IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 6, 2021, accepted October 26, 2021, date of publication November 2, 2021, date of current version November 10, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3124814*

# Edge-Aware Multi-Level Interactive Network for Salient Object Detection of Strip Steel Surface Defects

**XIAOFEI ZHOU[1], HAO FANG[1], XIAOBO FEI[1], RAN SHI[2], AND JIYONG ZHANG[1]**

[1]School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China
[2]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Corresponding author: Jiyong Zhang (jzhang@hdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61901145 and Grant 61801219; and in part by the Hangzhou Dianzi University (HDU) and the China Electronics Corporation DATA (CECDATA) Joint Research Center of Big Data Technologies under Grant KYH063120009.

**ABSTRACT** The performance of the salient object detection of strip surface defects has been promoted largely by deep learning based models. However, due to the complexity of strip surface defects, the existing models perform poorly in the challenging scenes such as noise disturbance, and low contrast between defect regions and background. Meanwhile, the detection results of existing models often suffer from coarse boundary details. Therefore, we propose a novel saliency model, namely an Edge-aware Multi-level Interactive Network, to detect the defects from the strip steel surface. Concretely, our model adopts the U-shape architecture where the two crucial points are the interactive feature integration and the edge-guided saliency fusion. Firstly, except the skip connection that combines the same stage of encoder and decoder, we deploy another connection, where the features from adjacent levels of encoder are transferred to the same stage of decoder. By this way, we are able to provide an effective fusion of multi-level deep features, yielding a well depiction for defects. Secondly, to give well-defined boundaries for prediction results, we add the edge extraction branch after each decoder block, where the progressive feature aggregation endows the edge with precise details and complete object cues. Meanwhile, together with the edge extraction branches, we deploy the saliency prediction branch at each decoder stage. After that, coupled with the fine edge information, we fuse all outputs of saliency prediction branches into the final saliency map, where the edge cue steers the saliency result to pay more attention to the boundary details. Following this way, we can provide a high-quality saliency map which can accurately locate and segment the defects. Extensive experiments are performed on the public dataset, and the results prove the effectiveness and robustness of our model which consistently outperforms the state-of-the-art models.

**INDEX TERMS** Salient object detection, surface defects, multi-level feature, fusion, edge.

## I. INTRODUCTION

Surface defect detection is a very important research area in the field of machine vision, which tries to locate the defect regions in the collected surface images. Here, this paper focuses on strip steel which is a kind of industrial material and is widely used in ships, bridges, cars, military, and so on. There are many types of surface defects on strip steel, such as inclusions, patches, and scratches, which are caused by the equipment, raw materials, technology, casting and other factors in the production process of strip steel [1]. The defects exert a negative influence on the quality of strip steel, the deep processing, and the aesthetics of products. Therefore, the strip steel surface defect detection technology is deployed on the production line to inspect the surface and locate the defects, so as to realize the effective control of the strip steel quality.

Generally, the defect detection is often conducted by the human vision based manual inspection, where the surface defect details cannot be observed in time. Besides, the manual inspection is easily affected by the working environment, equipment stability, and subjective factors. Thus, in recent years, manual inspection methods have been

The associate editor coordinating the review of this manuscript and approving it for publication was Michele Nappi.

gradually replaced by the machine vision based detection methods which are more efficient and more robust. Meanwhile, as we all known, saliency detection [2] tries to capture the most visually attractive regions of an image, and coincidentally, the defect regions of the strip steel surface can also be regarded as salient regions. Saliency detection is often treated as a preprocessing operation for many vision tasks such as tracking [3], [4], segmentation [5], [6], quality assessment [7], [8], and defect detection [9]–[17]. Therefore, this paper attempts to regard the strip steel surface defect detection as the salient object detection, so as to effectively highlight the defect regions.

Recently years, there are many efforts have been devoted to the research of saliency detection. Concretely, the traditional saliency models can be categorized into two classes. The first one is heuristic priors based models, such as the contrast-based saliency model [18], [19], center-surround differences based models [20], [21], and the background prior based models [22], [23]. The second one is traditional machine learning algorithm based models such as random forest [24], [25], support vector machine [26], conditional random field [27], and so on. However, the traditional saliency models are mainly designed based on the hand-crafted features which cannot give a well depiction for strip steel surface defects, especially some complex surface scenes such as low contrast between defects and backgrounds, small defect regions, and noise disturbance. This often results in the generated saliency map cannot pop-out the defects completely. To tackle this dilemma, the deep learning technology has been applied to this field [28]–[37]. Although the performance of saliency models has been elevated largely, the inference results still suffer from two problems. The first problem is the coarse boundaries, where the defects cannot be accurately segmented when there are multiple defect regions with different sizes and the defects with fine structure in the image. The second one is the robustness of the model, where the performance degrades largely when dealing with the noise interference data.

To address the above challenges, we propose a novel saliency model, namely Edge-aware Multi-level Interactive Network shown in Fig. 1, to detect the strip steel surface defects. The entire network is an U-shaped architecture [38], and the two crucial components of our model are the interactive feature integration and edge-guided saliency fusion. To be specific, the proposed model first extracts multi-level deep features by using the encoder part. Then, we deploy the decoder to aggregate the multi-level deep features. Particularly, the existing U-shape based saliency models [17], [28], [39] try to combine the deep features derived from the same stage encoder and decoder by using the skip connection. Following this way, each level deep feature can only give a scale-specific representation for defects, and it is lack of information exchange between different layers, where the shallow layer features may be impaired by the continuous combination of the features from deep layers. Thus, for each level feature, we attempt to integrate the features from adjacent levels of encoder which will provide more relevant and effective cues for the current level feature. By this way, we can realize the interaction of adjacent level features, and facilitate the flow of information from different levels. After that, to further promote the boundary quality of inference results, there are many models [17], [28], [39] attempt to introduce edge cues. Inspired by this, we also introduce edge information into our network. We deploy the edge extraction branch after each decoder block, and meanwhile, we add a saliency prediction branch at each decoder block, where the edge information not only conveys precise boundary details but also is endowed with complete object cues. Subsequently, we fuse the saliency inference results and the fine edge information into the final high-quality saliency map which can accurately locate and segment the defects.

Overall, the contribution of this paper can be summarized as follows:

1) We propose a novel saliency model, *i.e.* Edge-aware Multi-level Interactive Network, to detect strip steel surface defects, where the two key points are the interactive feature integration and the edge-guided saliency fusion.
2) To give an effective interaction of different level features, we integrate each level feature with its adjacent level features. Besides, to present high-quality boundary details of defect regions, we introduce the edge information to refine the saliency fusion.

## II. RELATED WORKS

There are numerous efforts have been devoted to the saliency detection of which the performance has been push forward significantly. Here, we mainly give a brief introduction for the two kinds of saliency models, namely the traditional models (the heuristic prior based models and the traditional machine learning based models) and the deep learning based model.

### A. TRADITIONAL SALIENCY MODELS

The pioneer work of saliency detection is conducted by Itti *et al.* [2], where the saliency is defined as the center-surround difference computed by color, intensity, and motion features. Following this mechanism, the Achanta *et al.* [40] defined the saliency by using frequency-tuned method. Besides, Cheng *et al.* [18] treated the saliency as a region contrast which is with respect to its nearby regions. In [19], a saliency prediction is designed from two contrast measures by the uniqueness and spatial distribution. There are also some other heuristic priors to build saliency measurement. For example, in [20], the discriminant center-surround hypothesis is proposed to estimate the saliency values of each image. In [21], the saliency of each pixel is defined as how much it discriminate from surroundings, which is computed by employing an anisotropic center-surround operator. In addition, the background prior is also adopted in many saliency efforts. For example, in [22], the boundary and connectivity priors about backgrounds are employed by geodesic saliency. In [23], the spatial layout of image regions with respect to image boundaries is employ to define the boundary connectivity which can be viewed as a saliency measurement.

In recent years, with the development of machine learning algorithms, the performance of saliency models has also been promoted to some degree. For example, in [24] and [25], the random forest is used to map the features to saliency values. In [26], Tong *et al.* employ the support vector machine based multiple kernel boosting method to estimate the saliency values. The conditional random field is also used to aggregate various unary saliency cues with pairwise information to highlight salient objects [27]. Besides, regularized random walk ranking [41] is employed to introduce prior saliency prediction to each pixel by simultaneously considering the region and pixel image features, thus generating high-quality saliency maps. In [42], Peng *et al.* employed the low-rank matrix theory to perform matrix decomposition, where the background and salient regions are represented by low-rank matrix and sparse matrix respectively. In [43], Huang *et al.* used the multiple instance learning to estimate each proposal's saliency value.

