Edge distraction-aware salient object detection

Sucheng Ren, South China University of Technology, Guangzhou, 510006, China

Wenxi Liu, Fuzhou University, Fuzhou, 350116, China

Jianbo Jiao, University of Birmingham, B15 2SQ, Birmingham, U.K.

Guoqiang Han, South China University of Technology, Guangzhou, 510006, China

Shengfeng He, Singapore Management University, Singapore, 178903

Published in IEEE Multimedia (2023) 30 (3), 63-73. DOI: 10.1109/MMUL.2023.3235936

Abstract: Integrating low-level edge features has been proven to be effective in preserving clear boundaries of salient objects. However, the locality of edge features makes it difficult to capture globally salient edges, leading to distraction in the final predictions. To address this problem, we propose to produce distraction-free edge features by incorporating cross-scale holistic interdependencies between high-level features. In particular, we first formulate our edge features extraction process as a boundary-filling problem. In this way, we enforce edge features to focus on closed boundaries instead of those disconnected background edges. Second, we propose to explore cross-scale holistic contextual connections between every position pair of high-level feature maps regardless of their distances across different scales. It selectively aggregates features at each position based on its connections to all the others, simulating the "contrast" stimulus of visual saliency. Finally, we present a complementary features integration module to fuse low- and high-level features according to their properties. Experimental results demonstrate our proposed method outperforms previous state-of-the-art methods on the benchmark datasets, with the fast inference speed of 30 FPS on a single GPU.

Keywords: Feature extraction, Image edge detection, Object detection, Visualization, Filling, Task analysis, Convolution

Salient object detection (SOD) aims at segmenting the most distinct objects and regions from images. It serves as a fundamental component for many image processing and computer vision tasks, such as image manipulation6 and people reidentification.[16] Traditional SOD is inspired by cognitive psychology, which aims to design hand-crafted features to capture salient stimuli in images. However, handcrafted features may not be reliable as they are designed based on assumptions for specific scenarios. With the development of deep learning, the power of representation learning boosts the development of SOD, especially after the emergence of fully convolution neural network (FCN).[5] FCN-based methods overcome the limitation of the capability of handcrafted features and detect saliency in an end-to-end manner. Whether an object is salient or not is determined from a global view, and therefore high-level features of a deep network is usually extracted. However, using high-level features always come with dilution of low-level features which leading to coarse salient object boundaries. There are some postprocessing methods, such as superpixel-based filter and conditional random filed for boundary refinement. However, these postprocessing methods are time consuming. As a result, most researchers focus on injecting low-level information in the networks. Latest research works[15,19] explicitly incorporates salient edges to refine the object boundaries. Notwithstanding the demonstrated success, the incorporated edges usually contain background nonsalient edges, which may distract the detection of saliency (see Figure 1). Although they

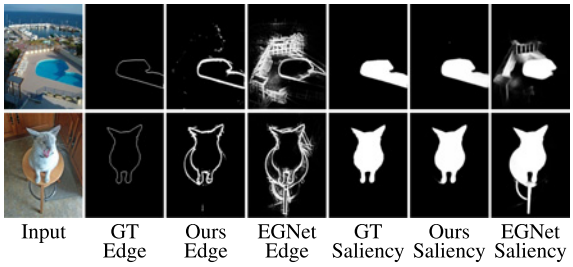


FIGURE 1. State-of-the-art edge-aware methods (e.g., EGNet) suffer from distraction of background edges. We propose to remove distraction from the detection of edges by converting it to a boundary filling problem.

are trained with salient edges data, the naive supervision cannot help distinguishing edges from foreground object or background distraction. Besides, edge feature extraction and integration modules are too heavy and lead to long inference time (less than 10 FPS of EGNet[19]) and memory overload.

We observe that foreground, salient objects exhibit closed boundaries, while background edges are usually disconnected. This observation motivates us to remove edge distraction from a new aspect. We formulate the process of salient edge features extraction as a new boundary filling problem. In this way, the predicted edges should be not only salient, but also restricted by the closed boundary constraint that can be filled to form the shape of objects. In the meantime, with the help of this supervision, our model can extract distraction-aware edge features even with superlightweight decoder, which helps reaching faster inference speed than others.[15,19] On the other hand, high-level contrast contextual information are crucial for both salient edge and object detections to prevent detecting incomplete objects. To this end, we propose a cross-scale holistic contrast features extraction (CSHC) module that explore contextual interdependencies between every position pairs cross feature scale in high-level feature maps regardless distances. It aggregates features according to positional similarities, simulating the "contrast" stimulus[7] in visual saliency. Finally, we present a complementary features integration (CFI) module to fuse low- and high-level features according to their unique properties.

In summary, the contributions of this article are threefold as follows.

- We propose a distraction-aware edge features extraction (DEFE) module and boundary-filling loss that extract edge features in a boundary filling manner. This novel solution enforces the network to produce closed boundaries instead of disconnected ones, and thus removing edge distraction from saliency detection.
- We propose a CSHC module that explores long-range positional similarities cross different feature scale. It simulates the "contrast" stimulus of visual saliency and thus enhancing high-level understanding for different size salient edge and object detections.
- Our proposed model outperforms state-of-the-art methods on six benchmark datasets on three evaluation metrics.

RELATED WORKS

Before deep learning era, SOD methods try to design hand-crafted features, but now it is dominated by deep learning-based methods.[1,2,8,9,10,17,19] Here we concentrate on the latter category.

