DSFMA: Deeply supervised fully convolutional neural networks based on multi-level aggregation for saliency detection

Inam Ullah¹, Muwei Jian^{2,4*}, Sumaira Hussain^{1,3}, Jie Guo¹, Li Lian¹, Hui Yu⁴, Kashif Shaheed¹, Yilong Yin^{1,*}

¹School of Software, Shandong University, Jinan 250101, China.

²School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, 250014, China.

³Department of Computer Science, Sindh Madressatul Islam University, Karachi-74000, Pakistan.

⁴School of Creative Technologies, University of Portsmouth, Portsmouth, UK.

*(Dr. Jian and Dr. Yin are co-corresponding authors) E-mail: jianmuwei@163.com; ylyin@sdu.edu.cn

Abstract:

In recent years, the emergence of fully convolutional neural networks (FCNs) has delivered significant success in the field of saliency detection. Although the different levels of FCNs layers can hold different types of information for salient object detection, it is still a challenging issue for the researchers to find a generic method while integrating all relevant information synthetically with multi-level aggregation. In this paper, we present a novel multi-level aggregation method by following a U-shaped architecture of the VGG-16 network. As the shallower layers of FCNs contain the low-level integrated features which are capable of capturing the more details of salient objects, while the more profound layers that hold the high-level integrated features have more contextual information. To exploit all the relevant information, we extend the last four side-outputs of U-Net at encoder and decoder sides and then utilize the concept of skip and short-connections to incorporate the high-level contextual knowledge with low-level details. Besides, we also integrate the recurrent convolutional layers (RCLs) into our model, which provide more deepness and enhance the capability to integrate more contextual knowledge. At last, we combine all the side-outputs into a final saliency map together for salient object detection. We evaluate the performance of the proposed model on six broadly used saliency detection benchmarks by comparing it with the other 11 state-of-the-art approaches. Experimental results demonstrate that our method achieves a favorable performance for all compared evaluation measures.

1. Introduction

Salient object detection, which focuses on discovering the most salient objects or parts (i.e. most fabulous and prominent objects) in an image, has achieved significant attention in recent years. Salient object detection has shown an excellent performance in computer vision as a preprocessing step, e.g., semantic-segmentation [1], image retrieval [2,3], object retargeting [4,5,6], visual tracking [7,8], Facial-Feature Detection[9], underwater vision[10,11] and scene classification[12,13]. Despite much research in the last two decades, salient object detection yet leftovers imperfect research problems. Because there is a wide variability of aspects that can play a different role in describing visual saliency, and it is very tough to collect all hand-tuned features or cues appropriately. However, detecting salient objects needs a semantic understanding of the entire image, along with the in-depth structures of the objects. Hence, salient object detection is still a critical and challenging issue in computer vision.

Motivated by the human visual attention system, numerous early-based approaches [17-23] in salient object detection mostly rely on several local and global low-level features. It is important to note that salient object detection and saliency detection both are interchangeable words that represent the same concept. Local-feature based [17,19] techniques aim at estimating the uniqueness of each region or pixel concerning local context for texture, color, and edge orientation, which naturally tend to focus object edges while usually neglect object interiors. Global-feature based [18,20] models consider the

Figure 1: Comparison of the proposed model with different types of models. Our approach generates more precise saliency maps as compared to other models in terms of the ground-truth (GT).

entire image to infer the salient objects. Unlike local -feature-based approaches that are responsive to the high-frequency image details like image boundaries, global strategies are usually less adequate when the textured regions of the prominent objects resemble the background. Seeing these problems, many researchers incorporated local with global contrasts [21,22] to integrate their courtesy interactions. However, these approaches are fragile in capturing high-level semantic information and mostly based on bottom-up hand-crafted features, which may flop to illustrate the images that have complex scenarios and object structures. It is challenging to find optimum integration.

Recently, the deep Convolutional Neural Networks (CNNs) has adhered far-ranging consideration in a variety of computer

vision methods for its superior performance over traditional hand-crafted features. CNN [24] has successfully been applied in the field of computer vision tasks such as semantic segmentation [25], image classification [26], edge detection [27], and object detection [28]. The arrival of fully convolutional neural networks (FCNs) [29] has further advanced the progress of these research spaces by providing more doctrine learning methods. As mentioned in many earlier models [30-32], due to the pyramid-like structural appearances of CNNs, shallower stages typically have bigger spatial sizes and possess detailed low-level information. In contrast, deeper stages usually have more high-level semantic information for better discovery of the contextual location of salient objects. Despite the FCNs based methods are very efficient and achieve a very encouraging performance in computer vision. However, in salient object detection, it needs two considerable complications to be attempted: how to highlight and ensure prominent objects against the cluttered background and how to retain the boundaries of salient objects. As the salient objects may contribute some identical visual features with its background distracters, and sometimes several salient objects intersect partially or totally with each other, saliency detection remains burdensome in computer vision tasks.

To address the above questions, several previous works [30,14,33,15] have attempted to leverage multi-level features to detect a salient object with clear boundaries. However, it is still a challenging task to produce an ideal multi-level aggregative strategy that can be used to detect a salient object with clear boundaries. In this work, we propose a novel multi-level aggregative strategy based on U-Net [34] architecture. For this purpose, we place a recurrent convolutional layer (RCL) at each multi-level stage in the decoder-side. Then we extend the last four multi-level side-outputs at encoder and decoder sides and deeply supervise the outputs. We also provide the short-connections among the extended side-outputs at the encoder-side. In the end, we integrate all the side-outputs to refine the final prediction of the network (i.e., also called saliency maps). Figure 1 shows that the results produced by our designed model are more accurate, and intimately near to the ground-truth. We summarize our main contribution as follows:

- We propose a deeply supervised, fully-convolutional neural network based on multi-level aggregation for salient object detection (DSFMA), which employs convolutional features from various side-outputs as saliency cues for salient object detection.
- As the deeper-sides, CNN contains the high-level (i.e., contextual) knowledge while the shallower side-outputs hold low-level spatial features. Different from existing saliency detection methods, we select extended side-outputs at both encoder and decoder sides by using the skip and short connections to integrate the high-level knowledge with low-level spatial information. That can better discover the salient objects along with transparent edges and recursively improve the coarse and irregular prediction maps of the deeper side.
- Besides, based on skip and short-connections, we provide recurrent convolutional layers at the decoder side for high-level contextual information.

