

Recursive Contour-Saliency Blending Network for Accurate Salient Object Detection

Yun Yi Ke

Computer Vision & AI Technology Lab
Open8 Singapore

yunyikeyyk@gmail.com

Takahiro Tsubono[†]

Computer Vision & AI Technology Lab
Open8 Singapore

tsubonot@open8.com

Abstract

Contour information plays a vital role in salient object detection. However, excessive false positives remain in predictions from existing contour-based models due to insufficient contour-saliency fusion. In this work, we designed a network for better edge quality in salient object detection. We proposed a contour-saliency blending module to exchange information between contour and saliency. We adopted recursive CNN to increase contour-saliency fusion while keeping the total trainable parameters the same. Furthermore, we designed a stage-wise feature extraction module to help the model pick up the most helpful features from previous intermediate saliency predictions. Besides, we proposed two new loss functions, namely Dual Confinement Loss and Confidence Loss, for our model to generate better boundary predictions. Evaluation results on five common benchmark datasets reveal that our model achieves competitive state-of-the-art performance.

1. Introduction

Salient object detection (SOD) aims to detect and segment the most attention-attractive region or object in a visual scene. Unlike eye fixation prediction (FP) [34], SOD requires obtaining the entire region with clear boundaries. Due to its essential role and wide applications in image understanding [53], image captioning [8][42], search engines [54], AR/VR [30] and video summarization [25], recently, various methods have been proposed in the field.

Since 2015, Convolutional Neural Networks (CNNs) [10][15] have been adopted for SOD tasks. Though algorithms like PiCANet [23], BMPM [45], and PAGRN [47] achieved significantly better results, the predicted object usually has poor boundaries. To obtain more precise boundaries, proposed by Qin *et al.*, BASNet [31] adopted a

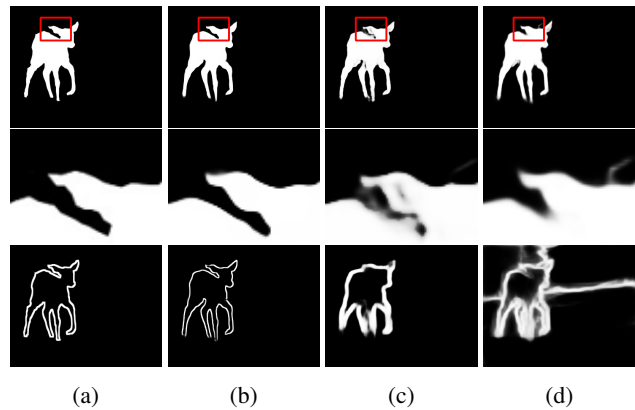


Figure 1: Visual comparisons between contour-based models. Saliency (first row) and corresponding contour predictions (last row) are listed. Ground truth contours are obtained via erosion and dilation with kernel size of 5. (a) ground truth, (b) ours, (c) ITSD [52] and (d) PoolNet [22].

boundary refinement U-Net [32] at the end of the saliency detection network and trained their model using various losses. In [5], Chen *et al.* proposed Contour Loss (CTLoss), which was a weighted Binary Cross-Entropy (BCE) loss, to improve the boundary predictions. Alternatively, models like EGNNet [49], PoolNet [22], and ITSD [52] fused contour predictions with saliency by explicitly supervising a contour branch. With contour cues, models yielded better boundary predictions.

However, above mentioned models still hold several problems that can be further improved. First, for better performance, many studies have introduced a huge number of trainable parameters. EGNNet contains 108 million parameters, BASNet and its extended work, U2Net [30], have more than 87 and 44 million parameters, respectively (Fig. 2). The huge number of parameters not only leads to increased consumption of computational resource, but also makes the model difficult to train. Second, though

[†]Corresponding author.

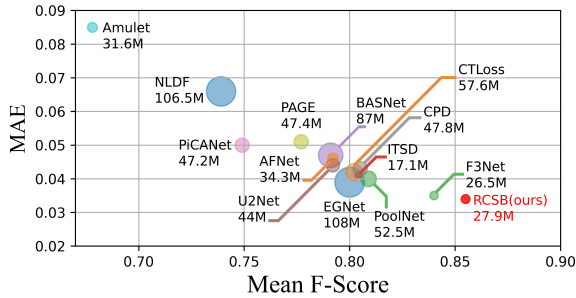


Figure 2: Mean F-score, total parameter and MAE comparison between RCSB with 13 state-of-the-art models, including EGNet [49], PiCANet [23], C2SNet [18], PoolNet [22], PAGE [37], Amulet [46], ITSD [52], BASNet [31], U2Net [30], NLDF [24], AFNet [9], F3Net [39] and CPD [40] on DUTS-TE [35] dataset. Bubble size represents model size.

object boundary is greatly improved for contour-based networks [22][49][52], predictions still have excessive false positives, as shown in Fig. 1c and 1d. Finally, to our best knowledge, for all deep learning-based models in the current field, intermediate saliency or contour predictions are generated and supervised via side branches, which introduce redundant parameters and inefficiency.

To address the abovementioned issues, we proposed a recursive contour-saliency blending network, namely, RCSBNet, for high accuracy salient object detection. We adopted a recursive CNN to reduce total trainable parameters while we can make our model deep. Unlike previous studies [22][49][52], where the contour and saliency are explicitly trained via two branches, we introduced a Contour-Saliency Blending (CSB) module in our network so that contour and saliency are intertwined and fused every step in the recursion. Meanwhile, to further improve the efficiency, we proposed a Stage-wise Feature Extraction (SFE) module to directly supervise intermediate saliency and contour predictions in the primary network without using any side branch. Lastly, we divided the training task into accuracy and confidence, and proposed Dual Confinement Loss (DCLoss) and Confidence Loss (CLoss) respectively for better model performance. To sum up, our contributions are as follows:

(1) We proposed an efficient and accurate network, RCSBNet. By using a recursive CNN and the proposed Stage-wise Feature Extraction (SFE) module, contour and saliency are fused more efficiently and effectively.

(2) We developed two loss functions, DCLoss and CLoss, to further help the boundary prediction.

(3) Our model has only 27.9 million parameters, which is significantly smaller and efficient than most of the networks in the field (Fig. 2).

(4) We conducted comprehensive evaluations on 5 widely used benchmark datasets and compared with 13 state-of-the-art methods. Our method achieves competitive state-of-the-art results among all datasets.

