

# Lightweight Salient Object Detection in Optical Remote-Sensing Images via Semantic Matching and Edge Alignment

Gongyang Li<sup>1</sup>, Zhi Liu<sup>1</sup>, *Senior Member, IEEE*, Xinpeng Zhang<sup>2</sup>, *Member, IEEE*,  
and Weisi Lin<sup>3</sup>, *Fellow, IEEE*

**Abstract**—Recently, relying on convolutional neural networks (CNNs), many methods for salient object detection in optical remote-sensing images (ORSI-SOD) are proposed. However, most methods ignore the number of parameters and computational cost brought by CNNs, and only a few pay attention to portability and mobility. To facilitate practical applications, in this article, we propose a novel lightweight network for ORSI-SOD based on semantic matching and edge alignment, termed *SeaNet*. Specifically, *SeaNet* includes a lightweight MobileNet-V2 for feature extraction, a dynamic semantic matching module (DSMM) for high-level features, an edge self-alignment module (ESAM) for low-level features, and a portable decoder for inference. First, the high-level features are compressed into semantic kernels. Then, semantic kernels are used to activate salient object locations in two groups of high-level features through dynamic convolution operations in DSMM. Meanwhile, in ESAM, cross-scale edge information extracted from two groups of low-level features is self-aligned through  $L_2$  loss and used for detail enhancement. Finally, starting from the highest level features, the decoder infers salient objects based on the accurate locations and fine details contained in the outputs of the two modules. Extensive experiments on two public datasets demonstrate that our lightweight *SeaNet* not only outperforms most state-of-the-art lightweight methods, but also yields comparable accuracy with state-of-the-art conventional methods, while having only 2.76 M parameters and running with 1.7 G floating point operations (FLOPs) for 288 × 288 inputs. Our code and results are available at <https://github.com/MathLee/SeaNet>.

**Index Terms**—Edge alignment, lightweight salient object detection (SOD), optical remote-sensing image (ORSI), semantic matching.

Manuscript received 7 August 2022; revised 24 October 2022; accepted 6 January 2023. Date of publication 11 January 2023; date of current version 20 January 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62171269 and Grant U1936214, in part by the China Postdoctoral Science Foundation under Grant 2022M722037, in part by the Science and Technology Commission of Shanghai Municipality under Grant 21010500200, and in part by the Singapore Ministry of Education Tier-2 Fund under Grant MOE2016-T2-2-057(S). (*Corresponding authors: Zhi Liu; Xinpeng Zhang.*)

Gongyang Li, Zhi Liu, and Xinpeng Zhang are with the Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Joint International Research Laboratory of Specialty Fiber Optics and Advanced Communication, Shanghai Institute for Advanced Communication and Data Science, and the School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China (e-mail: ligongyang@shu.edu.cn; liuzhisjtu@163.com; xzhang@shu.edu.cn).

Weisi Lin is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: wslin@ntu.edu.sg).

Digital Object Identifier 10.1109/TGRS.2023.3235717

1558-0644 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See <https://www.ieee.org/publications/rights/index.html> for more information.

## I. INTRODUCTION

**S**ALIENT object detection (SOD) aims at imitating the human vision system to quickly locate the objects/areas that attract the most attention [1]. As an important preprocessing step, the success of SOD has promoted the development of many fields, such as image quality assessment [2], [3], object segmentation [4], [5], and object tracking [6]. Different from most SOD methods proposed for single images [7], RGB and depth/thermal images (RGB-D/T) [8], [9], and video [10] photographed in natural scenes, in this article, we focus on SOD in optical remote-sensing images, or ORSI-SOD for short. Following the technical trend of SOD in natural scene images (NSI-SOD) [7], we are committed to addressing ORSI-SOD based on convolutional neural networks (CNNs) [11].

In the era of deep learning, numerous CNN-based NSI-SOD methods have been proposed, and the detection accuracy has been significantly improved. Among these methods, the classic encoder–decoder structure [16] is the most general and effective structure and is often accompanied by ingenious strategies such as deep supervision [17], gate mechanism [18], edge assistance [19], [20], progressive architecture [12], and so on. Although NSI-SOD methods cannot directly overcome the issue of complex scenes of optical remote-sensing images (ORSIs) (as PA-KRN [12] shown in the third column of Fig. 1), the strategies contained therein lay the foundation for CNN-based ORSI-SOD methods. The specialized methods for ORSI-SOD take into account the properties of salient objects and scenes in ORSIs. For example, LVNet [21] and EMFINet [22] take ORSIs with multiple resolutions as inputs to overcome the problem of variable sizes of salient objects. MCCNet [14] comprehensively integrates foreground, background, edge, and global information to deal with the complex background of ORSIs, producing good saliency maps as shown in the fifth column of Fig. 1.

However, the above methods may fall into the dilemma of the huge amount of parameters and computational cost, such as the parameters and floating point operations (FLOPs) of PA-KRN and MCCNet listed in Fig. 1. To address this issue, lightweight SOD methods are gradually emerging. Compared with PA-KRN, the pioneer of the lightweight NSI-SOD method HVPNet [13] reduces the number of parameters and computational cost by hundreds of times. But HVPNet is also stuck by ORSIs, as shown in the fourth column of Fig. 1. For

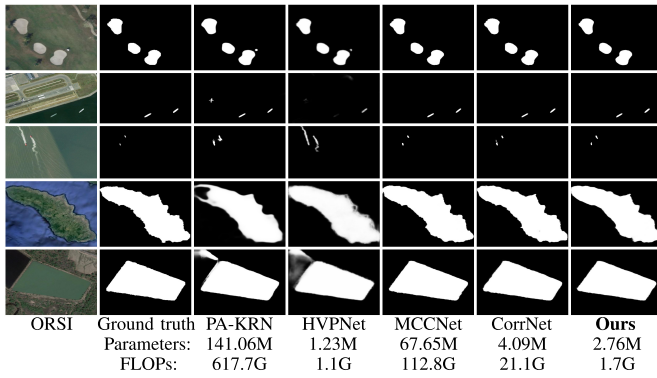


Fig. 1. Saliency maps produced by four types of methods and our method on ORSIs. PA-KRN [12] is an NSI-SOD method, HVPNet [13] is a lightweight NSI-SOD method, MCCNet [14] is an ORSI-SOD method, and CorrNet [15] is a lightweight ORSI-SOD method. Please zoom-in for details.

the lightweight ORSI-SOD method, as shown in the penultimate column of Fig. 1, CorrNet [15] significantly reduces the number of parameters and achieves good performance, but still consumes a lot of computational costs.

Driven by the aforementioned observation, in this article, we propose a novel semantic matching and edge alignment-based ORSI-SOD method, termed *SeaNet*, which aims to be more lightweight than CorrNet, while generating competitive performance. As we all know, the features extracted by the feature extraction network can be divided into low- and high-level features, where the former contains detail and texture information and the latter contains semantic and location information. Accordingly, the main idea of *SeaNet* is to explore high- and low-level features with different strategies in the encoder-decoder structure.

Specifically, we propose a dynamic semantic matching module (DSMM) to implement semantic matching in high-level features, that is, first compress semantic information and then match them with the high-level features to perceive the location of salient objects. We also propose an edge self-alignment module (ESAM) for edge alignment in low-level features, that is, align cross-scale edge information extracted from low-level features to correct edge errors and use them to enhance features. For efficiency, we adopt MobileNet-V2 [23] as the backbone and the depthwise separable convolution (DSconv) [23], [24] as the basic convolution component to control the number of parameters and computational cost. In this way, our *SeaNet* has only 2.76 M parameters, runs with 1.7 G FLOPs, and can generate accurate saliency maps, as shown in the rightmost column of Fig. 1. Concretely, compared with state-of-the-art lightweight methods, our *SeaNet* is competitive in detection accuracy, and when compared with state-of-the-art conventional methods, our *SeaNet* is competitive in computational complexity.

Our main contributions are threefold as follows.

- 1) We explore high- and low-level features of MobileNet-V2 with different strategies and propose a novel lightweight network for ORSI-SOD based on *SeaNet*, which has only 2.76 M parameters and runs with 1.7 G FLOPs for a  $288 \times 288$  image.

- 2) We propose a DSMM for high-level semantic features. DSMM perceives the location of salient objects through dynamic convolutions with semantic kernels, which not only improves the flexibility of feature interaction, but also effectively reduces the number of parameters. Moreover, it performs channel-wise correlation to activate the channel-wise interaction.
- 3) We propose an ESAM for low-level detail features. ESAM focuses on extracting edge information for detail enhancement and aligns cross-scale edge information through  $L_2$  loss to correct edge errors. Like DSMM, it also introduces channel-wise correlation.

We arrange the remainder of this article as follows. In Section II, we review the CNN-based SOD methods for natural scene images (NSIs) and ORSIs. In Section III, we elaborate our *SeaNet*. In Section IV, we present comprehensive experimental results. In Section V, we give the conclusion.

