Lightweight Salient Object Detection in Optical Remote Sensing Images via Feature Correlation

Gongyang Li, Zhi Liu, *Senior Member, IEEE*, Zhen Bai, Weisi Lin, *Fellow, IEEE*, and Haibin Ling, *Senior Member, IEEE*

***Abstract*— Salient object detection in optical remote sensing images (ORSI-SOD) has been widely explored for understanding ORSIs. However, previous methods focus mainly on improving the detection accuracy while neglecting the cost in memory and computation, which may hinder their real-world applications. In this article, we propose a novel lightweight ORSI-SOD solution, named CorrNet, to address these issues. In CorrNet, we first lighten the backbone (VGG-16) and build a lightweight subnet for feature extraction. Then, following the coarse-to-fine strategy, we generate an initial coarse saliency map from high-level semantic features in a correlation module (CorrM). The coarse saliency map serves as the location guidance for low-level features. In CorrM, we mine the object location information between high-level semantic features through the cross-layer correlation operation. Finally, based on low-level detailed features, we refine the coarse saliency map in the refinement subnet equipped with dense lightweight refinement blocks (DLRBs) and produce the final fine saliency map. By reducing the parameters and computations of each component, CorrNet ends up having only 4.09M parameters and running with 21.09G FLOPs. Experimental results on two public datasets demonstrate that our lightweight CorrNet achieves competitive or even better performance compared with 26 state-of-the-art methods (including 16 large CNN-based methods and two lightweight methods), and meanwhile enjoys the clear memory and run-time efficiency. The code and results of our method are available at https://github.com/MathLee/CorrNet.**

***Index Terms*— Cross-layer correlation, dense lightweight refinement block (DLRB), lightweight salient object detection (SOD), optical remote sensing image (ORSI).**

I. Introduction

SALIENT object detection (SOD) [1]–[3] focuses on extracting the visually distinctive objects or regions in a scene, and often serves as an important preprocessing step in computer vision. It has been successfully applied in image retargeting [4], image quality assessment [5], [6], object segmentation [7], [8], etc. In recent decades, there have been many branches of SOD, such as SOD in natural scene images (NSI-SOD) [1], video SOD [2], RGB-D SOD [9], [10], co-saliency detection [11], SOD in optical remote sensing images (ORSI-SOD) [12], etc. In this article, we are committed to an emerging topic in SOD, i.e., ORSI-SOD. ORSIs are photographed by satellites and aerial sensors, and have three optical bands (i.e., red, green, and blue bands), which are the same as NSIs. The scenes of ORSIs are completely different from NSIs and are very challenging. ORSI-SOD can discover attractive objects, which is conducive to quickly analyzing and understanding ORSIs.

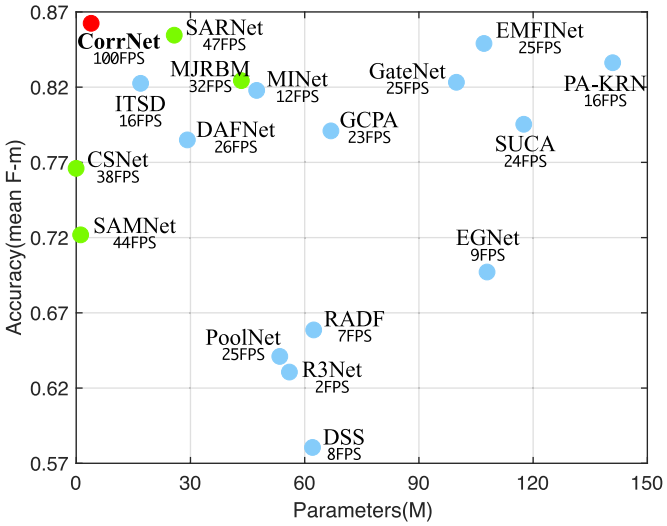


Fig. 1. Accuracy, parameters, and inference speed comparisons of our CorrNet and other CNN-based methods on the EORSSD dataset [13]. ● represents real-time methods, ● represents nonreal-time methods, and ● represents our CorrNet.

Early traditional NSI-SOD methods [14] mainly relied on hand-crafted features, which usually lead to unsatisfactory detection accuracy. Recently, convolutional neural networks (CNNs) [15] have demonstrated powerful capabilities in computer vision, and greatly promoted the development of NSI-SOD algorithms [1], which often generate satisfactory saliency maps. However, the improvement in detection accuracy often comes from more complicated network structures,

Manuscript received December 9, 2021; revised January 9, 2022; accepted January 19, 2022. Date of publication January 25, 2022; date of current version March 14, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62171269, in part by the China Scholarship Council under Grant 202006890079, and in part by the Singapore Ministry of Education Tier-2 Fund under Grant MOE2016-T2-2-057(S). *(Corresponding author: Zhi Liu.)*

Gongyang Li, Zhi Liu, and Zhen Bai are with Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China, and also with the School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China (e-mail: ligongyang@shu.edu.cn; liuzhisjtu@163.com; bz536476@163.com).

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

Haibin Ling is with the Department of Computer Science, Stony Brook University, Stony Brook, NY 11794 USA (e-mail: hling@cs.stonybrook.edu).

Digital Object Identifier 10.1109/TGRS.2022.3145483

which typically come with a large amount of parameters and increased computational complexity. Since ORSIs and NSIs have gaps in the scene, it is not appropriate to directly migrate NSI-SOD to ORSIs, but most of the existing ORSI-SOD methods [13], [16]–[18] are affected by NSI-SOD methods. Therefore, ORSI-SOD methods usually have a large computational consumption and memory burden, and are accompanied by limited inference speed.

In Fig. 1, we show the detection accuracy of recent NSI-SOD methods (PA-KRN [19], SUCA [20], GateNet [21], and EGNet [22]) and ORSI-SOD methods (DAFNet [13], EMFINet [16], MJRBM [17], and SARNet [18]) on an ORSI-SOD dataset, namely EORSSD [13]. We also report their parameters and inference speed in Fig. 1. Although these methods show good performance, their parameters are amazing and inference speeds are slow, for example, PA-KRN has 141.06M parameters with only 16 fps and EMFINet has 107.26M parameters with 25 fps. And the performance of the two lightweight NSI-SOD methods (CSNet [23] and SAMNet [24]) is slightly inferior. Considering the application scenarios of ORSI-SOD, we believe that ORSI-SOD is in urgent need of a lightweight solution with fewer parameters, faster speed, and good accuracy.

Inspired by the above observations, in this article, we propose a novel lightweight solution for ORSI-SOD, namely CorrNet, which is the first lightweight ORSI-SOD model as we know. In CorrNet, we mainly realize the lightweight framework from two aspects: 1) lightening the backbone and 2) designing lightweight modules. For the backbone, previous methods [13], [16]–[18] usually adopt the pretrained VGG [25] or ResNet [26] as the backbone, but such backbones suffer from a large number of parameters despite their powerful feature extraction capabilities. To achieve a balance between the feature extraction capabilities and the amount of parameters, we modify the vanilla VGG [25] and build a lightweight but powerful backbone for feature extraction. For the lightweight modules, we use the depthwise separable convolution [27], [28] instead of the regular one, which can reduce the parameters of regular convolution by about 90%. In this way, our CorrNet has only 4.09M parameters.

Moreover, to keep a good detection accuracy of CorrNet, we implement it following the coarse-to-fine strategy with two novel modules. For the coarse part, we explore the object location information among two groups of high-level semantic features in the correlation module (CorrM), and obtain the initial coarse saliency map. Then, we refine it with other low-level detailed features in the refinement subnet, which consists of several dense lightweight refinement blocks (DLRBs), and obtain the final fine saliency map. With all components working together, our CorrNet achieves excellent performance in accuracy (86.20% in mean F-measure on the EORSSD dataset [13]), parameters (4.09M), and inference speed (100 fps), as shown in Fig. 1.

Our main contributions are summarized as follows.

1) We explore the lightweight framework of ORSI-SOD for the first time. To this end, we propose a novel lightweight CorrNet (only 4.09M parameters) that uses the coarse-to-fine strategy.
2) We propose a CorrM to explore the cross-layer correlation of high-level semantic context, generating an initial coarse saliency map to low-level features for location guidance.
3) We propose a DLRB to merge the enhanced feature embeddings and the refined features for finely sculpting salient objects, gradually producing the final fine saliency map.
4) We evaluate the proposed CorrNet against 26 state-of-the-art methods on two ORSI-SOD datasets. Experiments demonstrate that the proposed CorrNet achieves better or competitive performance compared with previously proposed large CNN-based methods.

We organize the rest of this article as follows. In Section II, we review the related work of ORSI-SOD. In Section III, we elaborate our CorrNet. In Section IV, we conduct experiments and ablation studies. Finally, in Section V, we draw the conclusion.

II. Related Work

A. Lightweight Methods for NSI-SOD

The lightweight NSI-SOD task is an emerging direction in NSI-SOD, which aims to propose a solution suitable for edge computing devices. Gao *et al.* [23] proposed an extremely lightweight network with only about 100K parameters and 95.3-ms run-time on a single core i7-8700K CPU. Liu *et al.* [24] constructed a stereoscopic attention mechanism-based backbone for lightweight NSI-SOD. Liu *et al.* [29] imitated the primate visual hierarchies and proposed the hierarchical visual perception network for better multiscale learning. However, lightweight ORSI-SOD is still a desert. In this article, we propose an effective and efficient solution for lightweight ORSI-SOD for the first time. Different from the above lightweight NSI-SOD methods that focus on designing lightweight backbones, we focus on lightening the existing backbone (i.e., VGG-16) and proposing effective and lightweight modules.

