RETHINKING TWO-B-REAL NET FOR REAL-TIME SALIENT OBJECT DETECTION

Senyun Kuang[1], *Shijin Meng*[1], *Bo Xiao*[1], *Lv Tang and Bo Li*[2]

[1] School of Information Science and Technology, Southwest Jiaotong University, China

[2] Youtu Lab, Tencent, Shanghai, China.

ABSTRACT

Exploring a fast and accurate salient object detection (SOD) model is a promising research area. TBRS [1] has been proposed a two-branch network for real-time SOD. However, its principle of adding an extra path to encode spatial information is time-consuming. And its backbone is borrowed from image classification tasks, may be inefficient for SOD due to the deficiency of task-specific design. To handle these problems, we propose a novel and efficient structure named short-range concatenate module (SRCM) by removing structure redundancy. Specifically, we gradually reduce the dimension of feature maps and use the aggregation of them for image representation, which forms the basic module of SRCM network. Moreover, we propose an efficient detail guidance branch (DBG) to further encode detail structural information in low-level stages instead of the time-consuming *perceptual branch* used in TBRS. Finally, low-level features and high-level features are fused by the feature projection module (FPM). Extensive evaluations and analysis demonstrate that our proposed algorithm achieves the leading accuracy performance with real-time speed (216*fps*). We hope that our series of works can motivate future research for real-time SOD task.

Index Terms— Real-Time, Saliency Detection, Short-range Concatenate, Detail Guidance, Feature Projection.

1. INTRODUCTION

Salient object detection (SOD) is a classic and fundamental topic in computer vision, which aims to identify the most visually distinctive objects in an image. The prosperity of deep learning greatly promotes the performance of SOD by making various breakthroughs, coming with fast-growing demands in many computer vision tasks, such as weakly supervised semantic segmentation [2], visual video object segmentation [3], reversible data hiding [4] and object detection [5].

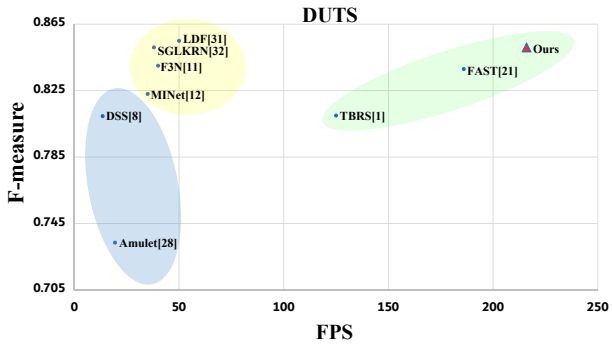

Fig. 1. Inference speed (FPS) and F-measure performance on DUTS [20] dataset. As can be seen that the proposed model has the fastest speed while achieves comparable performance with state-of-the-art methods.

Compared to traditional SOD methods [6, 7], deep-based methods use high-level semantic features to replace handcrafted features, leading to better SOD performance. Deep-based methods mainly refine the SOD results by improving the network structure, such as introducing attention mechanism [8, 9, 10, 11, 12, 13], iterative refining [14, 15], designing edge enhancement modules [16, 17, 18, 19]. As can be seen in Figure.1 (blue and yellow circles), the performance of SOD is improved yearly. However, the main inconvenience with these approaches is their low inference speed, which drags them on wide-ranging applications.

In order to achieve real-time SOD, TBRS [1] uses a lightweight backbone network to fast extract high-level semantic information and a parallel branch with shallow network for structural information as supplementary. FAST [21] proposes a novel cross-level feature aggregation strategy, which can enhance the model learning capacity and increase the receptive field simultaneously. However, the lightweight backbone (Xception [22]) used in these two works are borrowed from image classification tasks, which may not be perfect for SOD due to the deficiency of task-specific design.

In this paper, inspired by the two-branch architecture used in TBRS, we design a novel network for the purpose of faster inference speed. Firstly, we design a novel short-range concatenate module (SRCM) in backbone, to get variant scalable receptive fields with a few parameters. In detail, as shown

This paper is equally contributed by Senyun Kuang and Shijin Meng. Correspondence should be addressed to Bo Xiao. This work is supported by the Fundamental Research Funds for the Central Universities. Email: syKuang@my.swjtu.edu.cn, 869337710@my.swjtu.edu.cn and xiaobo@swjtu.edu.cn.