## II. RELATED WORK

### A. CNN-Based SOD in NSIs

NSI-SOD [7] has achieved remarkable success, especially CNN-based methods. Hou et al. [17] creatively introduced the pioneer deep supervision into NSI-SOD, which significantly enhances the representation of multiscale features for salient objects. This method effectively improves the detection accuracy and has a profound impact on subsequent CNN-based methods. Hu et al. [25] and Deng et al. [26] focused on the recurrent mechanism. The former first concatenates multilevel features and then combines them with features from different levels. The latter alternatively uses low- features and high-level features. Besides, some researchers were interested in edge/boundary information. Feng et al. [27] proposed a boundary-enhanced loss to improve the completeness of object boundaries and combined it with deep supervision. Qin et al. [28] proposed a hybrid loss, which integrates binary cross-entropy (BCE), structure similarity index measure (SSIM), and intersection-over-union (IoU) losses, to segment salient objects with fine structures and clear boundaries. Zhao et al. [19] modeled the explicit edge-through-edge supervision to preserve the salient object boundaries. Zhou et al. [20] proposed the saliency to contour and contour to saliency strategy for fast saliency detection. Lee et al. [29] proposed an attention-guided tracing module to highlight salient objects with explicit edges.

In addition to the above classic strategies, Pang et al. [30] enhanced the interaction of multiscale features for NSI-SOD. Chen et al. [31] considered the low-level, high-level, and global information to improve the completeness of the saliency map. Xu et al. [12] utilized the global localization and local segmentation policy in the knowledge review network to avoid salient information dilution. Li et al. [32] captured both the multireceptive field information of features and the complementary information of cross-level features. Ke and Tsubono [33] extended the atrous spatial pyramid pooling and embedded the channel and spatial attention into it to explore information dependencies in space and channel.

Although the above CNN-based methods achieve excellent performance in NSI-SOD, they cannot effectively handle the unique properties of ORSIs and often generate unsatisfactory saliency maps. Furthermore, they usually focus on accuracy and ignore computational complexity. Nonetheless, their strategies inspire our approach, such as the classic deep supervision, edge assistance, and differential feature processing.

### B. CNN-Based SOD in ORSIs

ORSI-SOD is a rising star in the SOD community, and recently numerous CNN-based methods are proposed. As a pioneer, Li et al. [21] proposed the first CNN-based ORSI-SOD method, named LVNet, in which a two-stream pyramid module cooperates with an encoder–decoder module with nested connections to perceive salient objects of different sizes. Moreover, Li et al. [34] explored the interactions of cross-level features for ORSI-SOD. Huang et al. [35] first roughly located salient objects in the semantic guided decoder and then refined the coarse saliency map in a top-down manner. Cong et al. [36] designed the relational reasoning encoder for high-level features and inferred salient objects in a multiscale attention decoder. Li et al. [37] adopted two groups of adjacent features to assist the current features, capturing contextual information to overcome challenging ORSI scenarios.

Similar to NSI-SOD, some researchers introduced edge/boundary information into ORSI-SOD. Zhang et al. [38] constructed a multitask structure to predict the edge map and saliency map simultaneously. Zhou et al. [22] and Tu et al. [39], following [19], extracted boundary information based on low- and high-level features to preserve boundaries of salient objects in two decoders. Li et al. [14] integrated edge with foreground, background, and global information, and took full account of the complementarity between this information to adapt to ORSIs.

The above-specialized methods for ORSI-SOD achieve satisfactory performance. However, they usually come with a large number of parameters and heavy computational costs, which are unfriendly to aerospace equipment and prevent practical applications. To this end, we propose SeaNet based on MobileNet-V2 [23] and two lightweight but effective modules, which are friendly to mobile devices while achieving competitive performance.

### C. Lightweight SOD

Lightweight SOD is a newly emerging task and is first explored in NSI-SOD. Gao et al. [40] proposed a flexible self-adaptive convolutional layer with strong multiscale representation abilities and constructed an extremely lightweight network (i.e., CSNet) for NSI-SOD. Liu et al. [13] proposed a hierarchical visual perception (HVP) module based on dense connections and built a lightweight HVPNet on HVP modules and residual attention to effectively learn multiscale contexts. Meanwhile, Liu et al. [41] proposed a stereoscopically attentive multiscale (SAM) module and built a lightweight SAMNet for multilevel and multiscale learning. To sum up, the above three works focus on learning effective multilevel and

multiscale information for SOD and building strong feature extraction backbones to achieve lightweight NSI-SOD. However, they do not further process and enhance the extracted features at different levels and directly infer salient objects based on these features. Moreover, they deal with NSIs rather than ORSIs, lacking pertinence. Therefore, in this article, we focus on making good use of features at different levels of existing feature extraction backbones and developing specific lightweight and effective modules for ORSI features, enabling a more effective dedicated ORSI-SOD solution.

For ORSI-SOD, Li et al. [15] proposed the first lightweight method, that is, CorrNet. They lightened the vanilla VGG-16 [42] for efficient feature extraction and adopted the coarse-to-fine strategy to detect salient objects in ORSIs with dense lightweight refinement blocks. The parameters of CorrNet were greatly reduced to only 4.09 M, but its computational cost was still very large, with 21.1 G FLOPs. For RGB-D SOD, Wu et al. [43] proposed the first lightweight method, that is, MobileSal, which is based on MobileNet-V2 [23].

Inspired by Wu et al. [43], in SeaNet, we adopt MobileNet-V2 as the backbone to overcome the issue of the large computational cost of the existing lightweight ORSI-SOD method, that is, CorrNet. Moreover, to reduce the number of parameters, we propose two lightweight and effective modules, that is, DSMM and ESAM. DSMM extends dynamic convolution [44] to dynamic depthwise convolution (DDconv), and ESAM corrects cross-scale edge features in a self-alignment way.

## III. PROPOSED METHOD

In this section, we elaborate on the proposed lightweight SeaNet. In Section III-A, we introduce the network overview of the proposed SeaNet. In Sections III-B and III-C, we elaborate on two lightweight modules, that is, DSMM and ESAM, respectively. In Section III-D, we depict the decoder and loss function.

### A. Network Overview

As shown in Fig. 2, the proposed lightweight SeaNet is based on the encoder–decoder structure commonly used in SOD [8], [9], [45], [46]. SeaNet includes an encoder, a semantic knowledge compression (SKC) unit, a DSMM, an ESAM, and a lightweight decoder. It first performs semantic matching for location activation of salient objects and then performs edge alignment for detail enhancement.

The input size of our SeaNet is  $3 \times 288 \times 288$ . For the encoder of our SeaNet, we adapt the lightweight MobileNet-V2 [23], that is, we keep the first 17 inverted residual bottlenecks and truncate the last three layers, that is, two convolution layers and one average pooling layer. We divide MobileNet-V2 into five blocks based on the first, third, sixth, 13th, and last bottlenecks, denoted as  $E^t$  ( $t = 1, 2, 3, 4, 5$ ). The output five-level features are denoted as  $\mathbf{f}_e^t \in \mathbb{R}^{c_t \times h_t \times w_t}$ , where  $h_t$  and  $w_t$  are  $(288/2^t)$ , and  $c_t \in \{16, 24, 32, 96, 320\}$ . We explore the generated high- and low-level features with different strategies for ORSI-SOD. For the high-level features, we first compress the highest-level features  $\mathbf{f}_e^5$  into two

