IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 13, 2021, accepted December 23, 2021, date of publication December 27, 2021,
date of current version December 31, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3138782

Salient Object Detection: An Accurate and Efficient Method for Complex Shape Objects

MIN QIAO, GANG ZHOU, QIU LING LIU, AND LI ZHANG
College of Information Science and Engineering, Xinjiang University, Ürümqi 830046, China

Corresponding author: Gang Zhou (gangzhou_xju@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62166040 and Grant U1803261, and in part by the Natural Science Foundation of XinJiang under Grant 2021D01C057.

ABSTRACT Deep learning-based salient object detection (SOD) methods have made great progress in recent years. However, most deep learning-based methods suffer from coarse object boundaries and expensive computations, especially in detecting objects with complex shapes. This paper presents an accurate and efficient SOD method that is based on a novel double-branch network that includes a body branch and an edge branch. To obtain an accurate edge, an edge profile enhancement module (EPEM) is embedded in the edge branch. In addition, a fusion feedback module (FFM) is embedded to integrate features from the two branches. To address the problem of expensive computations, channel attention module (CAM) is included to restrain redundant feature channels. Thus, the speed of the inference step can be improved with little reduction in the boundary accuracy. Experimental results on 9 datasets demonstrate that the proposed method performs favorably against 8 state-of-the-art methods in terms of both accuracy and efficiency. Additionally, our method achieves excellent detection results for objects with complex shapes.

INDEX TERMS Salient object detection, complex shape object, edge profile enhancement module, channel attention module, fusion feedback module.

I. INTRODUCTION

Salient object detection aims to locate the principal objects in an image or video. It is widely applied as a preprocessing procedure in computer vision tasks, including image compression [1]–[3], image segmentation [4], [5], image recognition [6], [7], image classification [8], [9], image retrieval [10], object tracking [11], scene classification [12], video segmentation [13] and action detection [14]. Most previous works were based on the contrast, such as color contrast [15] and global contrast [16], play the most important role in saliency detection. These methods can locate salient objects roughly, but failed in irregular shapes or low contrast.

Convolutional neural networks (CNN) have powerful representation capability in handling the large appearance variations of scenes and objects [17]–[20], and have significantly boosted salient object detection results in the past few years. In general, these methods utilize both highly semantic features from deep CNN layers and low-level features from shallow CNN layers. These CNN-based methods overtook the leaderboards for almost all of the widely used benchmarks. However, there are still two issues need to be improved for SOD. The first issue is that most methods suffer from the coarse boundary problem. When there are insufficient low-level features, it is infeasible to locate salient objects accurately, particularly for objects with complex shapes. Objects with complex shapes usually have irregular shapes, uneven contours, thin lines or nonconvex boundaries. The complexity of the object is defined as C in Eq. (8). As shown in Fig. 1, when detecting objects with simple shapes ($C < 100$), previously established methods have achieved good performance. However, when detecting objects with complex shapes ($C > 100$), these methods cannot predict the object contours accurately. These representative SOD methods are not efficient enough for implementing SOD as a preprocessing step for subsequent high-level tasks.

Significant efforts have been made toward addressing the first issue. [18], [41] focused on detecting salient object boundaries accurately by extracting edge features. Feng *et al.* [21] employed a dilated convolutional pyramid pooling module to generate a coarse prediction and to guide the residual learning process through several novel attentional

The associate editor coordinating the review of this manuscript and approving it for publication was Yizhang Jiang.

residual modules to gradually refine the coarse prediction. Liu *et al.* [22] designed a three-task network that can detect the object, body and edge at the same time and improve the overall detection performance. Wang *et al.* [23] proposed a salient edge detection module that provides a strong cue for better segmenting salient objects and refining object boundaries, which emphasizes the importance of salient edge information. Zhang *et al.* [24] embedded edge-aware feature maps from shadow layers into the deep learning framework. Mukherjee *et al.* [25] constructed a Bayesian probabilistic edge mapping that uses low-order edge features to assign a saliency value to each edge. However, edge preservation still has not been solved well, and specific guidance for the edge area is lacking. The complementarity between the salient edge information and the salient object information has not been identified. Regarding the second issue, some models [26], [27] focus mainly on improving the detection efficiency by simplifying the network. Li *et al.* [28] proposed a novel depthwise nonlocal module that implicitly models contrast by harvesting intrachannel and interchannel correlations to improve the detection efficiency. Zhang *et al.* [29] proposed fast salient object detection based on multiscale feature aggression, which processes images at a rate of 15 FPS. Although the above detection methods have made improvements, the accuracy of the detection results still needs to be further improved.

We propose a novel network that solves both problems. The body branch of the network detects the body of the object to locate the position of the salient object. The edge branch detects the edge profiles, which describe the object with a precise boundary. Because the proposed model is smaller than other models, its calculation efficiency is significantly higher.

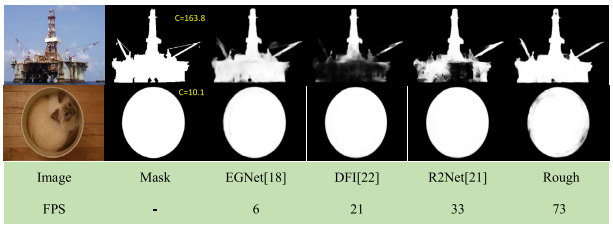


FIGURE 1. Compare the detection effects of different methods for objects with different complexity.

In summary, our main contributions are as follows:

1. To improve the detection accuracy for objects with complex shapes, an EPEM is embedded into the edge branch. An FFM is designed to combine the feature channels from the two branches.

2. To improve the detection efficiency, a CAM is added into both branches, which suppresses redundant channels. And the rough version without the FFM can accelerate the inference speed.

3. We compare the proposed method with 8 state-of-the-art SOD methods on 9 datasets. Our method obtains better results than other methods, especially for objects with complex shapes. It achieves better performance under various evaluation metrics while realizing a forward reasoning speed of 73 FPS on a GPU.

The remainder of this paper is organized as follows: Section II reviews related works. In Section III, we introduce our method in detail, including the network structure and corresponding learning algorithm. The experimental results are presented in Section IV, and the conclusions are discussed in Section V.

II. RELATED WORKS

In recent years, CNN have been successfully applied for saliency detection and have achieved substantial improvements over other methods due to their powerful feature representation abilities. In particular, various SOD methods [30]–[32] that are based on fully convolutional neural networks (FCN) have achieved accurate results. Guan *et al.* [30] added edge features as complementary information for SOD. Huang *et al.* [31] leveraged a multiscale iteration of a CNN with two complementary subnets on different spatial scales, which combined the predictions of the two subnets to generate a more accurate boundary. Han *et al.* [32] used edge information as a constraint and predicted the saliency map pixel by pixel. Tu *et al.* [33] extracted hierarchical global and local information in FCN to incorporate nonlocal features for effective feature representations. Wang *et al.* [34] developed a new saliency detection method based on recurrent fully convolutional networks. Ren *et al.* [35] proposed a densely connected refinement network that fully utilizes deep features that are derived from multiple convolutional layers. Zhang *et al.* [36] proposed a hierarchical image co-saliency detection framework as a coarse-to-fine strategy for capturing this pattern. Jin *et al.* [37] proposed a novel complementary depth network that well exploits salient information depth features for RGB-D SOD. Although these methods have good performance in detecting objects with simple shapes, they have difficulty detecting objects with complex shapes.

