TWO-B-REAL NET: TWO-BRANCH NETWORK FOR REAL-TIME SALIENT OBJECT DETECTION

Bo Li, Zhengxing Sun✉, Lv Tang, Anqi Hu

State Key Laboratory for Novel Software Technology, Nanjing University, 210046, China

ABSTRACT

As a hot topic in computer vision, recent researches on salient object detection (SOD) have focused on using the over-designed deep convolutional neural networks (CNNs) to improve the detection accuracy. However, these complex architectures constraint themselves to low speed and drag them on wide-ranging applications. In this paper, we simplify the over-designed networks and propose the Two-Branch Network for Real-time Salient Object Detection (Two-B-Real Net). Particularly, the Perceptual Branch and the Objectness Branch in our network can efficiently capture detailed information and distinctive objectness simultaneously. And we also design novel attention mechanisms to guide the network to focus on most saliency-related features and generate more accurate results. Extensive evaluations show that the proposed algorithm achieves the leading accuracy performance with real-time speed (*125fps*) which is significantly faster than the existing methods.

Index Terms— Two-Branch network, salient object detection, real-time, attention mechanism

1. INTRODUCTION

As a fundamental but challenging problem in computer vision, salient object detection is derived with the goal of discovering and locating most conspicuous objects or regions in an image which attract human attention. It endows many computer vision systems with the capability to take advantage of human attention for more promising processing and analysis, such as semantic segmentation [1], visual tracking [2], video summarization [3] and person re-identification [4], etc.

Generally, good salient object detection methods should be fast, accurate, and able to identify and localize a wide variety of objects. Recently, by introducing deep convolutional neural networks (CNNs), more algorithms focus on improving the detection accuracy of SOD. Such as Liu et al. [5] propose a fully convolutional network (FCN) based end-to-end method. A hierarchical recurrent CNN is adopted to progressively recover image details of saliency maps through integrating local context information. Zhang et. al. [6] learn to aggregate multi-level feature maps at each resolution and predict saliency maps in a recursive manner. Hou et al. in [7] densely introduce short connections by transforming high-level features to shallower side-output layers. The multi-scale feature maps at each layer can assist to locate salient regions and recover detailed structures. While these methods have achieved excellent salient object detection accuracy, the complex designs like recurrent structure and dense skip connection are very redundancy and computationally inefficient, which constraint these methods to low speed and drag them on wide-ranging applications. This is mainly caused by the over-designed CNN architectures they use [8].

To balance the speed and accuracy, we simplify the over-designed networks and achieve real-time SOD with a novel architecture. Following the principle pointed out in most previous works, a good salient object detection network should make full use of multi-level features to capture distinctive objectness and detailed information simultaneously, and we propose the Two-Branch Network for Real-time Salient Object Detection (Two-B-Real Net) with two parts: *Perceptual Branch* (PB) and *Objectness Branch* (OB). As their names imply, *Perceptual Branch* is designed to capture the detailed visual perception information such as color, texture and spatial structure which can localize the most attractive regions for human vision. We stack only three convolution layers to obtain affluent perception details as low-level information. While, for *Objectness Branch*, we use light backbone network Xception [9] to quickly shrink the receptive field and obtain contextual objectness from deep layers efficiently as high-level information. In pursuit of more accurate SOD results without loss of speed, we design the Spatial Attention (SA) Module and Channel Attention (CA) Module for *Perceptual Branch* and *Objectness Branch* respectively. We also research the fusion of two branches and introduce Attention based Feature Fusion (AFF) Module to better aggregate multi-level feature. The proposed attention mechanisms can guide network to focus on the salient objects and generates most saliency-related features. As our following experiments show, the proposed novel architecture achieves impressive results on three benchmarks with real-time speed (*125fps*).

*This work was supported by National High Technology Research and Development Program of China (No.2007AA01Z334), National Natural Science Foundation of China (Nos.61321491 and 61272219).
Email: njumagiclibo@gmail.com, szx@nju.edu.cn(corresponding author), tanglv@smail.nju.edu.cn , huaq@smail.nju.edu.cn.

