Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 3751053, 12 pages
https://doi.org/10.1155/2020/3751053

Research Article

Deeper and Mixed Supervision for Salient Object Detection in Automated Surface Inspection

Senbo Yan, Xiaowen Song, and Guocong Liu

State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China

Correspondence should be addressed to Xiaowen Song; songxw@zju.edu.cn

Received 19 September 2019; Revised 30 December 2019; Accepted 16 January 2020; Published 25 February 2020

Academic Editor: Daniel Zaldivar

In recent years, researches in the field of salient object detection have been widely made in many industrial visual inspection tasks. Automated surface inspection (ASI) can be regarded as one of the most challenging tasks in computer vision because of its high cost of data acquisition, serious imbalance of test samples, and high real-time requirement. Inspired by the requirements of industrial ASI and the methods of salient object detection (SOD), a task mode of defect type classification plus defect area segmentation and a novel deeper and mixed supervision network (DMS) architecture is proposed. The backbone network ResNeXt-101 was pretrained on ImageNet. Firstly, we extract five multiscale feature maps from backbone and concatenate them layer by layer. In addition, to obtain the classification prediction and saliency maps in one stage, the image-level and pixel-level ground truth is trained in a same side output network. Supervision signal is imposed on each side layer to realize deeper and mixed training for the network. Furthermore, the DMS network is equipped with residual refinement mechanism to refine the saliency maps of input images. We evaluate the DMS network on 4 open access ASI datasets and compare it with other 20 methods, which indicates that mixed supervision can significantly improve the accuracy of saliency segmentation. Experiment results show that the proposed method can achieve the state-of-the-art performance.

1. Introduction

Surface defect detection based on computer vision is an important task in the industry. In traditional cases, object surface defect detection is performed by the human eye. However, such artificial recognition-based detection methods are highly subjective, time-consuming, and lack of accuracy. To overcome the limitations of manual inspection, automatic surface inspection (ASI) technology arises to replace human decision.

In industry, automatic surface inspection task is detecting local anomalies in uniform textures. These textures can be divided into uniform textures and uneven textures. Surface inspection objects include steel [1], wood [2], stone [3], ceramic tile [4], and fabric [5].

To achieve automatic surface inception, many image processing-based methods have been proposed. Traditional ASI methods can be mainly divided into four categories: structural method, statistical method, filter-based method, and model-based method [6]. Structural methods simulate primitives and displacements and are often used in repetitive patterns, including roughness measurements, boundary features, and morphology [7]. Statistical methods, which measure the distribution of pixel values, are commonly used in the detection of random textures (wood, castings, and tiles), including histogram method [8], local binary pattern (LBP) [9], and gray-level co-occurrence matrix (GLCM) [10]. The filter-based method, which can be divided into the spatial domain method [11] and frequency domain method [12], directly applies a filter bank to the texture patterns. And the model-based approach builds a complete representation of the defect by modeling multiple features of the defect [4]. In general, despite the wide variety of automated surface inspection methods, the purpose of these traditional methods is to construct templates or features of the image. Model performance depends on the accuracy of modeling defects, which means the generalization ability of the model has great limitations.

In recent years, convolution neural networks have been widely used in computer vision tasks such as image classification, object detection, semantic segmentation, and salient object detection. The neural network has also become the mainstream of automatic surface inspection tasks. It has powerful image feature learning and generalization ability, which avoids the defect that the traditional ASI method relies too much on hand craft design features. In addition, the generic surface defect detection model has become possible. Park et al. [13] designed a simple CNN network to classify the surface defects of six different materials, which is far superior to the traditional feature extraction with classifier. Weimer et al. [14] used the CNN models to semantically segment the defect regions of the repeated texture patterns in the DAGM2007 dataset, and the segmentation performance was significantly improved.

With the introduction of neural network models and the continuous improvement in algorithm performance, the ASI task model is also evolving. The ASI task evolved from image-level defect recognition classification to finer-grained pixel-level segmentation or object detection.

Ren et al. [15] redefined the task mode of ASI task classification plus segmentation. Based on the Decaf network, they build a general automatic surface detection method. Then they perform image-level defect recognition and segment the defect area at the pixel level by a pixel-by-pixel hot zone algorithm. As the ASI dataset has the characteristics of clear defect categories and clear foreground background in a single picture, comparing with the general saliency segmentation in the free scene or the more complex semantic segmentation in the foreground, the task model of defect classification plus saliency segmentation is obviously a more reasonable choice. To solve the surface defect detection task of magnetic tile, Huang et al. [16] designed a surface defect detection network based on neural network and saliency detection method, realizing the real-time detection of surface defects of the magnetic tile. This research also fully demonstrates that the introduction of image saliency detection can greatly help solve the ASI task.

Although the deep learning method has achieved remarkable results in many computer vision tasks, its application of the ASI tasks has been limited by various factors. First, the deep learning method requires hundreds and thousands of training data to ensure the training effect of the model and prevent over-fitting. However, the collection of images in industrial scenes is difficult and expensive, only a few hundred or even dozens of images in the ASI dataset. In addition, unlike the general scene target detection and saliency detection tasks, most of the samples in practical industrial applications are negative samples, and it is expensive and inefficient to directly perform image detection or segmentation on all samples. Therefore, it is of great significance to discuss how to efficiently and accurately identify and segment surface defect regions with neural network. Finally, different from the saliency detection in natural scenes, the foreground of ASI tasks is usually the small-scale targets that are difficult to detect in traditional algorithms, such as holes and cracks. How to effectively divide these small-scale targets is also a huge algorithm challenge faced by ASI tasks.

In this paper, ASI task is defined as defect classification plus defect area segmentation. We propose the deeper and mixed supervision network (DMS), an innovative generic surface defect method to fulfill the multiscale classification and salient defects detection in one stage. To achieve this, as Figure 1 illustrates, we first extract five different layers as side outputs from the backbone network and then integrate them into three levels of feature maps. Second, we concatenate different level feature maps layer by layer in side outputs. Finally, we impose image-level ground truth and pixel-level ground truth in each feature layer to realize deeper and mixed supervision. In training, we designed a loss function to balance the weight of the classification and saliency segmentation. Finally, we refine the residual by the multi-bypass output of the DMS network to obtain the classification prediction and pixel-by-pixel prediction of the object defect. The test results applied to the four ASI open source data sets show that the mixed supervision mechanism of the DMS network can improve the saliency segmentation results. In the classification and segmentation tasks, the models we propose can achieve the best results of the current ASI tasks.

In summary, our contributions are four folds:

(i) Based on the multilevel side output architecture of HED, we propose a novel deep network architecture, i.e., DMS (including SDMS and BDMS) network, which combines the recurrent high-medium-low feature concatenate and residual refinement mechanism.

(ii) We propose a mixed supervision mechanism, which can fulfill defect classification and foreground segmentation in one stage. Besides, mixed supervision significantly improved the performance of SOD. We also propose our loss function to balance the weight of classification and foreground segmentation. In addition, mixed supervision provides a solution for processing the nonsalient samples, which is one of the most challenging tasks in generic SOD.

