Edge-Aware Convolution Neural Network Based Salient Object Detection

Wenlong Guan, Tiantian Wang, Jinqing Qi, Lihe Zhang, and Huchuan Lu

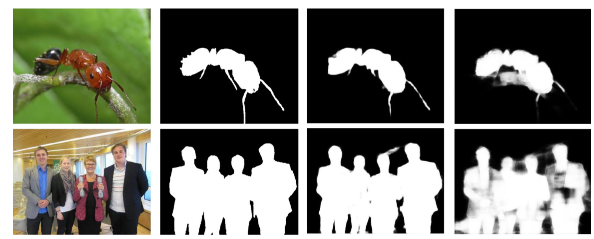
***Abstract*—Salient object detection has received great amount of attention in recent years. In this letter, we propose a novel salient object detection algorithm, which combines the global contextual information along with the low-level edge features. First, we train an edge detection stream based on the state-of-the-art holistically-nested edge detection (HED) model and extract hierarchical boundary information from each VGG block. Then, the edge contours are served as the complementary edge-aware information and integrated with the saliency detection stream to depict continuous boundary for salient objects. Finally, we combine pyramid pooling modules with auxiliary side output supervision to form the multi-scale pyramid-based supervision module, providing multi-scale global contextual information for the saliency detection network. Compared with the previous methods, the proposed network contains more explicit edge-aware features and exploit the multi-scale global information more effectively. Experiments demonstrate the effectiveness of the proposed method, which achieves the state-of-the-art performance on five popular benchmarks.**

***Index Terms*—Saliency detection, edge detection, pyramid pooling network, convolutional neural networks (CNNs).**

I. Introduction

VISUAL saliency, which aims to identify the most conspicuous objects or areas in an image, has intrigued great interest in recent years. It has been shown effective owing to various applications in many visual tasks, such as scene classification [1], image captioning [2] and visual tracking [3], [4]. Although visual saliency has achieved state-of-the-art performance through the use of deep convolutional neural networks (CNNs), it still suffers from two major problems in denoising the final prediction against the cluttered background and preserving the boundaries of salient objects.

The recent CNNs-based approaches [5]–[8] have made great progress towards tackling the above issues. It is known that the shallower layers of CNNs are encoded with fine but cluttered structure information, while the deeper layers contain much stronger semantic information at the expense of spatial resolution. Thus the existing methods mainly focus on the feature integration strategies through channel-wise fusion [6]–[8], with the purpose of capturing and integrating high-level objectness along with low-level structured information.



(a) Input image (b) Ground truth (c) Ours (d) Amulet [7]

Fig. 1. Visual comparison of different methods.

However, the low-level features from existing saliency networks cannot effectively help to preserve object boundaries because the supervision of saliency maps lacks specific attention on the boundary area. In other words, edge supervision has not been utilized explicitly by the previous works.

Besides, even though channel-wise fusion integrates features from different scales of CNNs, the network may still fail to distinguish the salient object from the background or to localize the object precisely because of the lack of global structure information, as described in [5], [9].

Therefore, from the perspective of edge preserving, we propose a novel edge-aware fusing module (EFM) to help learn the finer boundary information through two steps: first, we train a VGG-based Detection Stream(EDS) motivated by the holistically-nested edge detection (HED) [9] model on BSDS dataset [10] and fix the parameters when its training loss converges. Second, for each input image, we extract edge maps from the trained EDS and fuse them with the features generated by the side output of DenseBlocks [11] to help train the Saliency Detection Stream(SDS). To summarize, the EFM aims to provide the saliency detection task with the extra edge-related information.

Then, from the global perspective, we adopt a multi-scale pyramid-based supervision module (PSM) based on pyramid pooling module (PPM). PPM [12] implements a series of pooling operations in order to gather the global structural information with different scales, while the multiple side-output supervision can generate the saliency prediction hierarchically. Combining these two parts together, PSM can effectively improve the final prediction.

As shown in Fig. 1, compared with Amulet [7], the proposed method can generate more detailed object boundaries and effectively suppress background noise thanks to the use of EFM and PSM.

Our main contributions are summarized as follows:

- We propose a novel EFM to augment the saliency detection network with explicitly supervised edge-aware features.

