Pattern Recognition 121 (2022) 108212

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Recursive multi-model complementary deep fusion for robust salient object detection via parallel sub-networks

Zhenyu Wu[a], Shuai Li[a], Chenglizhao Chen[a,b,*], Aimin Hao[a,d,e], Hong Qin[c]

[a] State Key Laboratory of Virtual Reality Technology & Systems, Beihang University, China
[b] College of Computer Science & Technology, Qingdao University, China
[c] Department of Computer Science, Stonybrook University, USA
[d] Peng Cheng Laboratory, China
[e] Research Unit of Virtual Human & Virtual Surgery, Chinese Academy of Medical Sciences, China

ARTICLE INFO

Article history:
Received 22 March 2020
Revised 5 July 2021
Accepted 27 July 2021
Available online 3 August 2021

Keywords:
Salient object detection
Deep learning
Multi-model fusion

ABSTRACT

Fully convolutional networks have shown outstanding performance in the salient object detection (SOD) field. The state-of-the-art (SOTA) methods have a tendency to become deeper and more complex, which easily homogenize their learned deep features, resulting in a clear performance bottleneck. In sharp contrast to the conventional "deeper" schemes, this paper proposes a "wider" network architecture which consists of parallel sub-networks with totally different network architectures. In this way, those deep features obtained via these two sub-networks will exhibit large diversity, which will have large potential to be able to complement with each other. However, a large diversity may easily lead to the feature conflictions, thus we use the dense short-connections to enable a recursively interaction between the parallel sub-networks, pursuing an optimal complementary status between multi-model deep features. Finally, all these complementary multi-model deep features will be selectively fused to make high-performance salient object detections. Extensive experiments on several famous benchmarks clearly demonstrate the superior performance, good generalization, and powerful learning ability of the proposed wider framework.

1. Introduction

The objective of salient object detection is to identify the most visually distinctive object in the given image [1]. As a preprocessing tool, salient object detection (SOD) has a wide range of practical applications, including visual tracking [2], localization [3], video saliency [4], image captioning [5,6], image retrieval [7], visual question answering [8] and object retargeting [9].

Previous works frequently treat the SOD as a multi-level perception task [10–12], in which its key rationale is to make full use of the saliency clues at different perception levels [13]. Recently, the fully convolutional networks (FCNs) has been adopted for the robust SOD, in which such success should be attributed to its ability to learn hierarchical saliency clues. Thus, the current state-of-the-art (SOTA) models [14–16] generally focus on how to utilize the hierarchical deep features in "single network" to produce high-quality SOD. Nevertheless, the hierarchical deep features revealed in an identical network have a tendency to be homogenization, resulting in a limited performance eventually.

In the view of the neuroscience, human visual system mainly comprises two largely independent subsystems that mediate different classes of visual behaviors [17,18]. The subcortical projection from the retina to the cerebral cortex is strongly dominated by the two pathways that are relayed by the magnocellular (M) and parvocellular (P) subdivisions of the lateral geniculate nucleus (LGN), in which the parallel pathways generally exhibit two main characteristics: 1) the M cells contribute to the low-level transient processing (e.g., visual motion perception, eye movement, etc.) while the P cells contribute more to the high-level recognition tasks (e.g., object recognition, face recognition, etc.); 2) the M and P cells are separated in the LGN, but it is recombined in visual cortex latter.

Motivated by the above-mentioned theory, we propose to use two parallel networks (see Fig. 1) to mimic the binocular vision of human visual system. The key point of the proposed parallel network architecture is its ability to conduct multi-level saliency estimation while avoiding the conventional single network architecture inducted feature homogenization problem. To achieve it, we devise a novel multi-model deep fusion framework, which at-

* Corresponding author.
E-mail address: cclz123@163.com (C. Chen).

https://doi.org/10.1016/j.patcog.2021.108212
0031-3203/

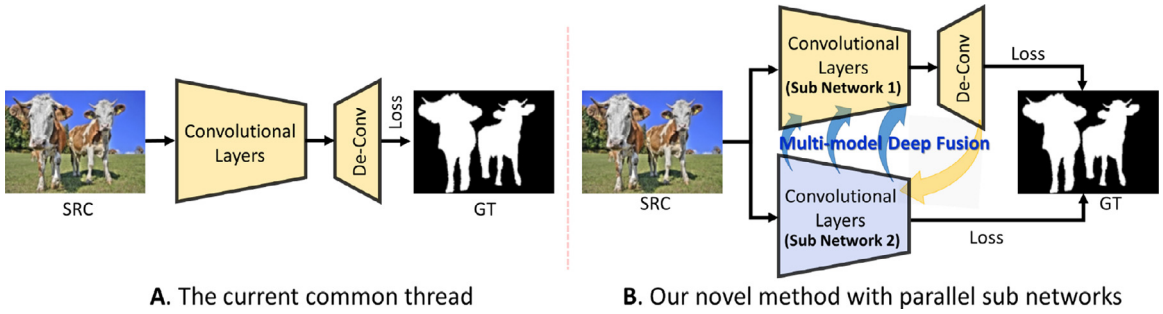

Fig. 1. The major difference between our method and the conventional methods.

tempts to fully exploit the complementary deep features from two different parallel subnetworks: the coarse-level saliency localization network and the fine-scale detail polishing network. Meanwhile, inspired by the aforementioned attributes, we adopt the inter-model short-connections to recursively ensure a complementary status between each of our subnetworks. Moreover, we utilize an FCNs based saliency regressor to conduct selective deep fusion over those inter-model deep features, achieving a high-performance SOD eventually.

It should be noted that our "wide" scheme is solely implemented by using simple network architecture, yet it has achieved remarkable performance improvement comparing to the conventional complicated "deeper" schemes. And such performance improvements are mainly induced by the newly designed multi-model fusion scheme, in which the adopted simple network architecture is a hallmark of the proposed method. Moreover, to our best knowledge, our paper is the first attempt to handle the SOD from the "wider" perspective.

To demonstrate the advantages of our method, we have conducted massive quantitative comparisons against 14 most representative SOTA methods over 5 widely used publicly available datasets. Also, we have conducted extensive ablation studies to comprehensively verify the effectiveness of each essential component in our method. Specifically, the salient contributions of this paper can be summarized as follows:

- We provide a deeper insight into the salient object detection task by imitating the binocular vision of human perception process;
- To alleviate the obstinate feature homogenization problem in single network case, we utilize parallel subnetworks to automatically reveal saliency clues at different spatial levels;
- We propose an end-to-end salient object detection model that learns diversity saliency clues in an iterative manner, aiming to achieve an optimal complementary status between the deep features extracted by our parallel subnetworks;
- We also provide a novel selective fusion strategy to fuse multi-model saliency clues for high-performance salient object detection, achieving the new SOTA performance over the five adopted datasets.
- The source code is available at: https://github.com/Diamond101010/RMMDF, which may has large potential to benefit the image salient object detection community in the future.

2. Related work

Early methods largely adopt various hand-crafted visual features [19–21] to model the human visual attention [22], which are limited in generalization and effectiveness. However, these methods have low computational efficiency and ignore rich contextual semantic information. More details about traditional methods are discussed in [23,24]. Here we mainly discuss deep learning based saliency detection models.

2.1. Single-stream network

The single-stream network is one of standard architecture adopted by many state-of-the-art methods, consisting of a sequential cascade of convolution layers, pooling layers and non-linear activation operations. Li et al. [25] first proposed a convolutional neural networks (CNNs) based computational model, which incorporates the multi-scale deep features via simply vector-wise feature concatenation. Inspired by the great success of fully convolutional network (FCN) [26] in semantic segmentation, recent deep SOD models adapt popular classification models, e.g., VGGNet and ResNet to predict the whole saliency maps directly. Similarly, Liu et al. [27] proposed a hierarchical refinement model in which the coarse saliency map by gradually combining shallower features by using recurrent layers. In [14], short connections are introduced from deeper side-outputs to shallower ones. In this way, higher-level features can help lower side-outputs to better locate the salient regions, while lower-level features can help enrich the higher-level side-outputs with finer details. Wang et al. [28] present progressive feature polishing network, a simple yet effective framework to progressively polish the multi-level features to be more accurate and representative. Zhang et al. [29] also proposed a novel method to aggregate multi-level features at multiple resolutions, in which the key rationale is to simultaneously integrate high-level semantical information with low-level details without considering the inter-layer relations. Instead of concatenating multi-level features directly, we devise the **s**elective **d**eep **f**usion (SDF) to make full use of the multi-level features. To be more specific, we first utilize the saliency maps in the previous stages, e.g., M^{t-1}, which contain both high-level semantic information and low-level details, to automatically 'select' which features in the next stage should be used. Moreover, after obtaining the refined features, the high-level fine-scale saliency will be fused with multi-scale features derived from another subnetwork.

Besides, Wang et al. [30] proposed a recurrent fully convolutional networks (RFCN) which recurrently refines the saliency prediction based on the input image and the saliency priors from heuristic calculation or prediction of previous time step. At the first glance, our proposed model is partially similar to the existing RFCN, where both our model and the RFCN follow the coarse-to-fine manner for the SOD task. However, our model is different from the RFCN in essence, and we would like to detail these critical differences as below: 1) The RFCN is heavily dependent on multiple saliency priors, taking them as the auxiliary input to boost the training convergency and yield more accurate SOD results; in sharp contrast, our method doesn't need such priors, where the performance gain is mainly ensured by seeking the complementary status between different feature backbones. 2) In view of the network design, the proposed model is a type of U-Net architecture, while

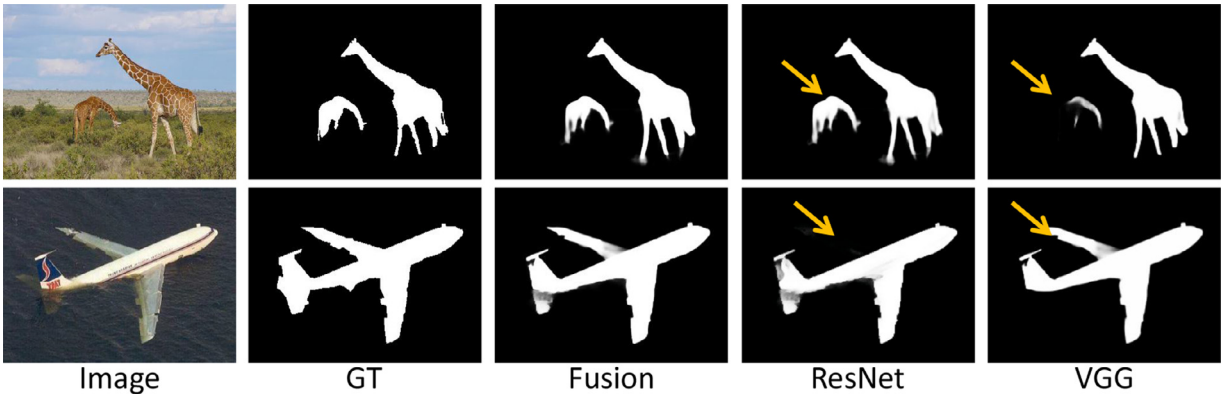

Fig. 2. Deep features in networks with different architectures are generally complementary, in which these feature maps are obtained from the last convolutional layer.

the RFCN is a typical FCN based model; besides, our model follows the bi-stream structure, where our main focus is in the interactions between different feature backbones.