(iii) We further explore the industrial application of salient object detection, while most of the current application focuses on wild scenes.

(iv) We evaluate our network in four ASI datasets and compare it with other methods. Overall, DMS network reaches the state-of-the-art performance for SOD in ASI.

2. Related Work

2.1. Fully Supervised Salient Object Detection. Saliency detection is a detection method that defines image content as background and foreground, detects the foreground according to the salient features, and divides it pixel by pixel. Many traditional methods are employed, fusing handcrafted features for salient object detection [17, 18]. In recent years, neural network algorithms, especially fully convolutional neural networks (FCN) [19], have dominated many fields of computer vision due to its convincing performance.

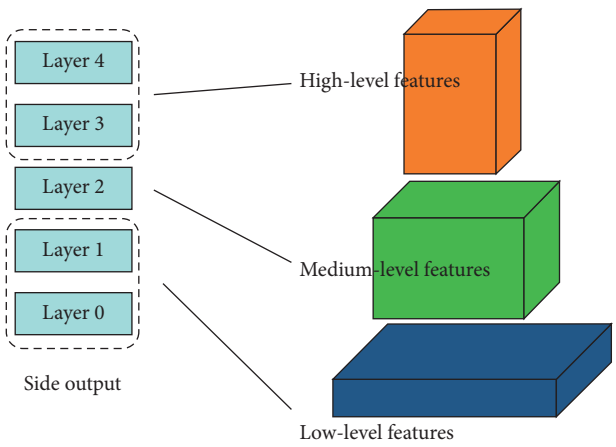

Figure 1: The setting of the side outputs.

For example, Zhang et al. [20] proposed a novel FCN-based structure to learn deep uncertain convolutional features and significantly encourage the robustness and accuracy of saliency detection. However, these SOD models generally require a large number of pixel-level images for training and can only give a foreground inference, which cannot meet the ASI task requirements. In contrast, we use image-level ground truth for enhanced training in our approach, and we can get defect classification and saliency segmentation results simultaneously.

2.2. Salient Object Detection with Image-Level Supervision. In early saliency segmentation tasks, training data were typically based entirely on expensive pixel-level image data, while data with image-level ground truth were rarely used in saliency detection. This is because typically the task of an image-level ground truth focuses on the category of the object in the image rather than the specific location, while the saliency detection is intended to detect the full extended area of the foreground object and ignore its specific category. However, research by Wang et al. [21] shows that image classification and saliency segmentation tasks are essentially interrelated, as the candidate regions provided by saliency detection help classify more accurately, while the categories provided by image-level ground truth are likely to be the foreground of the image. This method approaches or even surpasses the state-of-the-art fully supervised model by using image-level labels at that time. Particularly, in ASI tasks, the categories of defects are limited and normally have quite different features. It indicates that image-level ground truth may be more useful when inferring the foreground areas. WSS fully demonstrates the contribution to image-level supervision in saliency detection, which provides inspiration for our DMS network.

2.3. Feature Concatenate and Dense Supervision Refinement. Feature concatenate and shortcut connection are one of the hotspots of neural network model research in recent years. He et al. [22] who proposed ResNet is the first to propose the mechanism of shortcut, which is a challenge to the traditional neural network with the connection only between two adjacent layers. On this basis, DenseNet [23] applies more dense connections and bypass setting with the assumption that feature concatenate is a better learning method than its multiple learning redundancy features. In object detection, the FPN satisfies the requirements of the detection task by combining the location information of the low-level feature map and the classification information of the high-level feature map. These studies have fully proved that the potential of the feature layer in the traditional neural network has not been fully explored. Recently, many saliency detection models have enhanced the detection results by combining low-level structural features and high-level semantic features through short connection and obtained obvious effects.

Deng et al. [24] designed a residual refinement block (RRB). They concatenate the input rough saliency map with the depth feature layer, and the residual map is output and supervised to form a new saliency map, which is used as the input map for the next round of circular refinement. R3Net achieves the refinement of the saliency maps by repeatedly concatenating the high- and low-layer features, which improves the effect of the saliency detection. Zhang et al. [25] studied how to better aggregate multilevel convolutional feature maps for salient object detection. They proposed a novel structure to combine the multilevel feature maps at each resolution and predict saliency maps with the combined features. Those convincing studies indicate that multilevel feature maps that are generated by FCN are complementary.

Most recently, a large number of edge information enhancement methods have been proposed [26–30]. Zhao et al. [26] proposed to use the complementarity of edge information and saliency information to enhance the boundary and location information of saliency objects. Wu et al. [30] combined the SOD with edge detection and developed a novel mutual learning module (MLM) to help the foreground contour and edge detection tasks guide each other simultaneously. It is obvious that reasonable additional information is a useful complement to the SOD task.

The DMS network proposed in this paper combines the mechanism of multilevel feature concatenate, deep supervision, and residual refinement. The DMS backbone network is divided into three feature layers of low, medium, and high, and the network performs multiscale feature concatenate by means of short connection. The multi-bypass configuration satisfies the requirements of deep supervision and residual refinement.

3. Methodology

We show the structure of single deeper and mixed supervision network (SDMS) in Figure 2. Figure 3 shows the proposed structure of the bilateral deeper mixed supervision (BDMS) network. We first select five different scales feature maps of input images as side outputs through ResNeXt101 backbone network. The side outputs of different layers contain low-level details and high-level semantic information, respectively. We consider that the first-layer and the

