Triplet-Graph Reasoning Network for Few-Shot Metal Generic Surface Defect Segmentation

Yanqi Bao, Kechen Song, *Member, IEEE*, Jie Liu, Yanyan Wang, Yunhui Yan, Han Yu, and Xingjie Li

***Abstract*— Metal surface defect segmentation can play an important role in dealing with the issue of quality control during the production and manufacturing stages. There are still two major challenges in industrial applications. One is the case that the number of metal surface defect samples is severely insufficient, and the other is that the most existing algorithms can only be used for specific surface defects and it is difficult to generalize to other metal surfaces. In this work, a theory of few-shot metal generic surface defect segmentation is introduced to solve these challenges. Simultaneously, the Triplet-Graph Reasoning Network (TGRNet) and a novel dataset Surface Defects-4^i are proposed to achieve this theory. In our TGRNet, the surface defect triplet (including triplet encoder and trip loss) is proposed and is used to segment background and defect area, respectively. Through triplet, the few-shot metal surface defect segmentation problem is transformed into few-shot semantic segmentation problem of defect area and background area. For few-shot semantic segmentation, we propose a method of multi-graph reasoning to explore the similarity relationship between different images. And to improve segmentation performance in the industrial scene, an adaptive auxiliary prediction module is proposed. For Surface Defects-4^i, it includes multiple categories of metal surface defect images to verify the generalization performance of our TGRNet and adds the nonmetal categories (leather and tile) as extensions. Through extensive comparative experiments and ablation experiments, it is proved that our architecture can achieve state-of-the-art results.**

***Index Terms*— Few-shot semantic segmentation, metal generic surface defect segmentation, triplet graph reasoning network (TGRNet).**

I. Introduction

METAL surface defect inspection [1]–[5] can play a significant role in addressing the issue of product quantitative inspection in the industry. For quantitative metal surface defect inspection, there are two main different network architectures, detection network [6], [7] and segmentation network [8], [9]. Although many detection networks have reached a certain level of accuracy and speed, their prediction results only are bounding boxes, which cannot meet the high precision requirements in the industry. Therefore, the segmentation network of metal surface defects has received a lot of attention in recent years. However, the existing metal surface defect segmentation network has two challenges. First, metal surface defect images are often difficult to collect and label in industrial scenes, but pixel-level segmentation often requires large amounts of data to train. Therefore, existing fully supervised metal surface defect segmentation algorithms are difficult to achieve satisfactory results. Second, most of the existing metal surface defect segmentation works are only used for specific surface defects, such as aluminum [10], magnetic tile [11], rail [12]–[14], etc., which means it is difficult to generalize them to other metal surfaces. This also means that metal generic surface defect segmentation has been a largely underexplored domain.

To deal with the first challenge, the theory of few-shot learning is introduced to compensate for the lack of metal surface defect images. The existing few-shot semantic segmentation methods use the known support set to segment the query set of the same class. In extreme cases, the defect in the query set images (unlabeled) can be segmented only by using the K ($K \geq 1$) support images (labeled) defect and this method can achieve the state-of-the-art effect.

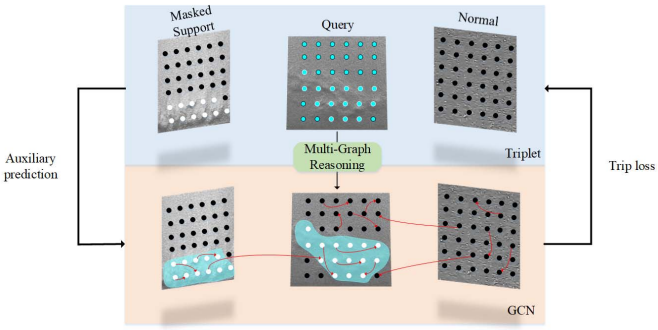


Fig. 1. Structure of TGRNet, it uses surface defect triplet, multigraph reasoning, and adaptive AP module to achieve few-shot generic surface defect segmentation (C-way N-shot W-normal).

Manuscript received March 4, 2021; revised April 23, 2021; accepted May 12, 2021. Date of publication May 25, 2021; date of current version June 9, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 51805078 and in part by the Fundamental Research Funds for the Central Universities under Grant N2103011 and Grant N2003021. The Associate Editor coordinating the review process was Dr. Lei Zhang. *(Corresponding authors: Kechen Song; Yunhui Yan.)*

Yanqi Bao, Kechen Song, Jie Liu, Yanyan Wang, and Yunhui Yan are with the School of Mechanical Engineering and Automation, Northeastern University, Shenyang 110819, China, and also with the Key Laboratory of Vibration and Control of Aero-Propulsion Systems Ministry of Education of China, Northeastern University, Shenyang 110819, China (e-mail: baoyq_neu@163.com; songkc@me.neu.edu.cn; liujie.neu.edu@gmail.com; wangyanyan_neu@163.com; yanyh@mail.neu.edu.cn).

Han Yu and Xingjie Li are with the State Key Laboratory of Light Alloy Foundry Tech for High-end Equipment, Shenyang 110027, China (e-mail: yuhan@chinasrif.com; lixj@chinasrif.com).

Digital Object Identifier 10.1109/TIM.2021.3083561

To deal with the second challenge, normal images are introduced to assist few-shot semantic segmentation. In the field of industrial defect segmentation, due to the large number of normal images, we propose adding $W(W \geq 1)$ normal images to the original setting of few-shot semantic segmentation. In essence, we divide the few-shot metal surface defect segmentation problem into the defect area and background few-shot semantic segmentation. For defect area few-shot semantic segmentation, same as the traditional few-shot semantic segmentation, the known defect areas of the support set are used to find similar areas in the query set. For background few-shot semantic segmentation, we treat normal images as support sets of background information. In other words, the known background areas of the normal images are used to find similar areas in the query set for the background areas. Finally, the defect area and background area of the query set are fused to obtain the final surface defect segmentation result. Through the recognition of the background area and defect area simultaneously, the generalization ability for different material defects is improved due to the similarity and regularity of the background area. In addition, to prove that our proposed theory and method, a novel dataset Surface Defects-4^i, is constructed. The Surface Defects-4^i contains aluminum, steel, rails, and magnetic tiles that belong to common metal surface defects and adds the nonmetal classes (leather and tile) as extensions to further prove generalization ability. In addition, the use of nonmetal classes can increase the diversity of backgrounds (leather and tile) and also increase the difficulty of the overall task. That is to say, the few-shot metal generic surface defect segmentation we proposed can make stable inferences about the similarity of intraclass, and has robustness for the difference of interclass. The dataset is randomly divided threefold for cross-validation. This ensures that the proposed method can be used to segment different classes of surface defects in different materials.

In general, a theory of few-shot generic surface defect segmentation is introduced to solve the two challenges mentioned above. To achieve it, a novel Triplet-Graph Reasoning Network (TGRNet) is proposed, as showed in Fig. 1. Specifically, different from the work of face recognition [15], [16], we propose the surface defect triplet (including triplet encoder and trip loss). Most existing works focus on defect areas and ignore the background information. However, it is known that the nondefect regions are highly similar and easy to distinguish on the industrial defect image. Therefore, we use surface defect triplet to segment the defect area and the background area simultaneously to achieve higher accuracy and improve generalization ability. In order to segment defect and background areas, the multigraph reasoning module is proposed to achieve better results of few-shot semantic segmentation. The multigraph reasoning module explores the similarity between different image regions by embedding features into the graph space and using graph convolutional network (GCN) for graph reasoning. In addition, we also proposed an adaptive auxiliary prediction (AP) module to improve the final metal surface defect segmentation performance.

