Contour-Aware Loss: Boundary-Aware Learning for Salient Object Segmentation

Zixuan Chen, Huajun Zhou, *Graduate Student Member, IEEE*, Jianhuang Lai, *Senior Member, IEEE*, Lingxiao Yang, *Member, IEEE*, and Xiaohua Xie, *Member, IEEE*

***Abstract*—We present a learning model that makes full use of boundary information for salient object segmentation. Specifically, we come up with a novel loss function, i.e., Contour Loss, which leverages object contours to guide models to perceive salient object boundaries. Such a boundary-aware network can learn boundary-wise distinctions between salient objects and background, hence effectively facilitating the salient object segmentation. Yet the Contour Loss emphasizes the boundaries to capture the contextual details in the local range. We further propose the hierarchical global attention module (HGAM), which forces the model hierarchically to attend to global contexts, thus captures the global visual saliency. Comprehensive experiments on six benchmark datasets show that our method achieves superior performance over state-of-the-art ones. Moreover, our model has a real-time speed of 26 fps on a TITAN X GPU.**

***Index Terms*—Salient object segmentation, deep learning, contour, attention.**

I. Introduction

SALIENT object segmentation, which aims to extract the most conspicuous object regions in visual range, has become an attractive computer vision research topic over the decades. A vast family of saliency algorithms has been proposed to tackle the saliency segmentation problem, which distinguishes whether a pixel pertains to a noticeable object or inconsequential background. Due to a mixture of both object and background, pixels closed to the boundary between objects and background are error-prone.

Early methods [1]–[3] determine the saliency by utilizing hand-crafted appearance features. These methods only focus on the low-level visual features so that they are difficult to achieve satisfactory results in the image with a complex scene. Compared with the methods based on hand-crafted features and prior knowledge, the fully convolution network (FCN) based frameworks [4]–[6] made remarkable progress by exploiting high-level semantic information. To learn the binarized saliency segmentation, these networks usually adopt the cross-entropy loss as an objective function. However, cross-entropy loss only considers sample distribution while neglecting the appearance cues of target objects, such as boundaries and inner textures. Especially, the boundary is regarded as very important for saliency segmentation. To leverage the boundary information, multi-task architecture [7], [8] was designed to aggregate the features acquired from boundary and saliency labels. Overall, the aforementioned saliency segmentation methods cannot well exploit boundary information to determine the contour pixels.

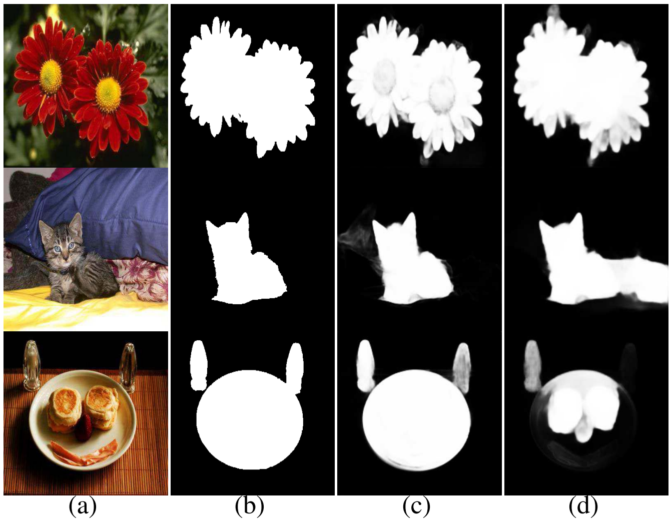


Fig. 1. Visual examples of the proposed method and PiCANet [9]. From left to right: (a) original image, (b) ground truth, (c) ours, (d) PiCANet [9].

Because the decision of salient object regions relies on the spatial contexts, exploiting contextual information in models can lessen the misdirections from insignificant features. Recently, the attention mechanism is exploited to obtain attended features for capturing global contexts by [9]–[11]. However, as these attended features are generated by softmax in global forms, it will emphasize significant pixels and reduce the contrast within other pixels which leads to error segmentation on the object borders in high resolutions. Therefore, for

Manuscript received October 10, 2019; revised May 31, 2020 and August 29, 2020; accepted October 25, 2020. Date of publication November 17, 2020; date of current version November 23, 2020. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2019B010155003, in part by the National Natural Science Foundation of China under Grant 62072482 and Grant 61672544, and in part by the Tip-top Scientic and Technical Innovative Youth Talents of Guangdong Special Support Program under Grant 2016TQ03X263. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sen-Ching Samson Cheung. *(Corresponding author: Xiaohua Xie.)*

The authors are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China, also with the Guangdong Province Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou 510006, China, and also with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Sun Yat-sen University, Guangzhou 510006, China (e-mail: chenzx3@mail2.sysu.edu.cn; zhouhj26@mail2.sysu.edu.cn; stsljh@mail.sysu.edu.cn; yanglx9@mail.sysu.edu.cn; xiexiaoh6@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/TIP.2020.3037536

the high-resolution image, it is not a good choice to capture the global contexts by softmax-wise attention modules.

To address the aforementioned issues of boundary-aware learning and attention mechanism for saliency segmentation, we propose a novel segmentation loss and an effective global attention module, i.e., Contour Loss and hierarchical global attention module (HGAM). The aim of Contour Loss is to guide the network to perceive the object boundaries to learn the boundary-wise distinctions between salient objects and background. Motivated by the focal loss [12], we apply spatial weight maps in cross-entropy loss, which assigns a relatively high value to emphasize the pixels near object borders in training. As a result, the trained model is sensitive to the boundary-wise distinctions in images. Since the Contour Loss focuses on local boundaries, HGAM is proposed to hierarchically attend to global contextual information for alleviating distractions from some inconsequential features. Different from the traditional attention modules which work with softmax, HGAM is based on color contrast thus can capture global contexts in all resolutions. Our baseline model is based on FPN [6] architecture with VGG-16 [13] backbone, which is refined by employing residual blocks instead of simple convolution layers in the decoder module. With the help of the abovementioned techniques, our network yields state-of-the-art performance on six benchmarks.

Followings are the summary of our main contributions:

1) We propose the Contour Loss to guide networks to perceive salient object boundaries. Consequently, boundary-aware features can be obtained to facilitate final predictions on object boundaries.

2) We present the hierarchical global attention module (HGAM) to attend to global contexts for reducing inconsequential feature distractions.

3) We construct a network based on FPN architecture and incorporate those proposed methods for joint training.

4) Comprehensive experimental results and extensive in-depth analysis can explain the outperformance of proposed methods. In addition, our model is very fast which has a speed of 26 fps on an NVIDIA TITAN X GPU.

II. Related Works

Salient object segmentation has been greatly evolved over the past decade. In this section, we only briefly introduce some closely related works. A comprehensive literature review can be found in the very recent surveys [14]–[16].

A. Conventional Methods

Conventional salient object segmentation methods utilize prior knowledge and humanlike intuitive, as well as hand-craft appearance features to capture salient regions. Considering obvious distinctions between salient regions and surrounding backgrounds in pictures, local contrast is used to determine the pixel is conspicuous or not by [17]. Inspired by the effectiveness of color contrast in saliency segmentation, Cheng *et al.* [2] proposed global contrast to capture the salient regions by color statistics features. Jiang and Davis [18] propose to measure the similarities of each pixel and rank the salient regions by a strategy of center prior. To exploit different appearance cues for refining the saliency quality, a multi-level segmentation model is designed by [19], [20] to hierarchically aggregate these cues. As conventional methods only leverage low-level visual features, the lack of semantic information can lead to failures in complex situations.

B. Deep Learning Based Methods

With the development of deep learning, remarkable progress has been made by FCN-based models. Different from conventional methods, high-level semantic features can be exploited by FCNs to achieve better results.

1) Fully Convolutional Networks (FCNs): As the hierarchical features of the network are complementary, FCNs [4]–[6], [21] integrate such hierarchical features for segmentation by lateral connections and skip-layer structures. Long *et al.* [4] first build an FCN for addressing semantic segmentation problems. Ronneberger *et al.* [5] propose the U-Net architecture, which consists of a contracting path, a symmetric expanding path, and lateral connections to integrate features with the same resolution. To exploit the potential of deep features, Lin *et al.* [6] present the feature pyramid architecture and employ hierarchical predictions. Since feature pyramid structures only integrate features from bottom to top in the decoder module, the features from the deep layer cannot leverage the local contexts from shallow layers. Yang *et al.* [22] propose a multi-scale bidirectional network that aggregates features through two-directional pathways to learn and consolidate multi-level context information. In their method, a coarse-to-fine path is also designed by hierarchical supervision in different resolutions. Hou *et al.* [21] propose a saliency architecture by embedding short connections into the skip-layer structures within the framework. These architectures are popularly followed by later related works.

