Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

Salient Object Detection by Lossless Feature Reflection

Pingping Zhang[1,3], **Wei Liu**[2,3], **Huchuan Lu**[1], **Chunhua Shen**[3]
[1] Dalian University of Technology, Dalian, 116024, P.R. China
[2] Shanghai Jiao Tong University, Shanghai, 200240, P.R. China
[3] University of Adelaide, Adelaide, SA 5005, Australia
jssxzhpp@gmail.com; liuwei.1989@sjtu.edu.cn; lhchuan@dlut.edu.cn; chunhua.shen@adelaide.edu.au

Abstract

Salient object detection, which aims to identify and locate the most salient pixels or regions in images, has been attracting more and more interest due to its various real-world applications. However, this vision task is quite challenging, especially under complex image scenes. Inspired by the intrinsic reflection of natural images, in this paper we propose a novel feature learning framework for large-scale salient object detection. Specifically, we design a symmetrical fully convolutional network (SFCN) to learn complementary saliency features under the guidance of lossless feature reflection. The location information, together with contextual and semantic information, of salient objects are jointly utilized to supervise the proposed network for more accurate saliency predictions. In addition, to overcome the blurry boundary problem, we propose a new structural loss function to learn clear object boundaries and spatially consistent saliency. The coarse prediction results are effectively refined by these structural information for performance improvements. Extensive experiments on seven saliency detection datasets demonstrate that our approach achieves consistently superior performance and outperforms the very recent state-of-the-art methods.

1 Introduction

As a fundamental yet challenging task in computer vision, salient object detection (SOD) aims to identify and locate distinctive objects or regions which attract human attention in natural images. In general, SOD is regarded as a prerequisite step to narrow down subsequent object-related vision tasks. For example, it can be used in image retrieval, sematic segmentation, visual tracking and person re-identification, etc.

In the past two decades, a large number of SOD methods have been proposed. Most of them have been well summarized in [Borji *et al.*, 2015]. According to that work, conventional SOD methods focus on extracting discriminative local and global handcrafted features from pixels or regions to represent their visual properties. With several heuristic priors, these methods predict salient scores according to the extracted features for saliency detection. Although great success has been made, there still exist many important problems which need to be solved. For example, the low-level handcrafted features suffer from limited representation capability, and are difficult to capture the semantic and structural information of objects in images, which is very important for more accurate SOD. What's more, to further extract powerful and robust visual features manually is a tough mission.

With the recent prevalence of deep architectures, many remarkable progresses have been achieved in a wide range of computer vision tasks, *e.g.*, image classification [Simonyan and Zisserman, 2014] and semantic segmentation [Long *et al.*, 2015]. Thus, many researchers start to make their great efforts to utilize deep convolutional neural networks (CNNs) for SOD and have achieved favourable performance, since CNNs have strong ability to automatically extract high-level feature representations, successfully avoiding the drawbacks of handcrafted features. However, most of state-of-the-art SOD methods still require large-scale pre-trained CNNs, which usually employ the strided convolution and pooling operations. These downsampling methods increase the receptive field of CNNs, helping to extract high-level semantic features, nevertheless they inevitably drop the location information and fine details of objects, leading to unclear boundary predictions. Furthermore, the lack of structural supervision also makes SOD an extremely challenging problem in complex image scenes.

In order to utilize the semantic and structural information derived from deep pre-trained CNNs, we propose to solve both tasks of complementary feature extraction and saliency region classification with an unified framework which is learned in the end-to-end manner. Specifically, we design a symmetrical fully convolutional network (SFCN) architecture which consists of two sibling branches and one fusing branch, as illustrated in Fig. 1. The two sibling branches take reciprocal image pairs as inputs and share weights for learning complementary visual features under the guidance of lossless feature reflection. The fusing branch integrates the multi-level complementary features in a hierarchical manner for SOD. More importantly, to effectively train our network, we propose a novel loss function which incorporates structural information and supervises the three branches during the training process. In this manner, our proposed model sufficiently captures the boundaries and spatial contexts of salient objects, hence significantly boosts the performance of SOD.