2. Related Works

In this section, we shortly review typical and recurrent-based approaches for saliency detection. We also discuss those approaches of FCN, which assimilate multi-level features for salient object detection.

Over the last two decades, many salient object detection approaches have been proposed. The significant part of these approaches is based on low-level hand-crafted features, e.g., local contrast [35], global-contrast[36], color [24] and texture [37] [38]. The complete details of these approaches are beyond the scope of this work, and hence, we refer the interested readers to relevant survey papers [39,40] for further knowledge.

Compared with conventional methods that exploit hand-crafted features, deep learning approaches, especially the convolutional neural network (CNN) based techniques, have broken all the earlier state-of-the-art records in salient object detection. For example, Wang et al. [41] proposed the integration of two deep neural networks for global proposal search and for local pixel estimation to achieve salient object detection. Li et al. [42] extracted multi-scale features from deep CNNs to get a saliency map. Zhao et al.[43] also presented an aggregated approach to obtain multi-context CNN networks and acquired a saliency map for each superpixel by considering its local and global contextual information. Although these approaches achieved superior performance than conventional methods, none of them grasps low-level facts thoroughly, and all of their strategies comprise many fully connected layers, which are computationally costlier and may drop spatial features of input images.

To overcome the above issues, Lee et al. [44] encoded high-level semantic features of deep CNNs with low-level distance maps for salient object detection. Li et al. [45] proposed a fully convolutional network that contains a spatial pooling stream between encoder and decoder stage. Though a significant performance gain has been achieved, especially in the last two years, however, there is still ample space for further improvement to create a generic saliency-detection model for all aspects.

Recently, recurrent convolutional networks have been adopted in the field of saliency detection. Wang et al. [46] devised a recurrent, fully convolutional network (RFCN) for detecting salient objects. At each time step, they feed forward the input RGB image and the earlier saliency map through the RFCN to get the predicted saliency feature map, which is then considered as a saliency-prior for the subsequent time step. This type of recurrent network can polish the saliency map to a certain amount. In order to yield adequate and useful features for saliency detection, Tang et al. [47] proposed a method for saliency detection by incorporating the recurrent modules into each convolutional layer for learning contextual information. In [15], Liu et al. designed the DHSNet method and used a systematic, recurrent convolutional network, which enhances the saliency maps hierarchically and progressively step by step

Figure 2: Features visualization in different layers of Vgg-16, which show different levels of semantic for different layers.

via incorporating local context details. In [48], Kuen et al. used a recurrent attentional convolution-deconvolution network (RACDNN) to upgrade the saliency map progressively.

2.1 Feature aggregation in FCNs

As shown in Figure 2, visualizing various features of deep CNNs [49-52] at different levels have distinct views for the same object and its background. Deeper sides of CNN contain high-level semantic visual features, which support to recognize the class of image regions. In contrast, the shallower side of deep CNNs provides low-level visual features that can give a high-resolution prediction by generating sharp and details in the edges of the salient object. However, how to organize these multi-level features efficiently and effectively remains an open problem.

To this expiration, many attempts have been executed to tackle this issue. In [53], FCN based method has been introduced with skip connections that integrate deep-level layers and intermediate-level layers to build a pixel-wise prediction over several resolutions. Meanwhile, the Hypercolumn based method [54] incorporates convolutional features from various central-level layers and trains for high-level dense classification layers. The DeconvNet [27] and SegNet [55] followed a convolutional encoder-decoder approach with skip connections to utilize the features from multi-convolutional layers for image segmentation. Besides, the U-Net [34] also use skip-connections between index pooling at encoder and decoder layer with a systematic way for pixel-wise object detection. In [56], the authors have integrated Fast R-CNN and multi-level networks for salient object detection with sharp boundaries. The holistically-nested edge detection (HED) model [57] designed a deeply supervised method that fuses different multi-level features to deal with the confusing indecision in edges and object boundaries. Meanwhile, the deeply supervised salient object detection (DSS) model [16] enhanced the HED model by introducing short connections among extended side-outputs.

In this paper, our devised model further enhances the idea of SegNet [55] and DSS [16] in such a way that we introduce extended side-outputs at both encoder and decoder sides along with short-connections at the encoder and skip connections at the decoder-sides. We use a pre-trained vgg16 [58] and integrate all side-outputs saliency maps to refine the final saliency map recursively. The predicted saliency map of our model provides competitive state-of-the-art results and learns well to preserve object boundaries.

3. Our proposed method

Our primary model is based on the inspiration of U-Net [34] architecture, where the up-sampling layers at the decoder-side have the skip connections with its corresponding max-pooling layers at the encoder-side. The overall framework of our proposed work is illustrated in Figure 3.

The previous works in the deep salient object-detection era demonstrate that a more in-depth architecture enhances the result performance. When the model goes more profound, more multi-level features can be trained. Furthermore, it can have various stages with many strides and have the capability to learn more inherent characteristics from varied scales. It is also observed that deep level features contain more semantic features, which are useful for objectness but have less boundary information. While the features from the shallow layers commonly include more vibrant and detailed information, which are helpful for a clear boundary in high-resolution saliency-map prediction. The HED model [57] is one of the candidate representatives of such deep models, which accumulates different multi-level-side-outputs at each max-pool of the VGGNet [58]. Experimental results show that the HED model is not appropriate for saliency detection, because it is designed for edge detection, and saliency detection requires sophisticated methods. A good saliency detection algorithm should find the most visual and attentive objects, instead of only seeing the edge information from the image.

To overcome the limitations as mentioned above, we propose a U-Net [34] architecture to aggregate the multi-level features at both encoder and decoder side-outputs using the skip-layers and short layers connections for precise saliency detection. For this purpose, we recommend a novel architecture to extends the idea of HED [57] and DSS [16]. Like DSS, we also attach extended multi-level side-outputs at decoder-side and an additional recurrent convolutional layer at each upsampling layer. The following subsections describe the exhaustive description.