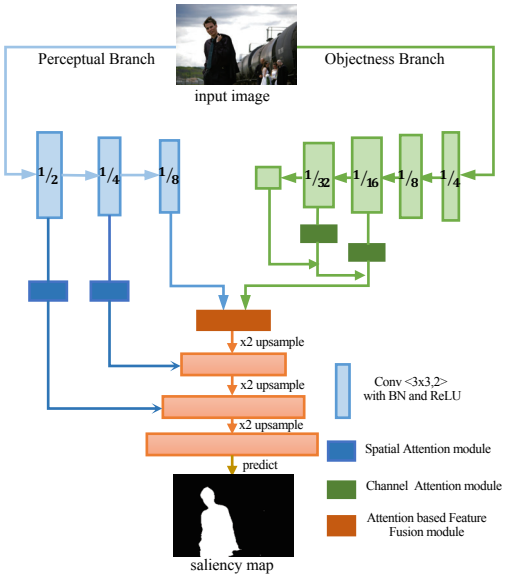


Fig. 1. Architecture of proposed network.

2. OUR METHOD

In this section, we first illustrate our proposed Two-Branch Network for Real-time Salient Object Detection (Two-B-Real Net) in detail. Furthermore, we elaborate on the effectiveness of Perceptual Branch and Objectness Branch with their Spatial Attention (SA) Module and Channel Attention(CA) Module correspondingly. Finally, the whole architecture of our Two-B-Real Net and Attention based Feature Fusion Module (AFF) will be introduced.

2.1. Perceptual Branch with Spatial Attention

In the SOD task, most state-of-the-art methods [6, 7, 8] directly use the shallower side-output layers of pre-trained deep CNN or complex convolutional modules to capture low-level features. For real-time SOD task, these modules inevitably need more computation and run-time. The popular approaches to accelerate this process are resizing input image to a small size to reduce the computation complexity or lightening the network by channel pruning. However, those approaches damage the detailed information especially spatial structure. To preserve affluent detailed visual perception information with high speed, we propose the Perceptual Branch which contains only three convolution layers. Each layer includes a convolution with $stride = 2$, followed by batch normalization [10] and ReLU. Therefore, this branch extracts the output feature maps that is 1/8 of the original image. It encodes rich detailed information due to the large spatial size of feature maps. Figure 1 presents the details of the structure.

*Spatial Attention Module:*In general, salient objects only correspond to partial regions of the input image. And there exist some background regions which can distract human attention. Therefore, directly exploiting convolutional features to predict saliency can lead to sub-optimal results because of the distraction of non-salient regions. Instead of considering all spatial positions equally, spatial attention is able to focus more on the saliency-related regions, which helps to generate effective features for SOD task. Inspired by SENet [8], we propose a Spatial Attention (SA) Module to refine the low-level features. As Figure 2(a) shows, SA module first employs a convolutional layer with 1×1 kernels, followed by batch normalization and ReLU. Then attention weight of feature map at each pixel is obtained by applying Softmax operation. Finally, we concatenate the spatial attention map with the low-level feature maps instead of directly multiplying them. Because, multiplying the attention maps with the low-level feature maps causes fake edges, which may lead to wrong saliency predictions.

2.2. Objectness Branch with Channel Attention

While the Perceptual Branch encodes affluent detailed visual perception information, the Objectness Branch is designed to provide sufficient high-level contextual objectness. Directly using the deep side-output features of very deep pre-trained CNN like VGG-19 [11] or ResNet-101/152 [12] is computation demanding and memory consuming. Considering efficient computation and sufficient high-level features with large receptive field simultaneously, we propose to use a lightweight model as the backbone of Objectness Branch. The lightweight model, like Xception [9], can downsample the feature map fast to obtain a large receptive field, which encodes high-level semantic objectness information. Then, we combine the up-sampled features of the last three stages as the final output of Objectness Branch.