The contributions of this article can be summarized as follows:

1) To the best of our knowledge, it is the first time that the theory of few-shot metal generic surface defect segmentation is introduced (C-way N-shot W-normal).
2) A novel surface defect triplet (including triplet encoder and trip loss) is proposed to segment the defect area and the nondefect area simultaneously in the field of defect segmentation.
3) A novel multigraph reasoning module is proposed to solve the problem of few-shot semantic segmentation by exploring the similarities between images.
4) We propose a novel dataset Surface Defects-4^i, and use existing few-shot semantic segmentation methods to conduct a large number of ablation experiments and comparison experiments.
5) A novel adaptive AP module is proposed to improve model performance in the industrial scenario.

In the remainder of the work, Section II proposes related work in surface defect segmentation, few-shot semantic segmentation, and GCN in recent years. Section III describes the detailed structure of TGRNet. Section IV describes the dataset Surface Defects-4^i and the setup of the experiment. In Section V, we conduct comparative experiments and ablation experiments, and the conclusion and future work are in Section VI.

II. Related Work

In this section, we briefly review some related works on surface defect segmentation, few-shot semantic segmentation, and GCN.

A. Surface Defect Segmentation

In the field of industrial inspection, surface defect segmentation has high precision, so it has received a lot of attention in recent years. Bian *et al.* [17] proposed a multiscale fully convolutional network to segment the defect of aeroengine blades. Yu *et al.* [18] proposed two-stage fully convolutional networks to predict defect areas in an industrial environment. Tabernik *et al.* [19] designed a segmentation-based deep-learning architecture on a specific domain of surface-crack detection. The first stage used a segmentation network to segment defect areas, and the second stage includes an additional decision network to predict whether the entire image is abnormal. Liong *et al.* [20] proposed a method of segmenting leather defects by employing both the convolution and deconvolution neural networks. Delconte *et al.* [21] proposed a new structure using relief map image and convolutional neural network to segment wood defect. And Xie *et al.* [22] proposed a main net and secondary net for defect segmentation of textured surfaces. The secondary net is used to extract features in the frequency domain, and the main net is used to extract features in the spatial domain and fuse the features extracted from the secondary net. In addition, Wu *et al.* [9] designed a ResMask GAN framework to generate generic defect images and a coarse-to-fine module to detect and segment generic defects.

From the above works, it is not difficult to find that the most existing methods are often aimed at a specific material

or defect. In other words, these works cannot be applied to multiple defects of different materials. Therefore, the study of generic surface defect segmentation as [9] is of great significance.

B. Few-Shot Semantic Segmentation

Few-shot semantic segmentation receives a lot of attention in recent years. OSLSM [23] proposed the concept of few-shot semantic segmentation, in which the dual branch network is used. SG-One [24] proposed the mask global average pooling to reduce background information and used cosine similarity to guide the segmentation branch to segment target. In recent years, (CANet) [25], a novel dense comparison module, has been proposed. It effectively uses the multilevel feature extracted from CNN for dense feature comparison and proposed an iterative optimization module to improve prediction performance. PGnet proposed by Zhang *et al.* [26] uses a multiscale graph attention module to propagate similarity information of nodes between two images. Tian *et al.* [27] proposed PFENet, which used prior information that is free-training and a horizontal and vertical feature fusion module to fuse features of cross-image and cross-scale for few-shot semantic segmentation. And the PMMs [28] were proposed recently, which used the method of expectation–maximization to calculate multiple foreground and background prototypes for metric learning. In addition, the FSS-1000 [29] constructed a few-shot semantic segmentation dataset, which includes 1000 classes of few-shot images and uses existing methods to conduct sufficient experiments.

C. Graph Convolutional Network

Analog convolution operation, graph convolution, was introduced by Bruna *et al.* [30] for the first time. GCN is mainly aimed at graph structures and used in non-European space, such as social networks, chemical molecular structure, knowledge maps, etc. [31]. After that, a lot of works proposed improvements on the original GCN to improve computing efficiency [32] and stable optimization process [33]. Recently, there have been many articles [34] using GCN in the field of image processing and achieved satisfactory results. For features extracted from images, how to embed them to graph space nodes is the main research problem. For this question, graph-based global reasoning network [35] was proposed, and it uses graph reasoning to extract global information for downstream tasks. After embedding features to graph space nodes, they used GCN to update node status. Then, Li *et al.* [34] proposed SpyGR to apply the idea of graph reasoning to semantic segmentation. Through these articles, we found that GCN can break the spatial constraints of image features to extract local features. Therefore, GCN can be utilized to break the boundaries between images and explore the relationship between multiple images in few-shot semantic segmentation problems.

III. Method

A. Problem Setting

As the problem setting proposed by Shaban [23], few-shot semantic segmentation contains two sets: a training set D_{train} and a testing set D_{test}. Our model is trained on the training dataset D_{train} and evaluated on the testing dataset D_{test}. We divide all samples into C_{seen} and C_{unseen}. Assuming that the training dataset D_{train}only contains the C_{seen}, and the D_{test} only contains C_{unseen} in episodes, that is to say $C_{\text{seen}} \cap C_{\text{unseen}} = \emptyset$, we further define our dataset according to the following rules.

1) $D_{\text{train}} = (x_i^S, x_i^Q, x_i^U, y(c)_i^S, y(c)_i^Q)_{i=1}^N$, among that, x_i^S is the support image, x_i^Q is the query image, and x_i^U is the normal image. Note that x_i^Q and x_i^U must belong to the same class with x_i^S. $y(c)_i^S$ and $y(c)_i^Q$ are the mask corresponding to x_i^S and x_i^Q, $c(c \in C_{\text{seen}})$ represents the category of training, and N represents the number of training episodes.
2) $D_{\text{test}} = (x_j^S, x_j^Q, x_j^U, y(c)_j^S)_{j=1}^M$, same as D_{train}, x_j^S, x_j^Q and x_j^U represent support, query, and normal images, respectively, in the testing phase. $y(c)_j^S$ represents the mask corresponding to the support image, c represents the category in the test set, which means $c \in C_{\text{unseen}}$. M represents the number of episodes in the testing phase.

Same as the existing few-shot semantic segmentation setting methods [23], [27], the task has been appropriately simplified. There is only one class defect (one-way) in the support images (labeled) and the query image (unlabeled), and the other classes of defects will be regarded as background. If the support set includes K-labeled support images and W normal images, we can call such few-shot generic surface defect segmentation problem as one-way K-shot W-normal generic surface defect segmentation task. Specifically, each episode consists of K supporting images, W normal images, and one query image for training and testing model. Through multiple episodes, the generalization and distinguishing ability of our model can be improved in the case of few-shot.