B. Traditional Methods for ORSI-SOD

Similar to traditional NSI-SOD methods [14], traditional ORSI-SOD methods also mainly rely on hand-crafted features. Faur *et al.* [30] presented a rate distortion-based estimation method and considered the mean-shift algorithm to segment the remote sensing images. Zhang *et al.* [31] applied color information content analysis to ORSIs and then computed the saliency scores of each color channel and fused color components for final results. Zhao *et al.* [32] introduced the high-level global and background cues for saliency map integration. Zhang *et al.* [33] combined the super-pixel and statistical saliency feature analysis for ORSI-SOD. Based on low-rank matrix recovery, Zhang *et al.* [34] proposed a self-adaptive fusion method to fuse color feature, intensity feature, texture, and global contrast for saliency detection in ORSIs. Huang *et al.* [35] proposed a contrast-weighted dictionary learning-based method for VHR RSI saliency detection, which follows the procedure of discriminant dictionary construction, saliency measurement, and saliency fusion.

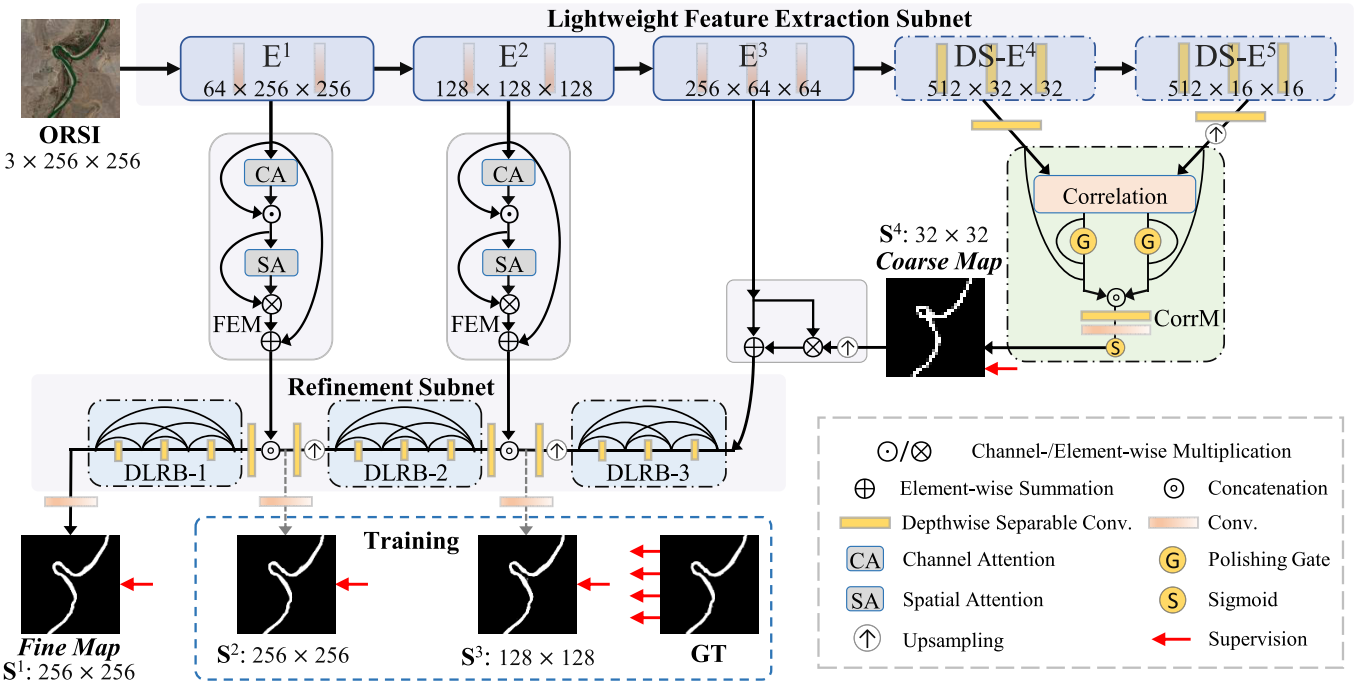


Fig. 2. Overall framework of the proposed CorrNet. First, we adopt a lightweight feature extraction subnet, which is a variant of classic VGG-16 [25], to extract the basic feature embeddings. Then, we model the cross-layer correlation between two groups of high-level semantic features in the CorrM, and generate the initial coarse saliency map $\mathbf{S}^4$. Meanwhile, the low-level detailed features are enhanced in the general feature enhancement module (FEM). Finally, we refine the coarse saliency map with the enhanced features in the refinement subnet, which consists of three DLRBs, and generate the final fine saliency map $\mathbf{S}^1$. Notably, in the training phase, we adopt the deep supervision.

In addition to general ORSI-SOD methods, there are some traditional methods for specific scenes of ORSIs. For ship detection, Chen *et al.* [36] proposed a contour refinement and the improved generalized Hough transform-based method to handle complex harbor scenes. For oil tank detection, Liu *et al.* [37] constructed a color Markov chain in the CIE Lab space to generate a bottom-up latent saliency map. For airport detection, Zhang *et al.* [38] proposed a complementary saliency analysis and saliency-oriented active contour model. For residential areas extraction, based on complementarities, Zhang *et al.* [39] merged two global maps and one local map to achieve complete residential areas.

Since hand-crafted features are usually accompanied by a large amount of computational consumption and memory burden, the above traditional methods are not efficient enough for practical applications.

C. CNN-Based Methods for ORSI-SOD

Taking the advantage of the powerful feature representation capabilities of CNNs, many CNN-based ORSI-SOD methods have shown good performance. In order to meet the data requirements of CNN-based methods, Li *et al.* [12] and Zhang *et al.* [13] constructed two challenging datasets for ORSI-SOD, namely ORSSD and EORSSD. Based on these two datasets, a large number of CNN-based methods have emerged.

Li *et al.* [12] constructed a L-shape module (i.e., the two-stream pyramid module) and a V-shape module (i.e., the encoder–decoder module with nested connections) based on features extracted from five different-resolution ORSIs. Zhou *et al.* [16] followed the multiinput strategy, and introduced edge features to complete salient regions in feature level. Edge information plays an important role in ORSI-SOD. Zhang *et al.* [13] additionally introduced the edge supervision for network training and constructed a multitask framework. Tu *et al.* [17] extracted edge features from the local cues and the global information, and embed the boundary features into region features. In addition to edge information, Li *et al.* [40] additionally introduced foreground, background, and the global image-level content to explore the complementarity of multiple content. Differently, Zhang *et al.* [41] solved this problem based on the weakly supervised learning. Li *et al.* [42] focused on cross-level feature fusion and inferred saliency map from a parallel down-up fusion network. Huang *et al.* [18] used the high-level features as a guide for locating multiscale objects and combined cross-level features and semantic information to refine the objects.

The above-mentioned existing methods have achieved high detection accuracy on the ORSSD and EORSSD datasets. However, these methods neglect the parameter and computational complexity of models, which prevents them from being deployed into practical systems. By contrast, in this article, we no longer focus on improving detection accuracy blindly, but open up a new direction for ORSI-SOD, that is, lightweight ORSI-SOD, which is to achieve a balance among accuracy, parameters, and computational complexity. To this end, we propose the first lightweight framework, namely CorrNet, for ORSI-SOD. In CorrNet, we implement

all components in a lightweight manner while maintaining competitive or even better performance.

III. Proposed Method

In this section, we elaborate the proposed CorrNet. In Section III-A, we depict the network overview of our CorrNet. In Section III-B, we show how to lighten the backbone. In Sections III-C and III-D, we elaborate the CorrM and the DLRB, respectively. In Section III-E, we formulate the loss function.

A. Network Overview

We present the overall framework of the proposed CorrNet in Fig. 2. CorrNet comprises three main components: a lightweight feature extraction subnet [equipped with the general feature enhancement module (FEM)], a CorrM, and a refinement subnet (equipped with the DLRB). It follows the coarse-to-fine strategy, that is, first generating a coarse saliency map and then sculpting it to generate a fine saliency map.

For feature extraction, we modify the classic vanilla VGG-16 [25] and construct a lightweight feature extraction subnet, named LFE-VGG. There are five convolution blocks in LFE-VGG. The first three convolution blocks are denoted as E^t ($t = 1, 2, 3$) and their output features as $\boldsymbol{f}_e^t \in \mathbb{R}^{c_t \times h_t \times w_t}$. The last two convolution blocks are denoted as DS-E^t ($t = 4, 5$), and their output features as $\boldsymbol{f}_{\mathrm{dse}}^t \in \mathbb{R}^{c_t \times h_t \times w_t}$. The size of input is $\mathbf{I} \in \mathbb{R}^{3\times 256\times 256}$, so h_t is $(256/2^{t-1})$, w_t is $(256/2^{t-1})$, and c_t belongs to $\{64, 128, 256, 512, 512\}$. Then, we apply the channel and spatial attentions[1] [43], [44] to $\boldsymbol{f}_e^1$ and $\boldsymbol{f}_e^2$ in the FEM, and get the enhanced features $\hat{\boldsymbol{f}}_e^1$ and $\hat{\boldsymbol{f}}_e^2$. To reduce parameters and computational complexity, we compress the channel of $\boldsymbol{f}_{\mathrm{dse}}^4$ and $\boldsymbol{f}_{\mathrm{dse}}^5$ (i.e., 512) to 128, and then upsample $\boldsymbol{f}_{\mathrm{dse}}^5$ to be the same size as $\boldsymbol{f}_{\mathrm{dse}}^4$. This way, we get $\hat{\boldsymbol{f}}_{\mathrm{dse}}^4$ and $\hat{\boldsymbol{f}}_{\mathrm{dse}}^5$, which belong to $\mathbb{R}^{\hat{c}_4\times h_4\times w_4}$ ($\hat{c}_4$ is 128).