Manuscript received August 10, 2018; revised November 6, 2018; accepted November 8, 2018. Date of publication November 22, 2018; date of current version December 3, 2018. This work was supported by the Natural Science Foundation of China under Grants 61725202, 61751212, 61771088, 61632006, and 91538201. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Daniel P. K. Lun. *(Corresponding author: Huchuan Lu.)*

W. Guan, T. Wang, and H. Lu are with the Dalian University of Technology, Dalian 116024, China (e-mail: guanwenlong@mail.dlut.edu.cn; wangtiantian@mail.dlut.edu.cn; lhchuan@dlut.edu.cn).

J. Qi and L. Zhang are with the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: jinqing@dlut.edu.cn; zhanglihe@dlut.edu.cn).

Digital Object Identifier 10.1109/LSP.2018.2881835

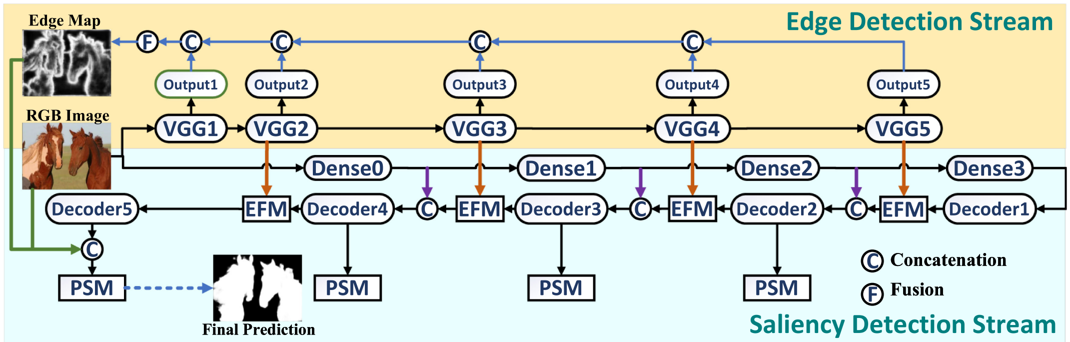


Fig. 2. The overall architecture of the proposed model. The whole network contains two stream: Edge detection stream (EDS) and saliency detection stream (SDS). Both streams take original RGB image as input.

- We adopt a multi-scale PSM to exploit multi-scale global context information via varying-scale feature representation, and make saliency prediction with each auxiliary side output feature, which effectively boost the robustness of the proposed network.
- Our proposed model (trained on the DUTS dataset) achieves the state-of-the-art performance on five widely used large-scale salient object detection datasets, including DUTS, ECSSD, HKU-IS, PASCAL-S and THUR15K.

II. The Proposed Method

The motivation for the proposed network is two-fold. First, previous methods fail to deblur the object edges when applying object-level saliency supervision. Therefore we seek for a solution by providing the saliency detection network with direct edge-aware supervision.

Second, existing saliency detection approaches only use the side outputs through direct channel-wise operation (addition or concatenation) and ignore the importance of global structure information. Hence we combine the multi-scale pyramid pooling module with auxiliary supervision to fill the gap.

The structure of the proposed method is illustrated in Fig. 2, which consists of two streams. The EDS is based on the VGG-16 network, and the SDS is based on the DenseNet which composed of two components: EFM and PSM.

A. Densely Connected Encoder-Decoder Network

We choose the recently proposed Densely Connected Net (DenseNet121) as our encoder-decoder baseline model due to its superior performance in classification task and memory-efficient implementation. Compared to the widely used VGG16 or ResNet-50, the training process based on DenseNet can converge faster with the dense skip connections.

Specifically, we adopt the first convolutional layer, pooling layer (DenseBlock0) and the first three DenseBlocks (DenseBlock1, DenseBlock2, DenseBlock3) extracted from a pretrained DenseNet121 as our encoder structure and the resolution of the output feature maps is 1/32 of the input image. As is illustrated in Fig. 3, Each decoder block is designed as a residual transition block [13], where the 1×1 convolution is responsible for increasing and reducing the dimension, and the final layer aims to double the resolution of feature maps through bilinear interpolation.