2) Recurrent Structures: Inspired by RNNs, some recurrent structures [23]–[27] have been proposed to tackle saliency segmentation problems. Kuen *et al.* [23] first design a recurrent network with convolution and deconvolution layers to enhance saliency maps from coarse to fine. Liu and Han [24] present a U-Net based architecture, which refines saliency maps by recursively integrating hierarchical predictions. Wang *et al.* [25] utilize saliency results as feedback signals to improve saliency performance. Tang *et al.* [26] propose to recursively integrate multi-level features within a module. In [27], Zhang *et al.* propose a hierarchical recurrent structure to recursively extract and aggregate features. To enhance the saliency performance, Li *et al.* [28] build N cascade stages to orderly refine the intermediate predictions by gradually increasing the resolution of feature maps. For further filtering out the errors of saliency results, a recurrent architecture is employed at each stage. These structures serve as refinement modules to leverage the backward signals by feeding intermediate results as well as learned features to convolutional layers. Although these advanced recurrent structures can better leverage the potentials of hierarchical features, they own a huge computational cost in testing and training.

3) Attention Networks: Attention mechanism aims to adaptively select significant features, in other words, alleviating distractions from some useless features. Since both attention and saliency have similar contextual meanings in pictures, recently many researchers adopt attention mechanism for salient object segmentation. To alleviate the distractions, Wang *et al.* [10] obtain attention maps from encoded features to attend to the global contexts, while Zhang *et al.* progressively utilize both spatial and channel-wise attentions in [11]. Zhang *et al.* [29] adopt gate strategy to select and integrate features in a certain direction. Because softmax may emphasizes important pixels in image as well as reduce the contrast within others in high resolution, softmax-wise attention modules are hard to capture precise global contexts in shallow layers.

To tackle this problem, Liu *et al.* [9] propose global and local attention modules to capture global contexts and local contexts in low-resolution and high-resolution respectively. Chen *et al.* [30] utilize the sigmoid function to capture the global contexts without straight supervisions. Wang *et al.* [31] design a non-local operation to capture long-range dependencies directly by computing interactions between any two positions. Chen *et al.* [32] employ hierarchical predictions as attention maps, which can attend to global contexts in all resolutions.

As not all the features in background regions are helpless for saliency determination especially in deep layers, the predicted maps which are trained to close to annotation masks may lose some crucial information. In contrast to the aforementioned attention modules, the proposed HGAM can not only capture global contexts in all resolutions but also considers some crucial information from background regions.

C. Boundary-Aware Learning

One of the major challenges in saliency segmentation is to determine the conspicuous object boundaries. Some researchers pay attention to this point. To combine global and local contexts including boundaries, Wang *et al.* [33] build two DNNs named DNN_L and DNN_G to capture these two cues respectively. They employ DNN_L and geodesic object proposal to capture the local contexts and boundary information, then use DNN_G to obtain global contrast information.

Since superpixel methods like SLIC [34] can obtain the regions by aggregating adjacent pixels with similar attributes, they are usually adopted to refine saliency results. Yang *et al.* [3] propose a background-prior method, which utilizes superpixel methods to obtain regions and detect salient regions by ranking the similarity of the foreground or background units. To revise the vague boundaries, [35]–[37] employ superpixel algorithms to generate the object contours by these over-segmented regions. Some methods [38], [39] calculate the salient values of superpixel regions by extracting the hand-craft features. Since color contrast can find the conspicuous pixels, deep contrast [40] is learned to find the salient regions from superpixels. Because superpixel relies on the distinction of pixel integration, it cannot well segment the pixels from low contrast regions. Besides, these superpixel-based methods often have a huge computational cost.

Instead of using superpixel methods for boundary determination, recent researches prefer to straightly leverage contour information in an entire framework. As a conventional method, [41] builds a two-stream framework for the mixture of texture and contour. Luo *et al.* [7] and Li *et al.* [8] present a multi-task network architecture based on U-Net, which predicts both saliency and contour maps of the corresponding salient objects. For better parsing different segments, Ding *et al.*'s method [42] generates an attention map by the predicted boundary confidence map. This map is utilized to assign higher value and a lower value to inner pixels and boundary pixels respectively. Therefore, their model puts more attention on pixels within the objects. Since pixels around boundary are hard examples [43], Ding *et al.*'s method [42] which relies on the predicted boundary maps obtains an inferior result due to the accumulated errors of boundary information in training and testing. Besides, thanks to the great distinctions between the segmentation map and boundary map, it leads to inconsistent interference by simply aggregating these features. Therefore, these models are difficult to converge and may generate sub-optimal results.

Compared with the abovementioned boundary-aware methods, the proposed Contour Loss can help the model to perceive the object boundaries by focusing on the boundary pixels, which is more robust and easier to be convergence.

III. Proposed Methods

Our proposed method mainly integrates a basic network with a Contour Loss and a hierarchical global attention module (HGAM), which aims at acquiring boundary-aware features and hierarchically integrating global contexts in all resolutions. We describe our methods and baseline network in the following subsections. The overall network structure is shown in Fig.2.

A. Baseline Network

As shown in Fig.2, the FPN [6] based baseline model mainly consists of two categories of modules: encoder module and decoder module.

For encoder module, we adopt the VGG-16 [13] backbone which is pretrained on ImageNet [44] for image classification. As the resolution of input image I is 224×224, to adapt the saliency segmentation task, we utilize the backbone to extract feature maps at 5 levels, which can be represented as encoded features $F^E = \{E_i, 1 \leq i \leq 5\}$ with the resolution $w_i \times h_i = \frac{224}{2^{i-1}} \times \frac{224}{2^{i-1}}$. Since F^E are extracted at multi-levels, they contain both low-level visual cues and high-level semantic information from different resolutions. To integrate this multi-level information, we transfer F^E to the decoder module.

Because the residual block is better than the pure convolution layer in aggregating the multi-scale features, our decoder module is constructed by 5 residual blocks corresponding to F^E. After the decoder module has received F^E, it generates the residual features $F^R = \{Res_i, 1 \leq i \leq 5\}$ and each Res_i can be formulated as:

$$Res_i = \begin{cases} \delta(\{Res_{i+1}\}^{up\times 2} \oplus E_i; \theta_i^R) & 1 \leq i < 5 \\ \delta(E_i; \theta_i^R) & i = 5 \end{cases} \tag{1}$$

where $\delta(\star; \theta)$ stands for the convolution together with ReLU layers with parameters $\theta = \{W, b\}$. $\oplus$ and $\{\star\}^{up\times 2}$ represent

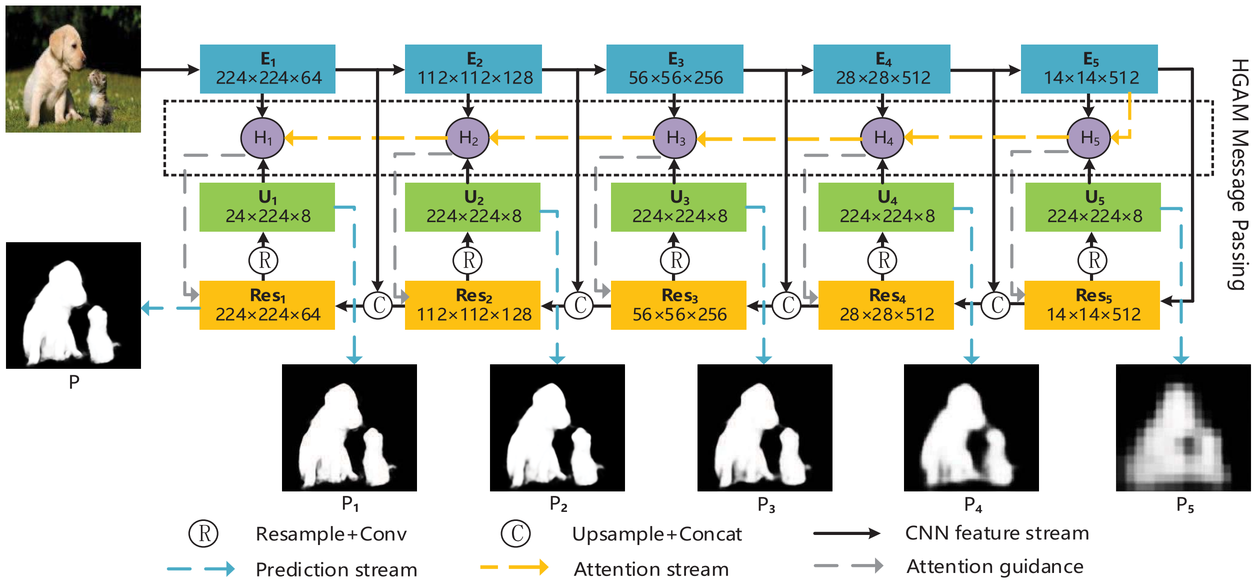


Fig. 2. Overall architecture of the proposed network with VGG-16 [13] backbone. E_i represents the feature of ith level in backbone. Res_i indicates the residual feature generated by the ith residual block. U_i denotes the ith resampled feature with 224 × 224 resolution, and P_i is generated by U_i. H_i is the ith HGAM, which receives E_i, U_i and previous HGAM message to guide Res_i. P denotes the final saliency output generated by guided Res_1.