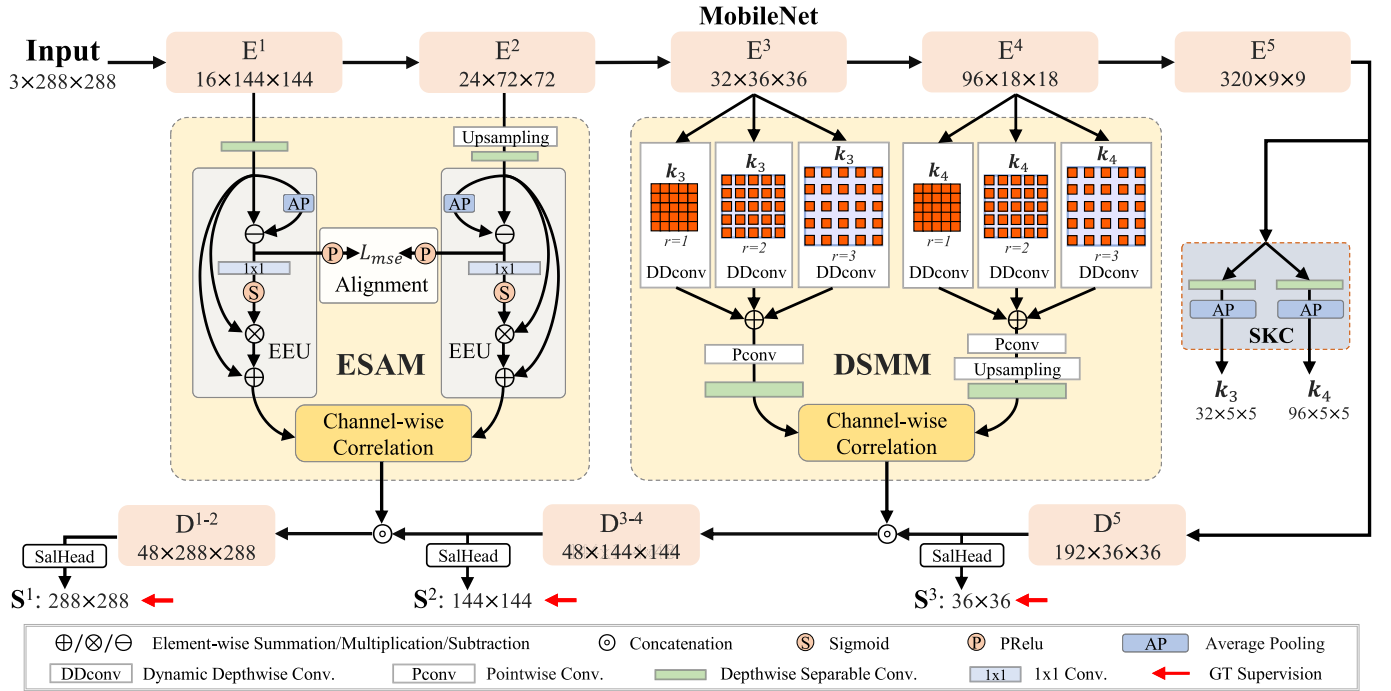


Fig. 2. Pipeline of the proposed lightweight SeaNet, which follows the encoder–decoder architecture. First, MobileNet-V2 [23] extracts basic feature embeddings from the input, resulting in five-level features. Next, we get two semantic kernels in the SKC unit. Semantic kernels are used to convolve with high-level features in the DSMM for location activation of salient objects. Then, we perform the channel-wise correlation [47], [48] in DSMM. Meanwhile, in the ESAM, we adopt EEUs to extract edge features from low-level features for detail enhancement and align edge features via  $L_{mse}$  loss (i.e.,  $L_2$  loss). Channel-wise correlation is also added to ESAM. Finally, we infer salient objects in the decoder based on the highest level features and the outputs of DSMM and ESAM and obtain the output saliency map  $S^1$ .

semantic kernels, that is,  $k_3$  and  $k_4$ , in the SKC unit, and then use them as kernels for DDconvs [24], [44] to, respectively, convolve with two groups of high-level features, that is,  $f_e^3$  and  $f_e^4$ , in DSMM. In addition to the above spatial semantic matching, we enhance the channel interaction via channel-wise correlation [47], [48], generating  $f_{dsmm}$ . Meanwhile, for the low-level features, we obtain edge information through the pooling-subtraction operation [27] and correct edge errors in a self-alignment way. Then, we adopt the corrected edge features to perform feature enhancement in edge-based enhancement units (EEUs) of ESAM and also perform channel-wise correlation, generating  $f_{esam}$ . The decoder of our SeaNet is comprised of three lightweight blocks denoted as  $D^{1-2}$ ,  $D^{3-4}$ , and  $D^5$ . Using  $f_e^5$ ,  $f_{dsmm}$ , and  $f_{esam}$ , we highlight salient objects in a progressive manner for better resolution recovery. We also introduce the saliency inference head (SalHead) after each decoder block for deep supervision [17] and final saliency map generation.

### B. Dynamic Semantic Matching Module

High-level features contain affluent semantic information, which is beneficial for salient object localization. Here, we propose DSMM to effectively activate salient regions using high-level features with limited parameters and computational cost. Inspired by the dynamic convolution [44], we generate convolution kernels with existing features rather than parameter initialization to reduce the number of parameters. Furthermore, we extend the dynamic convolution with the DSconv [24] and propose the DDconv to simultaneously

reduce the computational cost. DDconv plays an important role in the spatial semantic matching of our DSMM. However, considering only spatial interactions is not sufficient, and we therefore further introduce channel interactions into DSMM to enhance channel dependencies through the channel-wise correlation [47], [48].

We show the detailed structure of DSMM in the middle part of Fig. 2. Our DSMM can be divided into two parts, that is, the spatial semantic matching and the channel-wise correlation. In the following, we present DSMM based on these two parts. Since the SKC unit generates semantic kernels for DSMM, we first describe this unit in detail.

1) *SKC Unit*: As shown in the right part of Fig. 2, the SKC unit compresses the semantic information of  $f_e^5$  directly but effectively. Since the SKC unit generates semantic kernels for  $f_e^3$  and  $f_e^4$  that are the inputs of DSMM, we first compress the channel number of  $f_e^5$  by two parallel DSconv layers to fit that of  $f_e^3$  and  $f_e^4$  and then compress the resolution to a suitable size through two parallel adaptive average pooling layers, generating two semantic kernels  $k_3 \in \mathbb{R}^{32 \times 5 \times 5}$  and  $k_4 \in \mathbb{R}^{96 \times 5 \times 5}$ . We formulate the SKC unit as follows:

$$k_t = \text{AP}(\text{DSconv}(f_e^5)), \quad t = 3, 4 \quad (1)$$

where  $\text{DSconv}(\cdot)$  is the  $3 \times 3$  DSconv layer, and  $\text{AP}(\cdot)$  is the adaptive average pooling layer.

2) *Spatial Semantic Matching*: We adopt  $k_3$  and  $k_4$  containing global information as the kernel of dynamic convolution layers [44] to reduce the number of parameters. However, the computational cost of the traditional dynamic convolution layer is the same as that of the regular convolution layer.

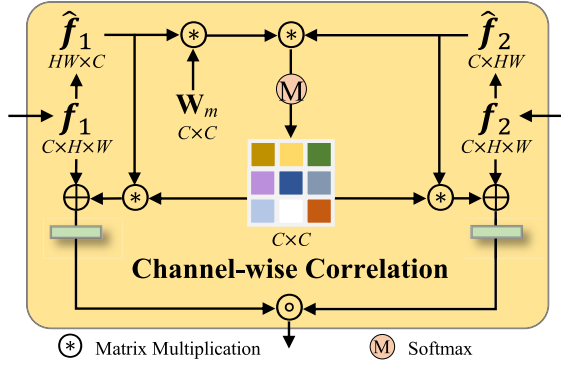


Fig. 3. Illustration of the channel-wise correlation.

Inspired by DSconv [24], we update dynamic convolution to DDconv, which performs dynamic convolution in a depthwise manner, that is, the group of dynamic convolution is set to the channel number of input features. Moreover, we introduce the dilation mechanism [49] into DDconv and use multiple dilated receptive fields to sufficiently perceive salient objects for accurate localization, which is effective for overcoming the scenes of multiple objects and objects with variable sizes in ORSIs.

As shown in DSMM of Fig. 2, we, respectively, employ three dilated DDconv layers with dilation rates  $\{1, 2, 3\}$  on  $f_e^3$  and  $f_e^4$ . In fact, DDconv is a parameter-free semantic matching process, which breaks the constraints of the trained parameters and improves the flexibility of the module and even the network. Different from DSconv that executes depthwise convolution and pointwise convolution sequentially, we integrate the output features of three dilated DDconvs through element-wise summation and then perform pointwise convolution to fuse the multiperception features. This not only further reduces the number of parameters, but also facilitates the interaction of features generated with different receptive fields. We formulate the spatial semantic matching as follows:

$$f_{sm}^t = \text{Pconv} \left( \sum_{i=1}^3 \text{DDconv}(f_e^t; k_t, r_i) \right), \quad t = 3, 4 \quad (2)$$

where  $f_{sm}^t \in \mathbb{R}^{c_t \times h_t \times w_t}$  denotes the output features of semantic matching,  $\text{DDconv}(\cdot; k_t, r_i)$  is the DDconv layer with dynamic kernel  $k_t$  and dilation rate  $r_i \in \{1, 2, 3\}$ ,  $\sum$  is the element-wise summation of multiple features, and  $\text{Pconv}(\cdot)$  is the pointwise convolution layer.

3) *Channel-Wise Correlation*: In addition to the spatial interactions, we perform the channel-wise correlation extended from spatial co-attention [47], [48] in DSMM. We first align  $f_{sm}^3$  and  $f_{sm}^4$  in channel and resolution via a DSconv layer and an upsampling operation to obtain the input features of channel-wise correlation, that is,  $\{\hat{f}_{sm}^3, \hat{f}_{sm}^4\} \in \mathbb{R}^{c_4 \times h_3 \times w_3}$ . Then, we define the channel-wise correlation, denoted by  $\text{CCorr}(\cdot)$ , as follows:

$$f_{dsmm} = \text{CCorr}(\hat{f}_{sm}^3, \hat{f}_{sm}^4) \quad (3)$$

where  $f_{dsmm} \in \mathbb{R}^{(2 \times c_4) \times h_3 \times w_3}$  denotes the output features of DSMM.

We depict the structure of channel-wise correlation in Fig. 3, where we simplify the input features to  $\{f_1, f_2\} \in \mathbb{R}^{C \times H \times W}$

for brevity. First, we reshape the size of  $f_1$  and  $f_2$  and obtain the flattened  $\hat{f}_1 \in \mathbb{R}^{HW \times C}$  and  $\hat{f}_2 \in \mathbb{R}^{C \times HW}$ . Then, we multiply  $\hat{f}_1$  by a trainable matrix  $W_m \in \mathbb{R}^{C \times C}$  using matrix multiplication to adaptively learn feature transformations. The channel-wise affinity matrix  $A \in \mathbb{R}^{C \times C}$  of  $\hat{f}_1$  and  $\hat{f}_2$  can be calculated through matrix multiplication as follows:

$$A = \hat{f}_2 \otimes (\hat{f}_1 \otimes W_m) \quad (4)$$

where  $\otimes$  is the matrix multiplication. In this way, we model feature dependencies along the channel.