FCN-Based SOD Method

As Long et al.[5] proposed FCN for dense prediction tasks, recent SOD methods are based on FCN-like structure. Li et al.dealt with SOD and salient segmentation together by integrating both FCN stream and special spatial pooling stream. In the next sections, we review three directions of SOD that inject additional information or explore internal information, i.e., edge-aware method, attention mechanism, and feature integration.

Edge-Aware Method

To solve the coarse boundary problem in SOD, latest research works aim at recovering the structural detail and edge information in SOD. On the one hand, edge/boundary aware losses, such as IoU boundary loss, boundary enhanced loss, and similarity structural similarity loss,[8] integrating the boundary detection into SOD for the more accurate edge of salient object. On the other hand, Zhao et al.[19] explicitly detected edge and used the edge information to complement SOD. Wu et al.[15] mutually considered the SOD and edge detection and propose a stacked cross refinement network to refine both of them. Wu et al.[13] took a multitask intertwined framework to guide SOD by the foreground contour detection and edge detection. Our method takes a different strategy that formulate the edge features extraction process as a boundary filling problem to remove edge distraction.

Attention Mechanism

Inspired by the study of human visual attention, attention mechanism is effective in computer vision tasks, such as image classification[12] and object recognition.[12]

Liu et al.[4] proposed to utilize the global and local pixel-wise information by the global and local attention to learn pixelwise contextual information. The attention mechanism is also used as a gating function.[17] Zhang et al.[18] proposed to learn spatial and channel attention to distinguish foreground and background. Pyramid network takes high-level features and low-level features, and consider the difference between them with the pyramid attention. However, these methods lack the consideration of the "contrast" in visual saliency, which is indeed a crucial stimulus of human visual system. Our method investigates the positional similarity and contrast to stimulate the visual saliency cross different scale and provide the guidance for both salient object and edge feature extraction.

Features Integration

Features between deep layers and shallow layers reveal different levels of details in SOD, and thus a lot of methods propose to integrate them in different manner. Hou et al.[2] proposed short connection and introduce multiscale output and side-output based on HED to refine the captured details. Amulet utilize the multilevel features to predict saliency maps in different resolutions and fuse them to generate the final saliency map. Wang et al.[11] extracted global information recurrently for better integration of high-level contextual information. Zhang et al.[17] passed information flow by a bi-directional structure between different level of features to predict saliency maps. However, these methods integrate different level of features from the same task, preventing the network from generating diverse and complementary features.

DISTRACTION-AWARE EDGE ASSISTED NETWORK

Model Overview

Our distraction-aware edge assisted network (DENet) is illustrated in Figure 2. Given an input image, it is first fed into a five-stage backbone network to extract multiscale features. Then high-level features are first enhanced by our CSHC module to extract by stimulating the "contrast" in visual saliency, while both low-level and enhanced high-level features are processed by our DEFE module for salient edge features. Finally, these enriched features together with multiscale salient object features from decoder are integrated in our CFI for final prediction.

Salient Object Features Extraction

Our model mainly based on FCN encoder–decoder structure and we also take skip connections. The backbone network has four stages adopted from a pretrained ResNet-50. We add a CSHC. Symmetric to the backbone network, the decoder also has four stages with much less channels than EGNet.[19] Each stage has three convolution layers and is followed by two extra convolution layers to generate saliency maps supervised by ground truth.

Distraction-Aware Edge Features Extraction

Given the features generated from the backbone network, we aim to produce distraction-free features with the boundary filling setting. This module takes two sets of features as input, the low-level features of the first stage $\mathcal{F}_{s1}$, and the enriched high-level features produced by our CSHC module (described in Section Cross-Scale Holistic Contrast Features Extraction). This is because detecting closed object boundaries relies not only on local, low-level information, but also a global view of the image. These two sets of features are fused as follows:

$$\mathcal{F}_f = \mathcal{F}_{s1} + \phi(\mathrm{Conv}(Up(\mathcal{F}_h, \mathcal{F}_1)); \theta) \tag{1}$$

where, $\mathcal{F}_f$ denotes the fused features, $Up(a, b)$ indicates interpolating a feature map a to the same size of b, ϕ is the ReLU activation function, and $\mathrm{Conv}(^\star; \theta)$ is convolution layer with parameter θ. Then, $\mathcal{F}_f$ is then fed into a convolution block with three convolution-batch normalization-ReLU layers. The output becomes salient edge features that governed by two tasks, edge detection and boundary filling.

Edge detection can be obtained by simply performing convolution on the salient edge features, and it is trained by salient edge ground truth. On the other hand, the salient edge features are fed to a boundary filling network, which inhibits background nonsalient edges. The boundary filling network is a simplified version of FCN for fast computation. It consists of an input convolution layer followed by ReLU and four stages downsample–upsample process. The input convolution layer has 64 filters of 7×7 kernel size. Each stage includes a convolution layer, a batch normalization layer, and a ReLU activation function. Every convolution layer has a 64 filters of 3×3 kernel size. The downsampling process uses the convolution layer with stride 2 and the upsample process uses bilinear interpolation operation. A final convolution layer with kernel size of 1×1 is used to obtain the output boundary-filling map denoted as S_b. The edge and boundary filling losses are described in Section Boundary Filling Loss.