B. Proposed Model

A novel TGRNet is proposed to achieve few-shot generic surface defect segmentation, as showed in Fig. 2. Our motivation is to use the support images and the normal images to segment the surface defects of the query image. The main innovation is that we propose a novel surface defects triplet network (including triplet encoder and trip loss) for defect segmentation and a novel multigraph reasoning module for few-shot semantic segmentation. The former uses the defect information of the support image and background nondefect information of the normal image to segment the surface defects and background of the query image. The latter uses the graph node relationship to reason the similarity between the images. In addition, a new adaptive AP module is proposed to further improve the performance of segmentation in our network.

C. Triplet Encoder

Different from existing work [15], surface defects triplet encoder is tried to extract features of support, query, and normal image, simultaneously. Using the surface defects triplet encoder not only reduces the number of parameters for optimization, but also ensures that the features of input are fixed

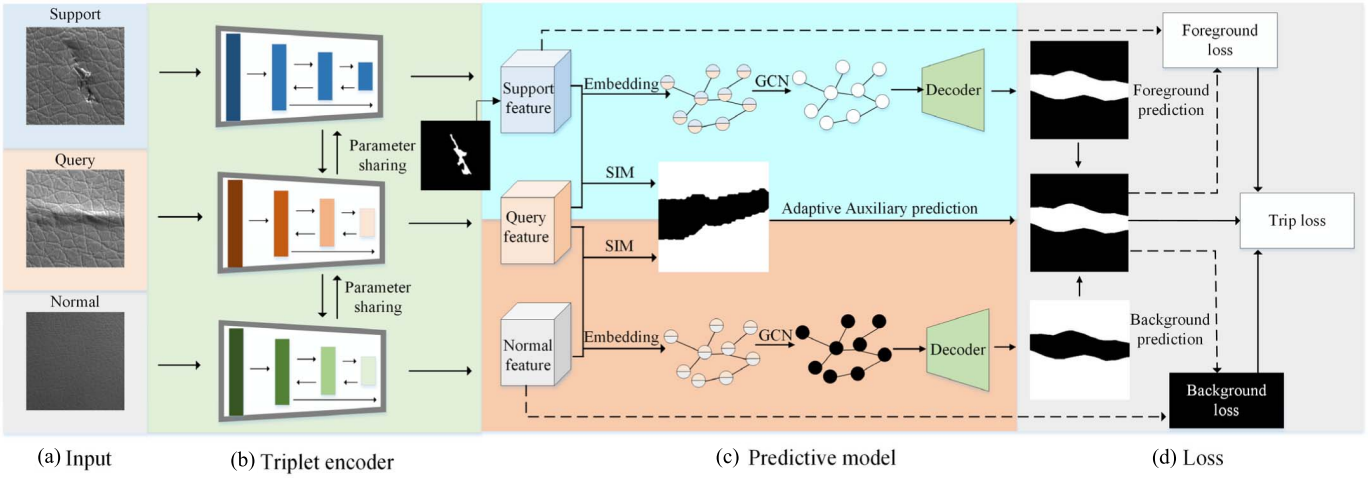


Fig. 2. In our proposed TGRNet, first, (a) extract features of support, query, and normal images through the (b) surface defects triplet encoder, then it uses multigraph reasoning module to predict foreground and background separately. In addition, (c) adaptive AP module is used to further improve the performance by calculating similarity. Finally, (d) it uses trip loss as an auxiliary loss to optimize network parameters.

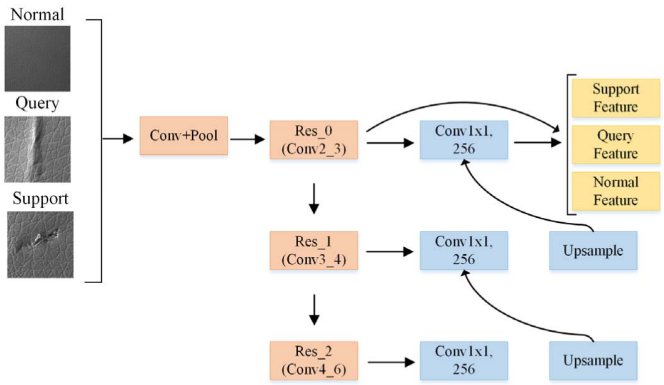


Fig. 3. Detailed illustration of the triplet encoder.

in the same feature subspace during the calculation process of trip loss.

For surface defect images, detailed texture features have great discriminating ability for defect segmentation, it is obvious that shallow features have more detailed information than deeply features. Therefore, our triplet encoder uses shallow features as the main features and deep features as auxiliary features in the feature extraction stage, as shown in Fig. 3. Specifically, our triplet encoder uses the pyramid structure to gradually fuse deep features into shallow features. Through experiments, we found that the features of Res-0 can characterize the image detailed information greatly. Therefore, we fuse the global features of Res-1 and Res-2 into the detailed features of Res-0 through convolution and upsampling. In addition, for supporting features, we multiply it with the corresponding mask at pixel level to get defect area features.

Here, we use ResNet-50 as the backbone network and train it on ImageNet [36] as the pretraining network for feature extraction. For surface defects triplet encoder, we share the encoder parameters for support images, query images, and normal images. There are industrial surface defect images and the distribution of the images is quite different, so we do not fix the backbone network parameter during the training process.

D. Multigraph Reasoning

Graph structure is composed of several nodes and edges that connect two nodes, and is used to describe the relationship between different nodes. Usually, the graph structure can be expressed as $\mathcal{G} = (V, E)$, where V represents the vertex and E represents the edge. Here, we define A and D as the adjacency matrix and the degree matrix of the graph structure. According to [27], the process of graph convolution can be expressed as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}\Theta^{(l)}\right) \tag{1}$$

where, $\tilde{A} = I + A$, I is the identity matrix, and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $H^{(l)}$ means the feature of lth layer and $H^{(l+1)}$ means the feature of $l + 1$th layer. $\Theta^{(l)}$ represents the trainable parameters corresponding to the features of the lth layer. In addition, σ represents nonlinear activation function. To simplify the representation, we introduce graph Laplacian matrix:

$$\tilde{L} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \tag{2}$$

Therefore, Eq. (1) can be regarded as:

$$H^{(l+1)} = \sigma\left(\tilde{L}H^{(l)}\Theta^{(l)}\right) \tag{3}$$

For image data, the features obtained by the surface defects triplet encoder should first be embedded into the graph space. As shown in Fig. 4, assume that $F_S, F_Q, F_N \in R^{C\times W\times H}$ are masked support feature, query feature, and normal feature output from the surface defects triplet encoder, respectively, where C is the number of channels, W is the width of the feature, and H is the height of the feature. We simplify the graph structure relationship $\mathcal{G}_{SQN}$ to the graph structure relationship $\mathcal{G}_{SQ}$ and the graph structure relationship of $\mathcal{G}_{SN}$. That is to say, the defect area is segmented by $\mathcal{G}_{SQ}$, and the background is segmented by $\mathcal{G}_{SN}$.

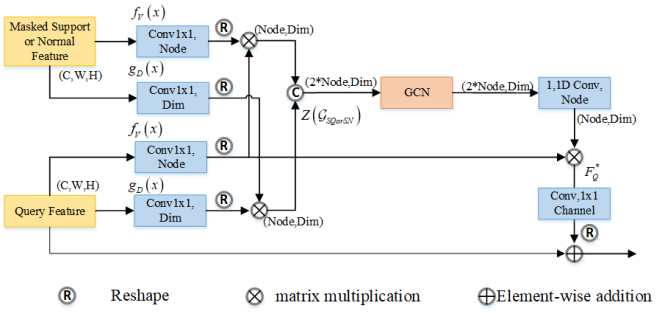


Fig. 4. Detailed illustration of the multigraph reasoning.

