Measurement 199 (2022) 111429

Contents lists available at ScienceDirect

Measurement

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

Cross-Scale Edge Purification Network for salient object detection of steel defect images

Tuo Ding, Gongyang Li[1], Zhi Liu [*], Yike Wang

Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China
School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China

ARTICLE INFO

Keywords:
Salient object detection
Steel defect images
Cross-scale calibration
Edge information

ABSTRACT

Salient object detection has achieved great success in natural scene images, but there is big room for exploration in steel defect images (SDIs). Moreover, the unique characteristic of SDIs makes salient object detection in SDIs (SDI-SOD) a challenging task, and many representative methods for natural scene images struggle to get satisfactory results in SDI-SOD. Existing SDI-SOD methods usually ignore the edge information, and focus only on enhancing the feature interaction of different layers. To this end, we propose a specialized Cross-Scale Edge Purification Network (CSEPNet) to explore the correlations of features at different scales for SDI-SOD. To be specific, our CSEPNet is based on the general encoder–decoder architecture. First, we adopt a generic Convolutional Block Attention Module (CBAM) to refine and enhance the features from the encoder. Then, we propose the Cross-Scale Calibration Module (CSCM) and Cross-Scale Feature Interweaving Module (CSFIM) to capture the relationship of inter-layer features and intra-layer features, respectively. In CSCM, adjacent features of different scales are effectively calibrated by each other and are re-calibrated by their collective weight map to generate informative features. In CSFIM, two branches of features interweave and fuse with their corresponding edge information purified to generate robust features. Besides, we employ deep supervision and use a hybrid loss to guide the training process. We perform comprehensive experiments on the public SD-saliency-900 dataset, and demonstrate that our method is superior to 26 state-of-the-art methods, including both traditional and CNN-based ones. The code and results of our method are available at https://github.com/showmaker369/CSEPNet.

1. Introduction

Visual attention is a unique signal processing mechanism of human vision and can automatically capture the regions that attract humans attention most. The salient object detection (SOD) task in computer vision aims to model this procedure. SOD has been widely used in other fields, such as image and video segmentation [1–3], object tracking [4], image quality assessment [5,6], *etc*. With the fast development of deep learning [7,8], CNN-based methods [9–12] meet the demand of fast inspection of steel defect for both accuracy and efficiency, and begin to replace the manual inspection operation. In this paper, we focus on SOD in steel defect images. Specifically, steel defect images (SDIs) refer to images of steel with some defects such as scratches, inclusion, patches, *etc*. As shown in Fig. 1, most of SDIs have a dark background and are often accompanied by noise interference. At the same time, the defect areas are rich in structural information.

Recently, SOD in steel defect images (SDI-SOD) [13–15] receive more attention, which benefits for accurately locating defects. Deep-learning based methods have significantly boosted the accuracy of SOD in nature scene images (NSI-SOD). Many meaningful strategies, such as multi-scale/layer feature aggregation [16,17], FPN based structure [18,19], self-attention mechanism [20], and global contextual information guided structure [21,22] have been proposed. However, directly using these NSI-SOD methods for steel defect images cannot get satisfactory results due to the unique characteristics of SDIs. As shown in Fig. 2, two representative CNN-based NSI-SOD methods, *i.e.*, PA-KRN [19] and R3Net [23], fail to highlight the salient defects.

As a result of the achievements in NSI-SOD, NSI-SOD methods largely inspire effective deep learning based solutions for SDIs, and some representative methods for SDI-SOD emerge. For example, Song et al. [14] proposed a residual network based on encoder–decoder framework and used the refinement module to refine the feature from

[*] Corresponding author at: Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China.
E-mail addresses: tding97@163.com (T. Ding), ligongyang@shu.edu.cn (G. Li), liuzhisjtu@163.com (Z. Liu), kevinwangyike@qq.com (Y. Wang).

[1] The first two authors contributed equally.

https://doi.org/10.1016/j.measurement.2022.111429
Received 16 March 2022; Received in revised form 13 May 2022; Accepted 31 May 2022
Available online 7 June 2022

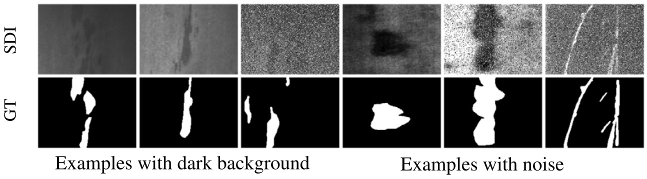


Fig. 1. Representative examples in steel defect images. GT is the ground truth.

the decoder network. Zhou et al. [15] used the cascaded feature integration module to fuse multi-branch features, and then exploited the decoder to progressively integrate multi-level deep features with the guidance of the dense attention structure. Although the above CNN-based SDI-SOD methods have made some progress, they cannot generate clear saliency maps when dealing with some challenging scenes. As shown in Fig. 2, EDRNet [14] and DACNet [15] fail to produce saliency maps with fine-grained detail and structural information. Inspired by above observations, we propose a novel Cross-Scale Edge Purification Network (CSEPNet) to highlight salient defects and maintain the edge information in SDIs. Our CSEPNet is based on the encoder–decoder architecture, and consists of one generic feature enhancement module CBAM [24] and our two proposed modules, *i.e.*, Cross-Scale Calibration Module (CSCM) and the Cross-Scale Feature Interweaving Module (CSFIM). To be specific, CSCM is proposed for inter-layer features, and CSFIM is proposed for intra-layer features. The former one focuses on calibrating the cross-scale features via attention mechanism. The latter one interweave the intra-layer features of two scales accompanied by an edge purification process, and then explores the complementarity between them.

Concretely, for each layer of our backbone, the basic features are firstly refined by CBAM. Features of two adjacent layers (*i.e.*, inter-layer features) are then fused in the CSCM, where features of two adjacent layers are calibrated by each other and re-calibrated by their concatenated mode. Then, the output features of CSCM are fed to the CSFIM to highlight salient defects in a cross-scale manner with the edge information purified by the contrast enhancement unit, and the two-scale refined features will be calibrated by each other in the variant of CSCM. In this way, we aggregate adjacent and intra-layer features in our CSEPNet to progressively infer the final saliency map. Notably, we take the deep supervision strategy and a comprehensive loss for network training. Experiments on SD-saliency-900 dataset [13] show that our specialized CNN-based SDI-SOD method achieves the best performance as compared with 26 state-of-the-art methods, and generates accurate saliency maps, as shown in Fig. 2.

Our main contributions are summarized as follows:

- We propose a novel *Cross-Scale Edge Purification Network* (CSEPNet) for SDI-SOD based on the encoder–decoder architecture. Our CSEPNet explores the complementarity of inter-layer and intra-layer features to progressively produce a precise saliency map.
- We propose a *Cross-Scale Calibration Module* to effectively capture the relationship of features from different layers so as to calibrate cross-scale inter-layer features flexibly and produce informative features.
- We propose a *Cross-Scale Feature Interweaving Module* to explore the complementarity of the intra-layer features. In this module, we present two-scale representation of the intra-layer features, and interweave them with the purified edge information in a cross-scale manner. Then, the refined two-scale intra-layer features are effectively aggregated by the cross-scale weighting module (a variant of CSCM).

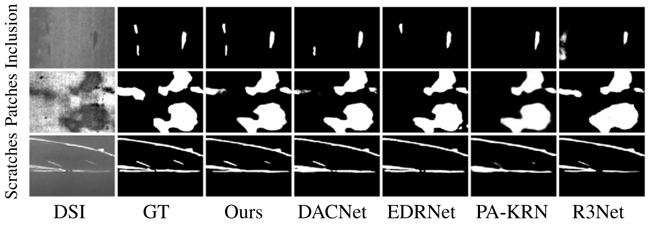


Fig. 2. Saliency maps generated by our method and four state-of-the-art methods, including DACNet [15], EDRNet [14], PA-KRN [19], and R3Net [23], on three challenging SDI scenes, *i.e.*, inclusion, patches, and scratches.

2. Related work

In this section, we introduce the SOD methods for NSIs and steel defect images. For the former one, we introduce both traditional and CNN-based methods and for the latter, we mainly focus on CNN-based methods.

2.1. Salient object detection in NSIs

(1) Traditional methods. As an early enlightening work, Itti et al. [25] proposed a model based on biologically-plausible architecture which parallelly uses human visual perception related factors like color, intensity and orientation to obtain the saliency maps. Similar to [25], Valenti et al. [26] used saliency features like isophotes, color distinctiveness and curvedness to produce initial saliency maps, and then linearly combined three initial saliency maps to produce the final result. Some methods [27,28] used matrix recovery to detect salient regions. For example, in [27], Shen et al. represented the image as a low-rank matrix added with sparse noise in a learned space, where saliency regions are represented by the sparse noise, and used the robust PCA method [29] to identify salient regions. In [30], a novel superpixel generation algorithm adapted self-adjustable distance measures to detect the amount of dissimilarity between the data points and performed well on the benchmark. Wei et al. [31] proposed a novel intuitive geodesic saliency measure method based on the contrast and two background priors (*i.e.*, boundary and connectivity) to produce the precise saliency map. Later, in [32], Ding et al. integrated the advantages of cellilar and Gauss filtering to calculate the background-based saliency map, and then fused it with another refined saliency map to get the final salieny map. In [33], Zhou et al. obtained two complementary maps based on the foreground and background seeds, and then generated the salieny map by the diffusion process. To some extent, traditional methods do not achieve satisfactory performance, but provide some inspiring ideas and experiences for CNN-based methods.

(2) CNN-based methods. With a better representation ability, CNN-based methods [34–36] break through the limitations of traditional methods [37,38] and push the performance to a higher level. For instance, Wu et al. [39] designed an partial decoder and a holistic attention module for features from higher layers to progressively produce initial saliency map and the final saliency map. Besides, to counter the problem of scale variation, a large number of CNN-based methods [16–18,21,39–42] take different strategies. For instance, Gupta et al. [43] used an attention-based module to integrate features from adjacent layers in a rational way, and then fused multiscale features in a top-down manner. Pang et al. [17] integrated multi-level features from adjacent layers and used a self-interaction module for better feature interaction. Zhao et al. [40] used features from the present encoder block and the previous decoder block to generate a gate so as to guide the two-stream feature fusion, extracting stabler features. Liu et al. [18] proposed a module based on FPN structure and used the feature aggregation module to allow the global guidance information to be delivered to feature maps at different pyramid levels, enlarging the receptive field of the

whole network. Hybrid loss is widely used in NSI-SOD. Qin et al. [44] proposed a U-Net [45] like densely supervised prediction module and use a hybrid loss function that consists of three different losses for the network training process. Moreover, to address the dependency of SOD models on high quality images, Zhou et al. [20] designed a multi-type self-attention module and proposed a network to detect salient object in the degraded images. Zhao et al. [35] introduced the edge information to guide the saliency map prediction process. In [36], a gate-based contextual information extraction model was proposed to control the information flows between different branches. For more details, please refer to the survey [46].

