Two-Stage Edge Reuse Network for Salient Object Detection of Strip Steel Surface Defects

Chengjun Han, Gongyang Li, and Zhi Liu, *Senior Member, IEEE*

***Abstract*— Recently, researchers have paid more attention to salient object detection (SOD) of strip steel surface defects, but there are lots of difficulties, such as a mass of noise, blurry defect boundaries, complex backgrounds, and various types of defects. The existing image SOD methods fail to overcome the above challenging scenes. To address them, in this article, we propose a novel two-stage edge reuse network (TSERNet), which consists of two stages, i.e., prediction and refinement. In the first prediction stage, we construct a primary net based on the encoder–decoder architecture. The encoder not only extracts multiscale features but also generates an edge map. In the decoder, we propose a novel edge-aware foreground–background integration (EFBI) module to distinguish the foreground and the background through edge features and reverse the attention mechanism, and then exploit the decoder to generate an initial saliency map. In the second refinement stage, we construct a sub-net based on the same architecture as that of the first stage. Its encoder extracts features from the initial saliency map, and its decoder deploys the edge-aware refinement module (ERM), which reuses the edge map generated from the first stage, to enhance these features for purifying the initial saliency map, resulting in the final saliency map. Comprehensive experiments on the public dataset show that our proposed TSERNet is consistently superior to 22 relevant state-of-the-art methods. The code and results of our method are available at https://github.com/monxxcn/TSERNet.**

***Index Terms*— Edge, reverse attention, salient object detection (SOD), strip steel surface defects.**

NOMENCLATURE

SOD	Salient object detection.
TSERNet	Two-stage edge reuse network.
EFBI	Edge-aware foreground–background integration.
ERM	Edge-based refinement module.
BIG-Part	Background information generation part.
FIG-Part	Foreground information generation part.
S_α	S-measure.
F_β	F-measure.
E_ξ	E-measure.
$\mathcal{M}$	Mean absolute error.
T.N.	Traditional SOD method for natural images.
D.L.N.	Deep-learning-based SOD method for natural images.
D.L.D.	Deep-learning-based SOD method for defect images.

Manuscript received 9 February 2022; revised 19 July 2022; accepted 9 August 2022. Date of publication 18 August 2022; date of current version 2 September 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62171269 and in part by the China Scholarship Council under Grant 202006890079. The Associate Editor coordinating the review process was Dr. Bardia Yousefi. *(Chengjun Han and Gongyang Li contributed equally to this work.) (Corresponding author: Zhi Liu.)*

The authors are with Shanghai Institute for Advanced Communication and Data Science, and the School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China (e-mail: monxxcn@gmail.com; ligongyang@shu.edu.cn; liuzhisjtu@163.com).

Digital Object Identifier 10.1109/TIM.2022.3200114

I. INTRODUCTION

AS THE foundation of the industrial society, the steel industry has always been valued by people. People take it as an important indicator to evaluate the degree of industrial modernization. So how to improve the quality of steel is an issue that has attracted much attention in the industry. Subject to the influence of production technology, surface defects in the steel produced are inevitable. As shown in the first and second rows of Fig. 1, the types of strip steel defects are varied, the first two columns are inclusions, the second two columns are patches, and the last two columns are scratches. These strip steel defects seriously affect the quality of steel. Therefore, an automated accurate steel defect detection method is very important for the production process.

With the fast development of computer vision, the use of computer vision for defect detection has become popular [3], [4], [5]. However, the traditional detection methods heavily rely on prior knowledge and a lot of constraints. On the other side, the traditional methods often fail in the highly complex background. Fortunately, the deep-learning-based methods can effectively solve these problems. In this article, we explore the salient object detection (SOD) of strip steel surface defects [1], which aims at segmenting the area that people are interested in. For conciseness, in the rest of the article, we specifically use defects to refer to strip steel surface defects. In general, visual saliency is the basic mechanism for people to understand the world, and SOD has been applied to other tasks successfully, such as object tracking [6], image compression [7], object segmentation [8], [9], and person reidentification [10].

Recently, SOD of defects has become popular. The traditional SOD methods of natural images are obviously not suitable for SOD of defects. The deep-learning-based SOD methods of natural images have greatly improved the detection accuracy of natural images. However, in most cases, directly using the traditional methods or deep-learning-based methods of natural images to defect images may not achieve good results. As shown in Fig. 2, the traditional natural image methods (i.e., 2LSG [11] and BC [12]) perform poorly and can only roughly calibrate the position of the salient defects. The

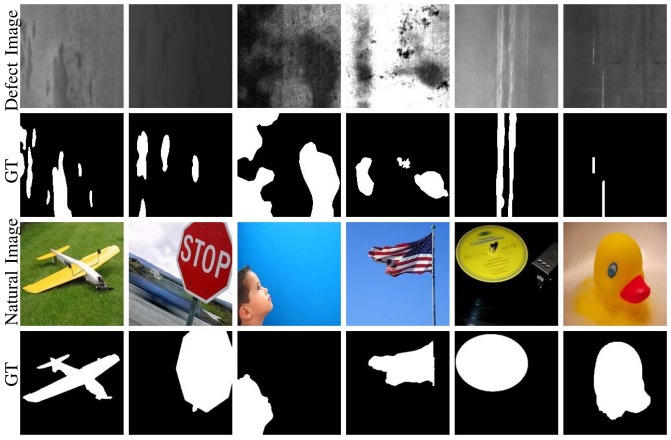

Fig. 1. Visual examples of strip steel surface defects from the SD-saliency-900 dataset [1] and natural images from the MSRA10K dataset [2]. GT means the ground truth.

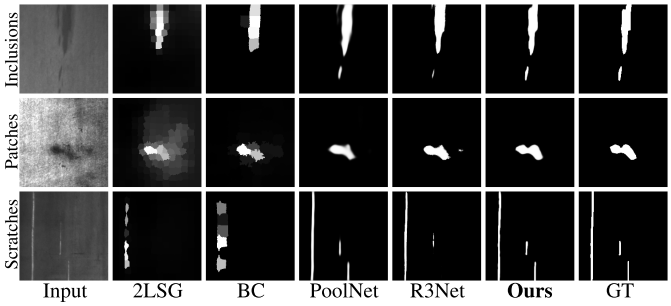

Fig. 2. Saliency maps generated by various categories of SOD methods, in which 2LSG [11] and BC [12] are the traditional natural image methods, PoolNet [13] and R3Net [14] are the deep-learning-based natural image methods, and our method is the deep-learning-based defect image method.

detection accuracy of the deep-learning-based natural image methods (i.e., PoolNet [13] and R3Net [14]) is improved, but the local details appear blurry or even missing. This is caused by the huge difference between natural images and defect images. As shown in Fig. 1, compared with natural images, defect images are mainly shot in industrial production scenes. Therefore, the scale, shape, category, illumination and, external interference of salient objects in strip steel surface images are different from those in natural images to some extent. So, it may be inappropriate to directly use the natural image SOD methods for SOD of defects.

However, we can still use the ideas and principles of natural image SOD methods to SOD of defects. For example, we can use the mainstream architectures of SOD, such as the encoder–decoder network [15] and side-fusion network [16], to build the model, deploy popular structures, such as VGG [17] and ResNet [18], to extract the basic features, introduce the attention mechanisms, such as squeeze-and-excitation block [19] and CBAM [20], to make the model pay more attention to the salient regions of features and use other information of the salient object, such as edge and background, to participate in the detection.

Based on the above motivation, we propose a novel two-stage edge reuse network (TSERNet) for SOD of defects, which follows a two-stage prediction refinement strategy. The networks of both the stages are built on the encoder–decoder architecture. The first stage is a primary net, which is responsible for generating the initial saliency map. To better predict the salient defects from multiple aspects, our primary net also generates an edge map based on the low-level and high-level basic features generated from the encoder. Meanwhile, we propose a novel edge-aware foreground–background integration (EFBI) module that uses the edge map to distinguish the foreground and background regions in features. With the cooperation of the encoder–decoder architecture, EFBI modules, and edge, the primary net generates an initial saliency map, which locates salient defects but with rough details. Therefore, we propose a sub-net with the edge-based refinement module (ERM) for saliency map refinement, whose input is the initial saliency map and the edge map. ERM not only reuses the edge features obtained by the primary net but also generates other edge features through the features extracted from the initial saliency map. Taking advantage of ERMs, our sub-net can gradually refine the details of salient defect objects and generate an accurate saliency map. In addition, inspired by the deep supervision mechanism [16], we deploy it in our TSERNet. This way is not only beneficial to network training but also can better enable the network to enhance the representation of salient defect objects and edge.

In summary, the main contributions of our work are threefold as follows.

1) We propose a novel TSERNet, which consists of a primary net and a sub-net, for SOD in the defect images. Our TSERNet follows a two-stage *prediction-refinement* strategy, i.e., prediction in the primary net and refinement in the sub-net, and uses edge information twice.
2) We propose an EFBI module in the primary net. It can recover saliency information well from foreground features and edge features through reverse attention. Thus, with the EFBI modules, the primary net can distinguish the foreground from the background based on edge, generating the initial saliency map.
3) We propose an ERM in the sub-net. It combines features of the initial saliency map and edge features of two different sources for salient region refinement. With ERMs, the sub-net can refine the initial saliency map of the primary net, generating the final saliency map with more accurate boundaries and more precise location for the defect objects.

The rest of this article is organized as follows. First, Section II reviews the related work of SOD for natural images and for defect images. Next, in Section III, we present our TSERNet in detail. Then, Section IV elaborates the experiments and ablation studies. Finally, we draw the conclusion in Section V.