To address this problem, many researchers designed networks for extracting powerful features for describing complex shapes objects. Qi *et al.* [38] designed a multiscale capsule attention module that captures functionality at multiple scales and generated an accurate saliency map by high-level semantic information and low-level spatial features. Guo *et al.* [39] proposed the AugFPN module, which narrows the semantic gaps between features of different scales before feature fusion through consistent supervision. Zhuge *et al.* [40] designed a novel integrity cognition network (ICON), which explores three important components to learn strong integrity features. Wei *et al.* [41] utilized an encoding-decoding network to detect the object body and edge separately and merged the two detection results into a saliency map. Li *et al.* [42] adopted a complementary perception network that uses a positive attention module to detect the foreground and a negative attention module to detect the background. Wei *et al.* [43] proposed F3Net for extracting powerful features, which consists a cross feature

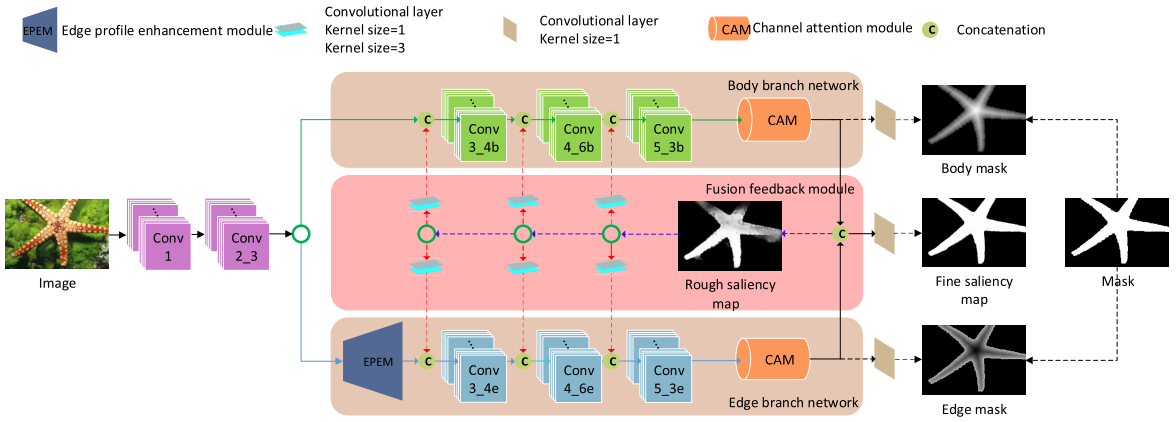


FIGURE 2. Overview of the proposed double branch networks.

module and a cascaded feedback decoder, and is trained by minimizing a new pixel position-aware loss. Liu *et al.* [44] designed a feature aggregation module that fuses the coarse-level semantic information with the fine-level features. Liu *et al.* [45] introduced a dual-branch method for SOD in which one branch provides a coarse-level saliency map that is optimized by the second branch. Wu *et al.* [47] proposed training saliency detection networks by exploiting supervision from not only salient object detection but also foreground contour detection and edge detection. Zhang *et al.* [48] proposed a novel attention-guided network that selectively integrates multilevel contextual information to detect objects. Although these methods perform relatively well in detecting objects with complex shapes, they reduce the computational efficiency.

To achieve high computational efficiency, researchers have focused on simplify the model. In [26], the authors proposed a reverse attention-based residual network, which simplifies the model size to 63 Mb and increases the detection speed to 35 FPS. An attentive feedback network of size 127 Mb and detection speed 26 FPS was designed by [27]. Zhang *et al.* [29] proposed a fast salient object detection algorithm with multiscale feature aggregation. Wu *et al.* [49] proposed a novel cascaded partial decoder framework for fast and accurate salient object detection. Li *et al.* [50] designed a novel depth nonlocal module that can model the contrast by collecting the correlations between channels using self-attention and realize fast and accurate detection in a single CPU thread. In practice, SOD methods should focus on both computing efficiency and object boundary accuracy.

III. PROPOSED METHOD

In this paper, the network is built on ResNet-50 [31]. As illustrated in Fig. 2, our network is divided into a body branch (which includes Conv3_4b, Conv4_6b, and Conv5_3b) and an edge branch (which includes Conv3_4e, Conv4_6e, and Conv5_3e), which share Conv1 and Conv2_3. To segment salient objects accurately, an EPEM is embedded into the input of the edge branch. To accelerate the inference speed, both the body branch and the edge branch are postprocessed with a CAM. The outputs of the two branches are combined by an FFM. The FFM is an important module that influences the detection accuracy and efficiency. Therefore, our method can be classified into two versions: The fine version, which uses the full FFM, can extract more accurate boundaries. The rough version, which simplifies the FFM, can obtain higher efficiency.

A. MASK COUPLING

To train the body branch and the edge branch individually, the original mask is converted into the original body mask and the original edge mask. These two masks can be generated with the same strategy as in [41]. The mask is a binary image M, and each pixel $g \in M$ is its corresponding value. We define the foreground as M_{f_g}(if $g \in M_{f_g}$, g equals 1) and the background as M_{b_g}($g \in M_{b_g}$, g equals 0). We use Euclidean distance to measure the distance between pixels, and we utilize Euclidean distance as the metric function:

$$f(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}, \tag{1}$$

where $p, q \in M$ and x, y are the pixel abscissa and ordinate, respectively. If pixel p belongs to the foreground and pixel q is the background pixel that is closest to p, then we use $f(p,q)$to calculate the distance between pixels p and q. If pixel p belongs to the background, their minimum distance is set to 0. After distance transformation, the original body mask, M, can be expressed as:

$$\tilde{M}(p) = \begin{cases} \min_{q \in M_{bg}} f(p,q), & p \in M_{fg}, \\ 0, & p \in M_{bg}, \end{cases} \tag{2}$$

We obtain the original edge mask by removing the original body mask $\tilde{M}$ from the original mask, M. Finally, we multiply

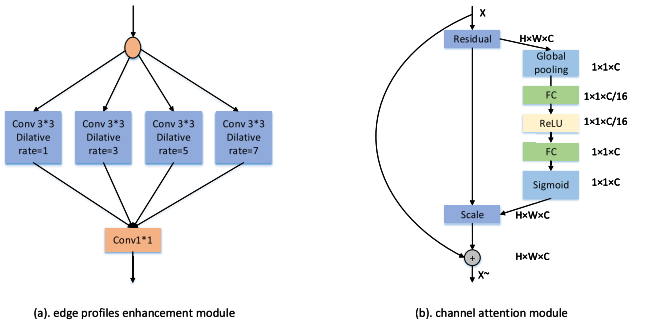
the newly generated body mask with the original mask, M, to remove the background interference:

$$Mask \Rightarrow \begin{cases} B = M * \tilde{M}, \\ E = M - B, \end{cases} \tag{3}$$

where B denotes the body mask, and E denotes the edge mask. The body mask is used to train the body branch; the edge mask is used to train the edge branch.

B. EDGE PROFILES ENHANCEMENT MODULE

According to [18], the edge profile feature can be extracted in the early stages of ResNet-50 (as block conv2_3 in our network). However, these edge feature maps are single scaled and cannot describe the salient object properly. The EPEM is embedded between conv2_3 and conv3_4e to extract multiscale edge features. As illustrated in Fig. 3(a), four dilated convolutional layers with different dilation rates are integrated into one module. The dilation rates are set to 1, 3, 5 and 7. Then, the outputs of the four dilated convolution layers are combined by 1×1 convolution. With different rate settings, the EPEM can extract edge profile features in parallel and concatenate the results together. Finally, 64 feature maps can be generated through a 1×1 convolution. The dilated convolutional layer provides a larger receptive field without changing the size of the CNN kernel. Therefore, this module improves the edge accuracy without significantly increasing the computational complexity.



