Deeply Supervised Salient Object Detection with Short Connections

Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, Philip H. S. Torr

Abstract—Recent progress on salient object detection is substantial, benefiting mostly from the explosive development of Convolutional Neural Networks (CNNs). Semantic segmentation and salient object detection algorithms developed lately have been mostly based on Fully Convolutional Neural Networks (FCNs). There is still a large room for improvement over the generic FCN models that do not explicitly deal with the scale-space problem. Holistically-Nested Edge Detector (HED) provides a skip-layer structure with deep supervision for edge and boundary detection, but the performance gain of HED on saliency detection is not obvious. In this paper, we propose a new salient object detection method by introducing short connections to the skip-layer structures within the HED architecture. Our framework takes full advantage of multi-level and multi-scale features extracted from FCNs, providing more advanced representations at each layer, a property that is critically needed to perform segment detection. Our method produces state-of-the-art results on 5 widely tested salient object detection benchmarks, with advantages in terms of efficiency (0.08 seconds per image), effectiveness, and simplicity over the existing algorithms. Beyond that, we conduct an exhaustive analysis on the role of training data on performance. Our experimental results provide a more reasonable and powerful training set for future research and fair comparisons.

Index Terms—Salient object detection, short connection, deeply supervised network, semantic segmentation, edge detection.

1 Introduction

The goal in salient object detection is to identify the most visually distinctive objects or regions in an image and then segment them out from the background. Different from other segmentation-like tasks, such as semantic segmentation, salient object detection pays more attention to very few objects that are interesting and attractive. Such a useful property allows salient object detection to commonly serve as the first step to a variety of computer vision applications including image and video compression [2], [3], image segmentation [4], content-aware image editing [5], [6], object recognition [7], weakly supervsied segmantic segmentation [8]–[11] visual tracking [12], non-photo-realist rendering [13], [14], photo synthesis [15], [16], information discovery [17], [18], image retrieval [19], [20], action recognition [21] *etc.*

Earlier salient object detection methods were mainly inspired by cognitive studies of visual attention [22] where contrast plays the most important role in saliency detection. Taking this fact into consideration, various hand-crafted features have been designed, employing either global or local cues (See [23], [24] for reviews). However, as these hand-crafted features are based on the prior knowledge of existing datasets, they cannot be extended to be successfully useful in all cases. Although some works have attempted to develop different schemes to combine these features rather than utilizing individual ones, the resulting saliency maps are still far away from being satisfactory, specially when encountering complex and cluttered scenes. To overcome the drawbacks caused by human priors, learning based methods (*e.g.* [25]) appear to better integrate different types of features to improve the generalization ability. Nevertheless, because many fusion details are designed manually, the enriched feature representations still suffer from low contrast and fail to detect salient objects in cluttered scenes.

In a variety of computer vision tasks, such as image classification [27], [28], semantic segmentation [29], edge detection [26], [30], object detection [31], [32], and pedestrian detection [33], convolutional neural networks (CNNs) [34] have successfully broken the limits of traditional hand-crafted features. The emergence of fully convolutional neural networks (FCNs) [29] have further boosted the development of these research areas, providing a more principled learning method. Such an end-to-end learning tool also motivates recent research efforts of using FCNs for salient object detection [35], [36]. Benefiting from the enormous amount of parameters in FCNs, a large margin of performance gain has been made compared to previous approaches. The holistically-nested edge detector (HED) [26] model, which explicitly deals with the scale space problem, has led to large improvements over generic FCN models in the context of edge detection. Though the mechanism of fusing the multi-level features extracted from different scales provides a much more natural way to edge detection, it is incompetent to do segmentation related tasks. Edge detection is an easier task since it does not rely too much on high-level semantic feature representations. This explains why skip-layer structure with deep supervision in the HED model does not lead to obvious performance gain for saliency detection.

- *Q. Hou, M.M. Cheng, and X. Hu are with CCCE, Nankai University. M.M. Cheng is the corresponding author (cmm@nankai.edu.cn).*
- *A. Borji is with the Center for Research in Computer Vision, University of Central Florida (aborji@crcv.ucf.edu)*
- *Z. Tuo is with the University of California at San Diego.*
- *P.H.S. Torr is with the University of Oxford.*
- *A preliminary version of this work appeared at CVPR [1]. The source code are publicly available via our project page: http://mmcheng.net/dss/.*

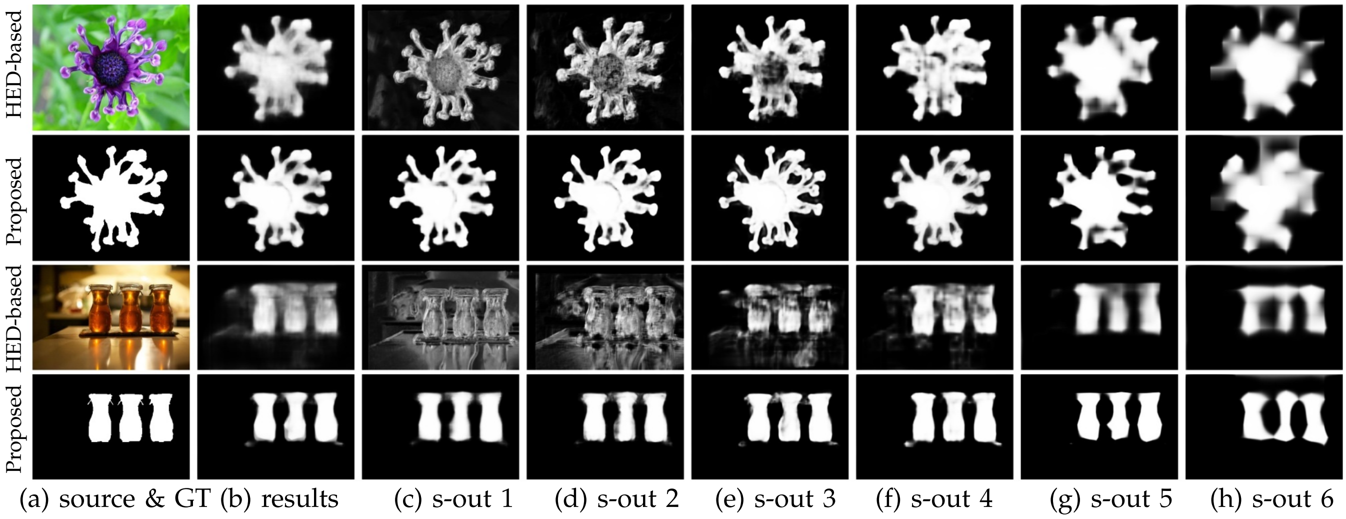


Fig. 1. Visual comparison of saliency maps produced by the HED-based method [26] and ours. Though saliency maps produced by deeper (4-6) side output (s-out) look similar, because of the introduced short connections, each shallower (1-3) side output can generate satisfactory saliency maps and hence a better output result.

Experimental results also support this statement as shown in Fig. 1.

In this paper, we focus on skip-layer structure with deep supervision. Instead of simply fusing the multi-level features extracted from different scales, we consider such a problem in a top-down view. As demonstrated in Fig. 1, we observe that 1) deeper side outputs encode high-level semantic knowledge and hence can better locate where the salient objects are. However, due to the down-sampling operations in FCNs, the predicted maps are normally with irregular shapes especially when the input image is complex and cluttered (see the bottle image), and 2) shallower side outputs capture rich spatial information. They are capable of successfully highlighting the boundaries of those salient objects in spite of the resulting messy prediction maps. Based on these phenomenons, an intuitive idea for yielding better saliency maps is to reasonably combine these multi-level features. This motivates us to develop a new method for salient object detection by introducing *short connections* to the skip-layer structure within the HED [26] architecture. By having a series of short connections from deeper side outputs to the shallower ones, our new framework offers two advantages:

1) high-level features can be transformed to shallower side-output layers and thus can help them better locate the most salient region, and
2) shallower side-output layers can learn rich low-level features that can help refine the sparse and irregular prediction maps from deeper side-output layers.

By combining features from different levels, the resulting architecture provides rich multi-scale feature maps at each layer, a property that is essentially needed to do efficient salient object detection. Our approach is fully convolutional and no other prior information such as superpixels is needed. It takes only 0.08s to produce a prediction map with resolution of 300×400 pixels. Other than improving the state-of-the-art results, we conduct exhaustive analysis on the behavior of different training sets as there is no universal training set for a fair comparison in the salient object detection field. Our goal is to offer a more unified training set and meanwhile build a fair benchmarking environment for future research.

2 Related Works

Over the past two decades, an extremely rich set of saliency detection methods have been developed. The majority of salient object detection methods are based on hand-crafted local features [38]–[40], global features [41]–[43], or both [25], [44]. A complete survey of these methods is beyond the scope of this paper and we refer the readers to recent survey papers [23], [45] for details. Here, we mainly focus on discussing recent salient object detection methods based on deep learning architectures.