the channel-wise concatenation and the upsample operation by a factor 2 respectively. To achieve the hierarchical predictions like FPN, we resample F^R to 224 × 224 resolution for obtaining the upsampled features $F^U = \{U_i, 1 \le i \le 5\}$, then utilize these feature maps to generate the hierarchical predictions $F^P = \{P_i, 1 \le i \le 5\}$. U_i and P_i can be formulated as:

$$\begin{aligned} U_i &= \delta(\{Res_i\}^{up\cdot 224}, \theta_i^U) \\ P_i &= \eta(U_i, \theta_i^P) \end{aligned} \tag{2}$$

where $\{\star\}^{up\cdot 224}$ denotes upsampling features to 224 × 224 resolution and $\eta(\star; \theta)$ stands for the convolution together with the Sigmoid layers with parameters $\theta = \{W, b\}$.

As the $P_5, \dots, P_1$ are based on F^R from low to high resolutions, these prediction maps can receive various supervised information from coarse to fine. To better leverage these various feedbacks from loss for updating parameters, in the training phase, the loss is calculated by the weighted sum of F^P like [21], it can be formulated as:

$$Loss(F^P; Y) = \sum_{i=1}^{5} W_i^L \cdot loss(P_i; Y) \tag{3}$$

where Y is the annotation mask, the $loss(\star; Y)$ and $Loss(\star; Y)$ represent the cross entropy loss and its weighted combination respectively. W_i^L is the hyperparameter of corresponding prediction P_i. In application, we adopt the P_1 as a saliency result.

B. Contour Loss

Salient object segmentation aims at extracting the most conspicuous objects in images. Suppose images only contain two parts: the backgrounds and salient objects. For most pixels, they are located at the inside of the objects or background, which indicates that they are far from the object borders. Intuitively, their contexts are relatively pure because the only object or background pixels are shown in receptive fields except for few noise pixels. Consequently, saliency networks can well classify these pixels without auxiliary techniques. However, pixels located at the boundary between background and salient objects are so ambiguous that it is difficult to determine their labels even for experienced people. From the perspective of features, these vectors extracted from motley image pixels fall near the hyperplanes, acting as hard examples. As general saliency networks only apply pixel-wise binary classification, while neglecting the boundary cues and train all pixels equally by cross-entropy loss, they usually predict broad outline of target objects but are inferior in precise boundaries.

Based on the above observations, we argue that border pixels, as well as the hard examples in saliency maps, deserve much higher attention in the training phase. Inspired by focal loss [12], assigning higher weights to focus on these hard examples is theoretically and technologically convincing. Towards this end, we apply spatial weight maps in cross-entropy loss, which assigns a relatively high value to emphasize pixels near the salient object borders. The contour map is generated by a morphological gradient. Specifically, we directly employ a morphological gradient edge detector [45], [46], which calculates the difference between the dilated and the eroded label map. For smoothing the edge learning, inspired by like [47], we also employ the Gaussian blur to contour map for tolerating considerable error rates. The spatial weight map M^C can be formulated as:

$$M^C = Guass(K \cdot (dilate(Y) - erode(Y))) + \mathbb{1} \tag{4}$$

where $dilate(Y)$ and $erode(Y)$ represent dilation and erosion operations with the 5 × 5 kernel respectively. K is a hyperparameter for assigning the high value to contour pixels which is set to 5 empirically. To endow the pixels which are closed to boundaries with a moderate weight, we adopt the Gaussian

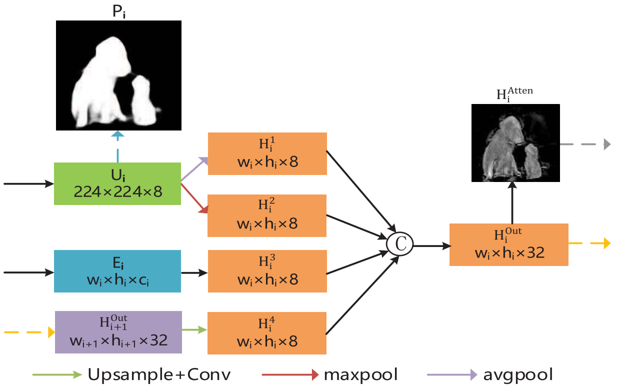


Fig. 3. Inner structure of the ith HGAM. Yellow, blue and gray dotted line represent attention stream, prediction stream and attention guidance respectively. w_i and h_i severally denote width and height of E_i. H_i^{Out} and H_i^{Atten} are the ith HGAM message and attention map respectively.

function with a 5 × 5 range. $\mathbb{1}$ denotes the one matrix with 224 × 224 resolution to set the pixels which are aloof from object boundaries to 1.

Generally, the proposed Contour Loss L^C is implemented as the following formula:

$$L^C = -\sum_{x,y} M^C_{x,y} \cdot (Y_{x,y} \cdot \log Y^*_{x,y} + (1 - Y_{x,y}) \cdot \log(1 - Y^*_{x,y})) \tag{5}$$

where $M^C_{x,y}$, $Y^*_{x,y}$ and $Y_{x,y}$ represent the spatial weight map, annotation map and predicted saliency map of the pixel (x, y) respectively. In the implementation, since our network outputs multiple intermediate saliency maps, Contour Loss is applied to all intermediate maps to supervise network in the training process. In other words, as we adopt Contour Loss, the *loss* in Eq.5 represents L^C.

C. Hierarchical Global Attention Module (HGAM)

Salient object segmentation aims to detect evident object regions, in other words, remove insignificant regions. Although an original picture may contain multi-objects, not all the objects are conspicuous for saliency maps. Thus, negligible information can lead to a sub-optimal result by distracting the models from salient regions.

The attention mechanism is a useful auxiliary module as it can leverage the contextual information and guide the model to abate the insignificant features. However, existing attention modules often adopt the softmax function, which reduces the contrast of inconsequential pixels. In other words, corresponding attention maps may have some ambiguous regions and cannot attend to global contexts in high resolution.

Saliency segmentation can be treated as a special case of attention mechanism, which attends to some primal objects or regions. Therefore, we build saliency-like heat maps auto-generated by the model itself without straight supervision. Our attention module is based on color contrast to capture global contextual information. We briefly introduce the traditional color contrast for saliency segmentation and then present our module in the following.

The saliency region is usually the visual conspicuous regions in the image, in other words, the color distribution of saliency pixels and background pixels has a contrast. Hence a salient pixel $S(I_k)$ can be defined using its color contrast to all other pixels in the image.

$$S(I_k) = \sum_{\forall I_i \in I} D(I_k, I_i) \tag{6}$$

where $D(I_k, I_i)$ is the distance metric between pixels k and i in the image I. Thus, we can select the saliency pixels I'_k by thresholding:

$$I'_k = \begin{cases} 1 & S(I_k) > t' \\ 0 & S(I_k) <= t' \end{cases} \tag{7}$$

where t' is the threshold for selecting the saliency pixels. In some conventional methods like [2] and [17], the saliency region is segmented by color contrast with some color statistics methods and color features.

Inspired by the above contrast method, we employ *feature contrast* modules to find salient pixels in feature maps. Similarly, a region should be significant so that each pixel in that region should have larger value than other pixels, in other words, the inconsequential features often have a smaller value in feature maps. Thus, we can segment these important pixels as well as other insignificant pixels by Eq.6 and Eq.7. To evaluate the distance between two features, we cut off the negative part of L_1 distance to calculate the distance $D(f_k, f_i)$. As a result, we conduct a pixel-wise classification in feature maps, where the positives represent salient features while the negative ones denote inconsequential features. Accordingly, for the given pixel (x, y), the attention map H^{Atten} can be generated as:

$$H^{Atten}_{(x,y)} = \begin{cases} \dfrac{F^{In}_{(x,y)} - (1-\lambda)Aver(F^{In})}{\sqrt{Var(F^{In}) + \epsilon}} & F^{In}_{(x,y)} - Aver(F^{In}) \geq 0 \\ \dfrac{\lambda Aver(F^{In})}{\sqrt{Var(F^{In}) + \epsilon}} & otherwise \end{cases} \tag{8}$$

where F^{In} is the input feature map, $Aver$ and Var represent the average and variance value of F^{In} respectively. λ is set to 0.1 empirically, which means to assign insignificant pixels which are under the average to a small value. ϵ is a small value to avoid zero-division as the default setting. Compared with softmax results, the pixel-wise disparity of our attention maps is more reasonable, in other words, our attention method can clarify conspicuous regions from feature maps in high-resolution. Fig.6 shows the attention maps of HGAM and other softmax-wise attention modules. We can see that HGAM selects the crucial features from feature maps as well as abnegate the insignificant features.