(a). edge profiles enhancement module (b). channel attention module

FIGURE 3. (a) shows the edge profiles enhancement module, and (b) shows the channel attention module.

C. CHANNEL ATTENTION MODULE

The number of output channels in each branch is 2048, which is redundant for SOD. The invalid information not only causes inaccurate results but also leads to additional expensive computations. Therefore, inspired by [46], we introduce a channel attention module to suppress unimportant feature channels, as illustrated in Fig. 3(b). The CAM is embedded between the output of each branch and the FFM. First, the feature maps are generated via global average pooling to generate channel-wise statistics. Second, to suppress redundant information, the gating mechanism forms a bottleneck with two fully connected layers, a ReLU and a sigmoid, thereby returning to the channel dimension of the transformation output. Finally, the output of the block is obtained by rescaling. Because the weights of unimportant feature channels are approximately 0, the amount of redundant information will be greatly reduced.

D. FUSION FEEDBACK MODULE

To fuse the information of the edge and the body, a feature interaction network is proposed in [41]. In this method, the feature maps of the two branches are fed back to the two branches through an interaction encoder. In the proposed method, we instead feed back the rough saliency map, which not only contains more detailed information but also can be fed back more quickly. This module is divided into three stages: fusion, feedback and refusion. First, the outputs of the two branches are concatenated to predict a rough saliency map through a 1×1 convolution layer. Second, this rough saliency map is fed back to each block using bilinear upsampling. Then, through a 3×3 convolution layer and a 1×1 convolution layer, feedback feature maps are generated with the same dimension of each block. Finally, all the feature maps are fused by concatenating the outputs of the two branches again. A fine saliency map is ultimately predicted by these three stages. To balance efficiency and accuracy, our method modifies the FFM in different ways. To achieve accurate edges, the fine saliency map is accepted as the final detection result. To achieve fast test speeds, the rough saliency map is accepted. In the latter situation, only the first stage of the FFM is retained. In Sec. IV.C, the rough version can achieve a speed of 73 FPS, which is higher than that of the fine version.

E. LOSS FUNCTION

As our method can generate a rough saliency map and a fine saliency map, there are two kinds of loss functions. For the rough prediction map, the loss can be calculated as:

$$L_R = \ell_e^r + \ell_b^r + l_{iou}^r, \tag{4}$$

where ℓ_e^r, ℓ_b^r and l_{iou}^r denote the edge loss, body loss and structural loss for rough saliency maps. We utilize binary cross entropy (BCE) to calculate ℓ_e^r and ℓ_b^r, which is defined as:

$$\ell_i = -\sum_{(x,y)} [g_i(x,y) \log (p_i(x,y)) + (1 - g_i(x,y)) \log (1 - p_i(x,y))] \quad i = b, e, \tag{5}$$

where $g_b(x,y) \in B$ and $g_e(x,y) \in E$, which are defined in Sec.III.A, and $p_b(x,y) \in [0,1]$ and $p_e(x,y) \in [0,1]$ are the predicted probabilities of the body and the edge, respectively, of the rough salient object. However, BCE calculates the loss for each pixel independently and ignores the global structure of the image. To remedy this problem, we utilize the IoU loss to calculate l_{iou}, which can measure the similarity of two images overall rather than with respect to a single pixel. It can

be expressed as follows:

$$l_{iou} = 1 - \frac{\sum_{(x,y)}[p(x,y) * g(x,y)]}{\sum_{(x,y)}[g(x,y) + p(x,y) - p(x,y) * g(x,y)]}, \quad (6)$$

where $g(x, y) \in M$ and $p(x, y)$ is the predicted probability of the saliency map. For the fine prediction map, the loss function is defined as:

$$L_F = \ell_e^f + \ell_b^f + l_{iou}^f. \quad (7)$$

where ℓ_e^f, ℓ_b^f and l_{iou}^f denote the edge loss, body loss and structural loss for fine saliency maps. The calculation method is the same as in Eq. (5) and Eq. (6), but the values (p_e, p_e and p) are optimized by adding the FFM.

F. TRAINING PROCESS

The training process of our method involves two versions, as presented in Algorithm 1. In the rough version, the body branch network, edge branch network, Conv2_ 3 and Conv1 are initialized. The image, mask, edge mask and body mask are inputted for training progress. In every epoch, the loss is calculated by Eq. (4). And the rough version weights (including the Body branch network, Edge branch network, Conv2_3 and Conv1) are updated. The rough saliency maps are then generated. In the fine version, the FFM are initialized and the rough version weights are loaded. In every epoch, the loss is calculated by Eq. (7). And the fine version weights (including the FFM, Body branch network, Edge branch network, Conv2_3 and Conv1) are updated. The fine saliency maps are finally generated.

Algorithm 1 Training

Rough version
Input: image, mask, edge mask, body mask
initialize (Body branch network, Edge branch network, Conv2_3 and Conv1);
for epoch $i \in [1, 50]$ **do**
  $L_R = \ell_e^r + \ell_b^r + l_{iou}^r$,
  update (Body branch network, Edge branch network, Conv2_3 and Conv1)
end for
Output: Rough saliency map
Fine version
Input: image, mask, edge mask, body mask
initialize FFM and load Rough version weights
for epoch $i \in [1, 50]$ **do**
  $L_F = \ell_e^f + \ell_b^f + l_{iou}^f$,
  update(FFM, Body branch network, Edge branch network, Conv2_3 and Conv1)
end for
Output: Fine saliency map

IV. EXPERIMENT

A. IMPLEMENTATION DETAILS

We select ResNet-50 as the backbone network and use a GTX 1080Ti GPU for training and testing. Our model is implemented in PyTorch. The hyperparameters are set as follows: batch size = (48), epoch = (50), momentum = (0.9), weight decay = (0.0005), and maximum learning rate = (0.08). The warm-up and linear decay strategies are used for the learning rate. During training and testing, each image is resized to 352 × 352.

TABLE 1. Benchmark datasets that are used in this study.

Dataset	Training set	Test set
RGBD135	-	135
NLPR	700	300
ECSSD	-	1000
STEREO	2985	797
HKU-IS	8894	4445
DUT-RGBD	800	400
DUTS-TE	10553	5019
DUT-COMPLEX	-	293
DUT-OMRON	-	5168

B. COMPARISON WITH STATE-OF-THE-ART METHODS

We adopt 5 widely applied and standard metrics to evaluate our model: the precision-recall (PR) curve, mean F-measure (mF), E-measure (E_m), S-measure (S_m) and mean absolute error (MAE). In addition, we carry out experiments on 9 datasets: RGBD135, NLPR, ECSSD, STEREO, HKU-IS, DUT-RGBD, DUTS-TE, DUT-COMPLEX and DUT-OMRON. These datasets are described in Tab. 1. To ensure a fair comparison, only the original images and masks are utilized. To evaluate the performance of our method in the detection of objects with complex shapes, we select the images of objects with complex shapes automatically. We define the salient object complexity as follows:

$$C = \frac{P \times P}{A}, \quad (8)$$

where P is the number of edge pixels of the ground truth, and A is the number of pixels in the object.

In Eq. (8), the Complexity C is a ratio between the square of the perimeter and the area. In the calculation, the number of edge pixels is represented for the perimeter of the object, and the number of pixels is represented for the area. It means that the complexity coefficient is independent to the number of pixels. We define that when the C > 100, the object is a complex shape object. From this definition, DUT-COMPLEX is established, containing 293 complex shape object images.