Generally, the aforementioned traditional models usually adopt hand-crafted features to compute the contrast, measure the background prior, train the machine learning algorithms, and so on. They cannot capture the rich semantic information of salient objects, *i.e.* defects. Therefore, when dealing with the challenging scenes, the existing traditional models are incapable of detecting the defect regions accurately and completely.

### B. DEEP LEARNING BASED SALIENCY MODELS

In recent years, the deep learning technologies have achieved a huge progress, and this is also benefit for the detecting of salient objects, where the performance of saliency models has been pushed forward significantly. For example, in [44], Luo *et al.* proposed a convolutional network that fuses the local and global cues via a multi-resolution 4 × 5 grid structure. In [45], Hou *et al.* inserted short connections to the skip-layer structures within holistically-nested edge detector. In [30], the recurrent residual refinement network is deployed to progressively refine the saliency maps by building a set of residual refinement blocks, where the low-level features and high-level features are alternatively utilized. In [46], the bi-directional message passing model is proposed to fuse multi-level features into the final saliency map, where the messages are flowed among multi-level features. In [47], pixel-wise contextual attention network is designed to selectively acquire an attention map, in which each attention map corresponds to the contextual relevance at each pixel. In [48], Zhao and Wu proposed the pyramid feature attention network consisting of context-aware pyramid feature extraction module and channel-wise attention module, which is employed to strength the high-level and low-level deep features. In [39], Liu *et al.* designed two pooling-based modules including the global guidance module and the feature aggregation module to promote the performance for saliency detection. In [29], Wu *et al.* designed the cascaded partial decoder framework to obtain a precise saliency map, where the low-level features are discarded and the high-level features are retained. In [28], the boundary-aware saliency detection network employed the hybrid loss to guarantee the accuracy of saliency maps. In [17], Song *et al.* proposed an encoder-decoder residual network to precisely segment the defect regions from the strip steel surface. Besides, in [31], a depth-quality-aware subnet is inserted into the classical bistream RGBD saliency network, which promotes the fusion of the RGB and depth information. In [32], Chen *et al.* proposed a lightweight temporal network to acquire the temporal information which can effectively interact with the corresponding spatial cues, which gives a well saliency inference on videos. In [49], the salient object detection is regarded as an object-level semantic re-ranking problem, where a lightweight deep network and a post-processing refinement are deployed successively. In [50], a stereoscopic attention mechanism is deployed to adaptively integrate various scale features.

Although the deep learning based saliency models have promoted the research of saliency detection, they are still weakness when approaching the complex defect scenes, especially the cluttered backgrounds and noise disturbance. In our model, we focus on the interaction of features from different layers and the effect of edge information, which gives an effective boost for the performance of salient object detection.

## III. THE PROPOSED METHOD

This section first gives an introduction for the proposed saliency model in Section III-A. Then, the interactive feature integration is detailed in Section III-B. After that, we detail the edge-guided saliency fusion in Section III-C. Lastly, the loss function will be elaborated in Section III-D.

### A. OVERALL ARCHITECTURE

The proposed saliency model shown in Fig.1 is built on the U-shape architecture consisting of encoder and decoder with pre-trained model ResNet-34 [51] as backbone, and the two key components of the network are interactive feature integration and edge-guided saliency fusion. Firstly, we discard the last average pooling layer and softmax function of ResNet-34. Then, the encoder contains six convolutional blocks. Concretely, the first convolutional block "Conv-E1" contains a 3 × 3 convolutional layer (channel = 64, stride = 1) and the residual learning block "conv2_x" from ResNet-34. The following convolutional blocks "Conv-E$i$" ($i$ = 2, 3,4) adopt the residual learning blocks of ResNet-34 (*i.e.*, "conv3_x", "conv4_x", and "conv5_x"). After that, to enlarge the receptive field of the entire network, a max pooling layer of stride 2 and another two convolutional blocks "Conv-E5" and "Conv-E6" are added after "Conv-E4", where each block is equipped with three basic residual blocks (channel = 512).

The overall process shown in Fig.1 can be summarized as follows: the input is the strip steel image **I**, and the

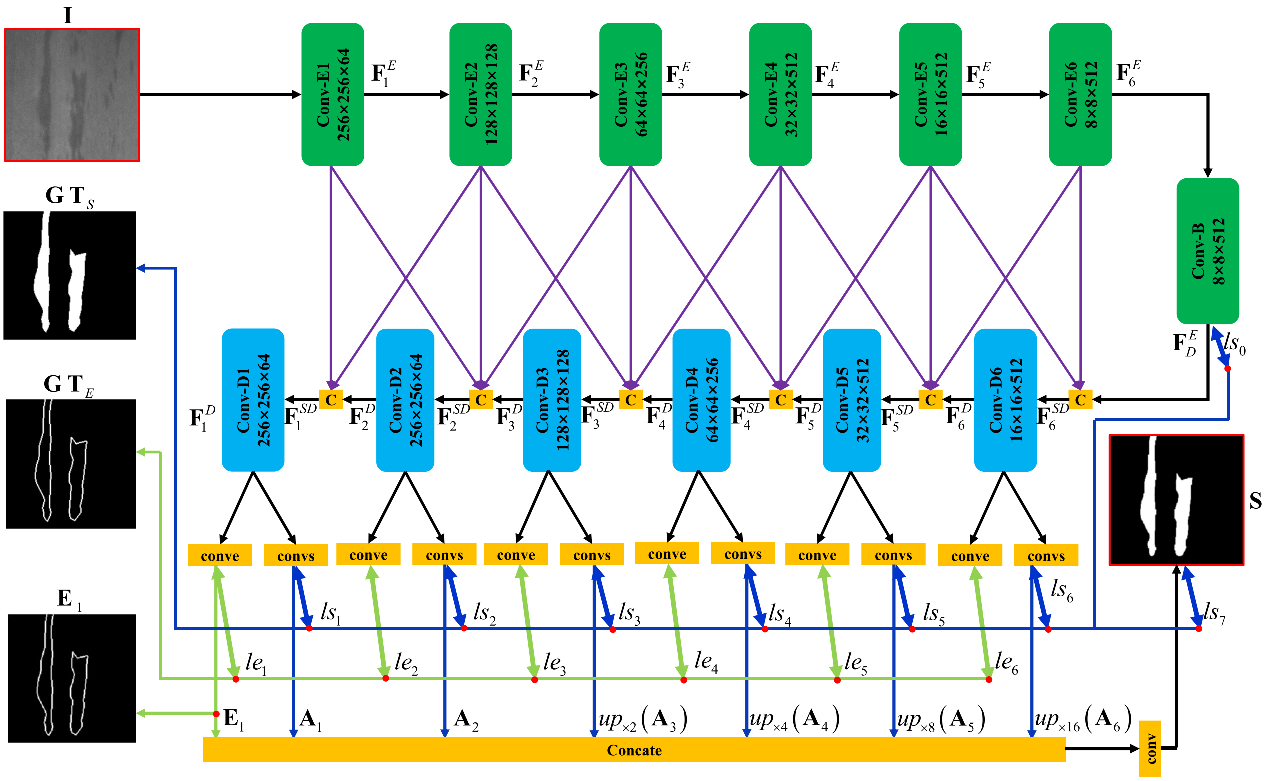


**FIGURE 1.** The overall architecture of the proposed saliency model: the encoder (*i.e.* Conv-E*i* (i = 1, ···, 6)) first extracts multi-level deep features $\{F_i^E\}_{i=1}^6$. Then, a deep feature $F_D^E$ is generated by using the bridge module(Conv-B). After that, the decoder (*i.e.* Conv-D*i* (i = 1, ···, 6)) progressively aggregate the multi-level deep features by integrating the adjacent level features, yielding the multi-level features $\{F_i^D\}_{i=1}^6$. Besides, we deploy the edge extraction branch "conve" and the saliency prediction branch "convs" after each decoder block. In addition, we employ the deep supervision to the network, namely $\{le_i\}_{i=1}^6$ and $\{ls_i\}_{i=0}^7$ presented by the blue and green arrows. Finally, the high-quality saliency map **S** is the aggregation of saliency prediction results $\{A_i\}_{i=1}^6$ and the fine edge cue $E_1$. Here, "up" means upsampling operation.

output of our model is the high-quality saliency map **S** which accurately highlights the defects. Firstly, the encoder extracts the multi-level deep feature $\{\mathbf{F}_i^E\}_{i=1}^6$. Then, through a bridge module "Conv-B" which consists of three dilated convolutional layers (channel = 512, dilation rate = 2) [52], we can obtain a global semantic information $\mathbf{F}_D^E$. After that, each decoder block integrates the feature from the corresponding encoder block, the features from adjacent encoder blocks, and the output from its previous decoder block. By this way, we can obtain six level deep features $\{\mathbf{F}_i^D\}_{i=1}^6$. Besides, to guarantee the accuracy of saliency prediction, we also deploy edge estimation branch after each side-path of decoder blocks. Correspondingly, the deep supervision is introduced to the entire network for the optimization of saliency prediction and edge extraction. Finally, by combing the saliency prediction $\{\mathbf{A}_i\}_{i=1}^6$ of all decoder blocks and the edge information $\mathbf{E_1}$ generated by the first decoder block, we can obtain the final saliency map **S**.