Next, we model the feature correlation among $\hat{\boldsymbol{f}}_{\mathrm{dse}}^4$ and $\hat{\boldsymbol{f}}_{\mathrm{dse}}^5$ in the CorrM, aiming to mine the object location information of high-level semantic context. In this way, we get an initial coarse saliency map $\mathbf{S}^4$. As shown in Fig. 2, the coarse saliency map $\mathbf{S}^4$ can accurately locate the salient objects. It is used to modulate $\boldsymbol{f}_e^3$ to focus on the salient regions, generating the modulated features $\hat{\boldsymbol{f}}_e^3$. Finally, $\hat{\boldsymbol{f}}_e^1$, $\hat{\boldsymbol{f}}_e^2$ and $\hat{\boldsymbol{f}}_e^3$ are fed to the refinement subnet to generate the final fine saliency map $\mathbf{S}^1$ through three DLRBs.

B. Lightweight Feature Extraction Subnet

Previous ORSI-SOD methods [13], [16], [40] generally adapt the vanilla VGG-16 [25] for basic feature extraction, i.e., the last four layers (i.e., one max-pooling layer and three fully connected layers) are abandoned. However, the amount of parameters of the modified vanilla VGG is still large, which is not suitable as the backbone of a lightweight model. Therefore, in this article, we propose a convenient way to lighten the vanilla VGG without compromising its feature extraction ability.

[1]Channel attention is implemented by a spatial-wise global max pooling (GMP) and two fully connected layers (the first one is with ReLU and the second one is with sigmoid); and spatial attention is implemented by a channel-wise GMP and a one-channel regular convolution layer with sigmoid.

TABLE I

Parameters Comparison (Including the Parameters of Convolutional Layer and Batch Normalization Layer) of Vanilla-VGG and Our LFE-VGG

Block	Vanilla-VGG		LFE-VGG	
	#Param(M)	Ratio	#Param(M)	Ratio
E^1	0.04	0.27%	0.04	1.24%
E^2	0.22	1.50%	0.22	6.83%
E^3	1.48	10.05%	1.48	45.96%
E^4/DS-E^4	**5.90**	**40.08%**	**0.67**	**20.81%**
E^5/DS-E^5	**7.08**	**48.10%**	**0.81**	**25.16%**
Total	**14.72**	100.00%	**3.22**	100.00%

In Table I, we present the amount of parameters (including the parameters of convolution layer and batch normalization layer [45]) of each convolution block in the vanilla VGG. We observe that the last two convolution blocks of the vanilla VGG (i.e., E^4 and E^5) have 12.98M parameters, which account for about 88.18% of all parameters. Recently, MobileNets [27], [28] use the depthwise separable convolution (DSConv) to replace the regular convolution. The DSConv can significantly reduce the parameters without weakening the feature representation ability. Motivated by Howard *et al.* [27] and Sandler *et al.* [28], we adopt DSConvs instead of regular convolutions in E^4 and E^5, and get the redefined convolution blocks DS-E^4 and DS-E^5. There are two reasons why we only redefine E^4 and E^5. First, according to the above analysis, E^4 and E^5 occupy the largest amount of parameters. Second, we hope to use as many pretrained parameters of the vanilla VGG as possible to inherit powerful feature extraction ability.

In this way, we construct our lightweight feature extraction subnet, named LFE-VGG. As presented in Table I, the amount of parameters of E^4 is reduced from 5.90M to 0.67M, and that of E^5 is reduced from 7.08M to 0.81M. Overall, our LFE-VGG reduces 11.50M parameters compared to the vanilla VGG, and only has 3.22M parameters in total, which is a qualified lightweight backbone. We will assess the effectiveness and efficiency of LFE-VGG in Section IV-C.

C. Correlation Module

In video object segmentation, the target objects usually exist in consecutive video frames with small differences. To accurately segment the target objects, Lu *et al.* [46] employed the co-attention mechanism [47] to effectively mine the inherent correlation among consecutive video frames. Inspired by Lu *et al.* [46], considering that the salient regions also exist in consecutive features of an ORSI, we make an attempt to explore the cross-layer correlation of continuous high-level semantic features and propose the CorrM. Different from [46], which generates corresponding segmentation maps of different video frames, we focus on generating an initial coarse saliency map from the continuous semantic features for location guidance of low-level features.

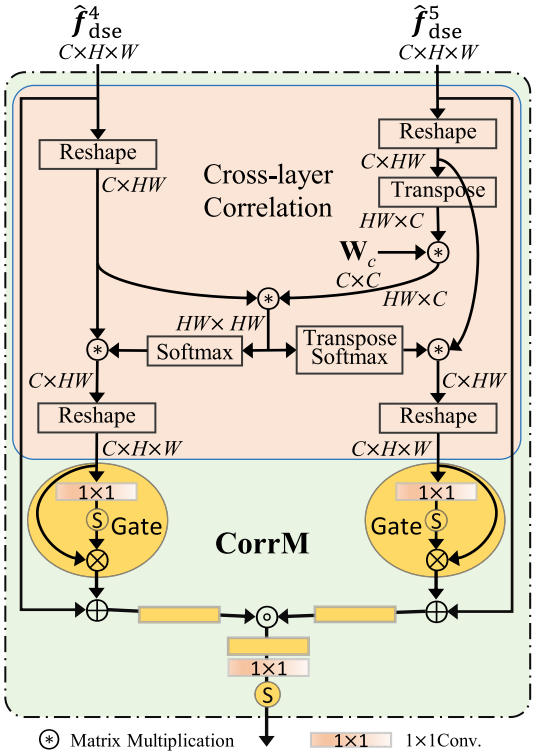


Fig. 3. Illustration of the CorrM.

We illustrate the CorrM in Fig. 3. It has three main components: cross-layer correlation operator, polishing gate, and initial coarse saliency map generation. In the following, we elaborate CorrM based on these three parts, and we also present the feature modulation process based on the coarse saliency map.

1) Cross-Layer Correlation: As shown in Fig. 3, we perform the cross-layer correlation operator on $\hat{f}^4_{\text{dse}}$ and $\hat{f}^5_{\text{dse}}$ (i.e., the continuous high-level semantic features), both belonging to $\mathbb{R}^{\hat{c}_4\times h_4\times w_4}$. Here, we simplify their sizes $\hat{c}_4 \times h_4 \times w_4$ as $C \times H \times W$ for notation conciseness.

First, we reshape $\hat{f}^4_{\text{dse}}$ to $\mathbb{R}^{C\times(HW)}$, and reshape and transpose $\hat{f}^5_{\text{dse}}$ to $\mathbb{R}^{(HW)\times C}$. Then, we define a learnable weight matrix $\mathbf{W}_c \in \mathbb{R}^{C\times C}$ for $\hat{f}^5_{\text{dse}}$, and construct a learning process for the cross-layer correlation operator, which makes our CorrM robust. Next, we compute the feature correlation via matrix multiplication to capture similarity between each row of the reshaped and transposed $\hat{f}^5_{\text{dse}}$ and each column of the reshaped $\hat{f}^4_{\text{dse}}$. We formulate the above process as follows:

$$\boldsymbol{r} = \left(\text{rshp}\left(\hat{f}^5_{\text{dse}}\right)\right)^{\top} \circledast \mathbf{W}_c \circledast \text{rshp}\left(\hat{f}^4_{\text{dse}}\right) \tag{1}$$

where $\boldsymbol{r} \in \mathbb{R}^{(HW)\times(HW)}$ is the cross-layer correlation matrix, rshp$(\cdot)$ is the reshape operation, $\top$ is the matrix transpose operation, and $\circledast$ is the matrix multiplication.

After obtaining the cross-layer correlation matrix $\boldsymbol{r}$, we use the softmax function to normalize it along the rows and columns, respectively, and exploit it to determine the location of salient regions of high-level semantic features, which can be formulated as follows:

$$f^4_{\text{corr}} = \text{rshp}\left(\text{rshp}\left(\hat{f}^4_{\text{dse}}\right) \circledast \text{softmax}(\boldsymbol{r})\right) \tag{2}$$

$$f^5_{\text{corr}} = \text{rshp}\left(\text{rshp}\left(\hat{f}^5_{\text{dse}}\right) \circledast \text{softmax}(\boldsymbol{r}^{\top})\right) \tag{3}$$

where $\{f^4_{\text{corr}}, f^5_{\text{corr}}\} \in \mathbb{R}^{C\times H\times W}$ are features containing rich location information.

Since we perform the matrix-based cross-layer correlation operator on $\hat{f}^4_{\text{dse}}$ and $\hat{f}^5_{\text{dse}}$, whose sizes are $128 \times 32 \times 32$, its computational cost is limited. Besides, only the parameters of the weight matrix $\mathbf{W}_c$ need to be learned. Therefore, the cross-layer correlation operator is with few parameters and low computational cost, but has strong capabilities to locate salient objects in ORSIs.