2.1 CNN-Based Saliency Models

Compared with traditional methods that use hand-crafted features, CNN-based methods have refreshed all the previous state-of-the-art records in nearly every sub-field of computer vision, including salient object detection. In [46], He *et al.* presented a superpixel-wise convolutional neural network architectures by utilizing hierarchical contrast features. For each scale of superpixels, two contrast sequences were fed into convolutional networks for building more advanced features. Finally, different weights were learned to fuse the multi-scale saliency maps together, yielding a much more confident one. Li *et al.* [47] proposed to use multi-scale features extracted from a deep CNN to derive a saliency map. By feeding different levels of image segmentation into the deep CNN and aggregating multiple resulting features, a stack of fully connected layers are then used to determine on whether each segmented region is salient. Wang *et al.* [48] predicted saliency maps by integrating both local estimation and global search. A deep neural network is first used to learn

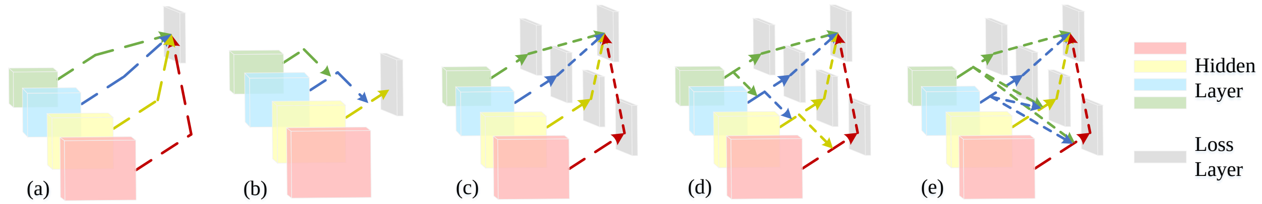


Fig. 2. Illustration of different architectures. (a) Hypercolumn [37], (b) FCN-8s [29] (c) HED [26], (d) and (e) different patterns of our proposed architecture. As can be seen, a series of short connections are introduced in our architecture for combining the advantages of both deeper layers and shallower layers. More interestingly, the last one can be viewed as a generalized version of all the formers.

local patch features to provide each pixel a saliency value. Then, the local saliency map, global contrast, and geometric information are merged together as the input to another deep neural network, which is used to predict the saliency score of each region. In [49], Zhao *et al.* presented a multi-context deep learning framework for salient object detection. Two different CNNs are designed to independently capture the global and local context information of each segment patch. A final regressor is used for final saliency decision of each segment patch. Lee *et al.* [50] took into account both high-level semantic features extracted from CNNs and hand-crafted features. To combine them together, a unified fully connected neural network was exploited to estimate saliency of each query region. Liu *et al.* [36] designed a two-stage deep network, in which a coarse prediction map was produced, followed by a recurrent CNN to refine the details of the prediction map hierarchically and progressively. In [35], a deep contrast network was proposed by leveraging the contrast information of the input images. It combined a pixel-level fully convolutional stream and a segment-wise spatial pooling stream. A fully connected conditional random field (CRF) is also used for further refining the prediction maps from the contrast network. In [51], Wang *et al.* proposed to leverage the advantages of recurrent fully convolutional networks. By doing so, their recurrent fully convolutional network allowed them to continuously refine previous prediction maps by correcting prediction errors. A pre-training strategy using semantic segmentation data is exploited for extracting generic representations of salient objects.

2.2 Skip-Layer Structures

Very recently, great progress has been made in segment detection because of CNNs and their flexible architectures. Of these versatile structures, skip-layer structures have been widely accepted by most researchers owning to their capability of fusing multi-level and multi-scale features. Early-stage skip-layer structures such as Hypercolumn [37] and DCL [35] have made breakthroughs in their respective fields. They, however, only simply fuse the skip layers with different scales for more advanced feature representation building as shown in Fig. 2(a). Differently, FCN-like structures [29] (see Fig. 2(b)) considered a better way to utilize multi-level features, gradually fusing the features from upper layers to lower ones. In [26], Xie and Tu proposed a scheme with deep supervision for each side output (skip layer). Other than fusing all skip layers together, a series of side losses are added after each side output for preserving more details of the edge information. Fig. 2(c) shows a simplified version of these architecture.

Despite the fact that multi-level and multi-scale features have been taken into account and significant progress has been made by these developments very recently, there is still a large room for improvement over the generic CNN models that do not explicitly deal with the scale-space problem.

3 Deep Supervision with Short Connections

This section describes our approach and some implementation details. Before that, let us first take a look at the observations.

3.1 Observations

As pointed out in most previous works, a good salient object detection network should be deep enough such that multi-level features can be learned. Further, it should have multiple stages with different strides so as to learn more inherent features from different scales. A good candidate for such requirements might be the HED network [26], in which a series of side-output layers are added after the last convolutional layer of each stage in the VGGNet [28]. However, experimental results show that this architecture is not suitable for salient object detection. Fig. 1 provides such an illustration. The reasons for this phenomenon are two-fold. On the one hand, saliency detection, requiring homogeneous regions, is quite different from edge detection that demands a special treatment. A good saliency detection algorithm should be capable of extracting the most visually distinctive objects and regions from an image instead of simple edge information. On the other hand, the features generated from lower stages are too convoluted and the saliency maps obtained from the deeper side-output layers are short of regularity.

To overcome the aforementioned problems, we propose a top-down method to reasonably combine both low-level and high-level features for accurate saliency detection. The following subsections are dedicated to a detailed description of the proposed approach.

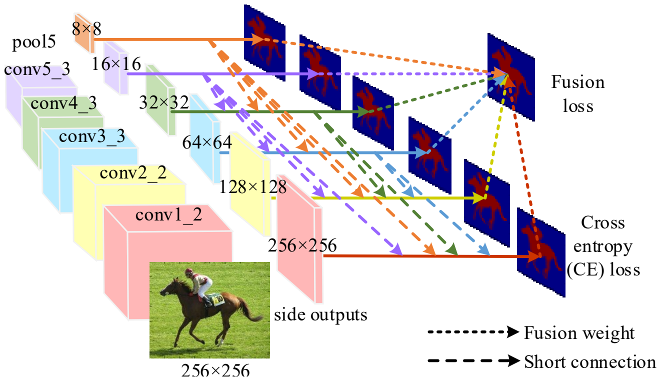


Fig. 3. The proposed network architecture. Our architecture is based on VGGNet [28] for better comparison with previous CNN-based methods. As there are totally 6 different scales in VGGNet, 6 side outputs are introduced, each of which is represented by different colors. Besides the side loss for each side output, a fusion loss is employed for capturing features of different levels.

3.2 HED-based saliency detection

To better understand our proposed approach, we start out with the standard HED architecture [26] as well as its extended version, a special case of this work, for salient object detection and gradually move on to our proposed architecture.

3.2.1 HED architecture

In the HED architecture [26], 5 side outputs are introduced, each of which is directly connected to the last convolutional layer of each stage. Let $T = \{(X_n, Z_n), n = 1, \dots, N\}$ denote the training data set, where $X_n = \{x_j^{(n)}, j = 1, \dots, |X_n|\}$ is the input image and $Z_n = \{z_j^{(n)}, j = 1, \dots, |X_n|\}, z_j^{(n)} \in [0, 1]$ denotes the corresponding continuous ground truth saliency map for X_n. In the sequel, we omit the subscript n for notational convenience since we assume the inputs are all independent of one another. We denote the collection of all standard network layer parameters as $\mathbf{W}$. Without loss of generality, we further suppose that there are totally M side outputs. Each side output is associated with a classifier, in which the corresponding weights can be represented by $\mathbf{w} = (\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \dots, \mathbf{w}^{(M)})$. Thus, the side objective function of HED can be given by

$$L_{\text{side}}(\mathbf{W}, \mathbf{w}) = \sum_{m=1}^{M} \alpha_m l_{\text{side}}^{(m)}\big(\mathbf{W}, \mathbf{w}^{(m)}\big), \tag{1}$$

where α_m is the weight of the mth side loss and $l_{\text{side}}^{(m)}$ denotes the image-level class-balanced cross-entropy loss function [26] for the mth side output. Besides, a weighted-fusion layer is added to better capture the advantage of each side output. The fusion loss at the fusion layer can be expressed as

$$L_{\text{fuse}}(\mathbf{W}, \mathbf{w}, \mathbf{f}) = \sigma\big(Z, h(\sum_{m=1}^{M} f_m A_{\text{side}}^{(m)})\big), \tag{2}$$

No.	Layer	1	2	3
1	conv1_2	$128, 3 \times 3$	$128, 3 \times 3$	$1, 1 \times 1$
2	conv2_2	$128, 3 \times 3$	$128, 3 \times 3$	$1, 1 \times 1$
3	conv3_3	$256, 5 \times 5$	$256, 5 \times 5$	$1, 1 \times 1$
4	conv4_3	$256, 5 \times 5$	$256, 5 \times 5$	$1, 1 \times 1$
5	conv5_3	$512, 5 \times 5$	$512, 5 \times 5$	$1, 1 \times 1$
6	pool5	$512, 7 \times 7$	$512, 7 \times 7$	$1, 1 \times 1$

Fig. 4. Details of each side output. $(n, k \times k)$ means that the number of channels and the kernel size are n and k, respectively. "Layer" means which layer the corresponding side output is connected to. "1" "2" and "3" represent three convolutional layers that are used in each side output. Note that the first two convolutional layers in each side output are followed by a ReLU layer for nonlinear transformation.

where $\mathbf{f} = (f_1, \dots, f_M)$ is the fusion weights, $A_{\text{side}}^{(m)}$ are activations of the mth side output, $h(\cdot)$ denotes the sigmoid function, and $\sigma(\cdot, \cdot)$ denotes the distance between the ground truth map and the fused predictions, which is set to be image-level class-balanced cross-entropy loss [26]. Therefore, the final loss function is given by

$$L_{\text{final}}\big(\mathbf{W}, \mathbf{w}, \mathbf{f}\big) = L_{\text{fuse}}\big(\mathbf{W}, \mathbf{w}, \mathbf{f}\big) + L_{\text{side}}\big(\mathbf{W}, \mathbf{w}\big). \tag{3}$$

HED connects each side output to the last convolutional layer in each stage of the VGGNet [28], respectively conv1_2, conv2_2, conv3_3, conv4_3, conv5_3. Each side output is composed of a one-channel convolutional layer with the kernel size 1×1 followed by an up-sampling layer for learning edge information.