### B. INTERACTIVE FEATURE INTEGRATION

Through the encoder of our network, we can obtain multi-level deep features which present different cues of the object. Particularly, the features from shallow layers focus on the spatial details, the middle-level deep features convey the spatial and semantic cues simultaneously, and the features from deep layers provide rich semantic information of salient objects. To aggregate the multi-level features, there are many efforts have been designed. Concretely, the existing models [44], [53] often try to transfer the current level features to the corresponding level decoder block, where the current level decoder block can only present the scale-specific cues. Further, some other existing models [45], [54] attempt to fuse the multi-level features in a dense way, where the integration process requires huge computation resource. Fortunately, inspired by the mutual learning [55], [56], we conduct the interactive feature integration for the aggregation of multi-level deep features, as presented in Fig.1.

Formally, firstly, to capture the global semantic information, we introduce a bridge stage (Conv-B) between the encoder and the decoder, yielding the deep feature $\mathbf{F}_D^E$. The corresponding process can be defined as:

$$\mathbf{F}_D^E = f_B\left(\mathbf{F}_6^E\right), \tag{1}$$

where $f_B$ denote the bridge Conv-B, which contains three dilated convolutional layers (channel = 512, dilation rate = 2) [52]. Meanwhile, each of the dilated convolution

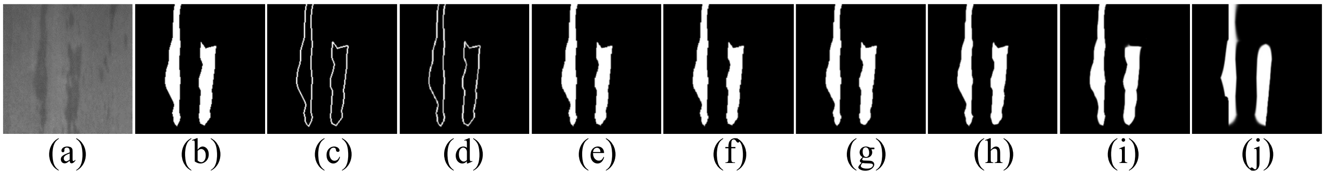


**FIGURE 2.** Visualization results of the edge extraction branch and the saliency prediction branches. (a): Input image, (b): GT of salient objects, (c): GT of salient edge, (d): $\mathbf{E}_1$, (e): $\mathbf{A}_1$, (f): $\mathbf{A}_2$, (g): $\mathbf{A}_3$, (h): $\mathbf{A}_4$, (i): $\mathbf{A}_5$, (j): $\mathbf{A}_6$. Here, the feature maps (d-j) are the results of the features $\{\mathbf{E}_1, \mathbf{A}_1 \sim \mathbf{A}_6\}$ after sigmoid activation function. Besides, for simplicity, we resize all maps to the same size as the input of our model.

layers is followed by a batch normalization (BN) layer and a ReLU layer, respectively.

Then, for each decoder block Conv-D$i$, its input $\mathbf{F}_i^{SD}$ not only contains the output of previous decoder block $\mathbf{F}_{i+1}^D$, but also includes the features $\{\mathbf{F}_i^E, \mathbf{F}_{i-1}^E, \mathbf{F}_{i+1}^E\}$ from the current-level encoder block Conv-E$i$ and its adjacent level encoder blocks (*i.e.* Conv-E$(i-1)$ and Conv-E$(i+1)$). By this way, we can obtain the input of the $i^{th}$ decoder block Conv-D$i$,

$$\mathbf{F}_i^{SD} = \begin{cases} \left[\mathbf{F}_D^E, \mathbf{F}_i^E, d_{\times 0.5}\left(\mathbf{F}_{i-1}^E\right)\right] (i=6) \\ \left[\mathbf{F}_{i+1}^D, \mathbf{F}_i^E, d_{\times 0.5}\left(\mathbf{F}_{i-1}^E\right), u_{\times 2}\left(\mathbf{F}_{i+1}^E\right)\right] \\ \qquad (i=2,3,4,5) \\ \left[\mathbf{F}_{i+1}^D, \mathbf{F}_i^E, u_{\times 2}\left(\mathbf{F}_{i+1}^E\right)\right] (i=1), \end{cases} \tag{2}$$

where $u_{\times 2}(\cdot)$ and $d_{\times 0.5}(\cdot)$ denote the upsampling and downsampling operations (sampling rate = 2, 0.5). Here, we should note that the upsampling and downsampling operations don't change the channel number of the features from adjacent levels.

Finally, with the generated initial fused deep features $\{\mathbf{F}_i^{SD}\}_{i=1}^6$, we pass them to the corresponding decoder blocks, respectively. This can be defined as

$$\mathbf{F}_i^D = f_i^D\left(\mathbf{F}_i^{SD}\right), \tag{3}$$

where the $f_i^D$ denotes the $i^{th}$ decoder block Conv-D$i$, and $\mathbf{F}_i^D$ is the output of Conv-D$i$. Here, Conv-D$i$ ($i = 2, \cdots, 6$) contains three $3\times3$ convolutional layers and a $2\times$ upsampling layer, where each convolutional layer is followed by batch normalization layer (BN) and a ReLU activation function. Conv-D1 only contains three $3 \times 3$ convolutional layers, and it isn't equipped with the upsampling layer. By this interactive feature integration process, we can obtain six-level deep features $\{\mathbf{F}_i^D\}_{i=1}^6$ which give a well representation for salient objects.

### C. EDGE-GUIDED SALIENCY FUSION

To obtain a high-quality saliency map with fine boundary details, many efforts have been paid their attention to the extraction and utilization of edge information [17], [28], [39]. Inspired by this, we also introduce edge information in our model. Differently, we deploy the edge extraction branch after each decoder block, namely "conve" shown in Fig.1. Meanwhile, we also add saliency prediction branch after each decoder block, namely "convs". Formally, the two operations are performed on the deep features $\{\mathbf{F}_i^D\}_{i=1}^6$, namely

$$\begin{cases} \mathbf{A}_i = f_s\left(\mathbf{F}_i^D\right) \\ \mathbf{E}_i = f_e\left(\mathbf{F}_i^D\right), \end{cases} \tag{4}$$

where $\mathbf{A}_i$ can be regarded as the attention map from the $i^{th}$ decoder block by conducting saliency prediction $f_s$, $f_e$ is the function of edge extraction branch, and $\mathbf{E}_i$ is the output of the $i^{th}$ edge extraction branch, *i.e.* the edge information. Here, both of the saliency prediction branch "convs" and edge extraction branch "conve" are set as a $3 \times 3$ convolutional layer. Besides, to give deep supervision of the saliency prediction branches and the edge extraction branches, we deploy the upsampling operation and the sigmoid activation function after the branches, namely the green and blue double-headed arrows shown in Fig. 1.