2) Polishing Gate: The above cross-layer correlation operator may leave some redundant information in f^4_{corr} and f^5_{corr}. To address this issue, we introduce a simple but effective gate mechanism to polish f^4_{corr} and f^5_{corr}, and achieve more pure location information.

In order to reduce the module parameters, here, we adopt the regular 1×1 convolution layer to separately produce a response map (which belongs to $[0, 1]^{1\times H\times W}$) for f^4_{corr} and f^5_{corr}. Based on the two response maps, we filter the redundant information of f^4_{corr} and f^5_{corr}. We formulate the above gate mechanism as follows:

$$\begin{aligned} f^4_{\text{gate}} &= \text{sigmoid}\left(\text{conv}_{1\times1}\left(f^4_{\text{corr}}\right)\right) \otimes f^4_{\text{corr}} \\ f^5_{\text{gate}} &= \text{sigmoid}\left(\text{conv}_{1\times1}\left(f^5_{\text{corr}}\right)\right) \otimes f^5_{\text{corr}} \end{aligned} \tag{4}$$

where $\{f^4_{\text{gate}}, f^5_{\text{gate}}\} \in \mathbb{R}^{C\times H\times W}$ are the polished features, $\text{conv}_{1\times1}(\cdot)$ is the regular 1×1 convolution operator, and $\otimes$ is the element-wise multiplication. Moreover, we adopt the residual connection to merge f^4_{gate} and $\hat{f}^4_{\text{dse}}$, and f^5_{gate} and $\hat{f}^5_{\text{dse}}$, respectively, producing $\hat{f}^4_{\text{gate}}$ and $\hat{f}^5_{\text{gate}}$ as follows:

$$\begin{aligned} \hat{f}^4_{\text{gate}} &= \text{DSconv}\left(f^4_{\text{gate}} \oplus \hat{f}^4_{\text{dse}}\right) \\ \hat{f}^5_{\text{gate}} &= \text{DSconv}\left(f^5_{\text{gate}} \oplus \hat{f}^5_{\text{dse}}\right) \end{aligned} \tag{5}$$

where DSconv$(\cdot)$ is the 3×3 DSConv and $\oplus$ is the element-wise summation. This original content preservation mode (i.e., the residual connection) is good for feature representation.

3) Initial Coarse Saliency Map Generation: Thanks to the above two effective parts of CorrM, the generated $\hat{f}^4_{\text{gate}}$ and $\hat{f}^5_{\text{gate}}$ are very informative. Based on them, we introduce the last part of our CorrM as follows:

$$\mathbf{S}^4 = \text{sigmoid}\left(\text{conv}_{1\times1}\left(\text{DSconv}\left(\hat{f}^4_{\text{gate}} \odot \hat{f}^5_{\text{gate}}\right)\right)\right) \tag{6}$$

where $\mathbf{S}^4 \in [0, 1]^{1\times32\times32}$ is the initial coarse saliency map and $\odot$ is the concatenation. In this way, we completely extract the location information of $\hat{f}^4_{\text{gate}}$ and $\hat{f}^5_{\text{gate}}$, accurately determining the salient regions in ORSIs, as $\mathbf{S}^4$ shown in Fig. 2.

4) Feature Modulation: Furthermore, we migrate location information to the basic features f^3_e as follows:

$$\hat{f}^3_e = \text{Up}\left(\mathbf{S}^4\right) \otimes f^3_e \oplus f^3_e \tag{7}$$

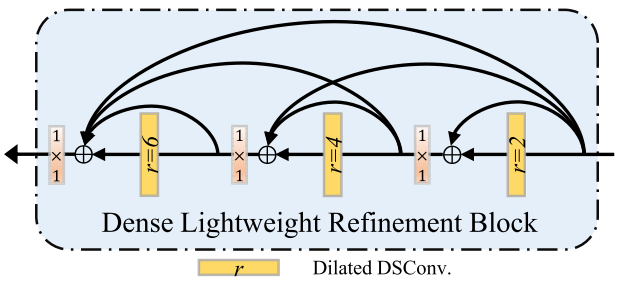


Fig. 4. Illustration of the DLRB.

where $\hat{\boldsymbol{f}}_e^3 \in \mathbb{R}^{c_3 \times h_3 \times w_3}$ is the modulated features and $\mathrm{Up}(\cdot)$ is the upsampling operation. This direct feature modulation mode provides accurate location information for the subsequent object refinement process, laying a solid foundation for the final fine saliency map.

In summary, our CorrM balances effectiveness and efficiency, that is, predicting a momentous coarse saliency map with few parameters. We will evaluate the importance of our CorrM in Section IV-C.

D. Dense Lightweight Refinement Block

The widely used refinement block usually follows a cascaded structure, i.e., several regular convolution layers are connected one by one. However, there are some challenging scenarios in ORSIs, such as multiple objects and small objects. The cascaded structure is not conducive to capturing multiscale information and is a suboptimal way for objects refinement in ORSIs. Besides, the cascaded refinement block is usually implemented by regular convolution layers, bringing lots of parameters. Inspired by DenseNet [48] and DSConv [27], [28], we implement a refinement block with the dense structure and DSConvs, constructing a DLRB for objects refinement in ORSIs, as shown in Fig. 4.

For each DLRB, there are three dilated DSConvs with progressive dilation rates {2,4,6} and three 1×1 convolution layers. Dilated DSConvs enlarge the receptive field, capturing multiscale features comprehensively. And 1×1 convolution layers are in charge of merging the captured features. The output feature of DLRB-t is denoted as $\boldsymbol{f}_{\mathrm{dlrb}}^t$. Here, we take DLRB-3 as an example. In DLRB-3, its input is $\hat{\boldsymbol{f}}_e^3$, and we decompose its dense structure into three stages, which are formulated as follows:

$$\boldsymbol{f}_{\mathrm{dlrb}}^{3,1} = \mathrm{conv}_{1\times1}\left(\mathrm{DSconv}_2\left(\hat{\boldsymbol{f}}_e^3\right) \oplus \hat{\boldsymbol{f}}_e^3\right) \tag{8}$$

$$\boldsymbol{f}_{\mathrm{dlrb}}^{3,2} = \mathrm{conv}_{1\times1}\left(\mathrm{DSconv}_4\left(\boldsymbol{f}_{\mathrm{dlrb}}^{3,1}\right) \oplus \boldsymbol{f}_{\mathrm{dlrb}}^{3,1} \oplus \hat{\boldsymbol{f}}_e^3\right) \tag{9}$$

$$\boldsymbol{f}_{\mathrm{dlrb}}^{3,3} = \mathrm{conv}_{1\times1}\left(\mathrm{DSconv}_6\left(\boldsymbol{f}_{\mathrm{dlrb}}^{3,2}\right) \oplus \boldsymbol{f}_{\mathrm{dlrb}}^{3,2} \oplus \boldsymbol{f}_{\mathrm{dlrb}}^{3,1} \oplus \hat{\boldsymbol{f}}_e^3\right) \tag{10}$$

where $\mathrm{DSconv}_r(\cdot)$ is the dilated 3×3 DSConv with dilation rate r, and $\boldsymbol{f}_{\mathrm{dlrb}}^{3,3}$ is the output feature of DLRB-3, i.e., $\boldsymbol{f}_{\mathrm{dlrb}}^3$.

In this way, our DLRB can perceive multiscale information and bring powerful feature representation during the refinement phase, which will facilitate the carving of salient objects in ORSIs and lead to good performance. We will evaluate the effectiveness of our DLRB in Section IV-C.

E. Loss Function

To effectively train CorrNet, we combine the classic BCE loss and IoU loss to construct a comprehensive loss function for network training, which is the same as previous SOD methods [40], [61], [62]. Moreover, we also adopt the deep supervision [54], [63] in the training phase to supervise two intermediate saliency maps of the refinement subnet as well as the coarse and fine saliency maps, as shown in Fig. 2. The intermediate saliency maps and fine saliency map are generated by the 1×1 convolution layer. We formulate the total loss function L_{total} as

$$L_{\mathrm{total}} = \sum_{t=1}^{4}\left(\ell_{\mathrm{bce}}\left(\mathrm{Up}\left(\mathbf{S}^t\right), \mathbf{G}\right) + \ell_{\mathrm{iou}}\left(\mathrm{Up}\left(\mathbf{S}^t\right), \mathbf{G}\right)\right) \tag{11}$$

where $\mathbf{G}$ is the ground truth, and $\ell_{\mathrm{bce}}(\cdot)$ and $\ell_{\mathrm{iou}}(\cdot)$ are BCE loss and IoU loss, respectively.

IV. Experiments

A. Implementation Details and Evaluation Metrics

1) Implementation Details: We train and test CorrNet on the ORSSD and EORSSD datasets, respectively. There are 800 ORSIs with corresponding ground truths in the ORSSD dataset [12], in which 600 images are used for training and 200 images for testing. And there are 2000 ORSIs with corresponding ground truths in the EORSSD dataset [13], in which 1400 images are used for training and 600 images for testing. We adopt the flipping and rotation for data augmentation, generating 4800 training pairs for ORSSD and 11 200 training pairs for EORSSD. All the experiments are conducted on the PyTorch [64] platform with an NVIDIA Titan X GPU (12-GB memory). In the training phase, we resized the training pairs to 256×256, and adopt the Adam optimization strategy [65] with the batch size 8 and the initial learning rate $1e^{-4}$, which will be divided by 10 after 30 epochs. We initialize the first three blocks of LFE-VGG by the pretrained VGG-16 model [25], and initialize other newly added DSConvs and 1×1 convolution layers by the normal distribution [66]. Notably, on the ORSSD dataset, we train our CorrNet for 44 epochs; and on the EORSSD dataset, we train our CorrNet for 34 epochs.