Most metrics cannot be used to evaluate the edge accuracy. Therefore, we use MAE_e to evaluate the accuracy of the salient object boundary. MAE_e is introduced in [25] to measure the average difference between the predicted edge and the ground-truth edge. The metric is defined as follows:

$$MAE_e = \frac{1}{H \times W} \sum_{x=1}^{H} \sum_{y=1}^{M} \left| P_{edge}(x,y) - G_{edge}(x,y) \right|, \quad (9)$$

TABLE 2. Performance evaluation metrics: mF, E_m, S_{m_e}, S_m, MAE_e and MAE. In the table, '-' indicates results that are not publicly available, ↑ and ↓ denote that larger and smaller values are better. Values in red indicate the best result, and values in blue indicate the second best. Size indicates the model size, and speed indicates the test efficiency.

dateset	metrics	Fine	Rough	EGNet [18]	R2Net [21]	DFI [22]	RAS [26]	LDF [41]	F3Net [43]	PoolNet [44]	ICON [40]
	size ↓	100	98	412	172	116	63	112	98	260	–
	speed↑	55	73	6	33	21	35	40	18	20	–
RGBD135	mF ↑	0.928	0.920	0.899	0.911	0.911	0.900	0.907	0.913	0.903	–
	E_m ↑	0.954	0.945	0.930	0.929	0.934	0.901	0.912	0.933	0.940	–
	S_{m_e} ↑	0.833	0.818	0.744	0.765	0.789	0.772	0.790	0.800	0.771	–
	S_m ↑	0.919	0.910	0.879	0.899	0.901	0.872	0.873	0.894	0.888	–
	MAE_e ↓	0.052	0.053	0.057	0.056	0.054	0.059	0.059	0.054	0.054	–
	MAE ↓	0.024	0.027	0.035	0.032	0.030	0.041	0.040	0.028	0.031	–
NLPR	mF ↑	0.915	0.911	0.864	0.887	0.874	0.871	0.844	0.866	0.841	–
	E_m ↑	0.947	0.943	0.896	0.908	0.913	0.903	0.873	0.896	0.899	–
	S_{m_e} ↑	0.839	0.835	0.762	0.769	0.786	0.788	0.785	0.796	0.763	–
	S_m ↑	0.921	0.916	0.881	0.893	0.880	0.875	0.853	0.873	0.867	–
	MAE_e ↓	0.051	0.053	0.060	0.057	0.056	0.058	0.067	0.057	0.060	–
	MAE ↓	0.026	0.028	0.046	0.040	0.041	0.043	0.059	0.044	0.046	–
ECSSD	mF ↑	0.948	0.947	0.943	0.942	0.947	0.931	0.942	0.925	0.945	–
	E_m ↑	0.926	0.925	0.926	0.922	0.925	0.919	0.920	0.924	0.923	–
	S_{m_e} ↑	0.836	0.833	0.787	0.774	0.812	0.792	0.819	0.833	0.799	–
	S_m ↑	0.927	0.924	0.926	0.920	0.927	0.907	0.924	0.924	0.922	–
	MAE_e ↓	0.085	0.088	0.093	0.095	0.089	0.094	0.090	0.088	0.091	–
	MAE ↓	0.033	0.034	0.040	0.043	0.039	0.044	0.037	0.033	0.038	–
STEREO	mF ↑	0.916	0.916	0.907	0.916	0.904	0.905	0.897	0.907	0.903	0.905
	E_m ↑	0.912	0.916	0.903	0.915	0.911	0.913	0.893	0.910	0.913	–
	S_{m_e} ↑	0.815	0.815	0.777	0.759	0.779	0.782	0.792	0.797	0.770	–
	S_m ↑	0.910	0.909	0.898	0.896	0.896	0.893	0.886	0.896	0.895	–
	MAE_e ↓	0.091	0.091	0.096	0.095	0.094	0.094	0.096	0.092	0.094	–
	MAE ↓	0.039	0.039	0.051	0.049	0.046	0.048	0.051	0.043	0.046	–
HKU-IS	mF ↑	0.968	0.946	0.946	0.946	0.934	0.941	0.951	0.951	0.944	0.905
	E_m ↑	0.976	0.957	0.960	0.947	0.958	0.959	0.963	0.962	0.955	0.953
	S_{m_e} ↑	0.852	0.823	0.805	0.780	0.814	0.811	0.833	0.835	0.800	–
	S_m ↑	0.954	0.922	0.924	0.903	0.927	0.919	0.930	0.929	0.923	0.921
	MAE_e ↓	0.076	0.081	0.085	0.087	0.082	0.083	0.079	0.078	0.084	–
	MAE ↓	0.020	0.029	0.032	0.038	0.030	0.030	0.025	0.025	0.030	0.027
DUT-RGBD	mF ↑	0.941	0.934	0.902	0.910	0.913	0.887	0.884	0.905	0.907	–
	E_m ↑	0.948	0.945	0.916	0.922	0.926	0.903	0.874	0.912	0.924	–
	S_{m_e} ↑	0.832	0.827	0.750	0.753	0.787	0.764	0.776	0.789	0.776	–
	S_m ↑	0.929	0.920	0.891	0.894	0.899	0.872	0.859	0.888	0.892	–
	MAE_e ↓	0.077	0.078	0.088	0.088	0.084	0.091	0.097	0.087	0.085	–
	MAE ↓	0.030	0.033	0.056	0.054	0.048	0.075	0.054	0.065	0.045	–
DUTS-TE	mF ↑	0.878	0.878	0.877	0.862	0.876	0.874	0.873	0.840	0.839	0.840
	E_m ↑	0.887	0.888	0.876	0.877	0.881	0.887	0.880	0.870	0.810	0.893
	S_{m_e} ↑	0.787	0.782	0.747	0.729	0.759	0.755	0.785	0.802	0.763	–
	S_m ↑	0.881	0.879	0.881	0.864	0.876	0.874	0.880	0.888	0.879	0.893
	MAE_e ↓	0.065	0.067	0.068	0.069	0.068	0.069	0.067	0.063	0.066	–
	MAE ↓	0.041	0.040	0.044	0.047	0.043	0.042	0.041	0.035	0.036	0.039
DUT-COMPLEX	mF ↑	0.829	0.802	0.799	0.779	0.788	0.782	0.753	0.798	0.801	–
	E_m ↑	0.870	0.852	0.802	0.833	0.824	0.813	0.832	0.843	0.845	–
	S_{m_e} ↑	0.740	0.736	0.731	0.723	0.719	0.703	0.725	0.733	0.729	–
	S_m ↑	0.839	0.834	0.799	0.813	0.809	0.813	0.816	0.822	0.819	–
	MAE_e ↓	0.089	0.091	0.104	0.105	0.103	0.110	0.105	0.103	0.097	–
	MAE ↓	0.071	0.076	0.090	0.088	0.089	0.088	0.093	0.083	0.082	–
DUT-OMRON	mF ↑	0.817	0.812	0.826	0.793	0.809	0.814	0.773	0.766	0.830	0.762
	E_m ↑	0.860	0.856	0.823	0.852	0.849	0.836	0.853	0.870	0.859	0.866
	S_{m_e} ↑	0.774	0.760	0.709	0.744	0.739	0.717	0.716	0.740	0.753	–
	S_m ↑	0.839	0.834	0.813	0.824	0.819	0.825	0.826	0.838	0.835	0.845
	MAE_e ↓	0.072	0.074	0.074	0.075	0.073	0.074	0.073	0.073	0.075	–
	MAE ↓	0.058	0.059	0.060	0.061	0.059	0.058	0.061	0.053	0.055	0.058

where H and W are the height and width of the picture and $P_{edge}(x, y)$ and $G_{edge}(x, y)$ are the predicted and mask values, respectively, for the salient object edge profile. In addition, inspired by the definition of S_m, we propose S_{m_e} for computing the boundary-aware structural similarities between the prediction edge E and the ground-truth edge G. The metric is defined as follows:

$$S_{m_e} = \alpha \times S_o(E, G) + (1 - \alpha) \times S_r(E, G). \quad (10)$$

