Deeply Supervised Salient Object Detection with Short Connections

Qibin Hou[1] Ming-Ming Cheng[1] * Xiaowei Hu[1] Ali Borji[2] Zhuowen Tu[3] Philip Torr[4]
[1]CCCE, Nankai University [2]CRCV, UCF [3]UCSD [4]University of Oxford
http://mmcheng.net/dss/

Abstract

Recent progress on saliency detection is substantial, benefiting mostly from the explosive development of Convolutional Neural Networks (CNNs). Semantic segmentation and saliency 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. Holisitcally-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 saliency method by introducing short connections to the skip-layer structures within the HED architecture. Our framework provides rich multi-scale feature maps 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.*

1. Introduction

The goal in salient object detection is to identify the most visually distinctive objects or regions in an image. Salient object detection methods commonly serve as the first step for a variety of computer vision applications including image and video compression [15, 20], image segmentation [10], content-aware image editing [9], object recognition [44, 48], visual tracking [2], non-photo-realist rendering [43, 16], photo synthesis [6, 14, 19], information discovery [35, 53, 8], image retrieval [18, 12, 8], *etc.*

Inspired by cognitive studies of visual attention [21, 41, 11], computational saliency detection has received great research attention in the past two decades [22, 4, 1, 5]. Encouraging research progress has been continuously observed by enriching simple local analysis [22] with global cues [7], rich feature sets [24], and their learned combination weights [37], indicating the importance of powerful feature representations for this task. Leading method [24] on a latest benchmark [1] uses as many as 34 hand crafted features. However, further improvement using manually enriched feature representations is non-trivial.

In a variety of computer vision tasks, such as image classification [27, 45], semantic segmentation [38], and edge detection [49], convolutional neural networks (CNNs) [28] have successfully broken the limits of traditional hand crafted features. This motivates recent research efforts of using Fully Convolutional Neural Networks (FCNs) for salient object detection [29, 46, 52, 13, 36]. The Holistically-Nested Edge Detector (HED) [49] model, which explicitly deals with the scale space problem, has lead to large improvements over generic FCN models in the context of edge detection. However, the skip-layer structure with deep supervision in the HED model does not lead to obvious performance gain for saliency detection.

As demonstrated in Fig. 1, we observe that i) deeper side outputs encodes high level knowledge and can better locate salient objects; ii) shallower side outputs capture rich spatial information. This motivated us to develop a new method for salient object detection by introducing *short connections* to the skip-layer structure within the HED [49] architecture. By having a series of short connections from deeper side outputs to shallower ones, our new framework offers two advantages: i) high-level features can be transformed to shallower side-output layers and thus can help them better locate the most salient region; 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 need to do salient object detection. Experimental results show that 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. To facilitate future research in our related areas, we release

*M.M. Cheng (cmm@nankai.edu.cn) is the corresponding author.

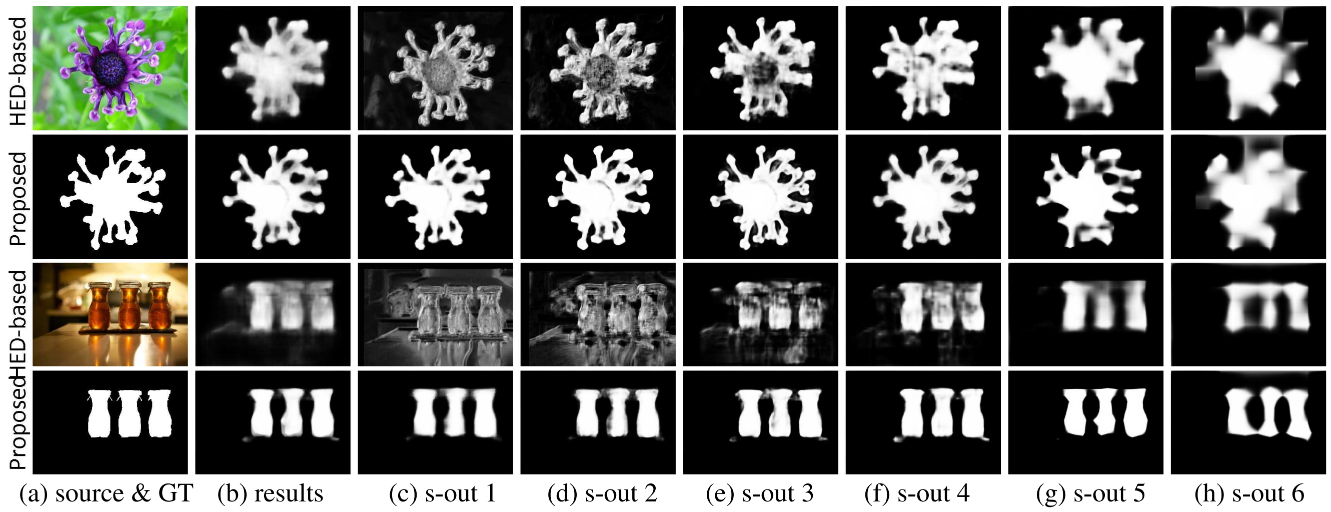


Figure 1: Visual comparison of saliency maps produced by the HED-based method [49] and ours. Though saliency maps produced by deeper (4-6) side output (s-out) look similar, because of the introduced short connections, each shallower (1-3) side output can generate satisfactory saliency maps and hence a better output result.