2) Evaluation Metrics: We employ five quantitative evaluation metrics to evaluate our method and other compared methods, including S-measure (S_α, $\alpha = 0.5$) [67], (maximum, mean, and adaptive) F-measure (F_β, $\beta^2 = 0.3$) [68], (maximum, mean, and adaptive) E-measure (E_ξ) [69], mean absolute error (MAE, $\mathcal{M}$), and precision-recall (PR) curve. The first three evaluation metrics are the bigger the better. MAE is the smaller the better. And PR curve is closer to the upper right, the better.

B. Comparison With State-of-the-Art Methods

We conduct a comprehensive comparison with 26 state-of-the-art NSI-SOD and ORSI-SOD methods, including eight

TABLE II

QUANTITATIVE COMPARISONS WITH 26 STATE-OF-THE-ART METHODS, INCLUDING FIVE TRADITIONAL NSI-SOD METHODS, THREE TRADITIONAL ORSI-SOD METHODS, ELEVEN CNN-BASED NSI-SOD METHODS, FIVE CNN-BASED ORSI-SOD METHODS, AND TWO LIGHTWEIGHT METHODS, ON EORSSD AND ORSSD DATASETS. THE TOP THREE RESULTS ARE HIGHLIGHTED IN **RED**, **BLUE**, AND **GREEN**, RESPECTIVELY

Methods	Type	Input size	Speed (fps)↑	#Param (M)↓	FLOPs (G)↓	EORSSD [13]								ORSSD [12]							
						$S_\alpha\uparrow$	$F_\beta^{\max}\uparrow$	$F_\beta^{\text{mean}}\uparrow$	$F_\beta^{\text{adp}}\uparrow$	$E_\xi^{\max}\uparrow$	$E_\xi^{\text{mean}}\uparrow$	$E_\xi^{\text{adp}}\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta^{\max}\uparrow$	$F_\beta^{\text{mean}}\uparrow$	$F_\beta^{\text{adp}}\uparrow$	$E_\xi^{\max}\uparrow$	$E_\xi^{\text{mean}}\uparrow$	$E_\xi^{\text{adp}}\uparrow$	$\mathcal{M}\downarrow$
$RRWR_{15}$ [49]	T.N.	-	0.3	-	-	.5992	.3993	.3686	.3344	.6894	.5943	.5639	.1677	.6835	.5590	.5125	.4874	.7649	.7017	.6949	.1324
$HDCT_{16}$ [50]	T.N.	-	7	-	-	.5971	.5407	.4018	.2658	.7861	.6376	.5192	.1088	.6197	.5257	.4235	.3722	.7719	.6495	.6291	.1309
DSG_{17} [51]	T.N.	-	0.6	-	-	.6420	.5232	.4597	.4012	.7260	.6594	.6188	.1246	.7195	.6238	.5747	.5657	.7912	.7337	.7532	.1041
SMD_{17} [52]	T.N.	-	-	-	-	.7101	.5884	.5473	.4081	.7697	.7286	.6416	.0771	.7640	.6692	.6214	.5568	.8230	.7745	.7682	.0715
$RCRR_{18}$ [53]	T.N.	-	0.3	-	-	.6007	.3995	.3685	.3347	.6882	.5946	.5636	.1644	.6849	.5591	.5126	.4876	.7651	.7021	.6950	.1277
VOS_{18} [38]	T.R.	-	-	-	-	.5082	.2765	.2107	.1836	.5982	.4886	.4767	.2096	.5366	.3471	.2717	.2633	.6514	.5352	.5826	.2151
$SMFF_{19}$ [34]	T.R.	-	-	-	-	.5401	.5176	.2992	.2083	.7744	.5197	.5014	.1434	.5312	.4417	.2684	.2496	.7402	.4920	.5676	.1854
CMC_{19} [37]	T.R.	-	-	-	-	.5798	.3268	.2692	.2007	.6803	5894	.4890	.1057	.6033	.3913	.3454	.3108	.7064	.6417	.5996	.1267
DSS_{17} [54]	C.N.	400×300	8	62.23	114.6	.7868	.6849	.5801	.4597	.9186	.7631	.6933	.0186	.8262	.7467	.6962	.6206	.8860	.8362	.8085	.0363
$RADF_{18}$ [55]	C.N.	400×400	7	62.54	214.2	.8179	.7446	.6582	.4933	.9140	.8567	.7162	.0168	.8259	.7619	.6856	.5730	.9130	.8298	.7678	.0382
$R3Net_{18}$ [56]	C.N.	300×300	2	56.16	47.5	.8184	.7498	.6302	.4165	.9483	.8294	.6462	.0171	.8141	.7456	.7383	.7379	.8913	.8681	.8887	.0399
$PoolNet_{19}$ [57]	C.N.	400×300	25	53.63	123.4	.8207	.7545	.6406	.4611	.9292	.8193	.6836	.0210	.8403	.7706	.6999	.6166	.9343	.8650	.8124	.0358
$EGNet_{19}$ [22]	C.N.	∼380×320	9	108.07	291.9	.8601	.7880	.6967	.5379	.9570	.8775	.7566	.0110	.8721	.8332	.7500	.6452	.9731	.9013	.8226	.0216
$GCPA_{20}$ [58]	C.N.	320×320	23	67.06	54.3	.8869	.8347	.7905	.6723	.9524	.9167	.8647	.0102	.9026	.8687	.8433	.7861	.9509	.9341	.9205	.0168
$MINet_{20}$ [59]	C.N.	320×320	12	47.56	146.3	.9040	.8344	.8174	.7705	.9442	.9346	.9243	.0093	.9040	.8761	.8574	.8251	.9545	.9454	.9423	.0144
$ITSD_{20}$ [60]	C.N.	288×288	16	17.08	54.5	.9050	.8523	.8221	.7421	.9556	.9407	.9103	.0106	.9050	.8735	.8502	.8068	.9601	.9482	.9335	.0165
$GateNet_{20}$ [21]	C.N.	384×384	25	100.02	108.3	.9114	.8566	.8228	.7109	.9610	.9385	.8909	.0095	.9186	.8871	.8679	.8229	.9664	.9538	.9428	.0137
$SUCA_{21}$ [20]	C.N.	256×256	24	117.71	56.4	.8988	.8229	.7949	.7260	.9520	.9277	.9082	.0097	.8989	.8484	.8237	.7748	.9584	.9400	.9194	.0145
$PA\text{-}KRN_{21}$ [19]	C.N.	600×600	16	141.06	617.7	.9192	.8639	.8358	**.7993**	.9616	.9536	.9416	.0104	**.9239**	.8890	**.8727**	**.8548**	.9680	**.9620**	**.9579**	.0139
$LVNet_{19}$ [12]	C.R.	128×128	1.4	-	-	.8630	.7794	.7328	.6284	.9254	.8801	.8445	.0146	.8815	.8263	.7995	.7506	.9456	.9259	.9195	.0207
$DAFNet_{21}$ [13]	C.R.	128×128	26	29.35	68.51	.9166	.8614	.7845	.6427	**.9861**	.9291	.8446	**.0060**	.9191	**.8928**	.8511	.7876	**.9771**	.9539	.9360	**.0113**
$MJRBM_{21}$ [17]	C.R.	352×352	32	43.54	95.7	.9197	.8656	.8239	.7066	.9646	.9350	.8897	.0099	.9204	.8842	.8566	.8022	.9623	.9415	.9328	.0163
$SARNet_{21}$ [18]	C.R.	336×336	47	25.91	129.7	**.9240**	**.8719**	**.8541**	**.8304**	.9620	**.9555**	**.9536**	.0099	.9134	.8850	.8619	.8512	.9557	.9477	.9464	.0187
$EMFINet_{21}$ [16]	C.R.	256×256	25	107.26	480.9	**.9290**	**.8720**	**.8486**	.7984	**.9711**	**.9604**	**.9501**	**.0084**	**.9366**	**.9002**	**.8856**	**.8617**	**.9737**	**.9671**	**.9663**	**.0109**
$CSNet_{20}$ [23]	L.W.	224×224	38	0.14	0.7	.8364	.8341	.7656	.6319	.9535	.8929	.8339	.0169	.8910	.8790	.8285	.7615	.9628	.9171	.9068	.0186
$SAMNet_{21}$ [24]	L.W.	336×336	44	1.33	0.5	.8622	.7813	.7214	.6114	.9421	.8700	.8284	.0132	.8761	.8137	.7531	.6843	.9478	.8818	.8656	.0217
Ours	L.W.	256×256	100	4.09	21.09	**.9289**	**.8778**	**.8620**	**.8311**	**.9696**	**.9646**	**.9593**	**.0083**	**.9380**	**.9129**	**.9002**	**.8875**	**.9790**	**.9746**	**.9721**	**.0098**

T.N.: Traditional NSI-SOD method, T.R.: Traditional ORSI-SOD method, C.N.: CNN-based NSI-SOD method, C.R.: CNN-based ORSI-SOD method, L.W.: Lightweight method.

traditional methods (RRWR [49], HDCT [50], DSG [51], SMD [52], RCRR [53], VOS [38], CMC [37], and SMFF [34]), 16 CNN-based methods (DSS [54], RADF [55], R3Net [56], PoolNet [57], EGNet [22], GCPA [58], MINet [59], ITSD [60], GateNet [21], SUCA [20], PA-KRN [19], LVNet [12], DAFNet [13], MJRBM [17], SARNet [18], and EMFINet [16]), and two lightweight methods (CSNet [23] and SAMNet [24]). Since some methods have different backbone versions, we only report their performance based on VGG backbone. For a fair comparison, we retain CNN-based NSI-SOD methods with their default parameter settings on the same training set as our method. And the saliency maps of other methods are provided by the authors or generated by public source codes.