2. Related Work

Early approaches based on hand-crafted priors [6][12][44] have limited effectiveness and generalization ability. The very first deep salient object detection (SOD) methods [17][50] used multi-layer perceptron to predict saliency score for each image. These methods suffered from low efficiency and damage of feature structures due to flattening. Later, some studies introduced a fully convolutional network (FCN) and achieved promising results.

Recurrent Networks. In [20], a recurrent convolutional neural network (RCNN) was proposed for object detection. The main idea was to unfold the same convolution layer several times while weights are shared. It had the advantage that model depth can now be deeper by unfolding, while the total number of trainable parameters remains the same. It also revealed that, by increasing the number of recursions, better results would be obtained.

In 2016, the recurrent CNN was introduced to salient object detection task, and proposed by Wang *et al.* RFCN [36] recursively refines the saliency prediction from previous time step. Later, proposed by Kuen *et al.* [16], a recurrent network was designed to refine selected image sub-regions iteratively. In [47], Zhang *et al.* designed a multi-path recurrent model for saliency detection by transferring global information from deep layers to shallower layers. Hu *et al.* [33] proposed their salient object detection model by concatenating multi-layer deep features recurrently. It was proved that saliency predictions will be refined by using recurrent mechanism.

Utilizing Contour Information. In recent years some studies explored and verified the effectiveness of involving contour information to improve the accuracy of saliency prediction. In [5], Chen *et al.* considered boundary pixels as hard samples and proposed a contour loss, which was a weighted BCE loss, to train their network. Qin *et al.* [31] combined Structural Similarity Index (SSIM), Intersection over Union (IOU), and BCE as their contour-aware loss function to achieve better boundary quality. In another seminal work, Salient Edge Detector (SED) [37] was introduced to simultaneously generate saliency and contour predictions by using a residual structure. Furthermore, PoolNet [22] applied multi-task training and fused the contour information with saliency predictions. Later, ITSD [52] proposed a two-stream network to convert saliency and contour interactively and yielded good boundary predictions. These studies further corroborated the importance of employing contour

information to improve saliency predictions.

3. Proposed Method

3.1. Overall Architecture

In previous contour-related networks [22][49][52], either the contour was supervised in a separate branch to guide the saliency prediction, or it was fused with saliency stage by stage to achieve better boundary predictions. Both approaches gave promising results, but there are two major disadvantages: 1) *late fusion*: contours are fused with saliency at the end of each stage. 2) *limited fusion*: the number of fusion is limited by the number of U-Net stages. For *late fusion*, we designed a Contour-Saliency Blending Unit (CSBU) so that contour and saliency information can be exchanged at a much earlier stage, while for *limited fusion*, a recursive mechanism was adopted to circumvent this constraint.

As shown in Fig. 3, the proposed RCSBNet is essentially a U-Net, where we employ pre-trained ResNet-50 as our encoder and a customized decoder adopting recursive CNNs. Contour and saliency are blended in the recursion block by the Contour-Saliency Blending Unit (CSBU). Then saliency and contour features are split and fed into the Stage-wise Feature Extraction (SFE) module for supervised learning. At the last stage of the decoder, we concatenate the prediction of contour and saliency with the input image, followed by an additional recursive block, to generate the final predictions.

3.2. ResNet-50 as the Encoder

We employ the pre-trained ResNet-50 as our encoder. Since it contains a large number of feature maps in high-level blocks, same as [52], we apply channel pooling (CP) to reduce the number of channels to 64, and the operation can be expressed as:

$$CP = collect_{j \in [0, m-1]}(max_{k \in [0, \frac{n}{m}-1]} X^{j \times \frac{n}{m} + k}) \quad (1)$$

where X represents the raw feature map with total n channels and j, k are integers. We divide X into m groups then apply max-pooling and concatenate all m maps. In our experiments, m is set to 64. A convolution layer is attached after the channel pooling layer to prepare the features for further processing by the decoder, as shown in Fig. 3.

3.3. Contour-Saliency Blending Unit (CSBU)

Contour information is usually supervised and fused with saliency at the end of each U-Net stage. For us, we want to fuse the contour and saliency at earlier stages and as many times as possible. Inspired by Shuffle-Net [48], where convolution was divided into groups and information is exchanged by shuffling the weight channels, we designed our Contour-Saliency Blending Unit (CSBU). As illustrated

in Fig. 4a, the CSBU has a contour branch and saliency branch. Features generated by each branch are concatenated and blended. Mathematically, let $F_{k \times k}^{sal}$ and $F_{k \times k}^{ctr}$ denote the convolution for saliency and contour with kernel size k respectively, while (C_{n-1}, S_{n-1}) and (C_n, S_n) represent contour and saliency features before and after CSBU. For simplicity, we omit ReLU [2] and BatchNorm [11] in our formula, and then the proposed CSBU can be modeled as:

$$CTR = F_{3 \times 3}^{ctr}(C_{n-1}) \quad (2)$$

$$SAL = F_{3 \times 3}^{sal}(S_{n-1}) \quad (3)$$

$$C_n = F_{3 \times 3}^{ctr}(F_{1 \times 1}^{ctr}([CTR, SAL])) \quad (4)$$

$$S_n = F_{3 \times 3}^{sal}(F_{1 \times 1}^{sal}([CTR, SAL])) \quad (5)$$

and thus:

$$S_n, C_n = CSBU(S_{n-1}, C_{n-1}) \quad (6)$$

By applying CSBU, contour and saliency information can be utilized by the network at a much earlier stage.

3.4. Recursive Block

We now introduce more details of the recursive block, as illustrated in Fig. 4b. Firstly let us consider a single recursive block then we extend to more general cases. We denote R as the total number of recursions in a single block, and $f^r, r \in [1, R]$ denote the r^{th} recursion of the CSBU. Based on Eq. 6, let X_{n-1} denote the input tuple of (S_{n-1}, C_{n-1}) , and X_R represent the output of the R^{th} recursion, then we have:

$$X_R = f^R(\dots(f^2(f^1(X_{n-1}) + X_{n-1}) + X_{n-1}) + \dots + X_{n-1}) \quad (7)$$

note that weights for CSBU are shared in recursion, though they have different superscripts in the formula. Then we apply a single convolution layer, denoted as $F_{3 \times 3}$, at the end of recursion with skip connection. Thus the output of our recursive block, X_n , is:

$$X_n = F_{3 \times 3}(X_R + X_{n-1}) + X_{n-1} = h_b(X_{n-1}) \quad (8)$$

where h_b represents the recursive block function.