both source code and trained models.

2. Related Works

Over the past two decades [22], 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 [22, 25, 50], global features [7, 42, 37, 24], or both (e.g., [3]). A complete survey of these methods is beyond the scope of this paper and we refer the readers to a recent survey paper [1] for details. Here, we focus on discussing recent salient object detection methods based on deep learning architectures.

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. Li *et al.* [29] proposed to use multi-scale features extracted from a deep CNN to derive a saliency map. Wang *et al.* [46] predicted saliency maps by integrating both local estimation and global search. Two different deep CNNs are trained to capture local information and global contrast. In [52], Zhao *et al.* presented a multi-context deep learning framework for salient object detection. They employed two different CNNs to extract global and local context information, respectively. Lee *et al.* [13] considered both high-level features extracted from CNNs and hand-crafted features. To combine them together, a unified fully connected neural network was designed to estimate saliency maps. Liu *et al.* [36] designed a two-stage deep network, in which a coarse prediction map was produced, followed by another network to refine the details of the prediction map hierarchically and progressively. A deep contrast network was proposed in [30]. It combined a pixel-level fully convolutional stream and a segment-wise spatial pooling stream. Though significant progress have been achieved by these developments in the last two years, 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

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 [49], in which a series of side-output layers are added after the last convolutional layer of each stage in VGGNet [45]. Fig. 2(b) provides an illustration of the HED model. However, experimental results show that such a successful architecture is not suitable for salient object detection. Fig. 1 provides such an illustration. The reasons for this phenomenon are two-fold. On one hand, saliency detection is a more difficult vision task than edge detection that demands special treatment. A good saliency detection algorithm should be capable of extracting the most visually distinctive objects/regions from an image instead of simple edge information. On the other hand, the features generated from lower stages are too messy while the saliency maps obtained from the deeper side-output layers are short of regularity.

To overcome the aforementioned problem, 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

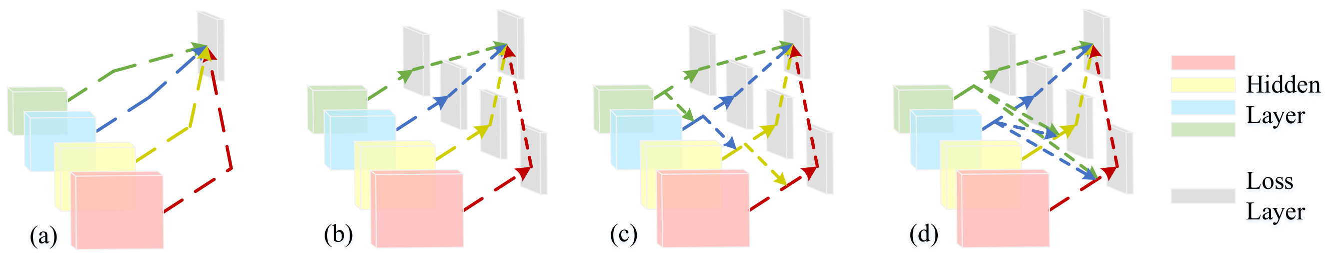


Figure 2: Illustration of different architectures. (a) Hypercolumn [17], (b) HED [49], (c) and (d) different patterns of our proposed architecture. As can be seen, a series of short connections are introduced in our architecture for combining the advantages of both deeper layers and shallower layers. While our approach can be extended to a variety of different structures, we just list two typical ones.

of the proposed approach.

3.1. HED-based saliency detection

We shall start out with the standard HED architecture [49] as well as its extended version, a special case of this work, for salient object detection and gradually move on to our proposed architecture.