For $\mathcal{G}_{SQ}$, firstly, the support and query features are embedded in the nodes ($F_{S_V}, F_{Q_V} \in R^{V\times W\times H}$) and dimensions ($F_{S_D}, F_{Q_D} \in R^{D\times W\times H}$) interaction subspace through the mapping functions $f_V(x)$ and $g_D(x)$, respectively. Then, they are reshaped to $F_{S_V}, F_{Q_V} \in R^{V\times L}$ and $F_{S_D}, F_{Q_D} \in R^{D\times L}$, where $L = W \times H$. Different from existing work [35], in order to ensure that two different images are embedded in the same graph space, we use the method of cross embedding:

$$Z(\mathcal{G}_{SQ}) = \left(F_{S_V} \times F_{Q_D}^T\right) | \left(F_{Q_V} \times F_{S_D}^T\right) \tag{4}$$

where $\times$ represents matrix multiplication, $|$ represents concatenate at the node level, and $Z(\mathcal{G}_{SQ}) \in R^{2V\times D}$ is the matrix representation of graph space. If we use cross embedding, we believe $\mathcal{G}_{SQ} = (2V, 4E)$. If not use cross embedding and concatenate directly, it will lose the interaction information between the two images, $\mathcal{G}_{SQ} = (2V, 2E)$. It should be noted that we use convolution instead of $f_V(x)$ and $g_D(x)$, and share parameters of the F_S and F_Q.

After converting the image features to the graph space, we use Eq. (3) to graph convolution for updating graph nodes. It is known that graph convolution can get the similarity relationship between nodes. Therefore, the defective area of the query image similar to the defective area of the support image will be found through graph convolution. Specifically, due to the large number of multigraph nodes, we treat the adjacency matrix as a learnable matrix to reduce the amount of calculation. We use the 1-D convolution in the two directions of $\mathcal{G}_{SQ}$ to replace the matrix multiplication ($\tilde{L}$ and $\Theta^{(l)}$) in Eq. (3). Then 1-D convolution at the level of the node is used to fuse the $2V$ nodes in the $\mathcal{G}_{SQ}$to the $\mathcal{G}_Q = (V, E)$. For $\mathcal{G}_Q$ graph space structure, we use F_{Q_V} to re-project the graph space to the feature space:

$$F_Q^* = \left(F_{Q_V}^T \times Z(\mathcal{G}_Q)\right)^T \tag{5}$$

where, $F_Q^* \in R^{D\times L}$, we flatten F_Q^* to $D \times W \times H$ and use convolution to change the number of channels to get the final $F_Q^* \in R^{C\times W\times H}$. Finally, the residuals connect F_Q^* and F_Q for optimization.

We use the three residual connections and Atrous spatial pyramid pooling [37] (ASPP) to process the output of the multigraph reasoning module to obtain the segmentation result of the defect area. For the graph structure $\mathcal{G}_{SN}$ composed of the support image and the normal image, we use the same method as $\mathcal{G}_{SQ}$ for multigraph reasoning and obtain the segmentation result of the background. Finally, we subtract the obtained defect area prediction and background prediction to get the final segmentation result P_{graph}.

E. Adaptive AP

The essence of few-shot semantic segmentation is to find a similar part between the support image and the query image. Therefore, we directly use the similarity between features for AP that is free-training. Same as the multigraph reasoning module, we still divide this module into a defect area part and a background part. For the defect area, we use the similarity of the masked support feature F_S and the query feature F_Q to predict the defect area. First, we explore the similarity between F_S and F_Q, and enrich the similar part of them at the channel level. Specifically, we flatten them to $F_S, F_Q \in R^{C\times L}$ and calculate the similarity matrix between them as:

$$S_C = F_S \times F_Q^T \tag{6}$$

where, $S_C \in R^{C\times C}$. And then matrix multiplication of S_C and F_S (or F_Q) is used to enrich similarity features as

$$F_{S_C}(F_{Q_C}) = S_C \times F_S(F_Q) \tag{7}$$

where, $F_{S_C}, F_{Q_C} \in R^{C\times L}$ represents enriched features. The similarity information between F_{S_C} and F_{Q_C} enriched at the channel level, thereby retaining a large amount of similar information and removing redundant information.

Then we explore the most similar point of F_{S_C} for F_{Q_C} at the spatial level. We use the cosine similarity of the vector to express the similarity between $f_j \in F_{S_C}$ and $f_i \in F_{Q_C}$ as:

$$S_{\text{COS}(i,j)} = \frac{f_i^T f_j}{\|f_i\| \|f_j\|} \quad i, j \in \{1, 2, \ldots, L\} \tag{8}$$

where, $S_{\text{COS}} \in R^{C\times C}$, S represents the similarity between the ith element in F_{Q_C} and the jth element in F_{S_C}. Then we find the maximum value in the direction of the column to find the most similar point. Finally, we reshape it to get $P_F \in R^{W\times H}$ and normalize it by min-max normalization to between 0 and 1.

For the background, the similarity between the background of query feature F_Q and the normal feature F_N is calculated to predict the background area $P_B \in R^{W\times H}$ by the same method as the defect area.

Finally, we concatenate obtained P_F and P_B together to get $P_{\text{AUX}} \in R^{2\times W\times H}$, then use adaptive coefficients to fuse P_{graph} that obtained from multigraph reasoning to P_{AUX} as:

$$P = \partial(P_F | P_B) + P_{\text{graph}} \tag{9}$$

where, ∂ represents trainable parameters and is initialized to 0.

F. Trip Loss

In order to update the obtained prediction, we need to back-propagate through the loss function. For L_{main}, similar to most few-shot semantic segmentation work, cross-entropy is used. In addition, we also introduce surface defects trip loss as L_{AUX} for triplet, as shown in Fig. 5. For the relationship between the three of support, query, and normal, we use the

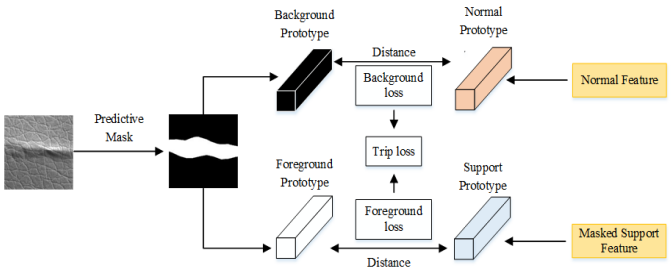


Fig. 5. Detailed illustration of the trip loss.

defect area prototype distance between support and query, and the background prototype distance between query and normal to calculate L_{AUX}. The L_{AUX} is also called trip loss. The trip loss can optimize the parameters to alleviate the situation where there is a large loss in foreground prediction or background prediction. Specifically, we calculate the prototype for the defect area of F_S, the defect area and background of F_Q, and F_N as:

$$Pt = \frac{\sum_{w=1,h=1}^{W,H} y(c) * F}{\sum_{w=1,h=1}^{W,H} y(c)} \tag{10}$$

where, $*$ represents element-wise multiplication. For the defect area prototype Pt_{SF} of F_S, $y(c)$ represents support mask $y(c)_i^S$, F represents F_S. For the defect area (and background) prototype Pt_{QF} (and Pt_{QB}) of F_Q, $y(c)$ represents predicted defect area (and background) mask P (and $1 - P$), F represents F_Q. For the prototype Pt_N of F_N, $y(c)$ represents a matrix with all elements of 1, F represents F_N. L_{AUX} is calculated by:

$$L_{\text{AUX}} = \text{distance}\left(Pt_{QF} - Pt_{SF}\right) + \text{distance}\left(Pt_{QB} - Pt_N\right) \tag{11}$$

where, distance represents the two-norm of distance. And then Loss represented as:

$$\text{Loss} = L_{\text{main}} + kL_{\text{AUX}} \tag{12}$$

where, k represents a hyperparameter and is set to 0.2 in the article. We proved its superiority in ablation experiments.

