layer-to-layer connection. Therefore, the boundary features are generated between every saliency layer, which force the network to learn the correlation and recalibrate the diversity. Besides, the saliency layers are more than the edge layers, which ensures the advantage of saliency. For the second step, different layers are aggregated to a whole layer by addition. Two boundary feature layers in MSFRC increase the proportion of edges, which reduces the parameter imbalance.

Specifically, the features $f_{CRC_{out}}^i$ from CRC are transferred to MSFRC, and each $f_{CRC_{out}}^i$ goes through three convolutional layers with batch normalization and ReLU activation function. Then, the five feature maps in MSFRC are added to aggregate multi-level features. The resolution and the channel are unchanged during the process. The whole process can be denoted as follows.

$$f_{MSFRC}^i = F_2^3 \left(\sum_{i=1}^5 F_1^3 (f_{CRC_{out}}^i; \theta_1^3); \theta_2^3 \right), \quad (3)$$

where, F_1^3 and F_2^3 are the combination of three convolution, batch normalization and ReLU. θ_1^3 and θ_2^3 are the parameters of F_1^3 and F_2^3 . Finally, if $i = 1, 3, 5$, f_{MSFRC}^i is supervised by saliency ground truth after a 1×1 channel adjustment layer. If $i = 2, 4$, f_{MSFRC}^i is supervised by edge ground truth after a 1×1 channel adjustment layer.

2.3 Loss Function

In mathematical optimization, loss function represents the degree of inconsistency between the predicted value and the ground truth value. In this paper, loss function is divided into saliency loss function and edge loss function.

Saliency Loss Function. Currently, most of the methods adopt edge supervision to refine the salient edges. However, the boundary features are usually generated after multiple convolutional layers, which may lead to the feature inconsistency between edge and saliency. Even though in MSFRC, the feature inconsistency is applied to correct the possible errors occurred in previous layers, the salient edges directly from saliency feature layers still need to refine. Therefore, we propose an edge recalibration loss function (ERC) as follows.

$$L_{ERC}^{(s)} = \begin{cases} - \sum_{(m,n)} [\gamma M G \log P], & \text{if } M = 1, G = 1 \\ - \sum_{(m,n)} [(1 - \gamma)(1 - M) G \log P], & \text{if } M = 0, G = 1 \\ - \sum_{(m,n)} [\gamma M (1 - G) \log(1 - P)], & \text{if } M = 1, G = 0 \\ - \sum_{(m,n)} [(1 - \gamma)(1 - M)(1 - G) \log(1 - P)], & \text{if } M = 0, G = 0 \end{cases} \quad (4)$$

where, $L_{ERC}^{(s)}$ is the loss of s -th layer. M , G , P represent the abbreviations $M(m, n)$, $G(m, n)$, $P(m, n)$, respectively. $M(m, n) \in \{0, 1\}$ is the edge ground truth of the pixels (m, n) , γ is a hyper parameter which controls the edge superiority and we set γ to 0.75 in this paper. $G(m, n) \in \{0, 1\}$ is the saliency ground truth and $P(m, n)$ is the predicted saliency probability. Equation 4 means the saliency features are constrained directly by the edge ground truth. Considering the sparsity of edge, we expand the edges in a 3×3 neighborhood. Consequently, Eq. 4 can be concluded as Eq. 5.

$$L_{ERC}^{(s)} = - \sum_{(m,n)} [(\gamma(M \oplus \mathbf{B}) + (1 - \gamma)(1 - (M \oplus \mathbf{B}))) (G \log P + (1 - G)(1 - \log P))], \quad (5)$$

where, \mathbf{B} is a 3×3 matrix with all 1, and $M \oplus \mathbf{B} = \{m, n | \mathbf{B}_{mn} \cap M \neq \emptyset\}$ denotes the morphological dilation. For all pixels, ERC first pays more attention to the salient edges, then the saliency and finally the background. ERC has two superiorities. On the one hand, ERC is used for recalibrating the salient edges directly from salient features, which helps the network detect the edges of saliency features. On the other hand, the saliency features and the boundary features of them are optimized simultaneously, which means ERC can balance the two to some extent. Equation 5 means ERC loss mainly focuses on the saliency structure. Therefore, the optimization based on foreground and background should be considered as well. We adopt BCE loss and Dice loss [6] to further optimize the network.