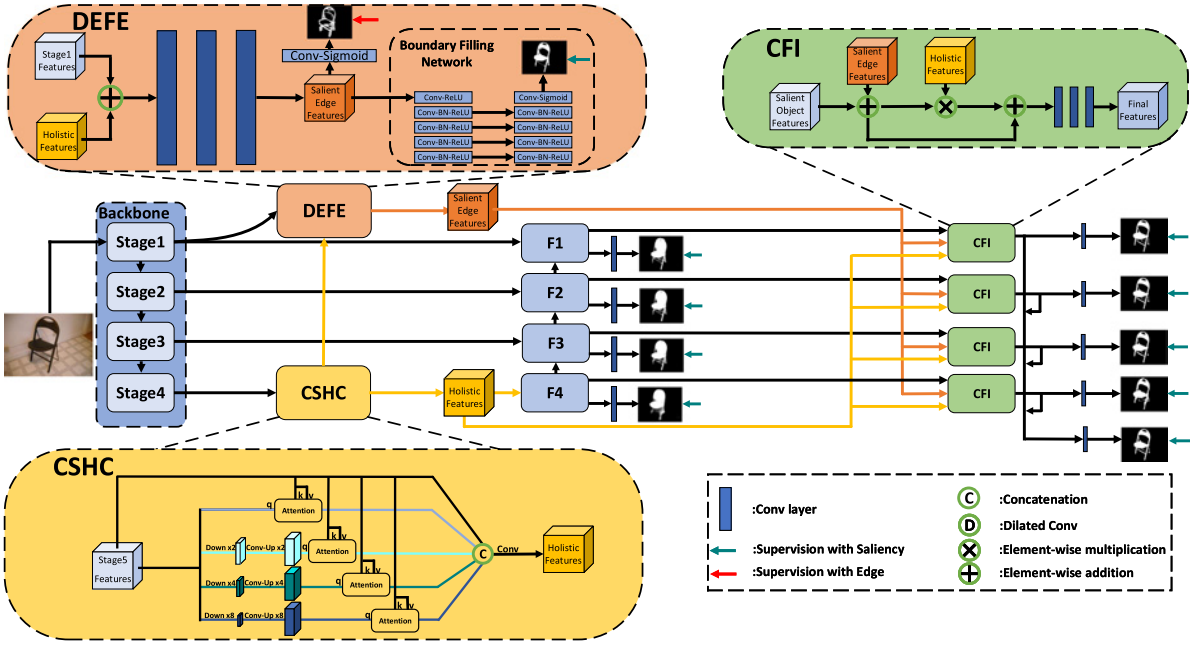


FIGURE 2. Network architecture of our DENet. It consists of three main components, DEFE, CSHC, and CFI. DEFE provides distraction-free edge features supervised by salient edge and boundary-filling network. CSHC stimulate "contrast" in visual saliency and provide enhance high-level features with long range contextual interdependencies over scale. CFI integrate different features according to their own characteristics.

Cross-Scale Holistic Contrast Features Extraction

The feature maps from the fourth stage of the backbone network cover rich semantic information. However, the receptive filed is constant and lacks diversity to capture multiscale contextual information. Comparing with simple fusion cross-scale feature, we aim to further enhance these features to capture cross-scale contextual interdependencies for simulating high-level contrast. We simulate the visual contrast by providing and dynamically adjusting the contrast information of every spatial position pairs across scales.

The feature maps from the fourth stage from backbone denoted as $\mathcal{F}_{s4} \in \mathbb{R}^{c\times h\times w}$ is downsampled–convolution–upsampled with scale rate $\{2, 4, 8\}$ to capture multiscale contextual information $\mathcal{F}^i_{s4}$, where i is the scale rate and $\mathcal{F}_{s4}$ can be regarded as $\mathcal{F}^0_{s4}$. To model cross-scale long range dependency, $\mathcal{F}^i_{s4}$ is feed into different positional attention module as "Query," where $\mathcal{F}_{s4}$ is as "Key" and "Value." In positional attention, "Query," "Key," "Value" are fed into three different linear projection layers to get the feature map $\{Q, K, V\} \in \mathbb{R}^{c'\times h\times w}$, respectively. The contrast map A is calculated by the matrix multiplication between the reshaped and transposed version of $Q \in \mathbb{R}^{hw\times c'}$ and reshaped version of $K \in \mathbb{R}^{c'\times hw}$ followed by a Softmax function:

$$A_{ji} = \frac{\exp(Q_i \cdot K_j)}{\sum_{k=1}^{wh} \exp(Q_k \cdot K_j)} \tag{2}$$

where, A_{ji} denotes the impact of the ith position on the jth position. A large value represents a high similarity (and thus lower contrast) between the two positions. Then we use V to multiply S and reshape it back to $\mathbb{R}^{c'\times h\times w}$ to generate the final output $\mathcal{F}'^i_{s4}$. Comparing with F_{s4}, every positions in F^{ri}_{s4} has a global view of contrast on the whole feature maps in scale rate $\{0, 2, 4, 8\}$, respectively. Finally, we concatenate all cross-scale position enhanced feature and the original feature F_{s4} followed by a convolution layer to generate final CSHC feature F_h. Comparing with F_{s4}, F_h covers different scale contextual information from the long range dependency guidance of different receptive field features.

Complementary Features Integration

In previous we extract salient edges features, holistic contrast features, and salient object features (from the decoder). All these features convey unique and complementary information, and we argue that they should be integrated in a simple, unified manner like Hou et al.,[2] Liu et al.,[3] and

Zhang et al.[17] did. We design our integration module to combine these features according to their special properties. Low-level salient edge features reveal important object boundaries, and therefore we use the addition to emphasize edges in salient object features. Holistic contrast features show rich contextual interdependencies, which helps locating salient objects, so we use multiply operation to inhibit nonsalient background (see Figure 2). The integration is formulated as follows:

$$F_i' = \mathrm{Conv\_block}((F_i + F_e) + F_h \times (F_i + F_e)); \theta) \tag{3}$$

where, F_i denotes the ith stage salient object features, and F_i' represents the final features of the ith stage. Every final features will followed by a convolution layer to output a saliency map S_i' from F_i'.