Although directly using NSI-SOD methods on steel defect images cannot produce satisfactory results, these methods still provide some worthwhile sights for SOD in steel defect images.

2.2. Camouflaged object detection

The characteristic of industrial images is that the foreground salient regions have a high similarity with their background and are always accompanied by ambiguous boundaries, which is similar to the characteristic of the camouflaged object detection (COD) images. Here, we introduce some classic COD methods. In [47], Fan et al. proposed a two-stage network for COD. In [48], Ji et al. designed an effective network, which incorporates diverse priors and produced comprehensive information on the basis of reversible re-calibration unit for COD. Mei et al. [49] designed a network based on a positioning module and a focus module to locate and identify target objects. To better analyze the attribute of the camouflage, Lv [50] proposed the first ranking based network which can simultaneously locate, segment, and rank camouflaged objects. Li [51] proposed an effective module to model the contradicting attributes of both COD and SOD tasks, and used an adversarial learning strategy for robust model training.

2.3. Salient object detection in steel defect images

Recently, some efforts are made on SOD in steel defect images. To address this task, Song et al. [13] constructed the first dataset, termed SD-saliency-900, for SOD in steel defect images, and proposed a saliency propagation algorithm which uses the generated label matrix and the results of multiple constraints to obtain a local diffusion function for defects detection. Moreover, influenced by deep learning, Song et al. [14] introduced the attention mechanism [24] into the encoder–decoder network to locate salient defect regions more efficiently. They adopted the prediction and refinement strategy, that is, first predicted a saliency map in an encoder–decoder network, and then refined the saliency map in a refinement network. However, they did not consider edge information, which is important for SOD of steel defect images. Therefore, the boundaries of the generated saliency maps are vague. Zhou et al. [15] employed three convolutional branches to extract multi-resolution features, and directly adopted the concatenation operation to aggregate them in a cascaded feature integration unit. Then, they exploited the decoder to progressively integrate multi-level deep features with the guidance of the dense attention structure. We believe the feature fusion strategy of [15] is rough and simple, and cannot effectively explore the feature interaction and generate valuable features. In addition, Zhou et al. also ignored the edge information. Besides, the above two CNN-based methods only focused on enhancing the feature interaction of different layers (i.e., inter-layer features), and ignored the feature interaction of intra-layer features.

Similar to SOD, some researchers focus on segmentation and object detection in steel defect images. Huang et al. [52] proposed a U-shape network to segment steel defects, which uses the depth-wise separable convolution to reduce parameters and the multi-scale module to extract multi-scale context information. To address the problem of big differences between intra-class surface defects, Dong et al. [53] proposed a network based on pyramid feature fusion and global context attention, termed PGA-Net, for surface defect segmentation. He et al. [54] first fused features from baseline into multi-level features and then fed the more representative features into a region proposal network to determine the location and class of defects. In [55], Tang et al. proposed an end-to-end network for defect detection. The network first embeds the attention mechanism module into the backbone to reduce interference of image noise in features, and then uses the multi-scale max-pooling module to increase the receptive field of the network before generating proposals.

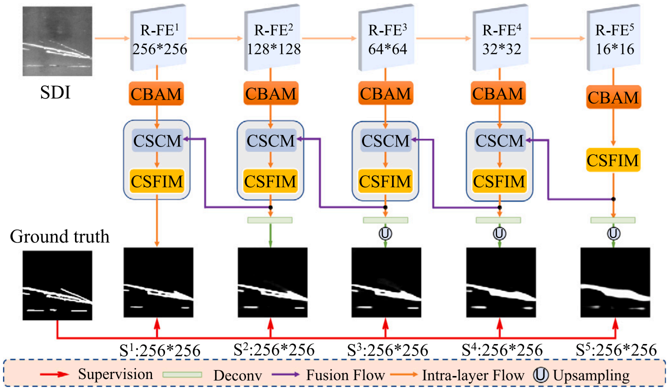


Fig. 3. The architecture of our CSEPNet, which is based on the general encoder–decoder architecture. We use the classic VGG-16 [65] as the feature extractor, and its input is a SDI with a size of 256 × 256 × 3. Features from five encoder blocks are elaborately refined by the CBAM. CSFIM is in charge of enhancing features with the edge information, and CSCM is in charge of coordinating features from adjacent layers to progressively aggregate multi-scale features. For the training phase, we employ deep supervision to the predicted saliency maps from CSFIM, and we adopt a hybrid loss, including BCE loss, IOU loss, CEL loss, and SSIM loss.

The above models are oriented in steel defects, however, there are defects in diverse materials in different industrial environments. Therefore, many works target on locating surface defects on other materials [56–59]. For example, in [60], Wei et al. used semantic-aware network and texture-aware network to capture semantic information and texture information, respectively, and then the diverse features are integrated for tire defects detection. In [61], Zhang et al. effectively resolved three difficulties in no-service rail surface defects segmentation. In [62], Wang et al. designed a self-attention module to coordinate the dependencies between side-output features to generate detail-enriched fabric defect detection results. Aslam et al. [63] designed a U-Net based encoder–decoder network for metal defect detection, which uses a combination of binary cross-entropy loss and dice loss as the loss function. Tabernik et al. [64] used a two-stage network for plastic surface detection. At the first stage, a segmentation network performs the pixel-wise localization of surface defects, and then an additional network that is built on top of the segmentation network uses both the segmentation output and the features of the segmentation network to generate the result. Similar to steel defect images, fabric defect images are always with interference of background noise while tire defect images are captured in insufficient illumination environments. For images of the plastic embedding in electrical commutators, the defects are also with dark background and have abundant structural information.

Inspired by the above works, our CSEPNet adopts the encoder–decoder structure as the backbone, and uses the popular attention module [24] for straightforward feature enhancement on different backbone layers. Moreover, we elaborately design the Cross-Scale Calibration Module to coordinate the dependencies between features of different backbone layers, and design the Cross-Scale Feature Interweaving Module to balance features within one backbone layer.

3. Proposed model

In this section, we elaborate our CSEPNet. We first introduce the overview of our CSEPNet in Section 3.1. In Sections 3.2 and 3.3, we give a detailed introduction of our Cross-Scale Calibration Module (CSCM) and Cross-Scale Feature Interweaving Module (CSFIM), respectively. At the end of this section, we clarify the loss function in Section 3.4.

3.1. Network overview

Many representative SOD methods [14,66–69] are based on the encoder–decoder architecture. Therefore, we build our CSEPNet on this popular architecture, as shown in Fig. 3. The backbone of our CSEPNet is the popular VGG-16 [65] pre-trained on ImageNet [70], and we remove the last four layers (*i.e.*, one max-pooling layer and three fully connected layers) of VGG-16 for feature extraction. Taking a SDI $\boldsymbol{I} \in \mathbb{R}^{256\times256\times3}$ as input, we denote the five blocks in feature extractor as R-FE^i and their output basic features as $\boldsymbol{f}_{\mathrm{e}}^{i}$, where $i \in \{1,2,3,4,5\}$ is the block index. Then, $\boldsymbol{f}_{\mathrm{e}}^{i}$ is refined by the CBAM to obtain the refined feature $\boldsymbol{f}_{\mathrm{r}}^{i}$. Specifically, $\boldsymbol{f}_{\mathrm{r}}^{5}$ is fed to a CSFIM, generating the output feature of CSFIM, $\boldsymbol{f}_{\mathrm{csfi}}^{5}$. Then, $\boldsymbol{f}_{\mathrm{csfi}}^{5}$ is integrated with the previous $\boldsymbol{f}_{\mathrm{r}}^{4}$ in a CSCM to calibrate them by each other, generating the output feature of CSCM, $\boldsymbol{f}_{\mathrm{csc}}^{4}$. As shown in Fig. 3, following such a data flow, we can accurately locate the salient defects and generate the predicted saliency maps. Notably, we employ five deconvolutional layers in our CSEPNet to upsample the predicted saliency maps to the resolution of GT for supervision. We choose classic binary cross-entropy (BCE) loss, intersection-over-union (IoU) loss, consistency-enhanced (CEL) loss, and patch-level structural similarity (SSIM) loss as the total loss to highlight the foreground region and improve structural information of salient objects for effective network training.

3.2. Cross-scale calibration module

The interaction between features of different scales plays an important role in the field of SOD. Since features from low layers have more detail information and features from high layers have more semantic information, the coordination of features at diverse scale is meaningful. Different from the previous SOD work [21] that simply upsamples low-scale features and multiply it with high-scale features, we propose a Cross-Scale Calibration Module to fuse features of different scales more effectively. Our CSCM can integrate features from adjacent layers and enhance the representation of salient defects in its output features. We illustrate the structure of CSCM in Fig. 4.

In CSCM, the input features are $\boldsymbol{f}_{\mathrm{r}}^{i}$ from low-level layer and $\boldsymbol{f}_{\mathrm{csfi}}^{i+1}$ from high-level layer. First, we upsample $\boldsymbol{f}_{\mathrm{csfi}}^{i+1}$ to the size of $\boldsymbol{f}_{\mathrm{r}}^{i}$, and then apply a convolutional layer with 3 × 3 kernel size on $\boldsymbol{f}_{\mathrm{csfi}}^{i+1}$ for channel adjustment, causing the two input features have the same size. After that, we perform the spatial attention [24] on them to get two distinctive attention maps $\boldsymbol{A}_{\mathrm{low}}^{i}$ and $\boldsymbol{A}_{\mathrm{high}}^{i}$ as follows:

$$\boldsymbol{A}_{\mathrm{low}}^{i} = \mathrm{SA}(\boldsymbol{f}_{\mathrm{r}}^{i}), \tag{1}$$

$$\boldsymbol{A}_{\mathrm{high}}^{i} = \mathrm{SA}(\boldsymbol{f}_{\mathrm{csfi}}^{i+1}), \tag{2}$$

where $\mathrm{SA}(\cdot)$ means spatial attention, which is implemented by a global average-pooling and global max pooling along channel axis, a convolutional layer and a sigmoid activation function.