2.2. Multi-stream networks

The recent development of network architecture has a tendency to become deeper and more complicated [31]. Zeiler et al. [32] have demonstrated that a deeper architecture can generally generate more discriminative features at the expense of more complex architecture, leading the network difficult to train. In sharp contrast to the "deeper" strategy, the "wider" architecture may become an intuitive choice, in this paper the term "wider" means to design network architecture with parallel sub-networks. For example, Lin et al. [33] proposed a bilinear architecture, utilizing two feature extractors to obtain multi-scale deep features for image recognition. Saito et al. [34] proposed a novel model for visual question answering, which attempts to learn discriminative features by using two independent sub-networks to conduct feature extraction for multi-source data. Kim et al. [35] proposed to utilize a newly designed parallel feature pyramid network for object detection. Yang et al. [36] present a deep compact code learning solution for efficient cross-modal similarity search. Deng et al. [37] propose a novel strategy to exploit the semantic similarity of the training data and design an efficient generative adversarial framework. Deng et al. [38] propose a novel two-stream ConvNet architecture, which learns hash codes with class-specific representation centers.

Recently, Multi-stream network, which typically has multiple network streams for explicitly learning multi-scale saliency features with different structures, is adopted in the image saliency detection and achieve promising results. Zhao et al. [39] designed a multi-context deep learning framework, in which the parallel revealed global context and local context are combined in an unified deep learning framework to jointly locate the salient object. Wang et al. [40] utilized parallel sub-networks to respectively conduct pixel-level/object-level saliency computation, and then the revealed saliency clues will be fused as the final predictions. Li et al. [41] built a multi-task deep network to explore the common saliency consistency between the salient object detection and the semantic segmentation. Wang et al. [42] design a two-stream network, i.e., a classification network and a caption generation network, to highlight the most important regions for corresponding tasks. Wu et al. [43] propose to integrate features of deeper layers in attention stream to get an initial saliency map, which is used to refine the features of the detection stream to generate the final map.

Actually, the "wider" structure has its merit to balance the trade-off between the saliency performance and the network complexity. However, because the parallel structure adopted by the above-mentioned methods are trained independently, those parallel learned deep features may not be able to effectively complement each other, not to mention those feature conflicts may lead the overall performance even worse.

In contrast to the above-mentioned methods, the proposed model is completely different in two aspects: 1) We utilize a novel recursive learning strategy to train parallel sub-networks to obtain a complementary status between two subnetworks; 2) As for those already learned complementary deep features, we utilize a selective fusion module to ensure an optimal fusion status for high-quality SOD result.

3. Network architecture

Motivation Existing state-of-the-art methods have a tendency to design deeper and more complicated network to improve the SOD performance along with expensive computation overhead. Recently, Zagoruyko et al. [44] suggest that wide residual network is far superior over their commonly used thin and very deep counterparts in terms of computational complexity and accuracy. Though the previous works have demonstrated that a wider network is effective, it has not been fully exploited in salient object detection task. On the other hand, as shown in Fig. 1-A, previous works [45–47] focused on how to effectively aggregate multi-level visual features within a single-stream network, ignoring the connection between different structure network. Saito et al. [34] show that the features extracted from different structure networks contain different information. As shown in Fig. 2, some information should be preserved (or lost) only by VGGNet, whereas some are preserved only by ResNet. Inspired by above-mentioned, we propose to design a bi-stream network consisting of two different subnetworks, in which these subnetworks will potentially be able to provide complementary discriminative saliency clues generated by different models. Our goal is to fully take advantage of complementary information present in different kinds of features.

As shown in Fig. 3, we redesigned the basic convolutional blocks of feature extractor. Compared to the original residual block of ResNet, we designed another parallel branches to mining complementary deep features. In other words, these two parallel subnetworks will focus on different saliency clues by using independent loss function to obtain diversity features.

We utilize $\mathbf{X} = \{\mathbf{X}_i, i \in [1, 5]\}$ to denote the input maps for each convolutional block in the VGG-16 subnetwork, in which the W_i and the b_i respectively represent the predefined kernel and bias. Thus, the learning procedure of our method can be uniformly for-

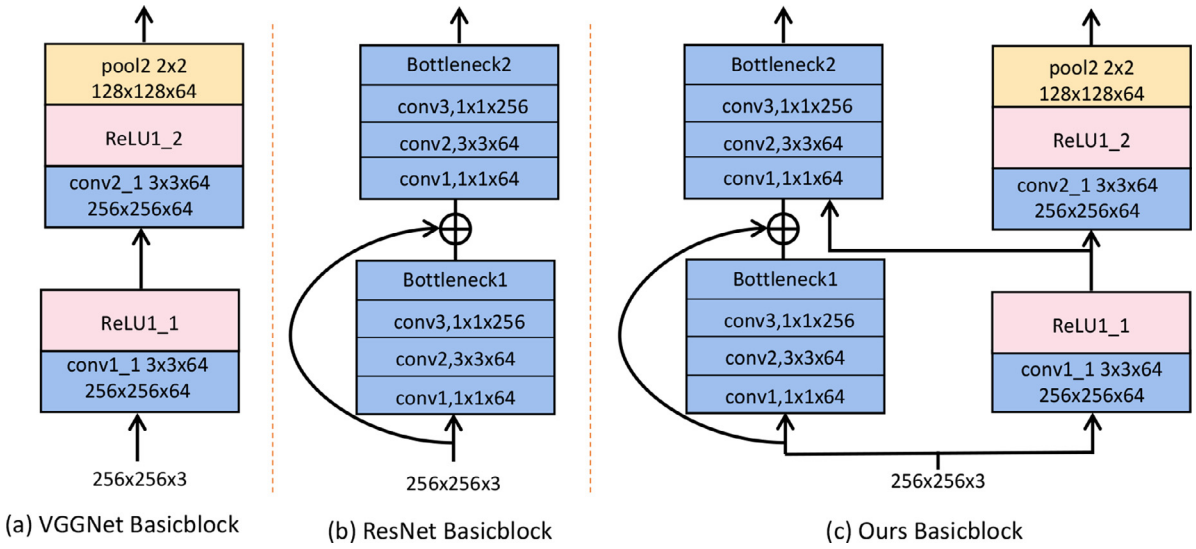

Fig. 3. Basicblock of VGGNet, ResNet and our wide network. Batch normalization and ReLU precede each residual block are omitted for clarity.

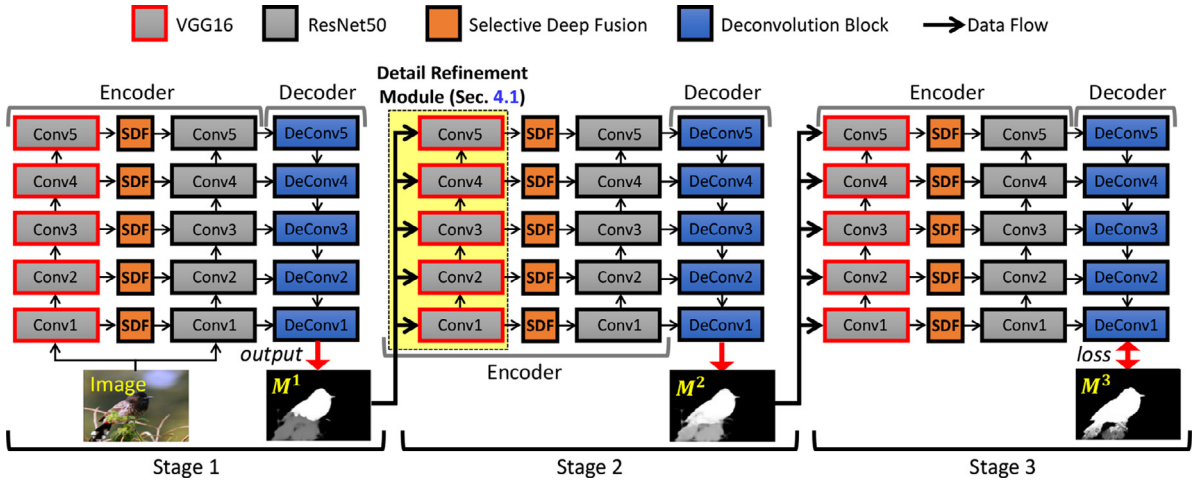

Fig. 4. The pipeline of our proposed method. Our network follows the encoder-decoder style, yet it different from previous methods, in which the encoder consists of two backbones with different structures, i.e., VGG16 and ResNet50. The input image is firstly passed through the encoder to extract multi-scale convolutional deep features. Then, we use both the newly proposed Dense Aggregation Module (Section 4.2) and Selective Deep Fusion Module (Section 5) to make full use the multi-scale deep features which are extracted from VGG16 and ResNet50 respectively. The decoder network takes the multi-scale convolutional features as input to generate a finer saliency prediction $\mathbf{M}^t$, which will latterly be refined by recursively using those low-level deep features in previous stage (Section 4.1). In each learning stage, our method simultaneously uses the detail refinement module (to alleviate the spatial info loss) and the dense aggregation module (to avoid the learning ambiguity) to ensure the complementary status between the parallel sub-networks. When our recursive learning reaches the final stage, we simultaneously feed the last feature layer of ResNet-50 and all side layers of VGG-16 into the selective deep fusion network to produce the final SOD results.

mulated as Eq. (1).

$$\mathbf{X}_{i+1} \leftarrow Conv(\mathbf{X}_i): \ W_i^s * \mathbf{X}_i + b_i, \tag{1}$$

where $Conv(\cdot)$ denotes the convolutional operation and the superscript s denotes the convolutional stride. Similarly, we represent the input features for convolutional blocks in our ResNet-50 subnetwork as $\mathbf{F} = \{\mathbf{F}_i, i \in [1, 5]\}$. Deconvolution layers are to progressively produce the fine-scale saliency score map $\mathbf{M}^t$, where the superscript t denotes the recursive learning stage.