Besides, we concatenate all enhanced salient object features to generate the fused saliency map, which will be the final saliency map in the inference

$$F_{\mathrm{fuse}} = \mathrm{cat}([F_1,' \ldots, F_k']) \tag{4}$$

$$S_{\mathrm{fuse}} = \mathrm{Sigmoid}(\mathrm{Conv}(F_{\mathrm{fuse}}; \theta)) \tag{5}$$

where, F_{fuse} is the concatenation of all enhanced features, $k = 5$ indicates five stage enhanced features, and S_{fuse} indicates the fused saliency map.

Objective Function

The total loss function $\mathcal{L}$ includes three parts: edge loss $\mathcal{L}_e$, boundary filling loss $\mathcal{L}_b$, and saliency loss $\mathcal{L}_s$

$$\mathcal{L} = \mathcal{L}_e + \mathcal{L}_b + \mathcal{L}_s. \tag{6}$$

Edge Loss

We use the edge map computed from the saliency map and the class-balanced cross entropy loss as the edge loss

$$\begin{aligned}\mathcal{L}_e = &-\beta \sum_{i \in \mathrm{GT}_e^+} \mathrm{GT}_e(i) \log E(i) \\ &- (1-\beta) \sum_{i \in \mathrm{GT}_e^-} (1 - \mathrm{GT}_e(i)) \log (1 - E(i))\end{aligned} \tag{7}$$

where, E is the predicted edge map, GT_e^+ and GT_e^- represent the edge pixel and background pixel of ground truth edge map and $\beta = \frac{\sum \mathrm{GT}_e^-}{\sum \mathrm{GT}_e}$. Cross-entropy loss is widely used in binary classification problem, and due to the small amount of positive samples in edge detection problem, class-balanced cross entropy shows better performance.

Boundary Filling Loss

We use cross entropy loss l_{ce} to measure the output boundary filling result

$$\begin{aligned}l_{\mathrm{ce}}(P, \mathrm{GT}) = &- \sum_{i \in \mathrm{GT}^+} \mathrm{GT}(i) \log P(i) \\ &- \sum_{i \in \mathrm{GT}^-} (1 - \mathrm{GT}(i)) \log (1 - P(i))\end{aligned} \tag{8}$$

where, P is the predicted map from boundary filling network. Besides, we use IoU loss l_{iou} to suppress background edges

$$l_{\mathrm{iou}}(P, \mathrm{GT}) = 1 - \frac{\sum_{i \in \mathrm{GT}} P(i) \mathrm{GT}(i)}{\sum_{i \in \mathrm{GT}} [P(i) + \mathrm{GT}(i) - P(i) G(i)]}. \tag{9}$$

To sum up, the total loss is

$$\mathcal{L}_b = l_{\mathrm{ce}}(S_b, \mathrm{GT}_s) + l_{\mathrm{iou}}(S_b, \mathrm{GT}_s) \tag{10}$$

where, the S_b is the boundary-filling map from DEFE and GT_s is the ground truth saliency map.

Saliency Loss

We need to supervise all the output saliency maps from salient object features extraction module S_i, feature integration module S_i', and fuse feature S_{fuse}. We adopt the cross entropy loss l_{ce} to compute the saliency loss as

$$\begin{aligned}l_{\mathrm{ce}}(S, \mathrm{GT}) = &- \sum_{i \in \mathrm{GT}^+} \mathrm{GT}_e(i) \log S(i) \\ &- \sum_{i \in \mathrm{GT}^-} (1 - \mathrm{GT}(i)) \log (1 - S(i))\end{aligned} \tag{11}$$

$$\begin{aligned}\mathcal{L}_s = &\sum_{i=1}^{4} l_{\mathrm{ce}}(S_i, \mathrm{GT}_s) + \sum_{i=1}^{4} l_{\mathrm{ce}}(S_i,' \mathrm{GT}_s) \\ &+ l_{\mathrm{ce}}(S_{\mathrm{fuse}}, \mathrm{GT}_s).\end{aligned} \tag{12}$$

EXPERIMENTS

Implementation Details

Our model is trained on DUTS-TR with random horizontal/vertical flipping and multiscale training. All the experiments are using an SGD optimizer with momentum of 0.9 and weight decay of 0.0005. The learning rate starts from 5e-4 and divided by 10 after 25 epochs. The four stages of the backbone network from ResNet-50 is pretrained on ImageNet (the same as EGNet[19] and SCRN[15]) and the rest layers are initialized following the default settings of PyTorch. During inference, the image will be resized to 352×352.

TABLE 1. Quantitative comparison with state-of-the-arts on six benchmarks. the best performances are marked in red. our method consistently outperforms previous state-of-the-arts on all datasets. methods can run in real-time (faster than 30 FPS) are bolded.