To obtain more comprehensive features, we perform the cross-scale calibration operation on $\boldsymbol{f}_{\mathrm{r}}^{i}$ from low-level layer and $\boldsymbol{f}_{\mathrm{csfi}}^{i+1}$ from high-level layer. Concretely, we adopt $\boldsymbol{A}_{\mathrm{high}}^{i}$ to calibrate $\boldsymbol{f}_{\mathrm{r}}^{i}$, and then add $\boldsymbol{f}_{\mathrm{r}}^{i}$. Similarly, we also adopt $\boldsymbol{A}_{\mathrm{low}}^{i}$ to calibrate $\boldsymbol{f}_{\mathrm{csfi}}^{i+1}$, and then add $\boldsymbol{f}_{\mathrm{csfi}}^{i+1}$. Based on the above operations, we can obtain $\boldsymbol{f}_{\mathrm{low}}^{i}$ and $\boldsymbol{f}_{\mathrm{high}}^{i}$, which can be computed as:

$$\boldsymbol{f}_{\mathrm{low}}^{i} = \boldsymbol{A}_{\mathrm{high}}^{i} \otimes \boldsymbol{f}_{\mathrm{r}}^{i} \oplus \boldsymbol{f}_{\mathrm{r}}^{i}, \tag{3}$$

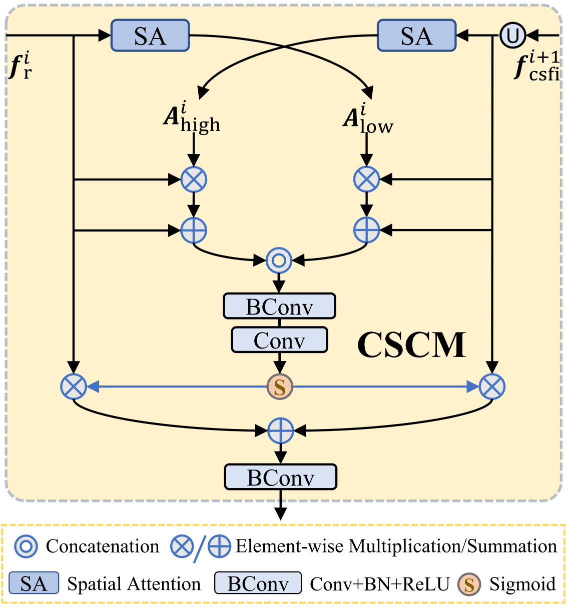

SA
SA
U
$\boldsymbol{f}_{\mathrm{r}}^{i}$
$\boldsymbol{f}_{\mathrm{csfi}}^{i+1}$
$\boldsymbol{A}_{\mathrm{high}}^{i}$
$\boldsymbol{A}_{\mathrm{low}}^{i}$
BConv
Conv
CSCM
S
BConv
Concatenation
Element-wise Multiplication/Summation
SA Spatial Attention
BConv Conv+BN+ReLU
S Sigmoid

Fig. 4. Illustration of the Cross-Scale Calibration Module.

$$\boldsymbol{f}_{\mathrm{high}}^{i} = \boldsymbol{A}_{\mathrm{low}}^{i} \otimes \boldsymbol{f}_{\mathrm{csfi}}^{i+1} \oplus \boldsymbol{f}_{\mathrm{csfi}}^{i+1}, \tag{4}$$

where $\otimes$ is element-wise multiplication, and $\oplus$ is element-wise summation.

To further enhance the interaction between the low-level and high-level features, we explore the useful information from them together. We concatenate $\boldsymbol{f}_{\mathrm{low}}^{i}$ and $\boldsymbol{f}_{\mathrm{high}}^{i}$, and employ two convolutional layers to fuse them, generating the collective weight map $\boldsymbol{W}^{i}$ as follows:

$$\boldsymbol{W}^{i} = s(\mathrm{conv}(\mathrm{BConv}(\boldsymbol{f}_{\mathrm{low}}^{i} \circledcirc \boldsymbol{f}_{\mathrm{high}}^{i}))), \tag{5}$$

where $s(\cdot)$ is the sigmoid activation function, $\mathrm{conv}(\cdot)$ is the convolutional layer, $\mathrm{BConv}(\cdot)$ consists of a convolutional layer with a batch normalization layer and ReLU activation function, and $\circledcirc$ is the cross-channel concatenation.

Finally, we adopt $\boldsymbol{W}^{i}$ to re-calibrate $\boldsymbol{f}_{\mathrm{r}}^{i}$ and $\boldsymbol{f}_{\mathrm{csfi}}^{i+1}$, and fuse them through a convolutional layer, generating the output feature of CSCM denoted to $\boldsymbol{f}_{\mathrm{csc}}^{i}$, as follows:

$$\boldsymbol{f}_{\mathrm{csc}}^{i} = (\boldsymbol{W}^{i} \otimes \boldsymbol{f}_{\mathrm{r}}^{i}) \oplus (\boldsymbol{W}^{i} \otimes \boldsymbol{f}_{\mathrm{csfi}}^{i+1}). \tag{6}$$

In this way, our CSCM uses features from adjacent layers to produce adaptive attention maps under the benefit of attention mechanism, and then calibrates each other in a cross-scale manner. At the same time, the collective weight map $\boldsymbol{W}^{i}$ is also generated through adaptive learning with two convolutional layers, which ensure that the module can re-optimize the input features in a more appropriate way to generate the output feature $\boldsymbol{f}_{\mathrm{csc}}^{i}$.

3.3. Cross-scale feature interweaving module

Cross-Scale Feature Interweaving Module plays a vital part in our CSEPNet. It generates two-scale representations from the input feature, and effectively enhances two branches of features in an interleaved manner. As illustrated in Fig. 5, our CSFIM can be divided into two stages, including the edge-based cross-scale feature interweaving stage and the cross-scale weighting stage. Specifically, in these two stages, there are two major components, termed contrast enhancement unit and cross-scale weighting module, and the latter one is a variant of CSCM. In the following, we elaborate CSFIM based on the above two stages.

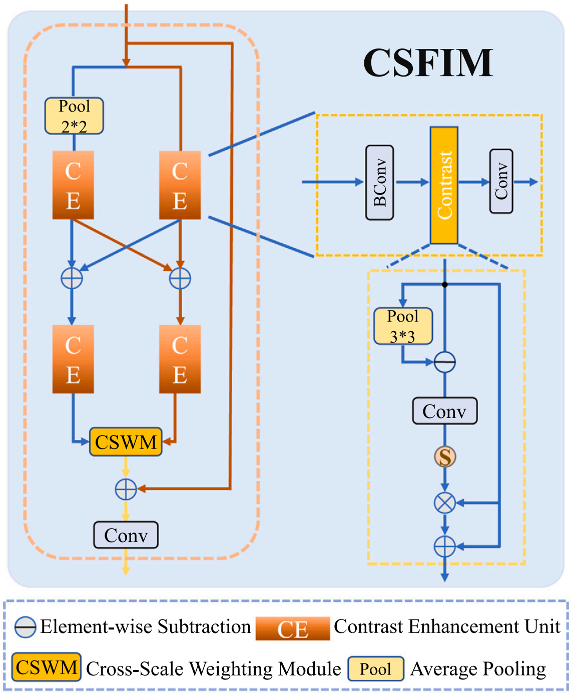

CSFIM
Pool 2*2
C E
C E
C E
C E
CSWM
Conv
BConv
Contrast
Conv
Pool 3*3
Conv
S
⊖ Element-wise Subtraction
CE Contrast Enhancement Unit
CSWM Cross-Scale Weighting Module
Pool Average Pooling

Fig. 5. Illustration of the Cross-Scale Feature Interweaving Module.

(1) Edge-based cross-scale feature interweaving stage. The input of CSFIM is $\boldsymbol{f}^i_{\mathrm{csc}}$ ($i \in \{1,2,3,4\}$) or $\boldsymbol{f}^i_{\mathrm{r}}$ (i =5). For simplicity, we define the input as $\boldsymbol{f}^i_{\text{csfi-in}}$. First, we generate a multi-scale representation of the input feature, that is, we downsample the input feature by the average pooling operation to obtain a low-resolution feature, and meanwhile keep the original-resolution feature. Subsequently, the two features are improved based on the edge information in the Contrast Enhancement (CE) unit, generating the low-resolution enhanced features $\boldsymbol{f}^i_{\mathrm{ls}}$ and the original-resolution enhanced features $\boldsymbol{f}^i_{\mathrm{os}}$ as follows:

$$\boldsymbol{f}^i_{\mathrm{ls}} = \mathrm{CE}(\mathrm{Avg}(\boldsymbol{f}^i_{\text{csfi-in}})), \tag{7}$$

$$\boldsymbol{f}^i_{\mathrm{os}} = \mathrm{CE}(\boldsymbol{f}^i_{\text{csfi-in}}), \tag{8}$$

where $\mathrm{Avg}(\cdot)$ is the average pooling operation, and $\mathrm{CE}(\cdot)$ is the CE unit. Specifically, the fusion of features with different scales by upsampling or downsampling followed by element-wise addition can easily cause the information of salient defects to be diluted [19], resulting in blurry edge. Hence, we adopt the CE unit to effectively purify the boundary part of the feature map thus maintaining more desired features. The contrast operation is the core of CE unit, and is in charge of extracting the edge features contained in the two-scale features for better highlighting the edge regions. $\mathrm{CE}(\cdot)$ can be computed as:

$$\mathrm{CE}(\boldsymbol{f}) = \mathrm{conv}(\mathrm{Contrast}(\mathrm{BConv}(\boldsymbol{f}))), \tag{9}$$

$$\mathrm{Contrast}(\boldsymbol{f}) = s(\mathrm{conv}(\boldsymbol{f} \ominus \mathrm{Avg}(\boldsymbol{f}))) \otimes \boldsymbol{f} \oplus \boldsymbol{f}, \tag{10}$$

where $\ominus$ is element-wise subtraction.