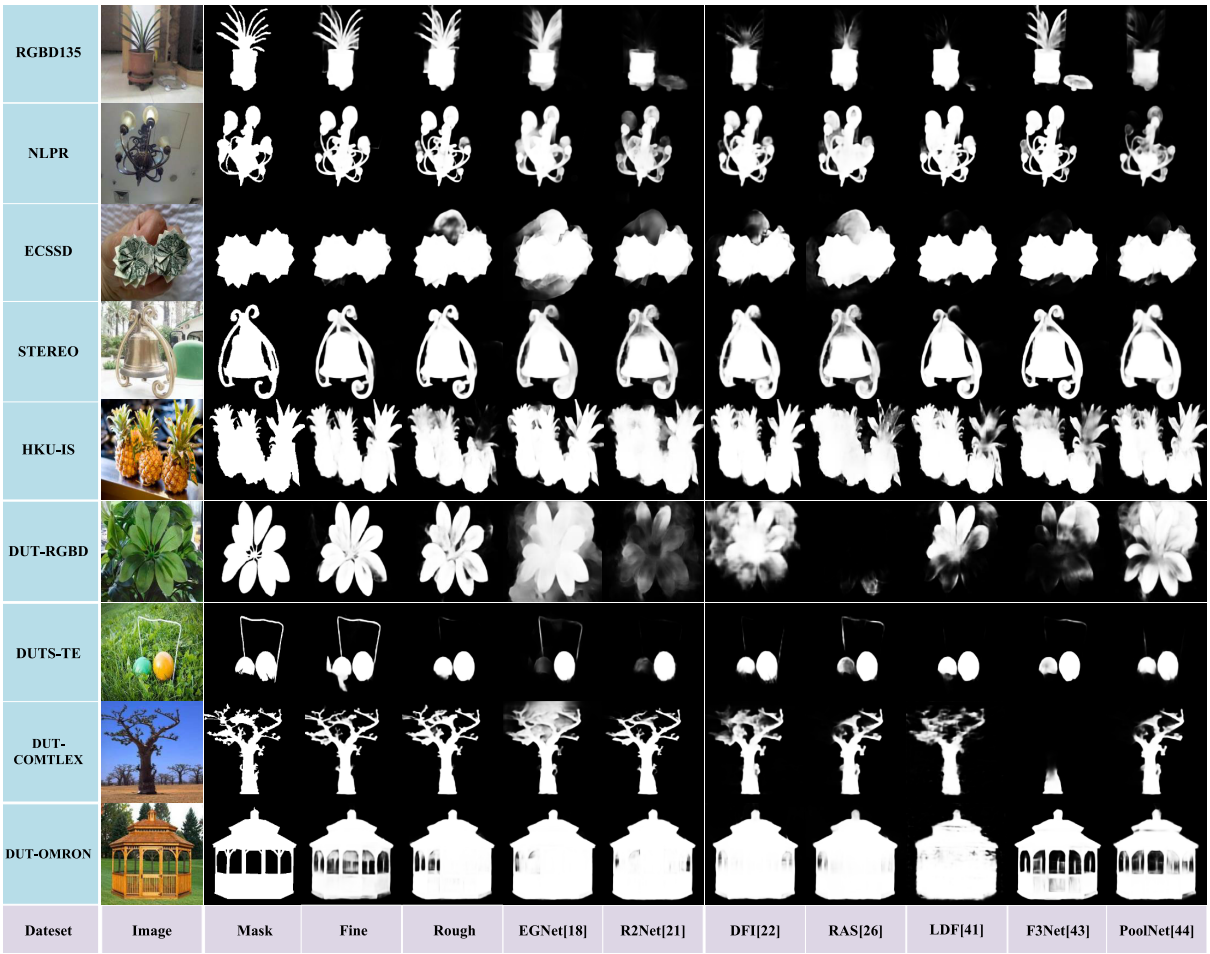


FIGURE 4. Visual comparison of the results obtained using each algorithm.

where S_o and S_r denote the object-edge-aware and region-edge-aware structural similarities and α is set to 0.5.

1) VISUAL COMPARISON

In Fig. 4, we compare our method with 7 state-of-the-art methods: EGNet [18], R2Net [21], DFI [22], RAS [26], LDF [41], F3Net [43] and PoolNet [44]. Each row presents the detection result on a dataset. The original image is displayed in the first column, the mask is shown in the second column, and each subsequent column represents a detection method. The visual comparison results show that the proposed method is more accurate than the compared methods, especially for the examples with complex shapes. From the example in RGBD135, our method can detect each leaf accurately. From the example in DUT-RGBD, it is found that when the foreground and background are similar, most methods cannot clearly distinguish the foreground and background. From the example in DUTS-TE, our method can detect the edge profiles of the microscopic object. From the example in DUT-COMPLEX, we find that when the contour of the object contains minor details, most methods can only detect the fuzzy contour but cannot accurately detect the minor details. By comparing the detection results of the above methods on various datasets, we find that the detection accuracies of the double-branch networks (such as ours, DFI, LDF and F3Net) are better than those of single-branch networks (such as EGNet, R2Net and RAS). As the EPEM can extract multi-scale features, our results are more accurate than LDF and DFI. Compared with the rough version, the fine version shows more detailed information for adding the FFM. We believe our method is superior because of the following two aspects: (1) In the rough version, although we reduce the number of model parameters as much as possible, we obtain better results because we use the double-branch network mode and the edge enhancement module to focus on edge information detection. (2) In the fine version, we give fine feedback based on the rough version, hence, we can obtain better results.

2) QUANTITATIVE ANALYSIS

In Fig. 5, we show the standard PR curves for 9 datasets. The red line in each figure represents the PR curve of fine version, and the blue line represents the PR curve of rough version. The proposed method yields better results than EGNet [18],

