Cross Position Aggregation Network for Few-Shot Strip Steel Surface Defect Segmentation

Hu Feng, Kechen Song, *Member, IEEE*, Wenqi Cui, Yiming Zhang, and Yunhui Yan

***Abstract*— Strip steel surface defect (S^3D) segmentation is a crucial method to inspect the surface quality of strip steel in the producing-and-manufacturing. However, existing S^3D semantic segmentation methods depend on quite a few labeled defective samples for training, and generalization to novel defect categories that have not yet been trained is challenging. Additionally, some defect categories are incredibly sparse in the industrial production processes. Motivated by the above problems, this article proposed a simple but effective few-shot segmentation method named cross position aggregation network (CPANet), which intends to learn a network that can segment untrained S^3D categories with only a few labeled defective samples. Using a cross-position proxy (CPP) module, our CPANet can effectively aggregate long-range relationships of discrete defects, and support auxiliary (SA) can further improve the feature aggregation capability of CPP. Moreover, CPANet introduces a space-squeeze attention (SSA) module to aggregate multiscale context information of defect features and suppresses disadvantageous interference from background information. In addition, a novel S^3D few-shot semantic segmentation (FSS) dataset FSSD-12 is proposed to evaluate our CPANet. Through extensive comparison experiments and ablation experiments, we explicitly evaluate that our CPANet with the ResNet-50 backbone achieves state-of-the-art performance on dataset FSSD-12. Our dataset and code are available at (https://github.com/VDT-2048/CPANet).**

***Index Terms*— Cross-position aggregation network (CPANet), few-shot learning, few-shot semantic segmentation (FSS), strip steel surface defect (S^3D) segmentation.**

I. Introduction

STRIP steel is an essential raw material in industrial production and manufacturing [1], [2], [3], [4], and the surface quality of strip steel will directly influence its production-grade and work performance. Numerous strip steel surface defects (S^3D), such as inclusion, punching, and scratch, are caused by rolling equipment status fluctuations and other adverse production factors [5]. Traditional S^3D inspection methods depend on manual implementation, which inevitably suffers a hefty workload, low detection efficiency, and unstable detection performance [6]. Therefore, some machine learning methods [7], [8] were proposed to achieve automatic S^3D detection in the early days. Traditional machine learning methods rely on manually constructing defect features. However, it is difficult to accurately represent S^3D features with irregular shapes and significant size variations. Not only do experts additionally design multiple defect template groups when novel defects arise, but they also have to perform complex feature post-processing. Therefore, machine learning-based methods struggle to detect S^3D in the modern production phases. With the emergence of deep learning, numerous defect detection models [9], [10], [11] based on the convolution neural network (CNN) framework have been applied to extract complicated defect features. These methods save the cost of hand-designed defect features and considerably improve detection accuracy. At present, the S^3D detection methods principally consist of image classification [12], object detection [13], and semantic segmentation [14]. Compared with the previous two paradigms, semantic segmentation-based methods proceed with dense classification of each pixel in the defective image [15], which has a powerful ability to predict defect region on a pixel-wise level.

Although existing CNN-based models for defect semantic segmentation have achieved better prediction capabilities, the segmentation performance of these methods drops dramatically when labeled samples are insufficient or the defect categories are not trained. With advances in production technology, the rate of defects in strip steel has been tightly controlled. As a result, it is challenging for researchers to take sufficient defect data. Moreover, the defect classes appear to have long-tailed distributions, which results in some defect classes becoming extremely sparse during production. Unfortunately, traditional CNN-based methods require a sufficient amount of annotated data to optimize their enormous trainable model parameters. In addition, these supervised methods work effectively only for defect classes that participate in the training phase. In other words, traditional segmentation methods typically struggle with generalization ability on novel defect classes with few labeled samples.

To resolve this challenge, the theory of the few-shot semantic segmentation (FSS) is introduced to our work. FSS is a practical application of meta-learning in segmentation. The FSS method aims to train a segmentation model using a few labeled samples, which can quickly apply to a novel defect category with only a few labeled data [16]. As shown in Fig. 1(a), the traditional CNN-based encoder-decoder defect segmentation methods are trained by numerous labeled defective images and tested on the identical defect category.

Manuscript received 13 January 2023; accepted 6 February 2023. Date of publication 20 February 2023; date of current version 1 March 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 51805078, in part by the Fundamental Research Funds for the Central Universities under Grant N2103011, in part by the Central Guidance on Local Science and Technology Development Fund under Grant 2022JH6/100100023, and in part by the 111 Project under Grant B16009. The Associate Editor coordinating the review process was Dr. Hongrui Wang. *(Corresponding authors: Kechen Song; Yunhui Yan.)*

The authors are with the School of Mechanical Engineering and Automation, the National Frontiers Science Center for Industrial Intelligence and Systems Optimization, and the Key Laboratory of Data Analytics and Optimization for Smart Industry, Ministry of Education, Northeastern University, Shenyang, Liaoning 110819, China (e-mail: fenghu@stumail.neu.edu.cn; songkc@me.neu.edu.cn; yanyh@mail.neu.edu.cn).