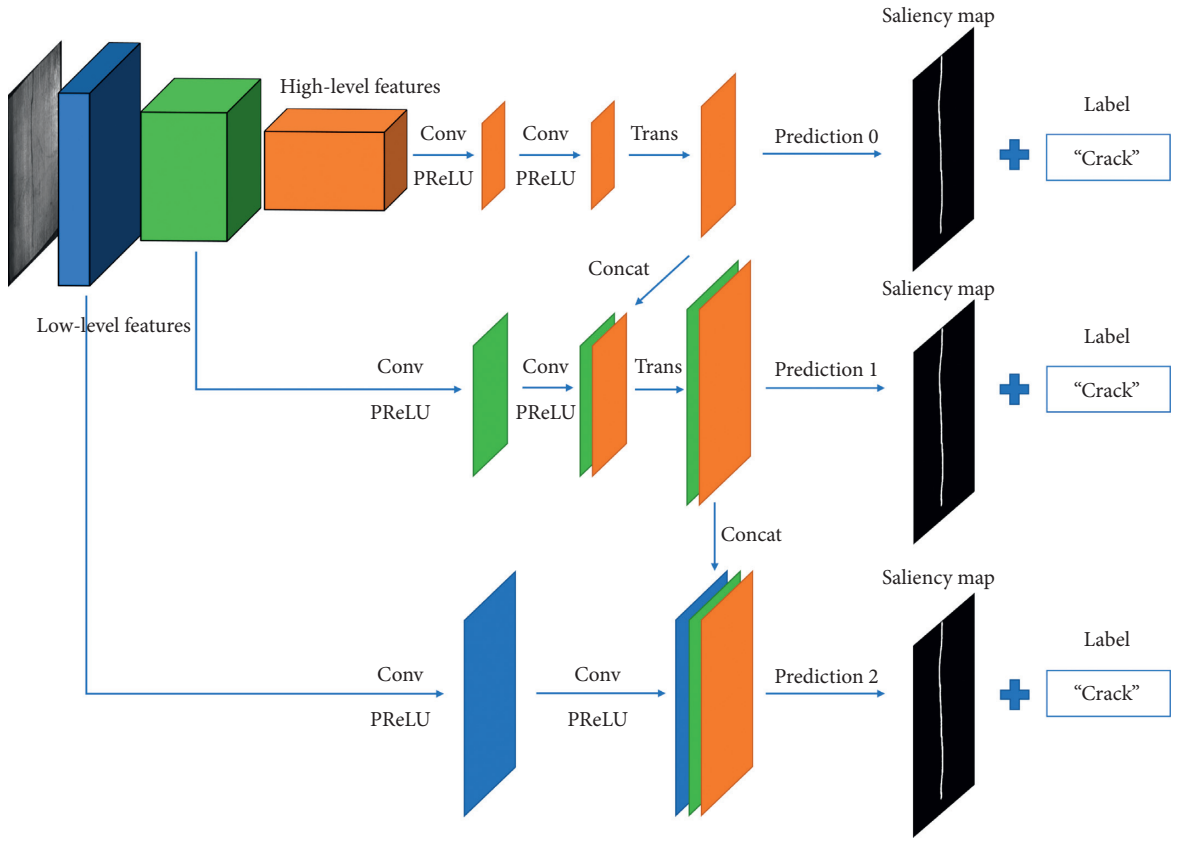

Figure 2: Structure of SDMS network.

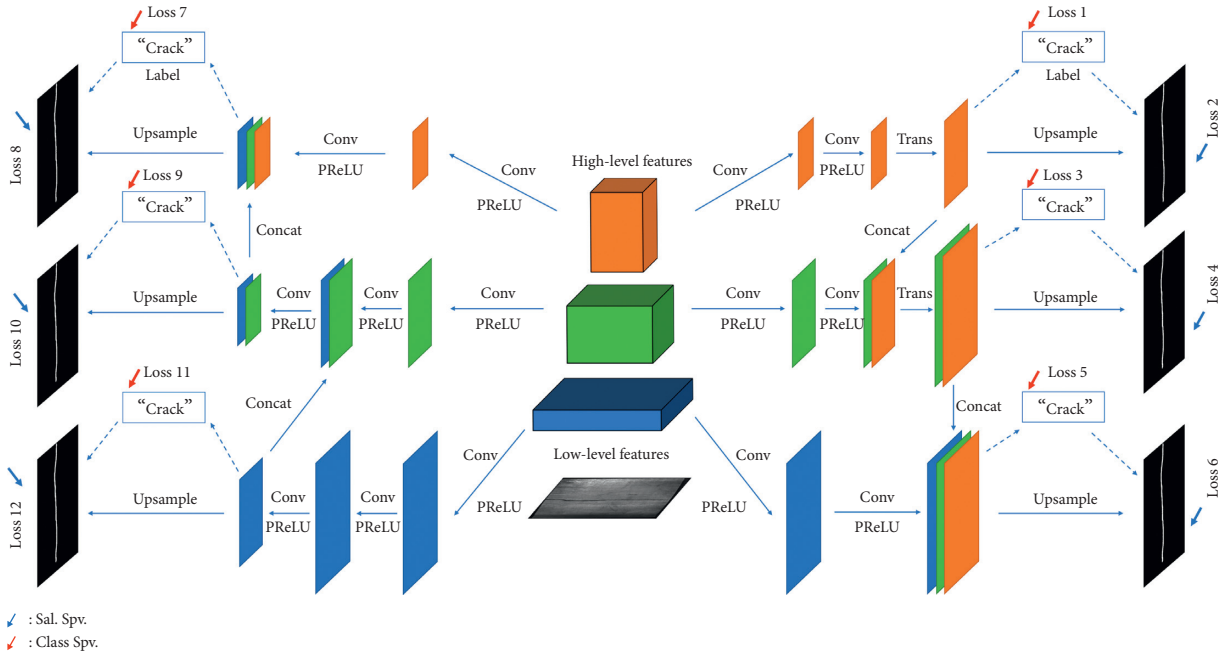

Figure 3: Schematic diagram of the BDMS network.

second-layer feature maps are integrated as the low-level feature (LF), third-layer feature maps are used as the middle-level feature (MF), and the fourth-layer and fifth-layer feature maps are integrated as the high-level feature (HF). For each feature layer (side output), we set an independent convolution filter to generate connectable feature maps and corresponding saliency map, and the parameters corresponding to each feature layer are as shown in Table 1. We

Table 1: Parameter of each side output.

Side output		Channel	Input	Conv filters			Feature maps
LF	Layer 0	64	75×75	Conv3@256	Conv3@256	Conv1@256	256@75×75
	Layer 1	256	75×75				
MF	Layer 2	512	38×38	Conv3@256	Conv3@256	Conv1@256	256@38×38
HF	Layer 3	1024	19×19	Conv3@256	Conv3@256	Conv1@256	256@19×19
	Layer 4	2048	10×10				

use the high-level features to generate the original saliency map and classification information and then concatenate the middle-level features and the original saliency map to generate detailed saliency maps and classification information. We finally concatenate the low-level features to generate saliency maps. We design a mixed loss function to adjust the loss weight of the saliency segmentation and defect classification and supervise the training of each level of the saliency map and the classification signal. In the SDMS and BDMS, we, respectively, adopt the saliency map after third-level/six-level feature reuse and residual refinement as the final output saliency map and obtain the image classification prediction. In the following subsections, we will elaborate on the specific architecture of SDMS and BDMS, weighted mixing losses, and the detail of surface defect detection task when the proposed model is applied.

3.1. Deeper and Mixed Supervision. The ultimate goal of this paper is to achieve classification of defect categories and saliency segmentation of defect regions. However, unlike general SOD tasks, there are too many normal samples in the ASI task, i.e., nonsalient samples. However, most of the current SOD methods have neglected the large number of nonsalient samples, especially in ASI. It is worth mentioning that the recent research by Fan et al. [31] also shows that most of the current SOD data sets and researchers' selective neglect of nonsalient samples lead to a large number of SOD models to perform huge difference in real-world scenes. In order to solve this drawback and combine the actual requirements of the ASI task, this paper proposes a mixed supervisory model based on multilevel side output implementation, which uses image-level weak supervised labels to enhance the saliency detection effectiveness and can effectively classify and process nonsalient samples in actual ASI tasks, and significantly enhances the ability of the SOD model to process nonsalient samples.