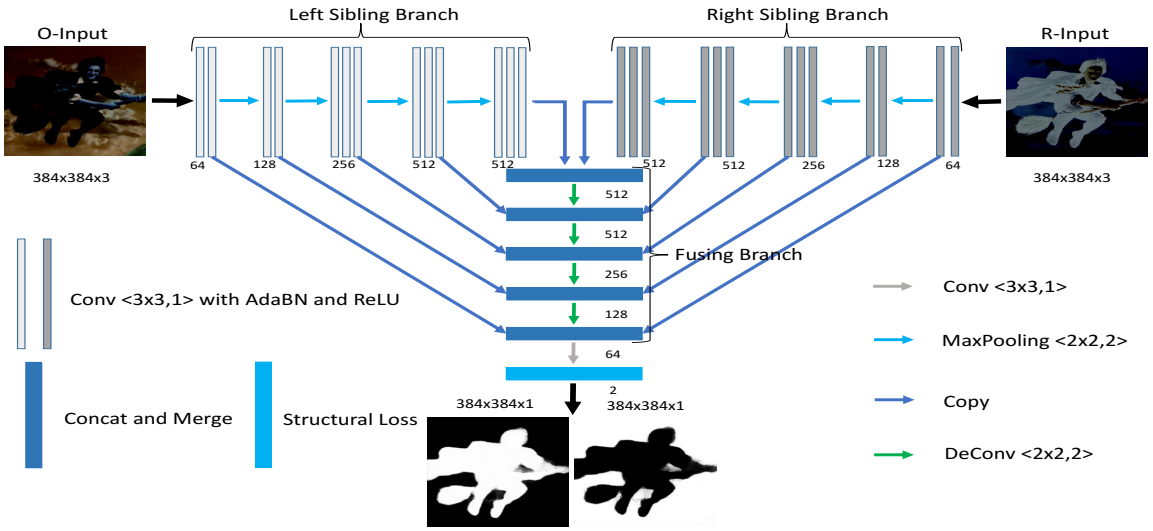


Figure 1: The semantic overview of our proposed SFCN.

In summary, **our contributions** are three folds:

- We present a novel network architecture, *i.e.*, SFCN, which is symmetrically designed to learn complementary visual features and predict accurate saliency maps under the guidance of lossless feature reflection.
- We propose a new structural loss function to learn clear object boundaries and spatially consistent saliency. This loss function is able to utilize the location, contextual and semantic information of salient objects to supervise the proposed SFCN for performance improvements.
- Extensive experiments on seven large-scale saliency benchmarks demonstrate that the proposed approach achieves superior performance and outperforms the very recent state-of-the-art methods by a large margin.

2 Related Work

Salient Object Detection. Recent years, deep learning based methods have achieved solid performance improvements in SOD. For example, [Wang *et al.*, 2015] integrate both local pixel estimation and global proposal search for SOD by training two deep neural networks. [Zhao *et al.*, 2015] propose a multi-context deep CNN framework to benefit from the local context and global context of salient objects. [Li and Yu, 2015] employ multiple deep CNNs to extract multi-scale features for saliency prediction. Then they propose a deep contrast network to combine a pixel-level stream and segment-wise stream for saliency estimation [Li and Yu, 2016]. Inspired by the great success of fully convolutional networks (FCNs) [Long *et al.*, 2015], [Wang *et al.*, 2016] develop a recurrent FCN to incorporate saliency priors for more accurate saliency map inference. [Liu and Han, 2016] also design a deep hierarchical network to learn a coarse global estimation and then refine the saliency map hierarchically and progressively. Then, [Hou *et al.*, 2017] introduce dense short connections to the skip-layers within the holistically-nested edge detection (HED) architecture [Xie and Tu, 2015] to get rich multi-scale features for SOD. [Zhang *et al.*, 2017a] propose a bidirectional learning framework to aggregate multi-level convolutional features for SOD. And they also develop a novel dropout to learn the deep uncertain convolutional features to enhance the robustness and accuracy of saliency detection [Zhang *et al.*, 2017b]. [Wang *et al.*, 2017b] provide a stage-wise refinement framework to gradually get accurate saliency detection results. Despite these approaches employ powerful CNNs and make remarkable success in SOD, there still exist some obvious problems. For example, the strategies of multiple-stage training reduce the efficiency. And the explicit pixel-wise loss functions used by these methods for model training cannot well reflect the structural information of salient objects. Hence, there is still a large space for performance improvements.

Image Intrinsic Reflection. Image intrinsic reflection is a classical topic in computer vision field. It aims to separate a color image into two intrinsic reflection images: an image of just the highlights, and the original image with the highlights removed. It can be used to segment and analyze surfaces with image color variations. Most of existing methods are based on the Retinex model [Land and McCann, 1971]. Recent years, researchers have augmented the basic Retinex model with non-local texture cues and sparsity priors. Sophisticated techniques that recover reflectance and shading along with a shape estimate have also been proposed. Inspired by these works, we construct a reciprocal image pair based on the input image (see Section 3.1). However, there are three obvious differences between our method and previous intrinsic reflection methods: 1) the objective is different. The aim of previous methods is explaining an input RGB image by estimating albedo and shading fields. Our aim is to learn complementary visual features for SOD. 2) the resulting image pair is different. Image intrinsic reflection methods usually factory an input image into a reflectance image and a shading image, while our method builds a reciprocal image pair for each image as the input of deep networks. 3) the source of reflection is different. The source of previous intrinsic reflection methods is the albedo of depicted surfaces, while our reflection is originated from deep features in CNNs. Therefore, our reflection is feature-level not image-level.