3.1 Multi-level Side-output Feature maps

In this section, we describe our network for saliency detection. We resize the input image into $256 \times 256 \times 3$ pixels. Then we utilize a pre-trained CNN model, i.e., VGG-16 [58], to extract the multi-level feature maps. VGG-16 model is very familiar with its sophistication and simplicity. In Figure 3, Conv1-2, Conv2-2, Conv3-3, Conv4-3, and Conv5-3 are the different levels of VGG-16. There are a total of 13 convolution layers and five max-pooling layers, and we deleted the top two

Figure 3: Architecture of our proposed encoder-decoder network with the VGG16 for salient object detection. The "Conv" mean convolutional layers at encoder-side while the D^i represent the decoding modules. The top arrows between encoder and decoder modules represent the skip-connections. The four branches at both encoder and decoder sides are the extended side-outputs with in-depth supervision. The arrows between side-outputs at the encoder-side represent the short-connections from top to bottom. The Final map represents the fusion of all side-outputs for predicting the saliency map. (Note: every convolutional block of the backbone and side-outputs are followed by BatchNorm and ReLU activation functions, except the last 1×1 convolution layers that are followed by Sigmoid activation function to normalize the value between 0 and 1.

fully connected layers. Every convolutional layer of the backbone is followed by a BatchNorm and ReLU activation layers. As there are several max-pooling layers in the VGG-16 model, which reduce the spatial resolutions of feature maps. For a given input image having $W \times H$ resolution, then the resulted feature maps after fifth max-pooling may have $\frac{W}{2^5}, \frac{H}{2^5}$ resolutions and hence reduces the feature maps resolutions to 32 times smaller than the original input. The reduction of spatial resolution is not beneficial for image segmentation, especially for image boundary detail.

To restore the spatial information that has been lost due to max pooling, and further enhance the performance, we connect a series of de-convolutional layers (i.e., D^5, D^4, D^3, D^2 , and D^1) to resize the precomputed feature maps. Instead of resizing specific feature maps by a ratio of $\{2, 4, 8, 16, 32\}$ as proposed by Long et al. [53], we followed a step-wise upsampling process, as displayed in Figure 3. At every Un-pooling layer, the current feature maps at the decoder-side are concatenated with the corresponding feature maps of encoder-side. Then the resulted feature maps are upsampled by a ratio of 2 using the bilinear-interpolation process. In Figure 3, the top skip-connections of the model represent the concatenated feature maps of the encoder side.

As the saliency detection is a class-agnostic process (i.e., whether an object/region is salient or not) depends on its backgrounds, i.e., context. As analyzed in previous works [49] [59], different CNN layers have different modes and show different levels of semantic. We also noticed that more profound layers could enhance better to locate the most salient regions. To integrate the different features of different levels, we extended the work of DSS[16]. We attach four extended side-outputs at every max-pooling of encoder-side (i.e., Conv1-2, Conv2-2, Conv3-3, and Conv4-3) and four extended side-

outputs at decoder side for every de-convolutional blocks (i.e., D^5, D^4, D^3 , and D^2) after applying the RCL layers. Every extended side-output contains three 3×3 convolutional layers followed by BatchNorm and ReLU layers. To facilitate the network training, we adopt the deep supervision in the last 1×1 convolutional layers of each side-output by utilizing the sigmoid activation function to generate the saliency maps of $W \times H$ size of the input image. Moreover, we adopted the "hole algorithm" [45] in the side-output4 to side-output7 with dilation rates 6, 4, and 2, which facilitate the network model to utilize the maximum global contextual knowledge. Besides, we adopted Recurrent Convolutional blocks at the decoder-side to further refine the feature maps and provide more deepness. The remaining details about kernels-sizes, dilation rates, and padding rates is mentioned in Table 1.

Finally, we integrate these all side-outputs along with the last-output of decoder block D^1 after RCL to find the final saliency map of our model. Table 2 represents the step-wise implementation overview of our proposed architecture.

3.2 Short connections

Our proposed approach relies on the observation that in-depth convolutional layers can better determine the location of salient regions but at the loss of spatial information, while' the emphasis on low-level features of shallower layers has a shortage of local contextual information. For this purpose, we use the short connections among different layers to exploit all information in a better way without any information loss. Like DSS [16], we also applied a top-down technique, which attaches a series of short connections from the deeper output side to the shallower outside. The logic of this consideration is to predict salient objects with more concise and clear boundaries because the superficial layers contain detailed information of edges. We apply the short-connections among

No.	Layer-Name	1	2	3	4
1	conv1-2 (128,3×3)	-	-	-	-
2	conv2-2 (128,3×3)	128,3×3	128,3×3	128, 3×3	1, 1×1
3	conv3-3 (256,3×3)	256,3×3	256,3×3	256, 3×3	1, 1×1
4	conv4-3(256, 3×3)	256,3×3,d=4,p=4	256,3×3 d=2, p=2	256, 3×3	1, 1×1
5	Conv5-3 (512,3×3)	512,3×3,d=6,p=6	512,3×3 d=4,p=4	512, 3×3 d=2,p=2	1, 1×1
6	D5 (512,3×3)	512,3×3,d=6,p=6	512,3×3 d=4,p=4	512, 3×3 d=2,p=2	1, 1×1
7	D4((256,3×3)	256,3×3,d=4, p=4	256,3×3 d=2,p=2	512, 3×3	1, 1×1
8	D3(256, 3×3)	256,3×3	256,3×3	512, 3×3	1, 1×1
9	D2 (128,3×3)	128,3×3	128,3×3	128, 3×3	1, 1×1
10	D1(1, 3×3)	-	-	-	-
11	RCL1(512,3×3)	-	-	-	-
12	RCL2(256,3×3)	-	-	-	-
13	RCL3(256,3×3)	-	-	-	-
14	RCL4(128,3×3)	-	-	-	-
15	RCL4(128,3×3)	-	-	-	-

Table 1. Detail of different convolutions of VGG-16 backbone and other connected branches in terms of output-channels and kernel-sizes. $(n, k \times k)$ denotes the number of channels and the corresponding kernel-size of convolutional layers. "Layer-Name" represents the encoder or decoder layers in the U-net-architecture. "1", "2", "3" and "4" represent the details of four convolutional layers that are used in each side-output for a specific Layer-Name or level. "d" represent dilation operation, and p represents padding function. (Note that in each side-output the first three convolutional layers are followed by a ReLU layer to apply the nonlinear transformation).

extended side-outputs at the encoder-side. For this, we take the output of side-output4, and before its last 1×1 convolutional layer, we feed it to the side-output3, side-output2, and side-output1. Similarly, from side-output3, we provide the short-connections to side-output2 and side-output1. A similar procedure is provided for side-output2 to side-output1. The short connections represent a simple integration of the corresponding side-outputs maps after the upsampling of the spatial-size of each side-output as the input image. By introducing the "hole algorithm" in a unique way, our method performs good than DSS[16] and HED [57] methods. We provide the short-connections among encoder side-outputs, which approximately boost the performance up to 4%, as shown in Table 4 (e). We do not give the short-connections among decoder side-outputs, because of negligible improvements.