Now let G represent the total number of recursive blocks, as shown in Fig. 4c, in our network we simply stack all the recursive blocks together. Thus given the input R_{n-1} , a tuple of (S_{n-1}, C_{n-1}) , the output of g -th block, R_n , is:

$$R_n = h_b^g(h_b^{g-1}(\dots h_b^2(h_b(R_{n-1})))) \quad (9)$$

where h_b^g represents the g -th recursive block function.

By applying recursive blocks and stacking them together, contour and saliency can now be fused $G \times R$ times at each stage of the U-Net, which will improve the network performance significantly. We will show more results in Sec. 4.

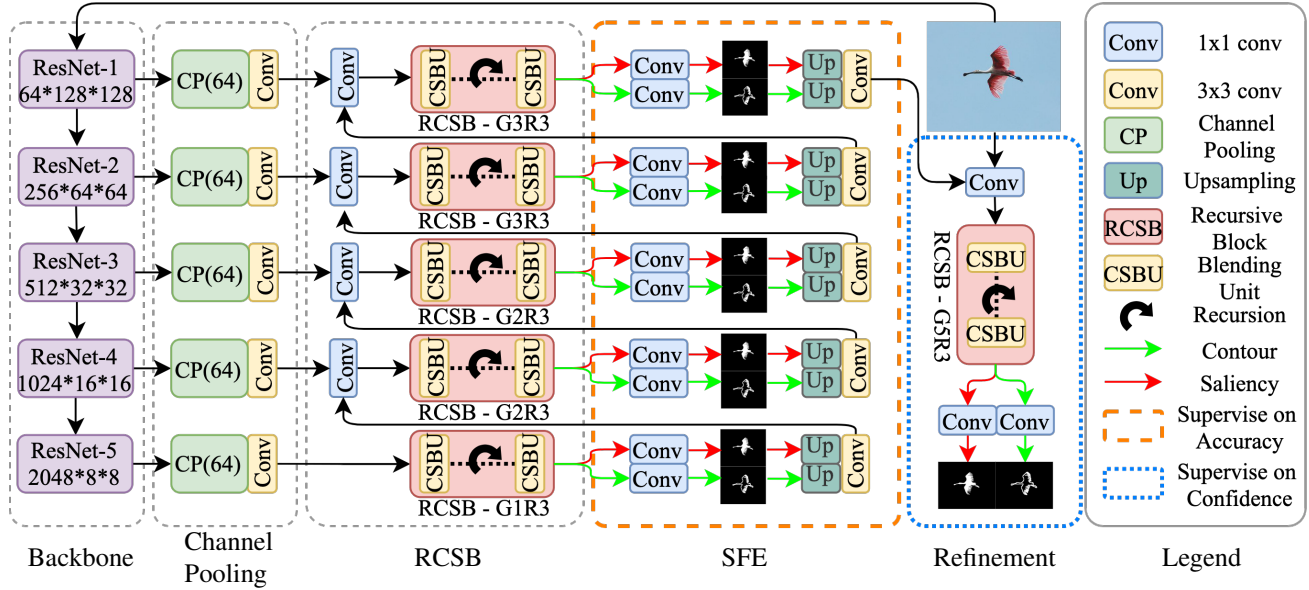


Figure 3: Network architecture for RCSBNet. Pre-trained ResNet-50 is used as the backbone, channel number is reduced to 64 via Channel Pooling (CP) layer. Recursive Contour-Saliency Blocks (RCSB) with G blocks and R recursions are then attached and followed by Stage-wise Feature Extraction (SFE) module to generate contour and saliency predictions. At the end of the network, a refinement module is adopted to further refine the predictions.

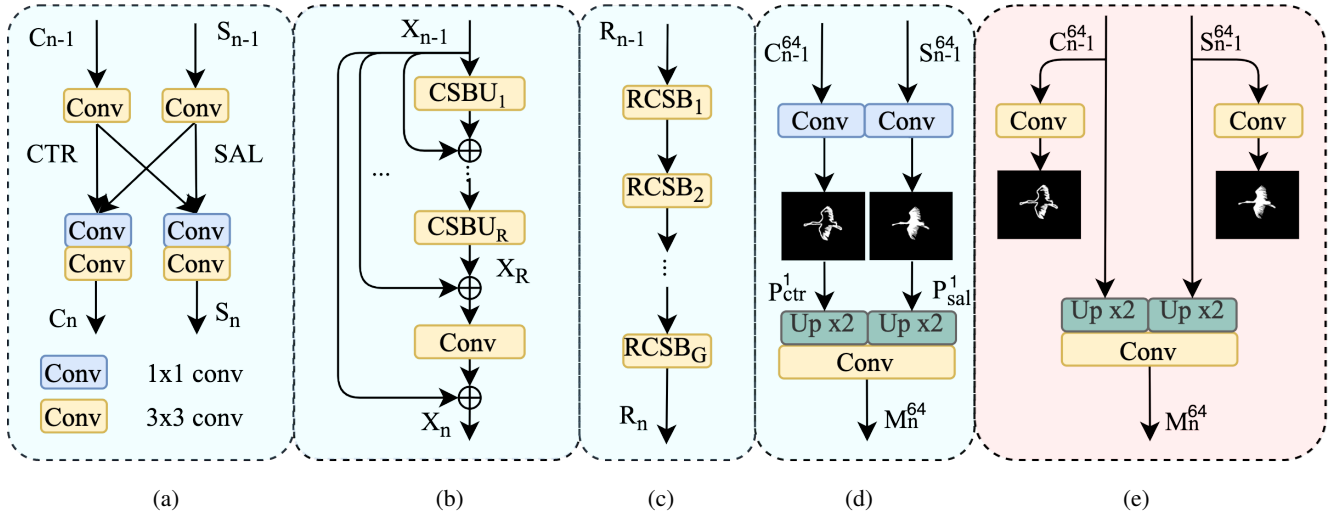


Figure 4: (a) Contour-Saliency Blending Unit (CSBU). It contains two streams where contour and saliency information are blended and intertwined. (b) Single Recursive Contour-Saliency Blending (RCSB) Block. In order to increase the number of contour-saliency fusion, recursive mechanism is applied. Weights are shared among all CSBU blocks used in the RCSB. (c) Network branch contains G blocks of RCSB, each with R times of recursion. (d) Stage-wise Feature Extraction (SFE) module. (e) Conventional methods for generating intermediate stage predictions by using side branches.