3.2.2 Enhanced HED architecture

In this part, we extend the HED architecture for salient object detection. During our experiments, we observe that deeper layers can better locate the most salient regions, so based on the architecture of HED we connect another side output to the last pooling layer (pool5) in VGGNet [28]. Besides, since salient object detection is a more difficult task than edge detection, we add two another convolutional layers with different filter channels and spatial sizes in each side output, which can be found in Fig. 4. We use the same bilinear interpolation operation as in HED for up-sampling. We also use a standard cross-entropy loss and compute the loss function over all pixels in a training image $X = \{x_j, j = 1, \dots, |X|\}$ and saliency map $Z = \{z_j, j = 1, \dots, |Z|\}$. Our loss function can be defined as follows:

$$\begin{aligned} \hat{l}_{\text{side}}^{(m)}\big(\mathbf{W}, \hat{\mathbf{w}}^{(m)}\big) = & - \sum_{z_j \in Z} z_j \log \Pr\big(z_j = 1 | X; \mathbf{W}, \hat{\mathbf{w}}^{(m)}\big) \\ & + \ (1 - z_j) \log \Pr\big(z_j = 0 | X; \mathbf{W}, \hat{\mathbf{w}}^{(m)}\big), \end{aligned} \tag{4}$$

where $\Pr\big(z_j = 1 | X; \mathbf{W}, \hat{\mathbf{w}}^{(m)}\big)$ represents the probability of the activation value at location j in the mth side output, which can be computed by $h(a_j^{(m)})$, where $\hat{A}_{\text{side}}^{(m)} = \{a_j^{(m)}, j = 1, \dots, |X|\}$ are activations of the mth side output. Similar to [26], we add a weighted-fusion

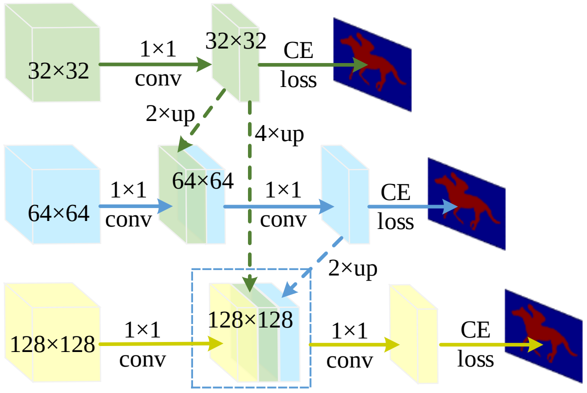


Fig. 5. Illustration of short connections in Fig. 3.

layer to connect each side activation. The loss function at the fusion layer in our case can be represented by

$$\hat{L}_{\text{fuse}}(\mathbf{W}, \hat{\mathbf{w}}, \mathbf{f}) = \hat{\sigma}\big(Z, \textstyle\sum_{m=1}^{\hat{M}} f_m \hat{A}_{\text{side}}^{(m)}\big), \tag{5}$$

where $\hat{A}_{\text{side}}^{(m)}$ is the new activations of the mth side output[1], $\hat{M} = M + 1$, and $\hat{\sigma}(\cdot, \cdot)$ represents the distance between the ground truth map and the new fused predictions, which has the same form as in Eqn. (4).

A comparison of salient object detection results between the original HED and enhanced HED is shown in Fig. 7. It can be easily found that a large margin of about 3% improvement has been achieved. In spite of such improvement, as shown in Fig. 1, the saliency maps from shallower side outputs still look messy and the deeper side outputs produce irregular results as well. In addition, the deeper side outputs can indeed locate the salient objects, but some detailed information is still lost.

3.3 Short connections

The insight of our approach is that deeper side outputs are capable of finding the location of salient regions but at the expense of the loss of details, while shallower ones focus on low-level features but are short of global information. These phenomenons inspire us to utilize the following way to appropriately combine different side outputs such that the most visually distinctive objects can be extracted.

3.3.1 Formulation

Mathematically, our new side activations $\tilde{R}_{\text{side}}^{(m)}$ at the mth side output can be given by

$$\tilde{R}_{\text{side}}^{(m)} = \begin{cases} \sum_{i=m+1}^{\hat{M}} r_i^m \tilde{R}_{\text{side}}^{(i)} + \hat{A}_{\text{side}}^{(m)}, & \text{for } m = 1, \dots, 5 \\ \hat{A}_{\text{side}}^{(m)}, & \text{for } m = 6 \end{cases} \tag{6}$$

where r_i^m is the weight of short connection from side output i to side output m ($i > m$). We can drop out some short connections by directly setting r_i^m to 0. The new side loss function and fusion loss function can be respectively represented by

$$\tilde{L}_{\text{side}}(\mathbf{W}, \tilde{\mathbf{w}}, \mathbf{r}) = \sum_{m=1}^{\hat{M}} \alpha_m \tilde{l}_{\text{side}}^{(m)}(\mathbf{W}, \tilde{\mathbf{w}}^{(m)}, \mathbf{r}) \tag{7}$$

and

$$\tilde{L}_{\text{fuse}}(\mathbf{W}, \tilde{\mathbf{w}}, \mathbf{f}, \mathbf{r}) = \hat{\sigma}\big(Z, \textstyle\sum_{m=1}^{M} f_m \tilde{R}_{\text{side}}^{(m)}\big), \tag{8}$$

where $\mathbf{r} = \{r_i^m\}, i > m$. Note that this time $\tilde{l}_{\text{side}}^{(m)}$ represents the standard cross-entropy loss which we have defined in Eqn. (4). Thus, our new final loss function can be written as

$$\tilde{L}_{\text{final}}(\mathbf{W}, \tilde{\mathbf{w}}, \mathbf{f}, \mathbf{r}) = \tilde{L}_{\text{fuse}}(\mathbf{W}, \tilde{\mathbf{w}}, \mathbf{f}, \mathbf{r}) + \tilde{L}_{\text{side}}(\mathbf{W}, \tilde{\mathbf{w}}, \mathbf{r}). \tag{9}$$

3.3.2 Construction

The backbone of our new architecture is the enhanced HED which has been described in Section 3.2.2. Fig. 5 illustrates how to construct short connections from side output 4 to side output 2. The score maps in side outputs 3 and 4 are first upsampled by simple bilinear interpolation and then concatenated to the original score map in side output 2. The hyper-parameters of bilinear interpolation can be derived according to the context. As salient object detection is a class-agnostic task, we further weight the foregoing score maps which have been enclosed by a dashed bounding box in Fig. 5 and introduce another 1×1 convolutional layer as the new score map of side output 2. A similar approach can be used for side outputs to which more than one short connection is connected. For instance, let us assume that 3 short connections are connected to side output 2. There would be 4 score maps being concatenated together within the dashed bounding box.

Our architecture can be functionally considered as two closely connected stages, which we call *saliency locating stage* and *details refinement stage*, respectively. The main focus of saliency locating stage is on looking for the most salient regions in a given image. For details refinement stage, we introduce a top-down method, a series of short connections from deeper side-output layers to shallower ones. The reason for such a consideration is that with the help of deeper side information, lower side outputs can both accurately predict the salient objects and refine the results from deeper side outputs, resulting in dense and accurate saliency maps. We further test the effectiveness of our proposed architecture by running a number of ablation experiments and showing the corresponding quantitative and visual results in the next section.

3.4 Implementation Details

Our network is based on the publicly available Caffe library [52] and the open implementation of FCN [29]. As mentioned above, we choose VGGNet [28] as our pre-trained model for better comparison with other works.

[1] We add a new side output in our enhanced HED architecture.

3.4.1 Inference

Although a series of short connections are introduced, the quality of the prediction maps produced by the deeper and the shallower side outputs is still unsatisfactory. Regarding this fact, during the testing phase, we adopt a more complicated combination of these side outputs. Let $\tilde{Z}_1, \cdots, \tilde{Z}_6$ denote the score map of each side output, respectively. They can be computed by $\tilde{Z}_m = h(\tilde{R}^{(m)}_{\text{side}})$. Recall that $h(\cdot)$ in our case is the sigmoid function. Therefore, the fusion output map can be computed by

$$\tilde{Z}_{\text{fuse}} = h\big(\sum_{m=2}^{4} f_m \tilde{R}^{(m)}_{\text{side}}\big). \tag{10}$$

To avoid the negative effect caused by the bad quality of the prediction map from the deepest and shallowest side outputs, we also use $\tilde{Z}_2, \hat{Z}_3$, and $\hat{Z}_4$ to help further fill in the lost details. As a result, the final output map during inference can be represented by

$$\tilde{Z}_{\text{final}} = \text{Mean}(\tilde{Z}_{\text{fuse}}, \tilde{Z}_2, \tilde{Z}_3, \tilde{Z}_4). \tag{11}$$

Surprisingly, we found that such a combination do help improve the results by a little margin. This is due to the fact that although the fusion output map incorporates the aggregation of each side output, some detailed information in the fusion output map is still missed. Regarding the quality of each side output map (see Fig. 1), we decide to use Eqn. (11) as the final output map.