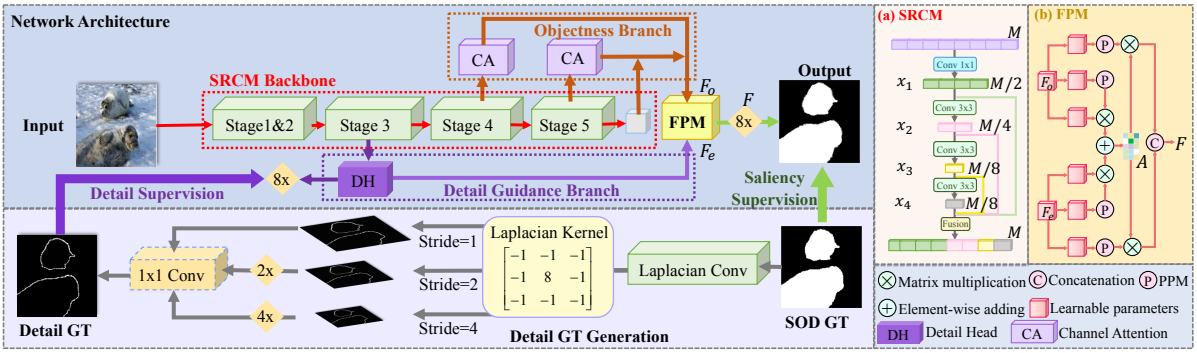

Fig. 2. The overall architecture of the proposed model. M is the resolution of each feature map.

in Figure.2(a), we concatenate response maps from multiple continuous convolutional blocks, each of which encodes input features in different scales and respective fields, leading to multi-scale feature representations which are important for SOD. To speed up, the filter size of convolutional layers is gradually reduced with negligible loss in SOD performance. Our proposed backbone is stacked by several SRCMs. Secondly, TBRS uses an extra time-consuming *perceptual branch* to guide the low-level stages for the learning of spatial details. In this paper, we replace *perceptual branch* by detail guidance branch (DGB), which uses boundary features as a guidance to make the network attend to finer spatial details. Different from FAST, which only uses "Canny" algorithm to generate detail GT. In this paper, we design a learnable generation to get multi-scale detail GT, which can better supervise DBG. Finally, the distribution of the spatial features from low-level stages and semantic features from high-level stages is different, directly fusing these two features would lead to deficiency results. To address this problem, we propose a feature projection module (FPM) to project spatial features and semantic features to the common feature subspace. This operation can bridge the gap between these two features. As shown in Figure.1 (green circle), compared with other methods, our proposed network achieves a better balance between speed and accuracy which can generate impressive SOD results with a far faster speed (216*fps*).

2. METHOD

In this section, we first elaborate on the effectiveness of our proposed short-range concatenate module (SRCM), detail guidance branch (DGB) and feature projection module (FPM). Then the whole architecture will be introduced.

2.1. Short-range Concatenate Module

As can be seen in Figure.2, our proposed backbone contains five stages. The first two stages are two 3×3 convolutional layers with $stride = 2$. Each of the remaining stages contains two short-range concatenate module (SRCM). The spatial resolution of the feature maps is reduced with a stride of 2 during each stage. The structure of SRCM module can be seen in Figure 2(a). Each SRCM is separated into 4 convolutional blocks and we denote the output of i-th block as x_i ($i \in [1, 4]$). We claim that our SRCM has two advantages: (1) we elaborately tune the filter size of blocks by gradually decreasing in geometric progression manner, leading to significant reduction in computation complexity. Because in each SRCM, low-level blocks need enough channels to encode more fine-grained information with small receptive field, while high-level blocks with large receptive field focus more on high-level information induction, setting the same channel with low-level layers may cause information redundancy. (2) the final output of SRCM module is concatenated from all 4 blocks, which preserves scalable respective fields and multi-scale information, and is written as: $x_{output} = \Psi(x_1, x_2, ..., x_n)$, where Ψ is the fusion operation. The SRCM module keeps the size of output feature maps unchanged. Different from lightweight backbone designed for image classification task, our proposed backbone can achieve multi-scale features with limited parameters, which is more suitable for SOD task. Our proposed SRCM can remove structure redundancy by avoiding an extra branch to encode spatial information, and uses the aggregation of multi-scale features within the module for better high-level semantic features generation, which can capture both spatial and context information efficiently.