Next, the two-scale enhanced features are integrated in an interleaved manner as follows:

$$\boldsymbol{f}^i_{\mathrm{los}} = \boldsymbol{f}^i_{\mathrm{ls}} \oplus \mathrm{Avg}(\boldsymbol{f}^i_{\mathrm{os}}), \tag{11}$$

$$\boldsymbol{f}^i_{\mathrm{ols}} = \boldsymbol{f}^i_{\mathrm{os}} \oplus \mathrm{Up}(\boldsymbol{f}^i_{\mathrm{ls}}), \tag{12}$$

where $\mathrm{Up}(\cdot)$ is the unsampling operation implemented by bilinear interpolation. Furthermore, we adopt the CE unit to enhance $\boldsymbol{f}^i_{\mathrm{ls}}$ and $\boldsymbol{f}^i_{\mathrm{os}}$ again based on the edge information, generating $\hat{\boldsymbol{f}}^i_{\mathrm{los}}$ and $\hat{\boldsymbol{f}}^i_{\mathrm{ols}}$.

(2) Cross-scale weighting stage. At the second stage, we use Cross-Scale Weighting Module (CSWM) to fuse features of different sizes, which is similar to the CSCM. CSWM is a variant of CSCM, and the difference between them is that in the re-calibration process, the collective weight map weights the two groups of features differently. Therefore, based on Eq. (6) of CSCM, the last operation of CSWM can be computed as follows:

$$\boldsymbol{f}^i_{\mathrm{csw}} = (\boldsymbol{W}^i \otimes \hat{\boldsymbol{f}}^i_{\mathrm{los}}) \oplus ((\mathbf{1} \ominus \boldsymbol{W}^i) \otimes \hat{\boldsymbol{f}}^i_{\mathrm{ols}}). \tag{13}$$

in which we adopt two different weights to emphasize the two forms of intra-layer feature. Finally, we adopt the residual connection operation to add the input feature of CSFIM to $\boldsymbol{f}^i_{\mathrm{csw}}$, and generate the output feature of CSFIM, $\boldsymbol{f}^i_{\mathrm{csfi}}$.

In summary, the two branches of features with different resolutions effectively complement each other and their corresponding edge information is also purified through the CE unit. At the second fusion stage, two branches of enhanced features are fused by CSWM to generate more robust features in an appropriate way.

3.4. Loss function

The decoder network produces the predicted saliency maps $\mathbf{S}^i$ ($i \in \{1,2,3,4,5\}$) with increased resolutions, and the structural details of salient defects gradually appear with the effect of CSCM and CSFIM. In addition to the above well-designed modules, we adopt the widely used deep supervision [71,72] in the training phase to supervise the decoder layers and to let them learn the characteristics of diverse-scale defects. Since using the hybrid loss achieves success in some representative SOD works [44,68,73,74], we adopt not only the classic pixel-level BCE loss and the map-level IoU loss, but also the CEL loss and the SSIM loss [75] in our loss function. We believe that the SSIM loss [75] can facilitate learning of structure details in our CSEPNet. Therefore, our loss function can be formulated as:

$$\begin{aligned}\mathbb{L}_{\mathrm{total}} = \sum_{i=1}^{5} \big(&\ell_{bce}(\mathbf{S}^i, \mathbf{G}) + \ell_{iou}(\mathbf{S}^i, \mathbf{G}) \\ &+ \ell_{cel}(\mathbf{S}^i, \mathbf{G}) + \ell_{ssim}(\mathbf{S}^i, \mathbf{G})\big),\end{aligned} \tag{14}$$

where $\mathbf{G}$ is the ground truth, and $\ell_{bce}(\cdot)$, $\ell_{iou}(\cdot)$, $\ell_{cel}(\cdot)$, and $\ell_{ssim}(\cdot)$ are BCE loss, IoU loss, CEL loss, and SSIM loss, respectively. The use of such a hybrid loss function helps our CSEPNet better adapt to the special scenes in SDIs.

4. Experiments

In this section, we first introduce the dataset, implementation details and evaluation metrics. Then, we compare our method with state-of-the-art methods and present ablation studies to comprehensively demonstrate the effectiveness of our method. Finally, we conduct extension experiments on two optical remote sensing images datasets to demonstrate the robustness of our method and discuss some failure cases.

4.1. Experimental setup

(1) Datasets. The detailed and convincing experiments are conducted on the public SD-saliency-900 [13] dataset which contains 900 SDIs and corresponding pixel-level annotations. To be specific, the dataset contains three defect categories, and each one includes 300 images.

(2) Implementation Details. Our proposed CSEPNet is implemented on the pytorch framework [85] with one NVIDIA Titan XP GPU (12 GB memory) and the input size of network is 256 × 256 × 3. During the training phase, we adopt several data augmentation operations (*i.e.*, random horizontal flipping, random rotating, random color jittering, and salt and pepper noise) to avoid overfitting. The parameters

Table 1
Quantitative performance comparisons with a total of 26 state-of-the-art methods across four categories on the SD-saliency-900 [13] dataset. The best three results for each metric are marked in **red**, **blue** and **green**, respectively. ↑ and ↓ mean larger and smaller are better, respectively.

Model	Type	#Param(M)↓	FLOPs(G)↓	S_α ↑	max F_β ↑	meanF_β ↑	adp F_β ↑	max E_ξ ↑	mean E_ξ ↑	adp E_ξ ↑	$\mathcal{M}$ ↓
BC_{14} [76]	TN.	–	–	.592	.470	.426	.470	.675	.637	.730	.156
SMD_{17} [77]	TN.	–	–	.582	.466	.415	.439	.648	.588	.716	.209
$2LSG_{17}$ [78]	TN.	–	–	.554	.435	.385	.416	.630	.559	.710	.246
$RCRR_{18}$ [79]	TN.	–	–	.533	.392	.335	.328	.628	.546	.644	.242
DSS_{17} [72]	CN.	62.2	104.4	.775	.786	.747	.804	.893	.814	.893	.032
$NLDF_{17}$ [80]	CN.	35.5	115.5	.811	.784	.723	.738	.904	.826	.884	.047
$PiCANet_{18}$ [81]	CN.	47.2	108.1	.873	.865	.807	.749	.958	.916	.895	.031
$BMPM_{18}$ [82]	CN.	22.1	724.3	.822	.827	.809	.805	.924	.891	.926	.037
$R3Net_{18}$ [23]	CN.	56.7	47.5	.824	.816	.809	.820	.927	.899	.927	.030
CPD_{19} [39]	CN.	29.2	59.5	.858	.853	.821	.794	.953	.915	.926	.031
$BASNet_{19}$ [44]	CN.	87.1	127.3	.866	.858	.841	.821	.957	.947	.948	.027
$PoolNet_{19}$ [18]	CN.	53.6	123.4	.866	.852	.815	.779	.954	.921	.920	.029
$EGNet_{19}$ [35]	CN.	108.1	291.9	.867	.858	.821	.781	.959	.928	.924	.028
$PFANet_{19}$ [83]	CN.	37.3	–	.742	.704	.593	.552	.855	.752	.725	.081
$GCPANet_{20}$ [21]	CN.	67.1	54.3	**.876**	**.871**	.830	.786	.956	.926	.925	.027
$MINet_{20}$ [17]	CN.	47.6	146.3	.868	.857	.836	.818	.948	.935	.942	**.025**
$SAMNet_{21}$ [84]	CN.	1.3	0.5	.830	.820	.764	.742	.933	.856	.900	.038
$SUCANet_{21}$ [42]	CN.	117.7	56.4	.869	.860	.834	.812	.956	.934	.939	.027
$PA\text{-}KRN_{21}$ [19]	CN.	141.1	617.7	.872	.865	.833	.807	**.963**	.947	.941	.027
$EDRNet_{20}$ [14]	CS.	39.3	42.0	**.877**	**.872**	**.854**	.834	**.964**	**.956**	**.953**	**.024**
$DACNet_{21}$ [15]	CS.	98.4	35.3	.875	.870	**.855**	**.836**	**.964**	**.957**	**.956**	**.024**
$SINet_{21}$ [47]	CC.	27.0	12.3	.871	.870	.836	.811	.960	.948	.937	.026
$PFNet_{21}$ [49]	CC.	46.5	26.5	.873	.861	.842	.822	.958	.948	.946	.026
$UJSC_{21}$ [51]	CC.	218.0	56.3	.874	**.872**	.847	**.835**	.961	.954	.952	**.024**
$SLSR_{21}$ [50]	CC.	50.9	32.4	.870	.867	.833	.805	.960	.946	.935	.027
$ERRNet_{22}$ [48]	CC.	69.8	20.1	.867	.862	.829	.806	.958	.946	.937	.026
Ours	CS.	18.8	59.3	**.884**	**.882**	**.859**	**.846**	**.966**	**.959**	**.959**	**.023**

TN.: Traditional NSI-SOD method, CN.: CNN-based NSI-SOD method, CS.: CNN-based SDI-SOD method, CC.: CNN-based COD method.

of the backbone network are initialized by the pre-trained parameters on the ImageNet. To make the network converge better, we use the momentum SGD optimizer with a weight decay of $5e^{-4}$, and set the corresponding parameters like initial learning rate and momentum to $1e^{-3}$ and 0.9, respectively. We apply the poly strategy [86] with a factor of 0.9 to train our CSEPNet for 100 epochs. The training set contains 540 noise-free images (*i.e.*, 180 images from per defect category). Notably, taking into account the overlap of images between the test set and the training set in [14,15], we strictly distinguish between the training set and the test set, and use the remaining 360 images as our test set, rather than the total 900 images as the test set like [14,15].

(3) Evaluation Metrics. We use four evaluation metrics including S-measure, F-measure, E-measure and Mean Absolute Error (MAE) to compare our method with other state-of-the-art methods. Besides, we present Precision-Recall (PR) curve to compare the statistic result of different methods. For F-measure and E-measure, we report their maximum, mean and adaptive values to deeply assess the performance of methods. $\mathbf{S-measure}$ (S_α, $\alpha = 0.5$) [87] is a metric based on the characteristics of human visual system and can effectively evaluate structural similarity between the saliency map and the GT. $\mathbf{F-measure}$ (F_β) [88] balances precision and recall, and we set β^2 to 0.3. $\mathbf{E-measure}$ (E_ξ) [89] is inspired by human visual characteristics and gets insight of the limitations of traditional metrics, jointly capturing image-level statistics and local pixel matching information. **MAE** ($\mathcal{M}$) is a basic metric used to evaluates the average pixel-level difference between the saliency map and the GT. **PRcurve** shows the overall statistic result of recall and precision.