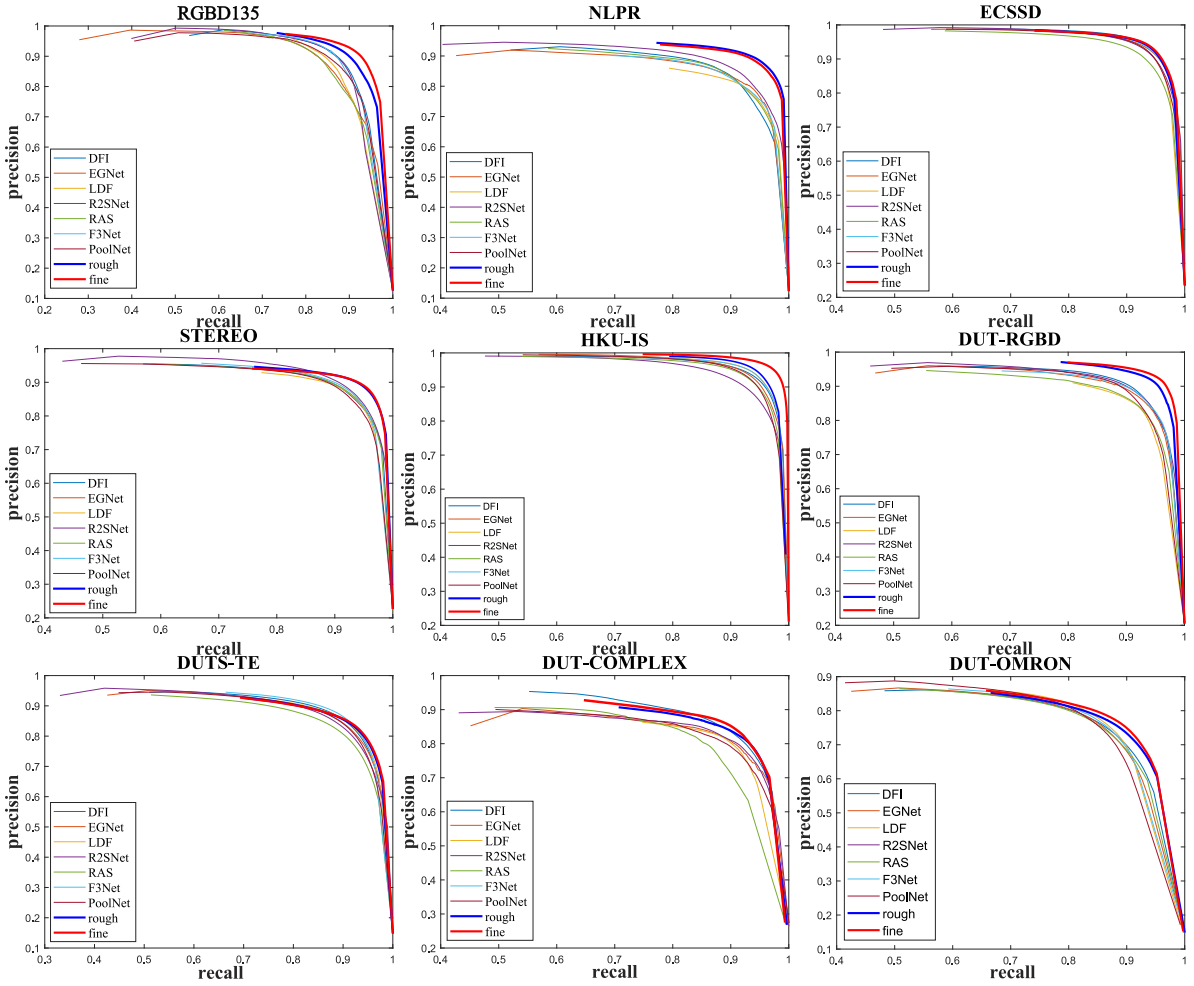


FIGURE 5. Precision-recall curves on 9 public benchmark datasets. It can be seen that the proposed method performs favorably against state-of-the-art methods.

R2Net [21], DFI [22], RAS [26], LDF [41], F3Net [43] and PoolNet [44] according to their PR curves. Especially on the NLPR, STEREO, HKU-IS, DUT-RGBD and DUT-COMPLEX, when the recall score is close to 1, our accuracy is much higher than those of other methods, which shows that the false-alarm rate of our saliency map is much lower.

In Tab. 2, we compare our method with 8 methods. In order to ensure the fairness of the process evaluation, we compared references [18], [21], [22], [26], [41], [43], [44] and our method on the same experimental platform. As reference [40] doesn't provide the code, the experiment results are directly from the original paper. It can be seen that our model performs favorably against the state-of-the-art methods under all evaluation metrics on the datasets RGBD135, NLPR, ECSSD, STEREO, HKU-IS, DUT-RGBD and DUT-COMPLEX. Although our method does not perform the best on DUTS-TE and DUT-OMRON, it is still closed to the best performance. In addition, the fine version achieves the best results on MAE_e and S_{m_e} on all datasets except the DUTS-OMRON, which shows that our method realizes a significant improvement in detecting object edge. Although the accuracy of the rough version is not the best, it achieves the best detection speed of 73 FPS. And the size of the rough version model is only 98 Mb, which is smaller than all compared methods except the RAS [26] model.

C. ABLATION ANALYSIS

In this section, we explore the effectiveness of various components of the proposed network using the DUT-OMRON and the DUT-COMPLEX. Through an ablation analysis on these two datasets, we compare the speed and accuracy performances of the EPEM, CAM and FFM. In Fig. 6, we present visualization results with and without these modules. Through comparison, we evaluate the improvement that is realized by each module in detecting complex shapes.

1) THE EFFECTIVENESS OF DIFFERENT BACKBONES

As shown in Tab. 3, we compared the results with ResNet-18, ResNet-50 and ResNet-101 as the backbone. The ResNet-18 gets the best efficiency and worst accuracy. On the contrary, the ResNet-101 gets the worst efficiency and best accuracy.

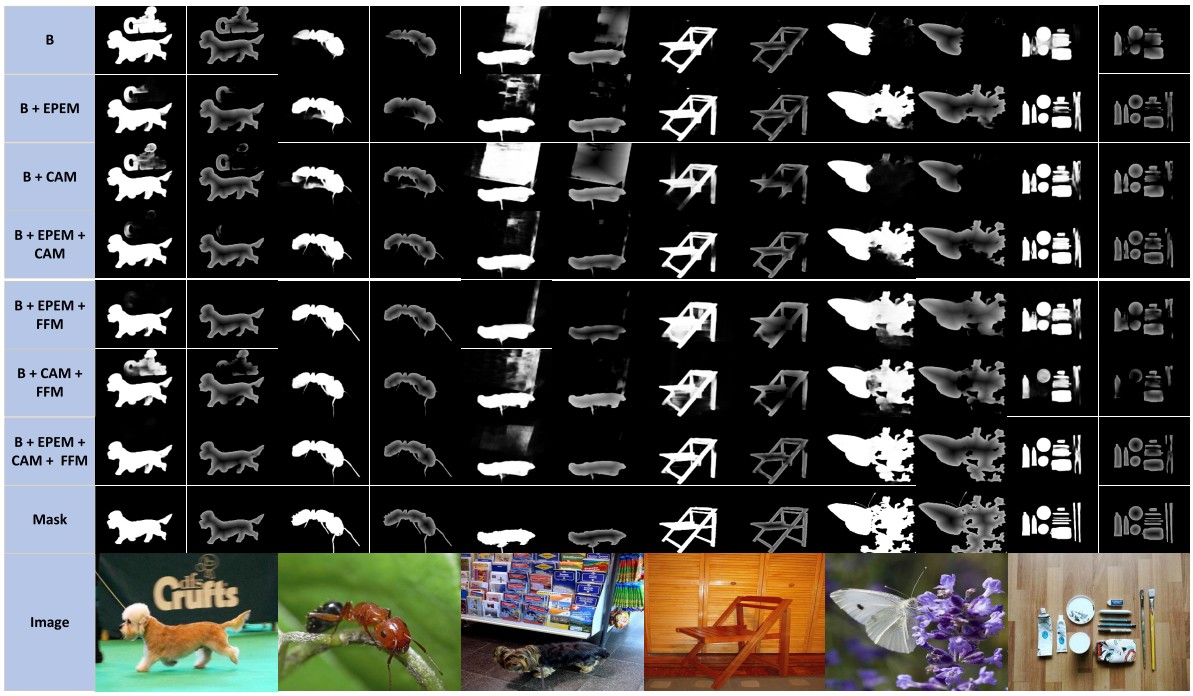


FIGURE 6. Visual examples with or without various modules.

TABLE 3. The effectiveness of different backbone.

Backbone	FPS	size	DUT-OMRON		DUT-COMPLEX	
			$mF\uparrow$	$E_m\uparrow$	$mF\uparrow$	$E_m\uparrow$
ResNet-18	76	47	0.794	0.832	0.782	0.813
ResNet-50	73	98	0.812	0.856	0.802	0.852
ResNet-101	33	172	0.817	0.862	0.812	0.859

TABLE 4. Effects of various convolutional layer kernel sizes of the EPEM.