Since attention maps do not hold the labels, they are usually generated or supervised by predicted maps or ground truth masks which only retains the salient regions. However, as models may also need information from background regions to determine salient objects, these attention maps may miss

TABLE I

QUANTITATIVE COMPARISONS OF DIFFERENT SALIENCY MODELS ON SIX BENCHMARK DATASETS IN TERMS OF MAXIMUM F_β-MEASURE AND MAE WHICH ARE MARKED AS F_β^* AND mae IN THIS TABLE. RED, BLUE AND GREEN TEXT INDICATE THE BEST, SECOND BEST AND THIRD BEST PERFORMANCE RESPECTIVELY. THE COMPUTATION SPEED (FPS) ARE OBTAINED ON AN NVIDIA TITAN X GPU. R, V_{19}, N, P, S_1, S_2 REPRESENT THE MODELS WHICH UTILIZE THE RESNET-50, VGG-19, NON-LOCAL BLOCK, PICANET, SIGMOID ATTENTION MAP, AND SOFTMAX ATTENTION MAP EMBEDDED INTO OUR MODEL, RESPECTIVELY. C MEANS THE MODEL TRAINED BY CONTOUR LOSS

	FPS	ECSSD [19]		PASCAL-S [48]		HKU-IS [37]		DUTS-TE [49]		DUT-O [3]		SOD [50]	
		F_β^*	mae	F_β^*	mae	F_β^*	mae	F_β^*	mae	F_β^*	mae	F_β^*	mae
conventional methods													
MR[3]	1.1	0.677	0.173	0.597	0.209	0.620	0.180	0.490	0.220	0.516	0.210	0.584	0.237
DRFI[20]	1.6	0.786	0.164	0.698	0.207	0.777	0.145	0.647	0.175	0.690	0.108	0.697	0.223
ResNet-50 [51] backbone													
SRM[52]	14	0.917	0.054	0.862	0.098	0.906	0.046	0.827	0.059	0.769	0.069	0.845	0.132
BRN[10]	5.9	0.921	0.045	0.861	0.071	0.916	0.037	0.829	0.051	0.790	0.063	0.858	0.104
Ours-R	25	0.936	0.035	0.874	0.067	0.928	0.033	0.871	0.042	0.833	0.054	0.870	0.094
VGG-19 [13] backbone													
PAGRN[11]	–	0.921	0.064	0.861	0.092	0.922	0.048	0.857	0.055	0.806	0.072	–	–
Ours-V_{19}	26	0.936	0.034	0.881	0.064	0.933	0.031	0.870	0.044	0.816	0.060	0.878	0.092
VGG-16 [13] backbone													
RFCN[25]	9	0.898	0.095	0.850	0.132	0.898	0.080	0.783	0.090	0.738	0.095	0.807	0.166
Amulet[27]	16	0.915	0.059	0.828	0.103	0.896	0.052	0.778	0.085	0.743	0.098	0.808	0.145
UCF[53]	23	0.911	0.078	0.846	0.128	0.886	0.074	0.771	0.117	0.735	0.132	0.803	0.169
NLDF[7]	12	0.905	0.063	0.845	0.112	0.902	0.048	0.812	0.066	0.753	0.080	0.842	0.130
RA[32]	35	0.918	0.059	0.834	0.104	0.913	0.045	0.826	0.055	0.786	0.062	0.844	0.124
RA-C	35	0.924	0.050	0.850	0.091	0.920	0.044	0.844	0.047	0.788	0.065	0.853	0.113
CKT[8]	23	0.910	0.054	0.846	0.081	0.896	0.048	0.807	0.062	0.757	0.071	0.829	0.119
CKT-C	23	0.918	0.049	0.863	0.070	0.895	0.043	0.817	0.055	0.766	0.068	0.840	0.108
BMP[29]	22	0.928	0.044	0.862	0.074	0.920	0.038	0.850	0.049	–	–	0.851	0.106
PiCANet[9]	5.6	0.931	0.047	0.873	0.088	0.921	0.042	0.851	0.054	0.794	0.068	0.855	0.108
PiCANet-C	5.6	0.929	0.049	0.880	0.073	0.927	0.038	0.854	0.049	0.807	0.062	0.868	0.096
Ours-N	2	0.933	0.040	0.880	0.062	0.929	0.035	0.876	0.048	0.827	0.057	0.860	0.102
Ours-P	5.2	0.931	0.035	0.875	0.077	0.925	0.034	0.853	0.051	0.807	0.059	0.863	0.098
Ours-S_1	26	0.931	0.042	0.878	0.068	0.922	0.038	0.862	0.048	0.814	0.062	0.864	0.010
Ours-S_2	26	0.925	0.045	0.875	0.067	0.926	0.036	0.859	0.047	0.817	0.061	0.854	0.109
Ours	26	0.933	0.037	0.883	0.063	0.932	0.031	0.872	0.042	0.825	0.058	0.873	0.095

some crucial information. As shown in Fig.7, in addition to focusing on the salient regions, the model also discovers some useful information from the background. It demonstrates that many pixels in the background are also helpful for segmentation.

In our method, the feature maps approximately to the predictions are exploited to generate unsupervised attention maps. Therefore, our attention maps can not only leverage the strong feedbacks of supervised information to update themselves but also contain the background information which may be crucial for saliency determination.

Along the above-mentioned line, we propose our hierarchical global attention module (HGAM) to capture the multi-scale global contexts. As shown in Fig.3, for a given HGAM H_i, it receives three inputs: the encoded feature E_i, the upsampled feature U_i which is close to the prediction P_i, and the previous HGAM message H_{i+1}^{Out}. To extract the global contextual information from input features, we adopt max-pooling and average-pooling layers to deal with U_i for obtaining contextual features H_i^1 and H_i^2 respectively, which is suggested by [54]. We also make channel-wise compression of E_i to obtain H_i^3, while H_i^4 can be generated by the previous HGAM message H_{i+1}^{Out} as:

$$H_i^4 = \begin{cases} \delta(\{H_{i+1}^{Out}\}^{up\times 2}; \theta_4^{H_i}) & 2 \le i < 5 \\ \{\delta(maxpool(E_5; \theta_4^{H_i}))\}^{up\times 2} & i = 5 \end{cases} \tag{9}$$

After obtaining $H_i^1, \ldots, H_i^4$, we make channel-wise concatenation of them to generate the ith HGAM message H_i^{Out}. The ith attention map H_i^{Atten} can be also generated by Eq.8 with H_i^{Out} as input feature. H_i^{Out} is transferred to next HGAM H_{i-1}, while H_i^{Atten} is utilized to guide the Res_i as:

$$Res_i^G = H_i^{Atten} \odot Res_i \tag{10}$$

where Res_i^G is the guided feature map and $\odot$ represents the element-wise multiplication.

IV. EXPERIMENTS

A. Experimental Settings

Different from the baseline network utilizes P_1 as final output, since H_1^{Atten} guides Res_1 by recursively aggregating the multi-scale HGAM messages, we exploit the guided feature

Res_1^G to generate the final prediction P, which is also included in F^P. Therefore, in training, $Loss$ in Eq.5 can be rewrote as:

$$Loss(F^P; Y) = loss(P; Y) + \sum_{i=1}^{5} W_i^L \cdot loss(P_i; Y) \quad (11)$$

In application, we adopt P as our final prediction to evaluate our model.

1) Evaluation Datasets: To evaluate the performance of our model, six public saliency segmentation datasets are exploited. **DUTS** [49] is a large scale saliency benchmark dataset that contains 10,553 images as a training set (DUTS-TR) and 5,019 images as testing set (DUTS-TE). In the experiments, we adopt DUTS-TR to train our model and DUTS-TE for evaluation. For comprehensive evaluation, we also utilize **SOD** [50], **PASCAL-S** [48], **ECSSD** [19], **HKU-IS** [37] and **DUT-O** [3] for testing, which contain 300, 850, 1,000, 4,447 and 5,168 images respectively. Note that for the testing on the abovementioned databases, no corresponding fine-tuning is carried.

2) Implementation Details: Our experiments are based on the Pytorch [55] framework and run on a PC machine with a single NVIDIA TITAN X GPU (with 12G memory).