HED architecture [49]. 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 $\mathbf{W}$. Suppose there are totally M side outputs. Each side output is associated with a classifier, in which the corresponding weights can be represented by

$$\mathbf{w} = (\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \dots, \mathbf{w}^{(M)}). \tag{1}$$

Thus, the side objective function of HED can be given by

$$L_{\text{side}}(\mathbf{W}, \mathbf{w}) = \sum_{m=1}^{M} \alpha_m l_{\text{side}}^{(m)}(\mathbf{W}, \mathbf{w}^{(m)}), \tag{2}$$

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 [49] 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

$$L_{\text{fuse}}(\mathbf{W}, \mathbf{w}, \mathbf{f}) = \sigma\big(Z, h(\sum_{m=1}^{M} f_m A_{\text{side}}^{(m)})\big), \tag{3}$$

where $\mathbf{f} = (f_1, \dots, f_M)$ is the fusion weight, $A_{\text{side}}^{(m)}$ 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 [49] here. Therefore, the final loss function can be given by

$$L_{\text{final}}(\mathbf{W}, \mathbf{w}, \mathbf{f}) = L_{\text{fuse}}(\mathbf{W}, \mathbf{w}, \mathbf{f}) + L_{\text{side}}(\mathbf{W}, \mathbf{w}). \tag{4}$$

HED connects each side output to the last convolutional layer in each stage of the VGGNet [45], respectively conv1_2, conv2_2, conv3_3, conv4_3, conv5_3. Each side output is composed of a one-channel convolutional layer with kernel size 1×1 followed by an up-sampling layer for learning edge information.

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 in VGGNet [45]. Besides, since salient object detection is a more difficult task than edge detection, we add two other convolutional layers with different filter channels and spatial sizes in each side output, which can be found in Table 1. 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:

$$\hat{l}_{\text{side}}^{(m)}(\mathbf{W}, \hat{\mathbf{w}}^{(m)}) = -\sum_{j \in Z} z_j \log \Pr\big(z_j = 1 | X; \mathbf{W}, \hat{\mathbf{w}}^{(m)}\big) + (1 - z_j) \log \Pr\big(z_j = 0 | X; \mathbf{W}, \hat{\mathbf{w}}^{(m)}\big), \tag{5}$$

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 [49], we add a weighted-fusion layer to connect each side activation. The loss function at the fusion layer in our case can be represented by

$$\hat{L}_{\text{fuse}}(\mathbf{W}, \hat{\mathbf{w}}, \mathbf{f}) = \hat{\sigma}\big(Z, \textstyle\sum_{m=1}^{\hat{M}} f_m \hat{A}_{\text{side}}^{(m)}\big), \tag{6}$$

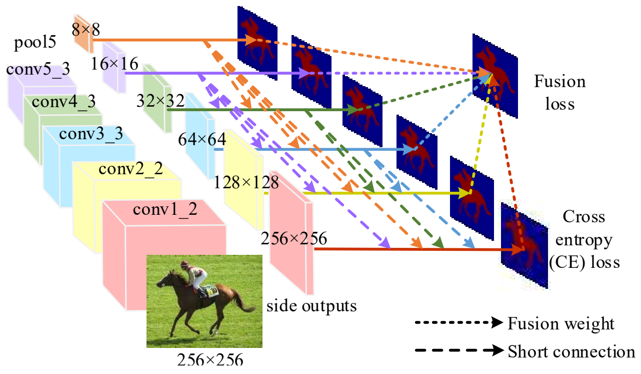


Figure 3: The proposed network architecture. The architecture is based on VGGNet [45] for better comparison with previous CNN-based methods.

where $\hat{A}_{\text{side}}^{(m)}$ is the new activations of the mth side output, $\hat{M} = M+1$, and $\hat{\sigma}(\cdot,\cdot)$ represents the distance between the ground truth map and the new fused predictions, which has the same form to Eqn. (5).

A result comparison between the original HED and enhanced HED for salient object detection can be found in Table 4. Despite a small improvement, as shown in Fig. 1, the saliency maps from shallower side outputs still look messy and the deeper side outputs usually produce irregular results. In addition, the deep side outputs can indeed locate the salient objects/regions, some detailed information is still lost.

3.2. Short connections

Our approach is based on the observation 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. Mathematically, our new side activations $\tilde{R}_{\text{side}}^{(m)}$ at the mth side output can be given by

$$\tilde{R}_{\text{side}}^{(m)} = \begin{cases} \sum_{i=m+1}^{\hat{M}} r_i^m \tilde{R}_{\text{side}}^{(i)} + \hat{A}_{\text{side}}^{(m)}, & \text{for } m = 1,\dots,5 \\ \hat{A}_{\text{side}}^{(m)}, & \text{for } m = 6 \end{cases} \tag{7}$$

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

$$\tilde{L}_{\text{side}}(\mathbf{W}, \tilde{\mathbf{w}}, \mathbf{r}) = \sum_{m=1}^{\hat{M}} \alpha_m \tilde{l}_{\text{side}}^{(m)}(\mathbf{W}, \tilde{\mathbf{w}}^{(m)}, \mathbf{r}) \tag{8}$$