II. Related Work

In this section, we briefly introduce the related work of SOD. First, we introduce the traditional SOD methods for natural images, and then introduce the deep-learning-based SOD methods for natural images. Finally, we introduce the SOD methods specifically designed for defect images.

A. Traditional SOD Methods for Natural Images

The SOD methods for natural images can be roughly divided into two types, i.e., the traditional methods and the deep-learning-based methods. As a pioneering work in the field of saliency detection, Itti *et al.* [21] expressed the problem of visual saliency using a computational model. The idea of center-surround difference proposed by Itti *et al.* was also adopted by many subsequent methods. For example, Liu and Gleicher [22] proposed scale-invariant saliency and region enhanced saliency on the basis of center-surround difference. To capture the contextual properties of image pixels or regions better, Li *et al.* [23] modeled an image as a hypergraph and analyzed the contrast at patch level. Shi *et al.* [24] achieved a lightweight model which is based on color and structure contrasts and accelerates the detection speed by fusing complementary contrast measures in a pixel-wise adaptive manner.

In addition, priors play an important role in the traditional methods. To obtain a better prediction result, various kinds of priors have been added to the traditional methods. Chang *et al.* [25] exploited shape information to address the SOD task using objectness to help estimate saliency. Wei *et al.* [26] focused more on the background rather than the object and used the background information as the priors to detect the salient regions. Just using the background information may be not reliable sometimes. Researchers realized that the boundary of objects is essential in SOD. In [12], the spatial layout of the image regions with respect to the image boundaries was used to help detect a robust background measure, named background connectivity prior. Niu *et al.* [27] further explored the potential of the background connectivity prior. They proposed a salient object segmentation method based on the superpixel and background connectivity prior and actually boosted the performance. Although the above traditional SOD methods cannot achieve satisfactory performance, they provide a lot of inspiration and ideas for follow-up works.

B. Deep-Learning-Based SOD Methods for Natural Images

The detection accuracy of the traditional methods relies on the handcrafted features. Recently, the deep-learning-based methods have become popular and have been introduced into the field of SOD. Relying on the powerful feature extraction ability of neural network [28], the deep-learning-based methods eliminate the need for handcrafted features and reduce the dependence on the prior bias.

Inspired by the solution of edge detection [29], Hou *et al.* [16] introduced short connections to the skip-layer structure for SOD. Zhang *et al.* [30] explored the way of aggregating multilevel convolution feature maps and predicted saliency maps with multiresolution features. Zhang *et al.* [31] further explored multiscale information fusion strategy and designed a bidirectional structure to pass messages between multilevel features. Chen *et al.* [32] rethought the gap between multilevel features and proposed GCPANet to integrate features from different levels. Gupta *et al.* [33] proposed a gate-based context extraction module which can retain relevant contextual information across multiple scales. Pang *et al.* [34] used the features of adjacent layers by learning from each other. Gupta *et al.* [35] paid more attention to the variability in scales of salient objects and proposed the adjacent layer attention block and partial encoder–decoder block to mine abundant scale features at the current resolution and fuse the multiscale features, respectively. In [36], Zhang *et al.* introduced the classic spatial attention and channel attention to achieve better performance. Then, Liu *et al.* [37] proposed the pixelwise contextual attention to focus more on contextual information. Most methods are based on the attention mechanisms to focus on the foreground regions, while Chen *et al.* [38] constructed the top–down reverse attention based on the background information. Zhang *et al.* [39] proposed a novel progressive dual-attention residual network (PDRNet) to exploit two complementary attention maps to guide residual learning, thus progressively refining prediction in a coarse-to-fine manner. In addition, researchers continued to improve the performance of SOD in other aspects. Liu *et al.* [40] explored the potential of pooling techniques for SOD. Wang *et al.* [41] deployed the weakly supervision strategy to detect salient objects.

In addition to the attention mechanism, researchers also realized the importance of edge information in SOD. Liu *et al.* [13] and Qin *et al.* [42] constructed the U-shape network, and both used the edge features in a residual way to refine the prediction map. In [43], [44], [45], and [46], researchers generated the initial prediction map and edge map in parallel and used the edge map to guide the prediction of saliency map. Wu *et al.* [47] introduced not only the edge information but also the skeleton information to construct an edge-aware SOD network. In [48], Tu *et al.* completed the SOD task with the edge map as the input to the network.

Although the deep-learning-based methods for natural images have achieved good performance, due to the complexity of defect images, it is not suitable to directly apply these methods to defect images. Compared with the edge-based methods, in our work, we not only generate an edge map to guide prediction but also obtain the final refinement result in a residual way. Meanwhile, due to the characteristic of strip steel surface images, we integrate the foreground information, background information, and edge information to predict the initial results as accurately as possible in the first stage. Then, we reuse the edge map in the second stage to refine the initial results to generate the final prediction map.

C. SOD Methods for Defect Images

Due to the differences between defect images and natural images, SOD of defect images is more challenging than SOD of natural images. Therefore, it is inevitable to explore the special solution for SOD of defect images. Zhou *et al.* [49] proposed a traditional SOD method for defect images. They considered the correlation between the foreground and the background and noted the double low-rank and sparse characteristics of defect images, resulting in a good performance. To better carry out the SOD task, in [50], Song *et al.* published the dedicated dataset of strip steel surface defects, called SD-saliency-900. Moreover, they noted the differences between the natural images and the strip steel surface defect

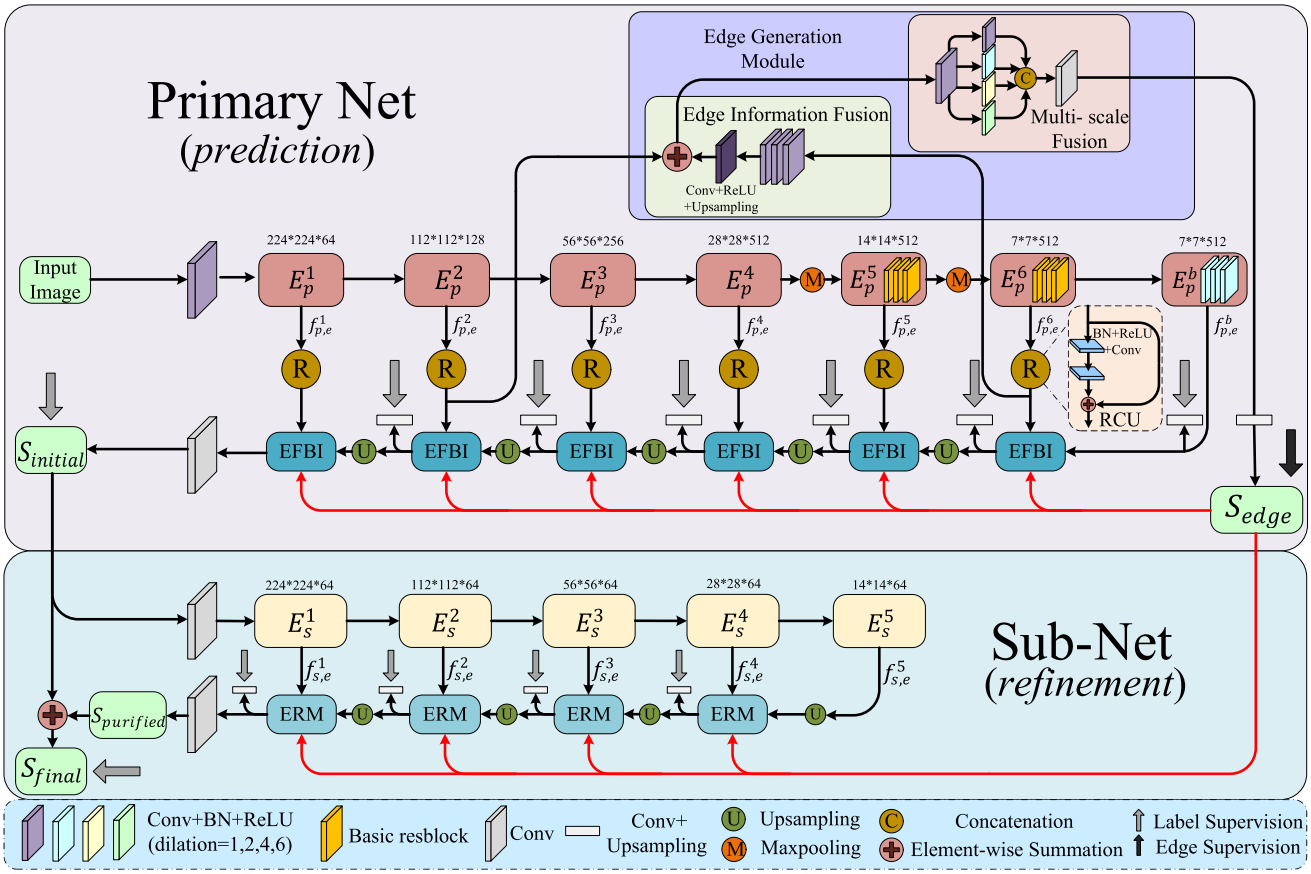

Fig. 3. Overall framework of our TSERNet, which is built on the encoder–decoder architecture and consists of two parts, i.e., primary net and sub-net. Primary net is responsible for prediction, while sub-net is responsible for refinement. In the primary net, we first extract the basic features from an input defect image with a size of $224 \times 224 \times 3$ and generate an edge map in the edge generation module (EGM) based on the low-level and high-level basic features. Then, we use the basic features and the edge information to generate an initial saliency map in the decoder with the EBFI modules. In the sub-net, we propose an ERM to reuse the edge information for saliency map refinement and generate the final saliency map. In the training phase, we use the supervision to each decoder and the EGM. For each decoder, we adopt a hybrid loss function, including the pixel-level BCE loss, the patch-level SSIM loss, the map-level IoU loss, and the metric-aware F-m loss to supervise the intermediate saliency maps. For the EGM, we adopt the BCE loss to supervise the edge map.

images and proposed an encoder–decoder residual network. To better integrate the low-level features and semantic features, they deployed the channel weighted block and the residual decoder block alternatively. Similar to SOD, Dong *et al.* [51] were committed to segmenting the defects and proposed a global context attention network, in which the feature pyramid fusion strategy allows the low-resolution information and high-resolution information to better integrate. To better detect the defect regions from the complex scenes of strip steel surface images, Zhou *et al.* [52] proposed a dense attention-guided cascaded network and deployed the dense attention mechanism to steer the deep features pay more concerns to the defect regions. Zhou *et al.* [53] also noted the importance of edge information in SOD of defects. They adopted the U-shape architecture where the two crucial points are the interactive feature integration and the edge-guided saliency fusion and added the edge extraction branch after each decoder block to preserve the salient object boundaries in the prediction results.

