PGA-Net: Pyramid Feature Fusion and Global Context Attention Network for Automated Surface Defect Detection

Hongwen Dong, Kechen Song, Yu He, Jing Xu, Yunhui Yan, and Qinggang Meng, *Senior Member, IEEE*

***Abstract*—Surface defect detection is a critical task in industrial production process. Nowadays, there are lots of detection methods based on computer vision and have been successfully applied in industry, they also achieved good results. However, achieving full automation of surface defect detection remains a challenge, due to the complexity of surface defect, in intraclass. While the defects between interclass contain similar parts, there are large differences in appearance of the defects. To address these issues, this article proposes a pyramid feature fusion and global context attention network for pixel-wise detection of surface defect, called PGA-Net. In the framework, the multiscale features are extracted at first from backbone network. Then the pyramid feature fusion module is used to fuse these features into five resolutions through some efficient dense skip connections. Finally, the global context attention module is applied to the fusion feature maps of adjacent resolution, which allows effective information propagate from low-resolution fusion feature maps to high-resolution fusion ones. In addition, the boundary refinement block is added to the framework to refine the boundary of defect and improve the result of the prediction. The final prediction is the fusion of the five resolutions fusion feature maps. The results of evaluation on four real-world defect datasets demonstrate that the proposed method outperforms the state-of-the-art methods on mean intersection of union and mean pixel accuracy (NEU-Seg: 82.15%, DAGM 2007: 74.78%, MT_defect: 71.31%, Road_defect: 79.54%).**

***Index Terms*—Boundary refinement, deep learning, deeply-supervised, global context attention, pyramid feature fusion, surface defect detection.**

Manuscript received September 4, 2019; revised October 29, 2019 and November 22, 2019; accepted December 2, 2019. Date of publication December 10, 2019; date of current version September 18, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 51805078 and 51374063, in part by the National Key Research and Development Program of China under Grant 2017YFB0304200, and in part by the Fundamental Research Funds for the Central Universities under Grant N170304014. Paper no. TII-19-4111. *(Corresponding authors: Kechen Song and Yunhui Yan.)*

H. Dong, K. Song, and Y. Yan are with the School of Mechanical Engineering and Automation, Northeastern University, Shenyang 110819, China, and the Key Laboratory of Vibration and Control of Aero-Propulsion Systems Ministry of Education of China, Northeastern University, Shenyang 110819, China (e-mail: donghongwenliran@163.com; songkc@me.neu.edu.cn; yanyh@mail.neu.edu.cn).

Y. He and J. Xu are with the School of Mechanical Engineering and Automation, Northeastern University, Shenyang 110819, China (e-mail: heyu142616@gmail.com; jing_xu@yeah.net).

Q. Meng is with the Department of Computer Science, Loughborough University, Loughborough LE11 3TU, U.K. (e-mail: q.meng@lboro.ac.uk).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2019.2958826

I. Introduction

THE QUALITY is an important component during the manufacturing process. To meet the growing demand, it is necessary to ensure the quality of products strictly while improving the production efficiency in the process of industrial production. Surface defect detection is a crucial step to control the quality of industrial products.

Because of the complexity of defects, there are three main challenges in the automatic defect detection task: 1) low-contrast: In the industrial production, the existence of dust and the change of light intensity result in the low contrast between defects and background in the image. Fig. 1(a) shows that the defects in red box are hardly visible; 2) intraclass difference: Unlike other applications, in industrial production, the shape of the defect is irregular. As shown in Fig. 1(b), the multiple scales of defects in the same kind are greatly different; 3) interclass similarity: Due to the uncertainty of the production process, some different kinds of defects have little difference. Fig. 1(c) presents the different types of defects (in yellow and blue boxes), which are very similar in texture and grayscale information.

Benefiting from the rapid development of computer vision, the above challenges are gradually being addressed in the industrial production. Zhang *et al.* [1] used curvature filter and Gaussian mixture model to the rail surface defect detection. Wang *et al.* [2] applied template-based methods to the strip surface defect detection. Other approaches based on hand crafted feature are used for defect detection in industrial applications (such as solar modules [3], metal [4], and steel [5]) and have achieved good result in recent years. However, these methods artificially design a set of features for a specific defect, which isn't universal.

Recently, deep learning-based [6] methods have proven to be effective in many vision tasks. Deep learning that uses nonlinear combination, and building convolutional neural network (CNN) architecture [7]–[9] the capacity of which is controlled by varying the breadth and depth to make strong and substantially correct assumptions about the nature of the image (i.e., the locality of the statistical stationarity and the pixel dependency) has been proved to be better than artificial design

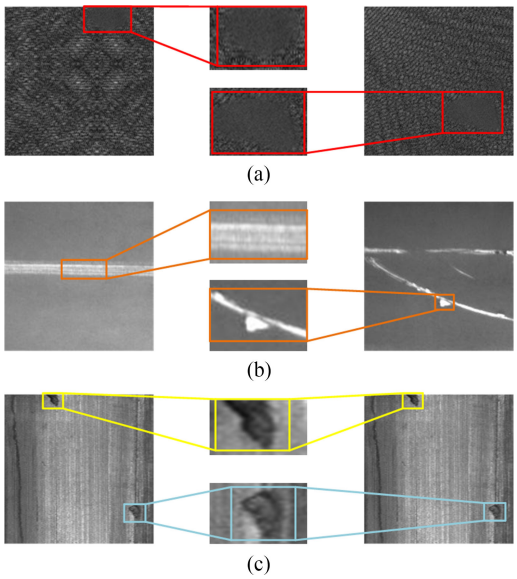


Fig. 1. Challenges of defect inspection from industry. (a) Defects with low-contrast. (b) Defects with great difference between intraclass. (c) Defects with similarity between intraclass.

features. Currently, the detection methods based on CNN have been widely used in industrial defect detection, which complete defect detection by designing different network models. For low-contrast, model needs to make effective use of features of object to distinguish the difference between object and background. As [10] indicates, features at different levels of CNN have different sensitivity to the objects. The low-level features have a higher resolution which can generate sharp and detailed boundaries but less context information, while the high-level features have more abstract semantic information which is skilled in making category classification but weak in shape and location. Most methods [11]–[13] mainly focus on the high-level features extracted from deep layers of network. Since the lack of low-level features extracted from shallow layers (e.g., boundary, texture and grayscale information) in these methods, which lead to poor prediction. Inspired by [14]–[17], this article utilizes the features extracted from last convolution layer of each stage of backbone network, which include low-level coarse features and high-level semantic features. For intraclass difference, model needs to be undeformed for various changes (such as shape, scale, and texture). Most methods based on fully convolutional network expand the receptive field to cover the whole defect to achieve the perception of the object change. In [18], dilated convolution (with different dilate rates) is used to expand the receptive field at the last convolution layer of the backbone network to enhance the cognition of the feature change. However, this will cause grid artifacts [19], [20]. Zhao *et al.* [21] fuses features extracted from backbone network under different scales by pyramid pooling module with different pooling kernels. However, pooling will cause the loss of information [22]. To solve these issues, this article proposes a pyramid feature fusion module, which uses multiscale convolution (with different size kernels) to weight the feature maps from last convolution layer of each stage of backbone network, so as to obtain the context information of different stages, and then these extracted features of the same resolution are fused at each stage. This not only avoids the gridding artifacts and lack of information but also the context information is fully extracted. Meanwhile, using the same size strip as the convolution kernel width will not bring large computation. For inter-class similarity, model also needs to realize the overall perception for different classes of objects (including the connection and difference between them) in the image, achieve each pixel needs to be classified in the correct location. In [23], the high-level features are up-sampled directly and then fused with low-level features, which is inefficient [24]. Lin *et al.* [25] obtains the multiple context information and aggregates features from high-level to low-level to refine the features detail, but it will result in a large number of parameters. To address this problem, we add global context attention module to adjacent resolution fusion maps, which extract global context information from low-resolution fusion map then weight high-resolution ones to refine the spatial location of category pixel. This not only ensures the effective dissemination of information but also does not increase the amount of computation.