3.4.2 Smoothing Method

Though our model can precisely find the salient objects in an image, the boundary information of the resulting saliency maps is still lost for those complex scenes. To further improve spatial coherence and quality of our saliency maps, we adopt the fully connected conditional random field (CRF) method [53] as a selective layer during the inference phase.

The energy function of CRF is given by

$$E(\mathbf{x}) = \sum_i \theta_i(x_i) + \sum_{i,j} \theta_{ij}(x_i, x_j), \tag{12}$$

where x is the label prediction for pixels. To make our model more competitive, instead of directly using the predicted maps as the input of the unary term, we leverage the following unary term

$$\theta_i(x_i) = -\frac{\log \hat{S}_i}{\tau h(\hat{S}_i)}, \tag{13}$$

where $\hat{S}_i$ denotes normalized saliency value of pixel x_i, $h(\cdot)$ is the sigmoid function, and τ is a scale parameter. The pairwise potential is defined as

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j)\Bigg[w_1 \exp\Big(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\Big) + w_2 \exp\Big(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\Big)\Bigg], \tag{14}$$

where $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$ and zero, otherwise. I_i and p_i are pixel value and position of x_i, respectively. Parameters $w_1, w_2, \sigma_\alpha, \sigma_\beta$, and σ_γ control the importance of each Gaussian kernel.

In this paper, we employ a publicly available implementation of [53], called PyDenseCRF [2]. Since there are only two classes in our case, we use the inferred posterior probability of each pixel being salient as the final saliency map directly.

3.4.3 Parameters

The hyper-parameters used in this work include learning rate (1e-8), weight decay (0.0005), momentum (0.9), loss weight for each side output (1). We use full-resolution images to train our network, and the mini-batch size is set to 10. The kernel weights in newly added convolutional layers are all initialized with random numbers. Our fusion layer weights are all initialized with 0.1667 in the training phase. The parameters in the fully connected CRF are determined using cross validation on the validation set. In our experiments, τ is set to 1.05, and w_1, w_2, $\sigma_\alpha, \sigma_\beta$, and σ_γ are set to 3.0, 3.0, 60.0, 8.0, and 5.0, respectively.

4 Experiments and Analyses

In this section, we introduce utilized datasets and evaluation criteria and report the performance of our proposed approach. Besides, a number of ablation experiments are performed for analyzing the importance of each component of our approach.

4.1 Datasets

We evaluate our approach on 5 representative datasets, including MSRA-B [43], ECSSD [54], HKU-IS [47], PASCALS [55], and SOD [56], [57], all of which are available online. These datasets all contain a large number of images as well as well-segmented annotations and have been widely used recently.

MSRA-B contains 5,000 images from hundreds of different categories. Because of its diversity and large quantity, MSRA-B has been one of the most widely used datasets in salient object detection literature. Most images in this dataset have only one salient object, and hence it has gradually become a standard dataset for evaluating the capability of processing simple scenes. ECSSD contains 1,000 semantically meaningful but structurally complex natural images. HKU-IS is another large-scale dataset that contains more than 4000 challenging images. Most of images in this dataset have low contrast with more than one salient object. PASCALS contains 850 challenging images (each composed of several objects), all of which are chosen from the validation set of the PASCAL VOC 2010 segmentation dataset. We also evaluate our system on the SOD dataset, which is a subset of the BSDS dataset. It contains 300 images, most of which possess multiple salient objects. All of these datasets consist of ground truth human annotations.

In order to preserve the integrity of the evaluation and obtain a fair comparison with existing approaches,

[2] https://github.com/lucasb-eyer/pydensecrf

we utilize the same training and validation sets as in [25] and test over all of the datasets using the same model.

4.2 Evaluation Metrics

We use three universally-agreed, standard metrics (see also [23], [23], [41], [58]) to evaluate our model including precision-recall curves, F-measure, and the mean absolute error (MAE). For a given continuous saliency map S, we convert it to a binary mask B using a threshold. Then its precision and recall are computed as $precision = |B \cap Z|/|B|$ and $recall = |B \cap Z|/|Z|$, respectively, where $|\cdot|$ accumulates the non-zero entries in a mask. Averaging the precision and recall values over the saliency maps of a given dataset yields the PR curve.

To comprehensively evaluate the quality of a saliency map, the F-measure metric is used, which is defined as

$$F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 Precision + Recall}. \tag{15}$$

As suggested by previous works, we choose β^2 to be 0.3 for stressing the importance of the precision value.

Let $\hat{S}$ and $\hat{Z}$ denote the continuous saliency map and the ground truth that are normalized to $[0,1]$. The mean absolute error (MAE) score can be computed as

$$MAE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |\hat{S}(i,j) = \hat{Z}(i,j)|. \tag{16}$$

4.3 Ablation Analysis

We experiment with different design options and different short connection patterns to illustrate the effectiveness of each component of our method.

4.3.1 Various Short Connection Patterns

Our architecture as shown in Fig. 3 is so flexible that can be regarded as the generalized model of most existing architectures, such as those depicted in Fig. 2. To better show the strength of our proposed approach, we use different network architectures as listed in Fig. 2 for salient object detection. Besides the Hypercolumns architecture [37] and the HED-based architecture [26], we implement three representative patterns using our proposed approach. The first one is formulated as follows, which is a similar architecture to Fig. 2(d).

$$\tilde{R}_{\text{side}}^{(m)} = \begin{cases} r_{m+1}^{m} \tilde{R}_{\text{side}}^{(m+1)} + \hat{A}_{\text{side}}^{(m)}, \text{ for } m = 1,\ldots,5 \\ \hat{A}_{\text{side}}^{(m)}. \text{ for } m = 6 \end{cases} \tag{17}$$

The second pattern is represented as follows which is much more complex than the first one.

$$\tilde{R}_{\text{side}}^{(m)} = \begin{cases} \sum_{i=m+1}^{m+2} r_i^m \tilde{R}_{\text{side}}^{(i)} + \hat{A}_{\text{side}}^{(m)}, \text{ for } m = 1,2,3,4 \\ \hat{A}_{\text{side}}^{(m)}. \text{ for } m = 5,6 \end{cases} \tag{18}$$

The last pattern, the one used in this paper, is given by

$$\tilde{R}_{\text{side}}^{(m)} = \begin{cases} \sum_{i=3}^{6} r_i^m \tilde{R}_{\text{side}}^{(i)} + \hat{A}_{\text{side}}^{(m)}, \text{ for } m = 1,2 \\ r_5^m \tilde{R}_{\text{side}}^{(5)} + r_6^m \tilde{R}_{\text{side}}^{(6)} + \hat{A}_{\text{side}}^{(m)}, \text{ for } m = 3,4 \\ \hat{A}_{\text{side}}^{(m)}. \text{ for } m = 5,6 \end{cases} \tag{19}$$

The quantitative results are listed in Fig. 7. As can be seen from Fig. 7, by adding another side output and two additional convolutional layers in each side output, we have a performance gain of 2.5 points in terms of F-measure. In addition, with the increase of short connections, our approach gradually achieves better performance. Although there is no performance gain obtained when Pattern 1 is used compared with the enhanced HED structure, a gain of 0.8 points can be achieved when we turn to Pattern 2. Another 0.6 points gain can also be obtained when Pattern 3 is considered.

4.3.2 Details of Side-Output Layers

We run several ablation experiments to explore the best side output settings. The detailed information of each side-output layer in each experiment has been shown in Fig. 6. We use Pattern 3 in Fig. 7 as our baseline model. To highlight the importance of different parameters, we adopt the variable-controlling method that only changes one parameter at a time. Besides, all the results are tested on PASCALS dataset for fair comparison. Compared with the fourth experiment, the first one exploits more channels but the same F-measure score is obtained. This means that more channels for each side output cannot bring in additional performance gain. In the second experiment, we tried to reduce 1 convolutional layer in each side output but it turns out that such an operation decreases the performance by 1.5 points. In spite of a small decrease, it is enough to account for the importance of introducing two convolutional layers in each side output. Furthermore, we attempt to reduce the large kernel size in deeper side outputs. Similarly, this leads to a slight decrease in F-measure. All the above experiments demonstrate that the side output settings we use are reasonable and appropriate.

4.3.3 Upsampling Operation

In our approach, we use the in-network bilinear interpolation to perform upsampling in each side output. As implemented in [29], we use fixed deconvolutional kernels for our side outputs with different strides. Since the prediction maps generated by deep side-output layers are not dense enough, we also try to use the "hole algorithm" to make the prediction maps in deep side outputs denser. We adopt the same technique as in [35]. However, according to our experiments, using such a method yields a worse performance. We notice that as the fusion prediction map gets denser, some non-salient pixels are wrongly predicted as salient ones even though the CRF is used thereafter. The F-measure score on the validation set is decreased by nearly 1%.