Then, we adopt the row-wise and column-wise softmax functions to normalize the affinity matrix, respectively, and transfer the established channel dependencies to  $\hat{f}_1$  and  $\hat{f}_2$ . Besides, we introduce the short connection and a  $3 \times 3$  DSconv layer to integrate the original spatially enhanced features (i.e.,  $f_1$  and  $f_2$ ) and the above channel-enhanced features. We formulate the above process as follows:

$$f_{sc}^1 = \text{DSconv} \left( (M_r(A) \otimes \hat{f}_1^\top) \oplus f_1 \right) \quad (5)$$

$$f_{sc}^2 = \text{DSconv} \left( ((M_c(A))^\top \otimes \hat{f}_2) \oplus f_2 \right) \quad (6)$$

where  $\{f_{sc}^1, f_{sc}^2\} \in \mathbb{R}^{C \times H \times W}$  are features of spatial and channel enhancement,  $M_r(\cdot)$  and  $M_c(\cdot)$  are the row-wise and column-wise softmax functions, respectively,  $\top$  is the matrix transpose operation, and  $\oplus$  is the element-wise summation. Notably, we omit feature size transformation for brevity. Finally, we concatenate  $f_{sc}^1$  and  $f_{sc}^2$  to produce the output features of channel-wise correlation  $f_{ccorr} \in \mathbb{R}^{2C \times H \times W}$ , that is, the output features of DSMM  $f_{dsmm}$ .

In summary, our DSMM is implemented in a lightweight manner with limited parameters and computational cost. And we fully consider spatial interactions and channel interactions in DSMM, providing comprehensive and accurate localization of salient objects, which is conducive to conquering the challenging scenes of ORSIs.

### C. Edge Self-Alignment Module

Low-level features contain rich texture and object detail information, which is conducive to delineating the fine structure of salient objects. We propose ESAM to effectively and efficiently explore edge information for detail enhancement to preserve the complex shapes of salient objects in ORSIs. Different from some edge-based ORSI-SOD methods [14], [22], [38], [39], our ESAM is lightweight and extracts edge information without using edge supervision, which is more convenient. Like DSMM, ESAM also fully considers the spatial interaction and channel interaction of features. As illustrated in the left part of Fig. 2, ESAM consists of two EEU and one channel-wise correlation. We describe them in turn.

1) *EEU With Self-Alignment*: The input features of ESAM are  $f_e^1$  and  $f_e^2$ . We align them via DSconv layer and upsampling operation and obtain  $\{\hat{f}_e^1, \hat{f}_e^2\} \in \mathbb{R}^{c_2 \times h_1 \times w_1}$ , which are the input features of EEUs. We adopt the pooling-subtraction operation [27] to extract two groups of edge features  $\{\hat{f}_{edge}^1, \hat{f}_{edge}^2\} \in \mathbb{R}^{c_2 \times h_1 \times w_1}$  from  $\hat{f}_e^1$  and  $\hat{f}_e^2$ , respectively, as follows:

$$f_{edge}^t = \hat{f}_e^t \ominus \text{AP}(\hat{f}_e^t), \quad t = 1, 2 \quad (7)$$

TABLE I

DETAILED STRUCTURE AND PARAMETERS OF THREE DECODER BLOCKS. ( $3 \times 3$ , 320, 192) DENOTES THAT THE KERNEL SIZE OF DSConv IS  $3 \times 3$ , THE INPUT CHANNEL NUMBER IS 320, AND THE OUTPUT CHANNEL NUMBER IS 192

Aspects	D <sup>5</sup>	D <sup>3-4</sup>	D <sup>1-2</sup>
Input size	[320 × 9 × 9]	[384 × 36 × 36]	[96 × 144 × 144]
DSconv	(3 × 3, 320, 320)	(3 × 3, 384, 192)	(3 × 3, 96, 48)
DSconv	(3 × 3, 320, 320)	(3 × 3, 192, 192)	(3 × 3, 48, 48)
Upsampling	4×	4×	2×
DSconv	(3 × 3, 320, 192)	(3 × 3, 192, 48)	(3 × 3, 48, 48)
Output size	[192 × 36 × 36]	[48 × 144 × 144]	[48 × 288 × 288]

where  $\ominus$  is the element-wise subtraction.  $f_{\text{edge}}^t$  is then used to delineate edge regions in  $\hat{f}_e^t$ . However, since there is no edge supervision, the obtained edge information is inevitably vulnerable to errors. Attacking this problem, we propose a novel self-alignment mechanism based on the mean squared error loss (i.e.,  $L_2$  or  $L_{\text{mse}}$  loss) and apply it to edge features, as shown in ESAM of Fig. 2. In this way, we can adaptively correct edge errors between  $\hat{f}_{\text{edge}}^1$  and  $\hat{f}_{\text{edge}}^2$  during the training phase and obtain accurate and consistent edge information.

Based on the corrected edge information, we perform the edge enhancement on  $\hat{f}_e^t$  and obtain the output features of EEU, denoted as  $f_{\text{eeu}}^t \in \mathbb{R}^{c_2 \times h_1 \times w_1}$ , as follows:

$$f_{\text{eeu}}^t = \text{conv}_{1 \times 1}^s \left( f_{\text{edge}}^t \right) \otimes \hat{f}_e^t \oplus \hat{f}_e^t, \quad t = 1, 2 \quad (8)$$

where  $\text{conv}_{1 \times 1}^s(\cdot)$  is the  $1 \times 1$  convolution layer with sigmoid activation function, and  $\otimes$  is the element-wise multiplication.

2) *Channel-Wise Correlation*: EEUs focus on feature enhancement at the spatial level, and we enhance the channel interaction of their outputs  $f_{\text{eeu}}^1$  and  $f_{\text{eeu}}^2$  using channel-wise correlation as follows:

$$f_{\text{esam}} = \text{CCorr}(f_{\text{eeu}}^1, f_{\text{eeu}}^2) \quad (9)$$

where  $f_{\text{esam}} \in \mathbb{R}^{(2 \times c_2) \times h_1 \times w_1}$  is the output features of ESAM.

In this way, ESAM saves a large number of parameters and computational costs, while providing powerful support for accurately highlighting the complex geometry and topology of salient objects in ORSIs.

#### D. Decoder and Loss Function

1) *Decoder*: Based on the locations and details of salient objects provided by the above two modules, we design a lightweight decoder to produce saliency maps. As shown at the bottom of Fig. 2, our lightweight decoder consists of three blocks, that is, D<sup>1-2</sup>, D<sup>3-4</sup>, and D<sup>5</sup>. Each decoder block in turn contains two DSconv layers, an upsampling operation, and another DSconv layer. Their detailed parameters are reported in Table I. In particular, we arrange SalHeads after these three decoder blocks to generate three saliency maps of different resolutions, that is,  $\mathbf{S}^3 \in [0, 1]^{1 \times 36 \times 36}$ ,  $\mathbf{S}^2 \in [0, 1]^{1 \times 144 \times 144}$ , and  $\mathbf{S}^1 \in [0, 1]^{1 \times 288 \times 288}$ , where the first two are used for deep supervision and the last one is the final output of our

SeaNet. SalHead is comprised of a dropout layer [51] and a  $1 \times 1$  convolution layer.

2) *Loss Function*: We impose the BCE loss and IoU loss to jointly train our SeaNet. Therefore, our total loss consists of two parts, that is, the saliency loss and the edge alignment loss. Moreover, we introduce a loss weight to treat these two losses differently for better training. The total loss function  $L_{\text{total}}$  can be formulated as follows:

$$L_{\text{total}} = \sum_{i=1}^3 (L_{\text{bce}}^i + L_{\text{iou}}^i) + \lambda \cdot L_{\text{mse}}(P(f_{\text{edge}}^1), P(f_{\text{edge}}^2)) \quad (10)$$

where  $L_{\text{bce}}^i$  and  $L_{\text{iou}}^i$  are BCE loss and IoU loss, respectively, which supervise  $\mathbf{S}^i$  by the ground truth (GT);  $\lambda$  is the loss weight and set to 0.5;  $L_{\text{mse}}(\cdot)$  is the mean squared error loss; and  $P(\cdot)$  is the parametric rectified linear unit [52].

## IV. EXPERIMENTS

### A. Experimental Setup

1) *Datasets*: We conduct experiments on the ORSSD [21] and EORSSD [38] datasets. The ORSSD dataset has 800 ORSIs and corresponding pixel-level annotations. It is divided into two parts, that is, the training set (600 images) and the test set (200 images). The EORSSD dataset adds 1200 ORSIs to the ORSSD dataset, resulting in 2000 ORSIs. This dataset is also divided into two parts, that is, 1400 images in the training set and 600 images in the test set.

2) *Evaluation Metrics*: We adopt quantitative evaluation metrics and computational complexity metrics to evaluate our method and all compared methods from two aspects.

Quantitative evaluation metrics include  $S$ -measure ( $S_\alpha$ ,  $\alpha = 0.5$ ) [53],  $F$ -measure ( $F_\beta$ ,  $\beta^2 = 0.3$ ) [54],  $E$ -measure ( $E_c$ ) [55], and mean absolute error ( $\mathcal{M}$ ). The first three are the higher the better, and the last one is the opposite. We report the maximum, mean, and adaptive  $F$ -measure and  $E$ -measure.

Computational complexity metrics include the inference speed measured in frames/s (fps), the parameter amount (#Param) measured in million (M), and the number of FLOPs measured in giga (G). The first one is the higher the better, and the last two are the opposite. The inference speed is reported with a batch size of 1 and no I/O time.