2.2. Detail Guidance Branch

TBRS uses an extra time-consuming *perceptual branch* to guide the low-level stages for the learning of spatial details. Because our proposed SRCM can help efficiently capture spatial information, we propose a detail guidance branch (DGB) to further encode detail structural information in low-level stages for final SOD prediction. As can be seen in Figure.2, we use a detail head (DH), containing two 3×3 convolutional operations to encode spatial information in stage3. Then, we should generate the binary detail ground-truth from the SOD ground-truth to supervise the DGB. This operation can be carried out by 2D convolution kernel named Laplacian kernel and a trainable 1×1 convolution. We use the Laplacian oper-

ator shown in Figure.2 to produce soft thin detail feature maps with different strides to obtain multi-scale detail information. Then we upsample the detail feature maps to the original size and fuse it with a trainable 1×1 convolution for dynamic re-weighting. Finally, we adopt a threshold 0.1 to convert the predicted details to the final binary detail ground-truth with boundary and corner information. Note that the generation of detail ground-truth (Gray Line) is discarded in the inference phase. Therefore, this side-information can easily boost the accuracy of SOD task without any cost in inference.

2.3. Feature Projection Module

Similar to TBRS, we use an objectness branch (OB) to provide sufficient high-level contextual objectness. The output of OB is F_o, and the output of DGB is F_e. However, the distribution of these two features is different, directly fusing these two features would to deficiency results. To address this problem, we propose a feature projection module (FPM) to project these two features to the common feature subspace. We first upsample the feature F_o to the same size of $F_e \in \mathbb{R}^{C\times H\times W}$. Then, as can be seen in Figure.2(b), we employ pyramid pooling module (PPM) [23] to reduce the dimension of feature maps F_o and F_e to save the computational cost. The PPM is composed of four-scale feature bins, which are then flattened and concatenated to form a matrix of size $C_1 \times N$, $N \ll HW$, $C_1 = C/2$. Thus, the self-affinity matrixes of F_o and F_e can be calculated as:

$$\begin{aligned} A_o &= (PPM(W_o^1 F_o))^\top (W_o^2 F_o), \\ A_e &= (PPM(W_e^1 F_e))^\top (W_e^2 F_e), \end{aligned} \tag{1}$$

where A_o and A_e denote the similarity matrixes. Their sizes are fixed to $N \times (HW)$ through the PPM. W_o^1, W_o^2, W_e^1 and $W_e^2 \in \mathbb{R}^{C_1\times C}$, indicate the learnable parameters. We further combine these two matrices as follows:

$$A = softmax((A_o + A_e)^\top). \tag{2}$$

Then, the row-wise normalized matrix $A \in \mathbb{R}^{(HW)\times N}$ is used to project two features to the same subspace:

$$\begin{aligned} \widetilde{F}_o &= A(PPM(W_o^3 F_o))^\top, \\ \widetilde{F}_e &= A(PPM(W_e^3 F_e))^\top. \end{aligned} \tag{3}$$

We concatenate $\widetilde{F}_o$ and $\widetilde{F}_e$ and use a 3×3 convolution to get the final output feature F.

2.4. Supervision of The Network

Generally, our whole network architecture contains two branches. We first use the SRCM backbone as our encoders. Then we use an OB to capture sufficient high-level contextual objectness, and design DGB to further encode detail structural information in the low-level stage. Finally, these two features are fused by our proposed FPM. Our model is jointly trained by detail supervision and saliency supervision. The detail prediction is supervised by pixel-wise cross entropy loss, which is defined as: $L_e = -\big(\mathcal{G}^e log(M^e) + (1-\mathcal{G}^e) log(1-M^e)\big)$, where M^e denotes detail prediction map and $\mathcal{G}^e$ is the detail ground truth. The saliency prediction is also supervised by pixel-wise cross entropy loss, which is defined as: $L_s = -\big(\mathcal{G}^s log(M^s) + (1-\mathcal{G}^s) log(1-M^s)\big)$, where M^s denotes saliency prediction map and $\mathcal{G}^s$ is the saliency ground truth. The total loss is: $L = L_s + L_e$.

3. EXPERIMENTS

3.1. Experimental Setup

Saliency Detection Datasets: Following most previous methods [16, 11, 12], we train our model on DUTS-TR [20] dataset for fair comparison with existing works. We only use the simple random horizontal flipping for data augmentation. For evaluating the effectiveness of the proposed network, we adopt another three widely-used benchmark datasets: DUTS [20], DUT-OMRON [4], HKU-IS [24].