3 The Proposed Method

Fig. 1 illustrates the semantic overview of our method. In the following subsections, we elaborate the proposed SFCN architecture and the weighted structural loss.

3.1 Symmetrical FCN

The proposed SFCN is an end-to-end fully convolutional network. It consists of three main branches with a paired reciprocal image input to achieve lossless feature reflection learning.

Reciprocal Image Input. To capture complementary image information, we first convert the given RGB image $X \in R^{W\times H\times 3}$ to a reciprocal image pair by the following reflection function,

$$\begin{aligned} Rec(X,k) &= (X-M, k(M-X))), && (1)\\ &= (X-M, -k(X-M)) && (2)\\ &= (X_O, X_R^k). && (3)\end{aligned}$$

where k is a hyperparameter to control the reflection scale and $M \in R^{W\times H\times 3}$ is the mean of an image or image dataset. From above equations, one can see that the converted image pair, *i.e.*, X_O and X_R^k, is reciprocal with a reflection plane. In detail, the reflection scheme is a pixel-wise negation operator, allowing the given images to be reflected in both positive and negative directions while maintaining the same content of images. In the proposed reflection, we use the multiplicative operator to measure the reflection scale, but it is not the only feasible method. For example, this reflection can be combined with other non-linear operators, such as quadratic form, to add more diversity. To reduce the computation burden, we use $k = 1$ and the well-known mean of the ImageNet dataset.

Sibling Branches with AdaBN. Based on the reciprocal image pair, we propose two sibling branches to extract complementary reflection features. More specifically, we build each sibling branch, following the VGG-16 model [Simonyan and Zisserman, 2014]. Each sibling branch has 13 convolutional layers (kernel size = 3×3, stride size = 1) and 4 max pooling layers (pooling size = 2×2, stride = 2). To achieve the lossless reflection features, the two sibling branches are designed to share weights in convolutional layers, but with adaptive batch normalization (AdaBN). In other words, we keep the weights of corresponding convolutional layers of the two sibling branches the same, while use different learnable BN between the convolution and ReLU operators [Zhang *et al.*, 2017a]. The main reason of this design is that after the reflection transform, the reciprocal images have different image domains. Domain related knowledge heavily affects the statistics of BN layers. In order to learn domain invariant features, it's beneficial for each domain to keep its own BN statistics in each layers.

Hierarchical Feature Fusion. After extracting multi-level reflection features, we adhere an additional fusing branch to integrate them for the saliency prediction. In order to preserve the spatial structure and enhance the contextual information, we integrate the multi-level reflection features in a hierarchical manner. Formally, the fusing function is defined by

$$f_l(X) = \begin{cases} h([g_l(X_O), f_{l+1}(X), g_l^*(X_R^k)]), l < L \\ h([g_l(X_O), g_l^*(X_R^k)]), l = L \end{cases} \quad (4)$$

where h denotes the integration operator, which is a 1×1 convolutional layer followed by a deconvolutional layer to ensure the same resolution. $[\cdot]$ is the concatenation operator in channel-wise. g_l and g_l^* are the reflection features of the l-th convolutional layer in the two sibling branches, respectively.

In the end, we add a convolutional layer with two filters for the saliency map prediction. The numbers in Fig. 1 illustrate the detailed filter setting in each convolutional layer.

3.2 Weighted Structural Loss

Given the SOD training dataset $S = \{(X_n, Y_n)\}_{n=1}^N$ with N training pairs, where $X_n = \{x_i^n, i = 1, ..., T\}$ and $Y_n = \{y_i^n, i = 1, ..., T\}$ are the input image and the binary ground-truth image with T pixels, respectively. $y_i^n = 1$ denotes the foreground pixel and $y_i^n = 0$ denotes the background pixel. For notional simplicity, we subsequently drop the subscript n and consider each image independently. In most of existing SOD methods, the loss function used to train the network is the standard pixel-wise binary cross-entropy (BCE) loss:

$$\begin{aligned} \mathcal{L}_{bce} = &- \sum_{i\in Y_+} \log \Pr(y_i = 1|X;\theta) \\ &- \sum_{i\in Y_-} \log \Pr(y_i = 0|X;\theta). \end{aligned} \quad (5)$$

where θ is the parameter of the network. $\Pr(y_i = 1|X;\theta) \in [0, 1]$ is the confidence score of the network prediction that measures how likely the pixel belong to the foreground.

However, for a typical natural image, the class distribution of salient/non-salient pixels is heavily imbalanced: most of the pixels in the ground truth are non-salient. To automatically balance the loss between positive/negative classes, we introduce a class-balancing weight β on a per-pixel term basis, following [Xie and Tu, 2015]. Specifically, we define the following weighted cross-entropy loss function,

$$\begin{aligned} \mathcal{L}_{wbce} = &-\beta \sum_{i\in Y_+} \log \Pr(y_i = 1|X;\theta) \\ &-(1-\beta) \sum_{i\in Y_-} \log \Pr(y_i = 0|X;\theta). \end{aligned} \quad (6)$$

The loss weight $\beta = |Y_+|/|Y|$, and $|Y_+|$ and $|Y_-|$ denote the foreground and background pixel number, respectively.