3.3 Recurrent Convolutional Layer

To enable the model to train for more contextual facts, we place recurrent convolutional layers (RCL) at the top of each deconvolutional layer at D^i at the decoder-side. The red blocks in the decoder-side of Figure 3 represent the RCL layers. RCL engross recurrent connections in their convolutional layers [60]. For a unit positioned at (i, j) on the k th feature map in an RCL, its state at the time interval t is given by

$$x_{ijm}(t) = h(f(z_{ijm}(t))), \quad (1)$$

Where f is the ReLU activation layer, and h represent the local response normalization (LRN) activation function [61] to prevent the states from exploding:

$$h(f_{ijm}(t)) = \frac{f_{ijm}(t)}{\left(1 + \frac{\alpha}{N} \sum_{m=\max(0, m-N/2)}^{\min(M, m+N/2)} (f_{ijm})^2\right)^\beta}, \quad (2)$$

Where $f(z_{ijm}(t))$ is shortened as $f_{ijm}(t)$, M represents the overall number of feature maps, N denotes the dimension of the

Figure 4: Recurrent Convolutional Layers. (a) simple feed-forward convolutional layer. (b) represents the Recurrent convolutional layer (RCL). (c) represents the unfolding subnetworks of RCL.

local neighbor feature maps which are convoluted in the normalization process, α and β represent the constants to control the normalization.

In Eq. (1), $z_{ijm}(t)$ is the input of the unit to include a feed-forward association and a recurrent linking:

Where $(\mathbf{u}^{(i,j)})$ and $x^{(i,j)}(t-1)$ denote the feed-forward input from the preceding layer and the recurrent input from the existing layer at time step $t-1$, respectively.

$$z_{ijm}(t) = (\mathbf{W}_m^f)^T \mathbf{u}^{(i,j)} + (\mathbf{w}_m^r)^T x^{(i,j)}(t-1) + bm, \quad (3)$$

Whereas, bm describes the bias, $\mathbf{W}_k^f$ and $\mathbf{w}_k^r$ represent the feed-forward weights and the recurrent weights, respectively.. For more details about RCL, see reference [60]. As displayed in Figure 4(b), the recurrent convolutional layer replaces the simple feed-forward layer in Figure 4 (a). Unfolding the RCL with time steps T , yields feed-forward subnetworks of $T+1$ depth, as displayed in Figure 4 (c). In our approach, we fixed "2" as the input value of T , and keep the recurrent and the feed-forward inputs value unchanged for all iterations in the current block. If $t=0$, then only simple feed-forward computation occurs, while if $t=2$, then the length feed-forward layers lead to the depth of 3. We can observe that many recurrent links guide the subnetwork has several paths from the input layer to the output layer, which expedites the learning process. Besides, the active receptive field of an RCL unit increases when the time interval t increases, it also expands the active receptive field of

the RCL module, which empower the RCL unit to see broader global contexts without growing the number of network parameters. Hence, the RCL can train for more contextual facts. We further examine the success of our proposed method by carrying out several ablation tests and displaying the corresponding numerical and visual results in the next section.

3.4 Deep Supervision and Inference

As shown in figure 3, we deeply supervise the training of each side-output stage at the encoder and decoder sides, as did by [57,16]. Each side-output produces a loss term L_{side} which is defined as below:

$$L_{side}(I, G, W, w) = \sum_{n=1}^N \alpha_n \ell_{side}^{(n)}(I, G, W, w^{(n)}), \quad (4)$$

Where N regards the total side-output numbers (i.e., in our case $N=9$), W represents the collection of all standard network layer parameters, I and G denote the input image and the corresponding ground truth, respectively. α_n represents the weight of cross-entropy loss for n th side output. Each side-output layer is considered as a pixel-wise classifier with the corresponding weights w which is represented by

$$w = (w^{(1)}, w^{(2)}, \dots, w^{(N)}). \quad (5)$$

Here, $\ell_{side}^{(n)}$ refers to the cross-entropy loss function [16] of the n th side-output, which is calculated by the following equation:

$$\begin{aligned} \ell_{side}^{(n)}(I, G, W, w^{(n)}) &= - \sum_{k=1}^{|I|} G(k) \log \Pr(G(k) = 1 | I(k); W, w^{(n)}) \\ &+ \sum_{k=1}^{|I|} (1 - G(k)) \log \Pr(G(k) = 0 | I(k); W, w^{(n)}), \end{aligned} \quad (6)$$

Where $\Pr(G(k) = 1 | I(k); W, w^{(n)})$ represents the probability of the activation value at position k in the n th side-output, k represents the spatial coordinate.

Since deep supervision is applied in each side-outputs, we empirically weight different losses (i.e., set value of α) according to their importance. In our case for nine side-outputs, the values of α are:

$$\{\alpha_1, \alpha_2, \dots, \alpha_9\} = \{0.8, 0.8, 0.5, 0.5, 0.5, 0.5, 0.5, 0.8, 1\}$$

For the testing phase, we compute the average of all $\ell_{side}^{(n)}$ to avoid the negative effect produced due to the variation among shallowest and deepest level layers. The final average saliency map can be obtained by

$$w_{final-map} = \text{Mean}((w^{(1)}, w^{(2)}, \dots, w^{(n)})). \quad (7)$$

In Table 4 (e), we see that extended multi-level convolutional side-outputs at the encoder and decoder-side boost the performance at least 6% than the simple Unit model.

Furthermore, a step-wise implementation detail of our proposed approach is described in Table 2.