After that, we attempt to combine all side-outputs of saliency prediction branches, namely the attention maps $\{\mathbf{A}_i\}_{i=1}^6$. Meanwhile, we choose the first edge cue $\mathbf{E}_1$ to take part in the saliency fusion process, where the $\mathbf{E}_1$ is with the biggest resolution than other edge cues. Besides, we should note that in the saliency fusion, the two kinds of features including attention maps and edge cue are not processed by sigmoid activation function. In addition, according to Fig. 1, to concatenate the attention maps and edge information, we should first resize the attention maps $\mathbf{A}_3 \sim \mathbf{A}_6$ to $256 \times 256$ by upsampling operation. Finally, under the guidance of edge information $\mathbf{E}_1$, the saliency fusion can be defined as

$$\begin{aligned} \mathbf{S} = f([\mathbf{E}_1, \mathbf{A}_1, \mathbf{A}_2, up_{\times 2}(\mathbf{A}_3), up_{\times 4}(\mathbf{A}_4), \\ up_{\times 8}(\mathbf{A}_5), up_{\times 16}(\mathbf{A}_6)]), \end{aligned} \tag{5}$$

where $\mathbf{S}$ is the final saliency map, $[\cdot]$ means the concatenation operation, $f$ denotes the convolution operation and a sigmoid activation function, and $up$ denotes upsampling operation. Furthermore, we present the features $\mathbf{E}_1$ and $\{\mathbf{A}_i\}_{i=1}^6$ in Fig. 2, where the features give a well depiction for defects. Notably, to present a well visualization, we exhibit the results of the features $\mathbf{E}_1$ and $\{\mathbf{A}_i\}_{i=1}^6$ after the sigmoid activation function, as shown in Fig. 2(d-j). Following this way, we can get the high-quality saliency map with well defined boundary details, which can effectively highlight the defect regions on the strip steel surface as presented in Fig.1.

### D. LOSS FUNCTION

To remit the over fitting, some models [28], [45] employ the deep supervision for the side-outputs. Here, we introduce the deep supervision to our network.

Formally, we deploy the supervision to the saliency prediction branches and edge extraction branches, namely $\{ls_i\}_{i=1}^{6}$ and $\{le_i\}_{i=1}^{6}$. Besides, we also introduce the supervision to the bridge module and the final output of the entire network, which are defined as $ls_0$ and $ls_7$, respectively. Thus, the total loss $\mathcal{L}$ of the entire network can be defined as

$$\mathcal{L} = \sum_{i=0}^{7} ls_i + \sum_{i=1}^{6} le_i. \tag{6}$$

Besides, similar as [17], [28], we also adopt the hybrid loss to define the saliency prediction loss $\{ls_i\}_{i=0}^{7}$, namely

$$ls_i = ls_i^B + ls_i^I + ls_i^S, \tag{7}$$

where $ls_i^B$, $ls_i^I$ and $ls_i^S$ denote BCE loss [57], IoU loss [58] and SSIM loss [59], respectively. For the edge extraction loss $\{le_i\}_{i=1}^{6}$, each of them adopts the BCE loss [57].

Here, the aforementioned three losses including BCE, IoU, and SSIM are detailed below. BCE [57] (Binary Cross Entropy) loss is usually employed by the binary classification task, and it can be written as

$$\begin{aligned} l^B = -\sum_{(x,y)} [\mathbf{GT}(x,y) log(\mathbf{S}(x,y)) \\ +(1-\mathbf{GT}(x,y)) log(1-\mathbf{S}(x,y))], \end{aligned} \tag{8}$$

where $l^B$, $\mathbf{GT}$ and $\mathbf{S}$ denote the BCE loss, the ground truth and the predicted saliency map, respectively.

IoU [58] (Intersection over Union) loss is often deployed to evaluate the similarity of $\mathbf{GT}$ and $\mathbf{S}$, which can be written as

$$l^I = 1 - \frac{\sum_{(x,y)} \mathbf{S}(x,y)\mathbf{GT}(x,y)}{\sum_{(x,y)} [\mathbf{S}(x,y) + \mathbf{GT}(x,y) - \mathbf{S}(x,y)\mathbf{GT}(x,y)]}. \tag{9}$$

where $l^I$ is the IoU loss.

SSIM [59] (Structural Similarity) loss is initially designed in image quality assessment task, which can be used to acquire the structural information. Specifically, $\mathbf{P}_S = \left\{ P_S^j : j = 1, \ldots, N^2 \right\}$ and $\mathbf{P}_{GT} = \left\{ P_{GT}^j : j = 1, \ldots, N^2 \right\}$ denote two patches (size $= N \times N$) which are cropped from the saliency map $\mathbf{S}$ and the ground truth $\mathbf{GT}$, respectively. The SSIM $l^S$ of patch $\mathbf{P}_S$ and patch $\mathbf{P}_{GT}$ can be defined as:

$$l^S = 1 - \frac{(2\mu_{P_S}\mu_{P_{GT}} + C_u)(2\sigma_{P_S P_{GT}} + C_\sigma)}{(\mu_{P_S}^2 + \mu_{P_{GT}}^2 + C_u)(\sigma_{P_S}^2 + \sigma_{P_{GT}}^2 + C_\sigma)} \tag{10}$$

where $\mu_{P_S}, \mu_{P_{GT}}$ and $\sigma_{P_S}, \sigma_{P_{GT}}$ refer to the mean and standard deviations of $\mathbf{P}_S$ and $\mathbf{P}_{GT}$ respectively, $\sigma_{P_S P_{GT}}$ denotes the covariance of two patches, and $C_u$ and $C_\sigma$ are usually set to $0.01^2$ and $0.03^2$, respectively.

## IV. EXPERIMENTAL RESULTS

This section first provides the experimental setup in Section IV-A. Then, in Section IV-B, we compare the proposed model with the state-of-the-art saliency models in quantitative and qualitative ways. Lastly, the ablation analysis is presented in Section IV-C.

### A. EXPERIMENTAL SETUP

Here, we perform extensive experiments on the public strip steel dataset SD-saliency-900 [60] to verify the effectiveness of our model. SD-saliency-900 has 900 images, and it consists of three types of strip steel surface defects including inclusion, patches and scratches, and each type of defects has 300 images with 200 × 200 resolution.

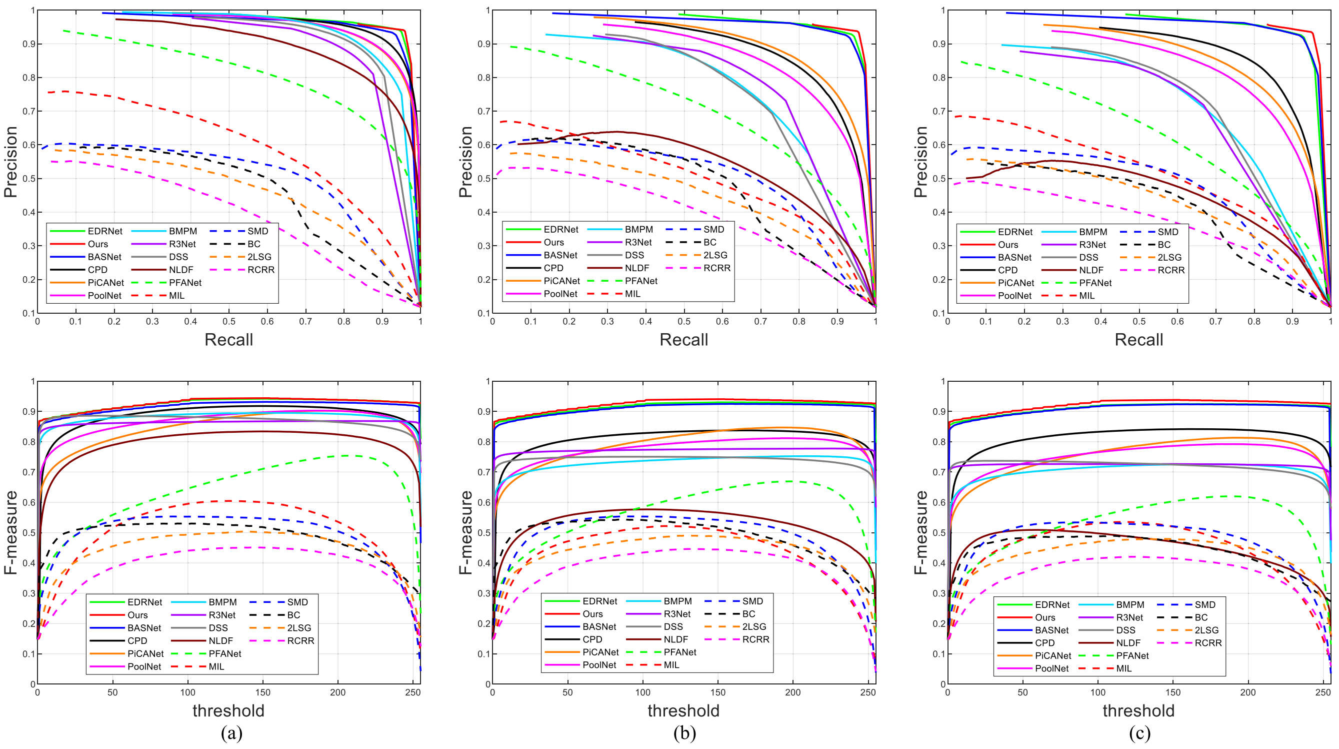
#### 1) PARAMETER SETTINGS AND IMPLEMENTATION DETAILS

Following [17], we generate the training set containing 1620 images. Concretely, we first choose 180 images from each type of defects, yielding the initial training set containing 540 images. Then, we further select 90 images from the each type of defects in the initial training set, and add salt & pepper noise ($\rho = 20\%$), generating the noise interference training set that consists of 270 images. Thus, we combine the initial training set and the noise disturbance training set, yielding the final training set which totally contains 810 images. After that, we perform horizontal flipping to augment the training set, yielding 1620 images. In addition, during the training phase, we resize each image $\mathbf{I}$ to 256 × 256, and then we perform normalization ($(\mathbf{I} - \mu)/\sigma$, $\mu = 0.4669$, and $\sigma = 0.2437$).