Method	Param. (M)	Speed (FPS)	ECSSD			DUT-O			HKU-IS			DUTS-TE			PASCAL-S		
			MAE↓	maxF_β ↑	S ↑	MAE↓	maxF_β ↑	S ↑	MAE↓	maxF_β ↑	S ↑	MAE↓	maxF_β ↑	S ↑	MAE↓	maxF_β ↑	S ↑
DSS	62.2	24	0.064	0.903	0.879	0.063	0.741	0.761	0.047	0.899	0.880	0.057	0.819	0.825	0.097	0.818	0.801
Amulet	33.2	15	0.060	0.902	0.891	0.098	0.711	0.785	0.051	0.881	0.884	0.084	0.747	0.803	0.101	0.798	0.814
UCF	24.0	9	0.086	0.897	0.886	0.128	0.731	0.745	0.071	0.878	0.875	0.109	0.775	0.778	0.131	0.810	0.805
R3Net	19	0.052	0.933	0.910	0.073	0.817	0.818	0.046	0.919	0.897	0.067	0.840	0.837	0.102	0.53	0.814	
BMPM	–	29	0.045	0.927	0.911	0.063	0.774	0.809	0.039	0.917	0.911	0.048	0.852	0.861	0.074	0.856	0.845
RAS	20.1	**58**	0.059	0.904	0.896	0.065	0.755	0.802	0.047	0.864	0.889	0.060	0.798	0.937	0.100	0.796	0.802
PiCANet	32.9	8	0.047	0.931	0.916	0.068	0.825	0.811	0.043	0.921	0.811	0.049	0.851	0.860	0.083	0.866	0.850
BASNet	87.1	28	0.037	0.931	0.916	0.056	0.801	0.836	0.032	0.919	0.909	0.047	0.849	0.866	0.078	0.847	0.837
PoolNet	68.3	**36**	0.039	0.941	0.921	0.055	0.821	0.834	0.033	0.928	0.915	0.040	0.881	0.882	0.075	0.878	0.850
PAGE	–	**38**	0.043	0.922	0.912	0.062	0.776	0.823	0.037	0.907	0.903	0.051	0.824	0.855	0.079	0.839	0.840
SCRN	–	16	0.038	0.941	0.927	0.056	0.814	0.837	0.033	0.925	0.919	0.039	0.886	0.884	0.065	0.885	0.868
EGNet	111.7	7	0.037	0.947	0.925	0.053	0.831	0.835	0.031	0.932	0.918	0.039	0.892	0.887	0.075	0.879	0.852
MINET	162.4	**20**	0.033	0.947	0.925	0.055	0.810	0.833	0.028	0.935	0.920	0.037	0.884	0.884	0.064	0.882	0.857
GateNet	–	**18**	0.040	0.949	0.927	0.054	0.816	0.839	0.032	0.935	0.919	0.038	0.890	0.889	0.064	0.888	0.863
EDN	42.9	-	0.034	0.948	0.923	0.050	0.821	0.835	0.027	0.936	0.922	0.035	0.893	0.886	0.062	0.883	0.866
Ours	59.6	**30**	0.031	0.953	0.929	0.048	0.842	0.848	0.026	0.937	0.927	0.035	0.899	0.885	0.061	0.891	0.870

Datasets

We test our model on five widely used benchmarks: HKI-IS, ECSSD, PASCAL-S, DUT-OMRON, and DUTS. HKU-IS consists of 4447 images with high-quality annotations. There are many disconnected objects with different sizes in this dataset. ECSSD contains 1000 semantic objects with complex backgrounds. PASCAL-S has 850 images from the PASCAL VOC segmentation dataset. DUT-OMRON contains 5168 image with various objects and complex backgrounds. DUTS is the largest SOD dataset including 10,533 images for training and 5019 images for testing. For a fair comparison, the performances of other methods are evaluated from the codes or models released by the authors.

Evaluation Metrics

For SOD, we take three widely adopted evaluation metrics, MAE, F-measure (max F_β), and structure-measure (S-measure) to evaluate our methods. MAE is the mean absolute value between the predicted saliency map and ground truth

$$\mathrm{MAE} = \frac{1}{H \times W} \sum_{i=1}^{w} \sum_{j=1}^{h} |S(i,j) - \mathrm{GT}(i,j)| \tag{13}$$

where, H and W are the height and width of the saliency map.

F_β is the weighted average mean of precision and recall

$$F_\beta = \frac{(1+\beta^2) \times \mathrm{Precision} \times \mathrm{Recall}}{\beta^2 \times \mathrm{Precision} + \mathrm{Recall}} \tag{14}$$

where, β^2 is usually set to 0.3 to strengthen the weight of precision.

S-measure consider both region and object structural similarity of saliency map

$$S = \mu * S_o + (1-\mu) * S_r \tag{15}$$

where, S_0 and S_r denotes the region-aware and object-aware structural similarity, respectively. We set μ as 0.5.

To evaluate the effectiveness of our detected edges, we measure the edge map of F1 score by two binarization methods: optimal dataset scale (ODS) sets the same threshold for all edge maps in the dataset, while optimal image scale (OIS) sets threshold for different edge maps in the dataset

$$F = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{16}$$

Salient Object Detection

We thoroughly compare with 14 latest CNN-based SOD methods: DSS,[2] Amulet, UCF, R3Net,[1] DGRL,[11] BMPM,[17] PiCANet,[4] PAGE,[18] BASNet,[8] PoolNet,[3] SCRN,[15] EGNet,[19] GateNet,[20] and EDN.[14]

Quantitative Evaluation

The quantitative performance of our methods and other state-of-the-art SOD methods are shown in Table 1. As can be seen, edge-aware method[8,15,19] has better performances than the other methods without considering object boundary.[2] Specifically, without any preprocessing and postprocessing, our method achieves the best performances on all evaluated datasets in MAE especially on the ECSSD dataset (about 16% improvement). In terms of F-measure, we perform the best in four dataset and achieve about 0.9% average improvements over the second best method EGNet in all six datasets. This indicates the effectiveness of the proposed distraction-aware edges and holistic contrast features. The improvements on S-measure shows our method achieves the better performance in object level and region level.