Based on the improvement of the above theories, this article carried out extensive experiments on four different defect datasets to show effectiveness and generalization of our approach.

In summary, five major contributions are as follows.

1) A surface defect detection method based on deep learning is introduced, which has achieved state-of-the-art performance on four different surface defect datasets. This proves that the method has certain generality and theoretical value.
2) A method was proposed for surface defect detection at pixel-wise rather than image-level or region-level. Meanwhile, the method aims to detect and distinguish different kinds of defects, not just highline the conspicuous regions in an image.
3) A pyramid feature fusion module was provided, which fuses multilevel features from all stages of backbone CNN into multiscale resolutions, and learns these resolutions, respectively.
4) A global context attention module was designed, which embedded in these resolutions to ensure efficient information transfer from low-resolution to high-resolution.
5) The deep supervision and boundary refinement are added to the proposed method to optimize the network for multibranches, and accelerating convergence during the training process. The final framework achieves outperformance on four defect datasets.

The rest of this article is organized as follows. Related works about surface defect detection are given in Section II. Next, the proposed PGA-Net is narrated thoroughly in Section III. Afterwards, Section IV describes the evaluation on four defect datasets and corresponding discussions. Finally, Section V concludes this article.

II. Related Works

In recent years, the methods based on computer vision for surface defect detection can be categorized into traditional detection approaches and deep-learning-based detection approaches.

A. Traditional Detection Approaches

This section refers to traditional detection approaches as no-deep-learning-based approaches. In the past decade, the traditional approaches can be categorized into statistical-based approaches, filter-based approaches, and model-based approaches.

1) Statistical-Based Approaches: The statistical-based approaches were applied to measure the distribution of pixel values. Popular statistical-based methods usually adopt histogram-of-oriented-gradient, cooccurrence matrix [26], and local-binary-pattern [27] for surface defects detection.

2) Filter-Based Approaches: Filter-based approaches adopt a bank of filters to describe the texture on images in a transformed domain, which are widely used for texture analysis. The filter-based approaches can be categorized into three domains of spatial, frequency [28], and spatial-frequency.

3) Model-Based Approaches: Model-based approaches obtain certain models with special distributions or other attributes using certain models, which require a high computational complexity [29].

Despite these techniques have achieved good performance on the description of texture features and the detection of texture defects, most of them are applied for homogeneous textures and heavily dependent on expertise.

B. Deep-Learning-Based Detection Approaches

According to different surface defect detection tasks, deep-learning-based approaches can be categorized into image-level defect classification, region-level defect inspection, and pixel-level defect segmentation.

Image-level defect classification: Masci *et al*. [30] proposed a multiscale pyramidal pooling network for classification of steel defect, which didn't require the size of all images to be equal. Natarajan *et al*. [31] proposed a flexible multilayered deep feature extraction through transfer learning and support vector machine (SVM) classifiers, which overcomes the problem of over-fitting caused by small datasets. He *et al*. [32] proposed a semi-supervised model of CNN for feature extraction and fed the representation features into a classifier for classification of steel surface defect. However, these methods can't give the exact location of defects. Meanwhile, when there are many kinds of defects in the image, the accuracy of these methods will also be reduced.

Region-level defect inspection: He *et al*. [33] proposed a multilevel-feature fusion network, which combined multilevel hierarchical features extracted from a backbone CNN into one resolution for steel plate defect inspection. Chen *et al*. [34] proposed an approach based on CNN, which analyzed individual video frames for crack detection through CNN and Naïve Bayes data fusion scheme. Zhou *et al*. [35] improved a deep convolution neural network, which applied a new anchor mechanism to generate suitable candidate boxes for objects, and combines multilevel features to construct discriminative hyper features for split pins defect inspection. The shortcomings of these methods are that they can only provide a coarse region of defects through one or more tight-fitting bounding boxes, but can't describe the defect boundary precisely.

Pixel-level defect segmentation: Currently, the most effective surface defect detection methods are based on the fully convolutional network [11]. A novel CNN was proposed in [36], which integrated context information from top-to-down in a feature pyramid way for pavement crack detection. Ren *et al*. [37] proposed a deep-learning-based framework for defect classification, then obtain the pixel-wise prediction through the trained classifier convoluted with raw image. Yang *et al*. [38] proposed a multiscale feature-clustering-based fully convolutional for texture surface defect inspection. Compared with image-level- and region-level-based methods, the methods based on pixel-level can locate the defect and describe the defect boundary more accurately. However, the results of these methods also need to be improved: 1) Most of these methods focus on the high-level features, ignoring the importance of low-level features information. Meanwhile, the output is only one-side prediction, the detect results is poor. 2) Part of these methods adopt more-side prediction, and then fuse these predictions directly to output the final prediction, which lack of the intrinsic relationship of different resolutions feature maps. In contrary, we propose a pyramid feature fusion module to utilize the feature information of different layers fully. We fuse these features into different resolutions, and adopt global context attention module to fuse them step by step.

III. Methodology and Design

A. System Overview

In this article, surface defect detection is regarded as a pixel-wise task. The architecture of the proposed approach includes five major components: i) Feature extract network for multilevel features extraction; ii) pyramid feature fusion module; iii) global context attention module; iv) boundary refinement block; and v) deep supervision, as shown in Fig. 2.

1) First, input a batch size of raw images and corresponding ground truth to network, and extract the multilevel features by feature extraction network with convolution and pooling operation. The model learns the effective features in each image of the training samples through forward propagation, and these features correspond to the ground truth one by one to inform the attributes of these features. At the forward propagation, the output feature maps and ground truth are used to calculate the loss. Then back propagation algorithm minimizes the loss and achieves the goal of optimizing the network.
2) Next, feeding these features into pyramid feature fusion module. Adjusting the dimension by convolution and deconvolution (with different kernels and strides) operations to make the fused feature maps have the same dimension. Through some dense skip connects and fuse these features into five resolutions at once.