Table 2: A Step-wise implementation details of our proposed approach

1	Pass the input image from encoder-decoder architecture
---	--

2	Link multi-level features of encoder-side with their corresponding decoder-side using skip connections
3	Insert the Recurrent Convolutional chunks (RCLs) at every decoder stage.
4	Insert four extended side-outputs in encoder and decoder sides with four additional convolutional layers.
5	Apply “hole algorithm” or convolutions with different dilation rates in the last three side-outputs of encoder and decoder, which have comparatively small resolutions.
6	Apply bilinear interpolation process for up-sampling
7	Integrate multi-level features by using short-connections
8	Apply Sigmoid activation function to generate saliency maps for each side-outputs
9	Assign a proper weight for each side-output and then integrate all saliency maps with a simple addition process to generate the final-saliency maps.

4. Experiments

In this region, we evaluate the proposed DSFMA approach on six available public benchmark datasets. We conduct both quantitative and qualitative evaluation comparison with 11 other contemporary methods.

4.1 Datasets

We examine our approach on six public datasets: ECSSD [37], DUT-OMRON[38], HKU-IS [42], SOD [37], PASCAL-S [24] and DUTS-TE [62]. All these datasets contain ground-truth with pixel-wise annotations. The ECSSD includes 1000 ordinary images, semantically meaningful, and composite structures, which have many objects of different sizes. The DUT- OMRON has a total of 5168 natural scene images, manually picked from a large set, i.e., more than 140,000 photographs. The images of this dataset are comparatively complicated scenes/backgrounds. Thus, this dataset is considered very challenging and can provide more space for further improvements in the related research area of saliency detection. The SOD [37] is another challenging dataset, designed initially for testing image segmentation models. Different types of single and multiple objects with low contrast and touching borders make this dataset challengeable for salient object detection. The HKU-IS is also a large dataset having 4447 natural images. It contains many multiple foreground objects comparatively having low contrast. The PASCAL-S dataset includes 850 natural scene images. It is considering a complex dataset for saliency detection and contains a cluttered background and different foreground objects. The DUTS[62] dataset is presently the common salient object detection benchmark dataset. It covers 10553 images in the training set, named DUTS-TR, and contains 5019 images in their test set, named DUTS-TE. It contains images with challenging scenarios.

4.2 Evaluation Metrics

We use five different evaluation metrics to estimate the performance of our approach with other different saliency detection algorithms. These metrics are F -measures, S -measure, mean absolute error (MAE), and the generally used precision-recall (PR)[39] and F -measure curves. The PR

Table 3: The F -measure, S -measure, and MAE of various saliency detection approach on five challenging saliency detection datasets. The top three results are displayed in red, green, and blue. Our proposed plan ranks first or 2nd on these datasets.

Methods	PASCAL-S			HKU-IS			SOD			DUT-OMRON			ECSSD			DUTS-TE		
	F_β	S_λ	MAE	F_β	S_λ	MAE	F_β	S_λ	MAE	F_β	S_λ	MAE	F_β	S_λ	MAE	F_β	S_λ	MAE
DSFMA	.876	.821	.093	.927	.910	.042	.850	.773	.082	.791	.821	.069	.932	.920	.042	.853	.858	.052
R3Net[[63]	.855	.800	.097	.917	.891	.038	.836	.732	.087	.792	.815	.061	.931	.900	.046	.824	.834	.067
SRM[64]	.847	.832	.085	.906	.887	.046	.843	.742	.127	.769	.797	.069	.917	.895	.054	.826	.834	.059
DSS[16]	.836	.797	.096	.910	.879	.041	.844	.751	.121	.771	.788	.066	.916	.882	.052	.825	.822	.057
Amulet[14]	.837	.820	.098	.895	.883	.052	.806	.758	.141	.742	.780	.098	.915	.894	.059	.777	.802	.085
UCF[32]	.828	.803	.126	.886	.866	.074	.803	.754	.164	.734	.758	.132	.911	.883	.078	.623	.777	.117
DHS[15]	.829	.807	.097	.890	.870	.053	.827	.750	.128	--	--	--	.907	.884	.059	.807	.817	.067
DCL[45]	.805	.754	.125	.885	.819	.072	.823	.735	.141	.739	.713	.097	.890	.828	.088	.781	.734	.088
ELD[65]	.773	.757	.123	.839	.820	.074	.764	.705	.155	.715	.750	.092	.867	.839	.079	.737	.753	.092
DS[66]	.765	.739	.176	.865	.852	.080	.784	.712	.190	.745	.750	.120	.882	.821	.122	.776	.792	.090
NLDF[31]	.831	.803	.099	.902	.879	.048	.841	.757	.124	.753	.770	.080	.905	.875	.063	.812	.815	.065
DSR[67]	.678	.603	.229	.734	.681	.201	-	-	-	.535	.672	.138	.735	.685	.171	.735	.685	.171
DRFI[68]	.712	.665	.234	.773	.712	.198	-	-	-	0.66	.697	.155	.786	.751	.164	.786	.751	.164

curves represent the average precision and recall values of the saliency maps at various thresholds of a specific dataset. The F-measure describes the weighted harmonic mean of average precision and average recall. It can be defined by

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{Precision} \times \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}, \quad (8)$$

where we adjust 0.3 as the value of β^2 to give more weight to precision than recall.

The above are overlapping-based evaluation methods, which generally gives a high score to those methods, which provide high saliency scores to salient pixels correctly. For a reasonable comparison, we also compute the mean absolute error (MAE). It calculates the average of absolute pixel-wise error between the ground truth G and their corresponding saliency map S . The MAE score is calculated by

$$MAE = \frac{1}{W \times H} (\sum_{X=1}^W \sum_{Y=1}^H |S(X, Y) - G(X, Y)|), \quad (9)$$

where W is the width, and H represents the height for the given image. $S(X, Y)$ and $G(X, Y)$ represent the pixel-wise values of the saliency map and their corresponding ground truth mask at (X, Y) .

Similarly to assess the spatial structure similarities of saliency maps, we also compute the S -measure [69] by

$$S_\gamma = \gamma * S_o + (1 - \gamma) * S_r, \quad (10)$$

Where $\gamma \in [0, 1]$ denotes the balance parameter. S_o represents the object-aware and S_r represents regional-aware structural similarity. We feed $\gamma = 0.5$ as recommended.