For training, we adopt DUTS-TR as the training set and utilize data augmentation, which resamples each image to 256 × 256 before random flipping, and randomly crops the 224 × 224 regions. We employ stochastic gradient descent (SGD) as the optimizer with a momentum 0.9 and a weight decay 1e-4. We also set the basic learning rate to 1e-3 and finetune the VGG-16 [13] backbone with 0.05 times smaller learning rate. Since the saliency maps of hierarchical predictions are coarse to fine from P_5 to P_1, we set the incremental weights with these predictions. Therefore $W_5^L, \ldots, W_1^L$ are set to 0.3, 0.4, 0.6, 0.8, 1 respectively in both Eq.5 and 11. The minibatch size of our network is set to 10. The maximum iteration is set to 150 epochs with the learning rate decay by a factor of 0.05 for every 10 epochs. As it costs less than 500s for one epoch including training and evaluation, the total training time is below 21 hours.

For testing, follow the training settings, we also resize the feeding images to 224×224 and only utilize the final output of P. Since the testing time for each image is 0.038s, our model achieves 26 fps speed with 224 × 224 resolution.

3) Evaluation Metrics: To evaluate different algorithms, we adopt three metrics for the quality of saliency maps, including the precision-recall (PR) curves, F_β-measure [56], and mean absolute error (MAE).

To evaluate the robustness of saliency results in different thresholds, we utilize the PR curve to demonstrate the relation of precision and recall by thresholding saliency maps from 0 to 255.

The F_β-measure is a weighted combination of precision and recall value for saliency maps, which can be calculated by

$$F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \quad (12)$$

where β^2 is set to 0.3 as suggested in [56]. To alleviate the unfairness caused by different thresholds in papers, we report the maximum F_β-measure as suggested by [7], [29], which selects the best score overall thresholds from 0 to 255.

For comprehensive comparisons, we also adopt the MAE metric to evaluate the pixel-wise average absolute difference between the saliency map S and its corresponding ground truth mask G,

$$MAE = \frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j=1}^{h} |S_{ij} - G_{ij}| \quad (13)$$

where w and h represent the width and height of a given picture respectively.

B. Comparison With State-of-the-Arts

To evaluate the performance, we compare our method with 13 state-of-the-art algorithms on the aforementioned six public benchmarks in terms of visual evaluation, PR curve, maximum F_β-measure, and MAE metrics. These methods include 2 conventional algorithms: DRFI [20], MR [3], as well as 11 deep learning models: RFCN [25], Amulet [27], UCF [53], NLDF [7], SRM [52], PAGRN [11], BRN [10], CKT [8], BMP [29], PiCANet [9] and RA [32].

1) Visual Comparison: The visual comparison between ours and other state-of-the-arts is shown in Fig.4. It can be observed that our method well detect the target objects in various situations, i.e., containing the object too huge or too small (rows 1 and 2), object touching image edges (row 1), object touching other inconsequential items (row 3), multi-objects (row 4) and object appearance similar with the background (row 5). It is also worth noting that our results have finer boundaries and more precise localization of salient regions, which thanks to the effect of Contour Loss and HGAM respectively. These details make our results live and easier to comprehend the original images, especially when the picture contains the complex foregrounds with multi-objects.

2) F-Measure and MAE: In Table.I, we show quantitative evaluation results between ours and other superior methods under maximum F_β-measure and MAE metrics. As utilizing the same backbones, our model can surpass all existing networks without any post-processing methods like CRF [57]. Moreover, only exploiting the VGG-16 pre-trained model as a backbone can apparently refresh state-of-the-art performance on benchmarks by 1 to 2 percent. Besides, the table also tells us that the proposed HGAM can favorably against the softmax-based and sigmoid-based attention modules on those benchmarks.

Moreover, since our approach even better over more difficult datasets, such as SOD [50], PASCAL-S [48] which have multi-objects with a complex background, and DUTS-TE [49], DUT-O [3] which contain a large number of various situations, our superior results indicate that our method can detect the salient regions in numerous complex situations, while other methods often fail.

3) PR curve: In Fig.5, we compare our approach with other state-of-the-art methods in terms of PR curve on 4 benchmarks. It can be observed that our model consistently outperforms all the other methods.

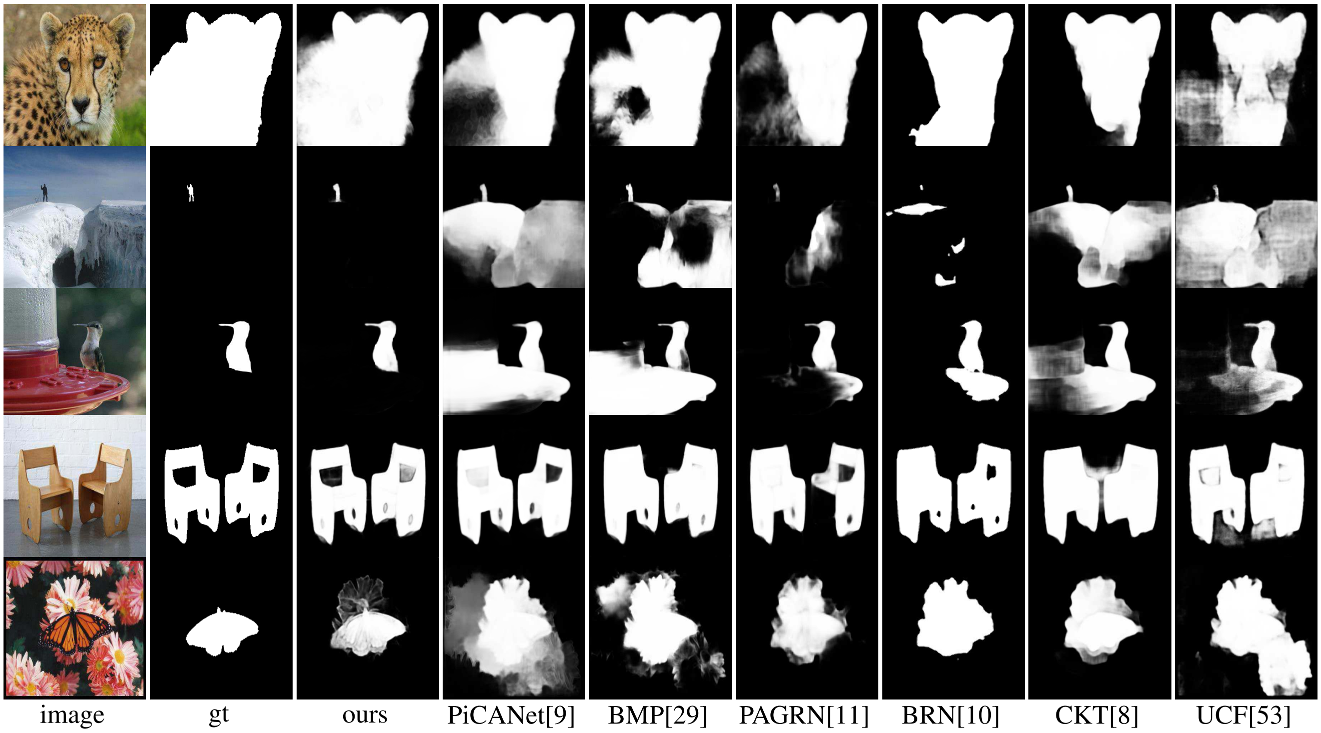


Fig. 4. Visual comparison between ours and other state-of-the-art methods.

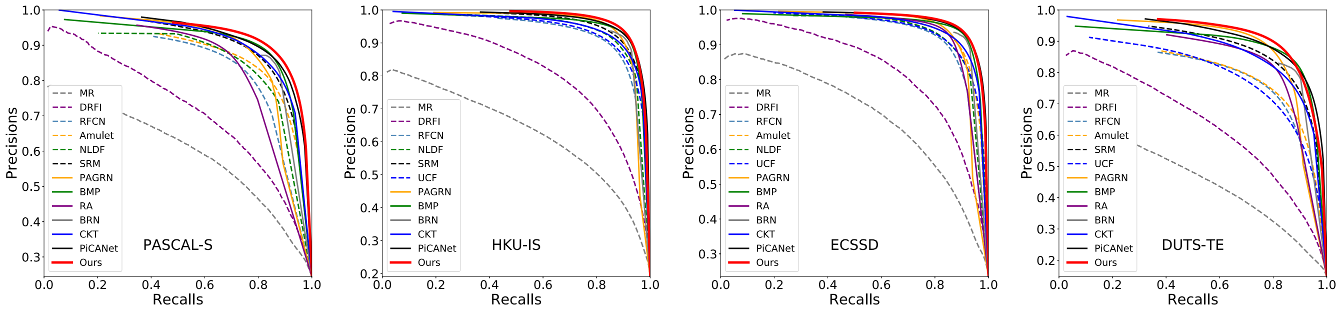


Fig. 5. PR curves of ours and other state-of-the-art methods.