As shown in Fig. 2, the input of each decoder block is the concatenation of two parts with the same spatial resolution, the first one is the feature map from each corresponding DenseBlock, the second one comes from EFM, which will be explained in Section II-B.

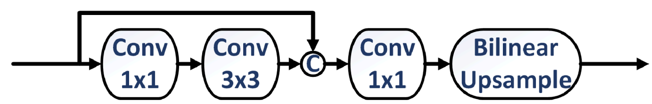


Fig. 3. The architecture details of decoder block, for simplicity the batch normalization and Relu layer after each convolutional layer are omitted.

B. Edge-Aware Fusing Module

The above-mentioned encoder-decoder feedforward network can produce coarse saliency prediction but cannot well preserve the edge structure because the upsampling operations fail to recover spatial information and finer details. To mitigate this, we propose a novel EFM that infuses edge-aware feature maps into the saliency detection task. Edge detection task aims to detect edges and object boundaries in natural images, which is a fundamental low-level computer vision task and is of great importance for tasks such as segmentation [14], [15] and object detection [16], [17]. Here we leverage complementary edge-related information to assist saliency prediction task.

We first train the EDS on the edge detection benchmark BSDS independently. EDS is based on the HED model [9], which combines multiple side outputs together through concatenation and use a fusion layer to get the unified output, in which the fusion layer is a 1×1 convolution with kernel size 5. The structure of EDS can be found in the yellow box of Fig. 2.

Then for each input image, we feed forward it into both EDS and SDS. The model parameters of EDS are fixed and multi-level edge-aware feature maps are fused into the SDS to help train the saliency stream. To be specific, we take feature maps at four levels from EDS including conv2-2 (128 channel), conv3-3 (256 channel), conv4-3 (512 channel) and conv5-3 (512 channel). Then fuse them respectively with the output of Decoder4, Decoder3, Decoder2 and Decoder1 blocks through concatenation operation. The fusion mechanism of EFM can be represented as follows,

$$F_i = Cat(X_i, E_j), \tag{1}$$

where F_i denotes the output feature map of EFM, X_i and E_j represent the intermediate features from the corresponding blocks of SDS and EDS respectively, in which $((i, j) \in$

$\{(5,1),(4,2),(3,3),(2,4)\}$). Cat is the channel-wise concatenation.

In addition, for Decoder5, which has the same spatial resolution as the input image, we additionally fuse the final edge detection result (1 channel) with the original RGB image (3 channel) to form the strongest complementary details.

C. Multi-Scale Pyramid-Based Supervision Module

Even though the encoder-decoder network mentioned in Section II-A already combines feature maps from different levels through concatenation, the network lacks the motivation to capture the multi-scale objects in the input images due to the shortage of global context structures.

In order to tackle this problem, we adopt the pyramid pooling module [12], which has been proved effective in tasks such as classification [18] and segmentation [12], [19]. This module first implements a four-level average pooling operation and then the pooled features are upsampled to the original input size. The last step is to concatenate the multi-level pooled features with the original features.

Based on the basic pyramid pooling operation, we propose a novel PSM to make better use of the multi-scale global context structure. Rather than applying one PPM only at the final prediction part, we combine the auxiliary side outputs with PPM in order to achieve a multi-scale supervision. The reason lies in the necessity of multi-scale supervision for dense prediction tasks [6], [8], [9]. To this end, we append PPM behind each EFM to further boost the learning process,

$$S_i = \sigma(W_i * Cat(W_{i,1} * PP_1(X_i), ..., W_{i,n} * PP_n(X_i))), \tag{2}$$

where $X_i (i \in \{2,3,4,5\})$ denotes output of the i-th Decoder block, $PP_n()$ denotes pyramid pooling operation at level $n (n \in \{1,2,3,4\})$, $W_{i,n}$ denotes the weight of 1×1 convolution, σ is the sigmoid function to rescale the prediction map and $S_i()$ denotes the final i-th side output prediction.

At the training phase, each auxiliary prediction S_i will be supervised by the ground truth saliency mask through pixel-wise cross entropy loss separately. During the inference time, we only pick the output of S_5, which has the same resolution as the input image.

III. Experiments

A. Experimental Setup

Saliency Detection Datasets: For fine-tuning our model, we utilize the DUTS [20] training dataset.