With the development of steel industry, the SOD of strip steel surface defects attracts researchers' attention, and researchers focus on coping with the complexity of defect images. In this article, we explore the edge information to address the challenging scenes of defect images. Specifically, we generate an edge map and use it to distinguish the foreground and the background. We also mine the complementary information contained in the generated edge map and features for object details' refinement.

III. Proposed Method

In this section, we describe our TSERNet in detail. First, the overview of our TSERNet is presented in Section III-A. Then, in Section III-B, we show the details of our EFBI module. In Section III-C, we elaborate our ERM. Finally, in Section III-D, we introduce the loss function.

A. Network Overview

As shown in Fig. 3, our TSERNet consists of two key parts, i.e., primary net for prediction and sub-net for refinement, and both the parts are based on the encoder–decoder architecture, which is widely used in the SOD task [1], [54], [55].

The primary net of our TSERNet can be divided into three parts, including the encoder network, the EGM, and the decoder network. We adopt ResNet-34 [18] as the encoder network for basic feature extraction, and it consists of an input

convolution layer, six encoder blocks, and a bridge block. The input convolution layer has 64 convolution filters with a kernel size of 3×3 and a stride of 1. Compared with the original ResNet-34, we do not deploy the pooling operation after the input layer. We denote these six encoder blocks as $\boldsymbol{E}_p^{i\in\{1,2,3,4,5,6\}}$ and the bridge block as $\boldsymbol{E}_p^b$. For $i = 1, 2, 3, 4$, similar to the original ResNet-34 [18], $\boldsymbol{E}_p^i$ has three, four, six, and three BasicBlocks [18], respectively; while for $i = 5, 6$, $\boldsymbol{E}_p^i$ has three BasicBlocks. $\boldsymbol{E}_p^b$ consists of three convolution layers with a kernel size of 3×3 and a dilation rate of 2. For an input image $\mathbf{I} \in \mathbb{R}^{224\times224\times3}$, we denote the extracted features of $\boldsymbol{E}_p^i$ as $\boldsymbol{f}_{p,e}^i \in \mathbb{R}^{h_i\times w_i\times c_i}$ where h_i is $(224/2^{i-1})$, w_i is $(224/2^{i-1})$, and $c_i \in \{64, 128, 256, 512, 512, 512\}$. The output feature of $\boldsymbol{E}_p^b$ is denoted as $\boldsymbol{f}_{p,e}^b$ which has the same size as $\boldsymbol{f}_{p,e}^6$. Inspired by [56], we add a residual convolution unit (RCU) [56] to each encoder block to refine the details of $\boldsymbol{f}_{p,e}^i$ via the residual connection for better feature representation. Therefore, $\boldsymbol{f}_{p,e}^i$ is redefined as the output features of RCU.

Notably, inspired by EGNet [43], in the primary net, we propose an EGM to integrate $\boldsymbol{f}_{p,e}^6$, which has rich context information, and $\boldsymbol{f}_{p,e}^2$, which contains details, to generate an edge map $\boldsymbol{S}_{\text{edge}} \in \mathbb{R}^{224\times224\times1}$. As shown in the top of Fig. 3, our EGM first implements the edge information fusion via the elementwise summation, and then uses the multiscale fusion operation, which has four branches with a kernel size 3×3 and different dilation rates, to collect the edge information in different receptive fields.

For the decoder network of primary net, we adopt the EFBI module as the decoder block, and its output features are denoted as $\boldsymbol{f}_{\text{efbi}}^i \in \mathbb{R}^{h_i\times w_i\times c_i}$, where $c_i \in \{64, 64, 128, 256, 512, 512\}$. With the above three parts working together, the primary net finally generates the initial saliency map $\boldsymbol{S}_{\text{initial}} \in \mathbb{R}^{224\times224\times1}$. To avoid overfitting [57], each decoder block and $\boldsymbol{S}_{\text{initial}}$ are supervised by the ground truth.

The sub-net is similar to the primary net, but simpler. Its encoder network consists of five encoder blocks denoted as $\boldsymbol{E}_s^{j\in\{1,2,3,4,5\}}$. For $j = 1, 2, 3, 4$, the encoder block has one convolution layer with a max-pooling operation; while for $j = 5$, the encoder block only has a convolution layer. Considering that the input of this encoder network is the initial saliency map $\boldsymbol{S}_{\text{initial}}$ which does not have as much information as the input image $\boldsymbol{I}$, the channel number of the encoder block in the sub-net is limited to 64. The extracted features of $\boldsymbol{E}_s^j$ are denoted as $\boldsymbol{f}_{s,e}^j \in \mathbb{R}^{h_j\times w_j\times c}$, where h_j is $(224/2^{j-1})$, w_j is $(224/2^{j-1})$, and c is 64. Correspondingly, we adopt the ERM as the decoder block of the sub-net, and its output features are denoted as $\boldsymbol{f}_{\text{erm}}^j \in \mathbb{R}^{h_j\times w_j\times c}$. Through the above operations, the sub-net purifies $\boldsymbol{S}_{\text{initial}}$ and generates the purified saliency map $\boldsymbol{S}_{\text{purified}} \in \mathbb{R}^{224\times224\times1}$, which has more accurate details. Finally, we get the final saliency map $\boldsymbol{S}_{\text{final}} \in \mathbb{R}^{224\times224\times1}$ by adding $\boldsymbol{S}_{\text{initial}}$ to $\boldsymbol{S}_{\text{purified}}$.

B. Edge-Aware Foreground–Background Integration Module

The foreground information plays an important role in the field of SOD. Some mainstream methods [30], [31] based on the foreground information have achieved good performance. Some methods [36], [37] deploy attention mechanism on SOD, which forces the network to better focus on the foreground regions in features. In addition to the foreground information, the background information steps into researchers' sight. Reverse attention [22] is a good practice to incorporate the background information into the network. The above successful SOD solutions of nature images are worth learning from.

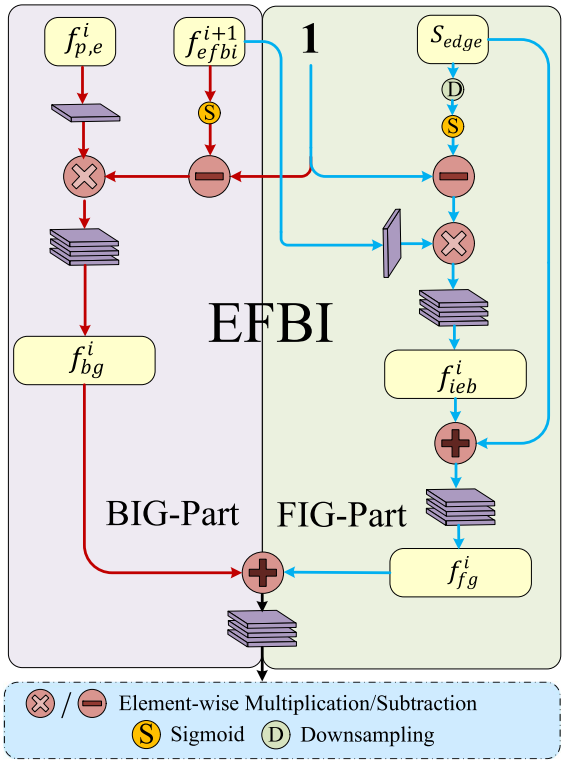

Fig. 4. Illustration of the edge-aware foreground–background integration (EFBI) module.

Compared with nature images that contain strong semantic information, defect images contain weaker semantic information, and the border of the salient defects and nonsalient area in defect images is usually blurry. This characteristic of defect images makes the encoder difficult to extract effective feature representations that can strongly distinguish salient defects and nonsalient area. The above methods for natural images that only rely on the foreground information or background information may not be able to cope with this characteristic well. In addition, some edge-based methods [13], [43], [53] use edge information in a simple way, such as the elementwise summation [13], [43] or concatenation [53]. To this end, we propose an EFBI module, which not only introduces the foreground and background information but also introduces the clear edge information to distinguish and integrate them. We illustrate the structure of the EFBI module in Fig. 4. Its inputs are $\boldsymbol{f}_{p,e}^i$, $\boldsymbol{f}_{\text{efbi}}^{i+1}$ ($\boldsymbol{f}_{p,e}^b$ when $i = 6$) and $\boldsymbol{S}_{\text{edge}}$, which are, respectively, generated from the corresponding stage in the encoder, the previous stage in the decoder, and the EGM. The EFBI module can be divided into two key parts, i.e., background information generation part (BIG-Part) and foreground information generation part (FIG-Part). In the

following, we elaborate the EFBI module based on these two parts.