Thanks to the effect of both Contour Loss and HGAM, the boundary and salient region of our results are more precise than other methods, which lead to higher precision value. Therefore our results achieve a higher PR curve than other methods.

4) Computational Cost: Table.I also provides the average testing time for each image among the state-of-the-art on an NVIDIA TITAN X GPU. We can see that our approach only takes 0.038s (corresponding to 26fps) to generate a saliency map, which is faster than other mainstream methods.

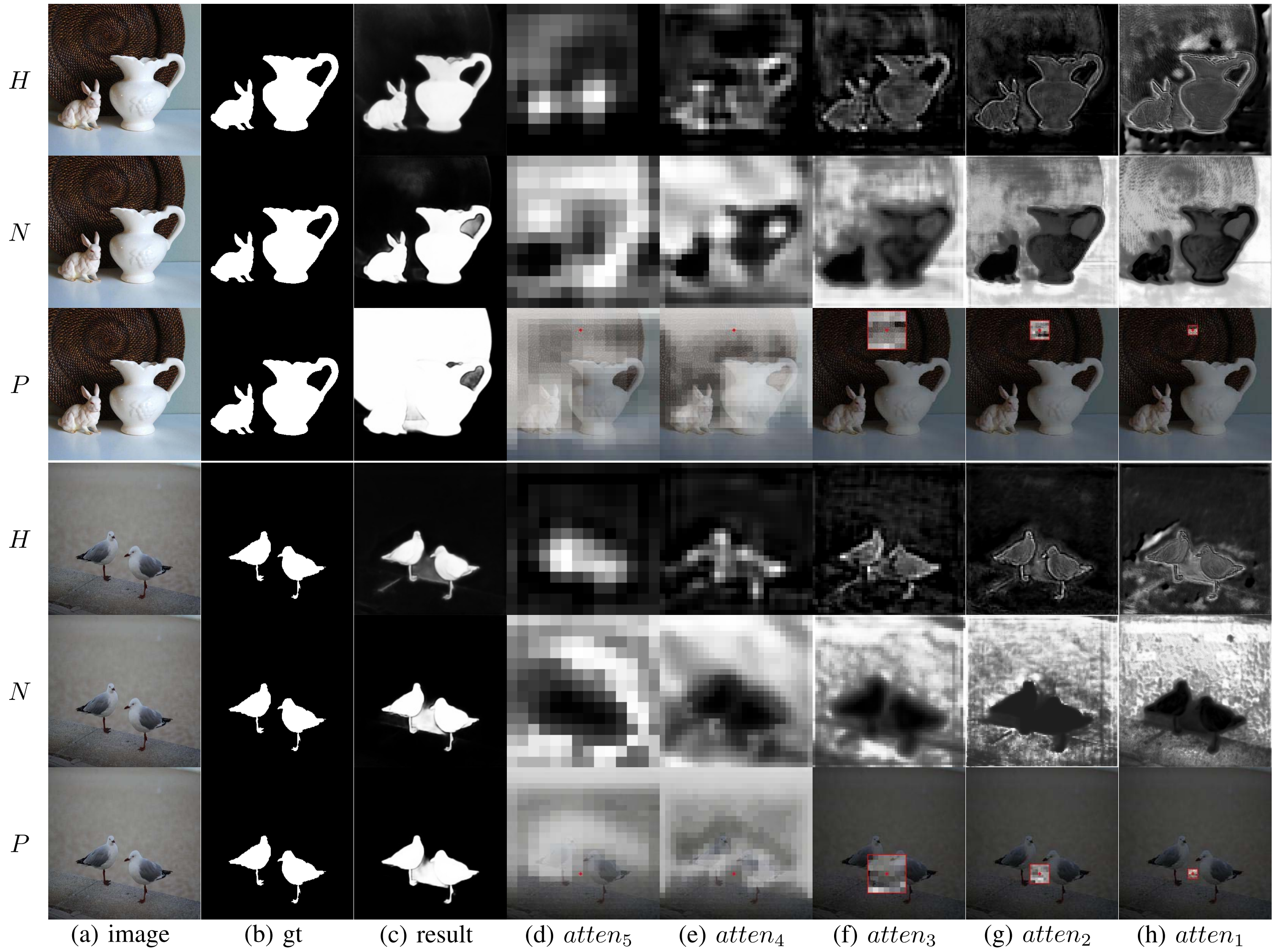
Because feature channels of ResNet-50 [51] from 2 to 5 layers are larger than VGG-16 [13] and VGG-19 [13], using ResNet-50 as backbone is more time consuming in feature concatenation. Therefore the fps of Ours-R is slightly smaller than Ours and Ours-V_{19}, although ResNet-50 is faster than VGG16 and VGG19. Similar findings can be found in [58]. Their VGG based model shows slightly faster speeds than the ResNet-50 based one.

TABLE II

COMPARISON OF DIFFERENT SETTINGS IN TERMS OF MAXIMUM F_β-MEASURE AND MAE METRIC, WHICH ARE MARKED AS F_β^* AND mae IN THIS TABLE. $\mathcal{B}$, $\mathcal{C}$ AND $\mathcal{H}$ REPRESENT THE BASELINE NETWORK, CONTOUR LOSS AND HGAM RESPECTIVELY. **BOLD** TEXT INDICATES THE BEST PERFORMANCE IN TABLE

	DUTS-TE[49]		DUT-O[3]	
	F_β^*	mae	F_β^*	mae
$\mathcal{B}$	0.848	0.050	0.787	0.070
$\mathcal{B}$+$\mathcal{C}$	0.861	0.044	0.806	0.063
$\mathcal{B}$+$\mathcal{H}$	0.860	0.048	0.801	0.069
ours($\mathcal{B}$+$\mathcal{C}$+$\mathcal{H}$)	**0.872**	**0.042**	**0.825**	**0.058**

5) Comparison Between Contour Loss, CKT [8] and Weighted Cross Entropy Loss [59]: Both our method and CKT use boundary information for segmentation. Our Contour Loss combines object boundary within a single segmentation map while CKT predicts both saliency and boundary maps of the corresponding salient objects. Contour Loss aims to refine the saliency map by using a precise boundary. As shown in Fig.4, although CKT uses more data for training, our model is still better in boundary regions than CKT. Table I shows



(a) image (b) gt (c) result (d) $atten_5$ (e) $atten_4$ (f) $atten_3$ (g) $atten_2$ (h) $atten_1$

Fig. 6. Visual comparison between HGAM and other attention methods. H, N and P represent embedding HGAM, Non-local and PiCANet attention modules into our model, respectively. For PiCANet, (d) and (e) represent the global version, while (f), (g) and (h) represent the local version. Because the channels of Non-local attention map corresponds to image space, here we show the map related to the pixel inside the object.

the performance of CKT trained by Contour Loss. Compared with the prototype CKT, Contour Loss can improve CKT performance by nearly 1 percent on F_β^*, which proves the effectiveness of Contour Loss.

Although [59] and Contour Loss are both weighted loss, they have differences. First, [59] aims to balance the positive examples and negative examples, while Contour Loss puts more weights on boundaries. Second, [59] leverages the number of positives and negatives to generate the weight, which means it will degrade to cross-entropy loss if the quantity of positives is equal to negatives. However, Contour Loss focuses on semantic boundary, it will not change its behavior even the number of boundary pixels is equal to the one of inside pixels.

6) Validate the Efficiency of Contour Loss: To comprehensively evaluate the efficiency of Contour Loss, we conduct a special experiment of training RA [32] and PiCANet [9] by using Contour Loss. As demonstrated in Table I, both RA and PiCANet can increase their performance by utilizing Contour Loss, for example RA has nearly 1 percent rise over six benchmark datasets.

7) Our Method vs. BFP [42]: BFP focuses on the task of scene segmentation, while our method aims at segmenting out the saliency object from an image. To infer the scene segments as well as their semantics simultaneously, BFP exploits boundary information to emphasize the importance of inner pixels of objects but alleviates the boundary distractions. In contrast, our method puts more attention on boundary pixels by treating them as hard samples for segmentation learning. Moreover, BFP uses a few Unidirectional Acyclic Graphs (UAGs) to capture global contexts which increase a large computational cost.

8) Comparison Between HGAM, PiCANet [9] and Non-Local Attention [31]: All HGAM, PiCANet and Non-local modules can be embedded into networks for emphasizing useful features as well as neglect insignificant ones. However, due to the difference of attention mechanism and structure, embedding HGAM, PiCANet and Non-local blocks into network respectively can obtain different inference results and computational costs.