Figure 4 illustrates the overview of the proposed model, which mainly consists of three components: 1) detail refinement module; 2) dense aggregation module; and 3) selective deep fusion module. All these components will cooperate our recursive multi-model deep learning, which will be respectively introduced in the following sections.

4. Inter-model deep fusion

4.1. Detail refinement module

Following the widely used encoder-decoder network architecture, the proposed detail refinement module (DRM) utilizes the ResNet-50 subnetwork to conduct an end-to-end saliency regression for the fine-scale saliency predictions, which will latter be applied to another parallel sub-network (VGG-16) to alleviate the spatial information loss problem, recursively.

Recently, the conventional networks usually adopt multiple convolution and pooling operations for their saliency regression, which easily degrade their performance due to the spatial information progressively vanishes in deep layers. To alleviate it, Hou et al. [14] proposed to resort short connections to integrate multi-level deep features to compensate the lost spatial details. However, deep features obtained by an identical single network have a ten-

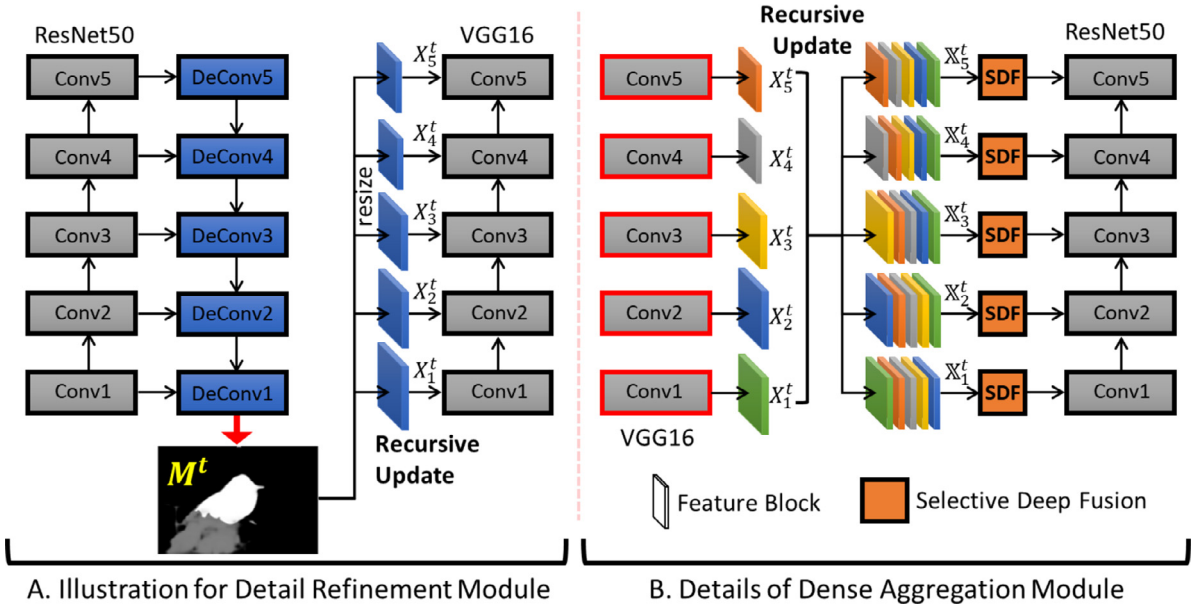

Fig. 5. The illustration of the proposed modules. The subfigure A is the detailed architecture of the detail refinement module (Section 4.1) in the tth stage. We resize the $\mathbf{M}^t$ to the same size of the $\mathbf{X}_i^t$ and concatenate them together by performing convolutional operation. Then, the combined features will be fed into the next stage, obtaining the $\mathbf{M}^{t+1}$ with better details. The subfigure B shows how to convert the multi-level deep features $\mathbf{X}_i^t$ into the integrated feature maps $\mathbb{X}_i^t$, which will latter prepare a set of finer deep features for the next learning stage (Section 4.2).

dency of homogenization, which heavily limits the complementary status between inter-layer deep features.

To address above limitations, we propose to construct dense connections between our parallel networks, see the pictorial demonstration in Fig. 5-A. Since the output of last layer of ResNet-50 can well represent the saliency details, we use it to recursively refine its parallel VGG-16 features ($\mathbf{X}_i^t$, $i \in [1, 2, 3, 4, 5]$). Also, we resize the resolution of $\mathbf{M}^t$ according to each target features $\mathbf{X}_i^t$, and then fuse these linked deep features by using a 3×3 convolution. Here we formulate the recursively fusion procedure as Eq. (2).

$$\mathbf{X}_i^{t+1} \leftarrow \begin{cases} Conv\{Cat(\mathbf{X}_i^t, \uparrow(\mathbf{M}^t))\}, & if\ \xi(\mathbf{M}^t) < \xi(\mathbf{X}_i^t) \\ Conv\{Cat(\mathbf{X}_i^t, \downarrow(\mathbf{M}^t))\}, & if\ \xi(\mathbf{M}^t) > \xi(\mathbf{X}_i^t) \end{cases}, \tag{2}$$

where $\uparrow(\cdot)$ and $\downarrow(\cdot)$ denote the upsampling and downsampling operations respectively. $Cat(\cdot)$ denotes concatenate operation and the function $\xi(\cdot)$ returns the feature size of the given input.

So far, by using Eq. (2), we have utilized the fine-scale saliency predicted by the ResNet-50 sub-network to refine its parallel sub-network VGG-16. Meanwhile, in order to achieve an optimal inter-model complimentary status, those deep features generated by VGG-16 in turn are used to reduce the false positive regions detected by the ResNet branch. Therefore, we recursively update $\mathbf{M}$ ($\mathbf{M}^{t+1} \leftarrow \mathbf{M}^t$) in the ResNet-50 subnetwork.

4.2. Dense aggregation module

Previous works [14,16,29] have shown that a good saliency model should take full advantage of its intermediate multi-level deep features, in which high-level deep features usually concentrate on the high-level semantical information while low-level features frequently focus on the subtle details.

As we all know, the lower-level features contain many spatial details along with non-salient distractors, while the higher-level features focus more on those discriminative regions, such non-salient distractors in deep features are gradually suppressed when the CNNs go deeper. Since the non-salient distractors are in lower-level features, the straightforward fusion strategies (e.g., the point-to-point style [48]) will easily introduce inconsistencies/conflictions. To address this issue, we have devised a novel dense aggregation scheme, which refines each layer of the ResNet branch by integrating all-level features of the VGGNet branch, see the pictorial demonstration in Fig. 5-B. In this way, the distractors hidden in the low-level features will be suppressed effectively.

For each recursive learning stage (i.e., noted by superscript t), we first utilize 1×1 convolution to reduce the feature channel. Thus, we can easily aggregate each feature block $\mathbf{X}_i^t$ to 32 channel feature map $\hat{\mathbf{X}}_i^t$. Then, for each $\hat{\mathbf{X}}_i^t$, we resize $\hat{\mathbf{X}}_j^t$ ($j \neq i$) to be an identical size of $\hat{\mathbf{X}}_i^t$ and aggregate all theses resized feature maps to an identical size of each ResNet-50' feature block $\mathbf{F}_i^t$ by using 1×1 convolution, which can be formulated as Eq. (3).

$$\mathbb{X}_i^t = \begin{cases} Conv\{Cat(\hat{\mathbf{X}}_1^t, \uparrow(\hat{\mathbf{X}}_2^t), \ldots, \uparrow(\hat{\mathbf{X}}_5^t))\} & if\ i = 1 \\ Conv\{Cat(\ldots, \downarrow(\hat{\mathbf{X}}_{i-1}^t), \hat{\mathbf{X}}_i^t, \uparrow(\hat{\mathbf{X}}_{i+1}^t), \ldots)\} & if\ i = \{2, 3, 4\} \\ Conv\{Cat(\downarrow(\hat{\mathbf{X}}_1^t), \ldots, \downarrow(\hat{\mathbf{X}}_4^t), \hat{\mathbf{X}}_5^t)\} & if\ i = 5, \end{cases} \tag{3}$$

where $\uparrow(\cdot)$ and $\downarrow(\cdot)$ respectively denote the upsampling/downsampling operation, $Cat(\cdot)$ denotes the concatenation operation.

In general, those computed deep feature $\mathbb{X}_i^t$ ($i \in \{1, 2, 3, 4, 5\}$) can well represent the intermediate coarse-level saliency clues in the VGG-16 subnetwork, and we recursively aggregate these features into the ResNet-50 subnetwork as Eq. (4).

$$\mathbf{F}_i^{t+1} \leftarrow Conv(\mathbf{F}_i^t, \mathbb{X}_i^t), \tag{4}$$

where $\mathbb{X}_i^t$ denotes the processed ith feature block in ResNet-50 at the t learning stage. Once the ResNet-50' deep features $\mathbf{F}_i^t$ have been updated to $\mathbf{F}_i^{t+1}$, we can achieve more finer saliency map $\mathbf{M}^{t+1}$ accordingly, which will be used to initiate another round of recursively learning in our detail refinement module.