No.	Side output 1	Side output 2	Side output 3	Side output 4	Side output 5	Side output 6	F_β
1	$(128, 3 \times 3) \times 2$	$(128, 3 \times 3) \times 2$	$(256, 5 \times 5) \times 2$	$(512, 5 \times 5) \times 2$	$(1024, 5 \times 5) \times 2$	$(1024, 7 \times 7) \times 2$	**0.830**
2	$(128, 3 \times 3) \times 1$	$(128, 3 \times 3) \times 1$	$(256, 5 \times 5) \times 1$	$(256, 5 \times 5) \times 1$	$(512, 5 \times 5) \times 1$	$(512, 7 \times 7) \times 1$	0.815
3	$(128, 3 \times 3) \times 2$	$(128, 3 \times 3) \times 2$	$(256, 3 \times 3) \times 2$	$(256, 3 \times 3) \times 2$	$(512, 5 \times 5) \times 2$	$(512, 5 \times 5) \times 2$	0.820
4	$(128, 3 \times 3) \times 2$	$(128, 3 \times 3) \times 2$	$(256, 5 \times 5) \times 2$	$(256, 5 \times 5) \times 2$	$(512, 5 \times 5) \times 2$	$(512, 7 \times 7) \times 2$	**0.830**

Fig. 6. Comparisons of different side output settings and their performance on PASCALS dataset [55]. $(c, k \times k) \times n$ means that there are n convolutional layers with c channels and size $k \times k$. Note that the last convolutional layer in each side output is unchanged as listed in Fig. 4. In each setting, we only modify one parameter while keeping all others unchanged so as to emphasize the importance of each chosen parameter.

Scheme	Architecture	F-measure
1	Hypercolumns [37]	0.818
2	Original HED [26]	0.791
3	Enhanced HED	0.816
4	Pattern 1 (Eqn. (17))	0.816
5	Pattern 2 (Eqn. (18))	0.824
6	Pattern 3* (Eqn. (19))	**0.830**

Fig. 7. The performance of different architectures on PASCALS dataset [55]. '*' represents the pattern used in this paper.

4.3.4 Data Augmentation

Data augmentation has been proven to be very useful in many learning-based vision tasks. As done in most previous works, we flip all the training images horizontally, resulting in an augmented image set with twice larger than the original one. We found that such an operation further improves the performance by more than 0.5%. In addition, we also try to crop the input images to a fixed size 321×321. However, experimental results show that such an operation decrease our performance by more than 0.5 points. This may be because input images with full size contain richer information that allows our network to better capture the salient objects.

4.3.5 Different Backbones

We also extend our work by replacing the VGGNet with ResNet-101 [59] as the backbone. Taking into account the network structure of ResNet-101, we only use the bottom 5 side outputs in Fig. 4, which are connected to conv1, res2c, res3b3, res4b22, and res5c, respectively. We keep other settings unchanged. We show the results on the bottom of Fig. 10. With the same training set, there is a further one-point improvement on each dataset in terms of F-measure score on average.

4.3.6 The Proposed CRF Model

Most previous works [35], [53] only use the negative log likelihood as the unary term in their CRF model. Differently from them, we introduce a modulating factor that aims to give positive predictions more confidence as shown in Eqn. (13). This is reasonable as most of the predictions are correct through observing the MAE scores. In our experiments, we found that adding such a modulating factor helps little on improving the F-measure scores but is able to further reduce the MAE scores (*i.e.* , reduce wrong predictions) by around 0.3 points.

4.4 Comparison with the State-of-the-art

We compare the proposed approach with 7 recent CNN-based methods, including MDF [47], DS [60], DCL [35], ELD [50], MC [49], RFCN [51], and DHS [36]. Four classical methods are also considered including RC [41], CHM [61], DSR [62], and DRFI [25], which have been proven to be the best in the benchmark study of Borji *et al.* [23]. It is worth mentioning that though more training images is able to bring us better results as shown in Fig. 14, our results here are mainly based on 2500 training images from MSRA-B dataset for fair comparison with existing works.

4.4.1 Visual Comparison

To exhibit the superiority of our proposed approach compared against the above-mentioned methods, we select multiple representative images from different datasets which incorporate a variety of difficult circumstances, including complex scenes, salient objects with center bias, salient objects with different sizes, low contrast between foreground and background, etc., and show the visual comparisons in Fig. 8. We manually split the selected images into multiple groups which are separated by solid lines. We also give each group multiple tags describing their properties.

Taking all circumstances into account, it can be easily seen that our proposed method not only highlights the right salient regions but also produces coherent boundaries. It is also worth mentioning that thanks to the short connections, our approach gives salient regions more confidence, yielding higher contrast between salient objects and the background. More importantly, it generates connected regions, which greatly strengthens the ability of our model. These advantages make our results very close to the ground truth and hence better than other methods in almost all circumstances which are shown in Fig. 8.

4.4.2 PR Curve

We compare our approach with the existing methods in terms of PR curve here. In Fig. 9, we depict the PR curves produced by our approach and previous state-of-the-art methods on 3 popular datasets. It is obvious that FCN-based methods substantially outperform other methods. More importantly, among all FCN-based methods, the PR curve of our approach is especially outstanding in the upper left corners of the coordinates. We can also find that the precision of

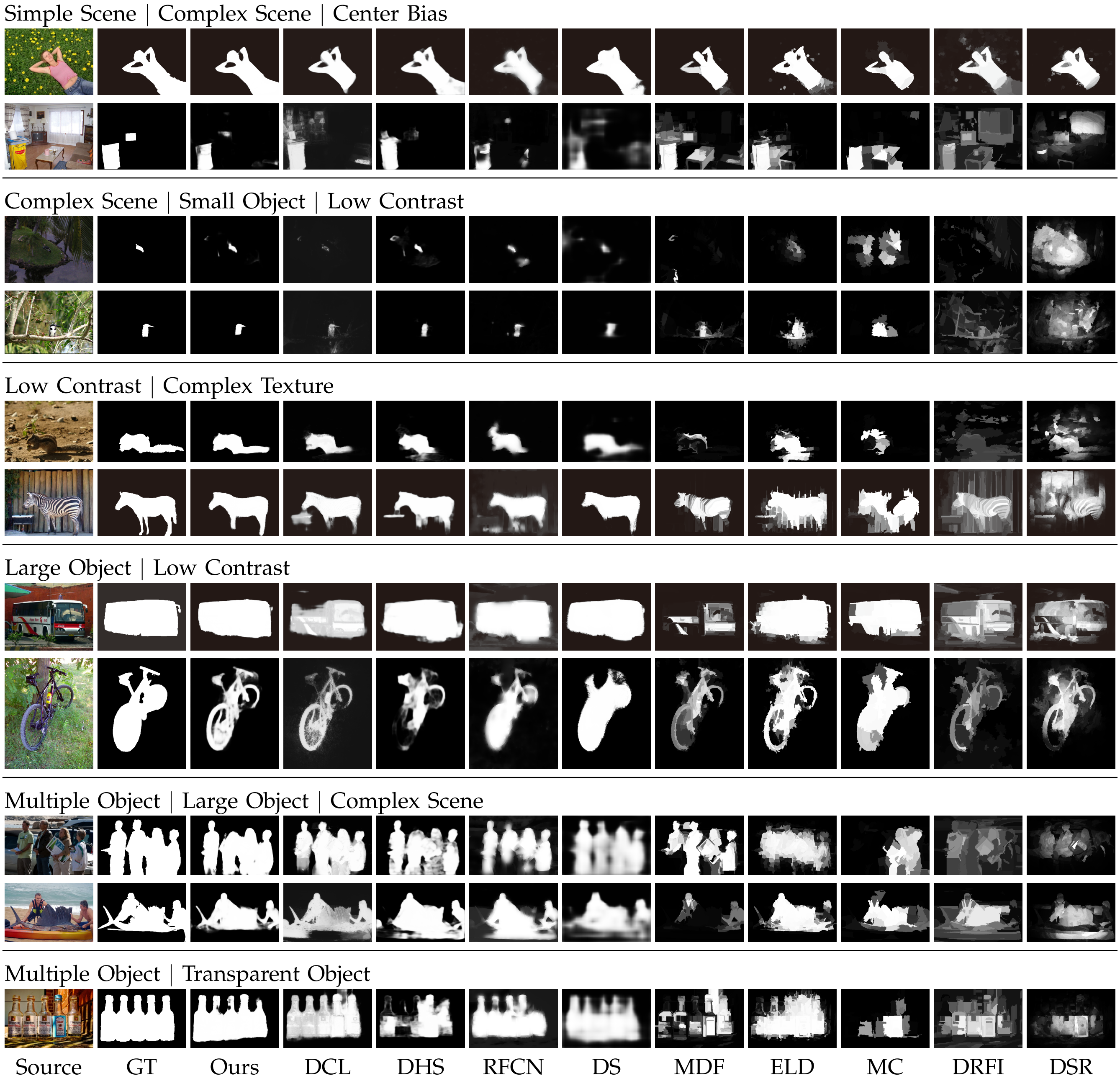

Simple Scene | Complex Scene | Center Bias
Complex Scene | Small Object | Low Contrast
Low Contrast | Complex Texture
Large Object | Low Contrast
Multiple Object | Large Object | Complex Scene
Multiple Object | Transparent Object
Source
GT
Ours
DCL
DHS
RFCN
DS
MDF
ELD
MC
DRFI
DSR