We implement our network with Pytorch 1.4.0, and the code is performed on a PC with an NVIDIA Titan XP GPU (with 12GB memory). Furthermore, to train the network, we initialize the encoder by using the ResNet-34 model [51], and the remaining convolutional layers are initialized by Glorot and Bengio [61]. Meanwhile, we adopt the Adam optimizer [62] to optimize our network, where the initial learning rate, betas, eps, and weight decay are set to $10^{-3}$, (0.9, 0.999), $10^{-8}$, and 0, respectively. The entire training process will continue until the loss converges. Besides, the training batch size is set to 10, and our training process runs about 15 hours. During test phase, we resize each image to 256 × 256, and the final saliency map is further resize to the same resolution as the input image by using bilinear interpolation. Generally, the average running speed of our model is about 48fps when dealing with 256 × 256 images.

#### 2) EVALUATION METRICS

In the experiment, we take the following metrics to evaluate the performance of our model, of which the metrics contain the precision-recall (PR) curve, the F-measure curve, mean absolute error (MAE), the weighted F-measure (WF) score, overlapping ratio (OR), structure-measure (SM), and Prattąŕs figure of merit (PFOM).



(a) (b) (c)

**FIGURE 3. Quantitative comparison results. The salt & pepper noise levels from (a) to (c) are set to 0, $\rho = 10\%$ and $\rho = 20\%$, respectively. The curves from top to down are PR curve and F-measure curve, respectively.**

**TABLE 1. Quantitative evaluation of different saliency models in terms of WF, SM, OR, PFOM, and MAE on the SD-saliency-900 datasets, where the salt & pepper noise levels are set to$\rho = 0$, 10%, 20%, respectively. Notice that, "↑" ("↓") denotes that the larger (smaller) the better, and the top three results in each column are marked in red, green and blue, respectively. Here, for saving space, we abbreviate MAE, WF, OR, SM, and PFOM as M, F, O, S, and P, respectively.**

| | | RCRR [41] | 2LSG [63] | BC [23] | SMD [42] | MIL [43] | PFANet [48] | NLDF [44] | DSS [45] | R3Net [30] | BMPM [46] | PoolNet [39] | PiCANet [47] | CPD [29] | BASNet [28] | EDRNet [17] | **Ours** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *$\rho$=0* | M↓ | 0.2439 | 0.2474 | 0.1554 | 0.2045 | 0.1824 | 0.0841 | 0.0474 | 0.0236 | 0.0254 | 0.0298 | 0.0215 | 0.0259 | 0.0211 | **0.0152** | **0.0130** | **0.0119** |
| | F↑ | 0.2591 | 0.3120 | 0.3855 | 0.3690 | 0.3376 | 0.5358 | 0.7134 | 0.7931 | 0.8160 | 0.8278 | 0.8476 | 0.8289 | 0.8192 | **0.9092** | **0.9225** | **0.9253** |
| | O↑ | 0.2526 | 0.3229 | 0.3588 | 0.3439 | 0.3827 | 0.4862 | 0.6558 | 0.7586 | 0.7286 | 0.7513 | 0.7456 | 0.7071 | 0.7750 | **0.8267** | **0.8416** | **0.8447** |
| | S↑ | 0.5342 | 0.5518 | 0.5942 | 0.5838 | 0.6182 | 0.7411 | 0.8028 | 0.8242 | 0.8397 | 0.8585 | 0.9018 | 0.8963 | 0.9033 | **0.9276** | **0.9374** | **0.9421** |
| | P↑ | 0.3267 | 0.3603 | 0.3449 | 0.3810 | 0.4114 | 0.5273 | 0.6143 | 0.7299 | 0.7407 | 0.7778 | 0.8649 | 0.8404 | 0.8856 | **0.8961** | **0.9133** | **0.9173** |
| *$\rho$=10%* | M↓ | 0.2552 | 0.2587 | 0.1519 | 0.1994 | 0.2128 | 0.1011 | 0.1175 | 0.0450 | 0.0382 | 0.0495 | 0.0345 | 0.0351 | 0.0353 | **0.0160** | **0.0139** | **0.0123** |
| | F↑ | 0.2557 | 0.3007 | 0.3733 | 0.3613 | 0.2921 | 0.4623 | 0.4196 | 0.6069 | 0.6961 | 0.6335 | 0.7263 | 0.7521 | 0.7235 | **0.9033** | **0.9125** | **0.9226** |
| | O↑ | 0.2586 | 0.3038 | 0.3509 | 0.3362 | 0.3315 | 0.4217 | 0.3781 | 0.5443 | 0.6020 | 0.5478 | 0.6300 | 0.6387 | 0.6654 | **0.8199** | **0.8290** | **0.8411** |
| | S↑ | 0.5302 | 0.5368 | 0.5881 | 0.5840 | 0.5683 | 0.6961 | 0.6339 | 0.7222 | 0.7660 | 0.7446 | 0.8213 | 0.8490 | 0.8308 | **0.9235** | **0.9299** | **0.9400** |
| | P↑ | 0.3138 | 0.3530 | 0.3352 | 0.3746 | 0.3485 | 0.4196 | 0.4125 | 0.4728 | 0.5835 | 0.5558 | 0.7060 | 0.7547 | 0.7343 | **0.8880** | **0.9051** | **0.9145** |
| *$\rho$=20%* | M↓ | 0.2842 | 0.2619 | 0.1753 | 0.1981 | 0.2083 | 0.1079 | 0.1253 | 0.0429 | 0.0430 | 0.0523 | 0.0373 | 0.0401 | 0.0324 | **0.0160** | **0.0146** | **0.0125** |
| | F↑ | 0.2431 | 0.2946 | 0.3327 | 0.3399 | 0.2947 | 0.4281 | 0.3637 | 0.5866 | 0.6170 | 0.6081 | 0.6981 | 0.7073 | 0.7325 | **0.9014** | **0.9056** | **0.9205** |
| | O↑ | 0.2464 | 0.2954 | 0.3152 | 0.3146 | 0.3441 | 0.3967 | 0.3391 | 0.5494 | 0.5349 | 0.5312 | 0.6030 | 0.5971 | 0.6743 | **0.8172** | **0.8202** | **0.8384** |
| | S↑ | 0.5147 | 0.5341 | 0.5623 | 0.5717 | 0.5770 | 0.6782 | 0.5892 | 0.7132 | 0.7258 | 0.7341 | 0.8104 | 0.8228 | 0.8384 | **0.9219** | **0.9244** | **0.9380** |
| | P↑ | 0.2908 | 0.3354 | 0.3088 | 0.3444 | 0.3428 | 0.3885 | 0.3136 | 0.4489 | 0.4849 | 0.5372 | 0.6871 | 0.7060 | 0.7566 | **0.8880** | **0.8970** | **0.9123** |

Formally, firstly, F-measure [40] is defined as the weighted harmonic mean of precision and recall, namely

$$F_\beta = \frac{\left(1+\beta^2\right) Precision \times Recall}{\beta^2 Precision + Recall}, \tag{11}$$

where we set $\beta^2$ to 1 as adopted in [24]. Correspondingly, the weighted F-measure [64] is a weighted version of F-measure, namely

$$WF = \frac{\left(1+\beta^2\right) Precision^w \times Recall^w}{\beta^2 Precision^w + Recall^w}. \tag{12}$$

Secondly, MAE [19] can be computed as

$$MAE = \frac{1}{W * H} \sum_{i=1}^{W*H} \left|\mathbf{S}(i) - \mathbf{GT}_s(i)\right|, \tag{13}$$

where $W$ and $H$ are the width and height of the saliency map $\mathbf{S}$, respectively. $\mathbf{GT}_s$ denotes the ground truth of salient objects.