In summary, there are totally three major advantages regarding the proposed dense aggregation module:

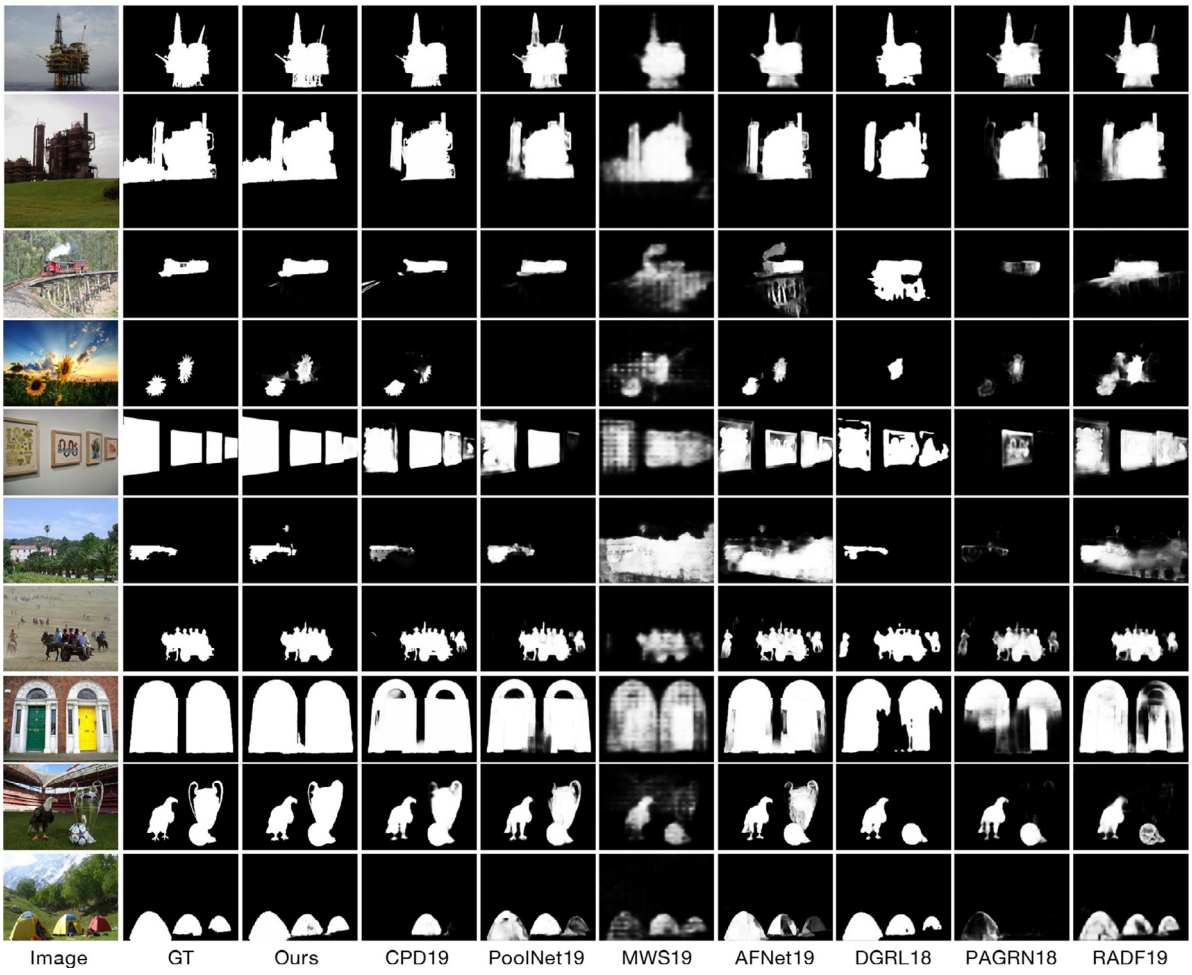

Fig. 6. Visual comparison of saliency maps. Note that GT stands for the groundtruth. Apparently, it can be observed that our proposed model is able to handle diverse challenging scenes.

(1) Each coarse-level deep features generated from VGG-16 facilitate the computation of fine-scale saliency prediction of current ResNet-50 network, which ensures an effective complementary status between our parallel sub-networks;
(2) The proposed dense aggregation scheme can correct the consistency of those intermediate multi-level deep features, making the fine-scale saliency prediction of ResNet-50 network more accurate;
(3) The coarse-level deep features produced by VGG-16 can be treated as attention maps to suppress the false positive regions detected by the ResNet branch.

5. Selective deep fusion

The conventional methods have investigated various handcrafted fusion strategies (e.g., the multiplicative based ones, the additive based ones, and the maximum combination based ones) to combine saliency clues which are revealed at different spatial-levels. However, these methods are elaborately designed for certain types of image scenes, which may fail to generalize well in other image scenes. To this end, we propose to utilize a newly designed selective deep fusion to address the above-mentioned limitation.

Concretely, when CNNs extract multi-level features from an input image, the distractors in features are gradually suppressed as CNNs go deeper. Since there are many distractors in lower-level features, we propose a novel selective deep fusion module, which refines each layer feature of one branch by integrating high-level features of the other branch. On the other hand, considering previous stage $\mathbf{M}^{t-1}$ contains both high-level semantic information (e.g., location) and low-level class-agnostic details information (e.g., edge), we combine the fine-scale saliency clue ($\mathbf{M}^{t-1}$) with our proposed selective deep fusion module, which can be formulated as Eq. (5).

$$\mathbf{S}_l^t = \sum_{i=l}^{5} Conv\Big(TF(\mathbf{M}^{t-1}) \otimes TF(\mathbb{X}_i^t)\Big), \tag{5}$$

where TF is a scale transformation operation along with a 1×1 convolutional layer with 32 output channel number, which aims to ensure the spatial size consistent with the corresponding $\mathbf{F}_l^t$. $\otimes$ is element-wise multiplication. $\mathbf{S}_l^t$ denotes the fused feature, which intrinsically contains complementary saliency clues. After obtaining $\mathbf{S}_l^t$, we feed it into another branch to learn complementary saliency clues. In this selective deep fusion module, the inputs ($\mathbb{X}_i^t$ and $\mathbf{M}^{t-1}$) usually vary between different levels of SDF. For example, the lowest-level SDF takes all levels (i.e., 5 levels) of deep features $\{\mathbb{X}_i^t, i = 1, 2, \ldots, 5\}$ as input, while the top-level SDF only takes a single $\mathbb{X}_5^t$ as input. This makes the state of the proposed SDF explicitly discriminative at different levels because each SDF has its own weights. Thus, we should assign one individual SDF module for each level of the proposed network. Besides, the selec-

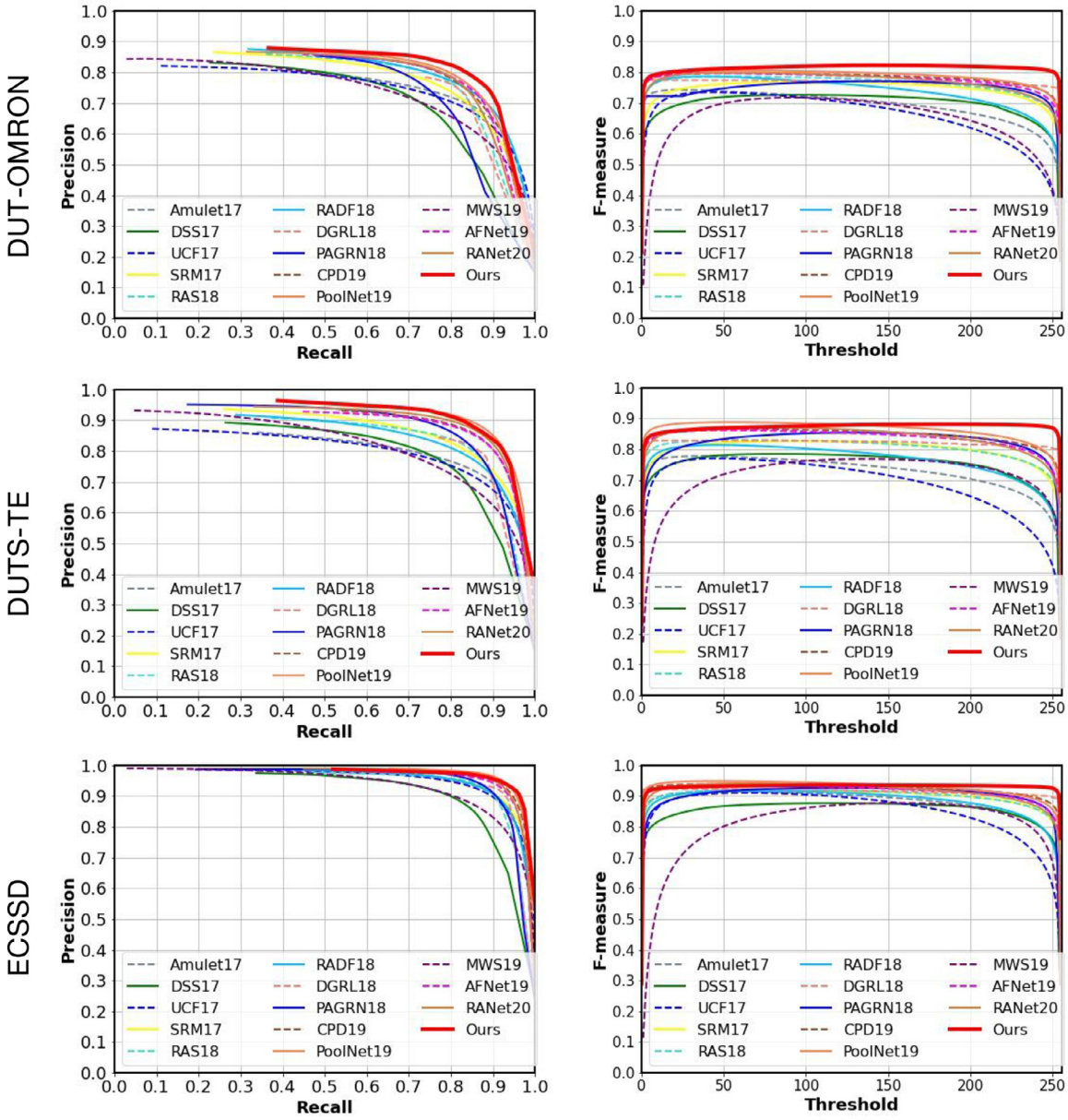

Fig. 7. Quantitative comparisons (Pecision-recall curves and F-measure curves) between our method and 14 state-of-the-art methods over 5 adopted datasets, in which the left part is the Precision-recall curve and the right part is the F-measure curve. Due to the limitation of space, we only provide the quantitative results over 3 datasets here, and the remaining 3 results can be found in Fig. 8.

Table 1
Details of our selective deep fusion module, in which the "DeC." denotes the deconvolution and the "ConvC." denotes the convolution and classifier. For simplicity, we have omitted the channel number of the "output" because they have an identical channel number (i.e., 64), excepting for the last ConvC which has 2 channels only.