1) Computational Complexity Comparison: In Table II, we report the inference speed with batch size of 1 (without I/O time), parameter amount (#Param), and FLOPs of our method and most compared methods. Notably, our method achieves competitive performance in these three computational complexity metrics. Our method (i.e., 100 fps) has 2.1× faster inference speed than the second-placed method SARNet (i.e., 47 fps). Compared with CNN-based methods, the parameter amount and FLOPs of our method are smaller than them. While compared with two lightweight methods, i.e., CSNet and SAMNet, our method is slightly inferior, but it is still good as the first lightweight ORSI-SOD solution. Therefore, we believe that our method is an efficient and promising lightweight ORSI-SOD framework.

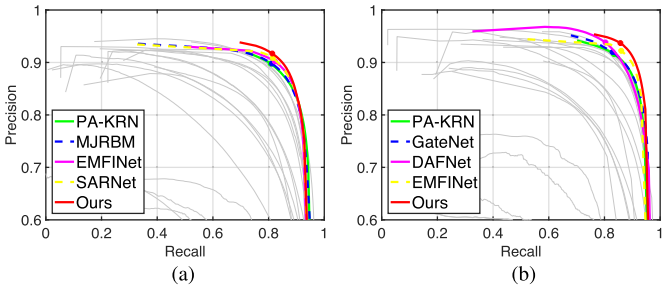


Fig. 5. Quantitative comparison in terms of PR curve on two datasets. The top five methods are shown in different colors, while the other compared methods are shown in gray. Zoom-in for better visualization of details. (a) EORSSD [13]. (b) ORSSD [12].

2) Quantitative Comparison: In Fig. 5, we plot the PR curves of our method and all compared methods on EORSSD and ORSSD. As visible, our method (i.e., the red one) is closest to the upper right than other methods on both datasets, showing a competitive performance.

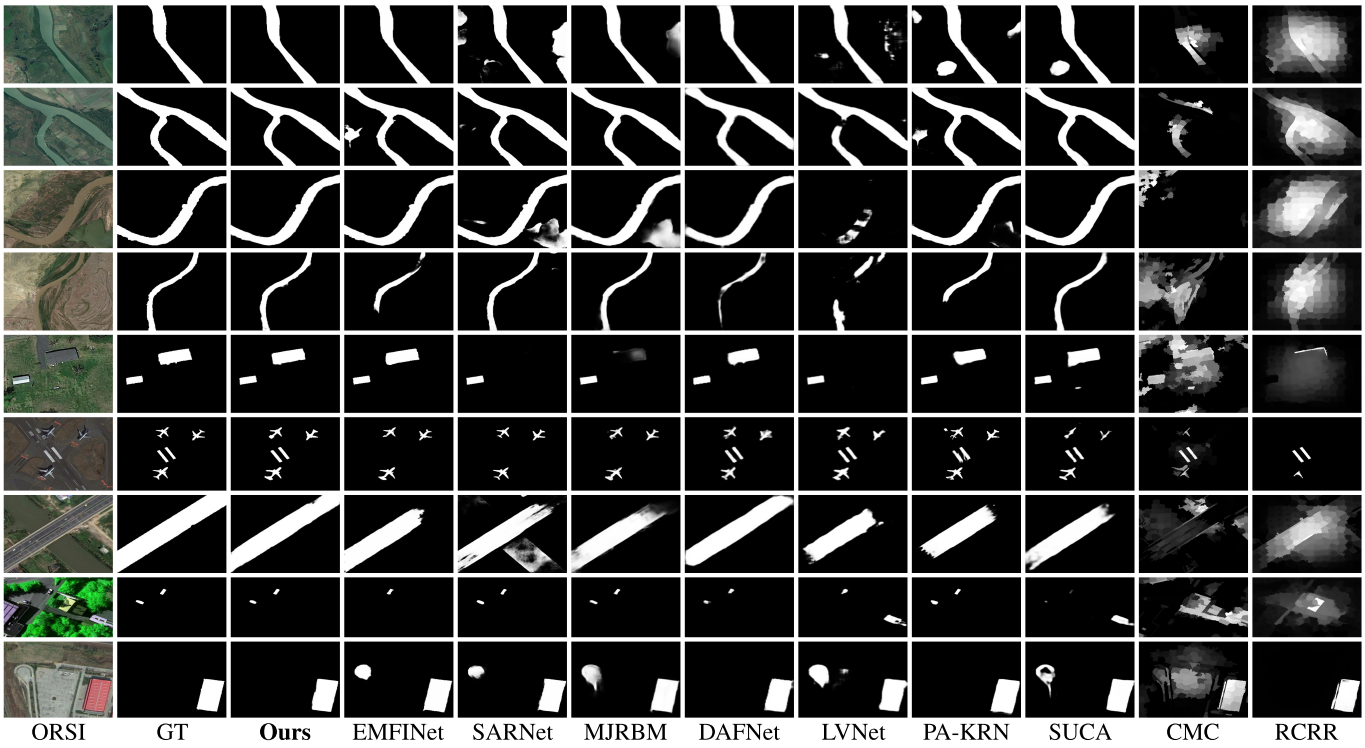


Fig. 6. Visual comparisons with nine representative state-of-the-art methods. Please zoom-in for the best view.

We report the quantitative performance of our method and all compared methods on EORSSD and ORSSD in Table II. On the EORSSD dataset, our method ranks first in five out of all eight metrics. Compared with EMFINet with similar performance, our method has 4× faster inference speed, 26× fewer parameters, and 23× fewer FLOPs than it. On the ORSSD dataset, our method outperforms all compared methods in all eight metrics. Specifically, in $F_\beta^{\max}$, F_β^{mean} and F_β^{apt}, our method is 1.27%, 1.46%, and 2.58% higher than the second-placed method EMFINet, respectively. Compared with the lightweight methods, i.e., CSNet and SAMNet, the performance of our method is significantly better than them. Moreover, we observe that the performance of CNN-based ORSI-SOD methods is generally better than that of the retrained CNN-based NSI-SOD methods, which indicates that the ORSI scenes are extremely challenging. Overall, the above analysis clearly demonstrates that our lightweight CorrNet achieves a favorable tradeoff between effectiveness and efficiency.

3) Visual Comparison: We present the visual comparison of our method and nine representative state-of-the-art methods on some challenging ORSI scenes in Fig. 6. As the first four rows of Fig. 6, the first scene is of low contrast. In these four cases, only our method can clearly highlight all salient objects. Other compared methods are interfered by similar backgrounds and fail in individual cases, such as EMFINet fails in the second case and MJRBM fails in the first and third cases. As the fifth and sixth rows of Fig. 6, the second scene is multiple objects, which is a difficult scene in ORSI-SOD. Our method can accurately locate all salient objects with fine details, while some models occasionally miss objects (such as SARNet and LVNet) or fail to outline details (such as DAFNet and SUCA). As the seventh row of Fig. 6, the third scene is the big object. In this scene, due to the large span of the bridge, most methods only segment its middle part and ignore its two ends. As the last two rows of Fig. 6, the fourth scene is cluttered background. The cluttered background confuses some methods, causing them to incorrectly include background or miss objects in their saliency maps. Overall, our method shows strong scene adaptability and overcomes the above scenes.

C. Ablation Studies

Here, we conduct comprehensive experiments to evaluate the effectiveness of important components of our CorrNet on EORSSD dataset. In particular, we investigate: 1) the efficiency of the lightweight feature extraction subnet; 2) the effectiveness of the coarse-to-fine strategy; 3) the individual contribution of each module in CorrNet; 4) the importance of cross-layer correlation and polishing gate in CorrM; and 5) the rationality of dilated DSConvs' dilation rates in DLRB. For each variant experiment, we rigorously retrain it with the same parameter settings and datasets as in Section IV-A.

1) Efficiency of the Lightweight Feature Extraction Subnet: To evaluate the efficiency of the lightweight feature extraction subnet (i.e., LFE-VGG), we replace it with two backbones, i.e., Vanilla-VGG and DS-VGG. Vanilla-VGG is all convolution layers of VGG-16 are regular convolution layers, and DS-VGG is all convolution layers of VGG-16 are DSConvs. We report the quantitative results in Table III.

TABLE III

ABLATION RESULTS OF EVALUATING THE EFFICIENCY OF THE LIGHTWEIGHT FEATURE EXTRACTION SUBNET

Models	#Param (M)↓	FLOPs (G)↓	EORSSD [13]			
			$S_\alpha \uparrow$	$F_\beta^{\max} \uparrow$	$E_\xi^{\max} \uparrow$	$\mathcal{M} \downarrow$
Vanilla-VGG	15.59	28.1	.9292	.8789	.9718	.0083
DS-VGG	2.55	10.4	.9063	.8447	.9512	.0108
LFE-VGG (**Ours**)	4.09	21.1	.9289	.8778	.9696	.0083

Vanilla-VGG: VGG-16 with regular convs.