Thirdly, OR describes the overlapping ratio between the segmentation result of saliency map (denoted by $\mathbf{S}'$) and the

**TABLE 2.** Comparison of the model size (MB) and the average running time (seconds per image) on the SD-saliency-900 dataset. Notably, "M", "C", "T", and "P" refer to Matlab, Caffe, TensorFlow, and PyTorch, respectively.

| | RCRR | 2LSG | BC | SMD | MIL | PFANet | NLDF | DSS | R3Net | BMPM | PoolNet | PiCANet | CPD | BASNet | EDRNet | **Ours** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | [41] | [63] | [23] | [42] | [43] | [48] | [44] | [45] | [30] | [46] | [39] | [47] | [29] | [28] | [17] | |
| **Code** | M | M | M+C | M+C | M+C | T | T | P | P | T | P | P | P | P | P | P |
| **Model size** | - | - | - | - | - | 61.9 | 56.5 | 237 | 214 | 57.1 | 260 | 180 | 183 | 332 | 150 | 378 |
| **Time** | 1.095 | .0639 | 0.054 | 0.319 | 24.134 | 0.023 | 0.051 | 0.110 | 0.117 | 0.037 | 0.030 | 0.116 | 0.055 | 0.046 | 0.037 | 0.021 |

**TABLE 3.** Quantitative comparison results of ablation studies. Notice that, the best result in each column is marked in bold face.

| Settings | MAE↓ | WF↑ | OR↑ | SM↑ | PFOM↑ |
|---|---|---|---|---|---|
| Baseline(B) | 0.0142 | 0.9140 | 0.8298 | 0.9324 | 0.9080 |
| B+Sup | 0.0131 | 0.9219 | 0.8411 | 0.9386 | 0.9155 |
| B + IFI | 0.0138 | 0.9158 | 0.8342 | 0.9329 | 0.9069 |
| B + IFI + Sup | 0.0126 | 0.9203 | 0.8400 | 0.9396 | 0.9094 |
| B + E | 0.0129 | 0.9218 | 0.8399 | 0.9389 | 0.9148 |
| B + E + Sup | 0.0125 | 0.9214 | 0.8410 | 0.9396 | 0.9137 |
| B + IFI + E | 0.0126 | 0.9217 | 0.8404 | 0.9394 | 0.9114 |
| B + IFI + E + Sup | **0.0119** | **0.9253** | **0.8477** | **0.9421** | **0.9173** |

ground truth $\mathbf{GT}_s$, namely

$$OR = \frac{|\mathbf{S}' \cap \mathbf{GT}_s|}{|\mathbf{S}' \cup \mathbf{GT}_s|}, \tag{14}$$

where $\mathbf{S}'$ can be generated by binarizing the saliency map $\mathbf{S}$ (the adaptive threshold can be set to twice the average value of $\mathbf{S}$).

Fourthly, SM [65] simultaneously considers the region-aware ($S_r$) and the object-aware ($S_o$) values to evaluate the structural similarity between saliency map $\mathbf{S}$ and ground truth $\mathbf{GT}_s$, which can be defined as

$$S = \alpha * S_o + (1 - \alpha) * S_r, \tag{15}$$

where $\alpha$ is the balance parameter, and it is set to 0.5.

Lastly, as a Prattąŕs figure of merit, PFOM [66] intuitively presents the boundary quality of the segmentation results, and it is often employed by the edge detection area, which can be defined as

$$PFOM = \frac{1}{max(N_G, N_S)\sum_{k=1}^{N_S} 1/(1 + \alpha d_k^2)}, \tag{16}$$

where $N_G$ and $N_S$ are the number of ideal and actual edge points extracted from ground truth map and binary saliency result, respectively. Besides, $\alpha$ denotes a scaling constant which is set to 0.1 or 1/9. In addition, $d_k$ denotes the Euclidean distance between the $k^{th}$ true edge point and the detected edge point.

### B. COMPARISON WITH THE STATE-OF-THE-ARTS

To quantitative and qualitative evaluate the performance of our model, we compare our model with totally 15 state-of-the-art models including RCRR [41], 2LSG [63], BC [23], SMD [42], MIL [43], PFANet [48], NLDF [44], DSS [45], R3Net [30], BMPM [46], PoolNet [39], PiCANet [47], CPD [29], BASNet [28] and EDRNet [17]. Notably, the saliency maps of all models are computed by executing the source codes or provided by the authors, where the deep learning based models are retrained by using the same training set as our model. Next, we successively present the quantitative and qualitative comparison results.

The quantitative comparison between our model and the state-of-the-art models are presented in Fig. 3 and Table 1. To be specific, Fig. 3 supplies the results of PR curves (top row) and F-measure curves (bottom row). Obviously, we can find that our model achieves the best performance when compared with other models in terms of PR curves and F-measure curves. Particularly, under the disturbance of salt & pepper noise ($\rho = 10\%$ and $\rho = 20\%$), our model still performs best, as presented in Fig. 3(b,c). In addition, when compared with the recently published work EDRNet [17], the improvement elevated by our model is still significantly. Furthermore, Table 1 provides the results in terms of MAE, WF, OR, SM and PFOM. We can find that our model still consistently outperforms other models with a large margin, especially on the noise disturbed data. Particularly, Compared with the performance of EDRNet [17] on the test set without noise disturbance ($\rho = 0\%$), the performance of EDRNet on the test set with noise disturbance ($\rho = 20\%$) degrades 12.3%, 1.8%, 2.5%, 1.4% and 1.8% in terms of MAE, WF, OR, SM and PFOM, respectively. In contrast, the performance degradation of our model is smaller, where the MAE, WF, OR, SM and PFOM only degrade 5.0%, 0.5%, 0.7%, 0.4% and 0.5%, respectively. Therefore, through the above quantitative comparisons, we can firmly demonstrate the superiority and effectiveness of our model.

In addition, to evaluate the running efficiency of different models, we make a comparison of different models in terms of the model size (MB) and the average running time (seconds per image), as presented in Table 2. Concretely, the average running time is computed by executing models on the SD-Saliency-900 dataset. It can be seen that our model runs fastest when compared with other models, where our model takes about 0.021s when handling a 200 × 200 image. For the model size, we can find that our model size is 378MB, which is slightly large when compared with the top-performance models. Thus, in our future work, we will attempt to compress the model, and reduce the model size.