3.5. Stage-wise Feature Extraction (SFE) Module

It is prevalent that saliency networks are densely supervised, where intermediate stage features are generated and supervised against ground truths to guide the model for better convergence. Common practices create a side branch

with a few convolution layers to generate predictions from current U-Net stage features, following which the stage features are passed on to the next U-Net stage in the primary network (Fig. 4e). Different from others, we regard stage predictions as the best result the network can obtain so far

and apply a new round of feature extraction based on current stage predictions. Therefore, stage predictions are now in the network’s main branch, acting as a single channel layer, and supervised against ground truths, as illustrated in Fig. 4d. By doing so, the next block is expected to extract valuable features from current best results and discard all useless features, which might otherwise be carried along by recursion and residual connections.

To generate stage predictions, unlike previous studies [3][22][52], we do not use max-pooling because it will bring false positives to the next stage, nor the element-wise multiplication between contour and saliency like SCRN [41] due to the introduction of false negatives. Instead, we employ a 1×1 convolution and a scaling factor to help sigmoid function classify saliency and background. To formulate the SFE module mathematically, let C^i and S^i represent incoming i channels of feature maps, and K represent the scaling factor learned by the network, then stage prediction P^j with j channels can be expressed as:

$$P_{sal}^1 = F_{1 \times 1}^{sal}(S_{n-1}^{64}) * K_{sal} \quad (10)$$

$$P_{ctr}^1 = F_{1 \times 1}^{ctr}(C_{n-1}^{64}) * K_{ctr} \quad (11)$$

where $*$ represents element-wise multiplication. For extracted feature maps M_n^j with j channels:

$$M_n^{64} = F_{3 \times 3}(Up^{\times 2}[\sigma(P_{sal}^1), \sigma(P_{ctr}^1)]) \quad (12)$$

where $[]$ stands for concatenating, $Up^{\times 2}$ is up-sampling, and σ stands for sigmoid function. SFE is a simple module but it is very effective, we will demonstrate more in ablation studies.

3.6. Loss Functions

Unsure prediction is very common in salient object detection in the form of shady areas (Fig. 5b). Many studies try to improve the performance by focusing on hard pixels near the boundary [5][39]. Boundary pixels are indeed hard samples, but not all the hard samples are near the boundary. We notice that the network can correctly predict the saliency for some images, but with low pixel values; while for some other images, the network is confident of generating false negatives (FN) or false positives (FP), as illustrated in Fig. 5e and 5f. It points out two kinds of difficulties encountered by the network: 1) unconfident but accurate predictions and 2) confident but inaccurate predictions. Hence in addition to accuracy, we factor in the confidence of predictions, and we introduce Confidence Loss (CLoss) to our training.

Confidence Loss. Recently focal loss [21][38] has been explored in saliency tasks due to its high weight on wrong predictions: $W_{focal} = \alpha(\hat{y}_{i,j} - x_{i,j})^\gamma$. However, focal loss becomes less sensitive as predictions approach ground truth, which eventually leaves a large area of unsure predictions.

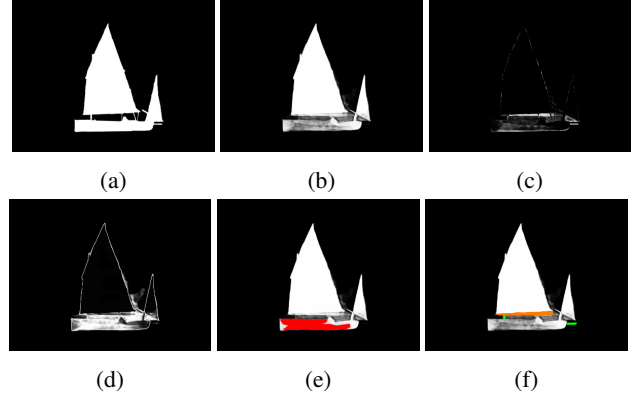


Figure 5: (a) ground truth saliency. (b) predicted saliency. (c) focal loss ($\alpha, \gamma = 2, 2$) weight map calculated using (a) and (b). (d) confidence loss ($\beta, \lambda = 2, 1$) weight map calculated using (a) and (b). (e) unconfident but correct area (red) in prediction. (f) confident false positives (orange) and false negatives (green) in prediction. Compare (d) with (c), focal loss is less sensitive to unsure predictions, while confidence loss generates high weights especially on object boundaries, which eventually help the model generate sharper edges.

In order to guide the model focus more on the unconfident predictions, we propose a confidence score, W_c , for each predicted pixel $x_{i,j}$:

$$W_c = \beta * x_{i,j} * (1 - x_{i,j}) \quad (13)$$

where β is empirically set to 2, and $x_{i,j}$ is the prediction after sigmoid. When $x_{i,j} = 0.5$ the score reaches its maximum. Then with ground truth $\hat{y}_{i,j}$, our confidence loss (CLoss), \mathcal{L}_{con} , is defined as:

$$\mathcal{L}_{con}(\beta, \lambda) = -\frac{1}{n} \sum_{i=1}^n [W_c * BCE(x_{i,j}, \hat{y}_{i,j}) + \lambda W_c] \quad (14)$$

where λ is set to 0.3 by parameter search. By applying this loss, it will encourage the model to make more confident predictions into either foreground (close to 1) or background (close to 0).