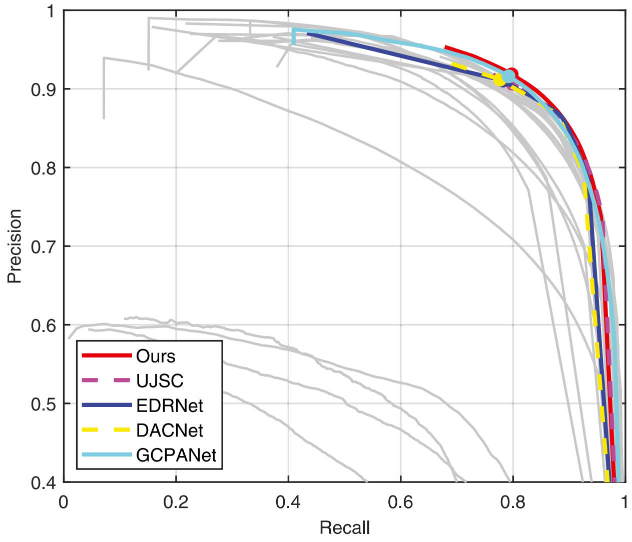


Fig. 6. Quantitative comparison in terms of PR curve on SD-saliency-90 [13] dataset. We show the top five methods in color.

4.2. Comparison with state-of-the-arts

For the comprehensive comparison, we compare with a total of 26 state-of-the-arts which can be divided into four categories. Traditional NSI-SOD method is the first category, and it includes BC [76], SMD [77], 2LSG [78], and RCRR [79]. CNN-based NSI-SOD method is the second category, and it includes DSS [72], NLDF [80], PiCANet [81], BMPM [82], R3Net [23], CPD [39], BASNet [44], PoolNet [18], EGNet [35], PFANet [83], GCPANet [21], MINet [17], SAMNet [84], SUCANet [42], and PA-KRN [19]. SDI-SOD method is the third category, and includes EDRNet [14] and DACNet [15]. For the last one, we also compare with five SOTA COD methods, including SINet [47], PFNet [49], UJSC [51] SLSR [50], and ERRNet [48]. To ensure a fair comparison, we use the saliency maps of sixteen methods provided by the representative DACNet [15] and EDRNet [14], and retrain ten CNN-based methods (*i.e.*, EGNet, GCPANet, SUCANet,

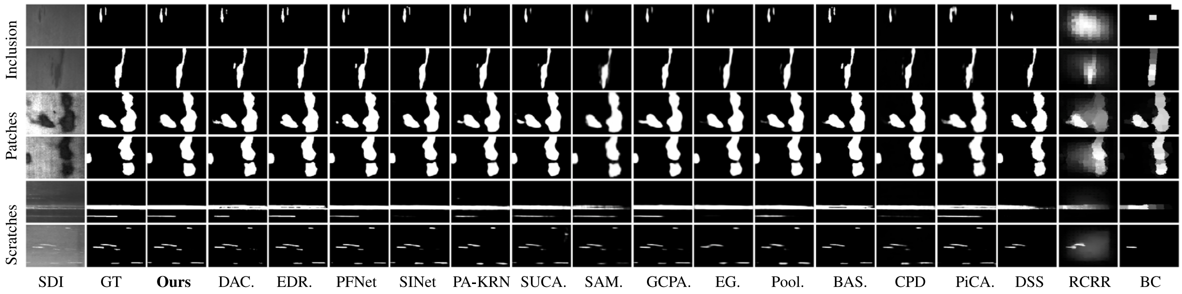


Fig. 7. Visual comparisons with sixteen representative methods on three categories of defects, including inclusion, scratches, and patches. These methods include CNN-based SDI-SOD methods (*i.e.*, DACNet [15] and EDRNet [14]), COD methods (*i.e.*, PFNet [49] and SINet [47]), CNN-based NSI-SOD methods (*i.e.*, PA-KRN [19], SUCANet [42], SAMNet [84], GCPANet [21], EGNet [35], PoolNet [18], BASNet [44], CPD [39], PiCANet [81], and DSS [72]), and traditional NSI-SOD method (*i.e.*, RCRR [79] and BC [76]). Specifically, we simplify the names of some methods, for example, we represent 'GCPANet' as 'GCPA.'.

PA-KRN, MINet, SINet, PFNet, UJSC, SLSR and ERRNet) on SD-saliency-900 [13] using the default parameters setting to generate their saliency maps.

4.2.1. Quantitative comparison

Table 1 shows the performance of our method and other 26 methods on eight metrics denoted as S_α, $F_\beta^{\max}$, F_β^{mean}, F_β^{adp}, $E_\xi^{\max}$, E_ξ^{mean}, E_ξ^{adp}, and $\mathcal{M}$. Briefly speaking, our method performs best on all metrics as compared with the other 26 methods on the SD-saliency-900 dataset, and DACNet and EDRNet rank second and third, respectively. Specifically, our method outperforms the second best model by 0.7%, 1.0%, 0.4%, 1.0%, 0.2%, 0.2%, 0.3%, and 0.001 on S_α, $F_\beta^{\max}$, F_β^{mean}, F_β^{adp}, $E_\xi^{\max}$, E_ξ^{mean}, E_ξ^{adp}, and $\mathcal{M}$, respectively. In comparison to the four traditional methods, our method is a lot ahead of them. Compared with the latest CNN-based SDI-SOD method DACNet, our method consistently outperforms it on all metrics, *e.g.*, 1.2% better than it on $F_\beta^{\max}$ and 0.1% lower than it on $\mathcal{M}$. Besides, we observe that the dedicated SDI-SOD methods generally perform better than the NSI-SOD methods, which proves the importance of development of dedicated SDI-SOD methods. The five COD methods perform better than most of NSI-SOD methods. However, when compared with the latest two SDI-SOD methods, they perform moderately. Our CSEPNet still outperforms these five COD methods on all metrics. In addition, we present PR curves of all compared methods in Fig. 6. It is obvious that the balance point of our PR curve is more towards the upper right corner and our curve wraps the other curves nicely, which demonstrates that our method outperforms other methods. Concurrently, the data shown in Table 1 also strongly supports this conclusion.

4.2.2. Computational complexity comparison

To further evaluate different methods, we provide the computational complexity performance, including the amount of parameters (#Param) and FLOPs, in Table 1. All data is calculated base on the source code released. Notably, #Param of our method is smaller than most of compared methods except SAMNet which is a lightweight model for SOD. For FLOPs, our method is 59.3G which is in the middle level of all compared methods. Therefore, the computational complexity of our method is competitive among all compared methods.

4.2.3. Visual comparison

To intuitively compare differences in the saliency maps of different methods, we provide the visual results of sixteen representative methods and our method in Fig. 7. For each category of defect (*i.e.*, inclusion, scratches and patches), we provides two representative cases. We summarize the characteristic of these three defect categories as follows: (1) inclusion is usually with tiny or relatively large area; (2) patches usually appears in center areas with close distance or appears in edge area; (3) scratches is usually in large coherent areas or in scattered areas. The above cases include the challenging scenes in SDIs, such as defects with fine structure, defects in large connected areas, scattered tiny defects, defects with low color contrast.

For all the above cases, we can observe that two traditional NSI-SOD methods get fuzzy localizations of defect regions or even miss them. In a word, these methods are always confused by the special scenes of SDIs. The ten CNN-based NSI-SOD methods can get the locations of main defects but still make some mistakes in case of multiple tiny defects, such as DSS, R3Net, BASNet and PA-KRN. Besides, the structural information of salient defects detected by these methods is still not refined enough. Similar to CNN-based NSI-SOD methods, the two COD methods can precisely locate the defect regions, but for the small targets they fail to detect certain regions. Moreover, the latest two SDI-SOD methods, *i.e.*, EDRNet and DACNet, are better than the above methods, and they can detect defects accurately and produce saliency maps with fine structures. In contrast, our CSEPNet generates clear saliency maps, which accurately locate salient defects and overcome the above problems, and shows strong adaptability in these scenes.

4.3. Ablation study

In order to demonstrate the effectiveness of each component of our CSEPNet, we provide some variants of our CSEPNet, and train them on SD-saliency-900 dataset with the same setting of our original CSEPNet as in Section 4.1. To be specific, our experiments include: (1) the contribution of each module in CSEPNet, (2) the robustness of our CSEPNet on different backbones, (3) the rationality of cross-scale calibration and re-calibration in CSCM, (4) the effectiveness of CE unit and CSWM in CSFIM.

1. The contribution of each module in CSEPNet. To evaluate the contribution of CBAM, CSCM, and CSFIM, we offer four variants: (1) the encoder–decoder network (*i.e.*, "Baseline"), (2) the baseline network with CBAM (*i.e.*, "Baseline+CBAM"), (3) the baseline network with CBAM and CSCM (*i.e.*, "Baseline+CBAM+CSCM"), and (4) the baseline network with CBAM and CSFIM (*i.e.*, "Baseline+CBAM+CSFIM"). We report the performance of the above variants and our complete CSEPNet in Table 2.

With a basic encoder–decoder structure, "Baseline" only achieves 86.39% on S_m, 0.0255 on $\mathcal{M}$, 94.76% on max E_ξ and 85.7% on max F_β. After embedding the general CBAM [24] into "Baseline", the performance increases by 0.09%, 0.01%, 0.2% and 0.12% on these four metrics, respectively. CBAM together with CSCM increases "Baseline" by 0.79%, 0.03%, 1.12% and 1.23% on these four metrics, respectively. Besides, CBAM together with CSFIM increases "Baseline" by 1.03%, 0.12%, 1.58% and 1.77% on these four metrics, respectively. The combination of CBAM, CSCM, and CSFIM increases "Baseline" by 1.96%, 0.25%, 1.79% and 2.5% on these four metrics, respectively. In summary, our method performs better with continuously adding main components, which demonstrates the effectiveness of each module.