1) Background Information Generation Part: BIG-Part is shown on the left of Fig. 4, and its inputs are $\boldsymbol{f}_{\text{efbi}}^{i+1}$ ($\boldsymbol{f}_{p,e}^{b}$ when $i = 6$) and $\boldsymbol{f}_{p,e}^{i}$. Since the output of the previous EFBI module $\boldsymbol{f}_{\text{efbi}}^{i+1}$ ($i \in \{1, 2, 3, 4, 5\}$) can reflect the salient defects and $\boldsymbol{f}_{p,e}^{b}$ contains more global semantic information, we extract the background information from them via reverse attention [22] and identify the background regions in $\boldsymbol{f}_{p,e}^{i}$, which has blurry border.

Concretely, in BIG-Part, we first generate a group of attention maps from $\boldsymbol{f}_{\text{efbi}}^{i+1}$ through the sigmoid activation function, and then adopt the reverse attention operation to obtain a group of background maps. After that, we adopt the background maps to modulate $\boldsymbol{f}_{p,e}^{i}$ and generate the background features $\boldsymbol{f}_{\text{bg}}^{i} \in \mathbb{R}^{h_i \times w_i \times c_i}$. The above operations in BIG-Part can be expressed as

$$\boldsymbol{f}_{\text{bg}}^{i} = \begin{cases} C_3\big(\big(\mathbf{1} \ominus s\big(\boldsymbol{f}_{\text{efbi}}^{i+1}\big)\big) \otimes C\big(\boldsymbol{f}_{p,e}^{i}\big)\big), & i = 1, 2, 3, 4, 5 \\ C_3\big(\big(\mathbf{1} \ominus s\big(\boldsymbol{f}_{p,e}^{b}\big)\big) \otimes C\big(\boldsymbol{f}_{p,e}^{i}\big)\big), & i = 6 \end{cases} \tag{1}$$

where $C(\cdot)/C_3(\cdot)$ is one/three convolution layers, and each convolution layer contains a 3×3 convolution layer, a batch normalization (BN) layer [58], and a ReLU activation function [59]. $\mathbf{1}$ is a matrix with the same dimension as the subtracted features and all the elements are 1, $s(\cdot)$ is a sigmoid activation function, and $\otimes/\ominus$ is the elementwise multiplication/subtraction.

2) Foreground Information Generation Part: FIG-Part is shown on the right of Fig. 4, and its inputs are $\boldsymbol{S}_{\text{edge}}$ and $\boldsymbol{f}_{\text{efbi}}^{i+1}$ ($\boldsymbol{f}_{\text{efbi}}^{b}$ when $i = 6$). $\boldsymbol{S}_{\text{edge}}$ indicates the clear boundary of salient defects, and $\boldsymbol{f}_{\text{efbi}}^{i+1}$ is sensitive to the edge. Therefore, we first reverse $\boldsymbol{S}_{\text{edge}}$ and multiply it with $\boldsymbol{f}_{\text{efbi}}^{i+1}$ to enhance the pure foreground region, generating the pure foreground features $\boldsymbol{f}_{\text{ieb}}^{i} \in \mathbb{R}^{h_i \times w_i \times c_i}$, which can be computed as

$$\boldsymbol{f}_{\text{ieb}}^{i} = \begin{cases} C_3\big(C\big(\boldsymbol{f}_{\text{efbi}}^{i+1}\big) \otimes \big(\mathbf{1} \ominus s\big(D\big(\boldsymbol{S}_{\text{edge}}\big)\big)\big)\big), & i = 1, 2, 3, 4, 5 \\ C_3\big(C\big(\boldsymbol{f}_{p,e}^{b}\big) \otimes \big(\mathbf{1} \ominus s\big(D\big(\boldsymbol{S}_{\text{edge}}\big)\big)\big)\big), & i = 6 \end{cases} \tag{2}$$

where $D(\cdot)$ is the downsampling operation.

Then, we directly integrate $\boldsymbol{S}_{\text{edge}}$ with $\boldsymbol{f}_{\text{ieb}}^{i}$ to complete the foreground region with fine boundaries via the elementwise summation and apply several convolution layers for feature smoothing, generating the foreground features $\boldsymbol{f}_{\text{fg}}^{i} \in \mathbb{R}^{h_i \times w_i \times c_i}$ as follows:

$$\boldsymbol{f}_{\text{fg}}^{i} = C_3\big(\boldsymbol{f}_{\text{ieb}}^{i} \oplus \boldsymbol{S}_{\text{edge}}\big). \tag{3}$$

Finally, we integrate $\boldsymbol{f}_{\text{bg}}^{i}$, which contains rich background information, and $\boldsymbol{f}_{\text{fg}}^{i}$, which indicates the foreground region, to generate the output features of the EFBI module, $\boldsymbol{f}_{\text{efbi}}^{i}$, as follows:

$$\boldsymbol{f}_{\text{efbi}}^{i} = C_3\Big(\boldsymbol{f}_{\text{fg}}^{i} \oplus \boldsymbol{f}_{\text{bg}}^{i}\Big) \tag{4}$$

where $\oplus$ is the elementwise summation.

The above fusion strategy makes $\boldsymbol{f}_{\text{efbi}}^{i}$ easier to distinguish the foreground information and background information by the clear edge information and facilitates the subsequent inference of salient defects.

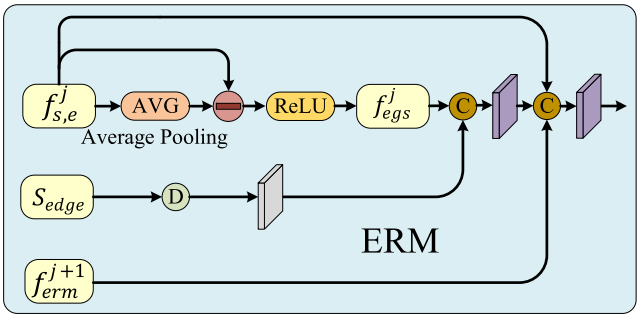

Fig. 5. Illustration of the edge-based refinement module (ERM).

C. Edge-Based Refinement Module

In our TSERNet, the primary net predicts an initial saliency map $\boldsymbol{S}_{\text{initial}}$, which reflects the salient defects with the blurry boundary and some noise. Therefore, we construct the second stage, i.e., sub-net, which aims to refine $\boldsymbol{S}_{\text{initial}}$. In the sub-net, we make full use of edge information for salient defects' refinement. Specifically, we propose the ERM, which plays an important role in our sub-net. In the ERM, we explore two types of edge information. The first one is the edge map $\boldsymbol{S}_{\text{edge}}$ generated in the first stage, and the second one is extracted from the basic features of the sub-net's encoder $\boldsymbol{f}_{s,e}^{j}$. We illustrate the structure of ERM in Fig. 5.

First, we extract the edge information carried by the features themselves as follows:

$$\boldsymbol{f}_{\text{egs}}^{j} = \text{ReLU}\big(\boldsymbol{f}_{s,e}^{j} \ominus \text{AVG}\big(\boldsymbol{f}_{s,e}^{j}\big)\big) \tag{5}$$

where $\boldsymbol{f}_{\text{egs}}^{j} \in \mathbb{R}^{h_j \times w_j \times c}$ is the self-edge features, $\text{AVG}(\cdot)$ is the average pooling operation, and $\text{ReLU}(\cdot)$ is the ReLU activation function.

Then, we adopt a direct and adaptive manner to integrate the self-edge features $\boldsymbol{f}_{\text{egs}}^{j}$ and the predicted edge map $\boldsymbol{S}_{\text{edge}}$ via the concatenation-convolution operation, which can be formulated as follows:

$$\boldsymbol{f}_{\text{edge}}^{j} = C\Big(\text{Cat}\Big(\boldsymbol{f}_{\text{egs}}^{j}, C\big(D\big(\boldsymbol{S}_{\text{edge}}\big)\big)\Big)\Big) \tag{6}$$

where $\boldsymbol{f}_{\text{edge}}^{j} \in \mathbb{R}^{h_j \times w_j \times c}$ is the fused edge features, and $\text{Cat}(\cdot)$ is the cross-channel concatenation. Notably, to prevent the edge information of $\boldsymbol{S}_{\text{edge}}$ from being overwhelmed by $\boldsymbol{f}_{\text{egs}}^{j}$, we convert $\boldsymbol{S}_{\text{edge}}$ into feature maps with the same dimension as $\boldsymbol{f}_{\text{egs}}^{j}$ via a convolution layer. Since $\boldsymbol{f}_{\text{egs}}^{j}$ and $\boldsymbol{S}_{\text{edge}}$ are two types of edge information from different sources, the above fusion operation can mine their complementary information, which is conducive to refining salient defects.