Layers	Conv1	Conv2	Conv3	Conv4	DeC.4	DeC.3	DeC.2	DeC.1	ConvC.
Kernel	33	3 × 3	3 × 3	3 × 3	3 × 3	3 × 3	3 × 3	3 × 3	1 × 1
Channel	64	64	64	64	64	64	64	64	2
Output	256 × 256	128 × 128	64 × 64	32 × 32	32 × 32	64 × 64	128 × 128	256 × 256	256 × 256

tive fusion costs less computation overhead than point-to-point fusion strategy. Moreover, the performance also achieves consistently increase because less distractors have been introduced in feature integration.

We show the architecture details of the proposed selective deep fusion module in Table 1. Actually, this module mainly consists of two components: the encoder block and the decoder block. The encoder block are composed of 4 convolutional layers. Each of these convolution layers is followed by a batch normalization and a ReLU activation function. Meanwhile, we assign each encoder layer with one corresponding decoder without using ReLU.

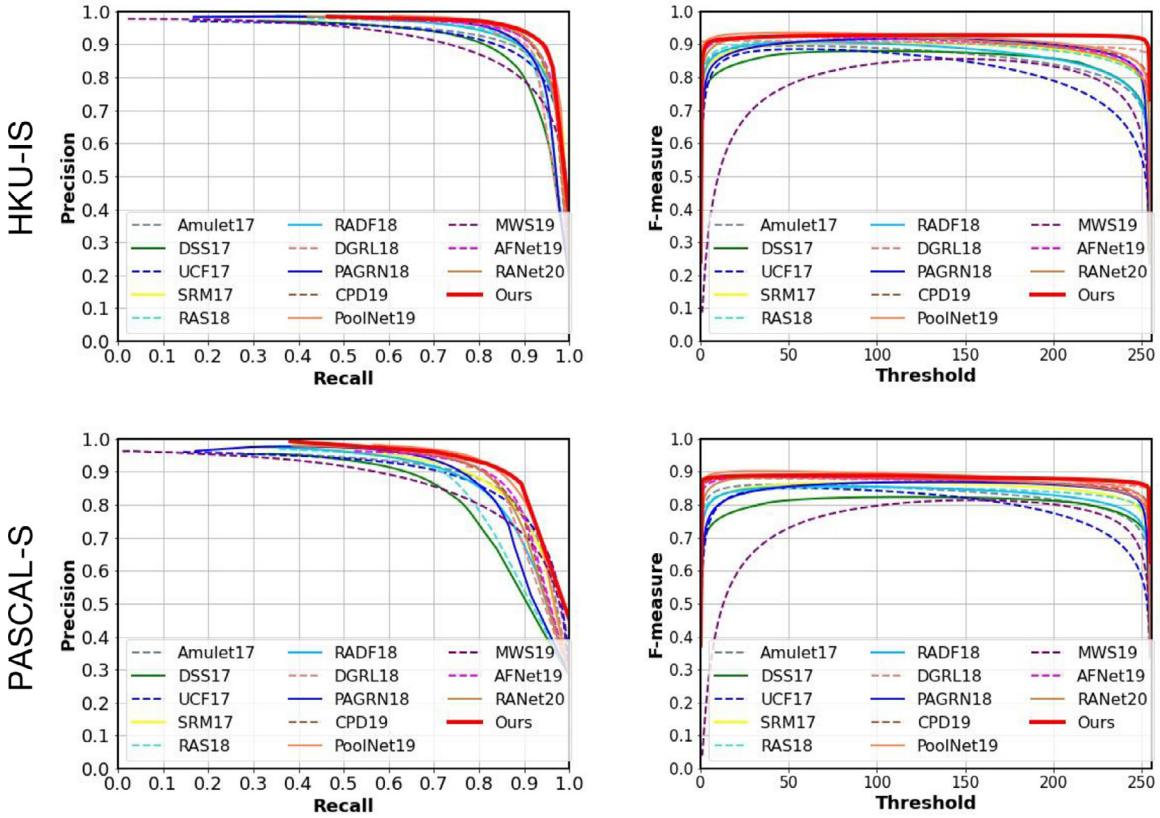

Fig. 8. Continued Quantitative comparisons (Precision-recall curves and F-measure curves) between our method and 14 state-of-the-art methods over 5 adopted datasets.

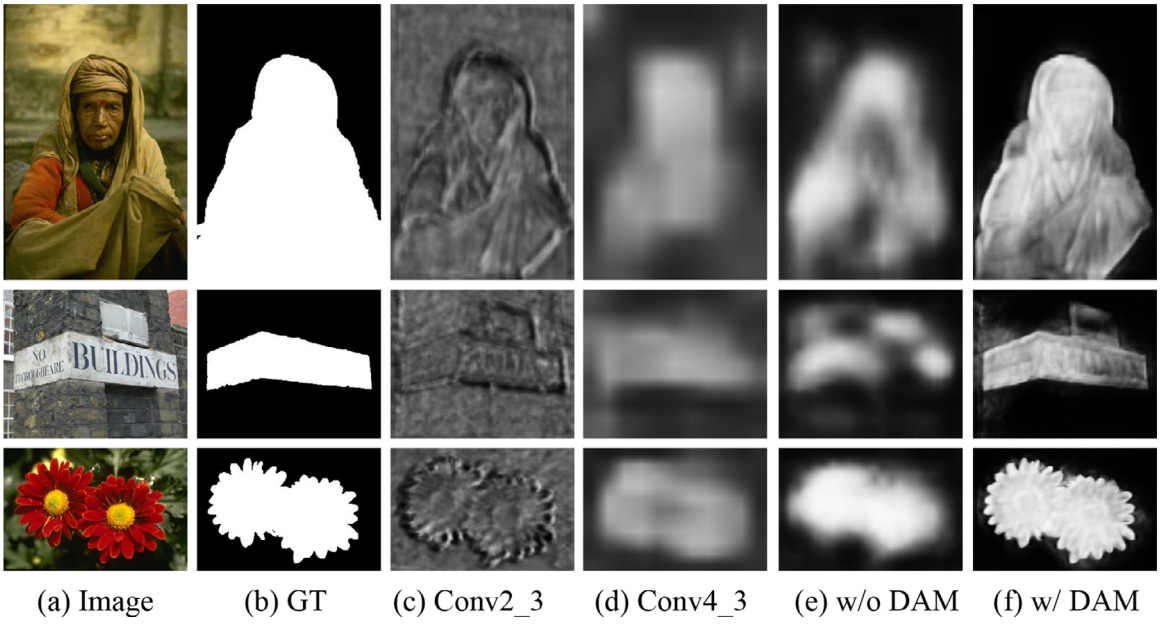

Fig. 9. Deep feature visualization. (c) and (d) respectively show the low-level and high-level features; (f) is the fused features by DAM.

6. Experiments and results

6.1. Implementation details

The proposed method is developed on the public deep learning framework Caffe. We run our model in a quad-core PC with an i7-6700 CPU (3.4 GHz and 8 GB RAM) and an NVIDIA GeForce GTX 1080 GPU (with 8G memory). Our model is trained on the MSRA10K dataset. Then, we test our model on other datasets. Due to the limited GPU memory, we set the mini-batch size to 4. We use the stochastic gradient descent (SGD) method to train with a momentum 0.99, and the same weight decay 0.0005. Meanwhile, for our feature integration module, we use SGD with a momentum 0.9, and weight decay 0.0005. We set the learning rate as 10^{-8} and it reduces by a factor of 0.1 at 10k iterations. The training process of our model takes about 14 hours. During testing, the proposed

Table 2
Comparison of quantitative results including the max F-measure, S-measure and MAE on five well-known SOD benchmarks: DUT-OMRON [57], DUTS-TE [39], ECSSD [58], HKU-IS [39] and PASCAL-S [59]. The top three results are highlighted in bold, italic, and bold-italic fonts, respectively.

Method	DUT-OMRON			DUTS-TE			HKU-IS			ECSSD			PASCAL-S		
	maxF_β	S-m	MAE	maxF_β	S-m	MAE	maxF_β	S-m	MAE	maxF_β	S-m	MAE	maxF_β	S-m	MAE
Ours	**0.824**	**0.843**	**0.053**	*0.887*	**0.895**	*0.041*	**0.941**	**0.925**	**0.028**	*0.945*	**0.928**	**0.031**	*0.895*	**0.861**	*0.078*
RANet20 [49]	***0.799***	0.825	0.058	***0.874***	***0.874***	***0.044***	***0.928***	***0.908***	0.036	***0.941***	0.917	0.042	0.866	*0.847*	*0.078*
R^2Net20 [50]	0.793	0.824	0.061	0.855	0.861	0.050	0.921	0.903	0.039	0.935	0.915	0.044	0.864	*0.847*	**0.075**
CPD19 [43]	0.797	0.825	***0.056***	0.865	0.868	***0.044***	0.925	0.906	***0.034***	0.939	***0.918***	***0.037***	0.884	0.828	0.089
PoolNet19 [51]	*0.805*	*0.831*	*0.054*	**0.889**	*0.886*	**0.037**	*0.936*	*0.919*	*0.030*	**0.949**	*0.926*	*0.035*	**0.902**	*0.847*	***0.081***
AFNet19 [52]	0.797	***0.826***	0.057	0.862	0.866	0.046	0.923	0.905	0.036	0.935	***0.918***	0.042	***0.885***	***0.833***	0.086
MWS19 [42]	0.718	0.756	0.109	0.769	0.757	0.092	0.856	0.818	0.084	0.878	0.828	0.096	0.814	0.753	0.151
PAGRN18 [53]	0.771	0.775	0.071	0.855	0.837	0.056	0.918	0.887	0.048	0.927	0.889	0.061	0.869	0.793	0.115
DGRL18 [54]	0.774	0.806	0.062	0.829	0.841	0.050	0.911	0.895	0.036	0.922	0.903	0.041	0.881	0.828	0.082
RADF18 [16]	0.786	0.813	0.072	0.814	0.824	0.072	0.907	0.888	0.050	0.917	0.895	0.060	0.857	0.797	0.119
RAS18 [55]	0.787	0.814	0.062	0.831	0.838	0.060	0.913	0.887	0.045	0.921	0.893	0.056	0.852	0.774	0.125
SRM17 [15]	0.769	0.798	0.069	0.827	0.835	0.059	0.906	0.888	0.046	0.917	0.895	0.054	0.868	0.817	0.100
Amulet17 [29]	0.743	0.781	0.098	0.778	0.803	0.085	0.896	0.883	0.052	0.915	0.894	0.059	0.862	0.820	0.103
UCF17 [56]	0.735	0.758	0.132	0.771	0.778	0.118	0.886	0.866	0.074	0.911	0.883	0.078	0.851	0.808	0.128
DSS17 [14]	0.727	0.748	0.092	0.785	0.790	0.081	0.880	0.852	0.067	0.877	0.836	0.090	0.824	0.749	0.144

Table 3
The number of model size, FLOPs and parameters comparisons of our method with 3 state-of-the-art models.