2. The robustness of our CSEPNet on different backbones. We replace the backbone of our CSEPNet with a different backbone to

Table 2
Ablation study on evaluating the contribution of each module in CSEPNet. Baseline removes all modules from CSEPNet, and it directly obtains features from each layer and fuses features of adjacent layers through upsampling and element-wise addition operations. The best result in each column is in **red**.

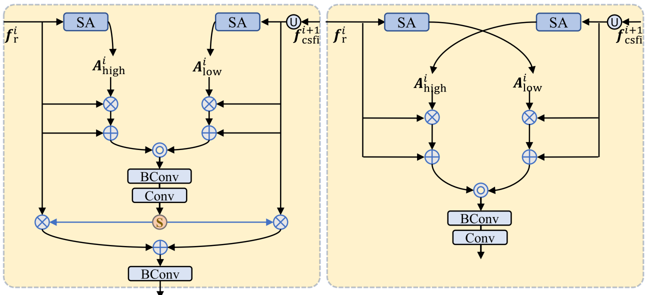
Dataset	Model	S_α ↑	$\mathcal{M}$ ↓	max E_ξ ↑	max F_β ↑
SD-saliency-900	Baseline	.8639	.0255	.9476	.8570
	Baseline + CBAM	.8648	.0254	.9496	.8582
	Baseline + CBAM + CSCM	.8718	.0252	.9588	.8693
	Baseline + CBAM + CSFIM	.8742	.0243	.9634	.8747
	Baseline + CBAM + CSCM + CSFIM	**.8835**	**.0230**	**.9655**	**.8820**

Table 3
Performance on ResNet-50 backbone and VGG-16 backbone with our CSEPNet.

Model	SD-saliency-900 [13]			
	S_α ↑	$\mathcal{M}$ ↓	max E_ξ ↑	max F_β ↑
CSEPNet-VGG	.8835	.0230	.9655	.8820
CSEPNet-ResNet	.8798	.0239	.9651	.8807

Table 4
Performance of variants of CSCM and CSFIM. The best result in each column is **bold**.

Dataset	Model	S_α ↑	$\mathcal{M}$ ↓	max E_ξ ↑	max F_β ↑
SD-saliency-900	*w/o CS*	.8786	.0234	.9646	.8813
	w/o re-calibration	.8768	.0237	.9640	.8795
	w/o CSWM	.8776	.0239	.9632	.8788
	w/o top-CE	.8788	.0238	.9633	.8777
	w/o bottom-CE	.8810	.0238	.9638	.8785
	CSFIM with CSCM	.8794	.0234	.9644	.8788
	Ours	**.8835**	**.0230**	**.9655**	**.8820**



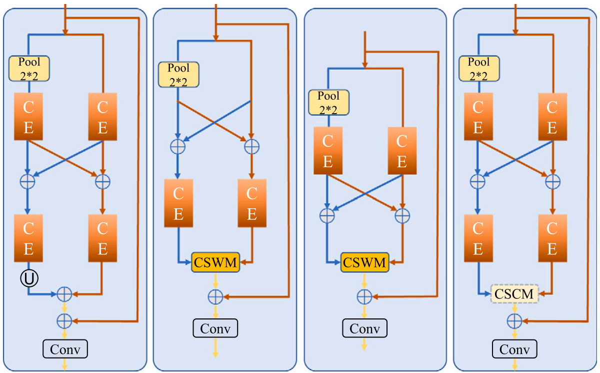
w/o CS   *w/o re-calibration*

Fig. 8. Illustration of two CSCM variants. Please zoom-in to view details.

illustrate the robustness of our CSEPNet. As shown in Table 3, CSEPNet-VGG denotes using VGG-16 [65] as the backbone of our CSEPNet, while CSEPNet-ResNet means using ResNet-50 [97] as the backbone of our CSEPNet. With the effective ResNet-50 backbone, CSEPNet-ResNet still achieves good performance, which demonstrates the robustness of our CSEPNet on different backbones.

3. The rationality of cross-scale calibration and re-calibration in CSCM. To evaluate the rationality of cross-scale calibration and re-calibration in CSCM, we present two variants. For the first one, we modify the cross-scale calibration to same-scale calibration, termed *w/o CS*. For the second one, we remove the re-calibration operation, termed *w/o re-calibration*. We show the structure of the above two variants in Fig. 8, and report their performance in the top part of Table 4.

We can observe that the performance degradation of *w/o CS*, *e.g.*, S_m: 88.35% → 87.86%, $\mathcal{M}$: 0.0230 → 0.0234, max E_ξ: 96.55% → 96.46%, and max F_β: 88.20% → 88.13%. This demonstrates that the cross-scale calibration operation can produce more comprehensive features. At the same time, we also observe the performance degradation of *w/o re-calibration*, *e.g.*, S_m: 88.35% → 87.68%, $\mathcal{M}$: 0.0230 → 0.0237, max E_ξ: 96.55% → 96.40%, and max F_β: 88.20% → 87.95%. In summary, we verify the rationality of two main components in our CSCM.



w/o CSWM   *w/o top-CE*   *w/o bottom-CE*   *CSFIM with CSCM*

Fig. 9. Illustration of four CSFIM variants. Please zoom-in to view details.

4. The effectiveness of CE unit and CSWM in CSFIM. To demonstrate the effectiveness of several important components in CSFIM, we present four variants, as shown in Fig. 9. We report their performance in the bottom part of Table 4.

We first verify the effectiveness of CSWM and replace it with up-sampling and element-wise addition operation, named *w/o CSWM*. We observe that the performance of *w/o CSWM* get worse, *e.g.*, S_m: 88.35% → 87.76%, $\mathcal{M}$: 0.0230 → 0.0239, max E_ξ: 96.55% → 96.32%, and max F_β: 88.20% → 87.88%. Then we verify the effectiveness of CE units at the top position and bottom position, respectively, and provide *w/o top-CE* and *w/o bottom-CE*. As a result, the performance of *w/o top-CE* also degrades, *e.g.*, S_m: 88.35% → 87.88%, $\mathcal{M}$: 0.0230 → 0.0238, max E_ξ: 96.55% → 96.33%, and max F_β: 88.20% → 87.77%. It is obvious that the performance of *w/o bottom-CE* drops slightly, *e.g.*, S_m: 88.35% → 88.10%, $\mathcal{M}$: 0.0230 → 0.0238, max E_ξ: 96.55% → 96.38%, and max F_β: 88.20% → 87.85%. Moreover, considering that CSCM is a variant of CSWM, we also conduct the experiment of replacing the CSWM in CSFIM with CSCM to further illustrate the effectiveness of CSWM, named *CSFIM with CSCM*. We can find out that the performance of *CSFIM with CSCM* also drops slightly. Therefore, we can conclude that it is a good manner to fuse the intra-layer features using the collective weight map as that in CSFIM. Therefore, we can conclude that each component in CSFIM is of great significance.

4.4. Extension experiment on optical remote sensing images datasets

To further demonstrate the compatibility and robustness of the proposed CSEPNet, we conduct experiments on two datasets for SOD in optical remote sensing images (ORSI-SOD), *i.e.*, EORSSD [90] and ORSSD [91]. The EORSSD dataset contains 1400 images for training and 600 images for testing, while the ORSSD dataset contains 600 images for training and 200 images for testing. We compare our method with seven SOTA ORSI-SOD methods, including LVNet [91], DAFNet [90], SARNet [92], MJRBM [93], EMFINet [94], ERPNet [95], and CorrNet [96]. As is shown in Table 5, our method dominates 6 out of 8 metrics on the EORSSD dataset and dominates 3 out of 8 metrics on the ORSSD dataset, which strongly proves the compatibility and robustness of our method. Besides, our CSEPNet also has advantages over most ORSI-SOD methods in terms of the amount of parameters and FLOPs.

Table 5
Quantitative and computational complexity comparisons with state-of-the-art ORSI-SOD methods on EORSSD and ORSSD datasets. The top three results are marked in **red**, **blue**, and **green**, respectively.

Methods	#Param	FLOPs	EORSSD [90]								ORSSD [91]							
	(M)↓	(G)↓	S_α ↑	$F_\beta^{\max}$ ↑	F_β^{mean} ↑	F_β^{adp} ↑	$E_\xi^{\max}$ ↑	E_ξ^{mean} ↑	E_ξ^{adp} ↑	$\mathcal{M}$ ↓	S_α ↑	$F_\beta^{\max}$ ↑	F_β^{mean} ↑	F_β^{adp} ↑	$E_\xi^{\max}$ ↑	E_ξ^{mean} ↑	E_ξ^{adp} ↑	$\mathcal{M}$ ↓
LVNet$_{19}$ [91]	-	-	.8630	.7794	.7328	.6284	.9254	.8801	.8445	.0146	.8815	.8263	.7995	.7506	.9456	.9259	.9195	.0207
DAFNet$_{21}$ [90]	29.35	68.5	.9166	.8614	.7845	.6427	**.9861**	.9291	.8446	**.0060**	.9191	.8928	.8511	.7876	**.9771**	.9539	.9360	.0113
SARNet$_{21}$ [92]	25.91	129.7	.9240	.8719	**.8541**	**.8304**	.9620	.9555	**.9536**	.0099	.9134	.8850	.8619	.8512	.9557	.9477	.9464	.0187
MJRBM$_{22}$ [93]	43.54	95.7	.9197	.8656	.8239	.7066	.9646	.9350	.8897	.0099	.9204	.8842	.8566	.8022	.9623	.9415	.9328	.0163
EMFINet$_{22}$ [94]	107.26	480.9	**.9290**	**.8720**	.8486	.7984	**.9711**	**.9604**	.9501	.0084	**.9366**	**.9002**	**.8856**	**.8617**	.9737	**.9671**	**.9663**	**.0109**
ERPNet$_{22}$ [95]	56.48	87.2	.9210	.8632	.8304	.7554	.9603	.9401	.9228	.0089	.9254	.8974	.8745	.8356	.9710	.9566	.9520	.0135
CorrNet$_{22}$ [96]	4.09	21.1	**.9289**	**.8778**	**.8620**	**.8311**	.9696	**.9646**	**.9593**	**.0083**	**.9380**	**.9129**	**.9002**	**.8875**	**.9790**	**.9746**	**.9721**	**.0098**
Ours	18.78	59.3	**.9305**	**.8799**	**.8620**	**.8497**	**.9734**	**.9686**	**.9675**	**.0068**	**.9387**	**.9081**	**.8933**	**.8905**	**.9743**	**.9692**	**.9697**	**.0093**

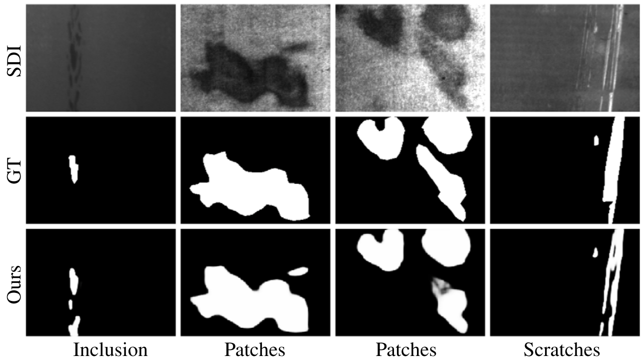


Fig. 10. Failure cases of our CSEPNet on challenging SDI images.