In Figure 3, we use the ResNeXt101 as backbone network to extract five sets of different scale feature maps of the input image as the side output. The first-layer and second-layer feature maps are combined as the middle-level feature (LF), the third-layer feature maps are used as the middle-level feature (MF), and the fourth-layer and fifth-layer feature maps are combined into the high-level feature (HF). We use the advanced features to generate the original saliency map and classification information, then concatenate the middle-level features and the original saliency map to generate detailed saliency maps and classification information, and finally concatenate the low-level features and generate saliency maps. Both of the saliency maps will be upsampled to the input size. Supervision signals were imposed on each level of saliency maps and classification results.

3.2. Improved Side Output Architecture. It is generally believed that, in neural networks, low-level features contain more detailed information, while higher-level features contain more semantic and positioning information, so the weighted average results of extracting multiscale side outputs in detection tasks tend to get better test results. Multiscale side output supervision was initially widely used in areas such as edge detection [32]. Hou et al. [33] obtained an improved DSS architecture based on the HED architecture by combining a specific short-connection structure with side outputs of different scales and successfully applied in the field of saliency detection. However, in DSS, since the side output layer is compressed into a single channel before making a short connection, there may be significant information loss between the short connections [34]. From the perspective of improving the efficiency of feature reuse, we do not intend to fully refer to the short-connection architecture in DSS but try to make a more complete concatenate of the side output.

The backbone of the DMS network uses a similar processing approach to the classic HED [32] architecture, taking five side output layers from different depths of the backbone network to ensure multiscale characteristics of the side output layer. We observe that, in the previous study, no researchers discussed the specific meaning of the five-layer side output architecture and they simply classified it into different information aggregations provided by low-level features and high-level features. In order to make the meaning of the side output layer more typical, we aggregate side output layers 0 and 1 into low-level feature (LF), side output layer 2 into the middle-level feature (MF), and side output layers 3 and 4 into high-level feature (HF) as shown in Figure 2. Since the shallower feature layer has a larger size, we upsample the relatively deep feature layer to the same size and then join them as follows:

$$\begin{aligned} \mathrm{LF} &= f_{\mathrm{conv}}\left(\mathrm{cat}\left(L_0, \mathrm{up}\left(L_1\right)\right)\right), \\ \mathrm{MF} &= f_{\mathrm{conv}}\left(L_2\right), \\ \mathrm{HF} &= f_{\mathrm{conv}}\left(\mathrm{cat}\left(L_3, \mathrm{up}\left(L_4\right)\right)\right), \end{aligned} \tag{1}$$

where LF, MF, and HF, respectively, represent the low-level features, middle-level features, and high-level features; L_n represents the n-th layer side output; up represents the feature layer upsampling, in the lower layer features L_1 is upsampled to the same scale as L_0, and in the high-level feature, L_4 is upsampled to the same scale as L_3; cat

represents the concatenate of feature layers; and f_{conv} represents a set of convolutional layers used to aggregate feature layers. On the one hand, reducing the number of feature layers effectively reduces the computational cost of side output stitching. On the other hand, the side output layer is more visual, which helps us to further discuss the association between the side output layers.

In the SDMS network, we first start from the high-level feature HF as F_0; after a set of transposed convolutions (so that the feature layer has the same scale as the next layer); it is upsampled to obtain the primary saliency map S_0; and then we aggregate the after-transposed-convolution high-level feature layer HF and middle-level feature layer MF into F_1. Next, we draw on the idea of residual refinement [24], stitching the primary saliency map S_0 and the feature layer F_1 into the residual layer R_1. The next saliency map S_1 is obtained by refining the initial saliency map S_0 by R_1. The specific definition is as follows:

$$\begin{aligned} F_i &= \varphi_{\text{trans}}\left(\text{cat}\left(F_{i-1}, F\right)\right), \\ R_i &= f_{\text{conv}}\left(\text{cat}\left(F_i, S_{i-1}\right)\right), \\ S_i &= S_{i-1} \oplus R_i, \end{aligned} \tag{2}$$

where F_i denotes the i-th feature layer; F represents the corresponding side output layer (including HF, MF, and LF); R_i represents the i-th residual layer (the number of channels is 1); S_i represents the i-th saliency map; and ϕ_{trans} indicates the transposed convolution corresponding to the size of the next-level feature layer.

When using the above formula for calculation, an obvious problem appears that, after the multilayer residual refinement, the constantly superimposed feature layer F_i generated for feature concatenate will cause huge computational overhead, so it is difficult to build a deeper architecture using SDMS network. To further optimize the saliency map, we propose the dual-stream architecture, the BDMS network. In BDMS, we use a dual-stream architecture with top-down and bottom-up feature concatenates. Specifically, we retain the resulting saliency map S_i after performing a top-down deep mixed supervision, reset F_i to F, and then perform a second round saliency map refinement from bottom to top:

$$\begin{cases} F_i = \varphi_{\text{trans}}\left(\text{cat}\left(F_{i-1}, F\right)\right), & \text{if } i \neq 3, \\ F_i = F, & \text{if } i = 3. \end{cases} \tag{3}$$

The practice of resetting the side output features effectively reduces the computational overhead of the neural network and provides greater scalability for the DMS network structure. In this paper, we chose the BDMS structure shown in Figure 3 as the final architecture and obtain the sixth-level saliency map as the final result. The experimental result shows that the training time of the model is significantly reduced after reset side output.

3.3. Weighted Loss Function. In order to satisfy the mixed supervision of classification result and saliency maps, we design a weighted mixed loss function for neural network training. We set open source data as $D = \{(X_i, Y_i, Y_i')\}_i^N$, where $X_i = \{X_i^k, k = 1, \ldots, P\}$ and $Y_i = \{Y_i^k, k = 1, \ldots, P\}$, respectively, represent an input image with a pixel value of P and a binarized truth value map and Y_i' represents a classification label corresponding to the image. We design our weighted mixed loss function based on the cross-entropy loss function. In particular, the formula for the weighted mixed loss function in the n-th output is as follows:

$$\begin{aligned} L_n\left(Y_i', Y_i, y_i', y_i\right) = &- \frac{w_1}{P} \sum_k^P \left[Y_i^k \log\, y_i^k + \left(1 - Y_i^k\right) \log\left(1 - y_i^k\right)\right] \\ &- w_2 \sum_{j=1}^N Y_{ij}' \log\, y_{ij}', \end{aligned} \tag{4}$$

where y_i' and y_i represent the classification prediction and the saliency maps, respectively; y_i^k represents the predicted value of the k-th pixel in the saliency maps, while the total number of pixels is P; Y_i^k represents the true value of the k-th pixel, where $Y_i^k = 1$ represents the foreground pixel and $Y_i^k = 0$ represents the background pixel; y_{ij}' represents the classification probability that the image belongs to the j-th class in the N categories; Y_{ij}' represents the true category of the image; w_1 and w_2, respectively, represent the weights of the foreground segmentation and the classification part; and n represents the n-th level output. In our experiments, we set the values of w_1 and w_2 to 1 and 0.01, respectively, to balance the loss function of foreground segmentation and classification.