Dual Confinement Loss. Most of the contour-based models use a separate contour branch to guide saliency predictions. In [5], and [52], a weight map generated from contour ground truth is applied to BCE loss to improve saliency boundary quality while the contour branch is supervised by BCE loss without any weights. It is reasonable because contour information is from a separate branch and is only used to guide saliency. Since in our network, saliency and contour streams are going to exchange information; thus we not only use contour to guide saliency predictions but also want to use saliency

to guide contour predictions. Based on this, we designed our Dual Confinement Loss (DCLoss), and losses for each stream, \mathcal{L}_{sal} and \mathcal{L}_{ctr} , are defined as:

$$\mathcal{L}_{sal} = -\frac{1}{n} \sum_{i=1}^n [W_{sal} * BCE(x_i^{sal}, \hat{y}_i^{sal})] \quad (15)$$

$$\mathcal{L}_{ctr} = -\frac{1}{n} \sum_{i=1}^n [W_{ctr} * BCE(x_i^{ctr}, \hat{y}_i^{ctr})] \quad (16)$$

where weight matrix W_{sal} and W_{ctr} are calculated by:

$$W_{sal} = \max(x_i^{ctr}, \hat{y}_i^{ctr}) * \theta + 1 \quad (17)$$

$$W_{ctr} = \max(x_i^{sal}, \hat{y}_i^{sal}) * \theta + 1 \quad (18)$$

where x, \hat{y} stand for prediction and ground truth. Max-pooling is used because higher weights need to be applied when x is not close to y . Empirically, we set $\theta = 4$ in our experiments. Then the DCLoss, \mathcal{L}_{DC} , is defined as:

$$\mathcal{L}_{DC} = \mathcal{L}_{sal} + \mathcal{L}_{ctr} \quad (19)$$

Loss for Training. Compared with low-level predictions, it should be relatively easy to emphasize accuracy on high-level predictions due to its small feature dimension. Meanwhile, a failure in high-level predictions will impact its following decoders and eventually cause false positives or negatives. Thus, we train stages 1 to 5 of our network against accuracy-related losses, i.e. DCLoss and weighted IOU loss mentioned in [39], and train the refinement module against CLoss only. Our loss functions for saliency and contour are defined as:

$$\mathcal{L}_{saliency} = \mathcal{L}_{DC}^{1-5} + \mathcal{L}_{wIOU}^{1-5} + \mathcal{L}_{con}^{ref}(\beta, \lambda = 2, 0.3) \quad (20)$$

$$\mathcal{L}_{contour} = \mathcal{L}_{DC}^{1-5} + \mathcal{L}_{con}^{ref}(\beta, \lambda = 2, 0.3) \quad (21)$$

where superscripts 1 – 5 and *ref* represent the five decoder stages and refinement module in Fig. 3.

4. Experiments

4.1. Datasets

We use DUT-OMRON [44](5168 images), ECSSD [43](1000 images), PASCAL-S [19](850 images), HKUIS [17](4447 images), and DUTS-TE [35](5019 Images) as our evaluation datasets.

4.2. Implementation Details

DUTS-TR [35] is used for training with input images resized to 256×256, then random horizontal flipping and 90° rotation are applied as the augmentation. We use pre-trained ResNet-50 [10] as the encoder. The number of recursive blocks G is set to (1,2,2,3,3,5) with recursion $R = 3$ for the network (Fig. 3) by parameter search.

Besides, we apply Leaky ReLU [13] and adopt the Adam optimizer [14] with default hyperparameters to train our network. Learning rates for encoder and decoder are set to 10^{-5} and 10^{-4} respectively, and they are halved every 20 epochs with a total of 100 epochs using a batch size of 4. During testing, images are resized to 256×256, and the predictions (256×256) are resized back to their original size by using bilinear interpolation. We use PyTorch [29] and a single RTX 3090 GPU for our model and experiments. Code is available at: <https://github.com/BarCodeReader/RCSB-PyTorch>.

4.3. Evaluation Metrics

Precision-Recall (PR) curve, F_β -measure [1], Mean Absolute Error (MAE), F_β^ω -measure [26], and E_ξ -measure [7] are adopted in our experiments.

PR-Curve. By applying different thresholds from 0 to 255, PR curve is obtained by comparing the ground truth masks against the binarized saliency predictions.

F-measure. The F_β -measure is calculated by precision and recall value of saliency maps: $F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}$ where β^2 is set to 0.3 [1]. We report the average score over all thresholds from 0 to 255 and denote as \bar{F}_β [24][45].

MAE. MAE is the mean value of the sum of pixel-wise absolute differences between predictions x and ground truths \hat{y} : $MAE = \frac{1}{n} \sum_{i=1}^n |x_i - \hat{y}_i|$.

Weighted F-measure. F_β^ω uses weighted precision and weighted recall to measure both exactness and completeness of the prediction against ground truth. It is designed to improve the existing F_β -measure.

E-measure. By using local pixel values and the image-wise mean, E_ξ calculates the similarity between the prediction and the ground truth.

4.4. Comparisons with State-of-the-art Results

We compare our results with 13 state-of-the-art salient object detection networks, including EGNNet [49], PoolNet [22], ITSD [52], AFNet [9], PAGE [37], CPD [40], BASNet [31], CAGNet [27], GateNet [51], U2Net [30], GCPA [4], MINet [28], and F3Net [39]. Saliency maps used are provided by authors.

Quantitative Evaluation. To compare our work with the state-of-the-art networks, detailed experimental results in terms of four metrics are listed in Table 1. Among all the models, RCSBNet achieves outstanding results across all four metrics on most datasets. Besides, PR and F-measure curves are demonstrated in Fig. 7. Our F-measure curves are flatter than all other models, which reveals that our results are closer to binary predictions and invariant to threshold changes.

Qualitative Evaluation. Visual comparisons are listed in Fig. 6. Compare with other contour-based network

Table 1: Quantitative comparisons between RCSBNet and other 13 methods on five benchmark datasets in terms of the average F-measure \bar{F}_β , MAE M , E_ξ and F_β^ω . \uparrow/\downarrow means the larger/smaller the value, the better the results. **Red**, **Green**, and **Blue** indicate the best, second best and third best performance.