Implementation Details: We use Pytorch to implement our model. During training, we use mini-batch stochastic gradient descent (SGD) with batch size 32, momentum 0.9 and weight decay 1e-5. The learning rate is set as 1e-4, and maximum epoch is set to 100. The training images are resized to 352×352 as the input to the whole network. For fair comparison, all experiments run on the same 1080Ti GPU.

Evaluation Metrics: Five metrics are used to evaluate the performance of our method. The first is Mean Absolute Error (MAE), which characterizes the average distance between prediction and groundtruth. F-measure (F_β) [25] and Structure Measure(S_m) [26] are also used to evaluate salient maps. Following [27, 21], we use B_μ to evaluate the structure alignment between saliency maps and their ground-truth, which shows the boundary quality of predictions. Smaller MAE and B_μ, larger F_β and S_m correspond to better performance. In addition, PR-curve is used to show the whole performance.

3.2. Performance Comparison

We compare our proposed method with other 12 state-of-the-art ones: Amulet [28], DSS [8], TBRS [1], CPD [29] BASNet [15], EGNet [16], F3N [11], MINet [12], GateNet [30], LDF [31], FAST [21], SGLKRN [32].

Qualitative Evaluation: Figure.3 shows the visual comparison of the proposed method with other algorithms. As shown, our method can not only preserve the completeness of objects, but also restore well edge details, producing overall better prediction results.

Quantitative Evaluation: As can be seen in Figure.4 and Table.1, our method achieves competitive accuracy performance among the state-of-the-art methods across all the datasets. Especially, we get the leading position in terms of

Table 1. quantitative comparison on three dataset. the best three results are shown in red, green and blue color respectively.

Data Set	Metric	Amulet	DSS	TBRS	CPD	BASNet	EGNet	F3N	MINet	GateNet	LDF	FAST	SGLKRN	Ours(Baseline)	Ours(+DBG)	Ours	Ours-1
DUTS	F_β	0.733	0.808	0.810	0.813	0.791	0.800	0.840	0.823	0.783	0.855	0.838	0.851	0.814	0.835	0.851	0.831
	S_m	0.804	0.820	0.850	0.867	0.866	0.866	0.888	0.875	0.870	0.890	0.876	0.893	0.862	0.880	0.892	0.863
	MAE	0.085	0.057	0.055	0.043	0.048	0.044	0.035	0.039	0.045	0.034	0.041	0.034	0.043	0.040	0.036	0.051
	B_μ	0.572	0.492	0.532	0.419	0.378	0.463	0.385	0.409	0.489	0.411	0.383	0.436	0.589	0.445	0.376	0.505
DUT-OMRON	F_β	0.647	0.740	0.674	0.745	0.766	0.744	0.766	0.741	0.723	0.773	0.750	0.783	0.725	0.755	0.770	0.723
	S_m	0.781	0.790	0.812	0.818	0.838	0.813	0.838	0.822	0.821	0.839	0.839	0.846	0.805	0.840	0.847	0.826
	MAE	0.098	0.062	0.093	0.057	0.056	0.057	0.053	0.057	0.061	0.052	0.055	0.049	0.068	0.057	0.053	0.070
	B_μ	0.717	0.640	0.720	0.558	0.490	0.576	0.526	0.571	0.625	0.544	0.488	0.574	0.558	0.490	0.479	0.551
HKU-IS	F_β	0.841	0.902	0.842	0.896	0.895	0.893	0.910	0.904	0.889	0.914	0.902	0.916	0.855	0.890	0.911	0.864
	S_m	0.886	0.878	0.881	0.904	0.909	0.910	0.917	0.912	0.910	0.919	0.920	0.921	0.880	0.915	0.922	0.892
	MAE	0.051	0.040	0.063	0.033	0.032	0.035	0.028	0.031	0.036	0.028	0.030	0.028	0.041	0.035	0.029	0.041
	B_μ	0.547	0.490	0.549	0.418	0.368	0.452	0.393	0.409	0.474	0.408	0.363	0.432	0.476	0.386	0.357	0.451