For saliency detection, it is also crucial to preserve the overall spatial structure and semantic content. Thus, we also minimize the differences between their multi-level features by a deep convolutional network [Johnson *et al.*, 2016]. The main intuition behind this operator is that minimizing the difference between multi-level features, which encode low-level fine details and high-level coarse semantics, helps to retain the spatial structure and semantic content of predictions. Formally, let ϕ_l denotes the output of the l-th convolutional layer in a CNN, our semantic content (SC) loss is defined as

$$\mathcal{L}_{sc} = \sum_{l=1}^{L} \lambda_l ||\phi_l(Y;w) - \phi_l(\hat{Y};w)||_2, \quad (7)$$

where $\hat{Y}$ is the overall prediction, w is the parameter of a pre-trained CNN and λ_l is the trade-off parameter, controlling the

influence of the loss in the l-th layer. In our case, we use the light CNN-9 model [Wu *et al.*, 2015] to calculate the above loss between the ground-truth and the prediction.

To overcome the blurry boundary problem [Li *et al.*, 2016], we also introduce the smooth L_1 loss which encourages to keep the details of boundaries of salient objects. Specifically, the smooth L_1 loss function is defined as

$$\mathcal{L}_{s1} = \begin{cases} \frac{1}{2}||D||_2^2, & ||D||_1 < \epsilon \\ \epsilon||D||_1 - \frac{1}{2}\epsilon^2, & otherwise \end{cases} \tag{8}$$

where $D = Y - \hat{Y}$ and ϵ is a predefined threshold. Following the practice in [Xiao *et al.*, 2018], we set $\epsilon = 0.5$. This training loss also helps to minimize pixel-level differences between the overall prediction and the ground-truth.

By taking all above loss functions together, we define our final loss function as

$$\mathcal{L} = \arg\min \mathcal{L}_{wbce} + \mu\mathcal{L}_{sc} + \gamma\mathcal{L}_{s1}, \tag{9}$$

where μ and γ are hyperparameters to balance the specific terms. All the above losses are continuously differentiable, so we can use the standard stochastic gradient descent (SGD) method to obtain the optimal parameters. In addition, we use $\lambda_l = 1$, $\mu = 0.01$ and $\gamma = 20$ to optimize the final loss function for our experiments without further tuning.

4 Experimental Results

4.1 Datasets and Evaluation Metrics

To train our model, we adopt the **MSRA10K** [Borji *et al.*, 2015] dataset, which has 10,000 training images with high quality pixel-wise saliency annotations. To combat overfitting, we augment this dataset by random cropping and mirror reflection, producing 120,000 training images totally.

For the performance evaluation, we adopt seven public saliency detection datasets as follows: **DUT-OMRON** [Yang *et al.*, 2013] dataset has 5,168 high quality natural images. Each image in this dataset has one or more objects with relatively complex image background. **DUTS-TE** dataset is the test set of currently largest saliency detection benchmark (DUTS) [Wang *et al.*, 2017a]. It contains 5,019 images with high quality pixel-wise annotations. **ECSSD** [Shi *et al.*, 2016] dataset contains 1,000 natural images, in which many semantically meaningful and complex structures are included. **HKU-IS-TE** [Li and Yu, 2015] dataset has 1,447 images with pixel-wise annotations. Images of this dataset are well chosen to include multiple disconnected objects or objects touching the image boundary. **PASCAL-S** [Li *et al.*, 2014] dataset is generated from the PASCAL VOC [Everingham *et al.*, 2010] dataset and contains 850 natural images with segmentation-based masks. **SED** [Borji *et al.*, 2015] dataset has two non-overlapped subsets, *i.e.*, SED1 and SED2. SED1 has 100 images each containing only one salient object, while SED2 has 100 images each containing two salient objects. **SOD** [Jiang *et al.*, 2013] dataset has 300 images, in which many images contain multiple objects either with low contrast or touching the image boundary.

To evaluate the performance of varied SOD algorithms, we adopt four metrics, including the widely used precision-recall (PR) curves, F-measure, mean absolute error (MAE) [Borji *et al.*, 2015] and recently proposed S-measure [Fan *et al.*, 2017]. The PR curve of a specific dataset exhibits the mean precision and recall of saliency maps at different thresholds. The F-measure is a weighted mean of average precision and average recall, calculated by

$$F_\eta = \frac{(1+\eta^2) \times Precision \times Recall}{\eta^2 \times Precision \times Recall}. \tag{10}$$

We set η^2 to be 0.3 to weigh precision more than recall as suggested in [Borji *et al.*, 2015].