Method	DUTS-TE				HKU-IS				PASCAL-S				ECSSD				DUT-OMRON			
	$\bar{F}_\beta \uparrow$	$M \downarrow$	$E_\xi \uparrow$	$F_\beta^\omega \uparrow$	$\bar{F}_\beta \uparrow$	$M \downarrow$	$E_\xi \uparrow$	$F_\beta^\omega \uparrow$	$\bar{F}_\beta \uparrow$	$M \downarrow$	$E_\xi \uparrow$	$F_\beta^\omega \uparrow$	$\bar{F}_\beta \uparrow$	$M \downarrow$	$E_\xi \uparrow$	$F_\beta^\omega \uparrow$	$\bar{F}_\beta \uparrow$	$M \downarrow$	$E_\xi \uparrow$	$F_\beta^\omega \uparrow$
Contour-based Methods																				
EGNet ₁₉ [49]	.815	.039	.891	.816	.901	.031	.950	.887	.817	.073	.848	.795	.920	.037	.927	.903	.755	.053	.867	.725
PoolNet ₁₉ [22]	.819	.037	.896	.817	.903	.030	.953	.889	.826	.064	.852	.809	.919	.035	.925	.904	.752	.054	.868	.725
ITSD ₂₀ [52]	.804	.041	.895	.824	.899	.031	.952	.894	.785	.065	.850	.812	.895	.034	.927	.911	.756	.061	.863	.750
Ours	.855	.034	.903	.840	.923	.027	.954	.909	.842	.058	.852	.816	.927	.033	.923	.916	.773	.045	.855	.752
Non-Contour-based Methods																				
AFNet ₁₉ [9]	.793	.046	.879	.785	.889	.036	.942	.872	.828	.078	.846	.804	.908	.042	.918	.886	.739	.057	.853	.717
PAGE ₁₉ [37]	.777	.051	.854	.769	.884	.037	.940	.868	.817	.078	.835	.792	.906	.042	.920	.886	.736	.066	.853	.722
CPD ₁₉ [40]	.805	.043	.886	.795	.891	.034	.944	.876	.831	.072	.849	.803	.917	.037	.924	.898	.747	.056	.866	.719
BASNet ₁₉ [31]	.791	.047	.884	.803	.898	.032	.946	.890	.781	.076	.847	.800	.879	.037	.921	.904	.756	.056	.869	.751
CAGNet ₂₀ [27]	.837	.040	.897	.817	.909	.030	.945	.893	.833	.066	.857	.808	.921	.037	.916	.902	.752	.054	.856	.728
GateNet ₂₀ [51]	.806	.040	.889	.809	.898	.033	.949	.879	.819	.068	.852	.797	.916	.040	.924	.894	.746	.055	.862	.729
U2Net ₂₀ [30]	.792	.045	.886	.804	.896	.031	.948	.889	.770	.076	.841	.792	.892	.033	.924	.910	.761	.054	.870	.751
GCPA ₂₀ [4]	.817	.038	.891	.821	.898	.031	.949	.888	.826	.061	.847	.808	.919	.035	.920	.903	.748	.056	.860	.734
MINet ₂₀ [28]	.828	.037	.898	.825	.909	.029	.953	.897	.829	.063	.851	.809	.924	.033	.927	.911	.755	.055	.865	.738
F3Net ₂₀ [39]	.839	.035	.902	.835	.909	.028	.953	.900	.835	.062	.859	.816	.925	.033	.927	.912	.766	.053	.870	.747
Ours	.855	.034	.903	.840	.923	.027	.954	.909	.842	.058	.852	.816	.927	.033	.923	.916	.773	.045	.855	.752

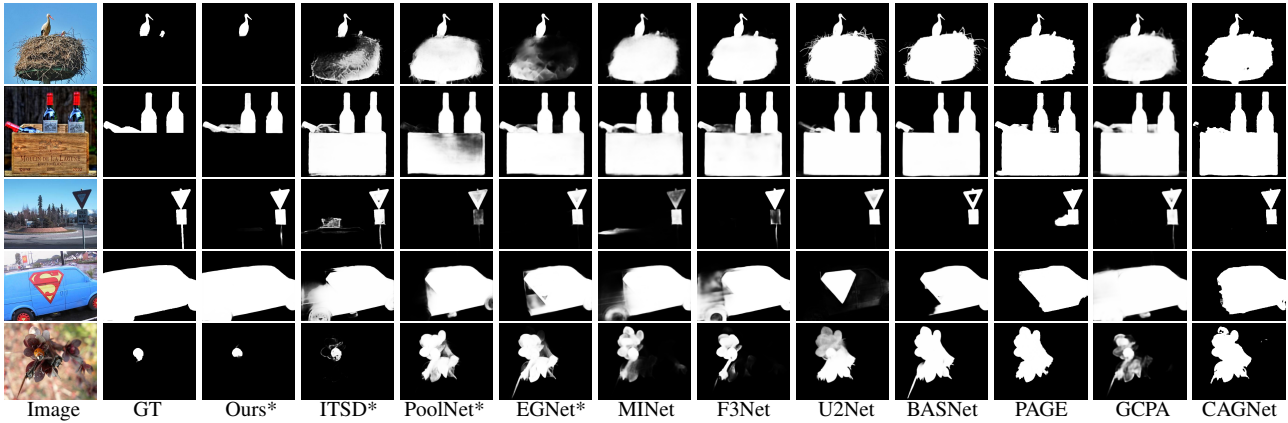


Figure 6: Visual comparisons between our method and 10 state-of-the-art networks. * stands for models utilizing contour information. More comparisons are provided in the supplementary material.

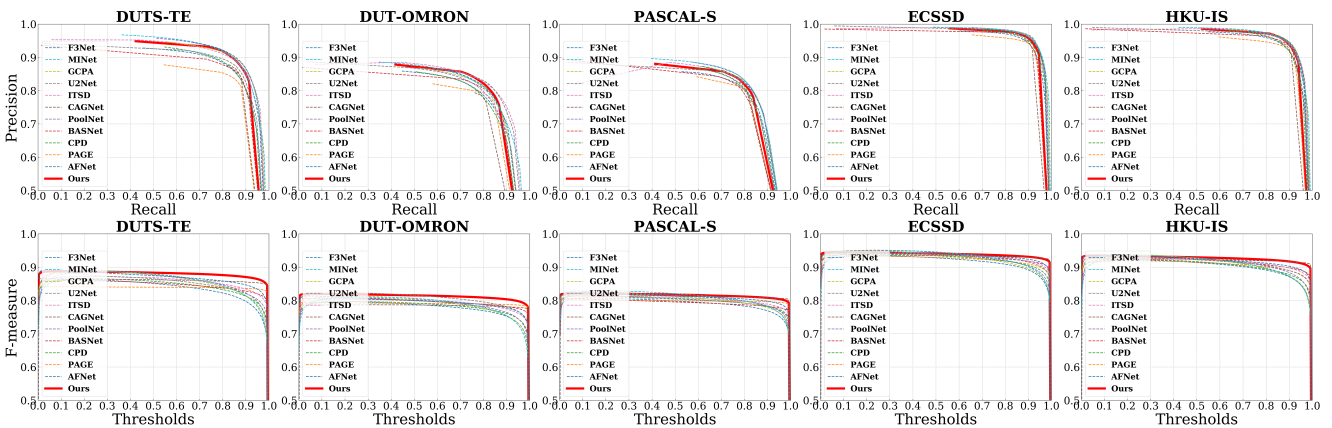
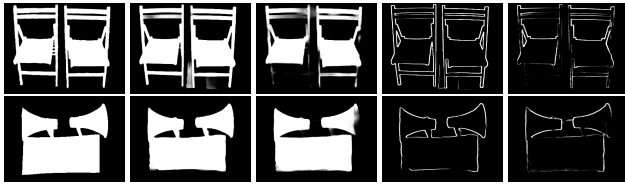


Figure 7: First row: Precision-Recall Curves comparison on five saliency benchmark datasets. Second row: F-measure Curves comparison on five saliency benchmark datasets.

results, our method yields better boundary predictions. As shown in the graph, our model can produce accurate and complete saliency maps with better edges.