Digital Object Identifier 10.1109/TIM.2023.3246519

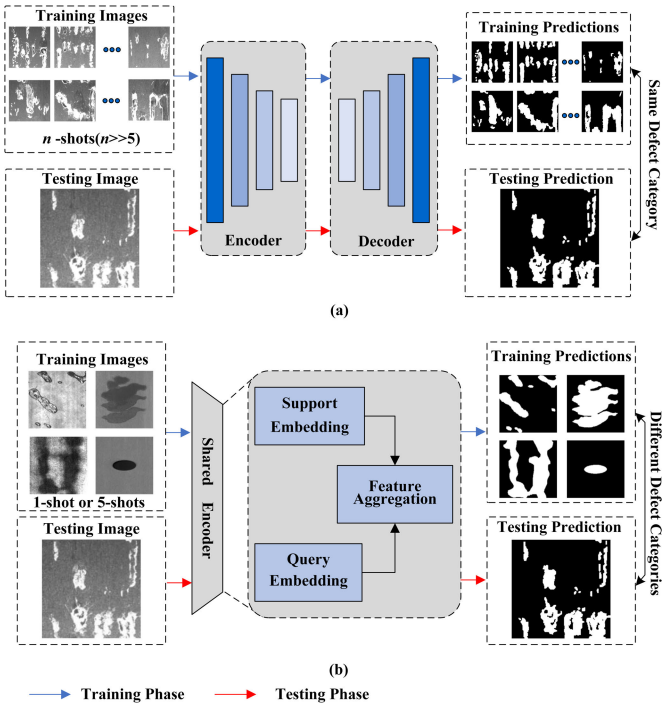

Fig. 1. Comparison of traditional defect segmentation network and few-shot defect segmentation network. (a) Traditional encoder-decoder structure defect segmentation method is trained on sufficient labeled defective images and tested on the identical defect category. (b) Few-shot defect segmentation method is trained on the multiple defect categories with insufficient labeled defective images and tested on novel defect category.

In contrast, as shown in Fig. 1(b), the few-shot defect segmentation methods are trained on known defect classes containing sufficient labeled samples. However, the prediction performance is evaluated using different defect categories with insufficient labeled samples.

However, existing FSS methods struggle to generalize to segment S^3D. General FSS methods mainly apply to the FSS benchmark, such as PASCAL-2012 [17], COCO-2014 [18], and FSS-1000 [19]. As far as we understand, only [20], [21] are proposed to segment surface defects under the FSS paradigm, which are evaluated on the same dataset, Surface Defect-4^i. In general, the pixel distribution of natural objects is continuous over a large region. Therefore, traditional FSS methods, such as CANet [22] and SG-One [23], commonly use a global average prototype to represent a specific semantic category with no difficulty. In contrast, S^3D typically distributes to discrete regions within the identical sample. Global average prototypes inevitably lose some essential local defect information, making it challenging to represent defect features comprehensively. Moreover, S^3D typically exhibits irregular shapes, considerable size variation, intra-class variance, inter-class similarity, low contrast, and ambiguity between normal and defective. In short, existing FSS methods are more likely to segment non-industrial surface defects, and their generalization is insufficient to address the S^3D segmentation task. Motivated by this, a novel few-shot S^3D segmentation method, cross position aggregation network (CPANet), is proposed. To overcome the traditional global average prototype restriction, we design a cross-position proxy (CPP) module to aggregate discrete defects and employ a support auxiliary (SA) module to strengthen the CPP module. In addition, a space-squeeze attention (SSA) module is used to aggregate the multiscale context information of S^3D and suppress the interference of background information. Finally, to address the lack of pixel-wise labeled defective samples in existing works and to evaluate the effectiveness of our method, we build a novel S^3D semantic segmentation dataset, FSSD-12.

The salient contributions of this article can be summarized as follows.

1) To conquer the existing challenge of S^3D inspection, a novel FSS method, CPANet, is proposed. Our CPANet can efficiently segment novel S^3D with insufficient labeled samples.

2) A novel CPP module is used to cross-spatial position to aggregate the long-range correlation among discrete defect areas.

3) A novel SSA module is used to simultaneously aggregate the multiscale foreground features and suppress the disadvantageous interference of background information.

4) We construct a novel S^3D dataset, FSSD-12, which is used to tackle the insufficient pixel-wise labeled defective samples in the existing datasets and evaluate the effectiveness of our method. Our CPANet is superior to other existing FSS methods and shows state-of-the-art results on FSSD-12 under 1-shot and 5-shot settings.