For fair comparison on non-salient regions, we also calculate the mean absolute error (MAE) by

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - G(x,y)|, \tag{11}$$

where W and H are the width and height of the input image. $S(x,y)$ and $G(x,y)$ are the pixel values of the saliency map and the binary ground truth at (x,y), respectively.

To evaluate the spatial structure similarities of saliency maps, we also calculate the S-measure, defined as

$$S_\lambda = \lambda * S_o + (1-\lambda) * S_r, \tag{12}$$

where $\lambda \in [0,1]$ is the balance parameter. S_o and S_r are the object-aware and region-aware structural similarity, respectively. We set $\lambda = 0.5$ as suggested in [Fan *et al.*, 2017].

4.2 Implementation Details

We implement our model based on the Caffe toolbox [Jia *et al.*, 2014] with the MATLAB 2016 platform. We train and test our method with an NVIDIA Titan 1070 GPU (8G memory) and an i5-6600 CPU. Following [Zhang *et al.*, 2017a; 2017b], we do not use validation set and train the model until its training loss converges. The input image is uniformly resized into $384 \times 384 \times 3$ pixels and subtracted the ImageNet mean [Deng *et al.*, 2009]. The weights of sibling branches are initialized from the VGG-16 model. For the fusing branch, we initialize the weights by the "msra" method. During the training, we use standard SGD method with batch size 12, momentum 0.9 and weight decay 0.0005. We set the base learning rate to 1e-8 and decrease the learning rate by 10% when training loss reaches a flat. The training process converges after 150k iterations. When testing, our proposed SOD algorithm runs at about **12 fps**. The source code is publicly available at http://ice.dlut.edu.cn/lu/.

4.3 Comparison with the State-of-the-arts

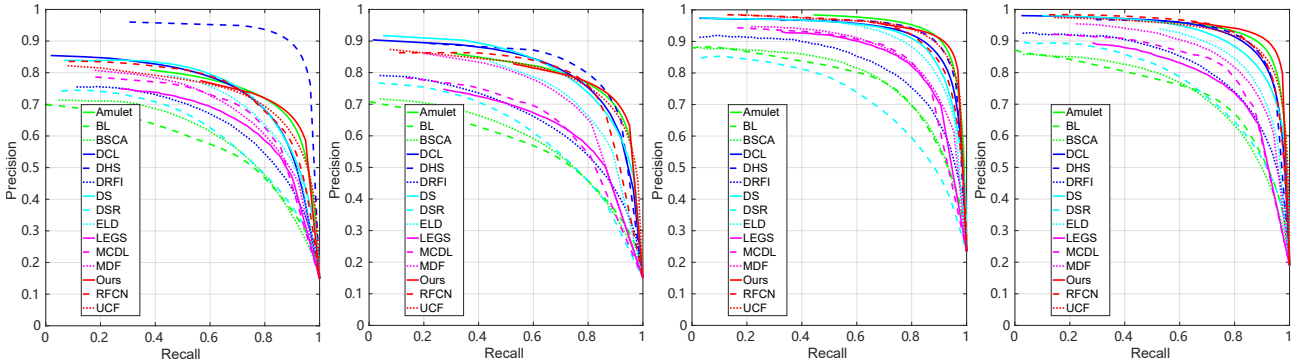
To fully evaluate the detection performance, we compare our proposed method with other 14 state-of-the-art ones, including 10 deep learning based algorithms (**Amulet** [Zhang *et al.*, 2017a], **DCL** [Li and Yu, 2016], **DHS** [Liu and Han, 2016], **DS** [Li *et al.*, 2016], **ELD** [Lee *et al.*, 2016], **LEGS** [Wang *et al.*, 2015], **MCDL** [Zhao *et al.*, 2015], **MDF** [Li and Yu, 2015], **RFCN** [Wang *et al.*, 2016], **UCF** [Zhang *et al.*, 2017b]) and 4 conventional algorithms (**BL** [Tong *et al.*,