For the sake of simplicity, it is described by Algorithm 1.

Algorithm 1 Training and evaluating TGRNet

Input: a training dataset D_{train} and a testing dataset D_{test}
Output: Trained parameters of TGRNet
for each episode $(x_i^S, x_i^Q, x_i^U)_{i=1}^N \in D_{\text{train}}$ **do:**
  Extract F_S, F_Q, F_N with surface defect triplet encoder;
  Project to graph space $\mathcal{G}_{SQ}$ and $\mathcal{G}_{SN}$ using Eqns.4;
  Use GCN for node update using Eqns.1-3;
  Re-project back to the image space using Eqns.5 and use residual connections and Atrous Spatial Pyramid Pooling to get P_{graph};
  Enrich similarity features using Eqns.6-7;
  Calculate P_{AUX} using Eqns.8;
  Get the final predicted P through an adaptive algorithm using Eqns.9;
  Compute trip loss L_{AUX} using Eqns.10-12 and Cross Entropy L_{main};
  Compute the gradient and optimize via SGD;
end
for each episode $(x_j^S, x_j^Q, x_j^U)_{j=1}^N \in D_{\text{test}}$ **do:**
  Extract F_S, F_Q, F_N with surface defect triplet encoder;
  Get P_{graph} through graph reasoning using Eqns.1-5;
  Get segmentation result P through adaptive auxiliary prediction using Eqns.6-9;
end

IV. Experiments

A. Dataset

To evaluate our method, we construct a novel dataset, Surface Defects-4^i, that uses images and annotations of multiple metal surface defect dataset. In addition, we also add two classes of nonmetal as an extension and further prove the generality of our method. Specifically, before experimenting, we grayscale all images and uniform size to 200×200 for ensuring consistency. There are a total of 12 different classes of surface defects in the dataset. Each class includes defective images, groundtruth (GT), and a large number of normal images. Referring to [23], we divide the 12 classes of datasets into three splits for cross-validation. The details are as following:

TABLE I
Details of Dataset

Dataset Surface Defects-4^i	Test classes
Surface Defects-4^0	Al-I(23), MT-I(61), Steel-I(48), Steel-II(50)
Surface Defects-4^1	Leather(53), Steel-III(50), Rail(66), Steel-IV(50)
Surface Defects-4^2	Al-II(40), MT-II(57), MT-III(26), Tile(37)

In the Table I, Al-I and Al-II represent defects of rub mark and convexity on the aluminum surface, respectively. MT-I, MT-II, and MT-III represent defects of uneven, break, and fray on the magnetic tile that was constructed in [38]. Steel-I, Steel-II, Steel-III, and Steel-IV represent defects of liquid, abrasion mask, patches, and scratches on the steel surface, respectively. In addition, leather and tile [39] are two kinds of nonmetal data as an extended part of our work. For leather, tile, and rail, although there are multiple types of defects, the difference in defect shape is small, which is regarded as one class in the dataset. We record the number of each category defective images in parentheses. It can be found that the number of defect images is small and the number gap between the categories is large. That is to say, the fully supervised algorithm and the traditional few-shot semantic segmentation algorithm cannot achieve great results.

As shown in Fig. 6, compared with the existing surface defect dataset, our proposed Surface Defects-4^i contains aluminum, steel, rails, magnetic tiles, tile, and leather. By setting this dataset, we can ensure that each split contains many different defects in different materials and verify our proposed

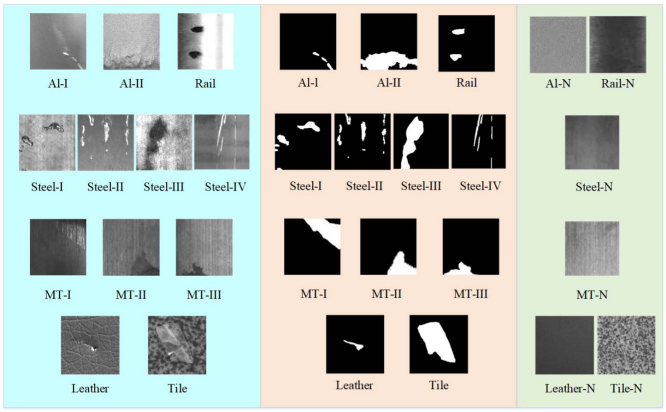


Fig. 6. Examples of defect images, corresponding GT, and normal of 12 defect classes in Surface Defects-4^i.

TGRNet has strong generalization performance for surface defects of different materials. In the process of training and testing, we randomly select K support set images, one query set image of the same category in the defect images, and W normal images.

B. Evaluation Metric

Previous few-shot semantic segmentations have mentioned two different evaluation metrics: mIoU and foreground and background intersection-over-union (FB-IoU). Specifically, they always used the (IoU) to describe the accuracy of each class foreground prediction, and used mIoU to calculate the average IoU of all classes. FB-IoU ignores the class information and calculates the mean value between the foreground IoU and the background IoU. In this article, we have tested the above two metrics, respectively, to comprehensively evaluate the performance of our proposed method.

C. Experimental Setup

Our network uses an end-to-end training mode, and the loss function uses cross-entropy and trip loss for backpropagation. Our network uses the SGD optimizer to train 200 epochs uniformly under the pytorch network framework. We used Nvidia RTX3080 (10G) and pytorch 1.7.1 for network training on Ubuntu system, set the training batchsize to 2 and test batchsize to 1, and unified the input image size to 200×200. During the experiment, the value of k was set to 0.2. And during the training process, the pretraining network will be fine-tuned, and its learning rate is one-tenth of the network learning rate. The Surface Defects-4^i dataset and our codes are available at the Github homepage (https://github.com/bbbbby-99/TGRNet-Surface-Defect-Segmentation).

D. Comparative Experiment

Since the existing few-shot semantic segmentation methods are all tested with C-way K-shot, in the process of the comparative experiments, we add the case of not adding normal images for comparison (C-way K-shot 0-normal). On this basis, we also compared the results of C-way K-shot one-normal to prove the superiority of adding normal images. It should be noted that due to the particularity of surface defect images, all our comparative experiments are set to one-way. To ensure the fairness of the experiment, when dealing with five-shot, we take the average of five supporting features and synthesize them into one feature for our network.