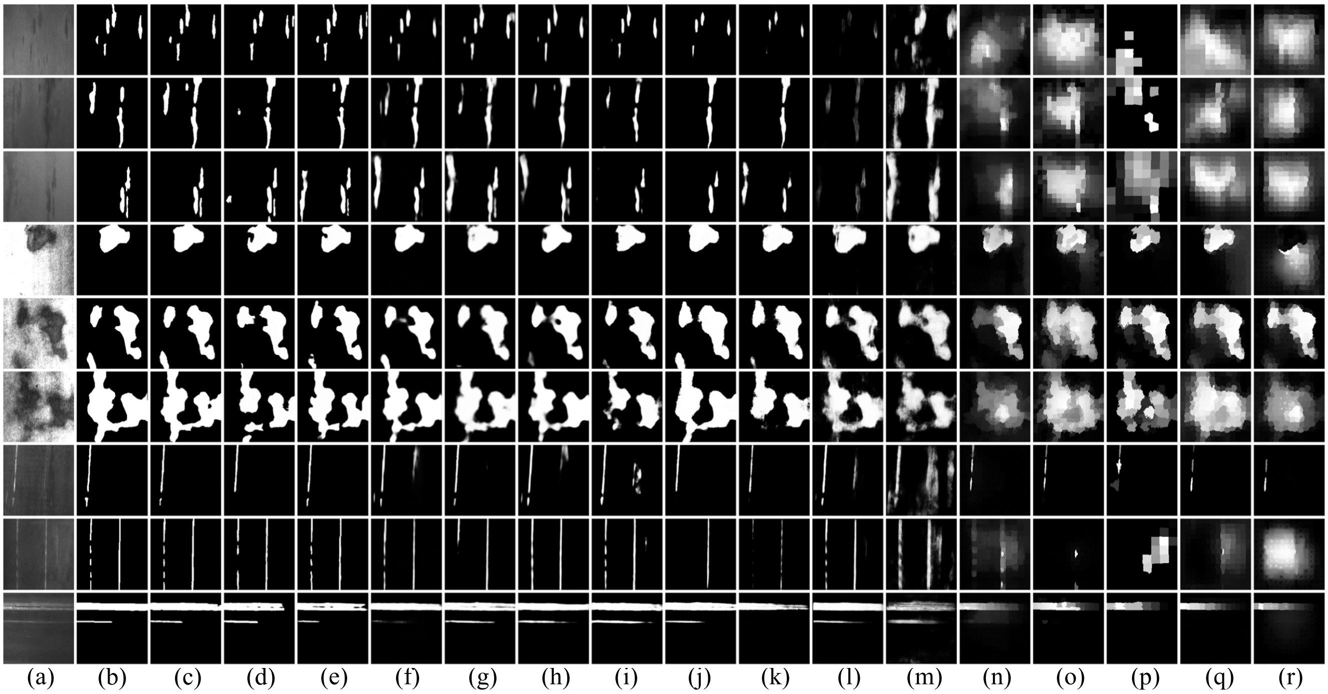


**FIGURE 4.** Visual comparison of saliency maps: (a) Input image, (b) Ground truth, (c) Ours, (d) EDRNet [17], (e) BASNet [28], (f) CPD [29], (g) PiCANet [47], (h) PoolNet [39], (i) BMPMBMPM [46], (j) R3Net [30], (k) DSS [45], (l) NLDF [44], (m) PFANet [48], (n) MIL [43], (o) SMD [42], (p) BC [23], (q) 2LSG [63], (r) RCRR [41].

The qualitative comparison results are shown in Figure 4, where the examples of the top three rows, the middle three rows, and the bottom three rows are selected from three types of defects (*i.e.* inclusion, patches, and scratches), respectively. It can be found that the results of our model are the closest one to the ground truth when compared with other models' results. Specifically, firstly, the examples of the top three rows present low contrast and scattered attributions. We can find that only our model provides complete and accurate results which are capable of highlighting the defects. By contrast, other models either falsely highlight the backgrounds or cannot detect complete defect regions. For example, in the third row, other models mistakenly recognize the background regions as the defects, while our model can distinguish the defects accurately. Secondly, for the middle three rows, the defect regions are large and the backgrounds are cluttered in the three examples. Fortunately, our model still give a perfect prediction for the defect regions. In contrast, the traditional models often highlight the backgrounds as shown in Figure 4(n-r), and the deep learning based models either incorrectly pop-out backgrounds or incompletely detect defects as shown in Figure 4(d-m). Lastly, from the $7^{th}$ row to the $9^{th}$ rows, the examples are with fine structures. It can be seen that our model still performs best, where the results shown in Figure 4(c) are with clear details. By contrast, most models fail to detect the salient objects, where they often loss parts of defect regions and even falsely highlight backgrounds. Therefore, the qualitative comparison results demonstrate the effectiveness and superiority of our model again.

### C. ABLATION STUDIES

To illustrate the effectiveness of our model and demonstrate the rationality of the design of our model, we give a comprehensive ablation studies shown in Table 3, where the quantitative comparison results are conducted in terms of five metrics including MAE, WF, OR, SM, and PFOM. Firstly, we design several variations of our model. Concretely, as depicted in Table 3, the "Baseline (B)" denotes the basic encoder-decoder network without any other components, which only contains Conv-E1∼E6, Conv-B, and Conv-D6∼D1. The final saliency map can be generated by deploying a 3 × 3 convolutional layer and a sigmoid activation function to the output of Conv-D1. "IFI" means interactive feature integration. "E" means the introduction of edge information. "Sup" denotes the deep supervision adopted by our network. Here, our model is denoted by "B+IFI+E+Sup", "B+IFI+Sup" means our model without edge extraction branches, "B+E+Sup" denotes our model without interactive feature integration, and "B+Sup" is the basic network with deep supervision. Correspondingly, "B+IFI+E" denotes our model without deep supervision, "B+IFI" means our model without edge extraction branches and deep supervision, "B+E" refers to our model without interactive feature integration and deep supervision.

From Table 3, we can find that our model achieves the best performs when compared with other variations in terms of MAE, WF, OR, SM, and PFOM. Particularly, compared with the baseline network (B), the WF, OR, SM, and PFOM of our model are improved by 1.2%, 2.2%, 1.0%, and 1.0%,

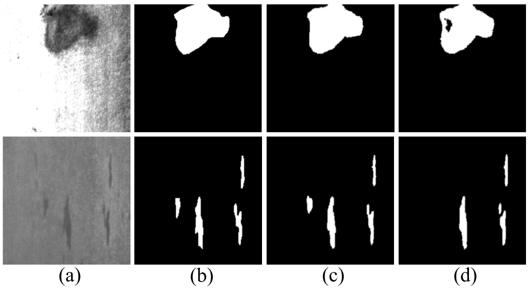


**FIGURE 5.** Qualitative comparisons between our model and the variant "B+IFI+Sup". (a): Input images, (b): GT, (c): Ours, (d): B+IFI+Sup.

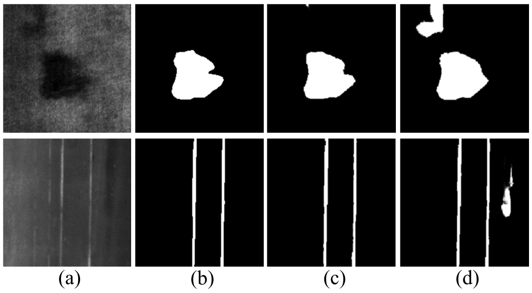


**FIGURE 6.** Qualitative comparisons our model and the variant "B+E+Sup". (a): Input images, (b): GT, (c): Ours, (d): B + E + Sup.

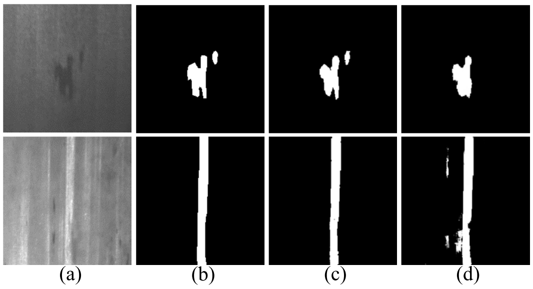


**FIGURE 7.** Qualitative comparisons our model and the variant "B+E+IFI". (a): Input images, (b): GT, (c): Ours, (d): B + E + IFI.

and the MAE is decreased by 16.2%. This can demonstrate the effectiveness of all components of our model, and further validate the rationality of the proposed network.

Besides, we also provide the qualitative comparison between our model and the variations including "B+IFI+Sup", "B+E+Sup", and "B+IFI+E", namely our model without edge, our model without interactive feature integration, and our model without deep supervision, as presented in Fig. 5, Fig. 6, and Fig. 7. To be specifically, firstly, comparing with the "B+IFI+Sup" shown in Fig. 5(d), we can find that the results of our model are more complete. Secondly, comparing with the "B+E+Sup" presented in Fig. 6(d), we can find that our model suppress the backgrounds effectively. Thirdly, comparing with the "B+IFI+E" depicted in Fig. 7(d), it is obviously that our model performs better. This presents the efforts of edge information, the interaction of different level features, and the deep supervision, where the edge indicates an accurate location cue of defect regions, the feature interaction gives a well depiction for defects, and deep supervision gives an effective constraint for feature learning. Thus, from Fig. 5, Fig. 6, and Fig. 7, we can prove the effectiveness of the crucial components of our model, and demonstrate the rationality of the design of our model.

## V. CONCLUSION

This paper proposes a novel saliency model, *i.e.* Edge-aware Multi-level Interactive Network, to pop-out defects on the strip steel surface. Specifically, the proposed network adopts an U-shape architecture where the two points are the interactive feature integration and the edge-guided saliency fusion. Firstly, for each level of the network, we fuse the features from the current level of encoder, the adjacent levels of encoder, and previous decoder stage. Particularly, the features of adjacent layers promote the flow of object cues, which is benefiting for the depiction of defects. Secondly, to acquire a saliency result with precise boundaries, we extract edge information together with saliency prediction at each decoder block. After that, the fusion of edge cues and saliency results provides a complete and accurate saliency map which can effectively highlight the defect regions from the strip steel surface. Comprehensive experiments are conducted on the public dataset, and the quantitative and qualitative results demonstrate the effectiveness of our model which consistently outperforms the state-of-the-art models in all evaluation metrics.

## REFERENCES

[1] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," *Appl. Surf. Sci.*, vol. 285, no. 21, pp. 858–864, Nov. 2013.

[2] 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.