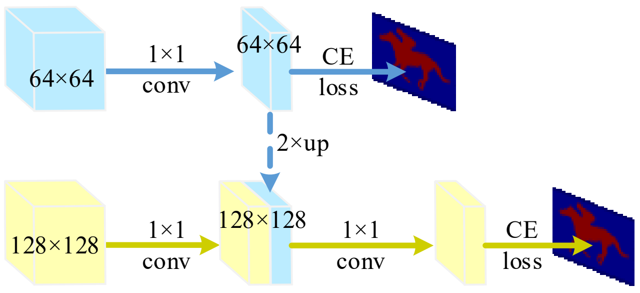


Figure 4: Illustration of short connections in Fig. 3.

No.	Layer	1	2	3
1	conv1_2	$128, 3\times3$	$128, 3\times3$	$1, 1\times1$
2	conv2_2	$128, 3\times3$	$128, 3\times3$	$1, 1\times1$
3	conv3_3	$256, 5\times5$	$256, 5\times5$	$1, 1\times1$
4	conv4_3	$256, 5\times5$	$256, 5\times5$	$1, 1\times1$
5	conv5_3	$512, 5\times5$	$512, 5\times5$	$1, 1\times1$
6	pool5	$512, 7\times7$	$512, 7\times7$	$1, 1\times1$

Table 1: 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.)

and

$$\tilde{L}_{\text{fuse}}(\mathbf{W}, \tilde{\mathbf{w}}, \mathbf{f}, \mathbf{r}) = \hat{\sigma}\big(Z, \textstyle\sum_{m=1}^{M} f_m \tilde{R}_{\text{side}}^{(m)}\big), \tag{9}$$

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. (5). Thus, our new final loss function can be written by

$$\tilde{L}_{\text{final}}(\mathbf{W}, \tilde{\mathbf{w}}, \mathbf{f}, \mathbf{r}) = \tilde{L}_{\text{fuse}}(\mathbf{W}, \tilde{\mathbf{w}}, \mathbf{f}, \mathbf{r}) + \tilde{L}_{\text{side}}(\mathbf{W}, \tilde{\mathbf{w}}, \mathbf{r}). \tag{10}$$

Our architecture can be considered as two closely connected stages from a functional standpoint, 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 for 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/regions and refine the results from deeper side outputs, resulting in dense and accurate saliency maps.

3.3. Inference

The architecture we use in this work can be found in Fig. 3. The illustration of how to build a short connection can be seen in Fig. 4. Although a series of short connections are introduced, the quality of the prediction maps produced by the deepest and the shallowest side outputs is unsatisfactory. Regarding this fact, we only fuse three side outputs during inference while throwing away the other three side outputs by directly setting f_1, f_5, and f_6 to 0. Let $\tilde{Z}_1, \cdots, \tilde{Z}_6$ denote the side output maps. They can be computed by $\tilde{Z}_m = h(\tilde{R}^{(m)}_{\text{side}})$. Therefore, the fusion output map and the final output map can be computed by

$$\tilde{Z}_{\text{fuse}} = h\Big(\sum_{m=2}^{4} f_m \tilde{R}^{(m)}_{\text{side}}\Big), \tag{11}$$

and

$$\tilde{Z}_{\text{final}} = \text{Mean}(\tilde{Z}_{\text{fuse}}, \tilde{Z}_2, \hat{Z}_3, \hat{Z}_4). \tag{12}$$

Smoothing method. Though our DCNN model can precisely find the salient objects/regions in an image, the saliency maps obtained are quite smooth and some useful boundary information is lost. To improve spatial coherence and quality of our saliency maps, we adopt the fully connected conditional random field (CRF) method [26] as a selective layer during the inference phase.

The energy function of CRF is given by

$$E(\mathbf{x}) = \sum_i \theta_i(x_i) + \sum_{i,j} \theta_{ij}(x_i, x_j), \tag{13}$$

where $\mathbf{x}$ 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

$$\theta_i(x_i) = -\frac{\log \hat{S}_i}{\tau h(x_i)}, \tag{14}$$

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

$$\begin{aligned}\theta_{ij}(x_i, x_j) = \mu(x_i, x_j)\Bigg[w_1 \exp\bigg(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \\ \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\bigg) + w_2 \exp\bigg(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\bigg)\Bigg],\end{aligned} \tag{15}$$

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 [26], called PyDenseCRF [1]. 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.

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

4. Experiments and Results

In this section, we describe implementation details of our proposed architecture, introduce utilized datasets and evaluation criteria, and report the performance of our proposed approach.