For evaluating the effectiveness of the proposed network, we adopt another four widely-used benchmark datasets: ECCSD [21], THUR15K [22], HKU-IS [23], PASCAL-S [24].

Implementation Details: The proposed method is implemented with the PyTorch deep learning framework [25]. During training, we use ADAM optimization algorithm and set the batch size to 1 for both streams. For EDS we set the learning rate to 10^{-6} and the training process takes 200 k iterations; for SDS the parameters are set to 10^{-6} and 300 k respectively. The training samples are resized to 384×384 as the input to the whole network. We use GTX 1080Ti GPU for training and inference. It takes us about 8 hours to train the edge detection network and another 7 hours for the saliency detection network. During the inference, our model can run at 25 fps.

B. Evaluation Metrics

We use three main metrics to evaluate the performance of saliency detection algorithms, including the precision-recall (PR) curves, F-measure and Mean Absolute Error (MAE) [26]. We first compute pairs of precision and recall values by thresholding the predicted saliency map for the use of plotting PR curves. Besides, F-measure value is used as another metric, which is computed on every binarized saliency map and is averaged on the whole dataset. The threshold is determined to be twice the mean saliency value. The F-measure value represents an overall performance of the corresponding algorithm and calculated as $F_\beta = \frac{(1+\beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precison + Recall}$, where β^2 is typically set to 0.3 to emphasize on the precision more than the recall [27].

In addition to the PR curve and F-measure, we also calculate the Mean Absolute Error (MAE) which is defined as the average per-pixel difference between the estimated saliency maps S and the ground truth G, $MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - G(x,y)|$, where W and H are width and height of the saliency map S, respectively.

C. Performance Comparison

We compare our method with 5 state-of-the-art methods, including SRM [5], BMP [28], DSS [8], MSR [29], DCL [30]. For fair comparison, we trained our model on three datasets to compare with the corresponding methods and utilize either the released codes or the saliency maps provided by the authors.

Quantitative Evaluation: PR curves are showed in Fig. 4, in terms of all three datasets (DUTS, HKU-IS, PASCAL-S), our method is among the top contenders.

We also list the comparison with respect to the F-measure and MAE scores. Since our method do not have a post-process, so the non-CRF version of DCL(DCL^-), DSS(DSS^-), MSR(MSR^-) are used for comparison. As shown in Table I. Our method outperforms the existing methods in most cases. *Qualitative Evaluation:* Fig. 5 shows the visual comparison of the proposed method with other algorithms. As illustrated in the picture, our method successfully preserves the boundary and suppress background noise simultaneously, producing overall better prediction maps.

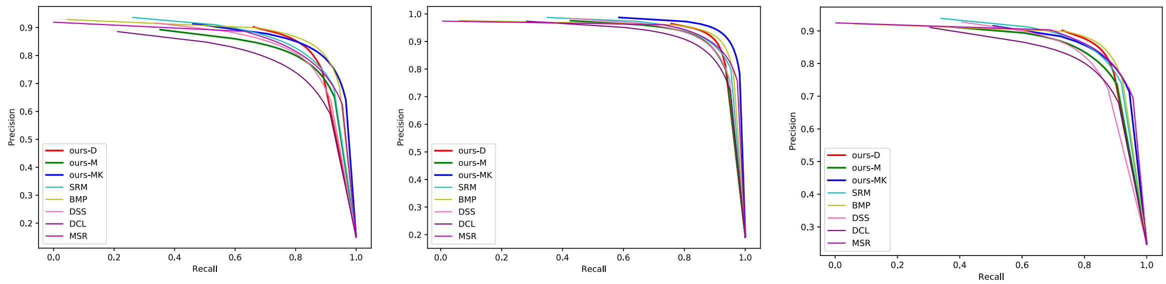
D. Ablation Analysis

In this section, we analyze the contribution of different model components. We first implement the encoder-decoder structure based on DenseNet121 as our baseline model, then apply PSM in the decoder part, namely PSM^+. On the basis of PSM^+, Edge-aware Fusing Module is additionally integrated to generate our final network EFM^+.

The F-measure and MAE scores are shown in Table II across all five datasets, here we use the results trained on DUTS dataset. It can be seen that both modules can improve the performance by a relatively large margin.