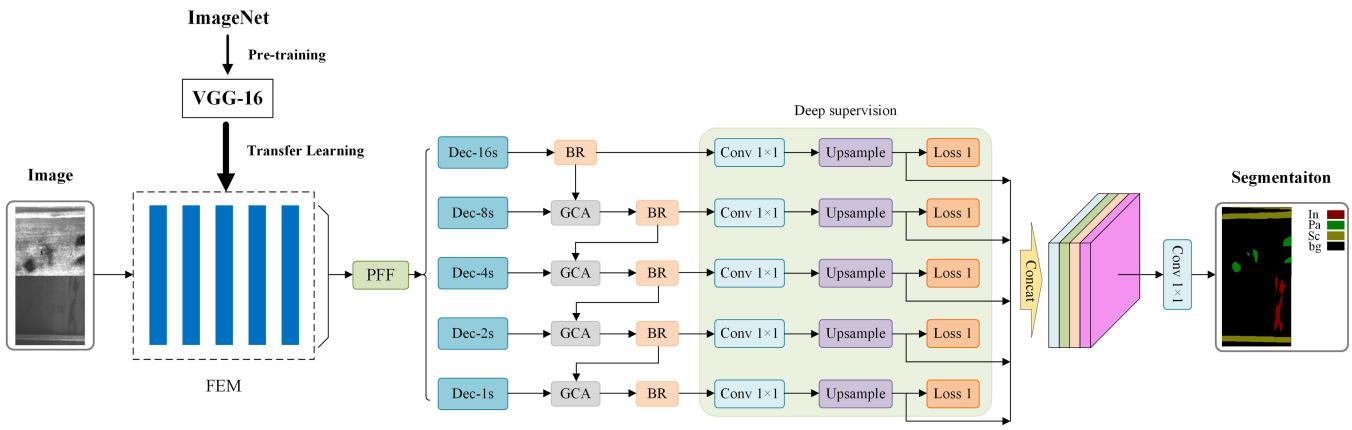


Fig. 2. Architecture of the proposed PGA-Net. Given an input image, we first use the pretrained CNN to get the feature maps from the last convolutional layer of each stage, then a pyramid feature fusion (PFF) module is applied to fuse these feature maps into five different resolutions, followed by global context attention (GCA) module and boundary refinement (BR) block to combine the adjacent resolutions and refine the predicted maps. Finally, the multiple outputs from boundary refinement are to carry out deep supervised learning. The final prediction is the fused of these multiple outputs.

3) Then, global context attention embedded in these resolutions to allow for effective information propagate from low-resolution to high-resolution. The output of each global context attention is followed by boundary refinement. Resize the dimension of each resolution to make it same as the raw image to yield prediction maps.
4) Finally, fuse these prediction maps and produces the final prediction.

B. Multilevel Features Extraction Module

CNNs are widely used to extract features from objects follow their characteristics, and these features can be learned by stacking multiple convolution and pooling layers.

In this article, the deep feature extraction module (FEM) was built on the VGG-16 [39] model pretrained with ImageNet [40] dataset to extract multilevel features for surface defect detection. The FEM includes five blocks, and these blocks extract appearance information on various, from shallow, fine layers (block_1 and block_2) to deep, coarse layers (block_4 and block_5). Each block consists of convolution layers, rectified linear unit activation function (ReLU), batch normalization, and max-pooling layer except the last block. The details of FEM can be referred to in Table I, all these layers are optimized by stochastic gradient descent in the process of back propagation to minimize the difference between prediction and ground truth.

C. Pyramid Feature Fusion Module

In deep CNN, the extent to how much context information is used roughly depends on the size of receptive field. For defect detection, some defects are intraclass difference and through the whole image [as shown in Fig. 1(b)], which need large receptive field to realize the overall perception of the defect in image. However, the size of actual receptive fields in the CNN is smaller than the theoretical ones [41]. Inspired by [42], [43], the pyramid feature fusion (PFF) module was proposed in this article as shown in Fig. 3, which can be divided into three steps. First, give an input

TABLE I
DETAILS OF FEATURES EXTRACTION MODULE

Stage	Type
	3×3 conv, stride = 1
	2×2 max pool, stride = 2
Block1	[conv 3×3 + BN + ReLU, $C = 64$] ×2 max pool 2×2
Block2	[conv 3×3 + BN + ReLU, $C = 128$] ×2 max pool 2×2
Block3	[conv 3×3 + BN + ReLU, $C = 256$] ×3 max pool 2×2
Block4	[conv 3×3 + BN + ReLU, $C = 512$] ×3 max pool 2×2
Block5	[conv 3×3 + BN + ReLU, $C = 512$] ×3

image $\mathbf{I}$ with size $W \times H$, and through FEM module generates multilevel features at different stages. The PFF module obtains last layer feature of each stage: conv1_2, conv2_2, conv3_3, conv4_3, and conv5_3. For simplicity, these five features could be denoted by a feature set $\boldsymbol{F}$:$\boldsymbol{F} = (f_1, f_2, f_3, f_4, f_5)$, where $\mathbf{f_1}$ denotes the conv1_2 features and so on. Second, multi-context information is generated by multiscale receptive fields weighted $\boldsymbol{F}$, and this information is mapped to five different resolution feature maps at the same time:$T_n = (W/2^n, H/2^n)$, where $n = (0, 1, 2, 3, 4)$, W and H represent the width and height of the input image, respectively. For $\mathbf{f}_1$ (resolution $R_1 = T_0$), the module down-scales it to five resolutions with a stack of convolution layers, and the output feature maps $\boldsymbol{Y}_1^i$ as follows:

$$\begin{aligned}\boldsymbol{Y}_1^i &= \Phi(f_1|\boldsymbol{W}, \boldsymbol{b}) \\ &= \boldsymbol{\sigma}\left(down{-}scale\left(W_{k\times k,\,s} * f_1 + \boldsymbol{b}\right)\right)(i = 1, \ldots, 5)\end{aligned} \tag{1}$$

where σ refers to the ReLU activation, *down-scale*$(\cdot)$ signifies through $\boldsymbol{W}_{k\times k}$ (kernel size is $k \times k$, stride $s = k$) to downscale the feature map $\mathbf{f}_1$, $\boldsymbol{b}$ denote bias, $*$ denotes convolution. For $\mathbf{f}_5$ (resolution $R_5 = T_4$), the module upsamples it into five

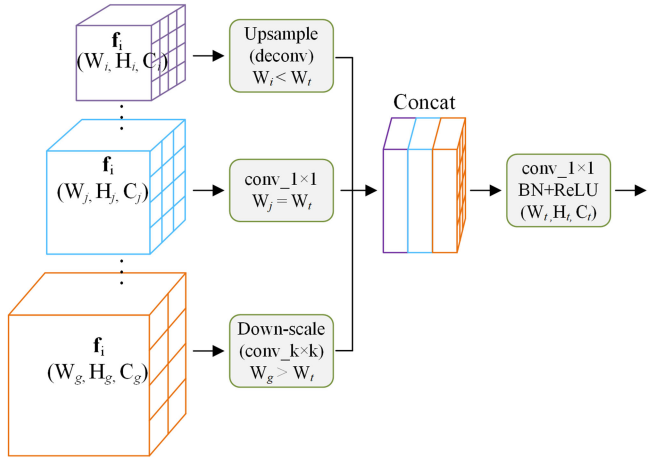


Fig. 3. Details of the PFF module. The PFF module first takes multi-scale features from FEM. Then through the up-scale, down-scale and conv operation to resize the features to the same dimension as the target fusion resolution. Finally, the concatenation and 1 × 1 convolution are used to output the final fusion feature maps.

resolutions, the output feature maps $\boldsymbol{Y}_5^i$ as follows:

$$Y_5^i = \Psi(f_5) = \boldsymbol{\sigma}\left(upsample\left(f_5;\ \psi\right)\right)(i = 1, \ldots, 5) \quad (2)$$

where σ refers to the ReLU activation, $upsample(\cdot;\psi)$ refers the deconvolution with parameters ψ which are learned during the training. For $\mathbf{f}_2$, $\mathbf{f}_3$, and $\mathbf{f}_4$, which resolution between T_0 and T_4, the model uses the combination of *down-scale* and *upsample* to resize them into five resolutions, and the output feature maps $\boldsymbol{Y}_l^i$ as follows:

$$\boldsymbol{Y}_l^i(f_l) = \boldsymbol{\sigma}\left(\Phi\left(f_l\right)\&\Psi\left(f_l\right)\right)(i = 1, \ldots, 5;\ l = 2, 3, 4) \quad (3)$$