4.1. Implementation

Our network is based on the publicly available Caffe library [23] and the open implementation of FCN [38]. As mentioned above, we choose VGGNet [45] as our pre-trained model for better comparison with previous works. We introduce short connections to the skip-layer structures within the HED network, which can be directly implemented using the split layer in Caffe.

Parameters. The hyper-parameters used in this work contain: 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 each time only one image is loaded. Taking training efficiency into consideration, each image is trained for ten times, i.e., the "iter_size" parameter is set to 10 in Caffe. 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.

Running time. It takes us about 8 hours to train our network on a single NVIDIA TITAN X GPU and a 4.0GHz Intel processor. Since there does not exist any other pre- and post-processing procedures, it takes only about 0.08s for our model to process an image of size 400×300 and another 0.4s is needed for our CRF. Therefore, our approach uses less than 0.5s to produce the final saliency map, which is much faster than most present CNN-based methods.

4.2. Datasets and evaluation metrics

Datasets. A good saliency detection model should perform well over almost all datasets [1]. To this end, we evaluate our system on 5 representative datasets, including MSRA-B [37], ECSSD [51], HKU-IS [29], PASCALS [34], and SOD [39, 40], all of which are available online. These datasets all contain a large number of images and have been widely used recently. MSRA-B contains 5000 images from hundreds of different categories. Most images in this dataset have only one salient object. Because of its diversity and large quantity, MSRA-B has been one of the

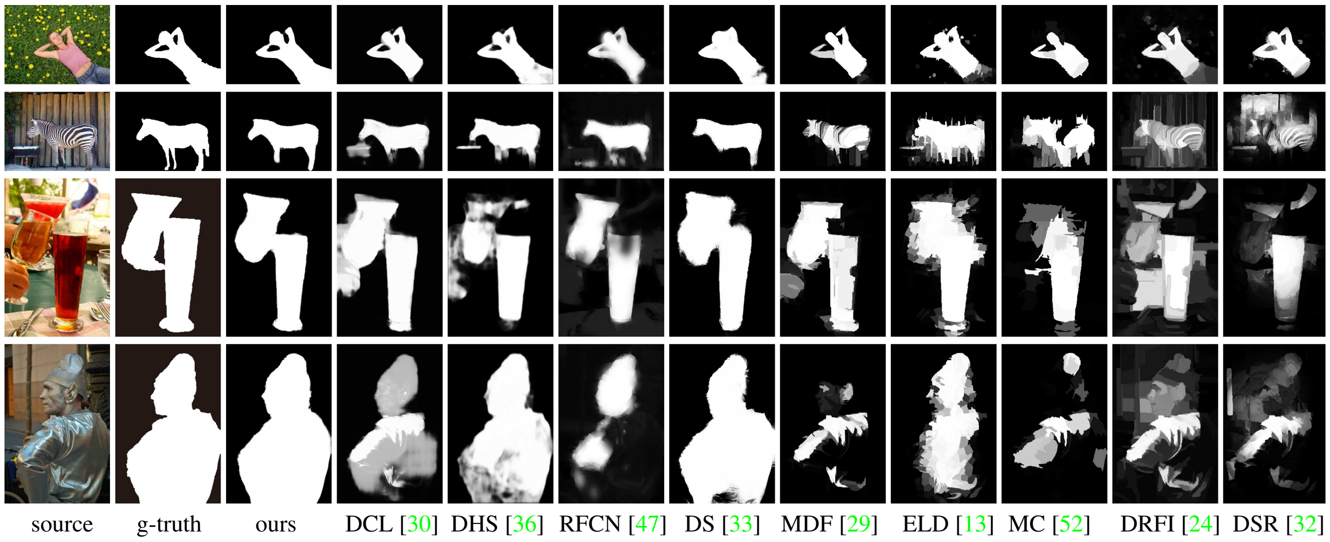


Figure 5: Visual comparison with nine existing methods. As can be seen, our proposed method produces more coherent and accurate saliency maps than all other methods, which are the closest to the ground truth.

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

Table 2: Comparisons of different side output settings and their performance on PASCALS dataset [34]. $(c, k \times k) \times n$ 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 Table 1. In each setting, we only modify one parameter while keeping all others unchanged so as to emphasize the importance of each chosen parameter.

most widely used datasets in salient object detection literature. ECSSD contains 1000 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 selected from the BSDS dataset. It contains 300 images, most of which possess multiple salient objects. All 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, we utilize the same training and validation sets as in [24] and test over all of the datasets using the same model.

Evaluation metrics. We use three universally-agreed, standard metrics to evaluate our model: precision-recall curves, F-measure, and the mean absolute error (MAE).