$$L_{BCE}^{(s)} = - \sum_{(m,n)} [G \log P + (1 - G)(1 - \log P)], \quad (6)$$

$$L_{Dice}^{(s)} = 1 - \frac{\sum_{(m,n)} GP}{\sum_{(m,n)} [G + P]} \quad (7)$$

Edge Loss Function. The edges generated in MSFRC are supervised by weighted BCE loss. Different from BCE loss, weighted BCE loss considers the sparsity difference between foreground pixels and background pixels. Weighted BCE loss is shown in Eq. 8.

$$L_{wBCE}^{(e)} = - \sum_{(m,n)} [\omega G \log P + (1 - G)(1 - \log P)], \quad (8)$$

where, $L_{wBCE}^{(e)}$ is the loss of e -th layer. ω is the abbreviation $\omega(m, n)$ and we set $\omega(m, n) = \frac{\#0\{m,n\}}{\#1\{m,n\}}$, in which $\#0\{m, n\}$ (or $\#1\{m, n\}$) is the number of 0 (or 1). Compared with the background, the sparse edge foreground are assigned higher weights by ω . The hybrid loss can be denoted as follows.

$$L = \sum_{s=1}^S [L_{ERC}^{(s)} + L_{BCE}^{(s)} + L_{Dice}^{(s)}] + \sum_{e=1}^E L_{wBCE}^{(e)}, \quad (9)$$

where, S and E represent the layer number and are set as 9 and 2, respectively.

Table 1. Comparison with state-of-the-art methods. max F-measure (F_M , larger is better), max E-measure (E_M , larger is better) and S-measure (S_m , larger is better) are applied to evaluate the performance. Values with bold fonts indicate the best performance of all results.

| Methods | ECSSD | | | DUT-OMRON | | | PASCAL-S | | | DUTS-Test | | | HKU-IS | | |
|---------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | F_M | E_M | S_m | F_M | E_M | S_m | F_M | E_M | S_m | F_M | E_M | S_m | F_M | E_M | S_m |
| HRSOD | .932 | .925 | .887 | .743 | .796 | .761 | .846 | .858 | .815 | .835 | .878 | .823 | .910 | .932 | .877 |
| EGNet | .947 | .954 | .924 | .815 | .863 | .838 | .865 | .886 | .851 | .889 | .922 | .885 | .935 | .957 | .917 |
| SCRN | .950 | .955 | .926 | .811 | .875 | .836 | .877 | .904 | .868 | .888 | .925 | .885 | .934 | .955 | .916 |
| AFNet | .935 | .946 | .913 | .797 | .856 | .825 | .858 | .890 | .847 | .863 | .908 | .866 | .923 | .948 | .905 |
| MLMS | .928 | .916 | .911 | .774 | .839 | .809 | .864 | .847 | .845 | .852 | .863 | .862 | .920 | .938 | .907 |
| BASNet | .942 | .950 | .915 | .805 | .871 | .836 | .854 | .881 | .837 | .859 | .902 | .866 | .928 | .951 | .909 |
| CPD | .939 | .950 | .917 | .797 | .868 | .825 | .859 | .885 | .847 | .865 | .914 | .869 | .925 | .950 | .905 |
| GateNet | .945 | .951 | .919 | .818 | .872 | .837 | .869 | .898 | .857 | .888 | .926 | .885 | .933 | .954 | .914 |
| F3Net | .945 | .953 | .923 | .813 | .867 | .838 | .872 | .897 | .860 | .891 | .926 | .888 | .937 | .957 | .917 |
| ITSD | .947 | .957 | .924 | .821 | .878 | .840 | .870 | .901 | .858 | .883 | .929 | .885 | .934 | .959 | .917 |
| MINet | .947 | .956 | .924 | .810 | .865 | .832 | .867 | .897 | .855 | .884 | .926 | .884 | .935 | .959 | .919 |
| cFRCNet | .944 | .953 | .924 | .821 | .879 | .841 | .872 | .900 | .861 | .887 | .929 | .886 | .933 | .954 | .912 |
| FRCNet | .951 | .958 | .928 | .828 | .885 | .849 | .879 | .905 | .865 | .893 | .932 | .890 | .938 | .960 | .920 |

3 Experiments

3.1 Datasets and Evaluation Metrics

The performance of FRCNet is evaluated on five benchmark datasets: DUT-OMRON with 5168 difficult images, DUTS-test with 5019 complex images, PASCAL-S with 850 images, ECSSD with 1000 images and HKU-IS with 4447 images. Same as the current methods, DUTS-train is used as the training dataset.