4.3 Implementation Training

Training data. The training data considerably affect the performance of a deep neural network. We used DUTS-TE[62] dataset to train our model. We allowed random rotation, cropping, and flipping for data augmentation. We do not provide a validation set and learn the model until the convergence of training loss. A quad-core PC machine with an i5-6600 CPU and an NVIDIA Titan 1080 GPU (with 8G memory) is used for training. We used Pytorch library[70] and followed the publically exposed implementation of FCN[53]. For training our model, we set $256 \times 256 \times 3$ resolution size to input into our model. We selected VGGNet [58] as the pre-trained model for our multi-level encoder layers and used the "Xavier" method [71] for initializing other layers. We used the Adam optimizer for training our network and provided 0.9 for

momentum and 0.0005 as a decay weight. Our learning rate begins from $1e-8$ and decreases at a 5% rate when the training loss reaches the convergence. We keep similar hyper-parameters for all side-outputs and final saliency map.

4.4 Comparison with state-of-the-art methods

We analyze our proposed approach with 11 typical methods on datasets, as mentioned above. These methods are DRFI[68], DSR[67], UCF[32], SRM[64], NLDF[31], ELD[65], DS[66], DCL[45], DHS[15], DSS[16] and Amulet[14]. The first two methods are classical unsupervised-based methods, while the remaining 9 are CNN supervised-based approaches. For a fair assessment, we use either the implementations with acclaimed parameter settings or the saliency maps distributed by the authors.

Quantitative Evaluation.

In Table 3, we present the quantitative comparison of all the approaches with F-measure, S-measure, and MAE based on DUT-OMRON[38], SOD[37], ECSSD[37], HKU-IS[42], PASCAL-S[24] datasets and DUTS-TE[62]. Besides, we illustrate the PR and F-measure curves for all methods, as shown in Figure 6 and Figure 7. From the results, we perceived that our model consistently attains comparable or superior feat than all other state-of-the-art models on most datasets, particularly in terms of the weighted F -measure. It is also worth noticing that without any post-processing method like CRF, our proposed method still attains the best performance and indicate noteworthy performance advances over existing approaches. The better performance approves that our model has excellent generalization capabilities on additional large-scale datasets.

Figure 5. Qualitative evaluation of saliency maps from six different challengeable datasets with other state-of-the-art methods.

Qualitative Evaluation.

In Figures 5, we show a visual comparison of our proposed method with other approaches. We note that our approach surpasses the equated criteria in different challenging cases, for images with complex backgrounds, foregrounds and low contrast (see rows 1, 2, 3, 6, 9 and 10), for objects touching the image borders (see row 4, and 6), and for objects comprise a similar look with the background (see row 7 and 9). Besides, for those complex images with small salient objects (see row 5) and multiple objects (see row 2, 8 and 10), For thread-like objects (see row, 6, 9, and 11). Our method gives more accurate saliency maps than others. Most importantly, even without any post-processing method, our approach highlights salient objects

more consistently than other methods, which indicates the effectiveness and strength of our proposed method.

4.5 Ablation analysis

To demonstrate the role of different modules in our model, we execute the following ablation studies on the PASCAL-S, HKU-IS, and ECSSD datasets. We trained all the models using VGG-16 as a base model under similar hyper-parameters and listed all the results in Table 4. (a) We start from a straightforward encoder-decoder network like FCNs-8s [29] to validate the importance of our model. (b) In this stage, we extend the model (a) by introducing pooling-based

Figure 6: Comparison of six different accessible datasets in terms of PR curves.

Figure 7: Comparison of six different accessible datasets in terms of F-Measure.

Table 4. The experimental results for different models on ECSSD, HKU-IS, and PASCAL-S datasets to examine the role of various components. All the models use the same hyper-parameters and are trained on the augmented DUTS dataset.

Models	Similar Methods	ECSSD		HKU-IS		PASCAL-S	
		$F\beta$	MAE	$F\beta$	MAE	$F\beta$	MAE
(a) Simple encoder-decoder	FCN-8s[53]	0.8120	0.1343	0.8006	0.1213	0.7198	.1933
(b) a + pooling-based skip connection	U-Net [34]	0.8245	0.1097	0.8147	0.1005	0.7271	.1687
(c) b + simple multi-level feature aggregation at encoder-side	HED [57]	0.8421	0.0910	0.8311	0.0801	0.7502	.1364
(d) c + multi-level enhanced feature integration at encoder-side	-	0.8643	0.0782	0.8591	0.0671	0.7831	.1282
(e) d + short connection at encoder-side	DSS [16]	0.913	0.055	0.902	0.042	0.832	.097
(f) e + multi-level enhanced feature integration at decoder-side	-	0.9207	0.0511	0.9104	0.0441	0.8491	.0941
(g) f + attaching Recurrent convolutional layers (RCLs) at decoder-side	DSFMA (ours)	0.9321	0.0423	0.9272	0.0420	0.8613	.0931

skip-connections like the U-Net [34]. (c) To examine the improvement of multi-level feature integration, next, we take the model (b) and attached side-outputs to each level with a Sigmoid classifier at the encoder-side like the work done in HED [57]. Next, we enhance the preceding model (c) side-outputs with four more convolutional layers at the last four max-pooling layers. (d) In this stage, we introduce the short connection among extended side-outputs at the encoder-side from top-to-bottom as the work done in DSS [16]. (e) Further, we apply the same procedure of model (d) (i.e., we attach side-outputs with extended four convolutional layers along with sigmoid function at the first four upsampling layers at the decoder side. (f) In this last stage, we extend the previous model (e) with Recurrent Convolutional Layers (RCL) at different levels at the decoder side. From Table 4, we see that our model consistently gains accuracy as we attach various components to our model. We also examined that side-outputs at encoder-side improve the performance more than decoder side outputs. Considering this fact, we only introduce short connections at the encoder-side, which significantly improves the results. However, in the decoder, the impact was just negligible, so we do not include it. In the last row of Table 4, we see that our RCL layer efficiently boosts the accuracy of our model.

4.6 Comparison with other similar type models

We also compared our proposed model with more similar models to verify the detection performance. We use the same augmented dataset (i.e., DUTS[62]) under the same hyper-parameters. For this, we select FCN8s[53], SegNet (SN) [55], Hypercolumn (HC) [54], HED [57], DeconvoNet(DN) [27] and DSS [16] models which use the VGG-16 model pre-trained on the ImageNet [58]. We only select the ECSSD dataset for performance comparison, as shown in Table 5.