where σ refers to the ReLU activation, $\Phi(\cdot)$ and $\Psi(\cdot)$ denote (1) and (2), respectively. The channel dimension of these resized feature maps $(\boldsymbol{Y}_1^i, \ldots, \boldsymbol{Y}_5^i)$ is 128. Finally, the features with same dimension in these output ones are fused to generate the final five fused feature maps. To be convenient, the five fused feature maps are named Dec-1s (n = 0), Dec-2s (n = 1), Dec-4s (n = 2), Dec-8s (n = 3), Dec-16s (n = 4), respectively. The five fused features could be defined as

$$\text{Dec}-i_s = \boldsymbol{\sigma}\left(\boldsymbol{W}_{1\times1} * \boldsymbol{CAT}\left(\boldsymbol{Y}_1^i, \ldots, \boldsymbol{Y}_3^i, \ldots \boldsymbol{Y}_5^i\right) + \boldsymbol{b}\right) \quad (4)$$

where σ refers to the ReLU activation, ***CAT*** denotes the element-wise concatenated. The channel of each fused features map Dec is 640. $\boldsymbol{W}_{1\times1}$ denotes a convolution with 1 × 1 kernel size to change the channel dimension of concatenated features (640 to 128), b refers bias. All convolution layers defined in PFF are followed by ReLU activation and batch normalization and these parameters are trainable, as shown in Table II. Through this way, the model effectively obtains the multiscale context information from different stages of CNN, and realizes the overall perception of the object.

D. Global Context Attention Module

The final fusion feature maps with different resolution generated from PFF contains various visual context information,

TABLE II
DETAILS OF PYRAMID FEATURE FUSION MODULE

Stage	Dec-1s	Dec-2s	Dec-4s	Dec-8s	Dec-16s
Conv1_2	1×1, s=1	2×2, s=2	4×4, s=4	8×8, s=8	16×16, s=16
Conv2_2	deconv	1×1, s=1	2×2, s=2	4×4, s=4	8×8, s=8
Conv3_3	deconv	deconv	1×1, s=1	2×2, s=2	4×4, s=4
Conv4_3	deconv	deconv	deconv	1×1, s=1	2×2, s=2
Conv5_3	deconv	deconv	deconv	deconv	1×1, s=1

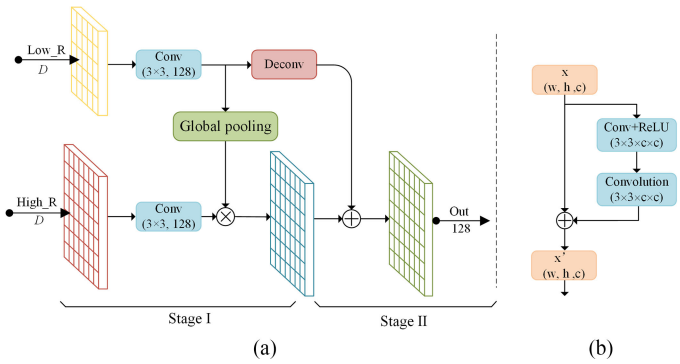


Fig. 4. The details of global context attention module and boundary refinement block are illustrated in (a) and (b), respectively.

and each of them can be used to yield the result prediction. One method using bilinear upsample to up-scale these fused features into the same dimension with the raw image, then change their channel through a convolutional layer to the number of classes to predict the segmentation result. However, the shortcomings of these approaches are: 1) They lack the inner relation information between different resolution predictions, ii) use bilinear upsample with a big kernel directly may lead to the missing of some detailed information and the parameters are not trainable. Other U-shape models [23], [44] combine the adjacent feature maps from low-resolution to high-resolution step-by-step in the decoding process. However, there are also two shortcomings in these methods: 1) The type of this combination between adjacent features maps in the decoding process is too single and lack diverse representation, 2) lack the global context information from low-resolution (high-level), which can enhance high-resolution (low-level) feature map in decoding process.

To address above issues, a global context attention module [shown in Fig. 4(a)] was proposed, which consists of two stages:

Stage one: A 3 × 3 convolution was applied to adjust the channels dimension of high-resolution and low-resolution fusion feature maps, then through global pooling to the low-resolution to obtain global context, following multiplied with the high-resolution feature map. The output f_{s1} as follows:

$$f_{s1} = \sigma\left(\boldsymbol{W}_{3\times3} * f^h + b\right) \otimes \sigma\left(\mathcal{G}(\sigma(\boldsymbol{W}_{3\times3} * f^l + b))\right) \quad (5)$$

where $\otimes$ and $*$ denote element-wise multiplication and convolution, respectively, $\mathcal{G}(\cdot)$denotes global pooling operation, σ refers to ReLU activation, f^h and f^l represent high-resolution and low-resolution fusion feature maps, $\boldsymbol{W}_{3\times3}$ indicates trainable parameters, b refers bias.

Stage two: The low-resolution fusion feature map is upsampled to the same dimension with the high-resolution, and then added with the f_{s1}. The output of stage two f_{s2} as follows:

$$f_{s2} = \sigma\left(upsample\left(f^{l};\psi\right) \oplus f_{s1}\right) \tag{6}$$

where $upsample(\cdot;\psi)$ refers to the deconvolution with parameters ψ which are learned during the training, $\oplus$ refers to element-wise addition.

In short, compared with simply adding the upsampled coarser-resolution feature maps to the finer-resolution ones, the proposed GCA module can utilize different resolution fusion feature maps to improve the efficiency of context obtain and corresponding pixel-wise localization.

E. Boundary Refinement Block

In this article, we add boundary refinement [45] block to further improve the detection accuracy, shown in Fig. 4(b). The boundary refinement was seen as a residual structure, the output refined score map $\tilde{S}$ as follows:

$$\tilde{S} = \sigma\left(\boldsymbol{W}_{1\times1} * (S \oplus \mathbb{R}(S)) + b\right) \tag{7}$$

where S and $\mathbb{R}(\cdot)$ signify the coarse score map and residual branch, respectively, $*$ represents convolution, σ refers to the ReLU activation, $\boldsymbol{W}_{1\times1}$ indicates trainable parameters, $\oplus$ is the cross-channel concatenation, b refers bias. The details were shown in Fig. 4(b).

F. Deep Supervision

Although multilevel features are fully utilized, the mount of parameters is also increased obviously, which may introduce additional optimization difficulty. To address the issue, we add deep supervision into our model, which aims to ease the process of training and accelerate the optimization of network model.

The fused feature maps generated from PFF module at each resolution can performs crack prediction individually. We add a per-pixel loss (cross-entropy) to each of the above five resolution fused maps. The loss function is described as

$$\boldsymbol{L}(T,P) = -\frac{1}{N}\sum_{i=1}^{N}\left[T_i \cdot \log P_i + (1-T_i)\log(1-P_i)\right] \tag{8}$$

where T_i and P_i represent ground truth and predicted probabilities of i^{th} image, respectively, N refers batch size. In the test phase, the predictions generated from the five branches are fused to output the result of detection, as shown in Fig. 2.

IV. Experiments and Results

A. Implementation Details

1) Parameters Setting: The initialization parameters of FEM are transformed from the pretrained VGG-16 network which for image classification task on the ImageNet dataset. Furthermore, the weights realize initialization of other convolutional layers through "Xavier" scheme. As for up-scale features, we use transposed convolution with learnable weights. For fine-tuning, we set the base learning rate is 10^{-5} with a decay of 0.005, the max-inter is 100k with mini-batch size 5. The model is saved every 5000 iterations.

2) Computation Platform: We implement our method on the PyCharm with the open source toolbox TensorFlow [46]. We run our method in a NVIDIA GTX TITAN GPU (with 12G memory) on Ubuntu 16.04 Linux.