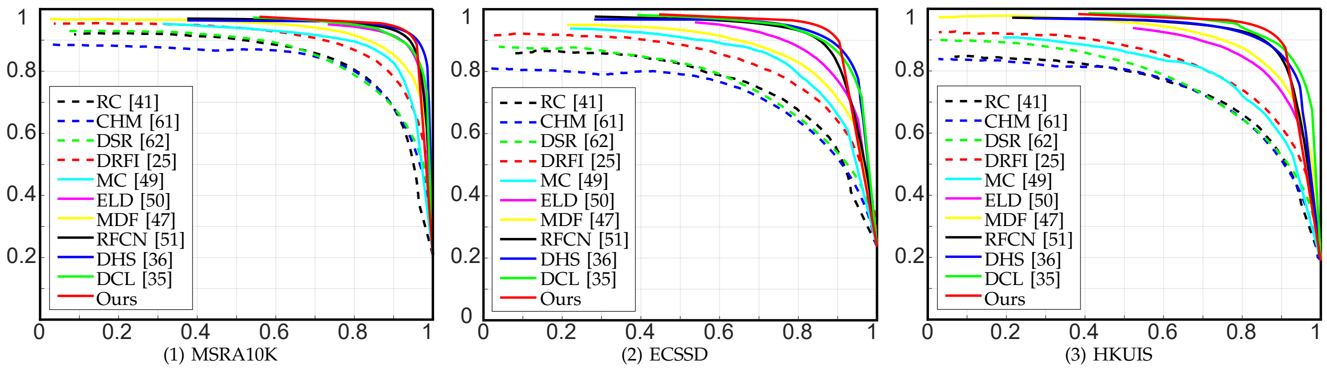
Fig. 8. Selected results from various datasets. We split the selected images into multiple groups, which are separated by solid lines. To better show the capability of processing different scenes for each approach, we highlight the features of images in each group.

our approach is much higher when the recall score is close to 1, reflecting that our false positives are much lower than other methods. This also indicates that our strategy of combining low-level and high-level features in terms of short connections is essential such that the resultant saliency maps look much closer to the ground truth.

4.4.3 F-measure and MAE

We also compare our approach with the existing methods in terms of F-meature and MAE scores. The quantitative results are shown in Fig. 10. As can be seen, our approach achieves the best score (maximum F-measure and MAE) on all datasets as listed in Fig. 10. On the ECSSD and SOD datasets, our approach improves the current best maximum F-measure by 1 point, which is a large margin as the values are already very close to ideal value 1. In regard to MAE scores, our approach achieves a more than 1-point decrease on MSRA-B and PASCALS datasets. On the other datasets, there are still at least 0.09 points improvements. This implies that the number of wrong predictions in our case is significantly less than the other methods.

Besides, we also observe that the proposed approach behaves even better on more difficult datasets, such as HKUIS [47], PASCALS [55], and SOD [56], [57], which contain a large number of images with multiple salient objects. This indicates that our method is capable of detecting and segmenting the most salient object, while other methods often fail at one of these stages.



(1) MSRA10K (2) ECSSD (3) HKUIS

Fig. 9. Precision (vertical axis) recall (horizontal axis) curves on three popular salient object datasets.

	Training		MSRA-B [43]		ECSSD [54]		HKU-IS [47]		PASCALS [55]		SOD [57]	
Methods	Dataset	#Images	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
RC [41]	-	-	0.817	0.138	0.741	0.187	0.726	0.165	0.640	0.225	0.657	0.242
CHM [61]	-	-	0.809	0.138	0.722	0.195	0.728	0.158	0.631	0.222	0.655	0.249
DSR [62]	-	-	0.812	0.119	0.737	0.173	0.735	0.140	0.646	0.204	0.655	0.234
DRFI [25]	MB	2,500	0.855	0.119	0.787	0.166	0.783	0.143	0.679	0.221	0.712	0.215
MC [49]	MK	8,000	0.872	0.062	0.822	0.107	0.781	0.098	0.721	0.147	0.708	0.184
ELD [50]	MK	9,000	0.914	0.042	0.865	0.981	0.844	0.071	0.767	0.121	0.760	0.154
MDF [47]	MB	2,500	0.885	0.104	0.833	0.108	0.860	0.129	0.764	0.145	0.785	0.155
DS [60]	MB	2,500	-	-	0.810	0.160	-	-	0.818	0.170	0.781	0.150
RFCN [51]	MK	10,000	0.926	0.062	0.898	0.097	0.895	0.079	0.827	0.118	0.805	0.161
DHS [36]	MK + D	9,500	-	-	0.905	0.061	0.892	0.052	0.820	0.091	0.823	0.127
DCL$^+$ [35]	MB	2,500	0.916	0.047	0.898	0.071	0.907	0.048	0.822	0.108	0.832	0.126
Ours	MB	2,500	**0.927**	**0.028**	**0.915**	**0.052**	**0.913**	**0.039**	**0.830**	**0.080**	**0.842**	**0.118**
Ours†	MB	2,500	0.936	0.030	0.928	0.048	0.920	0.035	0.838	0.092	0.850	0.119

Fig. 10. Quantitative comparisons with 11 methods on 5 popular datasets. The ResNet-101 [59] version of our approach (*i.e.* 'Ours†') clearly outperforms its VGGNet version. For fair comparison, we exclude 'Ours†' and highlight the best result of each column in **bold**. Here we use the initials of each dataset for convenience.

Methods	JSOD [63]	MSRA-B [43]	ECSSD [54]
Wang *et al.* [64]	90.64%	89.26%	70.50%
SSVM [63]	**99.22%**	98.66%	94.40%
Ours	98.84%	**99.05%**	**96.8%**

Fig. 11. The prediction accuracy of our saliency existence branch compared to SSVM [63] and Wang *et al.* [64]. The best result of each column is highlighted in **bold**.

4.5 The Existence of Saliency

To date, most existing salient object detection methods focus on datasets in which at least one salient object exists. However, in many real-world scenarios, salient objects do not always exists. Therefore, methods based on the above assumption may easily lead to incorrect prediction results when applied to scenes without any salient objects in them. To solve this problem, we propose to introduce another branch into our network to predict the saliency existence of the input image. The new branch is composed of a global average pooling layer, followed by a multi-layer perceptron (MLP) as the regressor to recognize the existence of saliency as done in many classification networks [28], [59]. The global average pooling layer is used to transform feature maps with different shapes into the same size so that the resulting feature vectors can be fed into the MLP. Like [28], [32], the MLP here consists of three fully-connected layers, all of which are with 1,024 neurons except the last one which has two. The softmax loss is used to optimize the new branch.

In our experiments, we use the same training set as in [63], which contains 5,000 background images (*i.e.* images without salient objects in them) and 5,000 images from MSRA10K [41]. For these background images, the gradients from the salient object detection module are not allowed to back-propagate so that the resulting prediction maps would not be interfered. We found that this operation is essential. The hyper-parameters used here are the same to our salient object detection experiments. We train our network for 24,000 iterations and decrease the learning rate by a factor of 10 at 20,000 iterations. We test our model on three datasets, including JSOD [63], MSRA-B [43] and ECSSD [54]. Fig. 11 lists the results compared to another two works SSVM [63] and Wang *et al.* [64]. Since there is a clear separation between JSOD dataset (mostly containing pure textures) and other two datasets (MSRA-B and ECSSD mostly contain images with clear salient objects), the classification results on all datasets have been already saturated (very close to the ideal value "1"). Thus, we expect more challenging dataset which better reflect real world difficulties would be developed in near future.

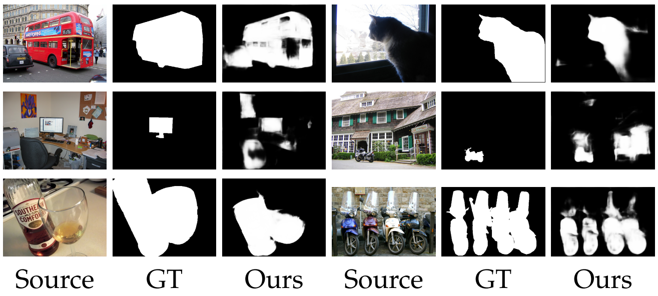


Fig. 12. Failure cases selected from multiple datasets. As can be seen, most cases are caused by complex background, low contrast between foreground and background, and transparent objects.

4.6 Timing

Our network is fully convolutional, which allows it to run very fast compared against most previous salient object detection methods. When trained on the MSRA-B dataset which contains 2,500 training images, our network takes less than 8 hours for 12,000 iterations. Interestingly, though 10,000 iterations are enough for convergence, we found another 2,000 iterations still bring us a small performance gain in MAE.

During the inference stage, it takes us about 0.08s to process an input image of size 300×400. This is extremely faster than most of the previous works, such as DCL [35] which need more than 1s for each image of the same size. With our CRF layer considered, another 0.4 seconds are needed. As a result, our overall time cost is less than 0.5s for an image of size 300×400.