The above formula shows how we calculate the loss function of the n-th-order output. For the entire neural network, our complete loss function θ is defined as the weighted sum of the output loss functions of each stage:

$$\theta = \sum_{n=1}^N \omega_n L_n, \tag{5}$$

where ω_n and L_n represent the weight and loss functions of the n-th stage output and N depends on the number of layers of feature concatenate and takes value 6 in the BDMS network. In the experiment of this paper, we will not discuss the weight of each feature layer, so ω_n is set to 1 uniformly.

4. Experiments

In this section, we mainly illustrate the experimental parameters and the training test details of the model. We focus on the ASI dataset we used and present the experimental results.

4.1. Training and Inference Settings

4.1.1. Training Parameters. We implement the model based on the PyTorch 0.4.0. All models were trained and tested on an NVIDIA GeForce GTX Titan Xp GPU (12 G Memory). As the number of saliency detection data sets, especially ASI data sets, is usually limited, a trained backbone network is necessary. In this paper, we use the weight of ImageNet pretrained ResNeXt101 [35] network as the initial

parameters of the backbone network. We use the standard stochastic gradient descent (SGD) training network, with the size of each batch is 16, the momentum is set to 0.9, and the weight attenuation is 0.0005. And the initial learning rate is set to 0.001, using a polynomial decay with a power of 0.9. Model training is finished after 20000 iterations, and we save the best and the latest model.

The SOD model is usually trained with the MSRA10K [36] dataset and verified on other datasets. Considering the limited size of the ASI datasets and the particularity of mixed supervision, we divided it into training set and test set in the same ASI dataset. The ASI datasets is introduced in Section 4.3.

4.1.2. Inference. During the testing stage, we input the test image into the trained network and obtain the saliency map and classification result of each side output, without any other preprocessing or postprocessing. We estimate the classification accuracy of the image and the F_β and MAE values of the saliency map while generating image reasoning and save the relatively better results. In general, multilevel optimized saliency map has better metrics.

4.2. Evaluation Metrics. In the classification task, we use the classification accuracy to evaluate the model classification effectiveness. For saliency testing, we use two commonly used metrics, F measure (F_β) and mean absolute error (MAE), to evaluate our DMS network. A good saliency network usually has a larger F_β and a smaller MAE. For a saliency map y with pixel value P, we linearly map the pixel values from [0, 255] to [0, 1] and compare it with the truth map Y. The MAE calculation formula is as follows:

$$\mathrm{MAE} = \frac{1}{P}\sum_{i=1}^{P}\left|y_i - Y_i\right|. \tag{6}$$

And the F measure calculation formula is

$$F_\beta = \frac{\left(1+\beta^2\right)\mathrm{precision}\times\mathrm{recall}}{\beta^2\mathrm{precision}+\mathrm{recall}}. \tag{7}$$

Usually, $\beta^2 = 0.3$ is chosen as the recommended value for the accuracy of the saliency maps. These metrics are discussed in our experimental results below.

4.3. ASI Datasets. In order to verify the effectiveness of the proposed automatic surface defect detection model, we select three surface defect detection public data for experiments. These datasets include the magnetic tile surface defect dataset [16], NEU surface defect dataset [37], rail surface defect dataset [38], and the road crack defect dataset [39].

4.3.1. Magnetic Tile Surface Defect Dataset. The first dataset is magnetic tile surface defect dataset (MTDD) [16]. MTDD contains 1344 images which was divided into six categories, including five types of defect and one type of defect-free map, named as Blowhole, Break, Crack, Fray, Uneven, and Free. Examples of different defect images and ground truth are shown in Figure 4. We divide this dataset into training set (1118 images) and test set (226 images). Both of the training set and test set share the same category distribution.

4.3.2. NEU Surface Defect Database. The second open source dataset is the NEU strip surface defect dataset [37]. The NEU dataset includes six defects of the hot rolled strip surface, including cracks (Cr), inclusions (In), plaques (Pa), pits (Ps), holes (Rs), and scratches (Sc), with 300 images each. Examples of defect images are shown in Figure 5.

4.3.3. Rail Surface Defect Datasets. The third open source dataset is the rail surface defect datasets (RSDDs) [38]. It contains two types of dataset: the first is Type-I RSDDs dataset captured from express rails, which has 67 challenging images, and the second is Type-II RSDDs dataset captured from common/heavy haul rails, which has 128 challenging images. Examples of defect images are shown in Figure 6.

4.3.4. Road Crack Dataset. The fourth open source dataset is the road crack dataset [39]. The road crack dataset does not classify road cracks but provides roadmaps and pixel-level labels, including a total of 151 images. Examples of defect images are shown in Figure 7.

4.4. Ablation Analysis. In order to fully evaluate the performance of mixed monitoring and feature concatenate mechanisms in the ASI dataset, we perform ablation analysis on DMS network. We use the same backbone network and the hyperparameters in a series of simplified SDMS networks and compare them with standard SDMS network to verify the concatenate of the side output and effectiveness of mixed supervision mechanism. The specific setting of the ablation models is shown in Table 2. The results of different models are show in Table 3.

Table 2 shows the test results for different settings. We can draw the following conclusions from the experiment: (1) The accuracy of saliency maps in each side output layer of the standard SDMS network are increased, indicating the effectiveness of the residual refinement mechanism. (2) The comparison result between the standard SDMS and SDMS-A networks proves that side output concatenate is effective to improve the accuracy of salient defect detection. (3) The comparison between the standard SDMS and SDMS-B networks shows that the mixed supervision mechanism has a significant improvement in the accuracy of the detection and proves the conclusion that image-level labels can effectively enhance the saliency segmentation results. (4) The comparison between the standard SDMS and SDMS-C networks shows that the introduction of the ASPP mechanism does not enhance the network effectiveness, meanwhile increased training time for the DMS network by about 40%. Due to this, we finally abandon the ASPP module. (5) Comparison between the standard and SDMS-D indicates that feature