Three widely used metrics are applied to evaluate the performance of FRCNet and other state-of-the-art methods. The first one is maximal F-measure, which has been adopted in most of SOD methods. E-measure [4] and structural similarity measure [3] are widely used metrics in recent years.

3.2 Implementation Details

We train FRCNet on DUTS-train dataset following previous works. For a fair comparison, ResNet-50 and U-net-like structure [9] are used as the encoder network and the decoder network, respectively. The whole framework is implemented in PyTorch on an NVIDIA 2080Ti GPU and FRCNet is trained end-to-end. We utilize stochastic gradient descent (SGD) optimizer and the hyper parameters are set as follows: maximum learning rate = 0.005, weight decay = 0.0005, momentum = 0.9. In addition, the learning rate is adjusted by warm-up and linear decay strategies. We train FRCNet for 100 epochs with a batchsize of 32. We do not use any pre-processing or post-processing techniques. The source code will be released.

3.3 Comparison with the State-of-the-Art

Quantitative Comparison. We compare the proposed FRCNet against 12 state-of-the-art SOD methods, including HRSOD [18], AFNet [5], MLMSNet [15], BASNet [8], SCRNet [17], EGNNet [19], CPD [16], GateNet [20], F3Net [14], ITSD [21], MINet [7]. For a fair comparison, the saliency maps of the above methods are provided by the authors. As illustrated in Table 1, FRCNet outperforms other methods across five datasets, especially with respect to MaxF metrics. Besides, we also remove all the parameters of MSFRC (cFRCNet). Therefore, the model size is the baseline size. We can observe that cFRCNet also achieves competitive results. This verifies that our CRC and MSFRC are effective.

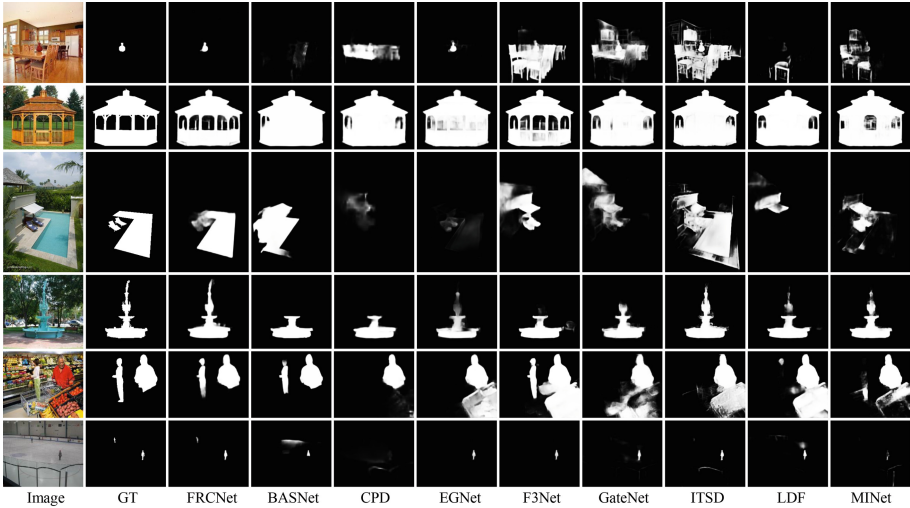


Fig. 3. Visual comparison with state-of-the-art methods.

Visual Comparison. As illustrated in Fig. 3, we visualize some results of FRCNet and 9 typical methods for saving room. The resulting saliency maps of FRCNet achieve superior performance, which are closer to the ground truth in visual. Specifically, with the help of CRC, our model not only enhances the salient regions, but also suppresses the background noises (see Fig. 3 row 1, 2 and 3). By the aid of the complementarity of saliency features and boundary features in MSFRC, FRCNet is able to generate more accurate and complete saliency maps even though in the complex background (see Fig. 3 row 4, 5 and 6). Furthermore, FRCNet achieves these results without any pre-processing or post-processing.