5 Discussion

In this section, we conduct useful analysis on our proposed approach, which we believe would be helpful for researchers to develop more powerful methods.

5.1 Failure Case Analysis

Some failure predictions of our approach have been shown in Fig. 12. As can be seen, these failure cases can be categorized into three circumstances in general. The first one is actually the common defect of CNN-based salient object detection methods, in which the salient objects cannot be completely segmented out, leaving a small part of the salient object missed. Typical examples are the images shown in the first row of Fig. 12. In the second circumstance, the main body of the salient object cannot be extracted or non-salient regions are predicted to be salient. As shown in the middle row of Fig. 12, this case is mostly caused by complex backgrounds and very low contrast. The last type of failure cases is caused by transparent objects as shown in the bottom row of Fig. 12. Though our approach can detect some parts of the transparent objects, to segment the complete objects out is still very difficult.

We argue that three possible remedies can be used to solve the aforementioned problems. First of all, a promising solution is to provide more prior knowledge on segment level so that regions with similar textures or colors can be detected simultaneously. Because of the internal structure of CNNs, the correlations of two positions in the score map are decided by the learnable weights of the former layers, making this problem difficult to be solved by the networks themselves. Segment-level information allows CNNs to correct those wrong predictions in the Circumstance 1 mentioned above. In addition, segment-level information can also serve as a post-processing tool to further refine the predicted saliency maps by a simple voting strategy. Secondly, more powerful training data should be presented, including both simple and complex scenes. As listed in Fig. 14, training data with complex scenes can substantially help improve the performance on both easy and difficult datasets. Another solution should be designing more advanced models and then extracting more powerful feature representations to deal with challenging inputs with complex structures [65].

5.2 Benchmarking Training Set

The selection of training set is one of the important aspects for a learning based algorithm. A good training set will definitely improve the learning ability, leading to a more generative model that can perform well on almost all scenes, even with complex background. However, the training sets of recent learning based approaches are different and none of these works have explored which training set is the best. Fig. 10 lists the details of different training sets that existing approaches have used. Furthermore, training on different datasets with different sizes makes the comparisons unfair. Albeit the number of training images is not proportional to the performance gain, the size and quality of different training sets break the fair comparisons among different approaches. One can observe in Fig. 10 that some of them only use a training set with 2,500 images while some others leverage around 10,000 images for training.

In this section, we attempt to thoroughly analyze the effect of utilizing different datasets for training based on our proposed approach. Our goal is to provide a new, unified, convincing, and large-scale training set based on existing datasets for future research. To do so, we perform a number of experiments and show exhaustive comparisons among 6 widely-used and publicly available datasets, which can be found in Fig. 13. Notice that all the training lists will be made *publicly available*. During testing phase, we use both the max F-measure score and MAE score as measuring metrics. Notice that since most datasets contain more than 5,000 images, each model is trained for 16,000 iterations here. An exception is the model trained on ECSSD with 6,000 iterations.

5.2.1 Dataset Quality Measuring

To exhibit the quality of datasets better, each time we train on one of them, except for the SOD dataset which has only 300 images and the PASCALS dataset which has a lowly consistent behavior, and test on all the test sets. As ECSSD contains less than 2,000 images, all the

	MSRA-B		ECSSD		HKU-IS		PASCALS		SOD		DUT-OMRON	
Training Set	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
MSRA-B (2500)	**0.920**	**0.043**	0.908	**0.064**	0.902	0.049	0.824	0.101	0.836	0.126	0.764	0.070
ECSSD (1000)	0.880	0.062	-	-	0.891	0.051	0.807	0.100	**0.840**	**0.107**	0.720	0.085
HKU-IS (2500)	0.893	0.057	0.898	0.070	**0.919**	**0.041**	0.817	**0.099**	0.820	0.133	0.737	0.085
DUT-OMRON (3103)	0.890	0.060	0.895	0.079	0.888	0.059	0.811	0.113	0.814	0.141	**0.828**	**0.051**
MSRA10K (6000)	-	-	**0.909**	0.068	0.901	0.054	**0.826**	0.107	0.822	0.140	0.769	0.074

Fig. 13. Performance when different training sets are used. The best results are highlighted in **bold**. Notice that all the results here are without CRF.

images are used for training and hence no image is left for testing. For the remaining large-scale datasets, if default splits are provided then they will be used directly. Otherwise, we split the dataset in a ratio of 6:1:3 for training, validation, and testing, respectively.

Detailed experimental results have been shown in Fig. 13. As there is a large overlap between MSRA-B and MSRA10K datasets, we only show the results on MSRA-B instead of both. According to the results shown in Fig. 13, the following conclusion can be drawn. First, the best result on each dataset is always obtained by training on the corresponding training set, and the phenomenon is especially obvious for DUT-OMRON. This might be caused by the characteristics of the images in each dataset, making different datasets favor different features. Consequently, we argue that it is *inappropriate* to directly compare performance numbers that are achieved by different models trained on different datasets (see also Fig. 13). Second, having more training images does not necessarily entail better performance. As can be seen in Fig. 13, training on ECSSD dataset allows us to achieve the best performance on the SOD dataset despite of having only 1,000 training images. In regard to the above-mentioned issues, a compromise solution is to construct a *unified*, *composite*, and *versatile* dataset.

5.2.2 Beyond Training on Individual Datasets

We select 4 datasets from Fig. 13 to build composite datasets for comparisons. Though the MSRA10K is more than twice bigger than MSRA-B dataset, models trained on it have a competitive performance compared to those trained on the MSRA-B dataset. Here we just keep MSRA-B for training due to its high-quality images and annotations. Therefore, there are totally 11 different combinations which have been shown in the second column of Fig. 14. During the testing phase, we also use the six test sets mentioned above for fair comparisons.

From the results in Fig. 14, the following conclusions can be drawn. First of all, a larger training set does not necessarily mean higher test performance. This phenomenon can be observed through comparing Scheme 3 with other schemes. Despite only 3,500 training images, this combination performs better than those with more than 6,000 training images. It is true that the quality of annotations might be an essential reason that causes such a problem. However, such a consideration is beyond the scope of this paper. All conclusions here are based on the assumption that each dataset we use is with well-segmented annotations.

Second, an inappropriate combination of datasets may result in worse performance compared with individual datasets. By comparing schemes 4 and 0, one can find that despite better performance on HKU-IS, PASCALS, and SOD datasets there are still slight decreases when testing on MSRA-B and DUT-OMRON datasets.

Through this series of experiments, we aimed to emphasis that a training set with a large quantity of images may not be capable of bringing in better performance gain. A good training set should take into account as many cases as possible. However, because of the diversity of existing datasets, it is hard to obtain a convincing dataset that can behave the consistency among all existing datasets. In regard to the current state in salient object detection, we recommend using our Scheme 11 in Fig. 14 as training set for fair comparison and fitting decreasing performance bias caused by different training sets. Another severe problem in salient object detection is that most datasets are no longer challenging. An explicit effect is that the differences between different models are difficult to be distinguished because of the close performance on existing datasets. We hope that more challenging datasets with complex scenes and high consistency would be presented in the near future.

6 Conclusion

In this paper, we presented a deeply supervised network for salient object detection. Instead of directly connecting loss layers to the last layer of each stage, we introduce a series of short connections between shallower and deeper side-output layers. With these short connections, the activation of each side-output layer gains the capability of both highlighting the entire salient object and accurately locating its boundary. A fully connected CRF is also employed for correcting wrong predictions and further improving spatial coherence. Our experiments demonstrate that these mechanisms result in more accurate saliency maps over a variety of images. Our approach significantly advances the state-of-the-art and is capable of capturing salient regions in both simple and difficult cases, which further verifies the merit of the proposed architecture.

Acknowledgments

This research was supported by NSFC (NO. 61572264, 61620106008), Huawei Innovation Research Program,

Scheme	Training Set	MSRA-B		HKU-IS		PASCALS		SOD		DUT-OMRON	
		F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
0	MB (2500)	0.920	**0.043**	0.902	0.049	0.824	0.101	0.836	0.126	0.764	0.070
1	D + E (4103)	0.901	0.053	0.907	0.048	0.832	0.090	0.846	0.109	0.832	0.050
2	H + E (3500)	0.897	0.054	0.923	**0.040**	0.825	0.092	0.849	**0.108**	0.753	0.078
3	D + H (5603)	0.905	0.053	0.924	0.042	0.832	0.096	0.839	0.130	0.833	0.052
4	MB + E (3500)	0.916	0.045	0.909	0.045	0.835	0.091	0.852	0.111	0.758	0.073
5	MB + H (5000)	0.920	0.045	0.925	**0.040**	0.834	0.095	0.845	0.121	0.774	0.072
6	MB + D (5603)	0.921	0.046	0.910	0.050	0.837	0.099	0.845	0.127	0.840	**0.049**
7	MB + E + D (6603)	0.921	0.046	0.915	0.048	0.842	0.091	0.858	0.115	0.839	0.051
8	MB + H + D (8103)	**0.923**	0.046	0.926	0.043	0.837	0.096	0.855	0.123	0.840	0.051
9	MB + E + H (6000)	0.921	0.045	0.926	**0.040**	0.841	0.090	0.860	0.111	0.786	0.069
10	E + D + H (6603)	0.911	0.050	0.925	0.041	**0.844**	**0.087**	0.854	0.110	0.835	0.051
11	MB + E + D + H (9103)	**0.923**	0.046	**0.927**	0.042	**0.844**	0.091	**0.864**	0.113	**0.843**	0.051

Fig. 14. Detailed information of different training sets and the corresponding results on 5 datasets. The best results are highlighted in *bold*. All the results are obtained without any post-processing. Here we use the initials of each dataset for convenience.