Finally, we adopt the concatenation-convolution operation to fuse $\boldsymbol{f}_{s,e}^{j}$ from the encoder, the fused edge features $\boldsymbol{f}_{\text{edge}}^{j}$, and the previous $\boldsymbol{f}_{\text{erm}}^{j+1}$, generating the output features of ERM, $\boldsymbol{f}_{\text{erm}}^{j}$, which can be computed as follows:

$$\boldsymbol{f}_{\text{erm}}^{j} = \begin{cases} C\Big(\text{Cat}\Big(\boldsymbol{f}_{\text{edge}}^{j}, \boldsymbol{f}_{s,e}^{j}, \boldsymbol{f}_{\text{erm}}^{j+1}\Big)\Big), & j = 1, 2, 3 \\ C\Big(\text{Cat}\Big(\boldsymbol{f}_{\text{edge}}^{j}, \boldsymbol{f}_{s,e}^{j}, \boldsymbol{f}_{s,e}^{j+1}\Big)\Big), & j = 4. \end{cases} \tag{7}$$

With the help of ERMs, our sub-net can gradually refine the boundary of salient defects based on two types of edge information and generate the purified saliency map $\boldsymbol{S}_{\text{purified}}$.

D. Loss Function

An effective and suitable loss function can improve the performance without modifying the network structure and increasing the model parameters. We adopt the widely used deep supervision [16] to force our TSERNet focus on salient regions at different stages. Two kinds of loss functions are used in our TSERNet, one for the edge supervision denoted as $\mathbb{L}_{\text{edge}}$ and the other for the saliency supervision. There are a total of 12 saliency supervisions in our TSERNet, including six ones in the primary net, four ones in the sub-net, one for $\boldsymbol{S}_{\text{initial}}$, and one for $\boldsymbol{S}_{\text{final}}$. For brevity, we denote the six generated saliency maps of the primary net as $\boldsymbol{S}^i_{\text{sal}}, i \in \{1, 2, 3, 4, 5, 6\}$, $\boldsymbol{S}_{\text{initial}}$ as $\boldsymbol{S}^7_{\text{sal}}$, the four generated saliency maps of the sub-net as $\boldsymbol{S}^i_{\text{sal}}, i \in \{8, 9, 10, 11\}$, and $\boldsymbol{S}_{\text{final}}$ as $\boldsymbol{S}^{12}_{\text{sal}}$. The corresponding saliency supervision for $\boldsymbol{S}^i_{\text{sal}}$ is denoted as $\mathbb{L}^i_{\text{sal}}$ Thus, the total loss $\mathbb{L}_{\text{total}}$ of our TSERNet can be formulated as:

$$\mathbb{L}_{\text{total}} = \mathbb{L}_{\text{edge}} + \sum_{i=1}^{12} \mathbb{L}^i_{\text{sal}}. \tag{8}$$

For each $\mathbb{L}^i_{\text{sal}}$, we resize the predicted saliency map to 224×224 by upsampling to fit the size of GT.

Specifically, we adopt the classic BCE loss [60] for edge supervision, which is denoted as

$$\mathbb{L}_{\text{edge}} = \ell_{\text{bce}}\big(\boldsymbol{S}_{\text{edge}}, \boldsymbol{G}_{\text{edge}}\big) \tag{9}$$

where $\ell_{\text{bce}}(\cdot)$ is the BCE loss, and $\boldsymbol{G}_{\text{edge}}$ is the ground truth of edge.

Inspired by some effective SOD solutions [42], [61], [62], [63], [64], we adopt a hybrid loss function for saliency supervision, including the classic pixel-level BCE loss, the patch-level SSIM loss [65], the map-level IoU loss [66], and the metric-aware F-m loss [67], which is denoted as

$$\begin{aligned}\mathbb{L}^i_{\text{sal}} = \ell_{\text{bce}}\big(\boldsymbol{S}^i_{\text{sal}}, \boldsymbol{G}_{\text{sal}}\big) + \ell_{\text{ssim}}\big(\boldsymbol{S}^i_{\text{sal}}, \boldsymbol{G}_{\text{sal}}\big) \\ + \ell_{\text{IoU}}\big(\boldsymbol{S}^i_{\text{sal}}, \boldsymbol{G}_{\text{sal}}\big) + \ell_{\text{Fm}}\big(\boldsymbol{S}^i_{\text{sal}}, \boldsymbol{G}_{\text{sal}}\big)\end{aligned} \tag{10}$$

where $\boldsymbol{S}_{\text{sal}}$ is the predicted saliency map, and $\boldsymbol{G}_{\text{sal}}$ is the ground truth of saliency map. Notably, ℓ_{ssim}, ℓ_{IoU}, and ℓ_{Fm} are, respectively, denoted as follows:

$$\ell_{\text{ssim}} = 1 - \frac{\big(2\mu_x\mu_y + C_1\big)\big(2\sigma_x\sigma_y + C_2\big)}{\big(\mu_x^2 + \mu_y^2 + C_1\big)\big(\sigma_x^2 + \sigma_y^2 + C_2\big)} \tag{11}$$

where x and y are the corresponding patches from $\boldsymbol{S}_{\text{sal}}$ and $\boldsymbol{G}_{\text{sal}}$, respectively, μ_x/μ_y and σ_x/μ_x are the mean and standard deviation, respectively, of x/y, and C_1 and C_2 are set to 0.01^2 and 0.03^2, respectively, to avoid the dividing by zero

$$\ell_{\text{iou}} = 1 - \frac{\sum_{i=1}^{W\cdot H} \boldsymbol{S}_{\text{sal}}(i) \cdot \boldsymbol{G}_{\text{sal}}(i)}{\sum_{i=1}^{W\cdot H} \big(\boldsymbol{S}_{\text{sal}}(i) + \boldsymbol{G}_{\text{sal}}(i) - \boldsymbol{S}_{\text{sal}}(i) \cdot \boldsymbol{G}_{\text{sal}}(i)\big)} \tag{12}$$

where $\boldsymbol{G}_{\text{sal}}(i) \in \{0, 1\}$ is the ground-truth value, and $\boldsymbol{S}_{\text{sal}}(i) \in [0, 1]$ is the predicted saliency score at the ith pixel

$$\ell_{\text{Fm}} = 1 - \frac{\big(1 + \beta^2\big) \cdot P\big(\boldsymbol{S}_{\text{sal}}, \boldsymbol{G}_{\text{sal}}\big) \cdot R\big(\boldsymbol{S}_{\text{sal}}, \boldsymbol{G}_{\text{sal}}\big)}{\beta^2 \cdot P\big(\boldsymbol{S}_p, \boldsymbol{G}_{\text{sal}}\big) + R\big(\boldsymbol{S}_{\text{sal}}, \boldsymbol{G}_{\text{sal}}\big)} \tag{13}$$

where $P = (\text{TP}/(\text{TP} + \text{FP}))$, $R = (\text{TP}/(\text{TP} + \text{FN}))$, $\text{TP}(\boldsymbol{S}_{\text{sal}}, \boldsymbol{G}_{\text{sal}}) = \sum_{i=1}^{W\cdot H} \boldsymbol{S}_{\text{sal}}(i) \cdot \boldsymbol{G}_{\text{sal}}(i)$, $\text{FP}(\boldsymbol{S}_{\text{sal}}, \boldsymbol{G}_{\text{sal}}) = \sum_{i=1}^{W\cdot H} \boldsymbol{S}_{\text{sal}}(i) \cdot (1 - \boldsymbol{G}_{\text{sal}}(i))$, $\text{FN}(\boldsymbol{S}_{\text{sal}}, \boldsymbol{G}_{\text{sal}}) = \sum_{i=1}^{W\cdot H} (1 - \boldsymbol{S}_{\text{sal}}(i)) \cdot \boldsymbol{G}_{\text{sal}}(i)$, and $\beta^2 = 0.3$.

Algorithm 1 Training of Our TSERNet

Require: dataSet($\mathbf{I}$, $\boldsymbol{G}_{sal}$, $\boldsymbol{G}_{edge}$)
Ensure: Trained TSERNet($.|\omega$)
  batchSize $\leftarrow$ 4, iteration $\leftarrow$ 26000, current $\leftarrow$ 0
  Set $\leftarrow$ dataSet($\mathbf{I}$, $\boldsymbol{G}_{sal}$, $\boldsymbol{G}_{edge}$, batchSize)
  $l \leftarrow 0$
  for current **in range** (iteration) **do**
    for ($\mathbf{I}$, $\boldsymbol{G}_{sal}$, $\boldsymbol{G}_{edge}$) **in enumerate** Set **do**
      $(\boldsymbol{S}^i_{sal}, \boldsymbol{S}_{edge}) \leftarrow$ TSERNet($\mathbf{I}|\omega$), $i = 1, 2, \ldots, 12$
      $l = \mathbb{L}_{\text{total}}$
      $\omega \leftarrow$ update($\partial l/\partial\omega$)
      if current % 1000 == 0 **then**
        Save(TSERNet($.|\omega$))
      end if
      if l is convergent **then**
        return TSERNet($.|\omega$)
      end if
    end for
  end for
  return TSERNet($.|\omega$)

IV. Validation of the Proposed Method

A. Experimental Protocol

1) Datasets: We use the public SD-saliency-900 dataset [50] to train and evaluate the proposed TSERNet. The SD-saliency-900 dataset includes three defect categories, i.e., inclusions, patches, and scratches, with a total of 900 images, and contains the corresponding pixel-level ground truth. All the images are with the resolution of 200×200.