B. Datasets

1) Datasets Description: In this article, four surface defect datasets are selected to prove and evaluate the applicability and generality of the proposed method, including NEU-DET defect dataset, DAGM 2007 defect dataset, MT defect dataset, and Road defect dataset.

NEU-Seg Dataset: NEU-Seg defect dataset is a standardized high-quality database, which was collected by [51] to solve the problem of automatic recognition for hot-rolled steel strip. This dataset includes six categories of surface defects from strip steel plates, including patch, crazing, pitted-surface, inclusion, scratches, and rolled-in scale. The resolution of each raw image is 200×200 and each class include 300 images with tightfitting bounding box annotations. However, in order to achieve the pixel-wise surface defect detection task, this form of annotation does not satisfy the training of CNN model. In this work, three typical defects (inclusion, patch, and scratches) are selected, and pixel-wise annotation is conducted by the open annotation tool: LabelMe. This dataset is named as NEU-Seg datasets. Due to the complexity of the situation of hot-rolled plates, there are large differences in the appearance of the defects between intraclass, while the defects between interclass contain similar parts, as well as the low contrast with background. All these factors bring great challenges to the surface defect detection of hot-rolled strip steel. Fig. 5 shows the visualization of partial NEU-Seg raw images and corresponding ground truth.

DAGM 2007 Dataset: This dataset [47] which produced by artificial represents defects under a textured background is very close to real-world. This dataset includes many categories defects and the resolution of each raw image is 512×512. In the label images of DAGM 2007, the defect regions are blanketed roughly by ellipses. In this experiment, six types of defects are selected and redefine the raw label (we didn't change the size of the raw defect area, just changed the index in the label image), the different indexes in the new label image represent different categories. Fig. 6 shows partial defect images and corresponding ground truth of DAGM 2007 datasets.

MT Defect Dataset: The magnetic-tile defect dataset is presented in [48] which contains 1344 defect images, and each raw defect image corresponds to a pixel-level label. MTdefect dataset includes five types of defects: uneven, fray, crack, blowhole, and break, all these defect images with different resolution. Most of these defect images contain a series of noise, e.g., the diversity of defect shape, complexity of texture, and the change of illumination intensity, all these factors bring a big challenge of detection. In this experiment, we detect five types defects (blowhole, crack, fray, break, and uneven) of magnetic-tile defect dataset. Fig. 7 shows the partial raw defect images and corresponding ground truth.

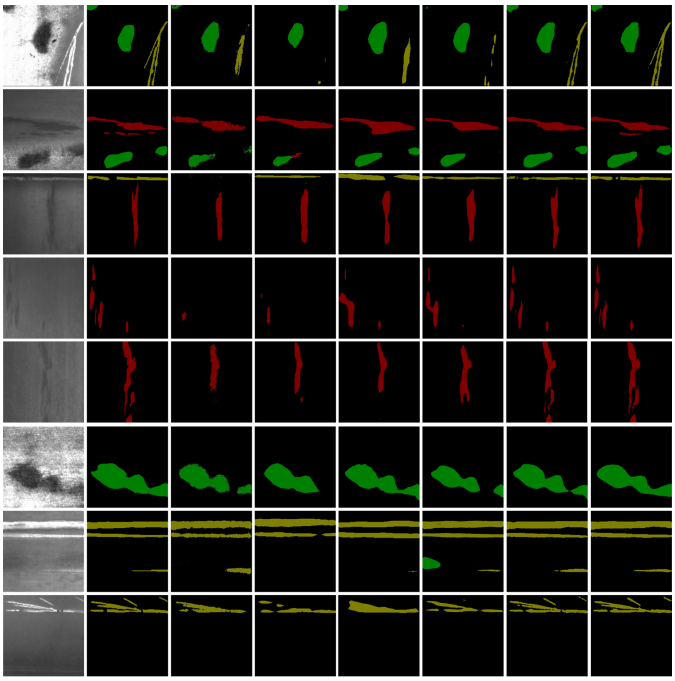


Fig. 5. Comparison of detection results on NEU-Seg dataset. Red, green and yellow represent inclusion (In), patches (Pa), and scratches (Sc) defect, respectively. (a) Original image. (b) Ground truth. (c) SegNet. (d) PSPNet. (e) DeepLab. (f) RefineNet. (g) FCN. (h) PGANet.

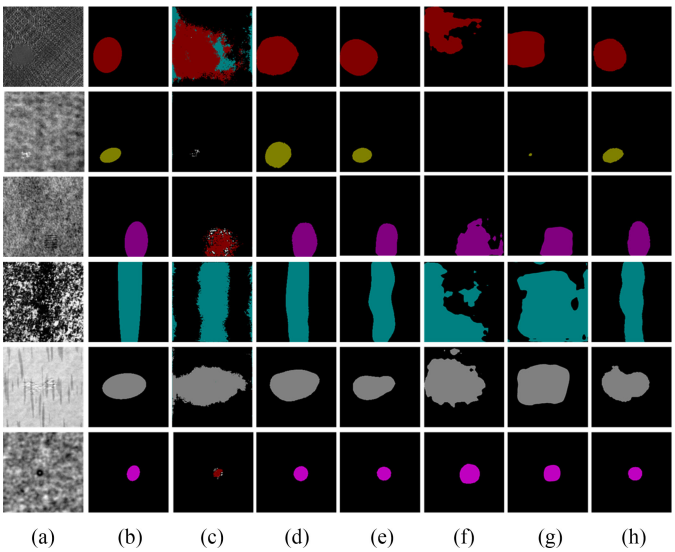


Fig. 6. Comparison of detection results on DAGM 2007 dataset. (a) Original image. (b) Ground truth. (c) SegNet. (d) FCN. (e) DeepLab. (f) PSPNet. (g) RefineNet. (h) PGA-Net.

Road Defect Dataset: This dataset contains two classes (crack, inlaid patch). The number of crack images are 500 with size around 2000 × 1500 pixels, which collected by [36]. The inlaid patch images that we collect by CCD contains 800 images of size around 3000 × 2000. Each defect image corresponds to a pixel-level label with different indexes. In this experiment, these raw images are randomly cut to 256 × 256 to improve diversity of the dataset, the effective area of the defect in each crop-image

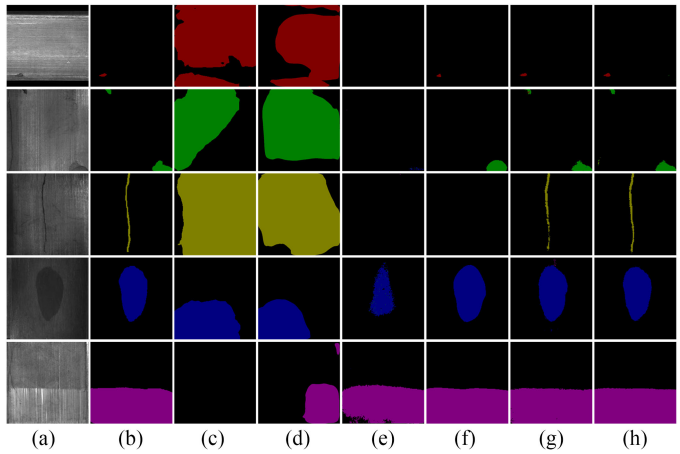


Fig. 7. Comparison of detection results on MT defect dataset. Different colors represent different kinds of defects, respectively. (a) Original image. (b) Ground truth. (c) PSPNet. (d) RefineNet. (e) SegNet. (f) DeepLab. (g) FCN. (h) PGA-Net.