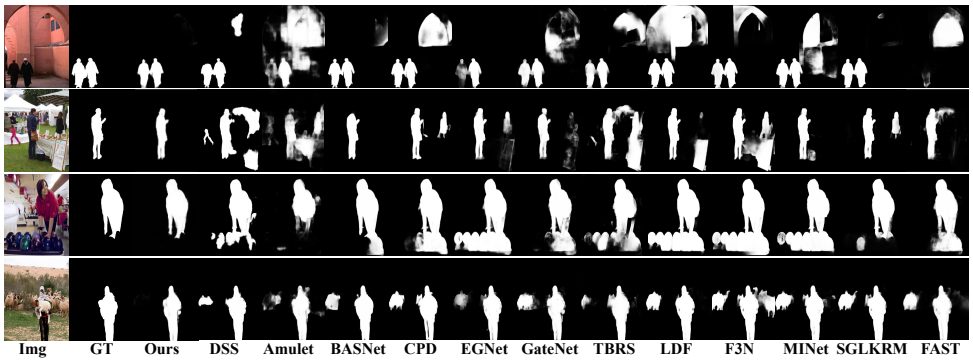

Fig. 3. Comparison examples of the proposed method with the state-of-the-art methods.

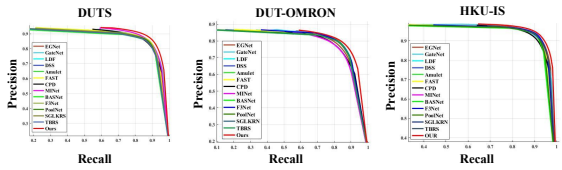

Fig. 4. Performance comparison in terms of PR curves.

Table 2. Average speed (FPS) comparisons.

	Ours	Ours	FAST	TBRS	MINet	F3N	EGNet
Size	400 × 300	352 × 352	400 × 300	400 × 300	320 × 320	352 × 352	400 × 300
FPS	218	216	186	125	35	40	12
	BASNet	DSS	Amulet	CPD	GateNet	LDF	SGLKRN
Size	256 × 256	224 × 224	256 × 256	352 × 352	384 × 384	352 × 352	352 × 352
FPS	25	12	16	50	33	50	38

S_m and B_μ two metrics, which shows our proposed method can predict saliency maps with richer boundary details. Note that this is achieved without any pre-processing and post-processing. The speed comparison is shown in Table.2. As can be seen, with our novel architecture, our method runs 216 *fps* which is far faster than the best existing methods.

3.3. Ablation Analysis

In this section, we analyze the contribution of different model components. Our proposed baseline model (Ours(Baseline)) only contains SRCM backbone and OB. As can be seen in Figure.5 and Table.1, our proposed baseline can only roughly locate the salient objects, and the detection results contain much background noise. When adding DBG (Ours(+DBG)), our model can indeed produce saliency map with more boundary details. After leveraging FPM (Ours), our proposed method can preserve the completeness of objects and suppress the noise regions. In Table.1, we also report the performance that replace our proposed SRCM backbone with Xception [22] (Ours-1). It can be seen that the performance is decreased a lot, which exhibits the superiority of the proposed . Noting that the speed of Ours-1 is 203 *fps*, which is slower than our proposed method. Because our proposed backbone can achieve multi-scale features, which needs fewer convolutional layers to get enough receptive field.

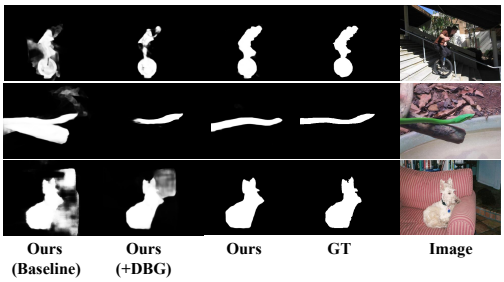

Fig. 5. Qualitative comparisons with different model settings.

4. CONCLUSION

In this paper, we first address the drawbacks of the previous two-branch network TBRS, then propose a novel and efficient structure named short-range concatenate module (SRCM) by removing structure redundancy. Moreover, we propose a detail guidance branch (DBG) to further encode detail structural information in low-level stages. Finally, low-level features and high-level features are fused by the feature projection module (FPM). Extensive evaluations and analysis demonstrate that the proposed algorithm achieves the leading accuracy performance with real-time speed (216*fps*).

5. REFERENCES

[1] Bo Li, Zhengxing Sun, Lv Tang, and Anqi Hu, "Two-b-real net: Two-branch network for real-time salient object detection," in *ICASSP*, 2019.

[2] Lian Xu, Mohammed Bennamoun, Farid Boussaïd, and Senjian An, "An improved approach to weakly supervised semantic segmentation," in *ICASSP*, 2019.

[3] Wenguan Wang and Jianbing Shen, "Saliency-aware video object segmentation," *TPAMI*, 2018.