In the remainder of the article, Section II discusses recent related works about S^3D detection, FSS, and attention mechanism. Section III provides the thoroughly detailed structure of CPANet. Section IV describes dataset FSSD-12, evaluation metric, experiment setup, comparison experiments, and ablation experiments. Finally, the conclusion is given in Section V.

II. Related Work

This section reviews recent research results about S^3D detection, FSS, and attention mechanism.

A. S^3D Detection

In S^3D detection, image classification, object detection, and semantic segmentation are the primary computer vision paradigms. Semantic segmentation-based methods with pixel-level accurate prediction results have recently received considerable attention. Nand and Neogi [14] conducted a machine-learning algorithm based on entropy, which used local entropy, background subtraction, and morphology to determine the position of defective pixels. Some works [24], [25] introduce unsupervised-learning methods to solve steel surface defect problems. Neven and Goedemé [26] used the multibranch U-Net network to segment different steel defects, and their severity is estimated. Zheng et al. [27] proposed replacing the traditional convolution layer with deep separable convolution. They employed a multiscale module to extract the contextual information of S^3D to improve the segmentation performance. In addition, some methods based on saliency detection [28], [29] are proposed to predict the saliency of defects. Song et al. [5] constructed the Encoder-Decoder Residual Network (EDRNet) to predict the saliency defects of strip steel. Wang et al. [30] released the first publicly few-shot defect dataset, NEU-DET, to alleviate the disadvantage of insufficient defect samples. Xiao et al. [13] designed

graph embedding and optimal transport to enhance the performance of few-shot classification. Bao et al. [20] constructed a Triplet-Graph reasoning network (TGRNet) based on few-shot segmentation to segment surface defects, including metal and non-metal. They conducted a novel dataset named Surface Defects-4^i, which contains four classes of surface defects of strip steel. Yu et al. [21] proposed a selective prototype network with a matrix decomposition attention mechanism to improve the segmentation performance on the Surface Defects-4^i.

B. Few-Shot Semantic Segmentation

In recent years, FSS has received further attention. After Shaban et al. [16] first proposed the theory of FSS, numerous FSS methods based on metric learning emerged. CANet [22] adopted a dense comparison module to extract features for feature comparison and proposed an iterative optimization module to improve the prediction performance. PGNet [31] leveraged used multiscale graph attention to propagate similarity information of nodes between two images. CRNet [32] applied cross-reference networks to achieve simultaneous prediction of the support set and query set. PFENet [33] conducted a feature enrichment module to aggregate contextual information of different scales to improve the segmentation performance. SAGNN [34] proposed to treat different scale features as the node of the graph neural network REF [35] utilized a rich embedding features method to explore multiple perspectives of support sets and multiscale decoder modules simultaneously to improve the segmentation capability. SCL [36] established a self-guiding mechanism to improve the loss of information due to masked-global average pooling. CWT [37] performed classifier weight transformers that effectively reduce the in-class differences between support and query sets. HSNet [38] applied multilevel feature correlation and more efficient 4-D convolution to extract features from intermediate convolutional layers. DCP [39] designed a divide-and-conquer method that decomposed the target into multiple prototypes, improving the characterization ability of prototypes by aggregating similar properties with the parallel decoder.

C. Attention Mechanism

Attention mechanisms have been widely used in image processing. The attention mechanism is divided into the spatial, channel, and hybrid domains according to the different ways and locations of attention weights. Concretely, spatial attention includes Self-Attention [40], Non-local Attention [41], and Spatial Transformer [42]. Self-Attention establishes global spatial information and has been widely used in visual processing. Non-local attention captures the long-range relationship between any pixel and the current pixel. SENet [43] used an extrusion excitation and attention model, which could dynamically complete the original feature recalibration. SKNet [44] used a set of dynamic convolutions. Moreover, hybrid domain attention is a combination of channel attention and spatial attention. In CBAM [45], maximum global pooling and global average pooling of channel and spital were used to extract more useful information from the model. DANet [46] used self-attention in both the channel and spatial domains to improve segmentation performance through long-range relationships. Motivated by recent attention mechanism advances, non-local attention is used to aggregate cross-position relations among discrete defect regions, and hybrid domain attention is used to aggregate the contextual defect features and suppress detrimental interference from background information.