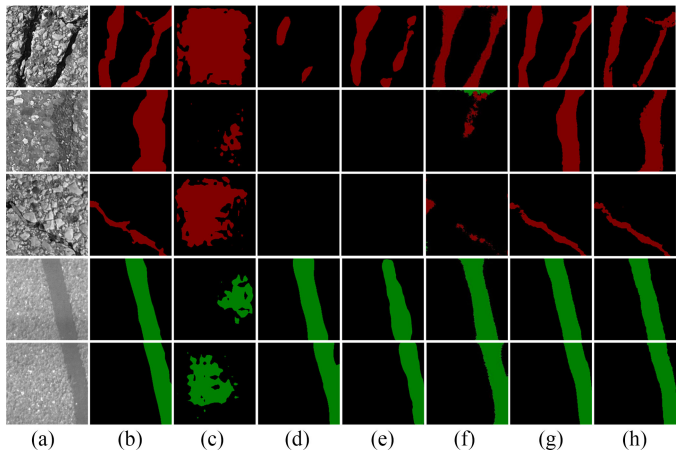


Fig. 8. Comparison of detection results on Road defect dataset. Red and green represents crack and inlaid patch, respectively. (a) Original image. (b) Ground truth. (c) PSPNet. (d) DeepLab. (e) RefineNet. (f) SegNet. (g) FCN. (h) PGA-Net.

is kept above twenty percent. Fig. 8 shows example raw defect images and corresponding ground truth.

2) Data Augmentation: Deep learning-based detection methods usually require a great many datasets to overcome over-fitting during training. However, in the actual production process, it is difficult to collect a large number of high-quality defect images due to factors such as environment and equipment, etc. The basic method to solve the over-fitting issue caused by the lack of dataset during training the model is data augmentation. In the early stage of the experiment, we cut all the images of four datasets with 200 × 200 resolution. Then we rotate the cropped images (90°, 180°, 270°) to increase the training samples, and the corresponding ground truth are also processed in the same way. To ensure the validity of the samples, samples with defect area less than 10% of the whole image are deleted. The details are shown in Table III. Inspired by [40], [44], in the process of training, the network randomly extracts 192 × 192

TABLE III
NUMBER OF FOUR DATASETS

Dataset	Train	Test
NEU-Seg	3630	840
DAGM 2007	3550	400
MT Defect	2840	300
Road Defect	6000	400

TABLE IV
QUANTITATIVE COMPARISONS OF DIFFERENT DETECTION METHODS

Dataset	Method	mIoU (%)	Time (s)
NEU-Seg Dataset	SegNet [23]	56.57	0.0528
	PSPNet [21]	72.25	0.0375
	DeepLab [18]	74.01	0.0104
	RefineNet [25]	75.37	0.0315
	FCN [11]	81.79	0.0665
	PGA-Net	82.15	0.0206

size areas (and their horizontal reflections) from input images, then train the framework on these extracted areas, which the training samples are increased 128 times. Although the random extract operation will not change the structure of the object in the image too much, so that each extracted region has high similarity in the object itself, but the operation can change the spatial position information and the semantic information of the target, so as to improve the number and diversity of dataset and avoid over-fitting.

C. Evaluation Metrics

Compared with other segmentation methods, mean intersection-over-union (mIoU) is used to performance the evaluation of the prediction result. The mathematical definitions are shown in [11]. We also use the average running time of process each image in this experiment to show the real-time performance of the proposed approach.

D. Experiment Results and Analysis

1) Detection Results on NEU-Seg Defect: The visual comparison of our approach and other methods for strip steel surface defect images are shown in Fig. 5. It can be observed that the proposed PGA-Net outstanding the performance than other methods in the challenging cases of defect detection, e.g., low-contrast (the 4–5 rows) and intraclass difference (the 7–8 rows), and the results of prediction are very similar to the ground truth. As the quantitative comparisons shown in Table IV, the proposed approach is superior to other compared counterparts in term of the evaluation metrics: the value of mIoU is improved to 82.15%.

2) Results of DAGM 2007 Dataset: The comparison of visual results of partial DAGM 2007 defect images detection shown in Fig. 6. The main detection challenging for this dataset is low-contrast between the backgrounds and defects (the 1–3 rows). It can be found that [23] miss or erroneously detects some defects. For some large area defects, [21] and [25] can't accurately locate defects. [11] and [18] magnify some low-contrast defect areas. In Contrary, the performance of proposed PGA-Net is closer to the real situation. As listed in Table V, PGA-Net improves performance to 74.78%.

TABLE V
QUANTITATIVE COMPARISONS OF DIFFERENT DETECTION METHODS

Dataset	Method	mIoU (%)	Time (s)
DAGM 2007 Dataset	SegNet [23]	21.95	0.0558
	RefineNet [25]	32.90	0.0322
	PSPNet [21]	41.21	0.0369
	FCN [11]	73.86	0.1041
	DeepLab [18]	74.61	0.0108
	PGA- Net	74.78	0.0229

TABLE VI
QUANTITATIVE COMPARISONS OF DIFFERENT DETECTION METHODS

Dataset	Method	mIoU (%)	Time (s)
MT Defect Dataset	PSPNet [21]	12.84	0.0401
	RefineNet [25]	13.52	0.0335
	SegNet [23]	33.32	0.0550
	DeepLab [18]	49.21	0.0124
	FCN [11]	67.83	0.2735
	PGA- Net	71.31	0.0246

TABLE VII
QUANTITATIVE COMPARISONS OF DIFFERENT DETECTION METHODS

Dataset	Method	mIoU (%)	Time (s)
Road Defect Dataset	PSPNet [21]	27.65	0.0374
	RefineNet [25]	46.21	0.0322
	DeepLab [18]	47.63	0.0109
	SegNet [23]	65.04	0.0568
	FCN [11]	78.74	0.1018
	PGA- Net	79.54	0.0218

3) Detection Results on MT Defect: Fig. 7 shows some samples of magnetic tile defects and the corresponding visual prediction. The main challenging of this dataset is inter-class similarity (the 1–3 rows). It can be observed from Fig. 7 that [25] and [21] are failure of predict result. [23] and [18] can locate and detect the defects (the region of defects is large and obvious), but they are easy to miss the defect detection in small area. [11] can effectively defects small defects, but the detected defect area is incomplete. However, the proposed PGA-Net achieves best performance in above aspects. In Table VI, PGA-Net improves the mIoU to 71.31%.

4) Detection Results on Road Defect Dataset: Fig. 8 demonstrates part of the road defect images and the corresponding prediction results. The main challenging cases of this dataset are low-contrast (the 2 row) and intraclass difference (the 1–3 rows). It can be observed from Fig. 8 that [21] detected defects roughly, but failed to locate the region of defect. For crack defects which are low contrast to the background, [25] and [18] failed to detect these defects, [23] and [11] lack of integrity in these defects. As listed in Table VII, the proposed PGA-Net increases the mIoU by 0.8% compared with the second best.

5) Analysis of Time to Test Each Image: Tables IV, V, VI, and VII list the average running time to process each image of different methods, and performed on four datasets using a computer introduced in Section IV-A. Compared with other state-of-the-art methods, the time of test each image of our method is not the shortest, but the speed can reach 41–49 fps/s, which is acceptable in the real detection process and does not harm the user experiences. In future research, we will further

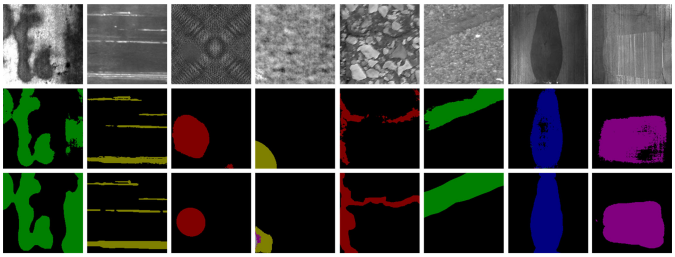


Fig. 9. Failure of proposed method PGA-Net on four datasets. (a) and (b) are failure detect of NEU-Seg dataset, (c) and (d) are failure detect of DAGM 2007, (e) and (f) are failure detect of Road Defect dataset, (g) and (h) are failure detect of MT Defect dataset.

optimize the code to accelerate the proposed method to meet the needs of real-time and high accuracy.