As shown in Table II, we use mIoU and FB-IoU to compare our method with existing few-shot semantic segmentation methods from the perspective of one-shot and five-shot. From the results, it can be found that we proposed TGRNet has achieved more state-of-the-art results than the existing methods from the perspective of mIoU and FB-IoU without adding normal images. For the backbone, it can be found from the table that the performance of resnet-50 is much higher than that of VGG-16, which proves the superiority of the triplet encoder. In addition, it can be found that after adding normal images, our result performance has also been greatly improved.

E. Ablation Study

1) Ablation Study: In Table III, the impact of each part of our proposed model on the performance of the result for all indicators can be clearly seen. Take one-shot mIoU as an example, it increased mIoU by 3.09% by using the multi-graph reasoning (using GCN to indicate in the table), indicating that using the multigraph reasoning can effectively find the similarity information between two images to solve the problem of few-shot semantic segmentation. Then it increased mIoU by 4.53% by adding a normal image indicating that using normal images to segment defect-free areas can greatly improve performance. For adaptive AP, it can bring a performance improvement of 0.67 for mIoU. At last, it increased mIoU by 1.58% by using the trip loss, indicating that using trip loss as an auxiliary loss can further improve network optimization performance. We think this may be because the trip loss we proposed improves the common accuracy of foreground prediction and background prediction. In total, TGRNet improves performance by 9.87% which is a significant improvement in semantic segmentation. This demonstrates the superiority of each part of TGRNet.

2) Ablation Experiment of Triplet Encoder: First, we conduct ablation experiments for different Resnet networks to verify the superiority of our backbone as shown in Table IV. It can be seen that Resnet-50 has better performance than other Resnet networks. For Resnet-34 and Resnet-18, they can only get shallow information, but our feature extractor needs the assistance of a deeper global feature (proved below). For Resnet-101 and Resnet-152, they have too many parameters that make it difficult to optimize in the context of a small number of samples.

Second, for few-shot semantic segmentation, the previous work [25] tends to focus on the features of the middle level rather than the deep features, because deep features often lose much detailed information. Therefore, we used Res-0, Res-1, and Res-2 as features for comparing in ResNet-50 (give up Res-3).

As the results (only have triplet encoder and multi-graph reasoning) are displayed in Table V, through the experimental

TABLE II
CLASS mIoU AND MEAN FB-IoU RESULTS ON THREEFOLD OF SURFACE DEFECTS-4^I

Method	1-shot mIoU				5-shot mIoU				Mean FB-IoU(1/5-shot)
	Fold-0	Fold-1	Fold-2	Mean	Fold-0	Fold-1	Fold-2	Mean	
VGG-16 Backbone									
SG-One	17.30	14.90	13.46	15.22	17.43	15.23	13.61	15.42	47.40/49.10
PFENet	24.16	20.19	20.21	21.52	32.56	33.38	26.10	30.68	47.87/53.23
Ours(5-normal)	46.78	26.88	22.07	31.91	52.07	30.04	25.04	35.72	55.74/57.63
ResNet-50 Backbone									
CANet	25.32	20.12	12.59	19.34	27.12	22.36	14.32	21.27	49.40/53.10
PGNet	20.04	19.37	21.33	20.25	22.30	20.03	21.63	21.32	48.89/50.01
PMMs	30.76	19.55	11.45	20.59	30.52	20.75	19.67	23.64	58.87/61.32
PFENet	28.45	34.08	19.94	27.49	37.71	34.68	22.59	31.66	53.46/54.06
Ours(0-normal)	45.11	30.13	23.15	32.80	44.06	34.97	23.82	34.82	55.85/53.93
Ours(1-normal)	56.75	**35.00**	26.98	**39.58**	58.44	34.33	28.91	40.56	**57.46/61.61**
Ours(5-normal)	**60.08**	30.59	**27.76**	39.48	**59.25**	**34.80**	**32.48**	**42.18**	56.78/59.75

TABLE III
ABLATION STUDY OF ONE-SHOT/FIVE-SHOT mIoU AND MEAN FB-IoU

Triplet encoder	GCN	Normal(1)	AP	Trip loss	1-shot/5-shot mIoU				1-shot/5-shot Mean FB-IoU
					Fold-0	Fold-1	Fold-2	Mean	
✓					41.90/46.25	25.25/33.21	21.99/23.66	29.71/34.37	52.99/57.48
✓	✓				45.11/44.06	30.13/34.97	23.15/23.82	32.80/34.28	55.85/56.96
✓	✓	✓			54.29/56.24	34.14/30.93	23.57/28.02	37.33/38.40	56.03/56.93
✓	✓	✓	✓		54.71/58.83	33.79/29.14	25.51/**29.06**	38.00/39.01	56.55/57.85
✓	✓	✓	✓	✓	**56.75/58.44**	**35.00/34.33**	**26.98**/28.91	**39.58/40.56**	**57.46/61.61**

TABLE IV
ONE-SHOT mIoU AND MEAN FB-IoU OF ABLATION STUDY FOR *Resnet*

Resnet	mIoU				Mean FB-IoU
	Fold-0	Fold-1	Fold-2	Mean	
Resnet-18	49.42	30.77	20.55	**33.58**	**53.19**
Resnet-34	48.12	31.09	22.34	**33.85**	**55.59**
Resnet-50	56.75	35.00	26.98	**39.58**	**57.46**
Resnet-101	46.04	30.37	26.70	**34.37**	**52.87**
Resnet-152	53.47	24.26	23.59	**33.77**	**53.55**

TABLE V
ABLATION STUDY OF TRIPLET ENCODER

Res_0	Res_1	Res_2	1-shot		5-shot	
			MIOU	MFB-IoU	MIOU	MFB-IoU
✓			30.52	54.25	33.81	55.90
	✓		28.18	50.40	30.90	53.83
		✓	25.84	48.99	25.96	49.26
✓(*)	✓(*)		31.75	52.32	32.60	50.39
	✓(*)	✓(*)	30.56	53.22	32.88	51.36
✓(*)	✓(*)	✓(*)	**32.80**	**55.85**	**34.28**	**56.96**

results of the first three rows, it is obvious that the feature of Res-0 has strong distinguishing ability. This may be because for surface defect images, the image structure is simple so that the image details feature extracted by shallow extractor is more distinguishable. From the last three lines of experimental results, we can find that fusing deep features into shallow features can have better performance. And fusing Res-1 and

TABLE VI
ONE-SHOT MIOU AND MEAN FB-IOU OF ABLATION STUDY FOR HYPERPARAMETER k

k	mIoU				Mean FB-IoU
	Fold-0	Fold-1	Fold-2	Mean	
0	54.71	33.79	25.51	**38.00**	**56.55**
0.2	56.75	35.00	26.98	**39.58**	**57.46**
0.4	51.70	28.99	18.03	**32.91**	**56.60**
0.6	54.28	21.70	25.70	**33.89**	**55.32**
0.8	54.02	27.71	24.50	**35.41**	**54.07**
1.0	50.10	27.58	24.50	**34.06**	**52.39**

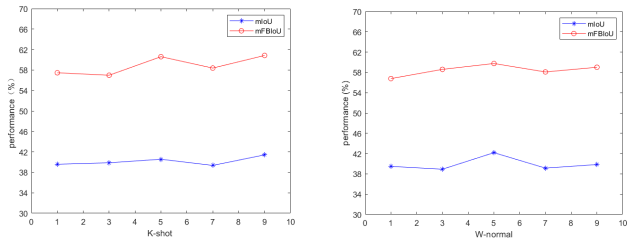


Fig. 7. Ablation experiment of K and W.