[4] Shilong Yang, "Saliency-based image contrast enhancement with reversible data hiding," in *ICASSP*, 2020.

[5] Kristian Fischer, Felix Fleckenstein, Christian Herglotz, and André Kaup, "Saliency-driven versatile video coding for neural object detection," in *ICASSP*, 2021.

[6] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu, "Global contrast based salient region detection," *TPAMI*, 2015.

[7] Jingdong Wang, Huaizu Jiang, Zejian Yuan, and Ming-Ming Cheng, "Salient object detection: A discriminative regional feature integratio approach," *IJCV*, 2017.

[8] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, and Zhuowen Tu, "Deeply supervised salient object detection with short connections," *TPAMI*, 2019.

[9] Bo Wang, Quan Chen, Min Zhou, Zhiqiang Zhang, Xiaogang Jin, and Kun Gai, "Progressive feature polishing network for salient object detection," in *AAAI*, 2020.

[10] Zuyao Chen and Qianqian Xu and, "Global context-aware progressive aggregation network for salient object detection," in *AAAI*, 2020.

[11] Jun Wei and Shuhui Wang, "F3net: Fusion, feedback and focus for salient object detection," *AAAI*, 2020.

[12] Youwei Pang and Xiaoqi Zhao, "Multi-scale interactive network for salient object detection," in *CVPR*, 2020.

[13] Lv Tang and Bo Li, "Class: Cross-level attention and supervision for salient objects detection," in *ACCV*, 2020.

[14] Tiantian Wang, Lihe Zhang, Shuo Wang, Huchuan Lu, Gang Yang, Xiang Ruan, and Ali Borji, "Detect globally, refine locally: A novel approach to saliency detection," in *CVPR*, 2018.

[15] Xuebin Qin and Zichen Zhang, "Basnet: Boundary-aware salient object detection," in *CVPR*, 2019.

[16] Jiaxing Zhao, Jiangjiang Liu, Deng-Ping Fan, and Yang Cao, "Egnet: Edge guidance network for salient object detection," in *ICCV*, 2019.

[17] Jinming Su, Jia Li, Yu Zhang, Changqun Xia, and Yonghong Tian, "Selectivity or invariance: Boundary-aware salient object detection," in *ICCV*, 2019.

[18] Huajun Zhou, Xiaohua Xie, Jian-Huang Lai, and Zixuan Chen, "Interactive two-stream decoder for accurate and fast saliency detection," in *CVPR*, 2020.

[19] Jiangjiang Liu, Qibin Hou, and Ming-Ming Cheng and, "A simple pooling-based design for real-time salient object detection," in *CVPR.*, 2019.

[20] Lijun Wang, Huchuan Lu, Yifan Wang, and Mengyang Feng, "Learning to detect salient objects with image-level supervision," in *CVPR*, 2017.

[21] Lv Tang, Bo Li, Yanliang Wu, Bo Xiao, and Shouhong Ding, "Fast: Feature aggregation for detecting salient object in real-time," in *ICASSP*, 2021.

[22] François Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR.*, 2017.

[23] Zhen Zhu, Mengdu Xu, Song Bai, Tengteng Huang, and Xiang Bai, "Asymmetric non-local neural networks for semantic segmentation," in *ICCV*, 2019.

[24] Guanbin Li and Yizhou Yu, "Visual saliency based on multiscale deep features," in *CVPR*, 2015.

[25] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li, "Salient object detection: A benchmark," *TIP*, 2015.

[26] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji, "Structure-measure: A new way to evaluate foreground maps," in *ICCV*, 2017.

[27] Jing Zhang, Xin Yu, Aixuan Li, Peipei Song, Bowen Liu, and Yuchao Dai, "Weakly-supervised salient object detection via scribble annotations," in *CVPR*, 2020.

[28] Pingping Zhang, Dong Wang, and Huchuan Lu, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *ICCV*, 2017.

[29] Zhe Wu and Li Su, "Cascaded partial decoder for fast and accurate salient object detection," in *CVPR*, 2019.

[30] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang, "Suppress and balance: A simple gated network for salient object detection," *CoRR*, 2020.

[31] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian, "Label decoupling framework for salient object detection," in *CVPR*, 2020.

[32] Binwei Xu, Haoran Liang, Ronghua Liang, and Peng Chen, "Locate globally, segment locally: A progressive architecture with knowledge review network for salient object detection," in *AAAI*, 2021.