2) Implementation Details: We train and test our proposed TSERNet on the NVIDIA Titan Xp GPU (12 GB memory) based on the PyTorch [68] platform. We show the network training process of our TSERNet in Algorithm 1, in which TSERNet($.|\omega$) is our TSERNet and ω denotes the learnable parameters. Following EDRNet [1], our training set contains 540 images (180 images per class). We adopt the salt and pepper noise for data augmentation. During the training phase, each image is first resized to 256×256 and randomly cropped to 224×224. For the test set, EDRNet [1] adopts all the 900 images as the test set, resulting in the test set containing all the training images, which is unreasonable. Therefore, different from EDRNet [1], we adopt the remaining 360 images from the SD-saliency-900 dataset as the test set, excluding images in the training set. We use the pretrained parameters of ResNet-34 [18] to speed up the loss convergence with the Adam optimizer [69]. The initial learning rate, betas, eps, weight decay, and batch size are set to $1e^{-3}$, (0.9, 0.999), $1e^{-8}$, 0, and 4, respectively, and we just train our TSERNet until the loss converges.

3) Evaluation Metrics: To evaluate our TSERNet and compare with other SOD methods, we use five evaluation metrics, including S-measure [70], F-measure [71], E-measure [72], mean absolute error, and precision–recall curve.

a) S-Measure: (S_α, $\alpha = 0.5$) can efficiently evaluate the structural similarity at object-aware and region-aware levels.

b) F-Measure: (F_β) pays more attention to the quality of boundary and efficiently evaluates the edge details of the prediction results. In our experiment, we adopt its maximum form, denoted as $F_\beta^{\max}$.

c) E-Measure: (E_ξ) proposed two kinds of information, i.e., image-level statistics and local pixel matching information. In our experiment, we adopt its maximum form, denoted as $E_\xi^{\max}$.

d) Mean Absolute Error: (MAE, $\mathcal{M}$) can better reflect the actual difference between the prediction and the corresponding ground truth at pixel level.

e) Precision–Recall Curve: reflects the relationship between the precision and the recall.

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

We compare our TSERNet with 22 state-of-the-art SOD methods, which can be divided into three categories. The first one is the traditional SOD methods for natural images, including BC [12], 2LSG [11], SMD [74], and RCRR [73]. The second class is the deep-learning-based SOD methods for natural images, including DSS [16], NLDF [76], PiCANet [37], R3Net [14], BMPM [31], PFANet [75], PoolNet [13], CPD [77], BASNet [42], EGNet [43], GCPANet [32], MINet [34], SAMNet [78], and SUCA [79]. The last one is the deep-learning-based methods for defect images, including EDRNet [1], DACNet [52], and EMINet [53]. For a fair comparison, the saliency maps of the following 14 methods, including BC, 2LSG, SMD, RCRR, DSS, NLDF, PiCANet, R3Net, BMPM, PFANet, PoolNet, CPD, BASNet, and EDRNet, are derived from the data published by Song *et al.* [1], and the saliency maps of DACNet and EMINet are released by the authors. We pick out the same 360 images from the published data as the test set we used. For EGNet, GCPANet, MINet, SAMNet, SUCA, and ENFNet, we retrained these six methods with their default settings using the same training set as our method.

1) Quantitative Comparison: We list the quantitative comparison results of our TSERNet and all the compared methods in Table I. We use four kinds of evaluation metrics, including S_α, $\mathcal{M}$, $E_\xi^{\max}$, and $F_\beta^{\max}$. Among these four metrics, we want $\mathcal{M}$ to be as small as possible while the remaining three to be as large as possible.

Overall, our method shows excellent performance compared with the three categories of SOD methods on the SD-saliency-900 dataset. Because of the inherent disadvantages of the traditional SOD methods mentioned before, our method is significantly better than these four traditional SOD methods listed in Table I. The traditional SOD methods cannot overcome the difficulties of defect images. Our method outperforms all the deep-learning-based SOD methods for natural images. GCPANet achieves the best performance among this category of methods on SD-saliency-900. Compared with GCPANet, our method achieves about 1.14%, 11.1%, 0.94%, and 1.95% percentage gain in S_α, $\mathcal{M}$, $E_\xi^{\max}$, and $F_\beta^{\max}$, respectively. Compared with EDRNet and DACNet for defect images, our method still shows better performance. Specifically, our method has the same value as EDRNet in $\mathcal{M}$ and slightly surpasses it in $E_\xi^{\max}$. Furthermore, our method has higher S_α and $F_\beta^{\max}$, e.g., S_α: 0.886 (Ours) versus 0.877 (EDRNet), and $E_\xi^{\max}$: 0.888 (Ours) versus 0.872 (EDRNet).

TABLE I

QUANTITATIVE RESULTS OF 22 STATE-OF-THE-ART METHODS, INCLUDING FOUR TRADITIONAL SOD METHODS FOR NATURAL IMAGES, 15 DEEP-LEARNING-BASED SOD METHODS FOR NATURAL IMAGES, AND THREE DEEP-LEARNING-BASED SOD METHODS FOR DEFECT IMAGES, AND OUR TSERNET ON SD-SALIENCY-900 DATASET. ↑/↓ MEANS A LARGER/SMALLER SCORE IS BETTER. TOP THREE RESULTS ARE, RESPECTIVELY, HIGHLIGHTED IN **RED**, **BLUE**, AND **GREEN**

Methods	Type	Param (M)↓	SD-saliency-900 [50]			
			S_α ↑	$\mathcal{M}$ ↓	$E_\xi^{\max}$ ↑	$F_\beta^{\max}$ ↑
BC_{14} [12]	T.N.	-	.592	.156	.675	.470
$2LSG_{17}$ [11]	T.N.	-	.554	.246	.630	.435
SMD_{17} [74]	T.N.	-	.582	.209	.648	.466
$RCRR_{18}$ [73]	T.N.	-	.533	.242	.628	.392
DSS_{17} [16]	D.L.N.	62.23	.775	.032	.893	.786
$NLDF_{17}$ [76]	D.L.N.	35.48	.811	.047	.904	.784
$PiCANet_{18}$ [37]	D.L.N.	47.22	.873	.031	.958	.865
$R3Net_{18}$ [14]	D.L.N.	56.16	.824	.030	.927	.816
$BMPM_{18}$ [31]	D.L.N.	22.09	.822	.037	.924	.827
$PFANet_{19}$ [75]	D.L.N.	37.27	.742	.081	.855	.704
$PoolNet_{19}$ [13]	D.L.N.	53.63	.866	.029	.954	.852
CPD_{19} [77]	D.L.N.	29.20	.858	.031	.953	.853
$BASNet_{19}$ [42]	D.L.N.	87.10	.866	**.027**	.957	.858
$EGNet_{19}$ [43]	D.L.N.	108.07	.867	.028	**.959**	.858
$GCPANet_{20}$ [32]	D.L.N.	67.06	**.876**	**.027**	.956	**.871**
$MINet_{20}$ [34]	D.L.N.	47.56	.868	**.025**	.948	.857
$SAMNet_{21}$ [78]	D.L.N.	1.33	.830	.038	.933	.820
$SUCA_{21}$ [79]	D.L.N.	117.71	.869	**.027**	.956	.860
$ENFNet_{21}$ [48]	D.L.N.	37.27	.800	.056	.919	.789
$EDRNet_{20}$ [1]	D.L.D.	39.31	**.877**	**.024**	**.964**	**.872**
$DACNet_{21}$ [52]	D.L.D.	98.39	.875	**.024**	**.964**	.870
$EMINet_{21}$ [53]	D.L.D.	99.13	**.877**	**.024**	.962	.867
Ours	D.L.D.	189.64	**.886**	**.024**	**.965**	**.888**

In addition, the PR curves of all the compared methods are plotted in Fig. 6. We highlight the top five methods and darken the PR curves of other methods to distinguish the PR curves of different methods easily. It is obvious that the PR curve of our method is the closest one to the top right corner, which indicates the best performance.

2) Visual Comparison: In Fig. 7, we show the saliency maps of all the compared methods and our TSERNet. The first three rows are inclusions, the middle three rows are patches, and the last three rows are scratches. Obviously, our saliency maps are very close to the ground truths and are the best of all the methods. For the traditional methods such as (c) RCRR and (d) 2LSG, they can only vaguely predict the approximate position of the salient defects, but cannot outline their edge details, resulting in unsatisfactory prediction accuracy. The retrained deep-learning-based SOD methods for natural images can predict salient defects more accurately. However, in their predicted saliency maps, there are cases where the background is predicted as the salient region and the edge of

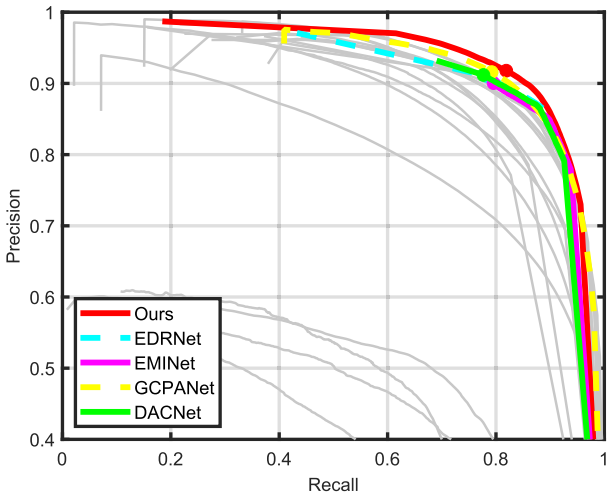

Fig. 6. Quantitative comparison in terms of PR curve on SD-saliency-900. We highlight the top five methods.

salient defects can only be roughly outlined. Compared with EDRNet, DACNet, and EMINet designed for defect images, the saliency maps of our TSERNet are visually better than theirs. EDRNet, DACNet, and EMINet do not perform as well as our TSERNet in the face of more complex backgrounds, such as the case of the sixth row, in which salient defects and background are similar and mixed together. EDRNet, DACNet, and EMINet generate the similar prediction results, and they cannot distinguish salient defects from the background in the right half of the defect image and wrongly highlight the background regions. Benefiting from the deployment of the edge reuse and reverse attention, our TSERNet can separate salient defects from complex backgrounds and generate more precise saliency maps.