	DUT-OMRON			DUTS-TE			ECSSD			HKU-IS-TE		
Methods	F_η	MAE	S_λ	F_η	MAE	S_λ	F_η	MAE	S_λ	F_η	MAE	S_λ
Ours	0.696	0.086	0.774	0.716	0.083	0.799	0.880	0.052	0.897	0.875	0.040	0.905
Amulet [Zhang *et al.*, 2017a]	0.647	0.098	0.771	0.682	0.085	0.796	0.868	0.059	0.894	0.843	0.050	0.886
DCL [Li and Yu, 2016]	0.684	0.157	0.743	0.714	0.150	0.785	0.829	0.149	0.863	0.853	0.136	0.859
DHS [Liu and Han, 2016]	–	–	–	0.724	0.066	0.809	0.872	0.060	0.884	0.854	0.053	0.869
DS [Li *et al.*, 2016]	0.603	0.120	0.741	0.632	0.091	0.790	0.826	0.122	0.821	0.787	0.077	0.854
ELD [Lee *et al.*, 2016]	0.611	0.092	0.743	0.628	0.098	0.749	0.810	0.080	0.839	0.776	0.072	0.823
LEGS [Wang *et al.*, 2015]	0.592	0.133	0.701	0.585	0.138	0.687	0.785	0.118	0.787	0.732	0.118	0.745
MCDL [Zhao *et al.*, 2015]	0.625	0.089	0.739	0.594	0.105	0.706	0.796	0.101	0.803	0.760	0.091	0.786
MDF [Li and Yu, 2015]	0.644	0.092	0.703	0.673	0.100	0.723	0.807	0.105	0.776	0.802	0.095	0.779
RFCN [Wang *et al.*, 2016]	0.627	0.111	0.752	0.712	0.090	0.784	0.834	0.107	0.852	0.838	0.088	0.860
UCF [Zhang *et al.*, 2017b]	0.621	0.120	0.748	0.635	0.112	0.777	0.844	0.069	0.884	0.823	0.061	0.874
BL [Tong *et al.*, 2015]	0.499	0.239	0.625	0.490	0.238	0.615	0.684	0.216	0.714	0.666	0.207	0.702
BSCA [Qin *et al.*, 2015]	0.509	0.190	0.652	0.500	0.196	0.633	0.705	0.182	0.725	0.658	0.175	0.705
DRFI [Jiang *et al.*, 2013]	0.550	0.138	0.688	0.541	0.175	0.662	0.733	0.164	0.752	0.726	0.145	0.743
DSR [Li *et al.*, 2013]	0.524	0.139	0.660	0.518	0.145	0.646	0.662	0.178	0.731	0.682	0.142	0.701

Table 1: Quantitative comparison with 15 methods on 4 large-scale datasets. The best three results are shown in red, green and blue, respectively. "–" means corresponding methods are trained on that dataset. Our method ranks first or second on these datasets.

	PASCAL-S			SED1			SED2			SOD		
Methods	F_η	MAE	S_λ	F_η	MAE	S_λ	F_η	MAE	S_λ	F_η	MAE	S_λ
Ours	0.772	0.104	0.809	0.913	0.048	0.905	0.871	0.048	0.870	0.789	0.123	0.772
Amulet [Zhang *et al.*, 2017a]	0.768	0.098	0.820	0.892	0.060	0.893	0.830	0.062	0.852	0.745	0.144	0.753
DCL [Li and Yu, 2016]	0.714	0.181	0.791	0.855	0.151	0.845	0.795	0.157	0.760	0.741	0.194	0.748
DHS [Liu and Han, 2016]	0.777	0.095	0.807	0.888	0.055	0.894	0.822	0.080	0.796	0.775	0.129	0.750
DS [Li *et al.*, 2016]	0.659	0.176	0.739	0.845	0.093	0.859	0.754	0.123	0.776	0.698	0.189	0.712
ELD [Lee *et al.*, 2016]	0.718	0.123	0.757	0.872	0.067	0.864	0.759	0.103	0.769	0.712	0.155	0.705
LEGS [Wang *et al.*, 2015]	–	–	–	0.854	0.103	0.828	0.736	0.124	0.716	0.683	0.196	0.657
MCDL [Zhao *et al.*, 2015]	0.691	0.145	0.719	0.878	0.077	0.855	0.757	0.116	0.742	0.677	0.181	0.650
MDF [Li and Yu, 2015]	0.709	0.146	0.692	0.842	0.099	0.833	0.800	0.101	0.772	0.721	0.165	0.674
RFCN [Wang *et al.*, 2016]	0.751	0.132	0.799	0.850	0.117	0.832	0.767	0.113	0.784	0.743	0.170	0.730
UCF [Zhang *et al.*, 2017b]	0.735	0.115	0.806	0.865	0.063	0.896	0.810	0.068	0.846	0.738	0.148	0.762
BL [Tong *et al.*, 2015]	0.574	0.249	0.647	0.780	0.185	0.783	0.713	0.186	0.705	0.580	0.267	0.625
BSCA [Qin *et al.*, 2015]	0.601	0.223	0.652	0.805	0.153	0.785	0.706	0.158	0.714	0.584	0.252	0.621
DRFI [Jiang *et al.*, 2013]	0.618	0.207	0.670	0.807	0.148	0.797	0.745	0.133	0.750	0.634	0.224	0.624
DSR [Li *et al.*, 2013]	0.558	0.215	0.594	0.791	0.158	0.736	0.712	0.141	0.715	0.596	0.234	0.596

Table 2: Quantitative comparison with 15 methods on 4 complex structure image datasets. The best three results are shown in red, green and blue, respectively. "–" means corresponding methods are trained on that dataset. Our method ranks first or second on these datasets.