Briefly speaking, both Non-local module and PiCANet computes the pair-wise similarities among all possible locations including spatial regions and channels from a feature map, while HGAM evaluates the feature contrast of each pixel only in spatial. Although PiCANet, including a global version and

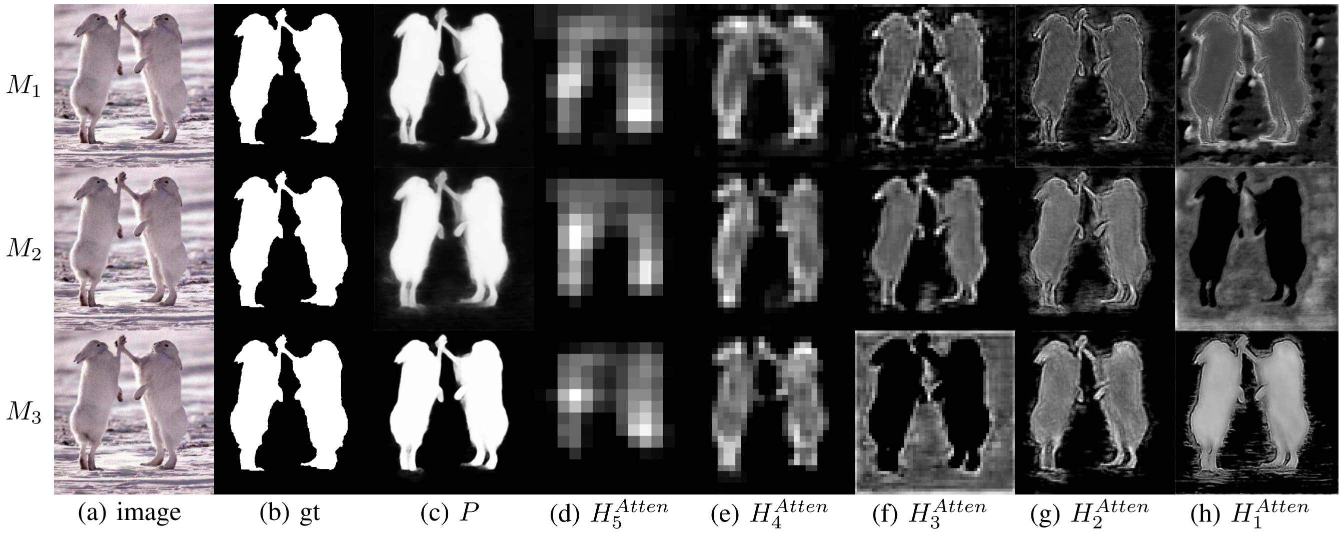


Fig. 7. Visualization of attention maps in the same model structure with different trained parameters. M_1, M_2, and M_3 represent three different trained parameters respectively.

a local version, computes pair-wise similarities in a small region, the embedded ReNet [60] still consumes a lot. In comparison, HGAM requires less time consumption. Besides, different from PiCANet and Non-local module which are isolated in each layer, HGAM owns a bottom-to-top pathway to integrate the contexts from the previous ones. Those aggregated contextual features can filter out the errors to refine the attention maps. Furthermore, HGAM works identically for all resolution, which is more flexible than PiCANet.

To demonstrate the effectiveness of HGAM, we conduct experiments and show all results in Table I. The proposed HGAM outperforms PiCANet and Non-local module in terms of accuracy and speed. Note that, results of PiCANet and Non-local module in Table I obtained by directly replacing our HGAM module with the corresponding attentions. Due to the massive computational cost of Non-local operation, the speed of Non-local module is the slowest one over three attention modules.

Fig.6 visualizes attention maps generated by HGAM, Non-local block and PiCANet, respectively. Obviously, instead of focusing on the target objects, Non-local block prefers to capture background information (the 2nd and 5th rows from (d) to (h)). Moreover, since softmax function reduces the contrast within insignificant features, the features closed to boundaries become ambiguous and lead to inferior results. For example, Non-local attention provides a low contrast between feature on object boundary and surrounding backgrounds (the 2nd and 5th rows in (h)), hence fails to capture object details and complex backgrounds as shown in Fig.6 2(c) and 5(c) respectively. Besides, the global PiCANet only captures the global contexts on feature maps in particular resolution so that it cannot attend more details within objects (3th and 6th from (d) to (e)). The local PiCANet only focuses on local ranges in high resolution so that it cannot leverage the global contexts to understand the distinctions between target objects and backgrounds (3th and 6th rows from (f) to (h)). Therefore, PiCANet is likely to fail in multi-objects situations (3rd and 6th rows in (c)). Besides, Non-local module is isolated in each layer, it cannot straightly leverage the knowledge learned by previous modules. It is noted that, even though Non-local module well determines the background regions in previous layer (2nd in (e) and 5th in (f)), it still makes error segmentation in backgrounds. Different from Non-local and PiCANet modules, our HGAM has a bottom-to-top pathway to integrate contextual information, thus it gradually refines the segmentation of objects by the assistance of previous attended information (1st row from (d) to (h)). Besides, HGAM pays more attention to boundary regions thus it owns high contrast between the boundary and surrounding backgrounds (1st and 4th rows from (g) to (h)). Also, HGAM employs a cut-off operation (Eq.8) to further reduce the importance of insignificant pixels. Hence, HGAM can perceive the distinctions between objects and backgrounds to well segment objects (1st and 4th rows in (c)).

C. Ablation Study

To evaluate the effectiveness of the proposed Contour Loss and HGAM, we show the results of quantitative and visual comparison under different settings. Table.II shows the quantitative comparison which demonstrates that only utilizing Contour Loss or HGAM can enhance the baseline performance by nearly 2 percent. As incorporating Contour Loss and HGAM can make a further improvement on two massive datasets by 1 to 2 precent, which proves that Contour Loss and HGAM refine the saliency results from different aspects. Since our result outperforms the other results both in boundaries and background elimination, it proves that incorporating Contour Loss and HGAM can lead to mutual promotion in training.

D. HGAM Visualizaiton

As shown in Fig.7, we visualize the attention maps generated by HGAM to further understand how it works. We train our model three times and name them as M_1, M_2 and M_3, respectively. Although M_1, M_2 and M_3 have the same model structure, their HGAM attention maps are different because the training involves some random process (e.g., random

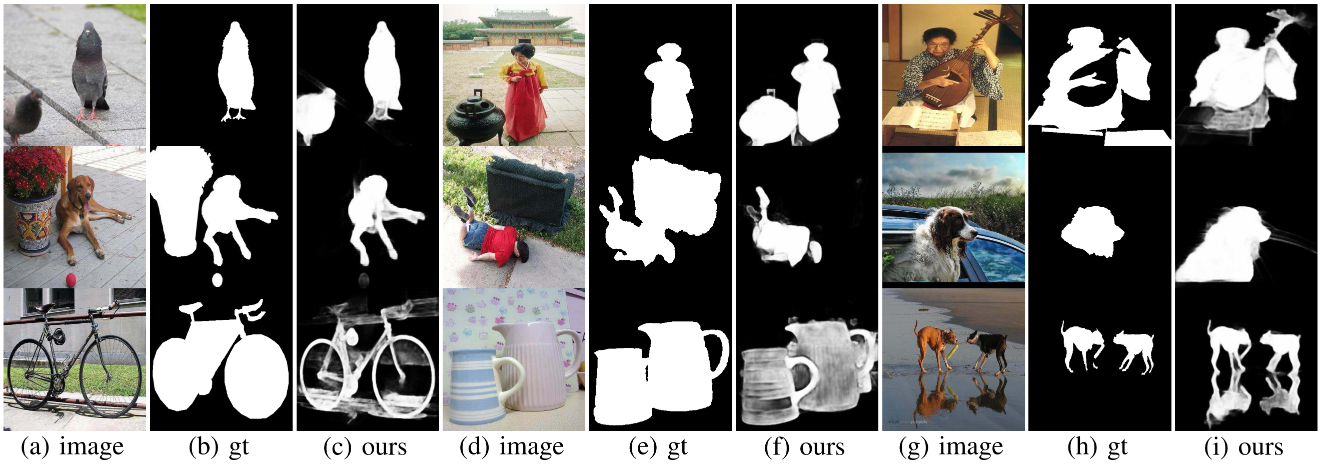


Fig. 8. Failure examples.

data shuffling, random initialization). As shown, all HGAMs preserve clear boundaries to distinguish the foreground and background. We can also observe that these attention maps show fine-to-coarse locations of salient objects from (d) to (h), which greatly matches the global attention mechanism in different resolution. It is worth pondering that, different from other attention maps which only focuses on salient regions, (h) in M_2 and (f) in M_3 assign a higher value to the background regions. Because our HGAMs are optimized by the loss from the top layers, without being directly supervised by ground-truth labels, our model sometimes automatically discover many useful information from backgrounds. In summary, our model puts more weights on backgrounds by chance due to the random process.