4.5. Ablation Studies

Effectiveness of the early fusion (EF). To prove that early fusion of contour and saliency information will boost model performance, an experiment was conducted by removing the fusion branch illustrated in Fig. 4a and make it into two separate streams. We list both quantitative and qualitative measures between the two approaches on DUTS-TE and ECSSD datasets, as shown in Table 2 and Fig. 8.



(a) GT (b) w/ EF (c) w/o EF (d) w/ EF (e) w/o EF

Figure 8: Qualitative comparisons for early fusion. (a) Ground-truth. (b) & (d) Saliency and contour prediction with early fusion. (c) & (e) Saliency and contour prediction without early fusion. As can be seen with early fusion, results are more complete for both contour and saliency predictions.

Table 2: Quantitative comparisons for the effectiveness of early fusion.

	DUTS-TE				ECSSD			
	$\overline{F_\beta} \uparrow$	$M \downarrow$	$E_\xi \uparrow$	$F_\beta^\omega \uparrow$	$\overline{F_\beta} \uparrow$	$M \downarrow$	$E_\xi \uparrow$	$F_\beta^\omega \uparrow$
w/ early fusion	.855	.034	.903	.840	.927	.033	.923	.916
w/o early fusion	.849	.037	.897	.833	.925	.035	.919	.908

Effectiveness of refinement module and supervised on confidence. To study the importance of the refinement module and prove the effectiveness of supervision on confidence, we conducted 4 experiments on DUTS-TE and ECSSD datasets covering all the cases for our comparison, as listed in Table 3. For conciseness, we denote reference module and supervision on confidence as *Ref.* and *Conf.*.

It can be observed that each approach will boost the performance, and when they are combined, we obtained the best results.

Effectiveness of SFE module and different loss functions. To study the importance of each loss function and SFE module, we conduct a series of controlled experiments on the DUTS-TE dataset. We train the model by using BCE loss only, then include weighted IOU Loss [39], DCLoss, and CLoss step by step. Detailed results are listed in Table 4.

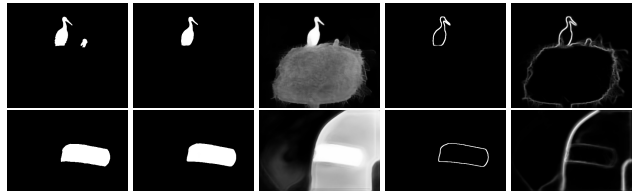
Table 3: Quantitative comparisons for different model configurations.

Ref.	Conf.	DUTS-TE				ECSSD			
		$\overline{F_\beta} \uparrow$	$M \downarrow$	$E_\xi \uparrow$	$F_\beta^\omega \uparrow$	$\overline{F_\beta} \uparrow$	$M \downarrow$	$E_\xi \uparrow$	$F_\beta^\omega \uparrow$
\times	\times	.842	.039	.849	.825	.916	.037	.914	.902
\checkmark	\times	.848	.038	.861	.830	.920	.036	.916	.908
\times	\checkmark	.849	.037	.870	.837	.922	.035	.921	.909
\checkmark	\checkmark	.855	.034	.903	.840	.927	.033	.923	.916

Table 4: Ablation study for different loss functions and presence of SFE module.

BCE	wIOU	DCLoss	CLoss	SFE	DUTS-TE			
					$\overline{F_\beta} \uparrow$	$M \downarrow$	$E_\xi \uparrow$	$F_\beta^\omega \uparrow$
\checkmark					.788	.058	.862	.776
\checkmark	\checkmark				.793	.047	.881	.789
	\checkmark	\checkmark			.829	.043	.890	.813
	\checkmark	\checkmark	\checkmark		.847	.040	.896	.825
	\checkmark	\checkmark	\checkmark	\checkmark	.855	.034	.903	.840

To prove the effectiveness of SFE module, qualitative comparisons are illustrated in Fig. 9. As can be seen, with SFE module, model can effectively suppress excessive wrong predictions. Though SFE is a simple feature extraction module, it improves overall performance.



(a) GT (b) w/ SFE (c) w/o SFE (d) w/ SFE (e) w/o SFE

Figure 9: Qualitative comparisons for SFE module. (a) Ground-truth. (b) & (d) Saliency and contour prediction with SFE module. (c) & (e) Saliency and contour prediction without SFE module.

5. Conclusions

In this paper, we have introduced an efficient and accurate model using a recursive CNN together with a Contour-Saliency Blending (CSB) module. To further improve the model's efficiency, a Stage-wise Feature Extraction (SFE) module is adopted. It is a simple module but can suppress wrong predictions effectively. Furthermore, we divided the training objectives into accuracy and confidence and proposed two loss functions to guide model convergence. The predicted salient objects achieved comparable state-of-the-art results on five benchmark datasets.