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

Figure 2: The PR curves of state-of-the-art methods.

Models	(a) SFCN-hf+$\mathcal{L}_{bce}$	(b) SFCN+$\mathcal{L}_{bce}$	(c) SFCN+$\mathcal{L}_{wbce}$	(d) SFCN+$\mathcal{L}_{wbce}$+$\mathcal{L}_{sc}$	(e) SFCN+$\mathcal{L}_{wbce}$+$\mathcal{L}_{s1}$	The overall
F_η	0.824	0.848	0.865	0.873	0.867	0.880
MAE	0.102	0.083	0.072	0.061	0.049	0.052
S_λ	0.833	0.859	0.864	0.880	0.882	0.897

Table 3: Results with different model settings on the ECSSD dataset. The best three results are shown in red, green and blue, respectively.

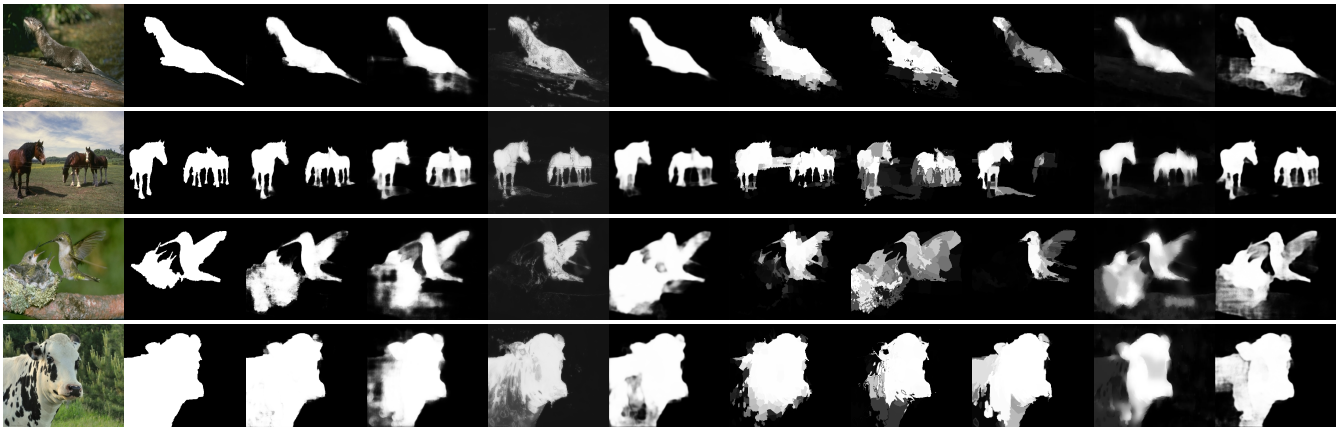


Figure 3: Comparison of typical saliency maps. (a) Input images; (b) Ground truth; (c) **Ours**; (d) **Amulet**; (e) **DCL**; (f) **DHS**; (g) **ELD**; (h) **MCDL**; (i) **MDF**; (j) **RFCN**; (k) **UCF**. Due to the limitation of space, we don't show the results of **DS**, **LEGS**, **BL**, **BSCA**, **DRFI** and **DSR**. We will release the saliency maps of all compared methods upon the acceptance.

2015], **BSCA** [Qin *et al.*, 2015], **DRFI** [Jiang *et al.*, 2013], **DSR** [Li *et al.*, 2013]). For fair comparison, we use either the implementations with recommended parameter settings or the saliency maps provided by the authors.

Quantitative Evaluation. As illustrated in Tab. 1, Tab. 2 and Fig. 2, our method outperforms other competing ones across all datasets in terms of near all evaluation metrics. Due to the limitation of space, we only present the PR curves on the DUTS-OMRON, DUTS-TE, ECSSD and HKU-IS. From these results, we have other notable observations: (1) deep learning based methods consistently outperform traditional methods with a large margin, which further proves the superiority of deep features for SOD. (2) our method achieves higher S-measure than other methods, especially on complex structure datasets, *e.g.*, the DUT-OMRON, SED and SOD datasets. We attribute this result to our structural loss. (3) without segmentation pre-training, our method only fine-tuned from the image classification model still achieves better results than the DCL and RFCN, especially on the HKU-IS and SED datasets. (4) compared to the DHS and Amulet, our method is inferior on the DUTS-TE and PASCAL-S datasets. However, our method still ranks in the second place.

Qualitative Evaluation. Fig. 3 provides several visual examples in various challenging cases, where our method outperforms other compared methods. For example, the images in the first two rows are of very low contrast, where most of the compared methods fail to capture the salient objects, while our method successfully highlights them with sharper edges preserved. The images in the 3-4 rows are challenging with complex structures or salient objects near the image boundary, and most of the compared methods can not predict the whole objects, while our method captures the whole salient regions with preserved structures.