Qualitative Evaluation

We qualitatively compare our method with state-of-the-arts in Figure 3. On one hand, our method is able to detect crispy boundary and structural details of salient objects. On the other hand, our method can better locate salient objects regardless background distraction. For example, in the second row, our method accurately captures three objects while the others either distracted by background or capture incomplete salient regions.

Time Statistics

We evaluation the average inference time over six datasets on the same platform: Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10 GHz and GTX1080Ti. The results are reported in Table 1. Note that our boundary-filling network is not necessarily activated during inference. Comparing with other edge prediction and edge information assisted methods, such as EGNet[19] and SCRN,[15] our model extracts and complements edge information with high quality and high efficiency. Compared with those methods without integrating boundary information explicitly, such as BASNet,[8] BMPM,[17] and PoolNet,[3] our method has almost the same inference speed but much better performance. Our model does not have any pre-/postprocess, so our model has better time performance than the methods require pre-/postprocess, such as DSS,[2] in practice.

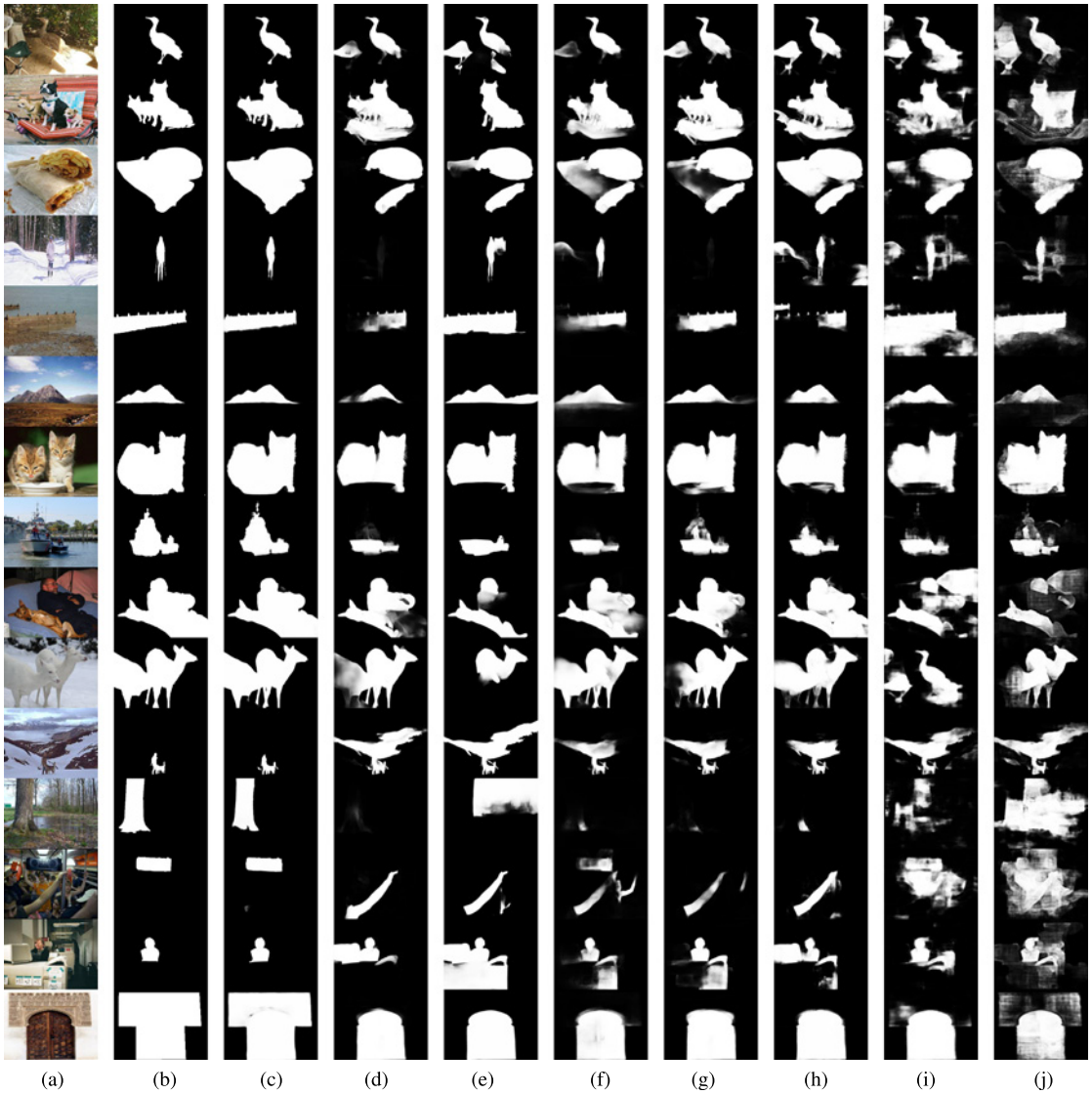


FIGURE 3. Qualitative comparison with state-of-the-art methods. Our proposed method can sharpen salient object boundary and locate objects accurately. (a) Img. (b) GT. (c) Ours. (d) EGNet. (e) BASNet. (f) SCRN. (g) PoolNet. (h) BMPM. (i) Amulet. (j) UCF.