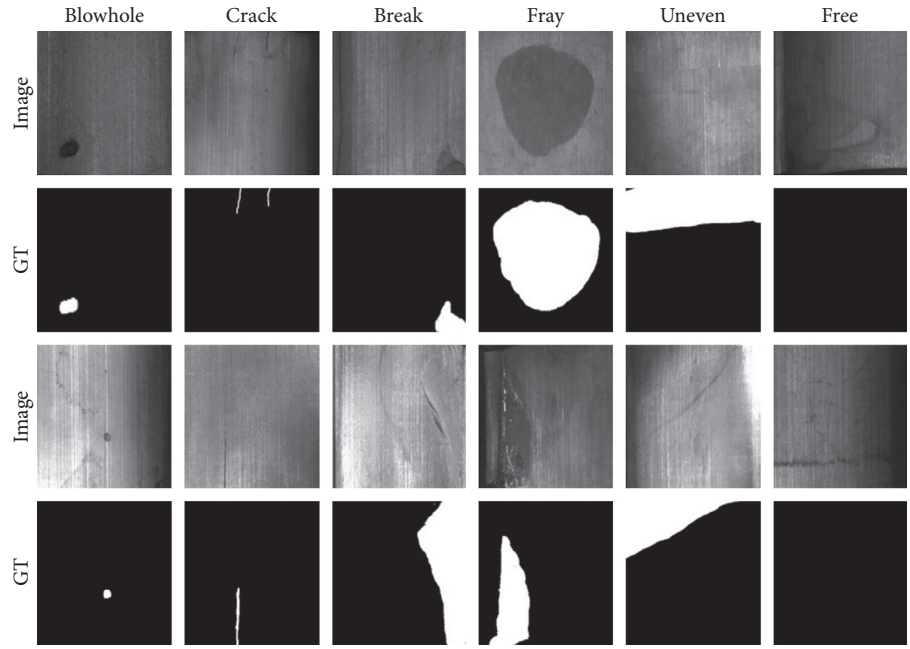

Figure 4: Examples of magnetic tile surface defects (https://github.com/abin24/Magnetic-tile-defect-datasets).

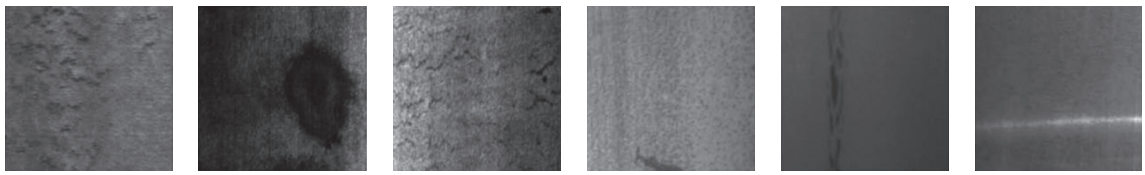

Figure 5: Examples of NEU surface defects datasets (http://faculty.neu.edu.cn/me/songkc/Vision-based_SIS_Steel.html). (a) Rolled-in scale. (b) Patches. (c) Crazing. (d) Pitted surface. (e) Inclusion. (f) Scratches.

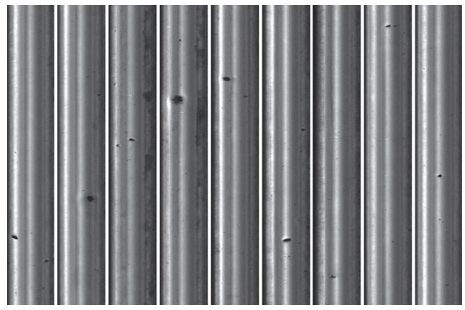

Figure 6: Examples of rail surface defect datasets (https://github.com/cuilimeng/CrackForest-dataset).

concatenate is a more advanced method than skip connection in reuse the feature layer. (6) It seems standard SDMS has better performance than SDMS-E, even though SDMS-E has better score in F measure. It indicates that residual refinement is an effective mechanism for optimizing the saliency maps.

4.5. Model Comparison. We compare the effectiveness of the DMS model with 16 saliency methods, including 12 traditional saliency algorithms (ITTI [40], LC [41], SR [42], AC [43], FT [44], MSS [45], PHOT [46], HC [47], RC [47], SF [48], BMS [49], and MBP [50]), and 8 deep learning methods (U-Net [51], FCN [19], R3Net [24], DSS [33], PiCANet [52], BASNet [29], PoolNet [53], and EGNet [26]). We implement the traditional saliency algorithm through the toolbox provided in [16]. In particularly, we test traditional saliency methods in the test dataset without free type, which causes significant interference with experimental results. For fair comparation, we realize the deep learning method by running the code directly

(a) (b) (c)

Figure 7: Examples of Road Cracks defects (https://github.com/cuilimeng/CrackForest-dataset).

Table 2: Setting of the ablation models.

Method	Standard	SDMS-A	SDMS-B	SDMS-C	SDMS-D	SDMS-E
Skip connection					✓	
Feature concatenate	✓		✓	✓		✓
Mixed supervision	✓	✓		✓	✓	✓
Residual refined	✓	✓	✓	✓	✓	
ASPP				✓		
F measure	0.9295	0.9245	0.9182	0.9254	0.9288	**0.9309**
MAE	**0.00979**	0.01294	0.01270	0.01196	0.01347	0.01063

Table 3: Results of models with different settings: the top 2 results of MAE and F_β^{Max} metrics in SDMS and top result in BDMS are marked with bold font.

Model		MAE↓	F_β^{Max}↑	Acc
Baseline	Output	0.0175	0.9160	—
SDMS-A	Output	0.0129	0.9245	90.3
SDMS-B	Output	0.0127	0.9227	—
SDMS-C	Output	0.0120	0.9254	89.8
SDMS-D	Output	0.0135	0.9288	88.3
SDMS-E	Output	**0.0106**	**0.9309**	87.6
Standard SDMS	Output	**0.0098**	**0.9295**	89.6
BDMS	Output 0	0.0134	0.9335	98.2
	Output 1	0.0103	0.9353	
	Output 2	0.0104	0.9355	
	Output 3	0.0104	**0.9369**	
	Output 4	0.0103	0.9360	
	Output 5	**0.0082**	0.9339	

provided by authors. Table 4 mainly shows the expert results on the MTDD dataset. Without any preprocessing and postprocessing, the proposed method outperforms those state-of-the-art methods. Figure 8 provides several examples of different defects, where our method is obviously better than others. Table 5 shows the test results on the other three datasets, which verified the effectiveness of the proposed method. In addition, our proposed method runs at about 7 FPS in GPU with input size $300 * 300$.

Table 4: DMS model is compared with other models in terms of performance metrics. The best three results are in bold.