3.4 Ablation Studies

In this paper, there is one hyper parameter (*i.e.*, γ) to be determined. γ is used in ERC loss function to adjust the weights of boundary features. Obviously, γ

Table 2. Ablation study on different modules optimized by BCE loss. Baseline: the baseline network.

| Baseline | CRC | MSFRC | ECSSD | | | DUT-OMRON | | | PASCAL-S | | |
|----------|-----|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | | | F_M | E_M | S_m | F_M | E_M | S_m | F_M | E_M | S_m |
| ✓ | | | .914 | .921 | .897 | .763 | .805 | .789 | .816 | .829 | .803 |
| ✓ | ✓ | | .921 | .928 | .904 | .774 | .818 | .802 | .825 | .844 | .812 |
| ✓ | | ✓ | .931 | .937 | .910 | .790 | .837 | .812 | .839 | .861 | .823 |
| ✓ | ✓ | ✓ | .938 | .943 | .918 | .803 | .845 | .823 | .852 | .875 | .835 |

Table 3. Ablation study on different losses.

| ω BCE | BCE | Dice | ERC | ECSSD | | | DUT-OMRON | | | PASCAL-S | | |
|--------------|-----|------|-----|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | | | | F_M | E_M | S_m | F_M | E_M | S_m | F_M | E_M | S_m |
| ✓ | ✓ | | | .942 | .945 | .920 | .807 | .856 | .831 | .857 | .879 | .840 |
| ✓ | | ✓ | | .933 | .938 | .912 | .791 | .847 | .820 | .842 | .864 | .826 |
| ✓ | | | ✓ | .944 | .951 | .922 | .811 | .860 | .835 | .861 | .886 | .846 |
| ✓ | ✓ | ✓ | | .946 | .949 | .925 | .818 | .869 | .839 | .862 | .892 | .851 |
| ✓ | | ✓ | ✓ | .947 | .951 | .927 | .822 | .873 | .841 | .866 | .896 | .854 |
| ✓ | ✓ | | ✓ | .948 | .954 | .926 | .824 | .878 | .844 | .871 | .900 | .860 |
| ✓ | ✓ | ✓ | ✓ | .951 | .958 | .928 | .828 | .885 | .849 | .879 | .905 | .865 |

should be larger than 0.5, which assigns higher weights to the edges. When $\gamma = 1$, ERC will only optimize the edges without the saliency features. As mentioned above, ERC should detect the edges and saliency simultaneously. Therefore, the value of γ should be between 0.5 and 1. If γ is close to 0.5, ERC prefers to detect the saliency. In contrast, if γ is close to 1, ERC prefers to detect the edges. Therefore, we set γ to 0.75, which balances the assigned weights between boundary features and saliency features.

To validate the effectiveness of each key module, a series of detailed analysis is conducted on DUT-OMRON dataset under various metrics. As shown in Table 2 and Table 3, the ablation study is divided into loss function part and module part. For module ablation, we observe that both CRC and MSFRC can refine the results. Furthermore, the combination of CRC and MSFRC achieves superior qualitative results. For loss ablation, the hybrid loss achieves the best performance. Besides, the single ERC loss performs better than the single BCE loss and Dice loss. The results in Table 2 and Table 3 verify the effectiveness of the proposed modules and losses.

4 Conclusion

In this paper, a novel model FRCNet is proposed for accurate and fast SOD. Firstly, to suppress the external noises and enhance the internal salient object,

we introduce the intersection and union mechanisms to CRC to recalibrate the consistency of encoder and decoder features. Secondly, to learn the correlation and recalibrate the difference between boundary features and saliency features, MSFRC is proposed to sufficiently couple the complementary features between edges and saliency by alternate feature transfer. Besides, MSFRC can reduce parameter imbalance and effectively aggregate different source features to refine the resulting saliency maps. Finally, to further guide the network to focus on the edges, we propose an ERC loss to recalibrate the equivocal edge pixels. Experimental results on five datasets demonstrate that the proposed FRCNet can achieve consistently superior performance under various metrics.