Salient Edge Detection

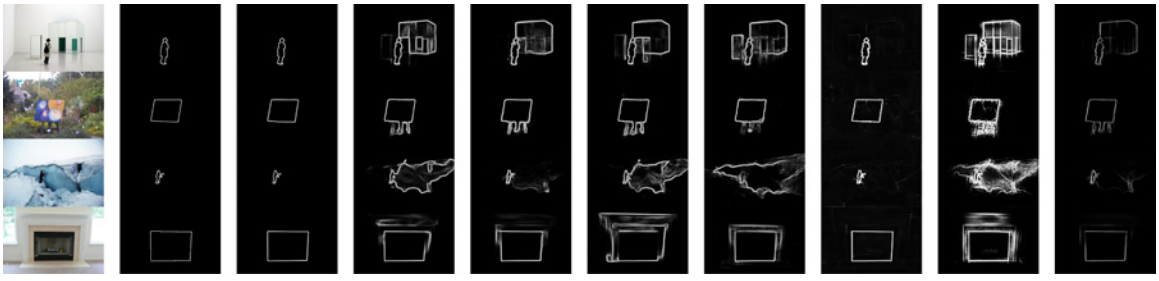
We choose two similar methods for comparing the detected salient edges, SCRN[15] and EGNet,[19] both of which explicit predict edge maps. We are interested in two points, the edge maps predicted by edge branches for measuring the quality of salient edge features, and the edge maps delivered from saliency maps for measuring the performance of salient edge detection. We also test the salient edge detection as a comparison with two methods (PoolNet and R3Net) without considering the boundary. We evaluate them on the DUT-O, HKU-IS, ECSSD datasets.

We report the quantitative results in Table 2. For edge maps predicted directly by edge branches, we achieve the best performance except under metric ODS on the ECSSD dataset. For salient edge measurement (with * that using the edges from salient maps), when comparing with the methods without considering of boundary (PoolNet and R3Net), the three edge-aware methods achieve an average improvement of 13.4% on ODS and 11.6% on OIS. Our methods significantly outperform the rest methods on salient object edge detection.

The visual results are shown in Figure 4. Our methods precisely locate salient objects and sharp details. More importantly, our methods suppress distracted background for all the four examples, leaving those closed salient boundaries.

TABLE 2. Qualitative evaluation on salient edge detection. no * indicates edge maps are derived from saliency maps. * indicates the edge maps are directly predicted by edge branches. our methods significantly boost the detection performances.

Method	DUT-O		HKU-IS		ECSSD	
	ODS	OIS	ODS	OIS	ODS	OIS
R3Net	0.516	0.541	0.609	0.625	0.627	0.643
PoolNet	0.579	0.609	0.703	0.715	0.723	0.736
SCRN	0.594	0.627	0.723	0.740	0.744	0.761
EGNet	0.614	0.646	0.752	0.766	0.771	0.756
Ours	**0.634**	**0.654**	**0.774**	**0.782**	**0.787**	**0.784**
SCRN*	0.461	0.491	0.667	0.683	0.675	0.688
EGNet*	0.496	0.523	0.702	0.714	**0.722**	0.734
Ours*	**0.511**	**0.555**	**0.711**	**0.732**	0.720	**0.741**



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE 4. Salient edge comparison with state-of-the-art methods. Our proposed method suppresses the background edge and predict sharp salient object edge. No * indicates edge maps are derived from saliency maps. * indicates the edge maps are directly predicted by edge branches. (a) Img. (b) GT. (c) Ours. (d) EGNet. (e) SCRN. (f) BMPM. (g) R3Net. (h) Ours* (i) EGNet* (j) SCRN*.

Ablation Study

In this section, we explore the effectiveness of our proposed modules. We test the performance on DUT-OMRON, which is the largest test set and covers many objects of different sizes. We set the model without salient edge features and holistic contrast features (and therefore without features integration) as the baseline.

TABLE 3. Ablation study on the proposed DEFE and CSHC modules. both two modules significantly boosts performance of the plain baseline.

Method	DUT-O		
	MAE$\downarrow$	max F_β $\uparrow$	S $\uparrow$
Baseline	0.073	0.695	0.724
Baseline + DEFE	0.050	0.836	0.822
Baseline + PPM	0.065	0.731	0.769
Baseline + ASPP	0.063	0.737	0.785
Baseline + CSHC	0.056	0.765	0.803
Baseline + DEFE + CSHC	0.048	0.842	0.848

TABLE 4. Ablation study on different features integration.

Settings				DUT-O		
High	Low	CFI	RCFI	MAE$\downarrow$	max F_β $\uparrow$	S $\uparrow$
				0.056	0.765	0.803
✓				0.056	0.774	0.815
	✓			0.052	0.833	0.818
✓	✓			0.049	0.834	0.841
✓	✓		✓	0.051	0.833	0.836
✓	✓	✓		0.048	0.842	0.848

Effectiveness of DEFE

To prove our boundary-filling network and the extracted closed boundary has positive effects, we activate the DEFE only, and the holistic feature are replaced by high-level features that are not enhanced. The architecture of DEFE is similar to EGNet[19] except extra supervision by boundary-fill network and our light weight encoder, decoder. Quantitative result is shown in Table 3. We can see that complementing with distraction-free salient edge features hugely boost the performance of baseline, and it outperforms state-of-the-art methods with this module only (comparing with Table 1).