DS-VGG: VGG-16 with DSConvs.

TABLE IV

ABLATION RESULTS OF EVALUATING THE EFFECTIVENESS OF THE COARSE-TO-FINE STRATEGY. THE BEST ONE IN EACH COLUMN IS BOLD

Models	EORSSD [13]			
	$S_\alpha \uparrow$	$F_\beta^{\max} \uparrow$	$E_\xi^{\max} \uparrow$	$\mathcal{M} \downarrow$
$\mathbf{S}^4$	.8741	.7800	.9578	.0126
$\mathbf{S}^3$	.9138	.8560	.9667	.0094
$\mathbf{S}^2$	.9265	.8755	**.9698**	.0083
$\mathbf{S}^1$ (**Ours**)	**.9289**	**.8778**	.9696	**.0083**

TABLE V

ABLATION RESULTS OF EVALUATING THE INDIVIDUAL CONTRIBUTION OF EACH MODULE IN CORRNET. THE BEST ONE IN EACH COLUMN IS BOLD

No.	Baseline	FEM	DLRB	CorrM	EORSSD [13]			
					$S_\alpha \uparrow$	$F_\beta^{\max} \uparrow$	$E_\xi^{\max} \uparrow$	$\mathcal{M} \downarrow$
1	✓				.9146	.8548	.9535	.0113
2	✓	✓			.9160	.8591	.9541	.0106
3	✓	✓	✓		.9231	.8679	.9607	.0086
4	✓	✓		✓	.9232	.8718	.9639	.0086
5	✓	✓	✓	✓	**.9289**	**.8778**	**.9696**	**.0083**

The complete Vanilla-VGG does improve performance, but the improvement is limited, e.g., $F_\beta^{\max}$ is increased by 0.11% and $E_\xi^{\max}$ is increased by 0.22%. However, along with improved performance comes a massive increase in parameters, e.g., #Param increases sharply from 4.09M to 15.59M. This means that our LFE-VGG is effective and efficient, and our modification is reasonable. As for DS-VGG, its performance is obviously degraded, e.g., $F_\beta^{\max}$ is reduced by 3.31% and $E_\xi^{\max}$ is reduced by 1.84%, and its parameter reduction is also limited, e.g., #Param is reduced by 1.54M. We think that the reason for the performance degradation of DS-VGG is because its parameters are initialized by the normal distribution [66], losing the benefits of pretrained parameters. Overall, our LFE-VGG maintains the powerful feature extraction ability of E^1, E^2 and E^3, and greatly reduces the parameters of DS-E^4 and DS-E^5, so it is a qualified lightweight backbone.

TABLE VI

ABLATION RESULTS OF EVALUATING THE IMPORTANCE OF CROSS-LAYER CORRELATION AND POLISHING GATE IN CORRM AND THE RATIONALITY OF DILATED DSCONVS IN DLRB. THE BEST ONE IN EACH COLUMN IS BOLD

Models	EORSSD [13]			
	$S_\alpha \uparrow$	$F_\beta^{\max} \uparrow$	$E_\xi^{\max} \uparrow$	$\mathcal{M} \downarrow$
CorrNet (**Ours**)	**.9289**	**.8778**	**.9696**	.0083
w/o Correlation	.9232	.8690	.9649	.0085
w/o Gate	.9253	.8728	.9661	.0086
w/o dilation rate	.9254	.8718	.9663	.0089
w/ 1-3-5	.9262	.8727	.9662	.0087
w/ 3-5-7	.9272	.8743	.9666	**.0077**

2) Effectiveness of the Coarse-to-Fine Strategy: To evaluate the effectiveness of the coarse-to-fine strategy, we quantitatively measure the performance of the initial coarse saliency map ($\mathbf{S}^4$), two intermediate saliency maps ($\mathbf{S}^3$ and $\mathbf{S}^2$) and the final fine saliency map ($\mathbf{S}^1$). As reported in Table IV, the quantitative performance is generally incremental, e.g., $F_\beta^{\max}$: 78.00% $\rightarrow$ 85.60% $\rightarrow$ 87.55% $\rightarrow$ 87.78%. And compared with $\mathbf{S}^4$, the improvement of $\mathbf{S}^1$ is greatly significant in all metrics, i.e., S_α, $F_\beta^{\max}$, $E_\xi^{\max}$ and $\mathcal{M}$ are improved by 5.48%, 9.78%, 1.18%, and 0.0043, respectively. This confirms that the coarse-to-fine manner is useful in our CorrM, and the refinement subnet demonstrates powerful refinement capabilities.

3) Individual Contribution of Each Module in CorrNet: To evaluate the individual contribution of each module, i.e., FEM, DLRB, and CorrM, we design four variants of the full CorrNet (i.e., No.5) in Table V: 1) Baseline; 2) Baseline + FEM; 3) Baseline + FEM + DLRB; and 4) Baseline + FEM + CorrM. For Baseline, we directly remove FEMs, replace CorrM with concatenation-convolution operation to generate the coarse saliency map, and replace DLRB with three cascaded regular DSConvs.

According to the quantitative performance in Table V, we observe that each module of CorrNet contributes to the ultimate excellent performance. Our full CorrNet boosts the primitive Baseline by 1.43%, 2.30%, 1.61%, and 0.0030 on S_α, $F_\beta^{\max}$, $E_\xi^{\max}$, and $\mathcal{M}$, respectively. As the key roles of CorrNet, DLRB improves $F_\beta^{\max}$ and $E_\xi^{\max}$ of Baseline + FEM by 0.88% and 0.66%, respectively, and CorrM improves $F_\beta^{\max}$ and $E_\xi^{\max}$ of Baseline + FEM by 1.27% and 0.98%, respectively. With the cooperation of DLRB and CorrM, the full CorrNet improves $F_\beta^{\max}$ and $E_\xi^{\max}$ of Baseline + FEM by 1.87% and 1.55%, respectively. Therefore, the above analysis verifies that each module in CorrNet is effective for ORSI-SOD.

4) Importance of Cross-Layer Correlation and Polishing Gate in CorrM: To evaluate the importance of cross-layer correlation and polishing gate in CorrM, we provide two variants in Table VI: 1) deleting cross-layer correlation operator in CorrM, namely w/o Correlation and 2) deleting two polishing gates in CorrM, namely w/o Gate. For w/o Correlation, there is no cross-layer correlation operator to capture the feature correlation among the high-level semantic context, the object

localization capabilities of CorrM are greatly weakened, and S_α and $F_\beta^{\max}$ drop to 92.32% and 86.90%, respectively. w/o Gate can capture the object location information with some redundant information, and obtain slightly better results than w/o Correlation, i.e., 92.53% of S_α and 87.28% of $F_\beta^{\max}$. The above variants verify that the cross-layer correlation and polishing gate are important to CorrM.

5) Rationality of Dilated DSConvs' Dilation Rates in DLRB: To evaluate the rationality of dilated DSConvs' dilation rates in DLRB, we provide three variants in Table VI: 1) replacing three dilated DSConvs with three regular DSConvs in DLRB, namely w/o dilation rate; 2) changing the original dilation rates {2,4,6} to {1,3,5}, namely w/ 1-3-5; and 3) changing the original dilation rates {2,4,6} to {3,5,7}, namely w/ 3-5-7. Based on the quantitative performance of w/o dilation rate and w/ 1-3-5, we observe that the relatively large receptive fields can capture more complementary multiscale information, which is very important for objects refinement in ORSI-SOD. However, when we continue to expand the receptive fields to {3,5,7}, we observe the performance degradation of w/ 3-5-7, possibly due to that excessively large receptive fields make DLRB impossible to accurately capture the salient objects with variable scales in ORSIs. Thus, we can come to a conclusion that the dilated DSConvs with dilation rates {2,4,6} are rational and exactly appropriate in DLRB.

D. Discussion

Here, we discuss the weaknesses of our method and our future works. For weaknesses, we summarize as follows: 1) since our method is based on GPU, its model size is still too large for edge computing devices; and 2) although the parameter amount and computations of our method are small as compared with most CNN-based methods, it is still difficult to run on the CPU in real time.

Therefore, in future works, we will work in the following two directions: 1) similar to the lightweight ORSI-SOD methods, we will focus on developing a lightweight backbone with smaller model size for ORSI-SOD; and 2) we will introduce the model pruning technology into our model to remove redundant layers and to further accelerate our model.

V. Conclusion

In this article, we propose an effective lightweight framework, namely CorrNet, for ORSI-SOD. In CorrNet, we first lighten the vanilla VGG and propose a lightweight feature extraction subnet, namely LFE-VGG. Then, we lighten other modules of CorrNet, that is, we use DSConvs instead of regular convolution layers in the modules. In addition, in order to obtain a good performance, CorrNet follows the coarse-to-fine strategy. It first generates an initial coarse saliency map from high-level semantic features via the CorrM, and then refines the salient objects via the refinement subnet equipped with DLRBs, producing the final fine saliency map. Experimental evaluations on two ORSI-SOD datasets demonstrate that though our CorrNet only has 4.09M parameters and 21.09G FLOPs, it achieves competitive or even better performance than large CNN-based methods and runs at 100 fps. The success of our CorrNet comes from three aspects: 1) the matrix-based cross-layer correlation operation that extracts salient regions effectively and only contains a few parameters; 2) the DSConv that maintains powerful feature representation capabilities and only has about 10% of the parameters of the regular convolution layer; and 3) the coarse-to-fine strategy that lays a high-accuracy foundation.