3) *Training Protocol*: We conduct experiments based on the PyTorch [56] on a computer with an NVIDIA Titan X GPU (12 GB memory). We list the training details and parameters as follows: the input size is  $288 \times 288$ , the data augmentation strategy includes flipping and rotation, the network optimizer is Adam [57], the batch size is 8, the base learning rate is  $1e^{-4}$ , the learning rate decays to 1/10 every 30 epochs, and the training epoch is 50. Besides, we initialize MobileNet-V2 with the pretrained parameters and initialize the newly added layers of two modules and decoder with the ‘‘Kaiming’’ method [52]. Notably, for each dataset, we train on its own training set and test on its own test set, as in [15], [22], and [38].

### B. Performance Analysis

We compare our SeaNet with 17 state-of-the-art conventional SOD methods, including 11 conventional

TABLE II

QUANTITATIVE AND COMPUTATIONAL COMPLEXITY COMPARISONS WITH STATE-OF-THE-ART NSI-SOD METHODS, ORSI-SOD METHODS, AND LIGHTWEIGHT METHODS ON EORSSD AND ORSSD DATASETS.  $\uparrow$  INDICATES THAT THE HIGHER THE BETTER, WHILE  $\downarrow$  IS THE OPPOSITE. THE TOP THREE RESULTS ARE MARKED IN **RED**, **BLUE**, AND **GREEN**, RESPECTIVELY

Methods	Type	Input size	Speed (fps) $\uparrow$	#Param (M) $\downarrow$	FLOPs (G) $\downarrow$	EORSSD [38]								ORSSD [21]							
						$S_\alpha \uparrow$	$F_\beta^{\max} \uparrow$	$F_\beta^{\text{mean}} \uparrow$	$F_\beta^{\text{adp}} \uparrow$	$F_\xi^{\max} \uparrow$	$E_\xi^{\text{mean}} \uparrow$	$E_\xi^{\text{adp}} \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta^{\max} \uparrow$	$F_\beta^{\text{mean}} \uparrow$	$F_\beta^{\text{adp}} \uparrow$	$E_\xi^{\max} \uparrow$	$E_\xi^{\text{mean}} \uparrow$	$E_\xi^{\text{adp}} \uparrow$	$\mathcal{M} \downarrow$
DSS <sub>17</sub> [17]	CN	400×300	8	62.23	114.6	.7868	.6849	.5801	.4597	.9186	.7631	.6933	.0186	.8262	.7467	.6962	.6206	.8860	.8362	.8085	.0363
RADF <sub>18</sub> [25]	CN	400×400	7	62.54	214.2	.8179	.7446	.6582	.4933	.9140	.8567	.7162	.0168	.8259	.7619	.6856	.5730	.9130	.8298	.7678	.0382
R3Net <sub>18</sub> [26]	CN	300×300	2	56.16	47.5	.8184	.7498	.6302	.4165	.9483	.8294	.6462	.0171	.8141	.7456	.7383	.7379	.8913	.8681	.8887	.0399
PoolNet <sub>19</sub> [50]	CN	400×300	25	53.63	123.4	.8207	.7545	.6406	.4611	.9292	.8193	.6836	.0210	.8403	.7706	.6999	.6166	.9343	.8650	.8124	.0358
EGNet <sub>19</sub> [19]	CN	~380×320	9	108.07	291.9	.8601	.7880	.6967	.5379	.9570	.8775	.7566	.0110	.8721	.8332	.7500	.6452	.9731	.9013	.8226	.0216
GCPA <sub>20</sub> [31]	CN	320×320	23	67.06	54.3	.8869	.8347	.7905	.6723	.9524	.9167	.8647	.0102	.9026	.8687	.8433	.7861	.9509	.9341	.9205	.0168
MINet <sub>20</sub> [30]	CN	320×320	12	47.56	146.3	.9040	.8344	.8174	.7705	.9442	.9346	.9243	.0093	.9040	.8761	.8574	.8251	.9545	.9454	.9423	.0144
ITSD <sub>20</sub> [20]	CN	288×288	16	17.08	54.5	.9050	.8523	.8221	.7421	.9556	.9407	.9103	.0106	.9050	.8735	.8502	.8068	.9601	.9482	.9335	.0165
GateNet <sub>20</sub> [18]	CN	384×384	25	100.02	108.3	.9114	.8566	.8228	.7109	.9610	.9385	.8909	.0095	.9186	.8871	.8679	.8229	.9664	.9538	.9428	.0137
SUCA <sub>21</sub> [32]	CN	256×256	24	117.71	56.4	.8988	.8229	.7949	.7260	.9520	.9277	.9082	.0097	.8989	.8484	.8237	.7748	.9584	.9400	.9194	.0145
PA-KRN <sub>21</sub> [12]	CN	600×600	16	141.06	617.7	.9192	.8639	.8358	.7993	.9616	.9536	.9416	.0104	.9239	.8890	.8727	.8548	.9680	.9620	.9579	.0139
LVNet <sub>19</sub> [21]	CR	128×128	1.4	-	-	.8630	.7794	.7328	.6284	.9254	.8801	.8445	.0146	.8815	.8263	.7995	.7506	.9456	.9259	.9195	.0207
DAFNet <sub>21</sub> [38]	CR	128×128	26	29.35	68.5	.9166	.8614	.7845	.6427	<b>.9861</b>	.9291	.8446	<b>.0060</b>	.9191	.8928	.8511	.7876	<b>.9771</b>	.9539	.9360	.0113
SARNet <sub>21</sub> [35]	CR	336×336	47	25.91	129.7	.9240	.8719	<b>.8541</b>	<b>.8304</b>	.9620	.9555	.9536	.0099	.9134	.8850	.8619	.8512	.9557	.9477	.9464	.0187
MJRBM <sub>22</sub> [39]	CR	352×352	32	43.54	95.7	.9197	.8656	.8239	.7066	.9646	.9350	.8897	.0099	.9204	.8842	.8566	.8022	.9623	.9415	.9328	.0163
EMFNet <sub>22</sub> [22]	CR	256×256	25	107.26	480.9	<b>.9290</b>	<b>.8720</b>	.8486	.7984	<b>.9711</b>	.9604	.9501	.0084	<b>.9366</b>	<b>.9002</b>	<b>.8856</b>	.8617	.9737	.9671	.9663	.0109
MCCNet <sub>22</sub> [14]	CR	256×256	95	67.65	112.8	<b>.9327</b>	<b>.8904</b>	<b>.8604</b>	<b>.8137</b>	<b>.9755</b>	<b>.9685</b>	<b>.9538</b>	<b>.0066</b>	<b>.9437</b>	<b>.9155</b>	<b>.9054</b>	<b>.8957</b>	<b>.9800</b>	<b>.9758</b>	<b>.9735</b>	<b>.0087</b>
CSNet <sub>20</sub> [40]	LN	224×224	38	0.14	0.7	.8364	.8341	.7656	.6319	.9535	.8929	.8339	.0169	.8910	.8790	.8285	.7615	.9628	.9171	.9068	.0186
SAMNet <sub>21</sub> [41]	LN	336×336	44	1.33	0.5	.8622	.7813	.7214	.6114	.9421	.8700	.8284	.0132	.8761	.8137	.7531	.6843	.9478	.8818	.8656	.0217
HVPNet <sub>21</sub> [13]	LN	336×336	26	1.23	1.1	.8734	.8036	.7377	.6202	.9482	.8721	.8270	.0110	.8610	.7938	.7396	.6726	.9320	.8717	.8471	.0225
CorrNet <sub>22</sub> [15]	LR	256×256	100	4.09	21.1	<b>.9289</b>	<b>.8778</b>	<b>.8620</b>	<b>.8311</b>	.9696	<b>.9646</b>	<b>.9593</b>	.0083	<b>.9380</b>	<b>.9129</b>	<b>.9002</b>	<b>.8875</b>	<b>.9790</b>	<b>.9746</b>	<b>.9721</b>	<b>.0098</b>
<b>SeaNet (Ours)</b>	LR	288×288	96	2.76	1.7	.9208	.8649	.8519	<b>.8304</b>	.9710	<b>.9651</b>	<b>.9602</b>	<b>.0073</b>	.9260	.8942	.8772	<b>.8625</b>	.9767	<b>.9722</b>	<b>.9670</b>	<b>.0105</b>

CN: CNN-based NSI-SOD method, CR: CNN-based ORSI-SOD method, LN: lightweight NSI-SOD method, LR: lightweight ORSI-SOD method.

NSI-SOD methods (i.e., DSS [17], RADF [25], R3Net [26], PoolNet [50], EGNet [19], GCPA [31], MINet [30], ITSD [20], GateNet [18], SUCA [32], and PA-KRN [12]) and six conventional ORSI-SOD methods (i.e., LVNet [21], DAFNet [38], SARNet [35], MJRBM [39], EMFNet [22], and MCCNet [14]). In addition, we also compare our lightweight SeaNet with four state-of-the-art lightweight SOD methods, including three lightweight NSI-SOD methods (i.e., CSNet [40], SAMNet [41], and HVPNet [13]) and the lightweight ORSI-SOD method CorrNet [15]. The above NSI-SOD methods are retrained on the same ORSI-SOD datasets as our SeaNet with default parameter settings to generate saliency maps. We obtain saliency maps for other methods from authors or public source codes.