To demonstrate that the improvement of EFM^+ comes from the edge-aware ability instead of ensemble of the VGG and DenseNet, we train the EDS and SDS together with the saliency detection dataset, namely $ENSEM^+$. The results show that $ENSEM^+$ has a slight improvement compared with PSM^+, but is not comparable with EFM^+.

Since our EFM module mainly focus on refining the boundary area, so we also made a comparison with the widespread post-processing method DenseCRF [8] by replacing EFM^+ with



DUTS dataset HKU-IS dataset PASCAL-S dataset

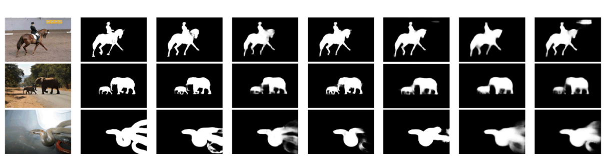
Fig. 4. Performance comparison in terms of PR curves.

TABLE I
QUANTITATIVE COMPARISON OF F-MEASURE AND MAE SCORES. THE BEST TWO RESULTS ARE SHOWN IN RED AND BLUE COLOR, RESPECTIVELY

	Training Set	DUTS-TEST		PASCAL-S		ECSSD		THUR		HKU-IS	
		F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE
Ours-D	DUTS	0.785	0.047	0.812	0.075	0.901	0.044	0.717	0.074	0.879	0.036
SRM	DUTS	0.757	0.059	0.80	0.085	0.892	0.056	0.708	0.077	0.874	0.046
BMP	DUTS	0.751	0.049	0.769	0.074	0.868	0.045	0.69	0.079	0.871	0.039
Ours-M	MB	0.721	0.082	0.778	0.106	0.888	0.058	0.698	0.081	0.861	0.050
DSS^-	MB	0.723	0.069	0.773	0.103	0.873	0.061	0.682	0.083	0.855	0.051
DCL^-	MB	0.714	0.149	0.714	0.125	0.827	0.151	0.676	0.161	0.853	0.136
Ours-MK	MB+HKU	0.737	0.065	0.782	0.092	0.891	0.045	0.695	0.079	0.881	0.032
MSR^-	MB+HKU	0.723	0.066	0.779	0.083	0.861	0.056	0.674	0.089	0.857	0.043

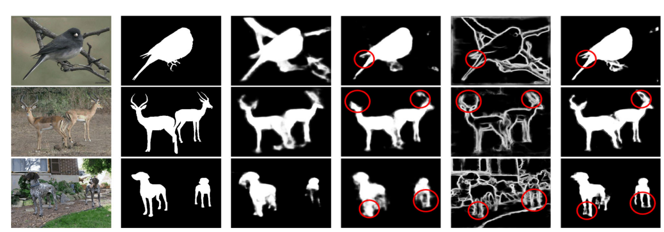
TABLE II
ABLATION ANALYSIS OF F-MEASURE AND MAE SCORES. THE BEST TWO RESULTS ARE SHOWN IN RED AND BLUE COLOR, RESPECTIVELY

	DUTS-TEST		PASCAL-S		ECSSD		THUR		HKU-IS	
	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE	F-measure	MAE
baseline	0.731	0.062	0.773	0.094	0.859	0.068	0.682	0.082	0.833	0.054
PSM^+	0.766	0.053	0.787	0.081	0.877	0.051	0.693	0.078	0.842	0.044
$ENSEM^+$	0.770	0.051	0.790	0.084	0.883	0.047	0.692	0.076	0.857	0.041
EFM^+	0.785	0.047	0.812	0.075	0.901	0.044	0.717	0.074	0.879	0.036
CRF^+	0.794	0.05	0.814	0.079	0.911	0.042	0.738	0.072	0.890	0.035



RGB GT Ours SRM BMP DSS^- MSR^- DCL^-

Fig. 5. Comparison examples of the proposed method with the state-of-the-art methods.



(a) (b) (c) (d) (e) (f)

Fig. 6. Illustration of ablation analysis. (a) Input images. (b) Ground truth. (c) Baseline. (d) PSM^+. (e) Edge map from EDS. (f) EFM^+.