6) Analysis of Failure cases: The experiment results show that the proposed approach outperforms state-of-the-art detection methods on the four datasets. However, some difficult images still posed challenged to our method as well as those comparative methods. As shown in second line images (a), (b), (e), (g), (h) of Fig. 9, we can see that our method is lack of integrity in detection the defect area of partial defect images. Because of the network over fits the benchmark data, when the difference between the test sample and the training sample is large, it will lead to missed detection. The main reason is the lack of dataset. Meanwhile, the generalization ability of network model needs to be improved. As shown in (c), (d) of the second line of Fig. 9, some defects are detected by mistake. The network model is over sensitive to image changes. When the defect area changes obviously, the network does not regard it as a whole. The main reason for this case is the lack of diversity and number of datasets. We will work on these problems in the future.

E. Ablative Study

To evaluate the proposed method, this article conducts a rank of ablation experiments, including down-scale type, the effects of fusion feature resolution, and the boundary refinement for detection result. All the evaluate of these ablative experiments based on NEU-Seg dataset.

1) The Ablation Studies for the Down-Sacle Type: For the down-scale structure in PFF module, this article uses large kernel convolution ($\mathrm{conv_{k\times k}}$) to replace max pooling. On the one hand, convolution reduces the dimension of feature and retains feature information, while the max pooling may lead to a large number of feature information loss. Meanwhile, using large kernel size doesn't bring too much computation burden. As shown in Table VIII, the performance is improved from 79.89% to 80.46%.

2) The Effects of Fusion Feature Resolution: Some fused feature structures (spatial and semantics) from FEM are easily destroyed when these features resolution are adjusted by convolution and deconvolution (with big kernel size and stride) which are very different from the resolution of target fusion feature map (Block5_3→Dec-1s / Block1_2→Dec-16s). To verify the effects of fusion feature resolution, this work additionally evaluates the five fusion feature maps (Dec-1s, Dec-2s, Dec-4s, Dec-8s, Dec-16s) come from PFF, and the corresponding performance are shown in Table VIII. As can be seen from table that with the increment of resolutions, the model performance better gradually, which demonstrate the necessary of selection of resolution for feature maps fusion for our method.

TABLE VIII
DETAILED PERFORMANCE OF OUR METHOD WITH DIFFERENT SETTINGS

Method	mIoU (%)
Dec-16s	76.84
Dec-8s	80.36
Dec-4s	81.68
Dec-2s	82.00
Dec-1s	82.07
PFF($\mathrm{Conv}_{k\times k}$) + GCA	80.46
PFF(Max pooling) + GCA	79.89
PFF($\mathrm{Conv}_{k\times k}$) + GCA + BR	82.15

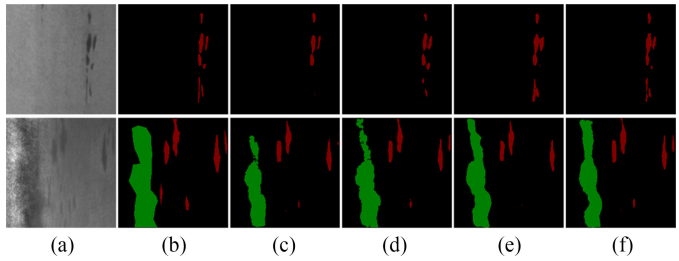


Fig. 10. Comparison of detection results on NEU-Seg dataset. (a) Original image. (b) Ground truth. (c) UN. (d) DLA. (e) DSS. (f) Ours.

TABLE IX
DETAILED PERFORMANCE OF DIFFERENT MULTISCALE FEATURES FUSION MODULES

Method	UN	DLA	DSS	Ours
mIoU (%)	81.43	81.58	81.66	82.15

3) The Ablation of Boundary Refinements for Inspection Result: Boundary refinement block is added into the proposed method to improve the performance of result. As shown in Table VIII, with the boundary refinement block yields results 82.15% in terms of mIoU, which proves the validity of BR for our method.

4) Compared With Other Multilevel Features Fusion Methods: To verify the advantage of the proposed PFF module, we compared other multilevel features fusion methods, including UN [44], DLA [49], and DSS [50]. We replace the PFF module of our proposed method with the multilevel fusion module of these methods. All the modules are based on the same backbone network VGG-16 network. In the experiment, we optimize the parameters of each multilevel features fusion module to achieve the best results. We evaluate the prediction results of these modules on the NEU-Seg dataset. The visual comparison of our approach and other methods is shown in Fig. 10. The quantitative evaluation is listed in Table IX. From Fig. 10 and Table IX we can see that compared with multilevel features fusion module, our PFF achieves better performance.

V. Conclusion

In this article, an automatic defect detection network for surface defect detection was proposed. In the framework, multilevel features from defect images were extracted by feature extraction module. Pyramid feature fusion module was introduced to fuse these multilevel features into different resolutions. Global context attention module makes the effective information propagate from low-resolution fusion feature maps to high-resolution fusion ones. The boundary refinement block was added in the framework to refine the object boundary prediction. Deep supervision was applied in the framework to speed up the process of network optimization. Experiments demonstrated that the proposed approach significantly advanced the state-of-the-art approaches on four surface defect datasets detection. However, due to the limitation of the number and diversity of datasets, some defects were missing and wrongly detected as shown in Fig. 9. Although the speed of detection can reach 41–49 fps/s, which is acceptable in the real detection process and does not harm the user experiences, the detection speed needed to be further improved to meet the needs of real-time and high accuracy. In addition, the training and test samples need to be labeled in the experiment, which consuming time.

In future research, we plan to seek an efficient data augmentation strategy combined with our approach to improve the detection performance, and optimize the framework to accelerate the proposed approach to meet real-time and high accuracy requirement. In addition, semi-supervised mechanism will be adopted in our future work.

References

[1] H. Zhang, X. Jin, Q. M. J. Wu, Y. Wang, Z. He, and Y. Yang, "Automatic visual fetection system of railway surface defects with curvature filter and improved gaussian mixture model," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 7, pp. 1593–1608, Jul. 2018.

[2] H. Wang, J. Zhang, Y. Tian, H. Chen, H. Sun, and K. Liu, "A simple guidance template-based defect detection method for strip steel surfaces," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2798–2809, May 2019.

[3] D. Tsai, S. Wu, and W. Chiu, "Defect detection in solar modules using ICA basis images," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 122–131, Feb. 2013.

[4] J. Wang, Q. Li, J. Gan, H. Yu, and X. Yang, "Surface defect detection via entity sparsity pursuit with intrinsic priors," *IEEE Trans. Ind. Informat.*, to be published. doi: 10.1109/TII.2019.2917522.

[5] Q. Luo, Y. Sun, P. Li, O. Simpson, L. Tian, and Y. He, "Generalized completed local binary patterns for time-efficient steel surface defect classification," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 3, pp. 667–679, Mar. 2019.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[7] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *Proc. IEEE 12th Int. Conf. Comput. Vision*, 2009, pp. 2146–2153.