*Channel Attention Module:*As different channels of feature in CNNs generate response to different semantics, it is unwise to treat all channels without distinction. To alleviate the interference of the irrelevant semantic information, we introduce a Channel Attention (CA) Module to assign larger weights to channels which show a higher response to salient objects.As Figure 2(b) shows, CA Module employs global average pooling to capture global context and computes an attention vector to guide the feature learning. It integrates the global contextual objectness easily at each stage. Therefore, it demands negligible computation cost. Considering there are fewer channels in low-level features and all of them can be useful, we don't apply Channel Attention on Perceptual Branch. And since most spatial information has already been lost in high-level features, it will be redundant to use Spatial Attention on Objectness Branch.

2.3. Network architecture

As illustrated in Figure 1, We use the pre-trained Xception model as the backbone of the Objectness Branch and three convolution layers as the Perceptual Branch. Then we fuse the output features of these two branches and followed by upsample layers with skip connection from low-level features to make the final prediction. Our Two-B-Real Net can achieve

real-time performance and high accuracy simultaneously. First, we simplify the over-designed networks, so both two branches are not computation intensive. Furthermore, these two branches compute concurrently, which considerably increases efficiency. Second, the attention mechanisms can guide the network to focus on most saliency-related features for more accurate results with negligible computation cost.

Attention based Feature Fusion: As the two branches encode different level information, we can not simply sum up these features. Instead, we use an Attention based Feature Fusion module to aggregate multi-level features. As shown in Figure 2(c), we first concatenate the output features of two branches. And then we utilize the batch normalization to balance the scales of the features. Next, we pool the concatenated feature to a feature vector and compute an attention weight vector. This weight vector can re-weight the features, which amounts to feature selection and combination.

Loss Function: Given the SOD training dataset S with N training pairs $S = \{(X_n, Y_n)\}_{n=1}^{N}$, where $X_n = \{x_i^n, i = 1, \ldots, T\}$ and $Y_n = \{y_i^n, i = 1, \ldots, T\}$ are the input image and the binary ground-truth image with T pixels, respectively. $y_i^n = 1$ denotes the foreground pixel and $y_i^n = 0$ denotes the background pixel. In most of existing SOD methods, the loss function used to train the network is the standard pixel-wise binary cross-entropy (BCE) loss. However, for a typical natural image, the class distribution of salient/non-salient pixels is heavily imbalanced: most of the pixels in the ground truth are non-salient. To automatically balance the loss between positive/negative classes, we introduce a class-balancing weight β on a per-pixel term basis, following [13]. Specifically, we define the following weighted cross-entropy loss function,

$$\begin{aligned} \mathcal{L}_{wbce} = &-\beta \sum_{i \in Y_+} logPr(y_i = 1|X;\theta) \\ &-(1-\beta) \sum_{i \in Y_-} logPr(y_i = 0|X;\theta). \end{aligned} \tag{1}$$

where θ is the parameter of the network. $Pr(y_i = 1|X;) \in [0, 1]$ is the confidence score of the network prediction that measures how likely the pixel belong to the foreground. The loss weight $\beta = |Y_+|/|Y|$, and $|Y_+|$ and $|Y_-|$ denote the foreground and background pixel number, respectively.

3. EXPERIMENTAL EVALUATION

In our experiments, we use three evaluation metrics. (i) The precision-recall (PR) curves, which exhibits the mean precision and recall of saliency maps at different thresholds. (ii) The F-measure is a weighted mean of average precision and recall, calculated by $F_\eta = \frac{(1+\eta^2)\times Precision \times Recall}{\eta^2 \times Precision + Recall}$. We set η^2 to be 0.3 as suggested in [14]. (iii) The mean absolute error (MAE), $MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - G(x,y)|$,where W and H are the width and height of the input image. $S(x, y)$ and $G(x, y)$ are the pixel values of the saliency map and the binary ground truth at (x, y), respectively. For the performance evaluation, we adopt three