Method	Model(MB)	Encoder(MB)	Decoder(MB)	FLOPs(G)	Params(M)
Ours	263.7	138.5	125.2	74.23	69.48
PoolNet19 [51]	278.5	94.7	183.8	88.91	68.26
BASNet19 [63]	348.5	87.3	261.2	127.32	87.06
DGRL18 [54]	573	95.6	477.4	215.62	146.37

model runs about 14 FPS with 256 × 256 resolution. We assign the number of recursive stage N=3 according to the qualitative results demonstrated in Fig. 10.

The performance improvements of our method are mainly brought by the newly-designed multi-model fusion scheme and we can implement the parallel subnetworks using "simple" networks. For each subnetwork, the complexity and memory requirements are more than the conventional single network cases, while the overall complexity and memory requirements for our parallel subnetworks is comparable to the mainstream single network cases (see in Table 3).

Algorithm 1 to show the details of our training process. We set

Algorithm 1 Training Procedure.

Require: Training data $I = \{(I_i, y_i)\}_{i=1}^{N}$; Max epoch number N=100; Number of iterations: T;
1: **for** $t = 1, \ldots, N$ **do**
2: **for** i= 1, …, T **do**
3: Data-loading: image, gt = DataLoader(I);
4: Predicting: pred = Model(image);
5: Computing Loss: loss= BCELoss(pred, gt);
6: Backpropagate loss and updating parameters: loss.backward().
7: **end for**
8: **end for**

the max epoch number N=100, and the iteration number T varies with the training data and batch size. As shown in Algorithm 1, we first first load the training set using the DataLoader(). Next, we begin to train the defined model using the binary cross entropy loss (BCEL). Finally, at the end of each iteration, we will back-propagate the loss and update the network parameters. The above procedure will be repeated until reaching the max epoch number.

6.2. Datasets and evaluation metrics

We have evaluated our method on 5 widely used publicly available datasets, including DUT-OMRON [57], DUTS-TE [39], ECSSD [58], HKU-IS [39] and PASCAL-S [59]. DUT-OMRON contains 5168 high-quality images. Images of this dataset have one or more salient objects with complex backgrounds. DUTS-TE has 5019 images with high-quality pixel-wise annotations, which is selected from the currently largest SOD benchmark DUTS. ECSSD has 1000 natural images, which contain many semantically meaningful and complex structures. As an extension of the complex scene saliency dataset, ECSSD is obtained by aggregating the images from BSD [60] and PASCAL VOC [61]. HKU-IS contains 4447 images. Most of the images in this dataset have low contrast with more than one salient object. PASCAL-S contains 850 natural images with several objects, which are carefully selected from the PASCAL VOC dataset with 20 object categories and complex scenes.

We have adopted 4 commonly used standard metrics to evaluate our method, including Precision-Recall curve, F-measure, S-measure [62], and Mean Absolute Error (MAE).

6.3. Comparison with the state-of-the-art methods

We have compared our method with 14 most representative SOTA methods, including Amulet17 [29], DSS17 [14], UCF17 [56], SRM17 [15], RAS18 [55], RADF18 [16], PAGRN18 [53], DGRL18 [54], MWS19 [42], CPD19 [43], AFNet19 [52], PoolNet19 [51], RANet20 [49] and R^2Net20 [50]. For all of these methods, we use the original codes with recommended settings or the saliency maps provided by the authors. Moreover, our results are diametrically generate by model without relying on any post-processing and all the predicted saliency maps are evaluated with the same evaluation code.

Quantitative Comparisons. As a commonly used quantitative evaluation venue, we first investigate our model using the precision-recall curves. As shown in the left of Figs. 7 and 8, our model can consistently outperform the state-of-the-art models on

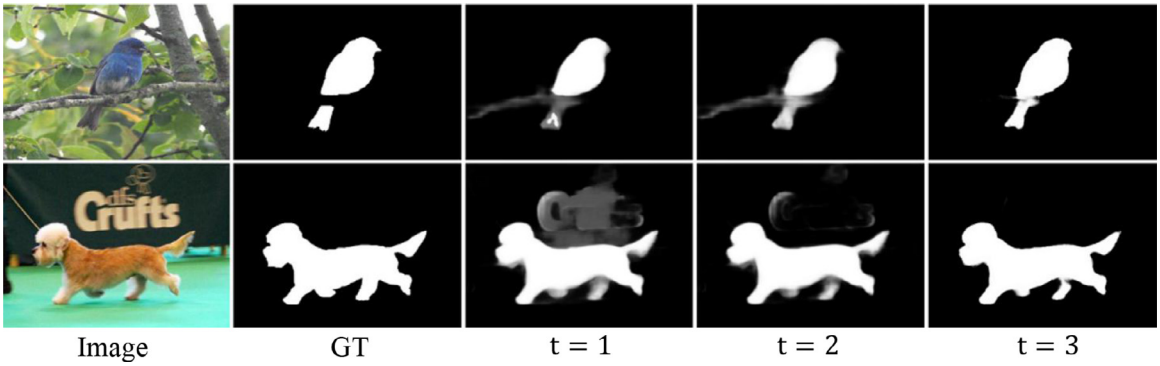

Fig. 10. Qualitative illustration of our recursive learning scheme, where t denotes the saliency maps obtained at different learning stages.

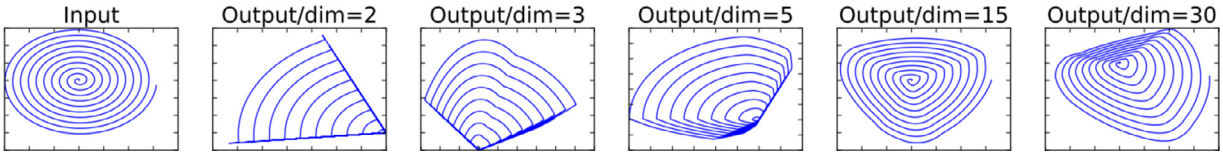

Fig. 11. Examples of ReLU transformations of low-dimensional manifolds embedded in higher-dimensional spaces. In these examples the initial spiral is embedded into an n-dimensional space using random matrix T followed by ReLU, and then projected back to the 2D space using T^{-1}. In examples above $n = 2, 3$ result in information loss where certain points of the mainfold collapse into each other, while for $n = 15$ to 30 the transformation is highly non-convex.

Table 4
Runtime comparison (GPU time) with previous deep learning based saliency models.

Method	Ours	CPD19	AFNet19	DGRL18	RADF18	SRM17	Amulet17	UCF17	DSS17	RFCN18
Time(s)	0.073	0.063	0.062	0.150	0.154	0.070	0.093	0.134	0.201	4.72

Table 5
Ablation study of our model on DUT-OMRON [57], DUTS-TE [39], ECSSD [58], and HKU-IS [39]. We change one component at a time, to assess individual contributions. VGGNet and ResNet are used as the backbone. DRM is Details Refinement Module, DAM denotes Dense Aggregation Module and SDF stand for Selective Deep Fusion Module.

Configurations	DUT-OMRON			DUTS-TE			HKU-IS			ECSSD		
	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓
Baseline VGG16	0.743	0.734	0.078	0.798	0.824	0.069	0.865	0.864	0.050	0.871	0.853	0.053
Baseline ResNet50	0.756	0.746	0.076	0.813	0.832	0.065	0.876	0.862	0.049	0.884	0.870	0.047
VGG16+ResNet50+DRM	0.772	0.793	0.065	0.846	0.864	0.057	0.893	0.905	0.046	0.910	0.884	0.042
VGG16+ResNet50+DRM+DAM	0.803	0.821	0.061	0.863	0.876	0.054	0.925	0.917	0.034	0.924	0.903	0.037
VGG16+ResNet50+DRM+DAM+SDF	**0.824**	**0.843**	**0.053**	**0.887**	**0.895**	**0.041**	**0.941**	**0.925**	**0.028**	**0.945**	**0.928**	**0.031**
VGG16+ResNet50+DRM+SDF	0.806	0.824	0.061	0.860	0.862	0.049	0.922	0.904	0.036	0.934	0.911	0.043
Stage1	0.791	0.817	0.061	0.853	0.872	0.049	0.923	0.894	0.037	0.916	0.886	0.040
Stage2	0.816	0.836	0.056	0.872	0.879	0.044	0.935	0.905	0.032	0.928	0.906	0.038
Stage3	**0.824**	**0.843**	**0.053**	**0.887**	**0.895**	**0.041**	**0.941**	**0.925**	**0.028**	**0.945**	**0.928**	**0.031**

Table 6
Ablation study for different scale input. For example, $\{S_{ResNet} = 1, S_{VGG} = 0.5\}$ denotes that the ResNet branch takes the whole image as input while the VGG branch reduces the image size by half.

Configrations	DUT-OMRON			DUTS-TE			HKU-IS			ECSSD			PASCAL-S		
	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓
$S_{ResNet} = 1, S_{VGG} = 1$	**0.824**	**0.843**	**0.053**	**0.887**	**0.895**	**0.041**	**0.941**	**0.925**	**0.028**	**0.945**	**0.928**	**0.031**	**0.895**	**0.861**	**0.078**
$S_{ResNet} = 0.5, S_{VGG} = 1$	0.813	0.837	0.062	0.873	0.888	0.047	0.935	0.921	0.032	0.926	0.915	0.034	0.878	0.855	0.085
$S_{ResNet} = 0.25, S_{VGG} = 1$	0.804	0.824	0.063	0.867	0.874	0.049	0.924	0.918	0.035	0.923	0.912	0.036	0.867	0.846	0.087
$S_{ResNet} = 1, S_{VGG} = 0.5$	0.821	0.836	0.056	0.882	0.887	0.046	0.934	0.912	0.032	0.938	0.925	0.035	0.887	0.854	0.083
$S_{ResNet} = 1, S_{VGG} = 0.25$	0.817	0.832	0.057	0.874	0.877	0.045	0.927	0.905	0.033	0.934	0.920	0.037	0.883	0.853	0.085

Table 7
Quantitative comparisons with other state-of-the-arts in term of F-measure (larger is better) and MAE (smaller is better) on five dataset. The best results are shown in bold. Ours-D represents for training on DUTS while Ours-DH represents for training on DUTS and HRSOD.