Table 5: The performance of similar types of models on the ECSSD dataset.

Models	FCN-8s[53]	HC[54]	DN[27]	SN[55]	HED[57]	DSS[16]	DSFMA
$F\beta$	0.812	0.805	0.8125	0.824	0.842	0.919	0.932
MAE	0.129	0.118	0.131	0.109	.0910	.0554	.0423

4.7 Timing

We train our network on the augmented DUTS dataset which contains 10553 training images, the training procedure takes nearly 16 hours and converges after 200k iterations. During the testing stage, it takes approximately 0.09 seconds to process a 300×400 input image. As compared to old and new FCN-based models, our model is comparatively fast but not much faster than some recent works.

5. Conclusion

In this paper, we propose a novel multi-level aggregation method by following a U-shaped architecture of the VGG-16 network. Our proposed model integrates the multi-level features by utilizing the extended side-outputs at encoder and decoder sides. We applied the skip and short-connections to refine the low-level details and use the recurrent convolutional layers to improve the high-level contextual knowledge. Experimental results show that our model demonstrates the best performance on five widely-used benchmark datasets by comparing it with the other 11 state-of-the-art methods.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (NSFC) (61876098, 61976123, 61601427); Royal Society - K. C. Wong International Fellowship (NIF\R1\180909) and the Taishan Young Scholars Program of Shandong Province.

6. References

1. Donoser M, Urschler M, Hirzer M, Bischof H Saliency driven total variation segmentation. In: 2009 IEEE 12th International Conference on Computer Vision, 2009. IEEE, pp 817-824
2. He J, Feng J, Liu X, Cheng T, Lin T-H, Chung H, Chang S-F Mobile product search with bag of hash bits and boundary reranking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012. IEEE, pp 3005-3012
3. Gao Y, Wang M, Tao D, Ji R, Dai Q (2012) 3-D object retrieval and recognition with hypergraph analysis. IEEE Transactions on Image Processing 21 (9):4290-4303
4. Vinyals O, Toshev A, Bengio S, Erhan D Show and tell: A neural image caption generator. In: Proceedings of the IEEE

- conference on computer vision and pattern recognition, 2015. pp 3156-3164
5. Jian M, Zhang W, Yu H, Cui C, Nie X, Zhang H, Yin Y (2018) Saliency detection based on directional patches extraction and principal local color contrast. *Journal of Visual Communication and Image Representation* 57:1-11
6. Jian M, Zhao R, Sun X, Luo H, Zhang W, Zhang H, Dong J, Yin Y, Lam K-M (2018) Saliency detection based on background seeds by object proposals and extended random walk. *Journal of Visual Communication and Image Representation* 57:202-211
7. Borji A, Frintrop S, Sihite DN, Itti L Adaptive object tracking by learning background context. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012. IEEE, pp 23-30
8. Mahadevan V, Vasconcelos N (2012) Biologically inspired object tracking using center-surround saliency mechanisms. *IEEE transactions on pattern analysis and machine intelligence* 35 (3):541-554
9. Jian M, Lam K-M, Dong J (2014) Facial-feature detection and localization based on a hierarchical scheme. *Information Sciences* 262:1-14
10. Jian M, Qi Q, Dong J, Yin Y, Lam K-M (2018) Integrating QDWD with pattern distinctness and local contrast for underwater saliency detection. *Journal of visual communication and image representation* 53:31-41
11. Jian M, Qi Q, Yu H, Dong J, Cui C, Nie X, Zhang H, Yin Y, Lam K-M (2019) The extended marine underwater environment database and baseline evaluations. *Applied Soft Computing* 80:425-437
12. Ren Z, Gao S, Chia L-T, Tsang IW-H (2013) Region-based saliency detection and its application in object recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 24 (5):769-779
13. Siagian C, Itti L (2007) Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE transactions on pattern analysis and machine intelligence* 29 (2):300-312
14. Zhang P, Wang D, Lu H, Wang H, Ruan X Amulet: Aggregating multi-level convolutional features for salient object detection. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2017. pp 202-211
15. Liu N, Han J Dhsnet: Deep hierarchical saliency network for salient object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. pp 678-686
16. Hou Q, Cheng M-M, Hu X, Borji A, Tu Z, Torr P Deeply supervised salient object detection with short connections. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. IEEE, pp 5300-5309
17. Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis machine intelligence* 20 (11):1254-1259
18. Global contrast based salient region detection (2015). *IEEE transactions on pattern analysis machine intelligence* 37 (3):569-582
19. Xie Y, Lu H, Yang M-H (2013) Bayesian saliency via low and mid level cues. *IEEE Transactions on Image Processing* 22 (5):1689-1698
20. Achanta R, Hemami S, Estrada F, Susstrunk S Frequency-tuned salient region detection. In: *Computer vision and pattern recognition, 2009. cvpr 2009. IEEE conference on*, 2009. IEEE, pp 1597-1604
21. Tong N, Lu H, Zhang Y, Ruan X (2015) Salient object detection via global and local cues. *Pattern Recognition* 48 (10):3258-3267
22. Shi K, Wang K, Lu J, Lin L Pisa: Pixelwise image saliency by aggregating complementary appearance contrast measures with spatial priors. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. pp 2115-2122
23. Arqub OA, Maayah B (2019) Modulation of reproducing kernel Hilbert space method for numerical solutions of Riccati and Bernoulli equations in the Atangana-Baleanu fractional sense. *Chaos, Solitons & Fractals* 125:163-170
24. Li Y, Hou X, Koch C, Rehag JM, Yuille AL The secrets of salient object segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. pp 280-287
25. Taigman Y, Yang M, Ranzato MA, Wolf L Deepface: Closing the gap to human-level performance in face verification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014. pp 1701-1708
26. Girshick R, Donahue J, Darrell T, Malik J Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014. pp 580-587
27. Noh H, Hong S, Han B Learning deconvolution network for semantic segmentation. In: *Proceedings of the IEEE international conference on computer vision*, 2015. pp 1520-1528
28. Liu N, Han J, Zhang D, Wen S, Liu T Predicting eye fixations using convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. pp 362-370
29. Shelhamer E, Long J, Darrell T (2016) Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:06211*
30. Hou Q, Cheng M-M, Hu X, Borji A, Tu Z, Torr PH Deeply supervised salient object detection with short connections. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. pp 3203-3212
31. Luo Z, Mishra AK, Achkar A, Eichel JA, Li S, Jodoin P-M Non-local Deep Features for Salient Object Detection. In: *CVPR*, 2017. vol 6. p 7
32. Zhang P, Wang D, Lu H, Wang H, Yin B Learning uncertain convolutional features for accurate saliency detection. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2017. pp 212-221
33. Zhang P, Wang L, Wang D, Lu H, Shen C (2018) Agile amulet: Real-time salient object detection with contextual attention. *arXiv preprint arXiv:06527*
34. Ronneberger O, Fischer P, Brox T U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*, 2015. Springer, pp 234-241
35. Klein DA, Frintrop S Center-surround divergence of feature statistics for salient object detection. In: *2011 International Conference on Computer Vision*, 2011. IEEE, pp 2214-2219