Acknowledgements. This work was supported in part by National Natural Science Foundation of China under grant 62176062.

References

1. Chen, C., Tan, Z., Cheng, Q., et al.: UTC: a unified transformer with inter-task contrastive learning for visual dialog. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, pp. 18103–18112. IEEE (2022)
2. Cheng, Q., Tan, Z., Wen, K., et al.: Semantic pre-alignment and ranking learning with unified framework for cross-modal retrieval. *IEEE Trans. Circuits Syst. Video Technol.* (2022)
3. Fan, D., Cheng, M., Liu, Y., et al.: Structure-measure: a new way to evaluate foreground maps. In: Proceedings of the IEEE International Conference on Computer Vision, Hawaii, pp. 4548–4557. IEEE (2017)
4. Fan, D., Gong, C., Cao, Y., et al.: Enhanced-alignment measure for binary foreground map evaluation. In: Proceedings of the International Joint Conference on Artificial Intelligence (2018)
5. Feng, M., Lu, H., Ding, E.: Attentive feedback network for boundary-aware salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision, California, pp. 1623–1632. IEEE (2019)
6. Fidon, L., et al.: Generalised wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks. In: Crimi, A., Bakas, S., Kuijf, H., Menze, B., Reyes, M. (eds.) *BrainLes 2017*. LNCS, vol. 10670, pp. 64–76. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75238-9_6
7. Pang, Y., Zhao, X., Zhang, L., et al.: Multi-scale interactive network for salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision, Seattle, pp. 9413–9422. IEEE (2020)
8. Qin, X., Zhang, Z., Huang, C., et al.: BASNet: boundary-aware salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision, California, pp. 7479–7489. IEEE (2019)
9. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) *MICCAI 2015*. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
10. Tan, Z., Hua, Y., Gu, X.: Salient object detection with edge recalibration. In: Farkas, I., Masulli, P., Wermter, S. (eds.) *ICANN 2020*. LNCS, vol. 12396, pp. 724–735. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61609-0_57

11. Tan, Z., Gu, X.: Scale balance network for accurate salient object detection. In: Proceedings of the International Joint Conference on Neural Networks, Glasgow, pp. 1–7. IEEE (2020)
12. Tan, Z., Gu, X.: Depth scale balance saliency detection with connective feature pyramid and edge guidance. *Appl. Intell.* **51**(8), 5775–5792 (2021)
13. Tan, Z., Gu, X.: Co-saliency detection with intra-group two-stage group semantics propagation and inter-group contrastive learning. *Knowl.-Based Syst.* **252**, 109356 (2022)
14. Wei, J., Wang, S., Huang, Q.: F3Net: fusion, feedback and focus for salient object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 12321–12328 (2020)
15. Wu, R., Feng, M., Guan, W., et al.: A mutual learning method for salient object detection with intertwined multi-supervision. In: Proceedings of the IEEE International Conference on Computer Vision, California, pp. 8150–8159. IEEE (2019)
16. Wu, Z., Su, L., Huang, Q.: Cascaded partial decoder for fast and accurate salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision, California, pp. 3907–3916. IEEE (2019)
17. Wu, Z., Su, L., Huang, Q.: Stacked cross refinement network for edge-aware salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision, California, pp. 7264–7273. IEEE (2019)
18. Zeng, Y., Zhang, P., Zhang, J., et al.: Towards high-resolution salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision, Seoul, pp. 7234–7243. IEEE (2019)
19. Zhao, J., Liu, J., Fan, D., et al.: EGNNet: edge guidance network for salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision, California, pp. 8779–8788. IEEE (2019)
20. Zhao, X., Pang, Y., Zhang, L., Lu, H., Zhang, L.: Suppress and balance: a simple gated network for salient object detection. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 35–51. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_3
21. Zhou, H., Xie, X., Lai, J., et al.: Interactive two-stream decoder for accurate and fast saliency detection. In: Proceedings of the IEEE International Conference on Computer Vision, Seattle, pp. 9141–9150. IEEE (2020)