Effectiveness of CSHC

To evaluate the effectiveness of CSHC, we replace our CSHC by two state-of-the-art multiscale context extraction modules, pyramid pooling module (PPM), atrous spatial pyramid pooling (ASPP). To fit the SOD problem, the dilation rate of ASPP is set as {1, 2, 4, 8}. Results are reported in Table 3. Our CSHC module outperforms the other two, especially on F-measure and S-measure, which reveals the effectiveness of our exploited globally contrast features.

Effectiveness of CFI

To better evaluate the effectiveness of different features and the CFI, we compare the performance on the DUT-OMRON dataset with and without CFI. We use our baseline model with CSHC for all the settings. "High" and "Low" indicates whether supplementing high-level contrast and low-level salient edge features, respectively. No CFI indicates simply concatenating different features. RCFI indicates adding the holistic feature and multiplying salient edge feature in CFI module

As can be seen in Table 4, low-level edge features play a more important role in our method comparing with high-level contrast features. It also means the negative effect of the loss of low-level information is more serious than that of the dilution of high-level information in an FCN setting. Our feature integration module emphasizes the properties of different features has a positive influence.

CONCLUSION

In this article, we delve into the edge distraction problem of SOD. Specifically, we formulate edge features extraction process as a boundary filling problem, and therefore enforcing the detection of closed foreground edges. Moreover, we propose a CSHC module to explore the interdependencies between every position pairs for global view cross feature map scale in salient object/edge detection. All these features are deliberately integrated by our CFI module. Extensive experiments demonstrate the effectiveness of our method.

ACKNOWLEDGMENTS

This project was supported in part by the National Natural Science Foundation of China under Grant 61972162, in part by Guangdong International Science and Technology Cooperation Project under Grant 2021A0505030009, in part by Guangdong Natural Science Foundation under Grant 2021A1515012625, and in part by Guangzhou Basic and Applied Research Project under Grant 202102021074.

REFERENCES

1. Z. Denget al., "R3Net: Recurrent residual refinement network for saliency detection," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 684–690.
2. Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5300–5309.
3. 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 Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3912–3921.
4. N. Liu, J. Han, and M.-H. Yang, "PiCANet: Learning pixel-wise contextual attention for saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3089–3098.
5. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
6. R. Mechrez, E. Shechtman, and L. Zelnik-Manor, "Saliency driven image manipulation," *Mach. Vis. Appl.*, vol. 30, no. 2, pp. 189–202, 2019.
7. H.-C. Nothdurft, "Salience from feature contrast: Additivity across dimensions," *Vis. Res.*, vol. 40, no. 10–12, pp. 1183–1201, 2000.
8. X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: Boundary-aware salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7471–7481.
9. S. Ren, C. Han, X. Yang, G. Han, and S. He, "TENet: Triple excitation network for video salient object detection," in *Eur. Conf. Comput. Vis.*, 2020, pp. 212–228.
10. S. Ren, Q. Wen, N. Zhao, G. Han, and S. He, "Unifying global-local representations in salient object detection with transformer," 2021, *arXiv:2108.02759*.

11. T. Wanget al., "Detect globally, refine locally: A novel approach to saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3127–3135.
12. S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.
13. 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 Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8142–8151.
14. Y.-H. Wu, Y. Liu, L. Zhang, M.-M. Cheng, and B. Ren, "EDN: Salient object detection via extremely-downsampled network," *IEEE Trans. Image Process.*, vol. 31, pp. 3125–3136, 2022.
15. Z. Wu, L. Su, and Q. Huang, "Stacked cross refinement network for edge-aware salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 7263–7272.
16. Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, and S. Z. Li, "Salient Color Names for Person Re-Identification," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 536–551.
17. L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, "A Bi-directional message passing model for salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1741–1750.
18. X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 714–722.
19. J.-X. 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 Int. Conf. Comput. Vis.*, 2019, pp. 8778–8787.
20. X. Zhao, Y. Pang, L. Zhang, H. Lu, and L. Zhang, "Suppress and balance: A simple gated network for salient object detection," in *Eur. Conf. Comput. Vis.*, 2020, pp. 35–51.

SUCHENG REN is a master student in the School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, China. His research interests include computer vision, image processing, and deep learning. Ren received his B.Eng. degree from the South China University of Technology. Contact him at oliverrensu@gmail.com.

WENXI LIU is a professor in the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, 350116, China. His research interests include crowd analysis, visual tracking, and image segmentation. Liu received his Ph.D. degree in computer science from the City University of Hong Kong, Kowloon Tong, Hong Kong. Contact him at wenxiliu@fzu.edu.cn.

JIANBO JIAO is an assistant professor in the School of Computer Science, University of Birmingham, B15 2SQ, Birmingham, U.K. His research interests include computer vision and machine learning. Jiao received his Ph.D. degree in computer science from the City University of Hong Kong, Kowloon Tong, Hong Kong. Contact him at j.jiao@bham.ac.uk.

GUOQIANG HAN is a professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, China. His research interests include multimedia, computational intelligence, machine learning, and computer graphics. Han received his Ph.D. degree from Sun Yat-sen University, Guangzhou, China. Contact him at csgqhan@scut.edu.cn.

SHENGFENG HE is an associate professor in the School of Computing and Information Systems, Singapore Management University, Singapore, 178903. His research interests include computer vision, image processing, computer graphics, and deep learning. He received his Ph.D. degree from the City University of Hong Kong, Kowloon Tong, Hong Kong. He is a corresponding author of this article and a senior member of the IEEE. Contact him at shengfenghe@smu.edu.sg.