References

- [1] Radhakrishna Achanta, Sheila S. Hemami, Francisco J. Estrada, and Sabine Süstrunk. Frequency-tuned salient region detection. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1597–1604. IEEE Computer Society, 2009.
- [2] Abien Fred Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018.
- [3] Quan Chen, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. Semantic human matting. In Susanne Boll, Kyoung Mu Lee, Jiebo Luo, Wenwu Zhu, Hyeran Byun, Chang Wen Chen, Rainer Lienhart, and Tao Mei, editors, *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018*, pages 618–626. ACM, 2018.
- [4] Zuyao Chen, Qianqian Xu, Runmin Cong, and Qingming Huang. Global context-aware progressive aggregation network for salient object detection. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 10599–10606. AAAI Press, 2020.
- [5] Zixuan Chen, Huajun Zhou, Jianhuang Lai, Lingxiao Yang, and Xiaohua Xie. Contour-aware loss: Boundary-aware learning for salient object segmentation. *IEEE Trans. Image Process.*, 30:431–443, 2021.
- [6] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):569–582, 2015.
- [7] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 698–704. ijcai.org, 2018.
- [8] Hao Fang, Saurabh Gupta, Forrest N. Iandola, Rupesh Kumar Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. From captions to visual concepts and back. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1473–1482. IEEE Computer Society, 2015.
- [9] Mengyang Feng, Huchuan Lu, and Errui Ding. Attentive feedback network for boundary-aware salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1623–1632. Computer Vision Foundation / IEEE, 2019.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.
- [11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015.
- [12] Bowen Jiang, Lihe Zhang, Huchuan Lu, Chuan Yang, and Ming-Hsuan Yang. Saliency detection via absorbing markov chain. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 1665–1672. IEEE Computer Society, 2013.
- [13] Muhammad Khalid, Junaid Baber, Mumraiz Khan Kasi, Maheen Bakhtyar, Varsha Devi, and Naveed Sheikh. Empirical evaluation of activation functions in deep convolution neural network for facial expression recognition. In *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pages 204–207, 2020.
- [14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012.
- [16] Jason Kuen, Zhenhua Wang, and Gang Wang. Recurrent attentional networks for saliency detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3668–3677. IEEE Computer Society, 2016.
- [17] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 5455–5463. IEEE Computer Society, 2015.
- [18] Xin Li, Fan Yang, Hong Cheng, Wei Liu, and Dinggang Shen. Contour knowledge transfer for salient object detection. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, volume 11219 of *Lecture Notes in Computer Science*, pages 370–385. Springer, 2018.
- [19] Yin Li, Xiaodi Hou, Christof Koch, James M. Rehg, and Alan L. Yuille. The secrets of salient object segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 280–287. IEEE Computer Society, 2014.

- [20] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3367–3375. IEEE Computer Society, 2015.
- [21] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007. IEEE Computer Society, 2017.
- [22] Jiangjiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3917–3926. Computer Vision Foundation / IEEE, 2019.
- [23] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3089–3098. IEEE Computer Society, 2018.
- [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 *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6593–6601. IEEE Computer Society, 2017.
- [25] Yu-Fei Ma, Lie Lu, HongJiang Zhang, and Mingjing Li. A user attention model for video summarization. In Lawrence A. Rowe, Bernard Merialdo, Max Mühlhäuser, Keith W. Ross, and Nevenka Dimitrova, editors, *Proceedings of the 10th ACM International Conference on Multimedia 2002, Juan les Pins, France, December 1-6, 2002*, pages 533–542. ACM, 2002.
- [26] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 248–255. IEEE Computer Society, 2014.
- [27] Sina Mohammadi, Mehrdad Noori, Ali Bahri, Sina Ghofrani Majelan, and Mohammad Havaei. Cagnet: Content-aware guidance for salient object detection. *Pattern Recognit.*, 103:107303, 2020.
- [28] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Multi-scale interactive network for salient object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9410–9419. IEEE, 2020.
- [29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019.
- [30] Xuebin Qin, Zichen Vincent Zhang, Chenyang Huang, Masood Dehghan, Osmar R. Zaiane, and Martin Jägersand. U²-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognit.*, 106:107404, 2020.
- [31] Xuebin Qin, Zichen Vincent Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jägersand. Basnet: Boundary-aware salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 7479–7489. Computer Vision Foundation / IEEE, 2019.
- [32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.
- [33] Chang Tang, Xinzhong Zhu, Xinwang Liu, and Pichao Wang. Salient object detection via recurrently aggregating spatial attention weighted cross-level deep features. In *IEEE International Conference on Multimedia and Expo, ICME 2019, Shanghai, China, July 8-12, 2019*, pages 1546–1551. IEEE, 2019.
- [34] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.
- [35] 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 *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3796–3805. IEEE Computer Society, 2017.
- [36] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 825–841, Cham, 2016. Springer International Publishing.
- [37] Wenguan Wang, Shuyang Zhao, Jianbing Shen, Steven C. H. Hoi, and Ali Borji. Salient object detection with pyramid attention and salient edges. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1448–1457. Computer Vision Foundation / IEEE, 2019.
- [38] Yupei Wang, Xin Zhao, Xuecai Hu, Yin Li, and Kaiqi Huang. Focal boundary guided salient object detection. *IEEE Trans. Image Process.*, 28(6):2813–2824, 2019.
- [39] Jun Wei, Shuhui Wang, and Qingming Huang. F3net: Fusion, feedback and focus for salient object detection. *CoRR*, abs/1911.11445, 2019.
- [40] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection.

- In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3907–3916. Computer Vision Foundation / IEEE, 2019.
- [41] Zhe Wu, Li Su, and Qingming Huang. Stacked cross refinement network for edge-aware salient object detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 7263–7272. IEEE, 2019.
- [42] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org, 2015.
- [43] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 1155–1162. IEEE Computer Society, 2013.
- [44] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 3166–3173. IEEE Computer Society, 2013.
- [45] Lu Zhang, Ju Dai, Huchuan Lu, You He, and Gang Wang. A bi-directional message passing model for salient object detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1741–1750. IEEE Computer Society, 2018.
- [46] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 202–211. IEEE Computer Society, 2017.
- [47] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 714–722. IEEE Computer Society, 2018.
- [48] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6848–6856. IEEE Computer Society, 2018.
- [49] Jiaying Zhao, Jiangjiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnnet: Edge guidance network for salient object detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 8778–8787. IEEE, 2019.
- [50] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1265–1274. IEEE Computer Society, 2015.
- [51] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12347 of *Lecture Notes in Computer Science*, pages 35–51. Springer, 2020.
- [52] Huajun Zhou, Xiaohua Xie, Jian-Huang Lai, Zixuan Chen, and Lingxiao Yang. Interactive two-stream decoder for accurate and fast saliency detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9138–9147. IEEE, 2020.
- [53] Jun-Yan Zhu, Jiajun Wu, Yan Xu, Eric I-Chao Chang, and Zhuowen Tu. Unsupervised object class discovery via saliency-guided multiple class learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(4):862–875, 2015.
- [54] Mingchen Zhuge, Dehong Gao, Deng-Ping Fan, Linbo Jin, Ben Chen, Haoming Zhou, Minghui Qiu, and Ling Shao. Kaleido-bert: Vision-language pre-training on fashion domain. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12647–12657. Computer Vision Foundation / IEEE, 2021.