EPEM	DUT-OMRON		DUT-COMPLEX	
	$mF\uparrow$	$E_m\uparrow$	$mF\uparrow$	$E_m\uparrow$
different dilation rates	0.817	0.860	0.829	0.870
different kernel sizes	0.788	0.824	0.812	0.852

Therefore, considering the advantage and disadvantage of the three backbones, we choose the ResNet-50.

2) THE EFFECTIVENESS OF THE EDGE PROFILE ENHANCEMENT MODULE

To evaluate the effectiveness of the EPEM, we compare the results with and without EPEM. As shown in Fig. 6, before adding the EPEM, most of our detection results have some false detections or detection loss. After adding the EPEM, our detection results are greatly improved. We select four metrics, namely, MAE_e, MAE, S_{m_e} and S_m, to compare the accuracy of the test results. As shown in Tab. 5 and Fig. 6, the proposed method achieves higher accuracy with the EPEM. On the DUT-COMPLEX, the performance on all four metrics is clearly improved with the EPEM. These results prove that the salient edge information is very useful for the SOD task and that multiscale dilated convolutions can improve the accuracy of object detection. In addition, we compared the experimental results of the EPEM in two ways: different dilation rates (the same kernel size 3×3 with different dilation rates 1, 3, 5 and 7) and different kernel sizes(the different kernel sizes 1×1, 3×3, 5×5 and 7×7 without dilation rates). As shown in Tab. 4, the EPEM can get better results in the same kernel size with different dilation rates.

3) THE EFFECTIVENESS OF THE CHANNEL ATTENTION MODULE

To evaluate the effectiveness of the CAM, we compare our method with and without the CAM. By comparing the test results on the DUT-OMRON and DUT-COMPLEX, as shown in Tab. 5 and Fig. 6, we find that the CAM improves the efficiency with little reduction in accuracy. As the CAM reduces the amount of redundant information, it improves the detection speed.

4) THE EFFECTIVENESS OF THE FUSION FEEDBACK MODULE

The FFM is used to combine the edge branch and the body branch. To evaluate the effectiveness of the FFM, we compared the results that are obtained with and without the FFM. As shown in Fig. 6, after adding the FFM, our results are further improved compared with adding a module alone and are more complete in terms of edge detail detection. As shown in Tab. 5, the FFM results in a slower detection speed, but the speed is still higher than those of other methods.

TABLE 5. Ablation analyses on DUT-COMPLEX and DUT-OMRON. B denotes the backbone, EPEM denotes the edge contour enhancement module, CAM denotes the channel attention module, FFM denotes the fusion feedback module, and red indicates the best result.

Model	FPS	DUT-OMRON				DUT-COMPLEX			
		$S_{m_e}\uparrow$	$S_m\uparrow$	$MAE_e\downarrow$	$MAE\downarrow$	$S_{m_e}\uparrow$	$S_m\uparrow$	$MAE_e\downarrow$	$MAE\downarrow$
B	67	0.686	0.792	0.087	0.079	0.620	0.730	0.110	0.108
B + EPEM	66	0.724	0.820	0.085	0.074	0.720	0.790	0.099	0.090
B + CAM	73	0.697	0.805	0.080	0.074	0.633	0.755	0.107	0.103
B + EPEM + FFM	53	0.764	0.836	0.074	0.059	0.738	0.810	0.090	0.072
B + CAM +FFM	55	0.754	0.832	0.073	0.059	0.710	0.808	0.092	0.077
B + EPEM + CAM(Rough)	73	0.760	0.834	0.074	0.059	0.736	0.824	0.091	0.076
B + EPEM + CAM + FFM(Fine)	55	0.774	0.839	0.072	0.058	0.740	0.839	0.089	0.071

The FFM feeds back the rough saliency maps to extract more accurate results. However, during the test, the FFM increases the number of calculations that are required.

V. CONCLUSION

In this paper, we propose a double-branch network for SOD. To improve the detection accuracy and efficiency, we propose the inclusion of EPEM, CAM and FFM components. The EPEM provides multiscale edge features for edge branch to improve the accuracy. Experimental results show that the EPEM yields more accurate results for complex objects than single-scale methods. Extracting features from two branches may cause information redundancy and require expensive computations. The CAM can suppress redundant information and raise the detection efficiency. The FFM can combine the information of the two branches and improve the accuracy through feedback. In this paper, we propose a fine version and a rough version. The fine version can greatly improve the detection accuracy. The rough version can greatly improve the detection speed with little reduction in accuracy. In addition, our method obtains better results in detecting objects with complex shapes. Using 9 datasets to evaluate the performance of the proposed method, we find that our method outperforms 8 state-of-the-art methods under a variety of evaluation metrics and has a real-time speed of 73 FPS.

REFERENCES

[1] S. Li, J. Wang, and Q. Zhu, "High dynamic range image compression based on visual saliency," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 21–25, doi: 10.1109/PCS.2018.8456247.

[2] S. Srivastava, P. Mukherjee, and B. Lall, "Adaptive image compression using saliency and KAZE features," in *Proc. Int. Conf. Signal Process. Commun. (SPCOM)*, Jun. 2016, pp. 1–5, doi: 10.1109/SPCOM.2016.7746680.

[3] L. Yang, D. Xin, L. Zhai, F. Yuan, and X. Li, "Active contours driven by visual saliency fitting energy for image segmentation in SAR images," in *Proc. IEEE 4th Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2019, pp. 393–397, doi: 10.1109/ICCCBDA.2019.8725646.

[4] Q. Li, Y. Zhou, and J. Yang, "Saliency based image segmentation," in *Proc. Int. Conf. Multimedia Technol.*, Jul. 2011, pp. 5068–5071, doi: 10.1109/ICMT.2011.6002178.

[5] H. Liu, W. Chu, and H. Wang, "Automatic segmentation algorithm of ultrasound heart image based on convolutional neural network and image saliency," *IEEE Access*, vol. 8, pp. 104445–104457, 2020, doi: 10.1109/ACCESS.2020.2989819.

[6] X. Tengjiao, Z. Danpei, S. Jun, and L. Ming, "High-speed recognition algorithm based on BRISK and saliency detection for aerial images," in *Proc. 3rd Int. Conf. Intell. Syst. Design Eng. Appl.*, Jan. 2013, pp. 816–819, doi: 10.1109/ISDEA.2012.194.

[7] Z. Ren, S. Gao, L.-T. Chia, and I. W.-H. Tsang, "Region-based saliency detection and its application in object recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 5, pp. 769–779, May 2014, doi: 10.1109/TCSVT.2013.2280096.

[8] L. Zhi-Jie, "Image classification method based on visual saliency and bag of words model," in *Proc. 8th Int. Conf. Intell. Comput. Technol. Automat. (ICICTA)*, Jun. 2015, pp. 466–469, doi: 10.1109/ICICTA.2015.122.

[9] J. Tang, Y. Ge, and Y. Liu, "Application of visual saliency and feature extraction algorithm applied in large-scale image classification," in *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, Oct. 2016, pp. 1–6, doi: 10.1109/CESYS.2016.7889903.

[10] Q. Kai, "A painting image retrieval approach based on visual features and semantic classification," in *Proc. Int. Conf. Smart Grid Electr. Automat. (ICSGEA)*, Aug. 2019, pp. 195–198, doi: 10.1109/ICSGEA.2019.00052.

[11] S. Pu, Y. Song, C. Ma, H. Zhang, and M.-H. Yang, "Deep attentive tracking via reciprocative learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1935–1945.

[12] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015.

[13] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3395–3402.

[14] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2678–2687.