Model	MAE↓	F_β^{Max}↑
ITTI	0.444	0.168
LC	0.160	0.147
SR	0.140	0.201
AC	0.119	0.161
FT	0.170	0.153
MSS	0.115	0.158
PHOT	0.108	0.161
HC	0.213	0.153
RC	0.275	0.185
SF	0.236	0.145
BMS	0.179	0.262
MBP	0.441	0.288
FCN	0.0532	0.898
PiCANet	0.0337	0.900
U-Net	0.0268	0.914
DSS	0.0177	0.919
R3Net	0.0109	0.914
BASNet	**0.0103**	0.922
PoolNet	0.0107	**0.930**
EGNet	0.0116	**0.933**
SDMS	**0.0098**	**0.930**
BDMS	**0.0082**	**0.934**

We also test the effectiveness of the DMS network on the three ASI datasets of RSDDs, road cracks, and NEU. The test result shows that the DMS network can mostly meet the

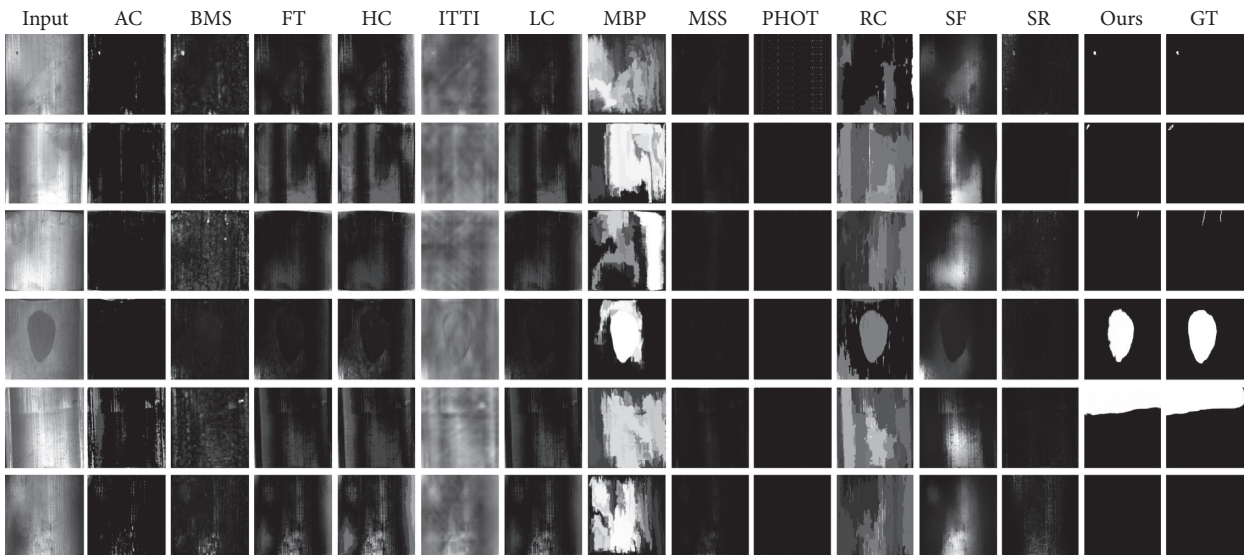

Figure 8: The saliency maps of different models were compared. The results obtained by our method are obviously different from those obtained by other saliency methods and can almost accurately segment defects of different scales in the image.

Table 5: SDMS network's performance on RSDDs, road cracks, and NEU datasets. We used the same model and hyperparameter settings in the training.

Dataset		RSDDs		Road cracks		NEU		Settings
Model		MAE↓	$F_\beta^{\text{Max}}\uparrow$	MAE↓	$F_\beta^{\text{Max}}\uparrow$	MAE↓	$F_\beta^{\text{Max}}\uparrow$	
SDMS	Output0	0.01132	0.5526	0.03044	0.6973	0.00891	0.7927	The same as the SDMS-D, which is shown in Table 3
	Output1	0.00549	0.7842	0.01955	0.8169	0.00616	0.8229	
	Output2	0.00527	0.7835	0.01934	0.8148	0.00593	0.8249	
	Output0+	0.00906	0.6884	0.02728	0.6549	0.00735	0.7913	
	Output1+	0.00602	0.7888	0.01494	0.7805	0.00611	0.8258	
	Output2+	0.00575	0.7882	0.01479	0.7786	0.00590	0.8306	

foreground division of surface defects, but there is still room for improvement in the accuracy of segmentation on small defects such as cracks.

5. Conclusion

This paper presents an optimized deep and mixed supervision network for surface defect saliency detection. The network is improved from the basic HED architecture and is equipped with layer-by-layer feature concatenate structure in the side output network. We design our loss function and add the classify module in DMS, in order to joint training classification and saliency segmentation in one stage. In the side output network, we divide the side output into high-level features, middle-level features, and low-level features and realize feature reuse on the basis of preserving feature layer information maximally. In addition, we generate saliency maps along each feature layer and apply supervisory signals, and the supervised map is passed to the next feature layer to achieve residual refinement for the saliency map.

One of our key contributions is proposal of a mechanism of classification and saliency segmentation joint training. We implement image classification plus segmentation in one model, and the classification information effectively enhances the saliency detection accuracy of the ASI dataset. In particular, it has a remarkable effect on removing normal samples (nonsalient images) with no defects in practical application. We believe that such a multitask model is a useful idea to promote saliency detection in more practical scenes. We conduct a simplified model test and tested our DMS network on 4 different test sets. Result shows the improvement mechanism we proposed increases the effectiveness of the saliency detection to different extents and can be effectively promoted to other ASI datasets.

We tried to employ the atrous spatial pyramid pooling (ASPP) from DeeplabV3 in DMS network, for which may improve the model effectiveness by expanding the convolution layer receptive field. However, the experimental result shows that this operation does not have a positive effect on the task but declined some evaluation metrics instead. After analysis, it may be that most detection objects in ASI saliency datasets are mainly small targets, so the idea of expanding the receptive field to collect global information is not very effective. That is to say, local information plays a more active role in resolving saliency surface defect detection. At present, saliency detection of small target objects is still a recognized difficulty in the field of saliency detection. Therefore, exploring how to solve the saliency segmentation of small task objectives may be one of the key points to further enhance the ASI task.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request. The saliency maps used to support the findings of this study have been deposited in the GitHub repository (https://github.com/Sssssbo/DMS). Previously reported ASI datasets were used to support this study and are available at https://github.com/abin24/Magnetic-tile-defect-datasets, http://faculty.neu.edu.cn/me/songkc/Vision-based_SIS_Steel.html, https://github.com/cuilimeng/CrackForest-dataset, and https://github.com/cuilimeng/CrackForest-dataset. These prior studies (and datasets) are cited at relevant places within the text as references [16, 37, 38, 39].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Nature Science Foundation of China (no. 51375439).

References

[1] F. Pernkopf, "Detection of surface defects on raw steel blocks using Bayesian network classifiers," *Pattern Analysis and Applications*, vol. 7, no. 3, pp. 333–342, 2004.

[2] O. Silvén, M. Niskanen, and H. Kauppinen, "Wood inspection with non- supervised clustering," *Machine Vision and Applications*, vol. 13, no. 5–6, pp. 275–285, 2003.

[3] J. J. Liu and J. F. MacGregor, "Estimation and monitoring of product aesthetics: application to manufacturing of "engineered stone" countertops," *Machine Vision and Applications*, vol. 16, no. 6, pp. 374–383, 2006.

[4] X. Xie and M. Mirmehdi, "TEXEMS: texture exemplars for defect detection on random textured surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1454–1464, 2007.

[5] V. Murino, M. Bicego, and I. Rossi, "Statistical classification of raw textile defects," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004)*, vol. 4, pp. 311–314, Cambridge, UK, August 2004.