4.5. Failure cases on challenging SDI images

We present some failure cases of the proposed CSEPNet in Fig. 10. The challenging scenes include three kinds of defects including patches, inclusion, and scratches. As is shown in Fig. 10, the first column shows a typical tiny object scene of inclusion defect. Our CSEPNet fails to locate the most salient region and makes some errors. For the second column of Fig. 10, our method cannot distinguish the small area closed to the salient defect region. For the third column, the scene involves multiple defect regions. Our method locates all the defect regions, but the result is not precise for the bottom-right defect region. The last scene of Fig. 10 contains a long scratch with a fine structure. Our method fails to detect the whole area.

5. Conclusion

In this paper, we propose a CSEPNet for SDI-SOD, which adopts the CSCM to properly aggregate features from adjacent layers, and the CSFIM to model the complementarity of intra-layer features. In CSCM, adjacent features are first calibrated via the cross-scale attention map, and then re-calibrated via the collective weight map. In CSFIM, the two branches of features of different scales interweave, and meanwhile are enhanced by CE units. Then, the two branches of features are effectively fused by CSWM to generate the output features. Further, we adopt deep supervision with a hybrid loss function to guide five-scale saliency maps produced by our CSEPNet, making the training process stable and efficient. Experimental results on the public SD-saliency-900 dataset demonstrates our CSEPNet outperforms 26 state-of-the-art methods as well as the effectiveness of the proposed modules.

CRediT authorship contribution statement

Tuo Ding: Conceptualization, Methodology, Software, Investigation, Writing – original draft. **Gongyang Li:** Validation, Writing – review & editing. **Zhi Liu:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition, Resources. **Yike Wang:** Formal analysis, Visualization.

Declaration of competing interest

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

Acknowledgments

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.

References

[1] W. Wang, J. Shen, R. Yang, F. Porikli, Saliency-aware video object segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 40 (1) (2018) 20–33.
[2] G. Li, Z. Liu, R. Shi, Z. Hu, W. Wei, Y. Wu, M. Huang, H. Ling, Personal fixations-based object segmentation with object localization and boundary preservation, IEEE Trans. Image Process. 30 (2021) 1461–1475.
[3] G. Li, Z. Liu, R. Shi, W. Wei, Constrained fixation point based segmentation via deep neural network, Neurocomputing 368 (2019) 180–187.
[4] V. Mahadevan, N. Vasconcelos, Biologically inspired object tracking using center-surround saliency mechanisms, IEEE Trans. Pattern Anal. Mach. Intell. 35 (3) (2013) 541–554.
[5] G. Ke, S. Wang, H. Yang, W. Lin, G. Zhai, X. Yang, W. Zhang, Saliency-guided quality assessment of screen content images, IEEE Trans. Multimedia 18 (6) (2016) 1098–1110.
[6] S. Jia, Y. Zhang, Saliency-based deep convolutional neural network for no-reference image quality assessment, Multimedia Tools Appl. (2018) 14859–14872.
[7] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.
[8] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2015) 640–651.
[9] X. Kou, S. Liu, K. Cheng, Y. Qian, Development of a YOLO-V3-based model for detecting defects on steel strip surface, Measurement 182 (2021) 109454.
[10] R. Tian, M. Jia, DCC-CenterNet: A rapid detection method for steel surface defects, Measurement 187 (2022) 110211.
[11] H. Dong, K. Song, Q. Wang, Y. Yan, P. Jiang, Deep metric learning-based for multi-target few-shot pavement distress classification, IEEE Trans. Ind. Inform. 18 (3) (2021) 1801–1810.
[12] L. Huang, K. Song, J. Wang, M. Niu, Y. Yan, Multi-graph fusion and learning for RGBT image saliency detection, IEEE Trans. Circuits Syst. Video Technol. 32 (3) (2021) 1366–1377.
[13] G. Song, K. Song, Y. Yan, Saliency detection for strip steel surface defects using multiple constraints and improved texture features, Opt. Lasers Eng. 128 (2020) 106000.
[14] G. Song, K. Song, Y. Yan, EDRNet: ENcoder–decoder residual network for salient object detection of strip steel surface defects, IEEE Trans. Instrum. Meas. 69 (12) (2020) 9709–9719.
[15] X. Zhou, H. Fang, Z. Liu, B. Zheng, Y. Sun, J. Zhang, C. Yan, Dense attention-guided cascaded network for salient object detection of strip steel surface defects, IEEE Trans. Instrum. Meas. (2021).
[16] P. Zhang, D. Wang, H. Lu, H. Wang, X. Ruan, Amulet: Aggregating multi-level convolutional features for salient object detection, in: Proceedings IEEE ICCV, 2017, pp. 202–211.
[17] Y. Pang, X. Zhao, L. Zhang, H. Lu, Multi-scale interactive network for salient object detection, in: Proceedings IEEE CVPR, 2020, pp. 9410–9419.

[18] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, J. Jiang, A simple pooling-based design for real-time salient object detection, in: Proceedings IEEE CVPR, 2019, pp. 3912–3921.

[19] B. Xu, H. Liang, R. Liang, P. Chen, Locate globally, segment locally: A progressive architecture with knowledge review network for salient object detection, in: Proc. AAAI, Vol. 35, 2021, pp. 3004–3012.

[20] Z. Zhou, Z. Wang, H. Lu, S. Wang, M. Sun, Multi-type self-attention guided degraded saliency detection, in: Proc. AAAI, Vol. 34, 2020, pp. 13082–13089.

[21] Z. Chen, Q. Xu, R. Cong, Q. Huang, Global context-aware progressive aggregation network for salient object detection, in: Proc. AAAI, Vol. 34, 2020, pp. 10599–10606.

[22] X. Chen, Q. Zhang, L. Zhang, Edge-aware salient object detection network via context guidance, Image Vis. Comput. 110 (2021) 104166.

[23] Z. Deng, X. Hu, L. Zhu, X. Xu, J. Qin, G. Han, P.-A. Heng, R^{3}Net: Recurrent residual refinement network for saliency detection, in: Proc. IJCAI, 2018, pp. 684–690.

[24] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: Convolutional block attention module, in: Proceedings ECCV, 2018, pp. 3–19.

[25] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1254–1259.

[26] R. Valenti, N. Sebe, T. Gevers, Image saliency by isocentric curvedness and color, in: Proceedings IEEE ICCV, 2009, pp. 2185–2192.

[27] X. Shen, W. Ying, A unified approach to salient object detection via low rank matrix recovery, in: Proc. IEEE CVPR, 2012, pp. 853–860.

[28] H. Peng, B. Li, R. Ji, W. Hu, W. Xiong, C. Lang, Salient object detection via low-rank and structured sparse matrix decomposition, in: Proc. AAAI, 2013, pp. 796–802.

[29] J. Wright, A. Ganesh, S. Rao, Y.G. Peng, Y. Ma, Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization, in: Proc. NeurIPS, 2009, pp. 2080–2088.

[30] A.K. Gupta, A. Seal, P. Khanna, O. Krejcar, A. Yazidi, AWkS: ADaptive, weighted k-means-based superpixels for improved saliency detection, Pattern Anal. Appl. 24 (2) (2021) 625–639.

[31] Y. Wei, F. Wen, W. Zhu, J. Sun, Geodesic saliency using background priors, in: Proceedings ECCV, 2012, pp. 29–42.

[32] M. Ding, X. Xu, F. Zhang, Z. Xiao, M. Wang, Saliency detection via background prior and foreground seeds, Multimedia Tools Appl. (2019) 14849–14870.

[33] L. Zhou, Z. Yang, Z. Zhou, D. Hu, Salient region detection using diffusion process on a two-layer sparse graph, IEEE Trans. Image Process. 26 (12) (2017) 5882–5894.

[34] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, R. Yang, Salient object detection in the deep learning era: An in-depth survey, IEEE Trans. Pattern Anal. Mach. Intell. (2021) http://dx.doi.org/10.1109/TPAMI.2021.3051099.

[35] J. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, M.-M. Cheng, EGNet: Edge guidance network for salient object detection, in: Proceedings IEEE ICCV, 2019, pp. 8779–8788.

[36] A.K. Gupta, A. Seal, P. Khanna, A. Yazidi, O. Krejcar, Gated contextual features for salient object detection, IEEE Trans. Instrum. Meas. 70 (2021) 1–13.

[37] A. Borji, M.-M. Cheng, H. Jiang, J. Li, Salient object detection: A benchmark, IEEE Trans. Image Process. 24 (12) (2015) 5706–5722.

[38] X.-L. Hao, H. Liang, A multi-class support vector machine real-time detection system for surface damage of conveyor belts based on visual saliency, Measurement 146 (2019) 125–132.

[39] Z. Wu, L. Su, Q. Huang, Cascaded partial decoder for fast and accurate salient object detection, in: Proc. IEEE CVPR, 2019, pp. 3902–3911.

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

[41] H. Zhou, X. Xie, J.-H. Lai, Z. Chen, L. Yang, Interactive two-stream decoder for accurate and fast saliency detection, in: Proc. IEEE CVPR, 2020, pp. 9138–9147.

[42] J. Li, Z. Pan, Q. Liu, Z. Wang, Stacked U-shape network with channel-wise attention for salient object detection, IEEE Trans. Multimedia 23 (2021) 1397–1409.