TABLE VII
ONE-SHOT OIOU AND MEAN FB-IOU OF ABLATION STUDY FOR *Main Loss*

Loss	mIoU				Mean FB-IoU
	Fold-0	Fold-1	Fold-2	Mean	
CE Loss + Trip Loss	56.75	35.00	26.98	**39.58**	**57.46**
Focal Loss + Trip Loss	41.34	29.27	24.40	**31.67**	**52.87**
Dice Loss + Trip Loss	36.76	31.95	27.84	**32.18**	**53.59**

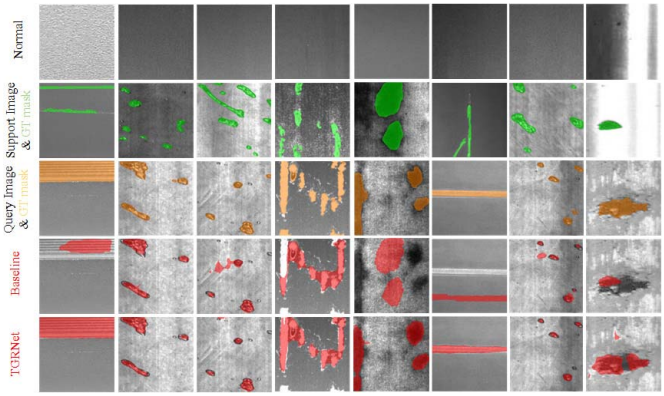


Fig. 8. Visualization of the results for baseline and TGRNet of metal. From top to bottom, each row represents normal, support images and their corresponding GT masks (green), query images and their GT masks (yellow), baseline predictions (red), TGRNet predictions (red), respectively.

Res-2 into Res-0 can get the best performance, which is 2.27% mIOU (5.45% mFB-IOU) higher than using Res-0 only.

3) Ablation Experiment of Hyperparameter k: k is a parameter that controls the ratio of L_{main} and L_{AUX} in *Loss*. It has a great influence on the performance of surface defect segmentation.

As the results are displayed in Table VI, take one-way one-shot one-normal as an example, it can be found that when $k = 0.2$, our proposed TGRNet can achieve the best performance. This may be due to the larger value of trip loss than L_{main}. If larger k is used, the effect of L_{main} may be reduced.

4) Ablation Experiment of K and W : For the number of support images K, we average the K masked support features to input the network. From the left image of Fig. 7 (fix $W = 1$), it can be found that as the value of K increases, the performance of the network is about to increase. This is similar to the results of other few-shot semantic segmentation works. For the number of normal images W, we average the W normal images directly and then put them into the encoder. From the right image of Fig. 7 (fix $K = 5$), it can be found that when the value of W is 5, the network can obtain the best performance. This may be because as the value of W increases, the amount of network calculations rises linearly, resulting in performance degradation.

5) Ablation Experiment of Main Loss: Recently, many different losses have been proposed to replace cross-entropy loss in the field of segmentation. They are usually used to alleviate the problem of imbalance between positive and negative samples in the foreground and background (such as Focal loss). Therefore, we conducted some ablation experiments to prove the superiority of our proposed loss as shown in Table VII. It can be seen that the cross-entropy loss has achieved better performance, which is probably due to the trip loss. For the trip loss mentioned in the article, the loss can be calculated separately from the perspective of the defect prediction and the background prediction, which can effectively alleviate the imbalance of positive and negative samples. On this basis, the cross-entropy loss seems to be easier to optimize than other losses. Therefore, the cross-entropy loss is used as the main loss, and works with trip loss in our network.

F. Visualization

As shown in Fig. 8, we visualize the metal prediction results of our proposed TGRNet and baseline. From the results, we can qualitatively prove the superiority of our proposed method.

As shown in Fig. 9, first of all, we visualize the nonmetallic prediction results, and it can be found that TGRNet can still achieve state-of-the-art results, which proves the generalization of our proposed algorithm. In addition, we visualize the failure case of the metal surface and analyze the reasons. It can be found that when the defect location is small and dense, or the gap between the support set and the query set is large, it may cause performance degradation. This may be due to

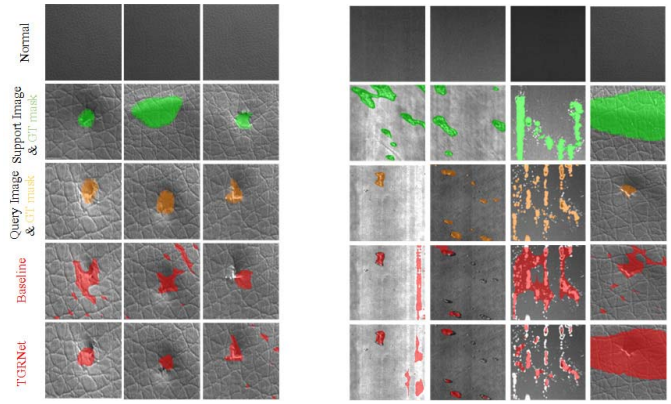


Fig. 9. Visualization of the results for baseline and TGRNet of nonmetal and failure case. From top to bottom, each row represents normal, support images and their corresponding GT masks (green), query images and their GT masks (yellow), baseline predictions (red), TGRNet predictions (red), respectively. From left to right, the first three columns indicate the prediction of nonmetal surface defects, and the last four columns indicate the prediction of defects on failure case.

the insufficient number of samples, causing the query image segmentation results to be too dependent on the support images and normal images. But because there is currently less work in the field of metal generic surface defect segmentation, we hope that our work could inspire some future work to address the aforementioned problems.

V. Conclusion

In the work, we introduce the theory of few-shot generic surface defect segmentation (C-way N-shot W-normal) to solve the challenge of insufficient samples and generic surface defect segmentation. To achieve it, the TGRNet is proposed, including surface defect triplet (including triplet encoder and trip loss), multi-graph reasoning module, and adaptive AP module. In addition, we propose a novel dataset, Surface Defects-4^i, and conduct a large number of comparative experiments and ablation experiments based on the existing few-shot semantic segmentation methods and each part of TGRNet to prove that it can achieve state-of-the-art results in few-shot generic surface defect segmentation.

References

[1] Z. Liu, B. Yang, G. Duan, and J. Tan, "Visual defect inspection of metal part surface via deformable convolution and concatenate feature pyramid neural networks," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9681–9694, Dec. 2020.

[2] Q. Luo *et al.*, "Automated visual defect classification for flat steel surface: A survey," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9329–9349, Dec. 2020.

[3] H. Zhang, X. Jin, Q. M. J. Wu, Y. Wang, Z. He, and Y. Yang, "Automatic visual detection 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.

[4] Y. Zhao, W. Xu, C. Z. Xi, D. Liang, and H. Li, "Automatic and accurate measurement of microhardness profile based on image processing," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 6006009, doi: 10.1109/TIM.2021.3067191.

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

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

[7] Q. Luo, X. Fang, L. Liu, C. Yang, and Y. Sun, "Automated visual defect detection for flat steel surface: A survey," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 3, pp. 626–644, Mar. 2020.