C. Ablation Studies

In this section, we present ablation studies on the SD-saliency-900 dataset to demonstrate the effectiveness of components of our TSERNet. To be more intuitive, we divide the ablation studies into the following two parts: 1) the individual contribution of each component in TSERNet and 2) the importance of each part in the EFBI module and ERM. For all the ablation studies, we use the same parameter configuration as in Section IV-A.

1) Individual Contribution of Each Component in TSERNet: We provide three variants to evaluate the contribution of sub-net, EFBI module, and ERM. The experimental results are presented in Table II.

For the first variant w/o sub-net, we remove the sub-net and regard $\boldsymbol{S}_{\text{initial}}$ as the network output to prove the effectiveness of our sub-net. As shown in the first row of Table II, we can observe that S_α and $F_\beta^{\max}$ of w/o sub-net have a 3.39% and 4.73% decrease, respectively, as well as $\mathcal{M}$ has a 20.83% increase without the sub-net. This shows that sub-net has a strong ability to refine the local details, and our two-stage architecture is reasonable.

For the second variant w/o EFBI, we only remove the EFBI modules from the primary net and adopt the concatenation-convolution operation to fuse various features to prove the importance of the EFBI module. The performance of w/o EFBI shows that the simple fusion operation makes S_α and $F_\beta^{\max}$, respectively, drop by 1.69% and 3.04% as well as a 12.5% increase in $\mathcal{M}$.

TABLE II

ABLATION STUDY ON EVALUATING THE INDIVIDUAL CONTRIBUTION OF EACH COMPONENT IN TSERNET. BEST RESULT IN EACH COLUMN IS GIVEN IN **BOLD**

No.	Models	SD-saliency-900 [50]		
		$S_\alpha \uparrow$	$\mathcal{M} \downarrow$	$F_\beta^{\max} \uparrow$
1	*w/o sub-net*	.856	.029	.846
2	*w/o EFBI*	.871	.027	.861
3	*w/o ERM*	.876	.026	.879
4	**Ours**	**.886**	**.024**	**.888**

For the third variant w/o ERM, we only remove the ERM from the sub-net and adopt the concatenation-convolution operation to fuse various features to prove the importance of ERM. Without ERM, there are a 1.13% and 1.01% decrease in S_α and $F_\beta^{\max}$ as well as a 8.33% increase in $\mathcal{M}$ for w/o ERM. This proves the superiority of our ERM compared with directly using the concatenation-convolution operation for information fusion.

2) Importance of Each Part in EFBI Module and ERM: First, we conduct ablation studies to evaluate the importance of BIG-Part and FIG-Part of the EFBI module and provide two variants. The first variant named w/o FIG removes FIG-Part from the EFBI module, and the second variant named w/o BIG removes BIG-Part from the EFBI module. We show their structures in Fig. 8 and report their performance in Table III. We can observe that the performance of these two variants is almost identical and is worse than our complete method, which proves the effectiveness and necessity of FIG-Part and BIG-Part in the EFBI module. Based on the foreground information and edge information, the addition of background information makes S_α and $F_\beta^{\max}$ of w/o BIG, respectively, increase by 1.03% and 2.54% as well as $\mathcal{M}$ decrease by 7.69%. This is due to the edge information that can better distinguish and fuse the foreground and background information in the EFBI module.

Second, we conduct ablation studies to evaluate the importance of two types of edge information in ERM and provide two variants. The first variant named w/o self-edge removes the edge information carried by $\boldsymbol{f}_{s,e}^{j}$, and the second variant named w/o $\boldsymbol{S}_{\text{edge}}$ removes $\boldsymbol{S}_{\text{edge}}$. We also show their structures in Fig. 8 and report their performance in Table III. We can observe that the addition of $\boldsymbol{S}_{\text{edge}}$ improves S_α and $F_\beta^{\max}$ of w/o $\boldsymbol{S}_{\text{edge}}$ by 0.68% and 1.60%, respectively. The addition of edge information carried by $\boldsymbol{f}_{s,e}^{j}$ improves S_α and $F_\beta^{\max}$ of w/o self-edge by 1.14% and 1.02%, respectively. This means that our complete method which fuses these two types of edge information can effectively outline the details of salient defects.

D. Failure Case Analysis

Although our TSERNet shows excellent performance on the SD-saliency-900 dataset, it still performs unsatisfactorily in

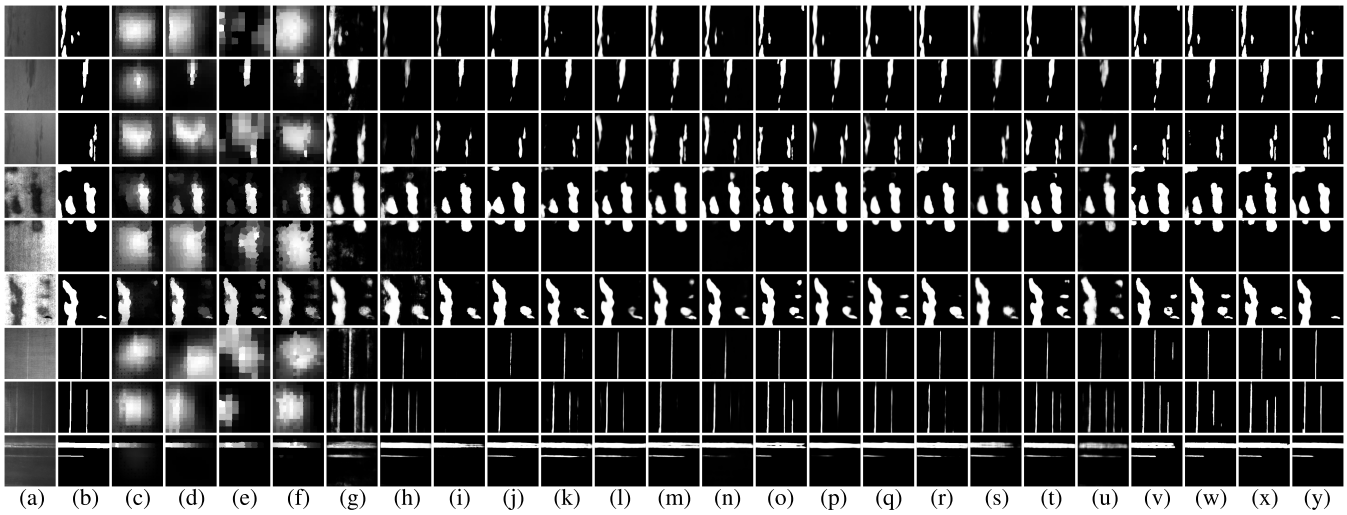

Fig. 7. Visual comparisons with 22 state-of-the-art methods. (a) Defect image. (b) GT. (c) RCRR [73]. (d) 2LSG [11]. (e) BC [12]. (f) SMD [74]. (g) PFANet [75]. (h) NLDF [76]. (i) DSS [16]. (j) R3Net [14]. (k) BMPM [31]. (l) PoolNet [13]. (m) PiCANet [37]. (n) CPD [77]. (o) BASNet [42]. (p) EGNet [43]. (q) GCPANet [32]. (r) MINet [34]. (s) SAMNet [78]. (t) SUCA [79]. (u) ENFNet [48]. (v) EDRNet [1]. (w) DACNet [52]. (x) EMINet [53]. (y) TSERNet (Ours). Zoom-in for the best view.

TABLE III

ABLATION STUDY ON EVALUATING THE IMPORTANCE OF EACH PART IN EFBI MODULE AND ERM. BEST RESULT IN EACH COLUMN IS GIVEN IN **BOLD**

No.	Models	SD-saliency-900 [50]		
		$S_\alpha \uparrow$	$\mathcal{M} \downarrow$	$F_\beta^{\max} \uparrow$
1	*w/o FIG*	.876	.026	.867
2	*w/o BIG*	.877	.026	.866
3	*w/o self-edge*	.876	.026	.879
4	*w/o* $\boldsymbol{S}_{edge}$	.880	.025	.874
5	**Ours**	**.886**	**.024**	**.888**

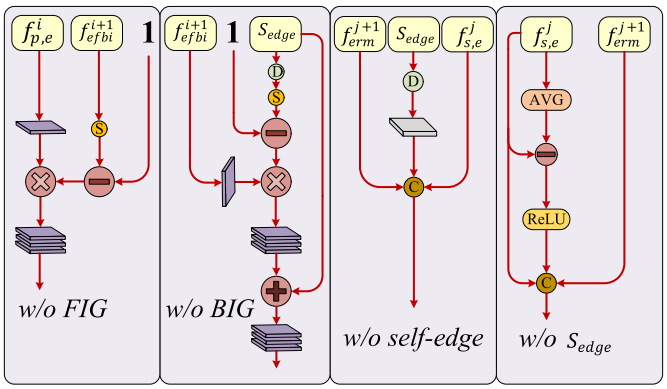

Fig. 8. Structures of four variants of the EFBI module and ERM.

some cases. We present three types of representative failure cases in Fig. 9. The first type is shown in the first two columns of Fig. 9. In this case, we can observe that GTs are labeled a bit inaccurately, while the saliency maps of our method are actually more accurate. The second type is shown in the middle two columns of Fig. 9. Our method does not discriminate

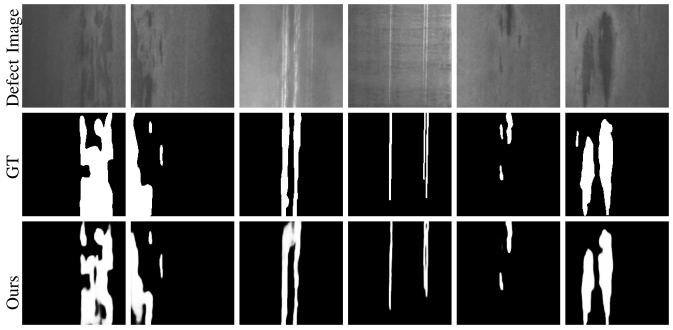

Fig. 9. Three types of failure cases of our method.

very close defects well. The last type is shown in the last two columns of Fig. 9. Our method may miss some tiny defects.