[6] X. Xie, "A review of recent advances in surface defect detection using texture analysis techniques," *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, vol. 7, no. 3, pp. 1–25, 2008.

[7] B. Mallik-Goswami and A. K. Datta, "Detecting defects in fabric with laser-based morphological image processing," *Textile Research Journal*, vol. 70, no. 9, pp. 758–762, 2000.

[8] C.-W. Kim and A. J. Koivo, "Hierarchical classification of surface defects on dusty wood boards," *Pattern Recognition Letters*, vol. 15, no. 7, pp. 713–721, 1994.

[9] M. Niskanen, O. Silvén, and H. Kauppinen, "Color and texture based wood inspection with non-supervised clustering," in *Proceedings of the 12th Scandinavian conference on image analysis (SCIA2001)*, pp. 336–342, Bergen, Norway, June 2001.

[10] R. W. Conners, C. W. Mcmillin, K. Lin, and R. E. Vasquez-Espinosa, "Identifying and locating surface defects in wood: Part of an automated lumber processing system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 6, pp. 573–583, 1983.

[11] F. Ade, N. Lins, and M. Unser, "Comparison of various filter sets for defect detection in textiles," in *Proceedings of the International Conference on Pattern Recognition*, vol. 1, pp. 428–431, Montreal, Canada, July 1984.

[12] S. A. H. Ravandi and K. Toriumi, "Fourier transform analysis of plain weave fabric appearance," *Textile Research Journal*, vol. 65, no. 11, pp. 676–683, 1995.

[13] J.-K. Park, B.-K. Kwon, J.-H. Park, and D.-J. Kang, "Machine learning-based imaging system for surface defect inspection," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, no. 3, pp. 303–310, 2016.

[14] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Annals*, vol. 65, no. 1, pp. 417–420, 2016.

[15] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 929–940, 2018.

[16] Y. Huang, C. Qiu, Y. Guo, X. Wang, and K. Yuan, "Surface defect saliency of magnetic tile," in *Proceedings of the 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pp. 612–617, Munich, Germany, August 2018.

[17] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1915–1926, 2012.

[18] H. Tian, Y. Fang, Y. Zhao, W. Lin, R. Ni, and Z. Zhu, "Salient region detection by fusing bottom-up and top-down features extracted from a single image," *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4389–4398, 2014.

[19] L. Jonathan, S. Evan, and D. Trevor, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, June 2015.

[20] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 212–221, Venice, Italy, October 2017.

[21] L. Wang, H. Lu, Y. Wang et al., "Learning to detect salient objects with image-level supervision," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Honolulu, HI, USA, July 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[23] G. Huang, Z. Liu, V. D. M. Laurens, and K. Weinberger, "Densely connected convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, Honolulu, HI, USA, July 2017.

[24] Z. Deng, X. Hu, L. Zhu et al., "R^{3}Net: recurrent residual refinement network for saliency detection," in *Proceedings of the 2018 International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 684–690, Stockholm, Sweden, July 2018.

[25] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: aggregating multi-level convolutional features for salient object detection," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 202–211, Venice, Italy, October 2017.

[26] J. Zhao, J. Liu, D. Fan et al., "EGNet: edge guidance network for salient object detection," in *Proceedings of the 2019 IEEE International Conference on Computer Vision (ICCV)*, pp. 8779–8788, Seoul, South Korea, November 2019.

[27] J. Su, J. Li, Y. Zhang et al., "Selectivity or invariance: boundary-aware salient object detection," in *Proceedings of the 2019 IEEE International Conference on Computer Vision (ICCV)*, pp. 3799–3808, Cardiff, UK, September 2019.

[28] M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," in *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1623–1632, Long Beach, CA, USA, June 2019.

[29] X. n Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: boundary-aware salient object detection," in *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7479–7489, Long Beach, CA, USA, June 2019.

[30] R. Wu, M. Feng, W. Guan, D. Wang, H. Lu, and E. Ding, "A mutual learning method for salient object detection with intertwined multi-supervision," in *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8150–8159, Long Beach, CA, USA, June 2019.

[31] D. Fan, M. Cheng, J. Liu et al., "Salient objects in clutter: bringing salient object detection to the foreground," in *Proceedings of the 2018 European Conference on Computer Vision (ECCV)*, pp. 186–202, Munich, Germany, September 2018.

[32] S. Xie and Z. Tu, "Holistically-nested edge detection," *International Journal of Computer Vision*, vol. 125, no. 1–3, pp. 3–18, 2015.

[33] Q. Hou, M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 4, pp. 815–828, 2018.

[34] S. Jia and N. Bruce, "Richer and deeper supervision network for salient object detection," 2019, http://arXiv.org/abs/1901.02425.

[35] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, Honolulu, HI, USA, July 2017.

[36] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[37] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," *Applied Surface Science*, vol. 285, pp. 858–864, 2013.

[38] J. Gan, Q. Li, J. Wang, and H. Yu, "A hierarchical extractor-based visual rail surface inspection system," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7935–7944, 2017.

[39] Y. Shi, L. Cui, Z. Qi, F Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 1–12, 2016.

[40] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[41] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," in *Proceedings of the ACM International Conference on Multimedia*, pp. 815–824, Barbara, CA, USA, October 2006.

[42] X. Hou and L. Zhang, "Saliency detection: a spectral residual approach," in *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, Minneapolis, MN, USA, June 2007.

[43] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, "Salient region detection and segmentation," in *Proceedings of the 2008 6th international conference on Computer vision systems*, Springer, Santorini, Greece, May 2008.

[44] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1597–1604, Miami, FL, USA, June 2009.

[45] R. Achanta and S. Ssstrunk, "Saliency detection using maximum symmetric surround," in *Proceedings of the 2010 IEEE International Conference on Image Processing (ICIP)*, pp. 2653–2656, Hong Kong, China, September 2010.

[46] D. Aiger and H. Talbot, "The phase only transform for unsupervised surface defect detection," in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 295–302, San Francisco, CA, USA, June 2010.

[47] M. Cheng, G. Zhang, N. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *Proceedings of the 2011 Computer Vision and Pattern Recognition (CVPR)*, pp. 409–416, Providence, RI, USA, June 2012.

[48] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, "Saliency filters: contrast based filtering for salient region detection," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 733–740, Providence, RI, USA, June 2012.

[49] J. Zhang and S. Sclaroff, "Saliency detection: a boolean map approach," in *Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV)*, pp. 153–160, Sydney, Australia, December 2013.

[50] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, "Minimum barrier salient object detection at 80 FPS," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1404–1412, Santiago, Chile, December 2015.

[51] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional net-works for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science, vol. 9351, pp. 234–241, Springer, Berlin, Germany, 2015.

[52] N. Liu, J. Han, and M. Yang, "PiCANet: learning pixel-wise contextual attention for saliency detection," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3089–3098, Salt Lake City, UT, USA, June 2018.

[53] J. Liu, Q. Hou, M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3917–3926, Long Beach, CA, USA, June 2019.