Ablation Analysis. We also evaluate the main components in our model. Tab.3 shows the experimental results with different model settings. All models are trained on the augmented MSRA10K dataset and share the same hyper-parameters described in subsection 4.2. Due to the limitation of space, we only show the results on the ECSSD dataset. Other datasets have the similar performance trend. From the results, we can see that the SFCN only using the channel concatenation operator without hierarchical fusion (model (a)) has achieved comparable performance to most deep learning methods. This confirms the effectiveness of reflection features. With the hierarchical fusion, the resulting SFCN (model (b)) improves the performance by a large margin. The main reason is that the fusion method introduces more contextual information from high layers to low layers, which helps to locate the salient objects. In addition, it's no wonder that training with the $\mathcal{L}_{wbce}$ loss achieves better results than $\mathcal{L}_{bce}$. With other two losses $\mathcal{L}_{sc}$ and $\mathcal{L}_{s1}$, the model achieves better performance in terms of MAE and S-measure. These results demonstrate that individual components in our model complement each other. When taking them together, the overall model, *i.e.*, $SFCN+L_{wbce}+L_{se}+L_{s1}$, achieves best results under all evaluation metrics.

5 Conclusion

In this work, we propose a novel end-to-end feature learning framework for SOD. Our method uses a symmetrical FCN to learn complementary visual features under the guidance of lossless feature reflection. For training, we also propose a new weighted structural loss that integrates the location, semantic and contextual information of salient objects to boost the detection performance. Extensive experiments on seven large-scale saliency datasets demonstrate that the proposed method achieves significant improvement over the baseline and performs better than other state-of-the-art methods.

References

[Borji *et al.*, 2015] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *IEEE TIP*, 24(12):5706–5722, 2015.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[Everingham *et al.*, 2010] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2010.

[Fan *et al.*, 2017] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, pages 4548–4557, 2017.

[Hou *et al.*, 2017] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. In *CVPR*, pages 5300–5309, 2017.

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, pages 675–678, 2014.

[Jiang *et al.*, 2013] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, pages 2083–2090, 2013.

[Johnson *et al.*, 2016] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

[Land and McCann, 1971] Edwin H Land and John J McCann. Lightness and retinex theory. *JOSA*, 61(1):1–11, 1971.

[Lee *et al.*, 2016] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. Deep saliency with encoded low level distance map and high level features. In *CVPR*, pages 660–668, 2016.

[Li and Yu, 2015] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *CVPR*, pages 5455–5463, 2015.

[Li and Yu, 2016] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *CVPR*, pages 478–487, 2016.

[Li *et al.*, 2013] Xiaohui Li, Huchuan Lu, Lihe Zhang, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via dense and sparse reconstruction. In *ICCV*, pages 2976–2983, 2013.

[Li *et al.*, 2014] Yin Li, Xiaodi Hou, Christof Koch, James Rehg, and Alan Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014.

[Li *et al.*, 2016] Xi Li, Liming Zhao, Lina Wei, Ming-Hsuan Yang, Fei Wu, Yueting Zhuang, Haibin Ling, and Jingdong Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE TIP*, 25(8):3919–3930, 2016.

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

[Long *et al.*, 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[Qin *et al.*, 2015] Yao Qin, Huchuan Lu, Yiqun Xu, and He Wang. Saliency detection via cellular automata. In *CVPR*, pages 110–119, 2015.

[Shi *et al.*, 2016] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE TPAMI*, 38(4):717–729, 2016.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[Tong *et al.*, 2015] Na Tong, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Salient object detection via bootstrap learning. In *CVPR*, pages 1884–1892, 2015.

[Wang *et al.*, 2015] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, pages 3183–3192, 2015.

[Wang *et al.*, 2016] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, pages 825–841, 2016.

[Wang *et al.*, 2017a] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, pages 136–145, 2017.

[Wang *et al.*, 2017b] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, pages 4019–4028, 2017.

[Wu *et al.*, 2015] Xiang Wu, Ran He, and Zhenan Sun. A lightened cnn for deep face representation. *arXiv:1511.02683*, 2015.

[Xiao *et al.*, 2018] Yi Xiao, Peiyao Zhou, and Yan Zheng. Interactive deep colorization with simultaneous global and local inputs. *arXiv:1801.09083*, 2018.

[Xie and Tu, 2015] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, pages 1395–1403, 2015.

[Yang *et al.*, 2013] Chuan Yang, Lihe Zhang, Ruan Xiang Lu, Huchuan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013.

[Zhang *et al.*, 2017a] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, pages 202–211, 2017.

[Zhang *et al.*, 2017b] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency detection. In *ICCV*, pages 212–221, 2017.

[Zhao *et al.*, 2015] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *CVPR*, pages 1265–1274, 2015.