For a given continuous saliency map S, we can convert it to a binary mask B using a threshold. Then its precision and recall can be computed by $|B \cap Z|/|B|$ and $|B \cap Z|/|Z|$, 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

$$F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 Precision + Recall}. \tag{16}$$

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 MAE score can be computed by

$$MAE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |\hat{S}(i,j) = \hat{Z}(i,j)|. \tag{17}$$

As stated in [1], this metric favors methods that successfully detect salient pixels but fail to detect non-salient regions over methods that successfully detect non-salient pixels but make mistakes in determining the salient ones.

Methods \ Datasets	MSRA-B [37]		ECSSD [51]		HKU-IS [29]		PASCALS [34]		SOD [39, 40]	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
RC [7]	0.817	0.138	0.741	0.187	0.726	0.165	0.640	0.225	0.657	0.242
CHM [31]	0.809	0.138	0.722	0.195	0.728	0.158	0.631	0.222	0.655	0.249
DSR [32]	0.812	0.119	0.737	0.173	0.735	0.140	0.646	0.204	0.655	0.234
DRFI [24]	0.855	0.119	0.787	0.166	0.783	0.143	0.679	0.221	0.712	0.215
MC [52]	0.872	0.062	0.822	0.107	0.781	0.098	0.721	0.147	0.708	0.184
ELD [13]	0.914	0.042	0.865	0.981	0.844	0.071	0.767	0.121	0.760	0.154
MDF [29]	0.885	0.104	0.833	0.108	0.860	0.129	0.764	0.145	0.785	0.155
DS [13]	-	-	0.810	0.160	-	-	0.818	0.170	0.781	0.150
RFCN [47]	0.926	0.062	0.898	0.097	0.895	0.079	0.827	0.118	0.805	0.161
DHS [36]	-	-	0.905	0.061	0.892	0.052	0.820	0.091	0.823	0.127
DCL [30]	0.916	0.047	0.898	0.071	0.907	0.048	0.822	0.108	0.832	0.126
Ours	0.927	0.028	0.915	0.052	0.913	0.039	0.830	0.080	0.842	0.118

Table 3: Quantitative comparisons with 11 methods on 5 popular datasets. The top three results are highlighted in red, green, and blue, respectively.

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.

Details of side-output layers. The detailed information of each side-output layer has been shown in Table 1. We would like to emphasize that introducing another convolutional layer in each side output as described in Sec. 3.1 is extremely important. Besides, we also perform a series of experiments with respect to the parameters of the convolutional layers in each side output. The side-output settings can be found in Table 2. To highlight the importance of different parameters, we adopt the variable-controlling method that only changes one parameter at a time. It can be shown that reducing convolutional layers (#2) decreases the performance but not too much. It can be observed that reducing the kernel size (#3) also leads to a slight decrease in F-measure. Moreover, doubling the number of channels in the last three convolutional layers (#1) does not bring us any improvement.

Comparisons of various short connection patterns. 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 [17] and the HED-based architecture [49], we implement three representative patterns using our proposed approach. The first one is formulated as follows, which is a similar architecture to Fig. 1(c).

$$\tilde{R}_{\text{side}}^{(m)} = \begin{cases} r_{m+1}^{m} \tilde{R}_{\text{side}}^{(m+1)} + \hat{A}_{\text{side}}^{(m)}, & \text{for } m = 1, \ldots, 5 \\ \hat{A}_{\text{side}}^{(m)}. & \text{for } m = 6 \end{cases} \tag{18}$$

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

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

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

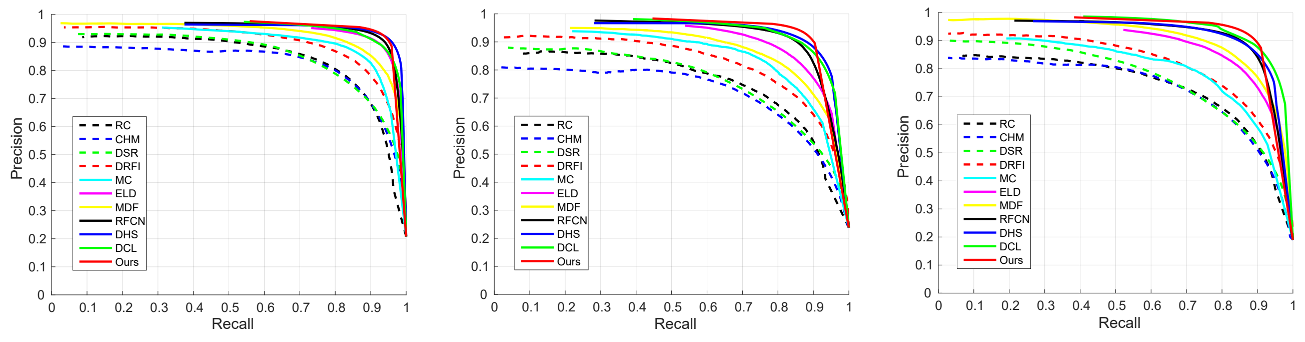
$$\tilde{R}_{\text{side}}^{(m)} = \begin{cases} \sum_{i=3}^{6} r_i^m \tilde{R}_{\text{side}}^{(i)} + \hat{A}_{\text{side}}^{(m)}, & \text{for } m = 1,2 \\ r_5^m \tilde{R}_{\text{side}}^{(5)} + r_6^m \tilde{R}_{\text{side}}^{(6)} + \hat{A}_{\text{side}}^{(m)}, & \text{for } m = 3,4 \\ \hat{A}_{\text{side}}^{(m)}. & \text{for } m = 5,6 \end{cases} \tag{20}$$