[8] S. C. Turaga *et al.*, "Convolutional networks can learn to generate affinity graphs for image segmentation," *Neural Computation*, vol. 22, pp. 511–538, 2010.

[9] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[10] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 5188–5196.

[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3431–3440.

[12] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 5455–5463.

[13] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Boston, MA, 2015, pp. 1265–1274.

[14] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *Proc. Int. Conf. Comput. Vision*, Oct. 2017, pp. 212–221.

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

[16] 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 *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 8150–8159.

[17] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Salient object detection with recurrent fully convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1734–1746, Jul. 2019.

[18] L. 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. 2018.

[19] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 636–644.

[20] P. Wang *et al.*, "Understanding convolution for semantic segmentation," in *Proc. IEEE Winter Conf. Applicat. Comput. Vision*, 2018, pp. 1451–1460.

[21] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 6230–6239.

[22] Z. Gao, L. Wang, and G. Wu, "LIP: Local importance-based pooling," in *Proc. Int. Conf. Comput. Vision*, Oct. 2019, pp. 3355–3364.

[23] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[24] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2016. [Online]. Available: https://arxiv.org/abs/1606.02147

[25] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 5168–5177.

[26] M. Win, A. R. Bushroa, M. A. Hassan, N. M. Hilman, and A. Ide-Ektessabi, "A contrast adjustment thresholding method for surface defect detection based on mesoscopy," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 642–649, Jun. 2015.

[27] M. Quintana, J. Torres, and J. M. Menéndez, "A simplified computer vision system for road surface inspection and maintenance," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 608–619, Mar. 2016.

[28] X. Bai, Y. Fang, W. Lin, L. Wang, and B.-F. Ju, "Saliency-based defect detection in industrial images by using phase spectrum," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2135–2145, Nov. 2014.

[29] X. Xie and M. Mirmehdi, "TEXEMS: Texture exemplars for defect detection on random textured surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1454–1464, Aug. 2007.

[30] J. Masci, U. Meier, G. Fricout, and J. Schmidhuber, "Multi-scale pyramidal pooling network for generic steel defect classification," in *Proc. 2013 Int. Joint Conf. Neural Netw.*, 2013, pp. 1–8.

[31] V. Natarajan, T. Hung, S. Vaikundam, and L. Chia, "Convolutional networks for voting-based anomaly classification in metal surface inspection," in *Proc. IEEE Int. Conf. Ind. Technol.*, 2017, pp. 986–991.

[32] Y. He, K. Song, H. Dong, and Y. Yan, "Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network," *Opt. Lasers Eng.*, vol. 122, pp. 294–302, 2019.

[33] 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.*, to be published. doi: 10.1109/TIM.2019.2915404.

[34] F. Chen and M. R. Jahanshahi, "NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4392–4400, May 2018.

[35] J. Zhong, Z. Liu, Z. Han, Y. Han, and W. Zhang, "A CNN-based defect inspection method for catenary split pins in high-speed railway," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2849–2860, Aug. 2019.

[36] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE Trans. on Intell. Transp. Syst.*, doi: 10.1109/TITS.2019.2910595.

[37] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 929–940, Mar. 2018.
[38] H. Yang, Y. Chen, K. Song, and Z. Yin, "Multiscale feature-clustering-based fully convolutional autoencoder for fast accurate visual inspection of texture surface defects," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1450–1467, Jul. 2019.
[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, *arXiv:1409.1556*.
[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems 25.*, Sydney, NSW, Australia: Curran Assoc. 2012, pp. 1097–1105.
[41] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs," 2014, *arXiv:1412.6856*.
[42] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.
[43] W. Wang, S. Zhao, J. Shen, Steven C, H. Hoi and A. Borji, "Salient object detection with pyramid attention and salient edges," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 1448–1457.
[44] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assisted Intervention – MICCAI 2015*, Cham, Switzerland: Springer, 2015, pp. 234–241.
[45] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–Improve semantic segmentation by global convolutional network," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1743–1751.
[46] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," 2016, *arXiv: 1605.08695*.
[47] M. Wieler and T. Hahn, *Weakly Supervised Learning for Industrial Optical Inspection*. Accessed: Jun. 25, 2017, [Online]. Available: https://hci.iwr.uni-heidelberg.de/node/3616.
[48] Y. Huang, C. Qiu, and K. Yuan, "Surface defect saliency of magnetic tile," *Vis. Comput.*, pp. 1–12, 2018.
[49] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 2403–2412
[50] Q. Hou, M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, Apr. 2019.
[51] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," *Appl. Surface Sci.*, vol. 285, pp. 858–864, Nov. 2013.

Hongwen Dong received the B.S. degree from the School of Mechanical Engineering and Automation, Liaoning University of Technology, Jinzhou, China, in 2016, and the M.S. degree form the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 2018, both in mechanical engineering. He is currently working toward the Ph.D. degree in mechanical design and theory with the School of Mechanical Engineering and Automation, Northeastern University.

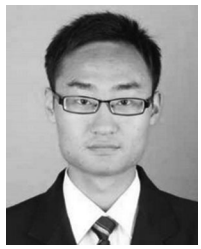
His research interests include deep learning, pattern recognition, and semantic segmentation.

Kechen Song received the B.S., M.S., and Ph.D. degrees in mechanical design and theory from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 2009, 2011, and 2014, respectively.

Between 2018 and 2019, he was an Academic Visitor with the Department of Computer Science, Loughborough University, U.K. He is currently an Associate Professor with the School of Mechanical Engineering and Automation, Northeastern University. His research interests cover vision-based inspection system for steel surface defects, surface topography, image processing, and pattern recognition.

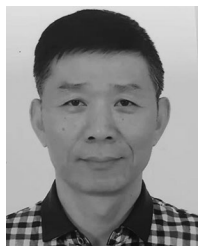
Yu He received the B.S. degree from the School of Mechanical Engineering and Automation, Liaoning Technical University, Fuxin, China, in 2014, and the M.S. degree from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 2016, both in mechanical engineering. He is currently working toward the Ph.D. degree in mechanical design and theory with the School of Mechanical Engineering and Automation, Northeastern University.

His research interests include deep learning, pattern recognition, and intelligent inspection.



Jing Xu received the B.S. and M.S. degrees in mechanical engineering from the School of Mechanical Engineering, Liaoning Shihua University, Fushun, China, in 2013 and 2016, respectively. He is currently working toward the Ph.D. degree in mechanical design and theory with the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China.

His research interests include robot motion planning and robot control.



Yunhui Yan received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 1981, 1985, and 1997, respectively.

He has been a Teacher with Northeastern University, since 1982, and became a Professor with the Department of Mechanical Engineering, Northeastern University, Shenyang, China, in 1997. During 1993–1994, he was a Visiting Scholar with the Tohoku National Industrial Research Institute. His research interests include intelligent inspection, image processing, and pattern recognition.

Qinggang Meng (M'06–SM'18) received the B.S. degree in electronic engineering and M.S. degree in signal and image processing from the School of Electronic Information Engineering, Tianjin University, Tianjin, China, and the Ph.D. degree in computer science from Aberystwyth University, Aberystwyth, U.K., in 1987, 1990, and 2002, respectively.

He is currently a Professor with the Department of Computer Science, Loughborough University, U.K. His research interests include biologically and psychologically inspired learning algorithms and developmental robotics, service robotics, robot learning and adaptation, multi-UAV cooperation, drivers distraction detection, human motion analysis and activity recognition, activity pattern detection, pattern recognition, artificial intelligence, and computer vision.

Dr. Meng is a fellow of the Higher Education Academy, U.K.