V. CONCLUSION

In this article, we propose a novel TSERNet to address SOD in strip steel surface defect images. In the first stage (i.e., primary net) of our TSERNet, by taking advantage of edge information, we fuse the foreground information and background information in an innovative way and distinguish them well in features, generating an initial saliency map. In the second stage (i.e., sub-net) of our TSERNet, we reuse the edge information generated from the first stage and explore another edge information contained in features to refine the local details of the initial saliency map, generating a purified saliency map. We fuse the two saliency maps generated from the above stages and obtain the final saliency map. In addition, two kinds of loss functions are used to train our TSERNet in a deep supervision manner during the training phase. Extensive experiments on the SD-saliency-900 dataset prove that our TSERNet achieves the best performance compared with 22 state-of-the-art methods.

REFERENCES

[1] G. Song, K. Song, and Y. Yan, "EDRNet: Encoder–decoder residual network for salient object detection of strip steel surface defects," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9709–9719, Dec. 2020.

[2] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.

[3] X. Cheng and J. Yu, "RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[4] Q. Luo *et al.*, "Waterdrop removal from hot-rolled steel strip surfaces based on progressive recurrent generative adversarial networks," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[5] T. Wang, J. Zhao, Q. Liu, and W. Wang, "Granular-based multilayer spatiotemporal network with control gates for energy prediction of steel industry," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.

[6] C. Fu, J. Xu, F. Lin, F. Guo, T. Liu, and Z. Zhang, "Object saliency-aware dual regularized correlation filter for real-time aerial tracking," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8940–8951, Dec. 2020.

[7] X. Zhang and X. Wu, "Attention-guided image compression by deep reconstruction of compressive sensed saliency skeleton," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13354–13364.

[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, R. Shi, and W. Wei, "Constrained fixation point based segmentation via deep neural network," *Neurocomputing*, vol. 368, pp. 180–187, Nov. 2019.

[10] L. He and W. Liu, "Guided saliency feature learning for person re-identification in crowded scenes," in *Proc. ECCV*, 2020, pp. 357–373.

[11] 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.

[12] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. IEEE CVPR*, Jun. 2014, pp. 2814–2821.

[13] 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 CVPR*, Jun. 2019, pp. 3917–3926.

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

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*. New York, NY, USA: Springer, Oct. 2015, pp. 234–241.

[16] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3203–3212.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Jun. 2016, pp. 770–778.

[19] J. Hu, L. Shen, and G. Sun, "Squeeze- and-excitation networks," in *Proc. IEEE CVPR*, Jun. 2018, pp. 7132–7141.

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

[21] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[22] F. Liu and M. Gleicher, "Region enhanced scale-invariant saliency detection," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 1477–1480.

[23] X. Li, Y. Li, C. Shen, A. Dick, and A. V. D. Hengel, "Contextual hypergraph modeling for salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3328–3335.

[24] K. Shi, K. Wang, J. Lu, and L. Lin, "Pisa: Pixelwise image saliency by aggregating complementary appearance contrast measures with spatial priors," in *Proc. IEEE CVPR*, Jun. 2013, pp. 2115–2122.

[25] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 914–921.

[26] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Proc. ECCV*, 2012, pp. 29–42.

[27] Y. Niu, C. Su, and W. Guo, "Salient object segmentation based on superpixel and background connectivity prior," *IEEE Access*, vol. 6, pp. 56170–56183, 2018.

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

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

[30] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 202–211.

[31] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, "A bi-directional message passing model for salient object detection," in *Proc. IEEE CVPR*, Jun. 2018, pp. 1741–1750.

[32] Z. Chen, Q. Xu, R. Cong, and Q. Huang, "Global context-aware progressive aggregation network for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, Apr. 2020, pp. 10599–10606.

[33] A. K. Gupta, A. Seal, P. Khanna, A. Yazidi, and O. Krejcar, "Gated contextual features for salient object detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[34] 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. 9413–9422.

[35] A. K. Gupta, A. Seal, P. Khanna, E. Herrera-Viedma, and O. Krejcar, "ALMNet: Adjacent layer driven multiscale features for salient object detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–14, 2021.

[36] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *Proc. IEEE CVPR*, Jun. 2018, pp. 714–722.

[37] N. Liu, J. Han, and M.-H. Yang, "PiCANet: Learning pixel-wise contextual attention for saliency detection," in *Proc. IEEE CVPR*, Jun. 2018, pp. 3089–3098.

[38] S. Chen, "Reverse attention-based residual network for salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 3763–3776, 2020.

[39] L. Zhang, Q. Zhang, and R. Zhao, "Progressive dual-attention residual network for salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Apr. 1, 2022, doi: 10.1109/TCSVT.2022.3164093.

[40] J.-J. Liu, Q. Hou, Z.-A. Liu, and M.-M. Cheng, "PoolNet+: Exploring the potential of pooling for salient object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 4, 2022, doi: 10.1109/TPAMI.2021.3140168.

[41] X. Wang *et al.*, "Weakly-supervised instance segmentation via class-agnostic learning with salient images," in *Proc. IEEE CVPR*, Jun. 2021, pp. 10220–10230.

[42] 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.

[43] 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.

[44] Z. Wu, L. Su, and Q. Huang, "Stacked cross refinement network for edge-aware salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7264–7273.

[45] J. Su, J. Li, Y. Zhang, C. Xia, and Y. Tian, "Selectivity or invariance: Boundary-aware salient object detection," in *Proc. IEEE ICCV*, Oct. 2019, pp. 3799–3808.

[46] W. Guan, T. Wang, J. Qi, L. Zhang, and H. Lu, "Edge-aware convolution neural network based salient object detection," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 114–118, Jan. 2019.

[47] Z. Wu, L. Su, and Q. Huang, "Decomposition and completion network for salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 6226–6239, 2021.

[48] Z. Tu, Y. Ma, C. Li, C. Li, J. Tang, and B. Luo, "Edge-guided non-local fully convolutional network for salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 582–593, Feb. 2021.

[49] S. Zhou, S. Wu, H. Liu, Y. Lu, and N. Hu, "Double low-rank and sparse decomposition for surface defect segmentation of steel sheet," *Appl. Sci.*, vol. 8, no. 9, p. 1628, Sep. 2018.

[50] G. Song, K. Song, and Y. Yan, "Saliency detection for strip steel surface defects using multiple constraints and improved texture features," *Opt. Lasers Eng.*, vol. 128, May 2020, Art. no. 106000.

[51] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7448–7458, Dec. 2020.

[52] X. Zhou *et al.*, "Dense attention-guided cascaded network for salient object detection of strip steel surface defects," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.
[53] X. Zhou, H. Fang, X. Fei, R. Shi, and J. Zhang, "Edge-aware multi-level interactive network for salient object detection of strip steel surface defects," *IEEE Access*, vol. 9, pp. 149465–149476, 2021.
[54] 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.
[55] 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.
[56] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.
[57] C.-Y. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," *Artif. Intell. Statist.*, vol. 38, pp. 562–570, May 2015.
[58] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 448–456.
[59] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. AISTATS*, vol. 15, Apr. 2011, pp. 315–323.
[60] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," in *Proc. AISTATS*, vol. 134, 2005, pp. 19–67.
[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] 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, pp. 1–13, 2022.
[63] 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, pp. 1–12, 2022.
[64] 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.*, early access, Apr. 13, 2022, doi: 10.1109/TCYB.2022.3162945.
[65] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, vol. 2, Nov. 2004, pp. 1398–1402.
[66] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proc. ACM MM*, 2016, pp. 516–520.
[67] K. Zhao, S. Gao, W. Wang, and M.-M. Cheng, "Optimizing the F-measure for threshold-free salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8849–8857.
[68] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NIPS*, Dec. 2019, pp. 8024–8035.
[69] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, May 2015, pp. 1–15.
[70] 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.
[71] 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.
[72] 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.
[73] 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.
[74] 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.
[75] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3085–3094.
[76] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P.-M. Jodoin, "Non-local deep features for salient object detection," in *Proc. IEEE CVPR*, Jul. 2017, pp. 6609–6617.
[77] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3907–3916.
[78] 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.
[79] 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.

Chengjun Han received the B.E. degree from Shanghai University, Shanghai, China, in 2020, where he is currently pursuing the M.S. degree with the School of Communication and Information Engineering.

His research interests include deep learning and saliency detection.

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 currently holds a post-doctoral position with the School of Communication and Information Engineering, Shanghai University. 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, 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 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*.