No.	Architecture	F_β
1	Hypercolumns [17]	0.818
2	Original HED [49]	0.791
3	Enhanced HED	0.816
4	Pattern 1 (Eqn. (18))	0.816
5	Pattern 2 (Eqn. (19))	0.824
6	Pattern 3* (Eqn. (20))	0.830

Table 4: The performance of different architectures on PASCALS dataset [34]. "*" represents the pattern used in this paper.

The performance is listed in Table 4. As can be seen from Table 4, with the increase of short connections, our approach gradually achieves better performance.

Upsampling operation. In our approach, we use the in-network bilinear interpolation to perform upsampling in each side output. As implemented in [38], 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 map in deep side outputs more denser. We adopt the same technique as done



(1) MSRA (2) ECSSD (3) HKUIS

Figure 6: Precision recall curves on three popular datasets.

in [30]. However, in our experiments, using such a method yields a worse performance. Albeit 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 more than 1%.

Data augmentation. Data augmentation has been proven to be very useful in many learning-based vision tasks. 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%.

4.4. Comparison with the state-of-the-art

We compare the proposed approach with 7 recent CNN-based methods, including MDF [29], DS [33], DCL [30], ELD [13], MC [52], RFCN [47], and DHS [36]. We also compare our approach with 4 classical methods: RC [7], CHM [31], DSR [32], and DRFI [24], which have been proven to be the best in the benchmark study of Borji *et al.* [1].

Visual comparison. Fig. 5 provides a visual comparison of our approach with respect to the above-mentioned approaches. It can be easily seen that our proposed method not only highlights the right salient region 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. It also generates connected regions. These advantages make our results very close to the ground truth and hence better than other methods.

PR curve. We compare our approach with the existing methods in terms of PR curve. As can be seen in Fig. 6, the proposed approach achieves a better PR curve than all the other methods. Because of the refinement effect of low-level features, our saliency maps look much closer to the ground truth. This also causes our precision value to be higher, thus resulting in a higher PR curve.

F-measure and MAE. We also compare our approach with the existing methods in terms of F-meature and MAE scores. F-measure and MAE of methods are shown in Table 3. As can be seen, our approach achieves the best score (maximum F-measure and MAE) over all datasets as listed in Table 3. Our approach improves the current best maximum F-measure by 1 percent.

Besides, we also observe that the proposed approach behaves even better over more difficult datasets, such as HKUIS [29], PASCALS [34], and SOD [39, 40], which contain a large number 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.

5. Conclusion

In this paper, we developed 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 We would like to thank the anonymous reviewers for their useful feedbacks. This research was supported by NSFC (NO. 61572264, 61620106008), Huawei Innovation Research Program (HIRP), and CAST young talents plan.

References

[1] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE TIP*, 24(12):5706–5722, 2015. 1, 2, 5, 6, 8

[2] A. Borji, S. Frintrop, D. N. Sihite, and L. Itti. Adaptive object tracking by learning background context. In *IEEE CVPRW*, pages 23–30. IEEE, 2012. 1

[3] A. Borji and L. Itti. Exploiting local and global patch rarities for saliency detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 478–485. IEEE, 2012. 2

[4] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207, 2013. 1

[5] A. Borji, D. N. Sihite, and L. Itti. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, 22(1):55–69, 2013. 1

[6] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: Internet image montage. *ACM TOG*, 28(5):124:1–10, 2009. 1

[7] M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S. Hu. Global contrast based salient region detection. *IEEE TPAMI*, 2015. 1, 2, 7, 8

[8] M.-M. Cheng, Q.-B. Hou, S.-H. Zhang, and P. L. Rosin. Intelligent visual media processing: When graphics meets vision. *JCST*, 2017. 1

[9] 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 TOG*, volume 29, page 83. ACM, 2010. 1

[10] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof. Saliency driven total variation segmentation. In *ICCV*, pages 817–824. IEEE, 2009. 1