References

[1] 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.*, early access, Jan. 12, 2021, doi: 10.1109/TPAMI.2021.3051099.

[2] W. Wang, J. Shen, J. Xie, M.-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.

[3] 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.

[4] Y. Fang, Z. Chen, W. Lin, and C.-W. Lin, "Saliency detection in the compressed domain for adaptive image retargeting," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 3888–3901, Sep. 2012.

[5] K. Gu *et al.*, "Saliency-guided quality assessment of screen content images," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 1098–1110, Jun. 2016.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5706–5722, Dec. 2015.

[15] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[16] 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, pp. 1–15, 2022, doi: 10.1109/TGRS.2021.3091312.

[17] 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, pp. 1–13, 2022, doi: 10.1109/TGRS.2021.3101359.

[18] 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.

[19] 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.

[20] 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.

[21] X. Zhao, Y. Pang, L. Zhang, H. Lu, and L. Zhang, "Suppress and balance: A simple gated network for salient object detection," in *Proc. ECCV*, Aug. 2020, pp. 35–51.

[22] J. 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.

[23] S.-H. Gao, Y.-Q. Tan, M.-M. Cheng, C. Lu, Y. Chen, and S. Yan, "Highly efficient salient object detection with 100k parameters," in *Proc. ECCV*, Aug. 2020, pp. 702–721.

[24] 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.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, May 2015, pp. 1–14.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[27] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[28] 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.*, Apr. 2018, pp. 4510–4520.

[29] 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, doi: 10.1109/TCYB.2020.3035613.

[30] D. Faur, I. Gavat, and M. Datcu, "Salient remote sensing image segmentation based on rate-distortion measure," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 4, pp. 855–859, Oct. 2009.

[31] L. Zhang, S. Wang, and X. Li, "Salient region detection in remote sensing images based on color information content," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 1877–1880.

[32] D. Zhao, J. Wang, J. Shi, and Z. Jiang, "Sparsity-guided saliency detection for remote sensing images," *J. Appl. Remote Sens.*, vol. 9, no. 1, pp. 1–14, Sep. 2015.

[33] L. Zhang, Y. Wang, and Y. Sun, "Salient target detection based on the combination of super-pixel and statistical saliency feature analysis for remote sensing images," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 2336–2340.

[34] L. Zhang, Y. Liu, and J. Zhang, "Saliency detection based on self-adaptive multiple feature fusion for remote sensing images," *Int. J. Remote Sens.*, vol. 40, no. 22, pp. 8270–8297, May 2019.

[35] Z. Huang, H.-X. Chen, T. Zhou, Y.-Z. Yang, C.-Y. Wang, and B.-Y. Liu, "Contrast-weighted dictionary learning based saliency detection for VHR optical remote sensing images," *Pattern Recognit.*, vol. 113, May 2021, Art. no. 107757.

[36] H. Chen, T. Gao, W. Chen, Y. Zhang, and J. Zhao, "Contour refinement and EG-GHT-based inshore ship detection in optical remote sensing image," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8458–8478, Nov. 2019.

[37] Z. Liu, D. Zhao, Z. Shi, and Z. Jiang, "Unsupervised saliency model with color Markov chain for oil tank detection," *Remote Sens.*, vol. 11, no. 9, pp. 1–18, May 2019.

[38] Q. Zhang, L. Zhang, W. Shi, and Y. Liu, "Airport extraction via complementary saliency analysis and saliency-oriented active contour model," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 7, pp. 1085–1089, Jul. 2018.

[39] L. Zhang, A. Li, Z. Zhang, and K. Yang, "Global and local saliency analysis for the extraction of residential areas in high-spatial-resolution remote sensing image," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 7, pp. 3750–3763, Jul. 2016.

[40] 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.*, early access, Nov. 30, 2021, doi: 10.1109/TGRS.2021.3131221.

[41] L. Zhang and J. Ma, "Salient object detection based on progressively supervised learning for remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9682–9696, Nov. 2021, doi: 10.1109/TGRS.2020.3045708.

[42] 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.

[43] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Jun. 2020.

[44] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. ECCV*, Sep. 2018, pp. 3–19.

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, vol. 37, Jul. 2015, pp. 448–456.

[46] 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.*, early access, Nov. 24, 2020, doi: 10.1109/TPAMI.2020.3040258.

[47] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Proc. NIPS*, Dec. 2016, pp. 289–297.

[48] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2261–2269.

[49] C. Li, Y. Yuan, W. Cai, Y. Xia, and D. D. Feng, "Robust saliency detection via regularized random walks ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2710–2717.

[50] J. Kim, D. Han, Y.-W. Tai, and J. Kim, "Salient region detection via high-dimensional color transform and local spatial support," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 9–23, Jan. 2016.

[51] L. Zhou, Z. Yang, Z. Zhou, and D. Hu, "Salient region detection using diffusion process on a two-layer sparse graph," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5882–5894, Dec. 2017.

[52] H. Peng, B. Li, H. Ling, W. Hu, W. Xiong, and S. J. Maybank, "Salient object detection via structured matrix decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 818–832, Apr. 2017.

[53] Y. Yuan, C. Li, J. Kim, W. Cai, and D. D. Feng, "Reversion correction and regularized random walk ranking for saliency detection," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1311–1322, Mar. 2018.

[54] 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. IEEE Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5300–5309.

[55] 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.

[56] Z. Deng *et al.*, "R^3Net: Recurrent residual refinement network for saliency detection," in *Proc. IJCAI*, Jul. 2018, pp. 684–690.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1395–1403.

[64] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NIPS*, Dec. 2019, pp. 8024–8035.

[65] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, May 2015, pp. 1–15.

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

[67] 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.

[68] 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.

[69] 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. IJCAI*, May 2018, pp. 698–704.

Gongyang Li received the B.E. degree from Shanghai Normal University, Shanghai, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Communication and Information Engineering, Shanghai University, Shanghai.

His research interests include image/video object segmentation and saliency detection.

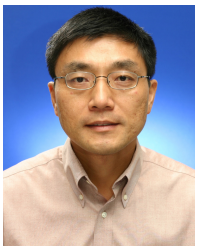
Zhi Liu (Senior Member, IEEE) received the B.E. and M.E. degrees from Tianjin University, Tianjin, China, in 1999 and 2002, respectively, and the Ph.D. degree from the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China, in 2005.

From August 2012 to August 2014, he was a Visiting Researcher with the SIROCCO Team, IRISA/INRIA, Rennes, France, with the support by 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, VCIP 2016, ICME 2014, and WIAMIS 2013. He co-organized special sessions on visual attention, saliency models, and applications at WIAMIS 2013 and ICME 2014. He is also an Area Editor of *Signal Processing: Image Communication*. He has served as the Guest Editor for the Special Issue on Recent Advances in Saliency Models, Applications and Evaluations in *Signal Processing: Image Communication*.

Zhen Bai received the B.E. degree from the Wuhan Huaxia University of Technology, Wuhan, China, in 2016, and the M.S. degree from the Zhengzhou University of Light Industry, Zhengzhou, China, in 2019. She is currently pursuing the Ph.D. degree with the School of Communication and Information Engineering, Shanghai University, Shanghai, China.

Her research interests include machine learning and saliency detection.



Weisi Lin (Fellow, IEEE) received the Ph.D. degree from 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. His research interests include image processing, visual quality evaluation, and perception-inspired signal modeling, with more than 340 refereed papers published in international journals and conferences.

Dr. Lin is also a fellow of IET, and an Honorary Fellow of the Singapore Institute of Engineering Technologists. He was the Technical-Program Chair of the 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*. He was a Distinguished Lecturer of the Asia-Pacific Signal and Information Processing Association (APSIPA) from 2012 to 2013 and the IEEE Circuits and Systems Society from 2016 to 2017.

Haibin Ling (Senior Member, IEEE) received the B.S. and M.S. degrees from Peking University, Beijing, China, in 1997 and 2000, respectively, and the Ph.D. degree from the University of Maryland, College Park, MD, USA, in 2006.

From 2000 to 2001, he was an Assistant Researcher with Microsoft Research Asia, Beijing. From 2006 to 2007, he worked as a Post-Doctoral Scientist with the University of California at Los Angeles, Los Angeles, CA, USA. In 2007, he joined Siemens Corporate Research, Princeton, NJ, USA, as a Research Scientist; then, from 2008 to 2019, he worked as a Faculty Member with the Department of Computer Sciences, Temple University, Philadelphia, PA, USA. In fall 2019, he joined Stony Brook University, Stony Brook, NY, USA, as a SUNY Empire Innovation Professor at the Department of Computer Science. His research interests include computer vision, augmented reality, medical image analysis, and human–computer interaction.

Dr. Ling received the Best Student Paper Award from ACM UIST in 2003, the NSF CAREER Award in 2014, the Yahoo Faculty Research and Engagement Award in 2019, and the Amazon Machine Learning Research Award in 2019, and the Best Journal Paper Award from IEEE VR in 2021. He has served as an Area Chair various times for CVPR and ECCV. He currently serves as an Associate Editor for several journals, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (PAMI), *Pattern Recognition* (PR), and *Computer Vision and Image Understanding* (CVIU).