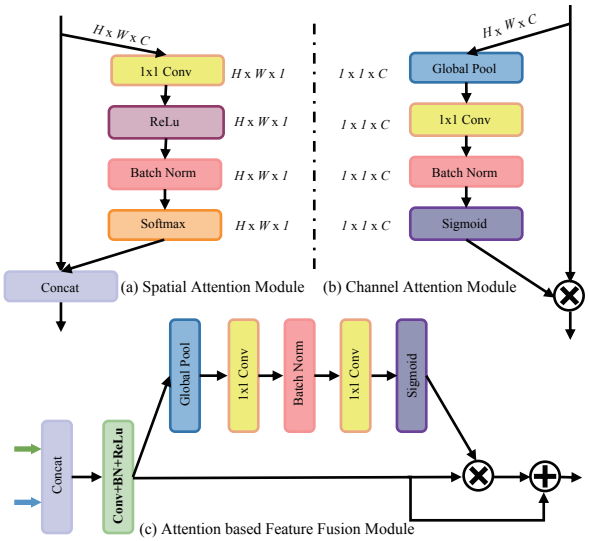


Fig. 2. Illustration of attention modules, where $\oplus$ means element addition and $\otimes$ means element multiplication.

well-known SOD datasets, including ECSSD [15], DUT-OMRON [16] and HKU-TE [17]. And we compare our proposed method with other 9 state-of-the-art ones, including 7 deep learning based algorithms: Amulet [6], UCF [18], DCL [19], DHS [5], RFCN [20], ELD [21], MDF [17] and 2 conventional algorithms: MST [22], DSR [23]. We use either the implementations with recommended parameter settings or the saliency maps provided by the authors.

To train our model, we adopt the MSRA10K [14] dataset, which has 10,000 training images with pixel-wise saliency annotations. And we augment this dataset by random cropping and mirror reflection, producing 120,000 training images totally. We implement our proposed model based on TensorFlow framework [24]. We train and test our method in a PC machine with an NVIDIA GTX 1080 GPU and an i7-6900 CPU. During the training, we use standard SGD method with batch size 12, weight decay 0.005. We set the base learning rate to 1e-4 and decrease the learning rate by 1% when training loss reaches a flat. The training process converges after 150k iterations.

As shown in PR curves and Table 1, for quantitative comparison, our method has already achieved competitive accuracy performance among the state-of-the-art methods. We further replace the backbone net with ResNet18 [12] a more powerful pre-trained CNN as Ours-2. With ResNet18, our method consistently outperforms existing methods across all the datasets in terms of almost all evaluation metrics. And we also examine the effectiveness of the proposed techniques by using two baselines: baseline 1 – without using the attention mechanisms and baseline 2 – without using skip connection. The results in Table 1 shows the attention mechanisms and skip connection can further improve our accuracy performance. For qualitative comparison, Figure 3 shows some sample saliency maps from three datasets for reference. Our