[11] W. Einhäuser and P. König. Does luminance-contrast contribute to a saliency map for overt visual attention? *European Journal of Neuroscience*, 17(5):1089–1097, 2003. 1

[12] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai. 3-D object retrieval and recognition with hypergraph analysis. *IEEE TIP*, 21(9):4290–4303, 2012. 1

[13] L. Gayoung, T. Yu-Wing, and K. Junmo. Deep saliency with encoded low level distance map and high level features. In *CVPR*, 2016. `https://github.com/gylee1103/SaliencyELD`. 1, 2, 6, 7, 8

[14] C. Goldberg, T. Chen, F.-L. Zhang, A. Shamir, and S.-M. Hu. Data-driven object manipulation in images. 31(21):265–274, 2012. 1

[15] C. Guo and L. Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE TIP*, 19(1):185–198, 2010. 1

[16] J. Han, E. J. Pauwels, and P. De Zeeuw. Fast saliency-aware multi-modality image fusion. *Neurocomputing*, pages 70–80, 2013. 1

[17] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, pages 447–456, 2015. 3, 7

[18] J. He, J. Feng, X. Liu, T. Cheng, T.-H. Lin, H. Chung, and S.-F. Chang. Mobile product search with bag of hash bits and boundary reranking. In *IEEE CVPR*, pages 3005–3012, 2012. 1

[19] 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 Visual Computer*, 29(5):393–405, 2013. 1

[20] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE TIP*, 13(10):1304–1318, 2004. 1

[21] L. Itti and C. Koch. Computational modeling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001. 1

[22] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, (11):1254–1259, 1998. 1, 2

[23] 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 Multimedia*, pages 675–678. ACM, 2014. 5

[24] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, pages 2083–2090, 2013. `http://people.cs.umass.edu/~hzjiang/`. 1, 2, 6, 7, 8

[25] D. A. Klein and S. Frintrop. Center-surround divergence of feature statistics for salient object detection. In *ICCV*, pages 2214–2219. IEEE, 2011. 2

[26] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 5

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[29] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *CVPR*, pages 5455–5463, 2015. `http://i.cs.hku.hk/~yzyu/vision.html`. 1, 2, 5, 6, 7, 8

[30] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *CVPR*, 2016. 2, 6, 7, 8

[31] X. Li, Y. Li, C. Shen, A. Dick, and A. Van Den Hengel. Contextual hypergraph modeling for salient object detection. In *ICCV*, pages 3328–3335, 2013. 7, 8

[32] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang. Saliency detection via dense and sparse reconstruction. In *ICCV*, pages 2976–2983, 2013. 6, 7, 8

[33] 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 Transactions on Image Processing*, 25(8):3919 – 3930, 2016. `https://github.com/zlmzju/DeepSaliency`. 6, 8

[34] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014. 5, 6, 7, 8

[35] H. Liu, L. Zhang, and H. Huang. Web-image driven best views of 3d shapes. *The Visual Computer*, pages 1–9, 2012. 1

[36] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, 2016. 1, 2, 6, 7, 8

[37] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *IEEE TPAMI*, 33(2):353–367, 2011. 1, 2, 5, 7

[38] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1, 5, 7

[39] 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 *ICCV*, volume 2, pages 416–423, 2001. 5, 7, 8

[40] V. Movahedi and J. H. Elder. Design and perceptual validation of performance measures for salient object segmentation. In *IEEE CVPRW*, pages 49–56. IEEE, 2010. 5, 7, 8

[41] D. Parkhurst, K. Law, and E. Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision research*, 42(1):107–123, 2002. 1

[42] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740. IEEE, 2012. 2

[43] P. L. Rosin and Y.-K. Lai. Artistic minimal rendering with lines and blocks. *Graphical Models*, 75(4):208–229, 2013. 1

[44] U. Rutishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition? In *CVPR*, 2004. 1

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 2, 3, 4, 5

[46] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, pages 3183–3192, 2015. 1, 2

[47] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, 2016. `http://202.118.75.4/lu/publications.html`. 6, 7, 8

[48] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE TPAMI*, 2016. 1

[49] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, pages 1395–1403, 2015. 1, 2, 3, 7

[50] Y. Xie, H. Lu, and M.-H. Yang. Bayesian saliency via low and mid level cues. *IEEE TIP*, 22(5):1689–1698, 2013. 2

[51] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013. 5, 7

[52] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *CVPR*, pages 1265–1274, 2015. `https://github.com/Robert0812/deepsaldet`. 1, 2, 6, 7, 8

[53] J.-Y. Zhu, J. Wu, Y. Wei, E. Chang, and Z. Tu. Unsupervised object class discovery via saliency-guided multiple class learning. In *IEEE CVPR*, pages 3218–3225, 2012. 1