[3] C. Sun, X. Zhang, Q. Zhou, and Y. Tian, "A model predictive controller with switched tracking error for autonomous vehicle path tracking," *IEEE Access*, vol. 7, pp. 53103–53114, 2019.

[4] Y. Zhao, Q. Chen, W. Cao, J. Yang, J. Xiong, and G. Gui, "Deep learning for risk detection and trajectory tracking at construction sites," *IEEE Access*, vol. 7, pp. 30905–30912, 2019.

[5] Z. Zeng, W. Xie, Y. Zhang, and Y. Lu, "RIC-Unet: An improved neural network based on Unet for nuclei segmentation in histology images," *IEEE Access*, vol. 7, pp. 21420–21428, 2019.

[6] J. Kang and J. Gwak, "Ensemble of instance segmentation models for polyp segmentation in colonoscopy images," *IEEE Access*, vol. 7, pp. 26440–26447, 2019.

[7] S. Athar and Z. Wang, "A comprehensive performance evaluation of image quality assessment algorithms," *IEEE Access*, vol. 7, pp. 140030–140070, 2019.

[8] L. Li, Y. Yan, Z. Lu, J. Wu, K. Gu, and S. Wang, "No-reference quality assessment of deblurred images based on natural scene statistics," *IEEE Access*, vol. 5, pp. 2163–2171, 2017.

[9] C. Li, G. Gao, Z. Liu, M. Yu, and D. Huang, "Fabric defect detection based on biological vision modeling," *IEEE Access*, vol. 6, pp. 27659–27670, 2018.

[10] R. Wang, Q. Guo, S. Lu, and C. Zhang, "Tire defect detection using fully convolutional network," *IEEE Access*, vol. 7, pp. 43502–43510, 2019.

[11] Q. Huangpeng, H. Zhang, X. Zeng, and W. Huang, "Automatic visual defect detection using texture prior and low-rank representation," *IEEE Access*, vol. 6, pp. 37965–37976, 2018.

[12] X. Zhou, Y. Wang, C. Xiao, Q. Zhu, X. Lu, H. Zhang, J. Ge, and H. Zhao, "Automated visual inspection of glass bottle bottom with saliency detection and template matching," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 11, pp. 4253–4267, Nov. 2019.

[13] L. Xu, H. Xu, X. Li, and M. Pan, "A defect inspection for explosive cartridge using an improved visual attention and image-weighted eigenvalue," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1191–1204, Apr. 2020.

[14] Y. He, K. Song, Q. Meng, and Y. Yan, "An End-to-End steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1493–1504, Apr. 2020.

[15] H. Yu, Q. Li, Y. Tan, J. Gan, J. Wang, Y.-A. Geng, and L. Jia, "A coarse-to-fine model for rail surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 3, pp. 656–666, Mar. 2018.

[16] 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. 2019.

[17] 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.

[18] 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.

[19] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 733–740.

[20] D. Gao, V. Mahadevan, and N. Vasconcelos, "The discriminant center-surround hypothesis for bottom-up saliency," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2007, pp. 497–504.

[21] R. Ju, L. Ge, W. Geng, T. Ren, and G. Wu, "Depth saliency based on anisotropic center-surround difference," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 1115–1119.

[22] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer, 2012, pp. 29–42.

[23] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 2814–2821.

[24] X. Zhou, Z. Liu, C. Gong, and W. Liu, "Improving video saliency detection via localized estimation and spatiotemporal refinement," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 2993–3007, Nov. 2018.

[25] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2083–2090.

[26] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via bootstrap learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1884–1892.

[27] K. Fu, I. Y.-H. Gu, and J. Yang, "Saliency detection by fully learning a continuous conditional random field," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1531–1544, Jul. 2017.

[28] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: Boundary-aware salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7479–7489.

[29] 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.

[30] Z. Deng, X. Hu, L. Zhu, X. Xu, J. Qin, G. Han, and P.-A. Heng, "R3net: Recurrent residual refinement network for saliency detection," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 684–690.

[31] C. Chen, J. Wei, C. Peng, and H. Qin, "Depth-Quality-Aware salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 2350–2363, 2021.

[32] C. Chen, G. Wang, C. Peng, Y. Fang, D. Zhang, and H. Qin, "Exploring rich and efficient spatial temporal interactions for real-time video salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 3995–4007, 2021.

[33] C. Li, R. Cong, J. Hou, S. Zhang, Y. Qian, and S. Kwong, "Nested network with two-stream pyramid for salient object detection in optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9156–9166, Nov. 2019.

[34] L. Pan, X. Zhou, R. Shi, J. Zhang, and C. Yan, "Cross-modal feature extraction and integration based RGBD saliency detection," *Image Vis. Comput.*, vol. 101, Sep. 2020, Art. no. 103964.

[35] X. Zhou, G. Li, C. Gong, Z. Liu, and J. Zhang, "Attention-guided RGBD saliency detection using appearance information," *Image Vis. Comput.*, vol. 95, Mar. 2020, Art. no. 103888.

[36] 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.

[37] X. Zhou, H. Wen, R. Shi, H. Yin, and C. Yan, "Depth-guided saliency detection via boundary information," *Image Vis. Comput.*, vol. 103, Nov. 2020, Art. no. 104001.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Cham, Switzerland: Springer, 2015, pp. 234–241.

[39] 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 Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3917–3926.

[40] 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.

[41] 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.

[42] 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.

[43] F. Huang, J. Qi, H. Lu, L. Zhang, and X. Ruan, "Salient object detection via multiple instance learning," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1911–1922, Apr. 2017.

[44] 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 Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2017, pp. 6609–6617.

[45] 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.

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

[47] N. Liu, J. Han, and M. H. Yang, "PiCANet: Learning pixel-wise contextual attention for saliency detection," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Aug. 2018, pp. 3089–3098.

[48] 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.

[49] G. Ma, S. Li, C. Chen, A. Hao, and H. Qin, "Rethinking image salient object detection: Object-level semantic saliency reranking first, pixel-wise saliency refinement later," *IEEE Trans. Image Process.*, vol. 30, pp. 4238–4252, 2021.

[50] 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.

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

[52] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.

[53] S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 234–250.

[54] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji, "Detect globally, refine locally: A novel approach to saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 3127–3135.

[55] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4320–4328.

[56] 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.

[57] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Ann. Oper. Res.*, vol. 134, no. 1, pp. 19–67, 2005.

[58] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *Proc. Int. Symp. Vis. Comput.* Cham, Switzerland: Springer, 2016, pp. 234–244.

[59] 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. 2003, pp. 1398–1402.

[60] 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.

[61] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, vol. 9, 2010, pp. 249–256.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[63] 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.

[64] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 248–255.

[65] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proc. ICCV*, Oct. 2017, pp. 4548–4557.

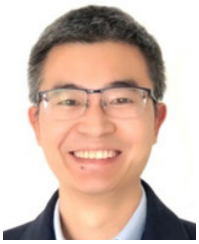
[66] I. E. Abdou and W. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," *Proc. IEEE*, vol. 67, no. 5, pp. 753–763, May 1979.

**XIAOFEI ZHOU** received the Ph.D. degree from Shanghai University, Shanghai, China, in 2018. He is currently a Lecturer with the School of Automation, Hangzhou Dianzi University, Hangzhou, China. His research interests include saliency detection, video segmentation, and video quality assessment.

**HAO FANG** is currently pursuing the B.E. degree with the School of Automation, Hangzhou Dianzi University, Hangzhou, China. His research interests include computer vision, visual saliency analysis, and defect detection.

**XIAOBO FEI** is currently pursuing the B.E. degree with the School of Automation, Hangzhou Dianzi University, Hangzhou, China. Her research interests include computer vision, visual saliency analysis, and video quality assessment.

**RAN SHI** received the B.S. degree in electronic science and technology from the Changshu Institute of Technology, Suzhou, China, in 2009, the M.S. degree in signal and information processing from Shanghai University, Shanghai, China, in 2012, and the Ph.D. degree from the Department of Electronic Engineering, The Chinese University of Hong Kong. He is currently an Assistant Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include object segmentation visual quality evaluation, interactive segmentation, and salient object detection.



**JIYONG ZHANG** received the B.S. and M.S. degrees in computer science from Tsinghua University, in 1999 and 2001, respectively, and the Ph.D. degree in computer science from the Swiss Federal Institute of Technology at Lausanne (EPFL), in 2008. He is currently a Distinguished Professor with Hangzhou Dianzi University. His research interests include artificial intelligence, machine learning, data mining, and image processing.