Configrations	HRSOD-Test			DAVIS-S			DUTS-Test			HKU-IS			THUR		
	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓
Ours-DH	**0.888**	**0.897**	**0.030**	**0.888**	**0.876**	**0.026**	0.791	0.822	**0.051**	0.886	0.877	**0.042**	**0.749**	**0.826**	**0.064**
Ours-D	0.857	0.876	0.040	0.850	0.875	0.029	**0.796**	**0.827**	0.052	**0.891**	**0.882**	0.042	0.740	0.820	0.067

Table 8
Quantitative comparison of the models with or without using the ReLU function in the proposed selective deep fusion module.

Configrations	DUT-OMRON			DUTS-TE			HKU-IS			ECSSD			PASCAL-S		
	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓	maxF_β ↑	S-m ↑	MAE ↓
w/o ReLU	**0.824**	**0.843**	**0.053**	**0.887**	**0.895**	**0.041**	**0.941**	**0.925**	**0.028**	**0.945**	**0.928**	**0.031**	**0.895**	**0.861**	**0.078**
w/ ReLU	0.822	0.840	0.053	0.882	0.889	0.042	0.938	0.924	0.028	0.943	0.925	0.033	0.891	0.856	0.079

all tested benchmark datasets. Specifically, the proposed model outperforms other models on DUT-OMRON datasets. Meanwhile, our model also is evaluated by F-measure curves as shown in the right of Figs. 7 and 8, which also demonstrates the superiority of our method. The detailed F-measure, MAE values are provided in Table 2, in which our method also performs favorably against other state-of-the-art approaches. As for the DUT-OMRON dataset, our model achieves 82.4% in max F-measure and 0.053 in MAE while the second best (PoolNet19) achieves 80.5% in max F-measure and 0.054% in MAE. Also, similar tendencies can be found in the HKU-IS dataset, which is one of the most challenge datasets. Compared to the recent published RANet20, our model increases 1.3% in max F-measure and decreases 8% in MAE.

Qualitative Comparisons. We demonstrate the qualitative comparisons in Fig. 6. The proposed method not only detects the salient objects accurately and completely, but preserves subtle details well. Specifically, the proposed model can adapt to various scenarios as well, including the small object case (row 3), the object occlusion case (raw 6), the complex background case (row 7), and the low contrast case (row 9). Moreover, our method can consistently highlight the foreground regions with sharp object boundaries.

Running Time and Model Complexity Comparisons. Table 4 shows the running time comparisons. This evaluation was conducted on the same machine with an i7-6700 CPU and a GTX 1080 GPU, in which our model achieves 14 FPS. Besides, Table 3 shows the model complexity comparisons, in which we may easily notice that most of the previous single-stream models resort to heavy decoders. For example, the total model size of the top-tier PoolNet is about 278MB, while its decoder part (about 183MB) takes more than half of the total size. In sharp contrast, the decoder size of our proposed method is only 125MB.

6.4. Component evaluation

To validate the effectiveness of our method, we have evaluated several key components of the proposed model on the DUT-OMRON, DUTS-TE, ECSSD and HKU-IS dataset. We start with two single-stream networks and progressively extend it with our newly designed modules, including the parallel backbones, the detail refinement module, the dense aggregation module and the selective deep fusion module.

As shown in Table 5, our newly designed parallel architecture equipped with detail refinement module only (see the 3rd row) can achieve much better performance than the single sub-network (the 1st row and 2nd row). Meanwhile, the overall performance of the proposed parallel architecture with dense aggregation module can get the overall performance further improved, see the 4th row in Table 5. Specially, we notice that the proposed selective deep fusion module obtains a significant performance improvement, see the 5th row. All these results have demonstrated the effectiveness of the proposed method.

For a better understanding, we also provide some qualitative demos in Fig. 9. The lower-level features (Fig. 9-c) usually contain many spatial details along with the non-salient distractors, while the higher-level features (Fig. 9-d) tend to focus more on the most discriminative regions, and the non-salient distractors would be gradually suppressed when the CNNs go deeper. The proposed DAM is able to take full advantage of both low-level and high-level features simultaneously, where the fused features (Fig. 9-f) can well suppress non-salient noises and retain spatial details simultaneously. As a result, it reveals the consistency of multi-level features, showing that the deep model with DAM can guarantee a better consistency between the multi-level features.

6.5. Recursive learning validation

As described in Section 4, our method is trained in a recursive manner. To validate the effectiveness of our stage-wise recursive learning scheme, we perform a detailed comparison of the proposed model at different recursive learning stages using max F-measure, S-measure and MAE scores. As shown in the last three rows of Table 5, the overall performance of our method becomes better as the stage-wise recursive learning continues, and the corresponding qualitative demonstrations can be found in Fig. 10.

6.6. Ablation study for different scale input

Considering that the M cells contribute to the low-level transient processing while the P cells contribute more to the high-level recognition tasks, we also investigate the effectiveness adopting different input scales for different networks. As shown in Table 6, we have newly conducted a series of quantitative experiments to validate it.

Instead of being beneficial to the SOD task, the experimental results show that different input scale for different network decreases the overall performances. The main reason could be that, as demonstrated by the previous work [64], the low-resolution image usually shows unimpressive representation, which is mainly induced by its limited information towards the SOD task (e.g., blurry object boundaries). In the proposed network, the VGG branch was designed to play a role of coarse localization, and thus it can ensure good performance (about 0.5% degeneration in its performance) when assigning a small scale to its features. Meanwhile, compared with the VGG sub-network, the ResNet branch that is designed for tiny saliency details has a significant performance degeneration (about 1%) due to a low-resolution input.

To make our experimental results more convincing, we would like to resort to third-party experimental results. Zeng et al. [64] proposed a high-resolution salient object detection dataset (HRSOD), aiming to solve the inherited defects (i.e., blurry boundary) of low-resolution image. And, their experiments demonstrate that the HRSOD dataset can improve the overall performance by a large margin. For convenience, we show the experiment results in Table 7. Their experiment results partly support our conclusion: a deep model trained on a high-resolution dataset can achieve better performance than the model trained on a low-resolution dataset. Thus, we prefer to inputs with the higher resolutions. Besides, using different scales of input for different networks would degenerate the overall performance, because, in our method, the multi-scale information has been fully considered from the perspective of the deep features, and using different input scales would limit the interactions between the multi-scale deep features.

6.7. The effectiveness of ReLU in SDF

In our SDF module, we get rid of the ReLU in decoder part. The role of the ReLU is to enhance the non-linear perception ability of the neural network. Unfortunately, the ReLU is not always playing a positive role in improving neural networks' capacities. For example, Sandler el al. [65] proposed the famous MobileNetV2 with the linear bottleneck, which discarded the ReLU in the bottleneck layer, where the authors have fully proved that using ReLU in the bottleneck layer indeed hurting the overall performance. On the one hand, as can be seen in Fig. 11, ReLU is capable of preserving complete information in a high-dimensional subspace, and it inevitably loses information in a low-dimensional subspace. On the other hand, to save the computational cost, we compressed the decoder feature channels to 32. These two insights provide us with an empirical hint: we can capture low-dimensional information by applying linear layers without ReLU. In addition, our experimental evidence (in Table 8) also suggests that using linear fusion is crucial as it prevents nonlinearities from destroying too much information.

6.8. Limitations

Compared with previous works, our method can capture more powerful saliency clues from different saliency perspective while avoiding the obstinate feature conflictions by using the proposed multi-model fusion scheme. In the clutter background case, our method can well suppress those non-salient regions and preserves subtle salient details, which is proved by the increased precision rate and F-measure score in Fig. 7 and Fig. 8. Nonetheless, we have noticed a slight decrease regarding the average recall rate, which is mainly induced by an unbalanced bias in our multi-model fusion when computing those complementary deep features. Another limitation of our model is the computational overhead for the stage-wise training. In the future, we plan to explore a more efficient fusion approach by using the off-the-shelf model compression techniques to alleviate the computational burden.

7. Conclusion

In this paper, we proposed a novel multi-model fusion scheme, in which two parallel subnetworks are coordinated to learn complementary deep features recursively. The key rationale of our proposed method is to take full advantage of the complementary features encoded in different subnetworks, revealing saliency clues from different perspectives. To achieve this goal, we newly design three components: 1) Detail Refinement Module, 2) Dense Aggregation Module, and 3) Selective Deep Fusion Module. Specifically, we propose a detail refinement module to recursively compensate for the lost spatial details, and the dense aggregation module is designed to make full use of multi-level deep features. Meanwhile, we propose a selective deep fusion module to effectively fuse complementary information encoded in different sub-branches. Experiments show that the proposed model outperforms existing state-of-the art algorithms on five benchmark datasets.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research is supported in part by National Natural Science Foundation of China (No. 61190120, 61190124, 61190125, 61300067, 61672077, 6167214, 61602341, 61532002 and 61772277), Open Project Program of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (VRLAB2021A05), the Natural Science Foundation of Shandong Province (ZR2019BF011), Research Unit of Virtual Human and Virtual Surgery, Chinese Academy of Medical Sciences (2019RU004), and National Science Foundation of USA (No. NSF IIS-1715985 and IIS-1812606).