As P well abnegates these background regions, we reckon that the model also needs to perceive background regions for eliminating the insignificant features. Moreover, as H_1^{Atten} shows the clear boundaries of salient objects, it is convincingly proved that the mutual promotion of Contour Loss and HGAM in boundary-aware learning.

E. Failure Example

Fig.8 shows some failure examples. It can be observed that our method fails in some situations, i.e., multi-objects including salient objects and other inconsequential items (1st and 2nd rows in (a) to (f)), object containing salient and insignificant parts (1st and 2nd rows in (g) to (i)). Also, our model is sensitive to object boundary due to the Contour Loss, it may be misleaded by object contextures to obtain an inferior result within object. (3rd row in (a) to (f)). Besides, our model cannot perform well in specular reflection situation, the reflections are segmented as salient objects (3rd row in (g) to (i)).

V. Conclusion

We propose the Contour Loss and the HGAM to help networks learn to better detect saliency objects in visual range. The Contour Loss forces to learn boundary-wise distinctions between salient objects and background, while HGAM enables the models to capture global contextual information in all resolutions. Experimental results on six datasets demonstrate that our proposed approach outperforms 13 state-of-the-art methods under different evaluation metrics.

References

[1] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[2] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, “Global contrast based salient region detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.

[3] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, “Saliency detection via graph-based manifold ranking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3166–3173.

[4] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[5] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervent. (MICCAI)*, 2015, pp. 234–241.

[6] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[7] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P.-M. Jodoin, “Non-local deep features for salient object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6609–6617.

[8] X. Li, F. Yang, H. Cheng, W. Liu, and D. Shen, “Contour knowledge transfer for salient object detection,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 370–385.

[9] N. Liu, J. Han, and M.-H. Yang, “PiCANet: Learning pixel-wise contextual attention for saliency detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3089–3098.

[10] T. Wang *et al.*, “Detect globally, refine locally: A novel approach to saliency detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3127–3135.

[11] 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.

[12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[13] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[14] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, “Salient object detection: A survey,” *Comput. Vis. Media*, vol. 5, pp. 117–150, Jun. 2019.

[15] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, “Salient object detection: A benchmark,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5706–5722, Dec. 2015.

[16] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, “Salient object detection in the deep learning era: An in-depth survey,” 2019, *arXiv:1904.09146*. [Online]. Available: http://arxiv.org/abs/1904.09146

[17] D. A. Klein and S. Frintrop, “Center-surround divergence of feature statistics for salient object detection,” in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2214–2219.

[18] Z. Jiang and L. S. Davis, “Submodular salient region detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2043–2050.

[19] Q. Yan, L. Xu, J. Shi, and J. Jia, “Hierarchical saliency detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1155–1162.

[20] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, “Salient object detection: A discriminative regional feature integration approach,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2083–2090.

[21] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, “Deeply supervised salient object detection with short connections,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3203–3212.

[22] F. Yang, X. Li, H. Cheng, Y. Guo, L. Chen, and J. Li, “Multi-scale bidirectional FCN for object skeleton extraction,” in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 1–8.

[23] J. Kuen, Z. Wang, and G. Wang, “Recurrent attentional networks for saliency detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3668–3677.

[24] N. Liu and J. Han, “DHSNet: Deep hierarchical saliency network for salient object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 678–686.

[25] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, “Saliency detection with recurrent fully convolutional networks,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 825–841.

[26] Y. Tang, X. Wu, and W. Bu, “Deeply-supervised recurrent convolutional neural network for saliency detection,” in *Proc. ACM Multimedia Conf.*, 2016, pp. 397–401.

[27] 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.

[28] X. Li, F. Yang, H. Cheng, J. Chen, Y. Guo, and L. Chen, “Multi-scale cascade network for salient object detection,” in *Proc. ACM Multimedia Conf. (MM)*, 2017, pp. 439–447.

[29] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, “A bi-directional message passing model for salient object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1741–1750.

[30] Z. Chen, C. Guo, J. Lai, and X. Xie, “Motion-appearance interactive encoding for object segmentation in unconstrained videos,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1613–1624, Jun. 2020.

[31] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[32] S. Chen, X. Tan, B. Wang, and X. Hu, “Reverse attention for salient object detection,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 236–252.

[33] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, “Deep networks for saliency detection via local estimation and global search,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3183–3192.

[34] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[35] P. Hu, B. Shuai, J. Liu, and G. Wang, “Deep level sets for salient object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2300–2309.

[36] X. Li *et al.*, “DeepSaliency: Multi-task deep neural network model for salient object detection,” *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3919–3930, Aug. 2016.

[37] G. Li and Y. Yu, “Visual saliency based on multiscale deep features,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5455–5463.

[38] G. Lee, Y.-W. Tai, and J. Kim, “Deep saliency with encoded low level distance map and high level features,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 660–668.

[39] Y. Tang and X. Wu, “Saliency detection via combining region-level and pixel-level predictions with CNNs,” in *Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 809–825.

[40] G. Li and Y. Yu, “Deep contrast learning for salient object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 478–487.

[41] A. Manno-Kovacs, “Direction selective contour detection for salient objects,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 2, pp. 375–389, Feb. 2019.

[42] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, “Boundary-aware feature propagation for scene segmentation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6819–6829.

[43] 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.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[45] J.-F. Rivest, P. Soille, and S. Beucher, “Morphological gradients,” *J. Electron. Imag.*, vol. 2, no. 4, pp. 326–337, 1993.

[46] A. N. Evans and X. U. Liu, “A morphological gradient approach to color edge detection,” *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1454–1463, Jun. 2006.

[47] H. Law and J. Deng, “Cornernet: Detecting objects as paired keypoints,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 734–750.

[48] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, “The secrets of salient object segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 280–287.

[49] L. Wang *et al.*, “Learning to detect salient objects with image-level supervision,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 136–145.

[50] V. Movahedi and J. H. Elder, “Design and perceptual validation of performance measures for salient object segmentation,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 49–56.

[51] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[52] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu, “A stagewise refinement model for detecting salient objects in images,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4019–4028.

[53] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, “Learning uncertain convolutional features for accurate saliency detection,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 212–221.

[54] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, “CBAM: Convolutional block attention module,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[55] A. Paszke *et al.*, “Automatic differentiation in PyTorch,” in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1–4.

[56] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, “Frequency-tuned salient region detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1597–1604.

[57] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected CRFs with Gaussian edge potentials,” in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2011, pp. 109–117.

[58] 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. 3907–3916.

[59] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.

[60] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio, “ReNet: A recurrent neural network based alternative to convolutional networks,” 2015, *arXiv:1505.00393*. [Online]. Available: http://arxiv.org/abs/1505.00393

Zixuan Chen received the B.E. degree in software engineering from Sun Yat-sen University, China, in 2016, where he is currently pursuing the master’s degree in computer science. His research interests include computer vision and pattern recognition, with a focus on image saliency, semantic segmentation, object segmentation, and video object segmentation.

Huajun Zhou (Graduate Student Member, IEEE) received the master's degree in computer science from the Nanjing University of Science and Technology, in 2018. He is currently pursuing the Ph.D. degree with Sun Yat-sen University. He has published three papers in several academic conferences and journals. His main research interest includes salient object detection.

Jianhuang Lai (Senior Member, IEEE) received the M.Sc. degree in applied mathematics and the Ph.D. degree in mathematics from Sun Yat-sen University, China, in 1989 and 1999, respectively. In 1989, he joined Sun Yat-sen University as an Assistant Professor, where he is currently a Professor with the School of Data and Computer Science. His current research interests include the areas of computer vision, pattern recognition, and its applications. He has published over 250 scientific papers in the international journals and conferences on image processing and pattern recognition, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B, *Pattern Recognition*, ICCV, CVPR, and ICDM. He serves as the Deputy Director of the Image and Graphics Association of China and the Standing Director of the Image and Graphics Association of Guangdong. He is also the Deputy Director of the Computer Vision Committee, China Computer Federation (CCF).

Lingxiao Yang (Member, IEEE) received the Ph.D. degree from Sun Yat-sen University, China, in March 2020. His research interests include computer vision and machine learning with a focus on object recognition, video understanding, and brain-inspired computational model.

Xiaohua Xie (Member, IEEE) received the B.Sc. degree in mathematics and applied mathematics from Shantou University in 2005 and the M.Sc. degree in information of computing science and the Ph.D. degree in applied mathematics from Sun Yat-sen University, China, in 2007 and 2010, respectively. He was an Associate Professor with the Shenzhen Institutes of Advanced Technology (SIAT), Chinese Academy of Sciences. He is currently an Associate Professor with Sun Yat-sen University. He has published more than a dozen articles in the prestigious international journals and conferences. His current research interests include cover image processing, computer vision, pattern recognition, and machine learning.