[15] X. Xu and W. Wu, "An adaptive computational method for color contrast based salient region detection," in *Proc. IEEE Int. Conf. Secur., Pattern Anal., Cybern. (SPAC)*, Oct. 2014, pp. 54–58, doi: 10.1109/SPAC.2014.6982656.

[16] S. Roy and S. Das, "Spatial variance of color and boundary statistics for salient object detection," in *Proc. 4th Nat. Conf. Comput. Vis., Pattern Recognit., Image Process. Graph. (NCVPRIPG)*, Dec. 2013, pp. 1–4, doi: 10.1109/NCVPRIPG.2013.6776270.

[17] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *Proc. CVPR*, Jun. 2011, pp. 409–416, doi: 10.1109/CVPR.2011.5995344.

[18] J. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "EGNet: Edge guidance network for salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8778–8787, doi: 10.1109/ICCV.2019.00887.

[19] S. Zhou, J. Wang, J. Zhang, L. Wang, D. Huang, S. Du, and N. Zheng, "Hierarchical U-shape attention network for salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 8417–8428, 2020, doi: 10.1109/TIP.2020.3011554.

[20] Y. Pang, X. Zhao, L. Zhang, and H. Lu, "Multi-scale interactive network for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9410–9419, doi: 10.1109/CVPR42600.2020.00943.

[21] M. Feng, H. Lu, and Y. Yu, "Residual learning for salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 4696–4708, 2020, doi: 10.1109/TIP.2020.2975919.

[22] J.-J. Liu, Q. Hou, and M.-M. Cheng, "Dynamic feature integration for simultaneous detection of salient object, edge, and skeleton," *IEEE Trans. Image Process.*, vol. 29, pp. 8652–8667, 2020, doi: 10.1109/TIP.2020.3017352.
[23] W. Wang, S. Zhao, J. Shen, S. C. H. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1448–1457, doi: 10.1109/CVPR.2019.00154.
[24] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 202–211.
[25] P. Mukherjee, B. Lall, and S. Tandon, "Salprop: Salient object proposals via aggregated edge cues," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2423–2429.
[26] S. Chen, X. Tan, B. Wang, H. Lu, X. Hu, and Y. Fu, "Reverse attention-based residual network for salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 3763–3776, 2020, doi: 10.1109/TIP.2020.2965989.
[27] M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1623–1632, doi: 10.1109/CVPR.2019.00172.
[28] H. Li, G. Li, B. Yang, G. Chen, L. Lin, and Y. Yu, "Depthwise nonlocal module for fast salient object detection using a single thread," *IEEE Trans. Cybern.*, vol. 51, no. 12, pp. 6188–6199, Dec. 2021, doi: 10.1109/TCYB.2020.2969282.
[29] X. Zhang and L. Zhu, "Fast salient object detection based on multi-scale feature aggression," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 5734–5738, doi: 10.1109/CCDC.2019.8832539.
[30] W. Guan, T. Wang, J. Qi, L. Zhang, and H. Lu, "Edge-aware convolution neural network based salient object detection," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 114–118, Jan. 2019, doi: 10.1109/LSP.2018.2881835.
[31] K. Huang and S. Gao, "Image saliency detection via multi-scale iterative CNN," *Vis. Comput.*, vol. 36, no. 7, pp. 1355–1367, Jul. 2020.
[32] L. Han, X. Li, and Y. Dong, "Convolutional edge constraint-based U-Net for salient object detection," *IEEE Access*, vol. 7, pp. 48890–48900, 2019, doi: 10.1109/ACCESS.2019.2910572.
[33] Z. Tu, Y. Ma, C. Li, C. Li, J. Tang, and B. Luo, "Edge-guided non-local fully convolutional network for salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 582–593, Feb. 2021, doi: 10.1109/TCSVT.2020.2980853.
[34] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Salient object detection with recurrent fully convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1734–1746, Jul. 2019, doi: 10.1109/TPAMI.2018.2846598.
[35] Q. Ren and R. Hu, "Densely connected refinement network for salient object detection," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS)*, Nov. 2018, pp. 1–6, doi: 10.1109/ISPACS.2018.8923354.
[36] K. Zhang, T. Li, B. Liu, and Q. Liu, "Co-saliency detection via mask-guided fully convolutional networks with multi-scale label smoothing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3090–3099, doi: 10.1109/CVPR.2019.00321.
[37] W.-D. Jin, J. Xu, Q. Han, Y. Zhang, and M.-M. Cheng, "CDNet: Complementary depth network for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 3376–3390, 2021, doi: 10.1109/TIP.2021.3060167.
[38] Q. Qi, S. Zhao, J. Shen, and K.-M. Lam, "Multi-scale capsule attention-based salient object detection with multi-crossed layer connections," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 1762–1767, doi: 10.1109/ICME.2019.00303.
[39] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "AugFPN: Improving multi-scale feature learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12592–12601, doi: 10.1109/CVPR42600.2020.01261.
[40] M. Zhuge, D.-P. Fan, N. Liu, D. Zhang, D. Xu, and L. Shao, "Salient object detection via integrity learning," 2021, *arXiv:2101.07663*.
[41] J. Wei, S. Wang, Z. Wu, C. Su, Q. Huang, and Q. Tian, "Label decoupling framework for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13022–13031, doi: 10.1109/CVPR42600.2020.01304.
[42] J. Li, Z. Pan, Q. Liu, Y. Cui, and Y. Sun, "Complementarity-aware attention network for salient object detection," *IEEE Trans. Cybern.*, early access, May 12, 2020, doi: 10.1109/TCYB.2020.2988093.
[43] J. Wei, S. Wang, and Q. Huang, "F3Net: Fusion, feedback and focus for salient object detection," 2019, *arXiv:1911.11445*.
[44] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3912–3921, doi: 10.1109/CVPR.2019.00404.
[45] Y. Liu, J. Han, Q. Zhang, and L. Wang, "Salient object detection via two-stage graphs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 4, pp. 1023–1037, Apr. 2019, doi: 10.1109/.TCSVT.2018.2823769.
[46] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020, doi: 10.1109/TPAMI.2019.2913372.
[47] R. Wu, M. Feng, W. Guan, D. Wang, H. Lu, and E. Ding, "A mutual learning method for salient object detection with intertwined multi-supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8142–8151, doi: 10.1109/CVPR.2019.00834.
[48] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 714–722, doi: 10.1109/CVPR.2018.00081.
[49] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3902–3911, doi: 10.1109/CVPR.2019.00403.
[50] H. Li, G. Li, B. Yang, G. Chen, L. Lin, and Y. Yu, "Depthwise nonlocal module for fast salient object detection using a single thread," *IEEE Trans. Cybern.*, vol. 51, no. 12, pp. 6188–6199, Dec. 2021.

MIN QIAO was born in 1994. He received the bachelor's degree from Xi'an Eurasia University, in 2017. He is currently pursuing the M.S. degree with the School of Information Science and Engineering, Xinjiang University. His research interests include image processing and object detection.

GANG ZHOU was born in 1981. He received the Ph.D. degree from the School of Information and Communication Engineering, Xi'an Jiaotong University, in 2013. He is now an Associate Professor at Xinjiang University. His current research interests include computer vision, image processing, and pattern recognition.

QIU LING LIU was born in 1995. She received the bachelor's degree from the Hunan Institute of Science and Technology, in 2017. She is currently pursuing the M.S. degree with the School of Information Science and Engineering, Xinjiang University. Her research interests include computer vision and image processing.

LI ZHANG was born in 1988. She received the master's degree from the School of Information Science and Engineering, Xinjiang University, in 2014. She is now a Lecturer at Xinjiang University. Her current research interests include image processing and circuit designing.