1) *Comparison With Conventional SOD Methods*: The top of Table II shows the quantitative evaluation and computational complexity evaluation results of our SeaNet and conventional SOD methods for NSIs and ORSIs. Compared with conventional NSI-SOD methods, our SeaNet achieves the best performance in terms of both accuracy and computational complexity. For example, compared with the best performing NSI-SOD solution PA-KRN [12], SeaNet has significant advantages on  $\mathcal{M}$ , for example, 0.0073 versus 0.0104 on the EORSSD dataset and 0.0139 versus 0.0105 on the ORSSD dataset and has 6× faster inference speed, 51.1× fewer parameters, and 363.3× fewer FLOPs than it. This shows the advantages of specialized methods, even our lightweight specialized method can outperform conventional NSI-SOD solutions.

Compared with conventional ORSI-SOD methods, our SeaNet shows comparable accuracy but with significantly

lower computational complexity. For example, compared with EMFNet [22], SeaNet achieves similar accuracy, for example,  $E_\xi^{\max}$ : 0.9710 versus 0.9711 on the EORSSD dataset and 0.9767 versus 0.9737 on the ORSSD dataset, while SeaNet is 3.6× faster in inference speed, and 24.5× and 282.8× fewer in parameters and FLOPs, respectively. Compared with the best-performing MCCNet [14], although the accuracy of SeaNet is not dominant, the lower computational complexity of SeaNet is still outstanding.

2) *Comparison With Lightweight SOD Methods*: At the bottom of Table II, we report the comparison results of our SeaNet with four lightweight SOD methods for NSIs and ORSIs. Compared with lightweight NSI-SOD methods, SeaNet has no advantages in parameters and FLOPs, but has obvious advantages in inference speed. This means that our lightweight SeaNet has room for improvement. On the other hand, the accuracy advantage of SeaNet is obvious, for example, SeaNet outperforms them by 3.50%–8.44% in  $S_\alpha$ , 4.87%–13.76% in  $F_\beta^{\text{mean}}$ , 5.51%–10.05% in  $E_\xi^{\text{mean}}$ , and 0.0037–0.0120 in  $\mathcal{M}$  on two datasets.

The original intention of our SeaNet is to reduce the number of parameters and FLOPs of existing lightweight ORSI-SOD method CorrNet [15], especially the latter one. The results in Table II show that SeaNet achieves this goal without significantly reducing accuracy. Specifically, in terms of computational complexity, SeaNet has 1.3× fewer parameters and 12.4× fewer FLOPs than CorrNet and has a comparable inference speed. In terms of accuracy, SeaNet shows a slightly lower  $E_\xi^{\text{adp}}$  (0.9670 versus 0.9721) than CorrNet on the ORSSD dataset, while achieves a slightly higher  $\mathcal{M}$  (0.0073 versus 0.0083) on the EORSSD dataset.

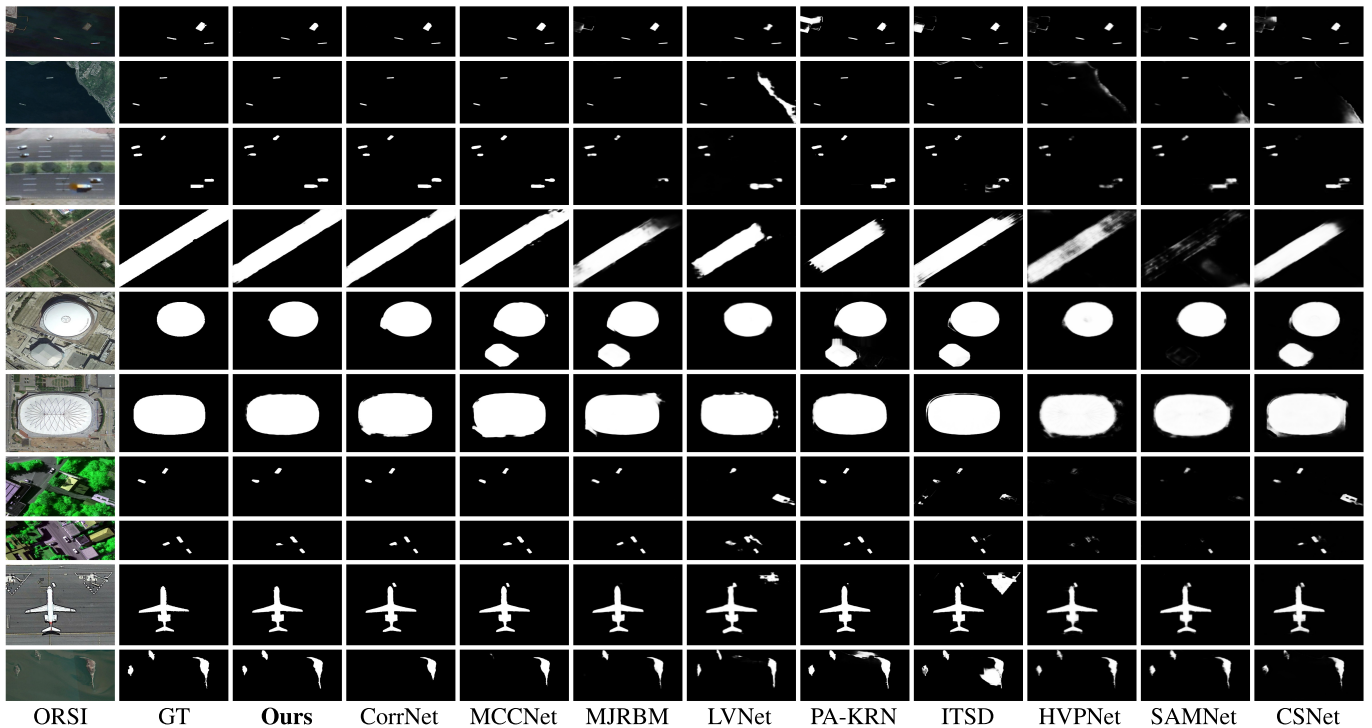


Fig. 4. Qualitative comparisons with four types of methods, including nine representative state-of-the-art methods.

Overall, SeaNet achieves one first place, two second places, and five third places in quantitative evaluation metrics with much more desirable computational complexity, striking a balance between effectiveness and efficiency. This means that SeaNet is a promising method that can be applied in practical applications.

3) *Qualitative Comparison*: Here, we show the qualitative comparison of our SeaNet and four types of methods, including nine representative state-of-the-art methods, on four unique and challenging ORSI scenes in Fig. 4. The first scene contains multiple objects or even multiple tiny objects, as shown in the first three cases of Fig. 4. We can observe that the saliency maps of our SeaNet are similar to those of specialized ORSI-SOD methods, such as CorrNet and MCCNet, which accurately highlight all salient objects, and are much better than those of conventional and lightweight NSI-SOD methods. This is attributed to the multiscale semantic matching of DSMM. The second scene contains a big object, such as the fourth to sixth cases of Fig. 4. Most methods can locate the big object, but cannot highlight them completely, while our SeaNet can highlight the entire big object with complete edges. This is attributed to the edge-based detail enhancement of ESAM. The third scene contains chaotic background, such as the seventh and eighth cases of Fig. 4. Our SeaNet can successfully find salient objects in complex backgrounds, while the three lightweight methods either miss objects (HVPNet and SAMNet) or introduce background regions (CSNet). The last scene contains objects with complex geometric shapes, such as the last two cases of Fig. 4. Thanks to the cooperation between DSMM and ESAM, our SeaNet has obvious advantages compared to the nine methods, that is, it can not only overcome the interference of the small object, but also precisely locate three islands with fine details. Overall, our SeaNet can catch up with

TABLE III

ABLATION RESULTS OF EVALUATING THE CONTRIBUTION OF TWO LIGHTWEIGHT MODULES. THE BEST ONE IN EACH COLUMN IS **BOLD**

Models	#Param (M)↓	FLOPs (G)↓	EORSSD [38]			
			$S_\alpha$ ↑	$F_\beta^{\text{mean}}$ ↑	$E_\xi^{\text{mean}}$ ↑	$\mathcal{M}$ ↓
SeaNet ( <b>Ours</b> )	2.76	1.66	<b>0.9208</b>	<b>0.8519</b>	<b>0.9651</b>	<b>0.0073</b>
w/o DSMM	2.75 -0.01	1.56 -0.10	0.9158	0.8440	0.9580	0.0093
w/o ESAM	2.70 -0.06	1.59 -0.07	0.9180	0.8404	0.9588	0.0084

specialized ORSI-SOD methods and outperform NSI-SOD methods.

### C. Ablation Studies

To evaluate the effectiveness of each component of our SeaNet, we conduct exhaustive ablation studies on the EORSSD dataset. Specifically, we analyze: 1) the contribution of two lightweight modules; 2) the effectiveness of each component of ESAM; and 3) the effectiveness of each component of DSMM. The parameter settings and datasets for each variant are the same as in Section IV-A.

1) *Contribution of Two Lightweight Modules*: To analyze the contribution of two lightweight modules, we provide two variants: 1) removing DSMM and SKC (i.e., *without DSMM*) and 2) removing ESAM (i.e., *without ESAM*). We report the quantitative results and computational complexity in Table III.