36. Cheng M-M, Mitra NJ, Huang X, Torr PH, Hu S-M (2014) Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (3):569-582
37. Yan Q, Xu L, Shi J, Jia J Hierarchical saliency detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. pp 1155-1162
38. Yang C, Zhang L, Lu H, Ruan X, Yang M-H Saliency detection via graph-based manifold ranking. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013. pp 3166-3173
39. Borji A, Cheng M-M, Jiang H, Li J (2015) Salient object detection: A benchmark. *IEEE transactions on image processing* 24 (12):5706-5722
40. Ullah I, Jian M, Hussain S, Guo J, Yu H, Wang X, Yin Y (2020) A brief survey of visual saliency detection. *Multimedia Tools and Applications*. doi:10.1007/s11042-020-08849-y
41. Wang L, Lu H, Ruan X, Yang M-H Deep networks for saliency detection via local estimation and global search. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. pp 3183-3192
42. Li G, Yu Y Visual saliency based on multiscale deep features. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015. pp 5455-5463
43. Zhao R, Ouyang W, Li H, Wang X Saliency detection by multi-context deep learning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. pp 1265-1274
44. Chen T, Lin L, Liu L, Luo X, Li X (2016) DISC: Deep Image Saliency Computing via Progressive Representation Learning. *IEEE Trans Neural Netw Learning Syst* 27 (6):1135-1149
45. Li G, Yu Y Deep contrast learning for salient object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. pp 478-487
46. Wang L, Wang L, Lu H, Zhang P, Ruan X Saliency detection with recurrent fully convolutional networks. In: *European Conference on Computer Vision*, 2016. Springer, pp 825-841
47. Tang Y, Wu X, Bu W Deeply-supervised recurrent convolutional neural network for saliency detection. In: *Proceedings of the 24th ACM international conference on Multimedia*, 2016. pp 397-401
48. Kuen J, Wang Z, Wang G Recurrent attentional networks for saliency detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. pp 3668-3677
49. Zeiler MD, Fergus R Visualizing and understanding convolutional networks. In: *European conference on computer vision*, 2014. Springer, pp 818-833
50. Wang L, Ouyang W, Wang X, Lu H Visual tracking with fully convolutional networks. In: *Proceedings of the IEEE international conference on computer vision*, 2015. pp 3119-3127
51. Mahendran A, Vedaldi A Understanding deep image representations by inverting them. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015. pp 5188-5196
52. Simonyan K, Vedaldi A, Zisserman A (2013) Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*
53. Long J, Shelhamer E, Darrell T Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015. pp 3431-3440
54. Hariharan B, Arbeláez P, Girshick R, Malik J Hypercolumns for object segmentation and fine-grained localization. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015. pp 447-456
55. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis machine intelligence* 39 (12):2481-2495
56. Wang X, Ma H, Chen X, You S (2018) Edge preserving and multi-scale contextual neural network for salient object detection. *IEEE Transactions on Image Processing* 27 (1):121-134
57. Xie S, Tu Z Holistically-nested edge detection. In: *Proceedings of the IEEE international conference on computer vision*, 2015. pp 1395-1403
58. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv*
59. Li Y, Yosinski J, Clune J, Lipson H, Hopcroft JE Convergent Learning: Do different neural networks learn the same representations? In: *Iclr*, 2016.
60. Liang M, Hu X Recurrent convolutional neural network for object recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. pp 3367-3375
61. Krizhevsky A, Sutskever I, Hinton GE Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, 2012. pp 1097-1105
62. Wang L, Lu H, Wang Y, Feng M, Wang D, Yin B, Ruan X Learning to detect salient objects with image-level supervision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. pp 136-145
63. Deng Z, Hu X, Zhu L, Xu X, Qin J, Han G, Heng P-A R3Net: Recurrent residual refinement network for saliency detection. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018. AAAI Press, pp 684-690
64. Wang T, Borji A, Zhang L, Zhang P, Lu H A stagewise refinement model for detecting salient objects in images. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2017. pp 4019-4028
65. Lee G, Tai Y-W, Kim J Deep saliency with encoded low level distance map and high level features. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. pp 660-668
66. Li X, Zhao L, Wei L, Yang M-H, Wu F, Zhuang Y, Ling H, Wang J (2016) Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE Transactions on Image Processing* 25 (8):3919-3930
67. Li X, Lu H, Zhang L, Ruan X, Yang M-H Saliency detection via dense and sparse reconstruction. In: *Proceedings of the IEEE international conference on computer vision*, 2013. pp 2976-2983

68. Jiang H, Wang J, Yuan Z, Wu Y, Zheng N, Li S Salient object detection: A discriminative regional feature integration approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013. pp 2083-2090
69. Fan D-P, Cheng M-M, Liu Y, Li T, Borji A Structure-measure: A new way to evaluate foreground maps. In: Proceedings of the IEEE international conference on computer vision, 2017. pp 4548-4557
70. Paszke AaG, Sam and Chintala, Soumith and Chanan, Gregory and Yang, Edward and DeVito, Zachary and Lin, Zeming and Desmaison, Alban and Antiga, Luca and Lerer, Adam (2017) Automatic differentiation in PyTorch.
71. Glorot X, Bengio Y Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010. pp 249-256