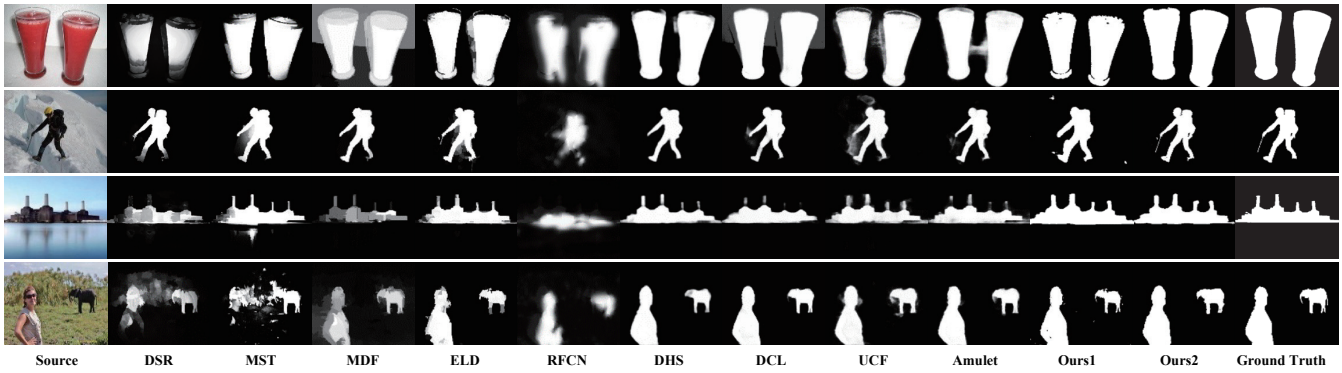
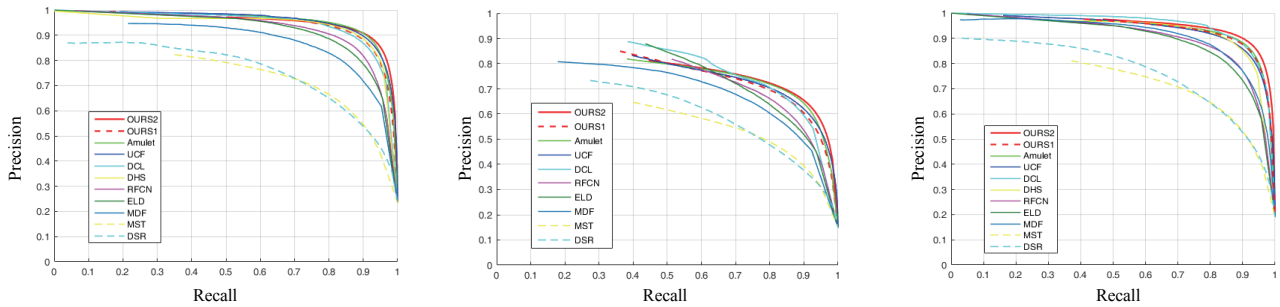


Fig. 3. Comparison of typical saliency maps.



(a)**ECSSD** (b)**DUT-OMRON** (c)**HKU-TE**

Fig. 4. The PR curves of the proposed algorithm and other state-of-the-art methods.

Table 1. Quantitative comparison on 3 famous datasets. The best three results are shown in red, green and blue, respectively.

Data Set	Metric	DSR	MST	MDF	ELD	RFCN	DHS	DCL	UCF	Amulet	BLine1	BLine2	Ours-1	Ours-2
ECSSD	F-m	0.662	0.724	0.807	0.810	0.834	0.872	0.829	0.844	0.868	0.825	0.831	0.848	0.887
	MAE	0.178	0.155	0.105	0.080	0.107	0.060	0.149	0.069	0.059	0.097	0.082	0.067	0.054
DUT-OMRON	F-m	0.524	0.588	0.644	0.611	0.627	-	0.684	0.621	0.647	0.642	0.655	0.674	0.695
	MAE	0.139	0.161	0.092	0.092	0.111	-	0.157	0.120	0.098	0.107	0.113	0.093	0.084
HKU-TE	F-m	0.682	0.707	0.802	0.776	0.838	0.854	0.853	0.823	0.843	0.822	0.834	0.842	0.871
	MAE	0.142	0.139	0.095	0.072	0.088	0.053	0.136	0.061	0.050	0.091	0.079	0.063	0.051

Table 2. Speed Comparison. Image size is 400×300.

Method	DSR	MST	MDF	ELD	RFCN	DHS	DCL	UCF	Amulet	Ours-1	Ours-2
fps	0.662	5	0.125	2	3	23	2	23	16	**125**	**68**
code	Matlab	C	Matlab	C	Matlab	Matlab	Matlab	Matlab	Matlab	Python	Python

model is able to detect salient objects in the scene with complex or highly textured background. The speed comparison is shown in Table 2. As can be seen, with the light backbone network Xception, our method runs 125 *fps* and is 5 times faster than the best existing methods. Even using ResNet18 as the backbone network, our method runs 68 *fps*, which is still significantly faster than the other methods. And the real-time speed will foster more applications.