We observe that our two modules are lightweight, that is, the combination of DSMM and SKC have 0.01 M parameters and 0.10 G FLOPs, and ESAM has 0.06 M parameters and 0.07 G FLOPs. Since DSMM can determine the location of salient objects, *without DSMM* reduces the accuracy of object localization, resulting in a drastic drop



TABLE IV

ABLATION RESULTS OF EMBEDDING THE TWO LIGHTWEIGHT MODULES INTO HVPNET AND SAMNET

Models	#Param FLOPs		EORSSD [38]			
	(M)↓	(G)↓	$S_\alpha \uparrow$	$F_\beta^{\text{mean}} \uparrow$	$E_\xi^{\text{mean}} \uparrow$	$\mathcal{M} \downarrow$
SeaNet (Ours)	2.76	1.7	0.9208	0.8519	0.9651	0.0073
SAMNet	1.33	0.5	0.8622	0.7214	0.8700	0.0132
SeaNet-SAM	1.51	1.4	0.9173	0.8507	0.9631	0.0074
HVPNet	1.23	1.1	0.8734	0.7377	0.8721	0.0110
SeaNet-HVP	1.48	1.8	0.9202	0.8486	0.9646	0.0069

in pixel-level evaluation metric, that is,  $\mathcal{M}$ : 0.0073→0.0093. *without EASM* only affects the details of salient objects, so the performance degradation is generally less severe than that of *without DSMM*, for example,  $\mathcal{M}$ : 0.0073→0.0084 and  $E_\xi^{\text{mean}}$ : 0.9651→0.9588. The cooperation of these two lightweight modules and MobileNet-V2 enables our SeaNet to achieve good performance without many parameters and computational costs.

In particular, to further illustrate the effectiveness, flexibility, and robustness of these two lightweight modules, we embed these two lightweight modules and our decoder into the feature extraction backbones proposed by SAMNet and HVPNet, forming two variants, named SeaNet-SAM and SeaNet-HVP, respectively. Notably, our decoder has 0.47 M parameters and 0.95 G FLOPs. As shown in Table IV, SeaNet-SAM is more lightweight than our original SeaNet. SeaNet-HVP has significantly fewer parameters and slightly higher FLOPs than our original SeaNet. Both variants have comparable performance to our original SeaNet, which shows that these two lightweight modules can be adapted to different backbones and are robust. In addition, SeaNet-SAM/SeaNet-HVP has a similar number of parameters as SAMNet/HVPNet, while having higher FLOPs which are mainly from the decoder. The performance of SeaNet-SAM/SeaNet-HVP is significantly better than that of SAMNet/HVPNet, such as leading by more than 10% on  $F_\beta^{\text{mean}}$ . Notably, the number of parameters and FLOPs of SeaNet-SAM (1.51 M and 1.4 G) and HVPNet (1.23 M and 1.1 G) are comparable, while SeaNet-SAM outperforms HVPNet by a large margin. This situation proves that when the number of parameters and FLOPs of our SeaNet variants are equivalent or comparable to those of other lightweight methods, our SeaNet variants still significantly outperform them.

2) *Effectiveness of Each Component of DSMM*: To analyze the effectiveness of each component of DSMM, we design three variants of DSMM in Table V and embed them into the network: 1) removing the spatial semantic matching (i.e., *without SM*); 2) changing dilated DDconvs to regular DDconvs (i.e., *without dilation*); and 3) removing the channel-wise correlation of DSMM (i.e., *without CCorr1*).

Based on the quantitative performance at the top of Table III, we observe that each component of DSMM is necessary. As the key part of DSMM, *without SM* truncates the object localization capability of DSMM, achieving the worst performance among these three variants, for example,

TABLE V

ABLATION RESULTS OF EVALUATING THE EFFECTIVENESS OF EACH COMPONENT OF DSMM AND EASM. THE BEST ONE IN EACH COLUMN IS **BOLD**

	Models	EORSSD [38]			
		$S_\alpha \uparrow$	$F_\beta^{\text{mean}} \uparrow$	$E_\xi^{\text{mean}} \uparrow$	$\mathcal{M} \downarrow$
	SeaNet (Ours)	<b>0.9208</b>	<b>0.8519</b>	<b>0.9651</b>	<b>0.0073</b>
DSMM	<i>w/o SM</i>	0.9170	0.8459	0.9595	0.0078
	<i>w/o dilation</i>	0.9194	0.8490	0.9626	0.0076
	<i>w/o CCorr1</i>	0.9187	0.8471	0.9619	0.0080
EASM	<i>w/o EEU</i>	0.9183	0.8441	0.9597	0.0078
	<i>w/o alignment</i>	0.9189	0.8476	0.9617	0.0078
	<i>w/o CCorr2</i>	0.9197	0.8494	0.9627	0.0076

$E_\xi^{\text{mean}}$ : 0.9651→0.9595. *without dilation* impairs the ability of DSMM to perceive salient objects of different sizes, resulting in a slight performance drop, for example, a drop of 0.29% on  $F_\beta^{\text{mean}}$ . *Without CCorr1* only focuses on the feature interactions at the spatial level, while ignoring that at the channel level, resulting in incomplete feature interaction and performance degradation. Therefore, the above analysis proves that the design of our DSMM is reasonable and effective.

3) *Effectiveness of Each Component of EASM*: To analyze the effectiveness of each component of EASM, we design three variants of EASM in Table V and embed them into the network: 1) removing two EEUs (i.e., *without EEU*); 2) removing the edge alignment (i.e., removing the edge alignment loss in (10), named *without alignment*); and 3) removing the channel-wise correlation of EASM (i.e., *without CCorr2*).

According to the quantitative performance at the bottom of Table III, we observe that each component of EASM contributes to the final performance. The two EEUs are responsible for enhancing the edge regions on the features, so *without EEU* does not achieve satisfactory performance, for example,  $F_\beta^{\text{mean}}$ : 0.8519→0.8441. The proposed edge alignment is an interesting mechanism that can improve the accuracy of edge information without increasing parameters and FLOPs. *without alignment* causes performance degradation on all four metrics. Like *without CCorr1*, *without CCorr2* also gives up the channel-level feature interactions in EASM, which hurts performance. Combining *without CCorr1* and *without CCorr2*, we can conclude that channel-level feature interactions are important to our SeaNet and cannot be discarded. The above analysis shows that these components of EASM are indispensable.

## V. CONCLUSION

In this article, we aim to treat low- and high-level features discriminatively, thereby proposing an efficient solution, named SeaNet, for lightweight ORSI-SOD. For high-level features, the lightweight DSMM is proposed to explore object locations through spatial semantic matching and takes into account the channel-level feature interactions. Spatial semantic matching utilizes dilated DDconvs to perceive multiple salient objects and objects with variable sizes, resulting in good adaptation to complex scenes of ORSIs. For low-level features, the

lightweight ESAM is proposed to enhance the details of salient objects based on edge information which is corrected in an innovative self-alignment manner. With the close cooperation of the two modules, SeaNet infers salient objects accurately in the decoder at a fast speed. Performance analysis and ablation studies demonstrate the effectiveness and efficiency of our SeaNet compared with state-of-the-art methods.