CAST YESS Program, and IBM Global SUR award.

References

[1] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[2] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 185–198, 2010.

[3] J. Guo, T. Ren, L. Huang, X. Liu, M.-M. Cheng, and G. Wu, "Video salient object detection via cross-frame cellular automata," in *Int. Conf. Multimedia and Expo*, 2017, pp. 325–330.

[4] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof, "Saliency driven total variation segmentation," in *Int. Conf. Comput. Vis.*, 2009, pp. 817–824.

[5] M.-M. Cheng, F.-L. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Repfinder: finding approximately repeated scene elements for image editing," in *ACM Trans. Graph.*, vol. 29, no. 4, 2010, p. 83.

[6] G.-X. Zhang, M.-M. Cheng, S.-M. Hu, and R. R. Martin, "A shape-preserving approach to image resizing," *Comput. Graph. Forum*, vol. 28, no. 7, pp. 1897–1906, 2009.

[7] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2004.

[8] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, Y. Zhao, and S. Yan, "Stc: A simple to complex framework for weakly-supervised semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2314–2320, 2017.

[9] Q. Hou, P. K. Dokania, D. Massiceti, Y. Wei, M.-M. Cheng, and P. H. S. Torr, "Mining pixels: Weakly supervised semantic segmentation using image labels," in *EMMCVPR*, 2017.

[10] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[11] Y. Wei, X. Liang, Y. Chen, Z. Jie, Y. Xiao, Y. Zhao, and S. Yan, "Learning to segment with image-level annotations," *Pattern Recognition*, vol. 59, pp. 234–244, 2016.

[12] A. Borji, S. Frintrop, D. N. Sihite, and L. Itti, "Adaptive object tracking by learning background context," in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2012.

[13] P. L. Rosin and Y.-K. Lai, "Artistic minimal rendering with lines and blocks," *Graphical Models*, vol. 75, no. 4, pp. 208–229, 2013.

[14] J. Han, E. J. Pauwels, and P. De Zeeuw, "Fast saliency-aware multi-modality image fusion," *Neurocomputing*, pp. 70–80, 2013.

[15] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, "Sketch2photo: Internet image montage," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 124:1–10, 2009.

[16] S.-M. Hu, T. Chen, K. Xu, M.-M. Cheng, and R. R. Martin, "Internet visual media processing: a survey with graphics and vision applications," *The Vis. Comput.*, vol. 29, no. 5, pp. 393–405, 2013.

[17] H. Liu, L. Zhang, and H. Huang, "Web-image driven best views of 3d shapes," *The Vis. Comput.*, pp. 1–9, 2012.

[18] J.-Y. Zhu, J. Wu, Y. Wei, E. Chang, and Z. Tu, "Unsupervised object class discovery via saliency-guided multiple class learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 3218–3225.

[19] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai, "3-D object retrieval and recognition with hypergraph analysis," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4290–4303, 2012.

[20] M.-M. Cheng, Q.-B. Hou, S.-H. Zhang, and P. L. Rosin, "Intelligent visual media processing: When graphics meets vision," *J. Comput. Sci. Tech.*, 2017.

[21] A. Abdulmunem, Y.-K. Lai, and X. Sun, "Saliency guided local and global descriptors for effective action recognition," *Computational Visual Media*, vol. 2, no. 1, pp. 97–106, 2016.

[22] L. Itti and C. Koch, "Computational modeling of visual attention," *Nature reviews neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[23] 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, 2015.

[24] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, 2013.

[25] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng, "Salient object detection: A discriminative regional feature integration approach," *Int. J. Comput. Vis.*, vol. 123, no. 2, pp. 251–268, 2017, http://people.cs.umass.edu/~hzjiang/.

[26] S. Xie and Z. Tu, "Holistically-nested edge detection," *Int. J. Comput. Vis.*, vol. 125, no. 1, pp. 3–18, 2017.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. Neural Inform. Process. Syst.*, 2012, pp. 1097–1105.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.*, 2015.

[29] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3431–3440.

[30] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[31] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[32] R. Girshick, "Fast r-cnn," in *Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[33] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster r-cnn doing well for pedestrian detection?" in *Eur. Conf. Comput. Vis.*, 2016, pp. 443–457.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *P. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[35] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[36] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 678–686.

[37] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 447–456.

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

[39] Y. Xie, H. Lu, and M.-H. Yang, "Bayesian saliency via low and mid level cues," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1689–1698, 2013.

[40] W. Qi, M.-M. Cheng, A. Borji, H. Lu, and L.-F. Bai, "Saliencyrank: Two-stage manifold ranking for salient object detection," *Computational Visual Media*, vol. 1, no. 4, pp. 309–320, 2015.

[41] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," vol. 37, no. 3, pp. 569–582, 2015.

[42] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," in *Int. Conf. Comput. Vis.*, 2013, pp. 1529–1536.

[43] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, 2011.

[44] A. Borji and L. Itti, "Exploiting local and global patch rarities for saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 478–485.

[45] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, "Salient object detection: A survey," *arXiv preprint arXiv:1411.5878*, 2014.

[46] S. He, R. Lau, W. Liu, Z. Huang, and Q. Yang, "Supercnn: A superpixelwise convolutional neural network for salient object detection," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 330–344, 2015, http://www.shengfenghe.com/.

[47] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 5455–5463, http://i.cs.hku.hk/~yzyu/vision.html.

[48] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3183–3192.

[49] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1265–1274, https://github.com/Robert0812/deepsaldet.

[50] L. Gayoung, T. Yu-Wing, and K. Junmo, "Deep saliency with encoded low level distance map and high level features," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, https://github.com/gylee1103/SaliencyELD.

[51] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *Eur. Conf. Comput. Vis.*, 2016.

[52] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[53] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Adv. Neural Inform. Process. Syst.*, 2011.

[54] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1155–1162.

[55] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 280–287.

[56] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Int. Conf. Comput. Vis.*, 2001, pp. 416–423.

[57] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2010, pp. 49–56.

[58] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 733–740.

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[60] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang, "Deepsaliency: Multi-task deep neural network model for salient object detection," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3919 – 3930, 2016, https://github.com/zlmzju/DeepSaliency.

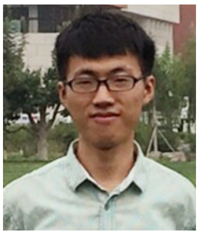
[61] X. Li, Y. Li, C. Shen, A. Dick, and A. Van Den Hengel, "Contextual hypergraph modeling for salient object detection," in *Int. Conf. Comput. Vis.*, 2013, pp. 3328–3335.

[62] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *Int. Conf. Comput. Vis.*, 2013, pp. 2976–2983.

[63] H. Jiang, M.-M. Cheng, S.-J. Li, A. Borji, and J. Wang, "Joint salient object detection and existence prediction," *Front. Comput. Sci.*, 2017.

[64] P. Wang, J. Wang, G. Zeng, J. Feng, H. Zha, and S. Li, "Salient object detection for searched web images via global saliency," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012.

[65] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A New Way to Evaluate Foreground Maps," in *Int. Conf. Comput. Vis.*, 2017.



Qibin Hou is currently a Ph.D. Candidate with College of Computer Science and Control Engineering, Nankai University, under the supervision of Prof. Ming-Ming Cheng. His research interests include deep learning, image processing, and computer vision.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then he did 2 years research fellow, with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests includes computer graphics, computer vision, and image processing. He received research awards including ACM China Rising Star Award, IBM Global SUR Award, CCF-Intel Young Faculty Researcher Program, *etc.*

Xiaowei Hu is currently a Master student with College of Computer Science and Control Engineering, Nankai University, under the supervision of Prof. Ming-Ming Cheng. His research interests include deep learning, image processing, and computer vision.

Ali Borji received the PhD degree in cognitive neurosciences from the Institute for Studies in Fundamental Sciences (IPM), 2009. He is currently an assistant professor at Center for Research in Computer Vision, University of Central Florida. His research interests include visual attention, visual search, machine learning, neurosciences, and biologically plausible vision models.

Zhuowen Tu received the BE degree from the Beijing Information Technology Institute, the ME degree from Tsinghua University, and the PhD degree in computer science from Ohio State University. He is an associate professor of cognitive science with the University of California, San Diego (UCSD). His main research interests include computer vision, machine learning, and neural computation.

Philip H.S. Torr received the PhD degree from Oxford University. After working for another 3 years at Oxford, he worked for 6 years as a research scientist for Microsoft Research, first in Redmond, then in Cambridge, founding the vision side of the Machine Learning and Perception Group. He is now a professor at Oxford University. He has won awards from several top vision conferences, including ICCV, CVPR, ECCV, NIPS *etc.* He is a Royal Society Wolfson Research Merit Award holder.