[8] D. Zhang, K. Song, J. Xu, Y. He, M. Niu, and Y. Yan, "MCnet: Multiple context information segmentation network of no-service rail surface defects," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 5004309.

[9] X. Wu, L. Qiu, X. Gu, and Z. Long, "Deep learning-based generic automatic surface defect inspection (ASDI) with pixelwise segmentation," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 5004010.

[10] D. J. Pasadas, H. G. Ramos, B. Feng, P. Baskaran, and A. L. Ribeiro, "Defect classification with SVM and wideband excitation in multilayer aluminum plates," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 1, pp. 241–248, Jan. 2020.

[11] L. Xie, L. Lin, M. Yin, L. Meng, and G. Yin, "A novel surface defect inspection algorithm for magnetic tile," *Appl. Surf. Sci.*, vol. 375, pp. 118–126, Jul. 2016.

[12] M. Nieniewski, "Morphological detection and extraction of rail surface defects," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 9, pp. 6870–6879, Sep. 2020.

[13] M. Niu, K. Song, L. Huang, Q. Wang, Y. Yan, and Q. Meng, "Unsupervised saliency detection of rail surface defects using stereoscopic images," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 2271–2281, Mar. 2021.

[14] D. Zhang, K. Song, Q. Wang, Y. He, X. Wen, and Y. Yan, "Two deep learning networks for rail surface defect inspection of limited samples with line-level label," *IEEE Trans. Ind. Informat.*, early access, Dec. 16, 2020, doi: 10.1109/TII.2020.3045196.

[15] A. Moeini, K. Faez, and H. Moeini, "Unconstrained pose-invariant face recognition by a triplet collaborative dictionary matrix," *Pattern Recognit. Lett.*, vol. 68, pp. 83–89, Dec. 2015.

[16] Y. Zhao, C. Yang, Y. Wang, J. Cai, and Y. Xue, "Face recognition for embedded system based on optimized triplet loss neural network," in *Proc. 3rd Int. Conf. Adv. Electron. Mater., Comput. Softw. Eng. (AEMCSE)*, Apr. 2020, pp. 260–263.

[17] X. Bian, S. N. Lim, and N. Zhou, "Multiscale fully convolutional network with application to industrial inspection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–8.

[18] Z. Yu, X. Wu, and X. Gu, "Fully convolutional networks for surface defect inspection in industrial environment," in *Proc. Int. Conf. Comput. Vis. Syst.*, 2017, pp. 417–426.

[19] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *J. Intell. Manuf.*, vol. 31, no. 3, pp. 759–776, Mar. 2020.

[20] S.-T. Liong, Y. S. Gan, Y.-C. Huang, C.-A. Yuan, and H.-C. Chang, "Automatic defect segmentation on leather with deep learning," 2019, *arXiv:1903.12139*. [Online]. Available: http://arxiv.org/abs/1903.12139

[21] F. Delconte, P. Ngo, I. Debled-Rennesson, B. Kerautret, V. T. Nguyen, and T. Constant, "Tree defect segmentation using geometric features and CNN," in *Reproducible Research in Pattern Recognition*. Cham, Switzerland: Springer, 2021, pp. 80–100.

[22] Y. Xie, F. Zhu, and Y. Fu, "Main-secondary network for defect segmentation of textured surface images," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 3520–3529.

[23] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," 2017, *arXiv:1709.03410*. [Online]. Available: https://arxiv.org/abs/1709.03410

[24] X. Zhang, Y. Wei, Y. Yang, and T. S. Huang, "SG-one: Similarity guidance network for one-shot semantic segmentation," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3855–3865, Sep. 2020.

[25] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, "CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5217–5226.

[26] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Wu, and R. Yao, "Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9587–9595.

[27] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 3, 2020, doi: 10.1109/TPAMI.2020.3013717.

[28] B. Yang, C. Liu, B. Li, J. Jiao, and Q. Ye, "Prototype mixture models for few-shot semantic segmentation," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2020, pp. 763–778.
[29] X. Li, T. Wei, Y. P. Chen, Y.-W. Tai, and C.-K. Tang, "FSS-1000: A 1000-class dataset for few-shot segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2869–2878.
[30] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*. [Online]. Available: https://arxiv.org/abs/1312.6203
[31] L. Zhao *et al.*, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.
[32] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 257–266.
[33] Y. You, T. Chen, Z. Wang, and Y. Shen, "L2-GCN: Layer-wise and learned efficient training of graph convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2124–2132.
[34] X. Li, Y. Yang, Q. Zhao, T. Shen, Z. Lin, and H. Liu, "Spatial pyramid based graph reasoning for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8947–8956.
[35] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 433–442.
[36] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2661–2671.
[37] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: http://arxiv.org/abs/1706.05587
[38] Y. Huang, C. Qiu, Y. Guo, X. Wang, and K. Yuan, "Surface defect saliency of magnetic tile," in *Proc. IEEE 14th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2018, pp. 612–617.
[39] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9584–9592.

Yanqi Bao received the B.S. degrees from the School of Mechanical Engineering and Automation, North University of China, Taiyuan, China, in 2019. He is currently pursuing an M.S. degree from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China.

His current research interests focus on few-shot learning, few-shot semantic segmentation, and multimodal fusion.

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

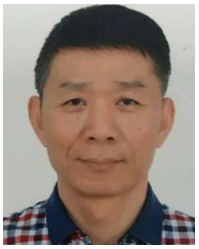
From 2018 to 2019, he was an Academic Visitor with the Department of Computer Science, Loughborough University, Loughborough, U.K. He is currently an Associate Professor with the School of Mechanical Engineering and Automation, Northeastern University. His research interests include vision-based inspection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

Jie Liu received the B.S. degrees from the School of Mechanical Engineering and Automation, North University of China, Taiyuan, China, in 2018. He is currently pursuing an M.S. degree from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China.

His current research interests focus on object recognition and semantic segmentation algorithms under data-scarcity scenarios, such as zero-shot/few-shot learning and few-shot semantic segmentation.

Yanyan Wang received a B.S. degree from the School of Mechanical Engineering, Qingdao University of Technology, Qingdao, China, in 2018. She is currently pursuing an M.S. degree from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China.

Her current research interests include deep learning and intelligent inspection.



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

Since 1982, he has been a teacher at Northeastern University and became a Professor in 1997. From 1993 to 1994, he stayed as a Visiting Scholar at the Tohoku National Industrial Research Institute, Sendai, Japan. His current research interests include intelligent inspection, image processing, and pattern recognition.

Han Yu received the B.S. degree from Shenyang Ligong University, Shenyang, China, in 2015, and an M.S. degree from Northeastern University, Shenyang, China, in 2020.

He is currently a researcher with the State Key Laboratory of Light Alloy Foundry Technology for High-end Equipment, Shenyang. His research interests include computer vision, image processing, and deep learning and their applications in NDT&E.

Xingjie Li received the B.S. degree from the Xi'an University of Technology, Xi'an, China, in 1987.

He is currently the Director with the Laboratory of testing technology, State Key Laboratory of Light Alloy Foundry Technology for High-end Equipment, Shenyang, China. His research interests include nondestructive testing and evaluation.