## REFERENCES

- [1] R. Cong, J. Lei, H. Fu, M.-M. Cheng, W. Lin, and Q. Huang, "Review of visual saliency detection with comprehensive information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 10, pp. 2941–2959, Oct. 2019.
- [2] K. Gu et al., "Saliency-guided quality assessment of screen content images," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 1098–1110, Jun. 2016.
- [3] S. Yang, Q. Jiang, W. Lin, and Y. Wang, "SGDNet: An end-to-end saliency-guided deep neural network for no-reference image quality assessment," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1383–1391.
- [4] G. Li, Z. Liu, R. Shi, and W. Wei, "Constrained fixation point based segmentation via deep neural network," *Neurocomputing*, vol. 368, pp. 180–187, Dec. 2019.
- [5] G. Li et al., "Personal fixations-based object segmentation with object localization and boundary preservation," *IEEE Trans. Image Process.*, vol. 30, pp. 1461–1475, 2021.
- [6] P. Zhang, W. Liu, D. Wang, Y. Lei, H. Wang, and H. Lu, "Non-rigid object tracking via deep multi-scale spatial-temporal discriminative saliency maps," *Pattern Recognit.*, vol. 100, Apr. 2020, Art. no. 107130.
- [7] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: An in-depth survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3239–3259, Jun. 2022.
- [8] G. Li, Z. Liu, and H. Ling, "ICNeT: Information conversion network for RGB-D based salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 4873–4884, 2020.
- [9] G. Li, Z. Liu, L. Ye, Y. Wang, and H. Ling, "Cross-modal weighting network for RGB-D salient object detection," in *Proc. ECCV*, Aug. 2020, pp. 665–681.
- [10] W. Wang, J. Shen, J. Xie, M. Cheng, H. Ling, and A. Borji, "Revisiting video saliency prediction in the deep learning era," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 220–237, Jan. 2021.
- [11] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
- [12] B. Xu, H. Liang, R. Liang, and P. Chen, "Locate globally, segment locally: A progressive architecture with knowledge review network for salient object detection," in *Proc. AAAI*, Feb. 2021, pp. 3004–3012.
- [13] Y. Liu, Y.-C. Gu, X.-Y. Zhang, W. Wang, and M.-M. Cheng, "Lightweight salient object detection via hierarchical visual perception learning," *IEEE Trans. Cybern.*, vol. 51, no. 9, pp. 4439–4449, Sep. 2021.
- [14] G. Li, Z. Liu, W. Lin, and H. Ling, "Multi-content complementation network for salient object detection in optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5614513.
- [15] G. Li, Z. Liu, Z. Bai, W. Lin, and H. Ling, "Lightweight salient object detection in optical remote sensing images via feature correlation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5617712.
- [16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.
- [17] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. Torr, "Deeply supervised salient object detection with short connections," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5300–5309.
- [18] X. Zhao, Y. Pang, L. Zhang, H. Lu, and L. Zhang, "Suppress and balance: A simple gated network for salient object detection," in *Proc. Conf. Artif. Intell. Assoc. Adv. Artif. Intell. (AAAI)*, Feb. 2020, pp. 35–51.
- [19] J.-X. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "EGNet: Edge guidance network for salient object detection," in *Proc. IEEE ICCV*, Oct. 2019, pp. 8779–8788.
- [20] H. Zhou, X. Xie, J. Lai, Z. Chen, and L. Yang, "Interactive two-stream decoder for accurate and fast saliency detection," in *Proc. CVPR*, Jun. 2020, pp. 9138–9147.
- [21] C. Li, R. Cong, J. Hou, S. Zhang, Y. Qian, and S. Kwong, "Nested network with two-stream pyramid for salient object detection in optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9156–9166, Nov. 2019.
- [22] X. Zhou, K. Shen, Z. Liu, C. Gong, J. Zhang, and C. Yan, "Edge-aware multiscale feature integration network for salient object detection in optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5605315.
- [23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jan. 2018, pp. 4510–4520.
- [24] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
- [25] X. Hu, L. Zhu, J. Qin, C.-W. Fu, and P.-A. Heng, "Recurrently aggregating deep features for salient object detection," in *Proc. AAAI*, Feb. 2018, pp. 6943–6950.
- [26] Z. Deng et al., "R<sup>3</sup>Net: Recurrent residual refinement network for saliency detection," in *Proc. IJCAI*, Jul. 2018, pp. 684–690.
- [27] M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1623–1632.
- [28] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: Boundary-aware salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7479–7489.
- [29] M. S. Lee, W. Shin, and S. W. Han, "TRACER: Extreme attention guided salient object tracing network," in *Proc. AAAI*, 2022, pp. 12993–12994.
- [30] Y. Pang, X. Zhao, L. Zhang, and H. Lu, "Multi-scale interactive network for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9410–9419.
- [31] Z. Chen, Q. Xu, R. Cong, and Q. Huang, "Global context-aware progressive aggregation network for salient object detection," in *Proc. AAAI*, Feb. 2020, pp. 10599–10606.
- [32] J. Li, Z. Pan, Q. Liu, and Z. Wang, "Stacked U-shape network with channel-wise attention for salient object detection," *IEEE Trans. Multimedia*, vol. 23, pp. 1397–1409, 2021.
- [33] Y. Y. Ke and T. Tsubono, "Recursive contour-saliency blending network for accurate salient object detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1360–1370.
- [34] C. Li et al., "A parallel down-up fusion network for salient object detection in optical remote sensing images," *Neurocomputing*, vol. 415, pp. 120–411, Nov. 2020.
- [35] Z. Huang, H. Chen, B. Liu, and Z. Wang, "Semantic-guided attention refinement network for salient object detection in optical remote sensing images," *Remote Sens.*, vol. 13, no. 11, p. 2163, May 2021.
- [36] R. Cong, Y. Zhang, L. Fang, J. Li, Y. Zhao, and S. Kwong, "RRNet: Relational reasoning network with parallel multiscale attention for salient object detection in optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5613311.
- [37] G. Li, Z. Liu, D. Zeng, W. Lin, and H. Ling, "Adjacent context coordination network for salient object detection in optical remote sensing images," *IEEE Trans. Cybern.*, vol. 53, no. 1, pp. 526–538, Jan. 2023.
- [38] Q. Zhang et al., "Dense attention fluid network for salient object detection in optical remote sensing images," *IEEE Trans. Image Process.*, vol. 30, pp. 1305–1317, 2021.
- [39] Z. Tu, C. Wang, C. Li, M. Fan, H. Zhao, and B. Luo, "ORSI salient object detection via multiscale joint region and boundary model," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5607913.
- [40] S.-H. Gao, Y.-Q. Tan, M.-M. Cheng, C. Lu, Y. Chen, and S. Yan, "Highly efficient salient object detection with 100 k parameters," in *Proc. ECCV*, Aug. 2020, pp. 702–721.
- [41] Y. Liu, X.-Y. Zhang, J.-W. Bian, L. Zhang, and M.-M. Cheng, "SAMNet: Stereoscopically attentive multi-scale network for lightweight salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 3804–3814, 2021.
- [42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.
- [43] Y.-H. Wu, Y. Liu, J. Xu, J.-W. Bian, Y.-C. Gu, and M.-M. Cheng, "MobileSal: Extremely efficient RGB-D salient object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 10261–10269, Dec. 2022.
- [44] Y. Zhang, J. Zhang, Q. Wang, and Z. Zhong, "DyNet: Dynamic convolution for accelerating convolutional neural networks," 2020, *arXiv:2004.10694*.
- [45] G. Li, Z. Liu, M. Chen, Z. Bai, W. Lin, and H. Ling, "Hierarchical alternate interaction network for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 3528–3542, 2021.

- [46] Z. Bai, Z. Liu, G. Li, and Y. Wang, "Adaptive group-wise consistency network for co-saliency detection," *IEEE Trans. Multimedia*, early access, Dec. 24, 2021, doi: 10.1109/TMM.2021.3138246.
- [47] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Proc. NeurIPS*, Dec. 2016, pp. 289–297.
- [48] X. Lu, W. Wang, J. Shen, D. Crandall, and J. Luo, "Zero-shot video object segmentation with co-attention Siamese networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2228–2242, Apr. 2022.
- [49] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. ICLR*, May 2016, pp. 1–13.
- [50] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3912–3921.
- [51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jun. 2014.
- [52] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2015, pp. 1026–1034.
- [53] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4548–4557.
- [54] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1597–1604.
- [55] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 698–704.
- [56] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, Dec. 2019, pp. 8024–8035.
- [57] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, May 2015, pp. 1–15.



**Xinpeng Zhang** (Member, IEEE) received the B.S. degree from Jilin University, Changchun, China, in 1995, and the M.S. and Ph.D. degrees from Shanghai University, Shanghai, China, in 2001 and 2004, respectively.

Since 2004, he has been with the Faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a full-time Professor. He is also with the Faculty of the School of Computer Science, Fudan University, Shanghai. He was with The State University of New York at Binghamton, Binghamton, NY, USA, as a Visiting Scholar from 2010 to 2011, and also with Konstanz University, Konstanz, Germany, as an Experienced Researcher, sponsored by the Alexander von Humboldt Foundation from 2011 to 2012. He has published over 200 research articles. His research interests include multimedia security, image processing, and digital forensics.

Dr. Zhang was an Associate Editor of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY from 2014 to 2017.



**Gongyang Li** received the Ph.D. degree from Shanghai University, Shanghai, China, in 2022.

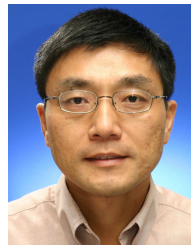
From July 2021 to June 2022, he was a Visiting Ph.D. Student with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He is currently a Post-Doctoral Fellow with the School of Communication and Information Engineering, Shanghai University. His research interests include image/video object segmentation, semantic segmentation, and saliency detection.



**Zhi Liu** (Senior Member, IEEE) received the B.E. and M.E. degrees from Tianjin University, Tianjin, China, in 1999 and 2002, respectively, and the Ph.D. degree from the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China, in 2005.

From August 2012 to August 2014, he was a Visiting Researcher with the SIROCCO Team, IRISA/INRIA-Rennes, Rennes, France, with the support of EU FP7 Marie Curie Actions. He is currently a Professor with the School of Communication and Information Engineering, Shanghai University, Shanghai. He has published more than 200 refereed technical papers in international journals and conferences. His research interests include image/video processing, machine learning, computer vision, and multimedia communication.

Dr. Liu was a TPC Member/Session Chair of ICIP 2017, PCM 2016, VVIP 2016, ICME 2014, and WIAMIS 2013. He has co-organized special sessions on visual attention, saliency models, and applications at WIAMIS 2013 and ICME 2014. He is an Area Editor of *Signal Processing: Image Communication* and served as a Guest Editor for the Special Issue on Recent Advances in Saliency Models, Applications and Evaluations in *Signal Processing: Image Communication*.



**Weisi Lin** (Fellow, IEEE) received the Ph.D. degree from the King's College London, London, U.K., in 1993.

He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He has published more than 340 refereed articles published in international journals and conferences. His research interests include image processing, visual quality evaluation, and perception-inspired signal modeling.

Dr. Lin is a fellow of IET and an Honorary Fellow of the Singapore Institute of Engineering Technologists. He was a Distinguished Lecturer of the IEEE Circuits and Systems Society from 2016 to 2017 and the Asia-Pacific Signal and Information Processing Association (APSIPA) from 2012 to 2013. He was a Technical-Program Chair for Pacific-Rim Conference on Multimedia 2012, the IEEE International Conference on Multimedia and Expo 2013, and the International Workshop on Quality of Multimedia Experience 2014. He has been on the Editorial Board of the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE SIGNAL PROCESSING LETTERS, and *Journal of Visual Communication and Image Representation*.