[43] A.K. Gupta, A. Seal, P. Khanna, E. Herrera-Viedma, O. Krejcar, ALMNet: ADjacent layer driven multiscale features for salient object detection, IEEE Trans. Instrum. Meas. 70 (2021) 1–14.

[44] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, M. Jagersand, BASNet: Boundary-aware salient object detection, in: Proceedings IEEE CVPR, 2019, pp. 7479–7489.

[45] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: Proceedings MICCAI, 2015, pp. 234–241.

[46] A.K. Gupta, A. Seal, M. Prasad, P. Khanna, Salient object detection techniques in computer vision-a survey, Entropy 22 (10) (2020) 1174.

[47] D.-P. Fan, G.-P. Ji, M.-M. Cheng, L. Shao, Concealed object detection, IEEE Trans. Pattern Anal. Mach. Intell. (2021) http://dx.doi.org/10.1109/TPAMI.2021.3085766.

[48] G.-P. Ji, L. Zhu, M. Zhuge, K. Fu, Fast camouflaged object detection via edge-based reversible re-calibration network, Pattern Recognit. 123 (2022) 108414.

[49] H. Mei, G.-P. Ji, Z. Wei, X. Yang, X. Wei, D.-P. Fan, Camouflaged object segmentation with distraction mining, in: Proceedings IEEE CVPR, 2021, pp. 8772–8781.

[50] Y. Lv, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, D.-P. Fan, Simultaneously localize, segment and rank the camouflaged objects, in: Proceedings IEEE CVPR, 2021, pp. 11591–11601.

[51] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, Y. Dai, Uncertainty-aware joint salient object and camouflaged object detection, in: Proc. IEEE CVPR, 2021, pp. 10071–10081.

[52] Z. Huang, J. Wu, F. Xie, Automatic surface defect segmentation for hot-rolled steel strip using depth-wise separable U-shape network, Mater. Lett. 301 (2021) 130271.

[53] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, Q. Meng, PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection, IEEE Trans. Ind. Inform. 16 (12) (2020) 7448–7458.

[54] Y. He, K. Song, Q. Meng, Y. Yan, An end-to-end steel surface defect detection approach via fusing multiple hierarchical features, IEEE Trans. Instrum. Meas. 69 (4) (2020) 1493–1504.

[55] M. Tang, Y. Li, W. Yao, L. Hou, Q. Sun, J. Chen, A strip steel surface defect detection method based on attention mechanism and multi-scale maxpooling, Meas. Sci. Technol. 32 (11) (2021) 115401.

[56] Y. Wang, K. Song, J. Liu, H. Dong, Y. Yan, P. Jiang, RENet: REctangular convolution pyramid and edge enhancement network for salient object detection of pavement cracks, Measurement 170 (2021) 108698.

[57] J. Xing, M. Jia, A convolutional neural network-based method for workpiece surface defect detection, Measurement 176 (2021) 109185.

[58] Y. Bao, K. Song, J. Liu, Y. Wang, Y. Yan, H. Yu, X. Li, Triplet-graph reasoning network for few-shot metal generic surface defect segmentation, IEEE Trans. Instrum. Meas. 70 (2021) 1–11.

[59] M. Niu, K. Song, L. Huang, Q. Wang, Y. Yan, Q. Meng, Unsupervised saliency detection of rail surface defects using stereoscopic images, IEEE Trans. Ind. Inf. 17 (3) (2020) 2271–2281.

[60] M. Wei, R. Wang, Q. Guo, Multi-scale defect detection network for tire visual inspection, in: Intelligent Computing, 2022, pp. 771–782.

[61] D. Zhang, K. Song, J. Xu, Y. He, M. Niu, Y. Yan, MCnet: Multiple context information segmentation network of no-service rail surface defects, IEEE Trans. Instrum. Meas. 70 (2020) 1–9.

[62] J. Wang, Z. Liu, C. Li, R. Yang, B. Li, Self-attention deep saliency network for fabric defect detection, in: Proceedings BIC-TA, 2020, pp. 627–637.

[63] Y. Aslam, N. Santhi, N. Ramasamy, K. Ramar, Localization and segmentation of metal cracks using deep learning, J. Ambient Intell. Humaniz. Comput. 12 (6) (2021) 4205–4213.

[64] Tabernik, Domen, Sela, Samo, Skvarc, Jure, Skocaj, Danijel, Segmentation-based deep-learning approach for surface-defect detection, J. Intell. Manuf. (2020) 759–776.

[65] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings ICLR, 2015, pp. 1–14.

[66] G. Li, Z. Liu, W. Lin, H. Ling, Multi-content complementation network for salient object detection in optical remote sensing images, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–13.

[67] G. Li, Z. Liu, H. Ling, ICNet: INformation conversion network for RGB-D based salient object detection, IEEE Trans. Image Process. 29 (2020) 4873–4884.

[68] G. Li, Z. Liu, M. Chen, Z. Bai, W. Lin, H. Ling, Hierarchical alternate interaction network for RGB-D salient object detection, IEEE Trans. Image Process. 30 (2021) 3528–3542.

[69] G. Li, Z. Liu, L. Ye, Y. Wang, H. Ling, Cross-modal weighting network for RGB-D salient object detection, in: Proceedings ECCV, 2020, pp. 665–681.

[70] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proceedings IEEE CVPR, 2009, pp. 248–255.

[71] S. Xie, Z. Tu, Holistically-nested edge detection, in: Proceedings IEEE ICCV, 2015, pp. 1395–1403.

[72] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, P.H.S. Torr, Deeply supervised salient object detection with short connections, IEEE Trans. Pattern Anal. Mach. Intell. 41 (4) (2019) 815–828.

[73] G. Li, Z. Liu, D.Z. Lin, H. Ling, Adjacent context coordination network for salient object detection in optical remote sensing images, IEEE Trans. Cybern. (2022) http://dx.doi.org/10.1109/TCYB.2022.3162945.

[74] G. Li, Z. Liu, W. Lin, H. Ling, Multi-content complementation network for salient object detection in optical remote sensing images, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–13.

[75] W. Zhou, E.P. Simoncelli, A.C. Bovik, Multiscale structural similarity for image quality assessment, in: Proc. ACSSC, Vol. 2, 2003, pp. 1398–1402.

[76] W. Zhu, S. Liang, Y. Wei, J. Sun, Saliency optimization from robust background detection, in: Proceedings IEEE CVPR, 2014, pp. 2814–2821.

[77] H. Peng, B. Li, H. Ling, W. Hu, W. Xiong, S.J. Maybank, Salient object detection via structured matrix decomposition, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 818–832.

[78] L. Zhou, Z. Yang, Z. Zhou, D. Hu, Salient region detection using diffusion process on a two-layer sparse graph, IEEE Trans. Image Process. 26 (12) (2017) 5882–5894.

[79] Y. Yuan, C. Li, J. Kim, W. Cai, D.D. Feng, Reversion correction and regularized random walk ranking for saliency detection, IEEE Trans. Image Process. 27 (3) (2018) 1311–1322.

[80] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, P.-M. Jodoin, Non-local deep features for salient object detection, in: Proceedings IEEE CVPR, 2017, pp. 6609–6617.

[81] N. Liu, J. Han, M.-H. Yang, Picanet: Learning pixel-wise contextual attention for saliency etection, in: Proc. IEEE CVPR, 2018, pp. 3089–3098.

[82] L. Zhang, J. Dai, H. Lu, Y. He, G. Wang, A bi-directional message passing model for salient object detection, in: Proceedings IEEE CVPR, 2018, pp. 1741–1750.

[83] T. Zhao, X. Wu, Pyramid feature attention network for saliency detection, in: Proceedings IEEE CVPR, 2019, pp. 3080–3089.

[84] Y. Liu, X.-Y. Zhang, J.-W. Bian, L. Zhang, M.-M. Cheng, SAMNet: STereoscopically attentive multi-scale network for lightweight salient object detection, IEEE Trans. Image Process. 30 (2021) 3804–3814.

[85] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Proc. NeurIPS, 2019, pp. 8024–8035.

[86] W. Liu, A. Rabinovich, A.C. Berg, Parsenet: Looking wider to see better, 2015, arXiv preprint arXiv:1506.04579.

[87] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, A. Borji, Structure-measure: A new way to evaluate foreground maps, in: Proceedings IEEE ICCV, 2017, pp. 4548–4557.

[88] R. Achanta, S. Hemami, F. Estrada, S. Susstrunk, Frequency-tuned salient region detection, in: Proceedings IEEE CVPR, 2009, pp. 1597–1604.

[89] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, A. Borji, Enhanced-alignment measure for binary foreground map evaluation, in: Proceedings IJCAI, 2018, pp. 698–704.

[90] Q. Zhang, R. Cong, C. Li, M.-M. Cheng, Y. Fang, X. Cao, Y. Zhao, S. Kwong, Dense attention fluid network for salient object detection in optical remote sensing images, IEEE Trans. Image Process. 30 (2021) 1305–1317.

[91] C. Li, R. Cong, J. Hou, S. Zhang, Y. Qian, S. Kwong, Nested network with two-stream pyramid for salient object detection in optical remote sensing images, IEEE Trans. Geosci. Remote Sens. 57 (11) (2019) 9156–9166.

[92] Z. Huang, H. Chen, B. Liu, Z. Wang, Semantic-guided attention refinement network for salient object detection in optical remote sensing images, Remote Sens. 13 (11) (2021) 2163.

[93] Z. Tu, C. Wang, C. Li, M. Fan, H. Zhao, B. Luo, ORSI salient object detection via Multiscale Joint Region and boundary model, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–13.

[94] X. Zhou, K. Shen, Z. Liu, C. Gong, J. Zhang, C. Yan, Edge-aware multiscale feature integration network for salient object detection in optical remote sensing images, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–15.

[95] X. Zhou, K. Shen, L. Weng, R. Cong, B. Zheng, J. Zhang, C. Yan, Edge-guided recurrent positioning network for salient object detection in optical remote sensing images, IEEE Trans. Cybern. (2022) http://dx.doi.org/10.1109/TCYB.2022.3163152.

[96] G. Li, Z. Liu, Z. Bai, W. Lin, H. Ling, Lightweight salient object detection in optical remote sensing images via feature correlation, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–12.

[97] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings IEEE CVPR, 2016, pp. 770–778.