CRF^+. The result shows that our EFM^+ can still achieve comparable results in MAE. But since CRF method is a post-processing method, it is much less time-efficient, can only run at 3fps during inference. Besides, its performance highly rely on the original saliency map, which makes it less robust than the learning-based method.

Fig. 6 shows the qualitative results. We find that the PSM helps highlight the salient objects and improve the spatial coherence of detected salient area, while the EFM produces sharp boundaries and help distinguish the partly overlapped objects. As illustrated in the figure, the edge map is able to emphasize the salient object boundary in most cases. Although it may also bring some noise into SDS, the supervised SDS can learn to leverage the emphasized boundary information while discard the unintended noise as the training processes.

IV. CONCLUSIONS

In this paper, we propose a novel edge-aware end-to-end saliency detection method based on the extra boundary information and multi-scale pyramid pooling layers. The complementary edge-related features are able to effectively preserve sharp boundaries of salient objects, while the combination of auxiliary side outputs with pyramid pooling layers can extract rich global context information.

Extensive quantitative and qualitative evaluations over five widely-used datasets verify that the proposed contributions can effectively improve state-of-the-art saliency detection performance.

REFERENCES

[1] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 300–312, Feb. 2007.

[2] H. Fang *et al.*, "From captions to visual concepts and back," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 2015, pp. 1473–1482.

[3] A. Borji, S. Frintrop, D. N. Sihite, and L. Itti, "Adaptive object tracking by learning background context," in *Proc. Comput. Vision Pattern Recogn. Workshops*, 2012, pp. 23–30.

[4] V. Mahadevan and N. Vasconcelos, "Biologically inspired object tracking using center-surround saliency mechanisms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 541–554, Mar. 2013.

[5] 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. Vision*, 2017, pp. 4039–4048.

[6] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *Proc. Comput. Vision Pattern Recogn.*, 2016, pp. 678–686.

[7] P. Zhang, D. Wang, H. Lu, H. Wang, and R. Xiang, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 202–211.

[8] Q. Hou, M. M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, pp. 5300–5309, 2016.

[9] S. Xie and Z. Tu, "Holistically-nested edge detection," *Int. J. Comput. Vision*, vol. 125, no. 1-3, pp. 3–18, 2015.

[10] P. Arbelez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[11] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 2017, pp. 2261–2269.

[12] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 2017, pp. 6230–6239.

[13] S. Jgou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proc. Comput. Vision Pattern Recogn. Workshops*, 2017, pp. 1175–1183.

[14] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 2009, pp. 2294–2301.

[15] J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proc. Comput. Vision Pattern Recogn.*, 2014, pp. 328–335.

[16] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of adjacent contour segments for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 36–51, Jan. 2008.

[17] S. Ullman and R. Basri, "Recognition by linear combinations of models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 10, pp. 992–1006, Oct. 1991.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–16, Sep. 2015.

[19] H. Zhang *et al.*, "Context encoding for semantic segmentation," in *Proc. Comput. Vision Pattern Recogn.*, pp. 7151–7160, 2018.

[20] L. Wang *et al.*, "Learning to detect salient objects with image-level supervision," in *Proc. Comput. Vision Pattern Recogn.*, 2017, pp. 3796–3805.

[21] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. Comput. Vision Pattern Recogn.*, 2013, pp. 1155–1162.

[22] M. M. Cheng, N. J. Mitra, X. Huang, and S. M. Hu, "Salientshape: Group saliency in image collections," *Visual Comput. Int. J. Comput. Graph.*, vol. 30, no. 4, pp. 443–453, 2014.

[23] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. Comput. Vision Pattern Recogn.*, 2015, pp. 5455–5463.

[24] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proc. Comput. Vision Pattern Recogn.*, 2014, pp. 280–287.

[25] A. Paszke *et al.*, "Automatic differentiation in pyTorch," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017.

[26] A. Borji, D. N. Sihite, and L. Itti, *Salient Object Detection: A Benchmark*. Berlin, Germany: Springer, 2012.

[27] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 2009, pp. 1597–1604.

[28] 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. Vision Pattern Recogn.*, 2018, pp. 1741–1750.

[29] G. Li, Y. Xie, L. Lin, and Y. Yu, "Instance-level salient object segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, pp. 247–256, 2017.

[30] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 2016, pp. 478–487.