4. CONCLUSION

In this paper, we present a novel Two-Branch Network with well-designed attention mechanisms to improve the speed and accuracy of real-time salient object detection simultaneously. Our proposed Two-B-Real Net contains two branches: Perceptual Branch (PB) and Objectness Branch (OB). The Perceptual Branch is designed to capture the detailed visual perception information from original images. And the Objectness Branch utilizes the lightweight model to quickly shrink the receptive field and obtain contextual objectness from deep layers efficiently. We also design special attention mechanisms to guide the network to focus on most saliency-related features for more accurate results. Extensive experiments demonstrate that our method performs favorably against state-of-the-art saliency approaches in both accuracy and speed.

5. REFERENCES

[1] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan, "STC: A simple to complex framework for weakly-supervised semantic segmentation," *TAPMI*, vol. 39, no. 11, pp. 2314–2320, 2017.

[2] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *ICML*, 2015, pp. 597–606.

[3] Ioannis Mademlis, Anastasios Tefas, and Ioannis Pitas, "Summarization of human activity videos using a salient dictionary," in *ICIP*, 2017, pp. 625–629.

[4] Rui Zhao, Wanli Ouyang, and Xiaogang Wang, "Unsupervised salience learning for person re-identification," in *CVPR*, 2013, pp. 3586–3593.

[5] Nian Liu and Junwei Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *CVPR*, 2016, pp. 678–686.

[6] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *ICCV*, 2017, pp. 202–211.

[7] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip H. S. Torr, "Deeply supervised salient object detection with short connections," in *CVPR*, 2017, pp. 5300–5309.

[8] Pingping Zhang, Luyao Wang, Dong Wang, Huchuan Lu, and Chunhua Shen, "Agile amulet: Real-time salient object detection with contextual attention," *CoRR*, vol. abs/1802.06960, 2018.

[9] François Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR*, 2017, pp. 1800–1807.

[10] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[11] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[13] Saining Xie and Zhuowen Tu, "Holistically-nested edge detection," *IJCV*, vol. 125, no. 1-3, pp. 3–18, 2017.

[14] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li, "Salient object detection: A benchmark," *TIP*, vol. 24, no. 12, pp. 5706–5722, 2015.

[15] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia, "Hierarchical image saliency detection on extended CSSD," *TPAMI*, vol. 38, no. 4, pp. 717–729, 2016.

[16] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang, "Saliency detection via graph-based manifold ranking," in *CVPR*, 2013, pp. 3166–3173.

[17] Guanbin Li and Yizhou Yu, "Visual saliency based on multiscale deep features," in *CVPR*, 2015, pp. 5455–5463.

[18] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin, "Learning uncertain convolutional features for accurate saliency detection," in *ICCV*, 2017, pp. 212–221.

[19] Guanbin Li and Yizhou Yu, "Deep contrast learning for salient object detection," in *CVPR*, 2016, pp. 478–487.

[20] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan, "Saliency detection with recurrent fully convolutional networks," in *ECCV*, 2016, pp. 825–841.

[21] Gayoung Lee, Yu-Wing Tai, and Junmo Kim, "Deep saliency with encoded low level distance map and high level features," in *CVPR*, 2016, pp. 660–668.

[22] Wei-Chih Tu, Shengfeng He, Qingxiong Yang, and Shao-Yi Chien, "Real-time salient object detection with a minimum spanning tree," in *CVPR*, 2016, pp. 2334–2342.

[23] Huchuan Lu, Xiaohui Li, Lihe Zhang, Xiang Ruan, and Ming-Hsuan Yang, "Dense and sparse reconstruction error based saliency descriptor," *TIP*, vol. 25, no. 4, pp. 1592–1603, 2016.

[24] Martín Abadi, Paul Barham, and Jianmin Chen et. al., "Tensorflow: A system for large-scale machine learning," in *OSDI*, 2016, pp. 265–283.