References

[1] X. Zheng, Z. Zha, L. Zhuang, A feature-adaptive semi-supervised framework for co-saliency detection, in: ACM International Conference on Multimedia, 2018, pp. 959–966.
[2] S. Hong, T. You, S. Kwak, B. Han, Online tracking by learning discriminative saliency map with convolutional neural network, in: International Conference on Machine Learning, 2015, pp. 597–606.
[3] X. He, Y. Peng, J. Zhao, Fine-grained discriminative localization via saliency-guided faster R-CNN, in: ACM International Conference on Multimedia, 2017, pp. 627–635.
[4] C. Chen, G. Wang, C. Peng, X. Zhang, H. Qin, Improved robust video saliency detection based on long-term spatial-temporal information, IEEE Trans. Image Process. 29 (8) (2019) 1090–1100.
[5] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: neural image caption generation with visual attention, in: International Conference on Machine Learning, 2015, pp. 2048–2057.
[6] H. Fang, S. Gupta, F. Iandola, R.K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J.C. Platt, et al., From captions to visual concepts and back, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 1473–1482.
[7] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, 3-D object retrieval and recognition with hypergraph analysis, IEEE Trans. Image Process. 21 (9) (2012) 4290–4303.
[8] Y. Lin, Z. Pang, D. Wang, Y. Zhuang, Task-driven visual saliency and attention-based visual question answering, (2017) arXiv:1702.06700
[9] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: a neural image caption generator, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 3156–3164.
[10] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1254–1259.
[11] C. Chen, S. Li, H. Qin, A. Hao, Real-time and robust object tracking in video via low-rank coherency analysis in feature space, Pattern Recognit. 48 (2015) 2885–2905.
[12] C. Chen, S. Li, A. Hao, H. Qin, Robust salient motion detection in non-stationary videos via novel integrated strategies of spatio-temporal coherency clues and low-rank analysis, Pattern Recognit. 52 (2016) 410–432.
[13] X. Shen, Y. Wu, A unified approach to salient object detection via low rank matrix recovery, in: IEEE International Conference on European Conference on Computer Vision, 2012, pp. 853–860.
[14] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, P. Torr, Deeply supervised salient object detection with short connections, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2017, pp. 5300–5309.
[15] T. Wang, A. Borji, L. Zhang, P. Zhang, H. Lu, A stagewise refinement model for detecting salient objects in images, in: IEEE International Conference on Computer Vision, 2017, pp. 4019–4028.
[16] X. Hu, L. Zhu, J. Qin, C. Fu, P. Heng, Recurrently aggregating deep features for salient object detection, in: AAAI Conference on Artificial Intelligence, 2018, pp. 6943–6950.
[17] W. Merigan, J. Maunsell, How parallel are the primate visual pathways? Annu. Rev. Neurosci. 16 (1) (1993) 369–402.
[18] P.H. Schiller, N.K. Logothetis, E.R. Charles, Parallel pathways in the visual system: their role in perception at isoluminance, Neuropsychologia 29 (6) (1991) 433–441.
[19] Q. Wang, L. Zhang, W. Zou, K. Kpalma, Salient video object detection using a virtual border and guided filter, Pattern Recognit. 97 (2020).
[20] Q. Zhang, Z. Huo, Y. Liu, Y. Pan, C. Shan, J. Han, Salient object detection employing a local tree-structured low-rank representation and foreground consistency, Pattern Recognit. 92 (2019) 119–134.
[21] P. Zhang, W. Liu, Y. Lei, H. Lu, Hyperfusion-Net: hyper-densely reflective feature fusion for salient object detection, Pattern Recognit. 93 (2019) 521–533.
[22] A. Borji, M. Cheng, H. Jiang, J. Li, Salient object detection: a benchmark, IEEE Trans. Image Process. 24 (12) (2015) 5706–5722.
[23] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, Salient object detection in the deep learning era: an in-depth survey, Comput. Vis. Pattern Recognit. (2019). arXiv
[24] A. Borji, M. Cheng, Q. Hou, H. Jiang, J. Li, Salient object detection: a survey, Comput. Visual Media 5 (2) (2019) 117–150.
[25] G. Li, Y. Yu, Visual saliency based on multiscale deep features, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 5455–5463.
[26] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[27] N. Liu, J. Han, DHSNet: deep hierarchical saliency network for salient object detection, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2016, pp. 678–686.

[28] B. Wang, Q. Chen, M. Zhou, Z. Zhang, X. Jin, K. Gai, Progressive feature polishing network for salient object detection, in: AAAI Conference on Artificial Intelligence, 2019.
[29] P. Zhang, D. Wang, H. Lu, H. Wang, X. Ruan, Amulet: aggregating multi-level convolutional features for salient object detection, in: IEEE International Conference on Computer Vision, 2017, pp. 202–211.
[30] L. Wang, L. Wang, H. Lu, P. Zhang, X. Ruan, Saliency detection with recurrent fully convolutional networks, in: European Conference on Computer Vision, 2016, pp. 825–841.
[31] Y. Tang, X. Wu, Salient object detection with chained multi-scale fully convolutional network, in: ACM International Conference on Multimedia, 2017, pp. 618–626.
[32] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, 2014, pp. 818–833.
[33] T.-Y. Lin, A. RoyChowdhury, S. Maji, Bilinear CNN models for fine-grained visual recognition, in: IEEE International Conference on Computer Vision, 2015, pp. 1449–1457.
[34] K. Saito, A. Shin, Y. Ushiku, T. Harada, DualNet: domain-invariant network for visual question answering, in: IEEE International Conference on Multimedia and Expo, 2017, pp. 829–834.
[35] S. Kim, H. Kook, J. Sun, M. Kang, S. Ko, Parallel feature pyramid network for object detection, in: European Conference on Computer Vision, 2018, pp. 234–250.
[36] E. Yang, C. Deng, C. Li, W. Liu, J. Li, D. Tao, Shared predictive cross-modal deep quantization, IEEE Trans. Neural Netw. 29 (11) (2018) 5292–5303.
[37] C. Deng, E. Yang, T. Liu, J. Li, W. Liu, D. Tao, Unsupervised semantic-preserving adversarial hashing for image search, IEEE Trans. Image Process. 28 (8) (2019) 4032–4044.
[38] C. Deng, E. Yang, T. Liu, D. Tao, Two-stream deep hashing with class-specific centers for supervised image search, IEEE Trans. Neural Netw. (2019) 1–13.
[39] R. Zhao, W. Ouyang, H. Li, X. Wang, Saliency detection by multi-context deep learning, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 1265–1274.
[40] L. Wang, H. Lu, X. Ruan, M. Yang, Deep networks for saliency detection via local estimation and global search, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 3183–3192.
[41] X. Li, L. Zhao, L. Wei, M. Yang, F. Wu, Y. Zhuang, H. Ling, J. Wang, DeepSaliency: multi-task deep neural network model for salient object detection, IEEE Trans. Image Process. 25 (8) (2016) 3919–3930.
[42] Y. Zeng, Y. Zhuge, H. Lu, L. Zhang, M. Qian, Y. Yu, Multi-source weak supervision for saliency detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6074–6083.
[43] Z. Wu, L. Su, Q. Huang, Cascaded partial decoder for fast and accurate salient object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3907–3916.
[44] S. Zagoruyko, N. Komodakis, Wide residual networks, (2016) arXiv:1605.07146
[45] Q. Cai, H. Liu, Y. Qian, S. Zhou, X. Duan, Y.-H. Yang, Saliency-guided level set model for automatic object segmentation, Pattern Recognit. 93 (2019) 147–163.
[46] W. Xie, Y. Shi, Y. Li, X. Jia, J. Lei, High-quality spectral-spatial reconstruction using saliency detection and deep feature enhancement, Pattern Recognit. 88 (2019) 139–152.
[47] M. Cheng, N.J. Mitra, X. Huang, P.H.S. Torr, S. Hu, Global contrast based salient region detection, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 569–582.
[48] Z. Wu, L. Su, Q. Huang, Stacked cross refinement network for edge-aware salient object detection, in: IEEE International Conference on Computer Vision, 2019, pp. 7264–7273.
[49] S. Chen, X. Tan, B. Wang, H. Lu, X. Hu, Y. Fu, Reverse attention-based residual network for salient object detection, IEEE Trans. Image Process. 29 (2020) 3763–3776.
[50] M. Feng, H. Lu, Y. Yu, Residual learning for salient object detection, IEEE Trans. Image Process. 29 (2020) 4696–4708.
[51] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, J. Jiang, A simple pooling-based design for real-time salient object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3917–3926.
[52] M. Feng, H. Lu, E. Ding, Attentive feedback network for boundary-aware salient object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1623–1632.
[53] X. Zhang, T. Wang, J. Qi, H. Lu, G. Wang, Progressive attention guided recurrent network for salient object detection, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2018, pp. 714–722.
[54] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, A. Borji, Detect globally, refine locally: a novel approach to saliency detection, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2018, pp. 3127–3135.
[55] S. Chen, X. Tan, B. Wang, X. Hu, Reverse attention for salient object detection, in: European Conference on Computer Vision, 2018, pp. 236–252.
[56] P. Zhang, D. Wang, H. Lu, H. Wang, B. Yin, Learning uncertain convolutional features for accurate saliency detection, in: IEEE International Conference on Computer Vision, 2017, pp. 212–221.
[57] C. Yang, L. Zhang, H. Lu, X. Ruan, M.-H. Yang, Saliency detection via graph-based manifold ranking, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2013, pp. 3166–3173.
[58] Q. Yan, L. Xu, J. Shi, J. Jia, Hierarchical saliency detection, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2013, pp. 1155–1162.
[59] Y. Li, X. Hou, C. Koch, J.M. Rehg, A.L. Yuille, The secrets of salient object segmentation, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2014, pp. 280–287.
[60] D.R. Martin, C.C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, IEEE Trans. Pattern Anal. Mach. Intell. 26 (5) (2004) 530–549.
[61] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[62] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, A. Borji, A new way to evaluate foreground maps, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 24548–24557.
[63] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, M. Jagersand, BASNet: boundary-aware salient object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7479–7489.
[64] Y. Zeng, P. Zhang, Z. Lin, J. Zhang, H. Lu, Towards high-resolution salient object detection, in: IEEE International Conference on Computer Vision, 2019, pp. 7234–7243.
[65] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

Zhengyu Wu received the B.S. degree in computer science from Beijing Jiaotong University 2017. He is currently pursuing the Ph.D. degree in Technology of Computer Application from Beihang University, Beijing, China. His research interests include pattern recognition, computer vision, and machine learning.

Shuai Li received the Ph.D. degree in computer science from Beihang University. He is currently a professor at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His research interests include computer graphics, pattern recognition, computer vision, and medical image processing.

Chenglizhao Chen received the Ph.D. degree from Beihang University, Beijing, China. He is currently an associate professor at Qingdao University. His research interests include pattern recognition, computer vision, and machine learning.

Aimin Hao is a professor in Computer Science School and the Associate Director of State Key Laboratory of Virtual Reality Technology and Systems at Beihang University. He received his B.S., M.S., and Ph.D. in Computer Science at Beihang University. His research interests are on virtual reality, computer simulation, computer graphics, geometric modeling, image processing, and computer vision.

Hong Qin received the B.S. and M.S. degrees in computer science from Peking University. He received the Ph.D. degree in computer science from the University of Toronto. He is a professor of computer science in the Department of Computer Science, Stony Brook University. His research interests include geometric and solid modeling, graphics, physics-based modeling and simulation, computer-aided geometric design, visualization, and scientific computing.