III. Method

A. Problem Setting

FSS can effectively address the bottleneck of traditional CNN segmentation methods. Following the few-shot episodic paradigm proposed by Shaban et al. [16], we use a meta-learning approach setting (*1*-way *k*-shot). Our strategy aims to construct a model to segment unseen S^3D categories with only one or a few labeled defective images. In contrast to traditional CNN segmentation methods, all S^3D classes are divided into a meta-training set D_{train} and a meta-testing set D_{test}. Note that there is no overlap between the defect categories D_{train} and D_{test}, $D_{\text{train}} \cap D_{\text{test}} = \emptyset$. To mimic the few-shot scenario, both D_{train} and D_{test} contain multiple episodes. Each episode consists of a support set $S(c) = (I_i^S, M(c)_i^S)_{i=1}^k$ and a query set $Q(c) = (I^Q, M(c)^Q)$, where I^* is the defect image and M^* is the corresponding ground truth (GT). k ($k \geq 1$) denotes the shots of support samples, and c is the identical defect class.

B. Architecture Overview

To address the existing challenges in S^3D detection, a novel few-shot segmentation method, a cross-position aggregation network, is proposed to explicitly segment defect regions in defective samples. As shown in Fig. 2, our method consists of a shared-parameter backbone network, a CPP module, a SA module, and an SSA decoder module. Initially, our CPANet uses the backbone network to extract initial feature maps of the support and query images. Then, the CPP module aggregates the long-range relations among discrete defect regions and constructs a CPP of defect features. After that, the SA module receives the proxy and generates an auxiliary prediction to reinforce the CPP module. Simultaneously, the proxy is fed to the SSA decoder module to obtain query prediction. Finally, the total loss is calculated by the above predictions and their corresponding GT.

C. Feature Extractor Encoder

Inspired by predecessors [22], [33], we use the ResNet-50 [47] with dilated convolution as the backbone network to extract defect features. The above works show that the backbone network is pre-trained on ImageNet [48]. ResNet consists of four blocks (*block1-block4*), representing different semantic information levels. Different from [16], the middle-level blocks, *block2* and *block3*, are used to extract the discriminative S^3D features.

At first, the preprocessed support image I^S and query image I^Q are fed to the backbone network simultaneously. Then, the concatenated outputs of the backbone network are

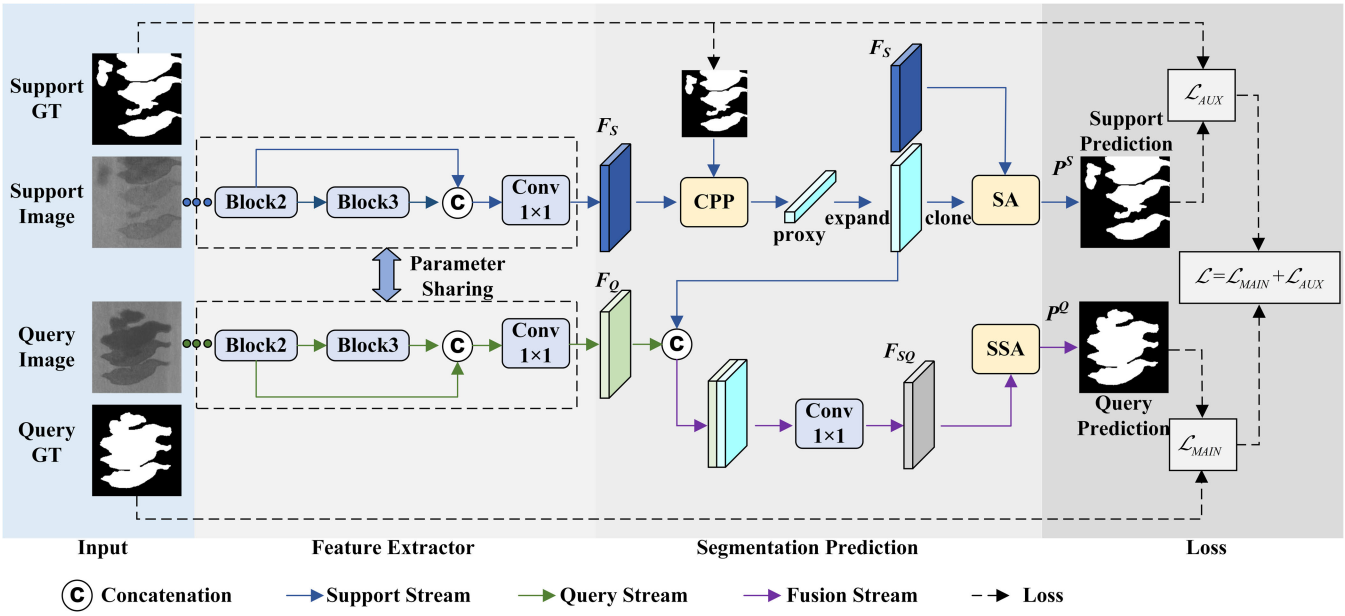

Fig. 2. In our proposed CPANet, first, a parameter-sharing backbone is used to extract the initial feature of defective images. Second, the support feature and its corresponding label are fed into the CPP module to obtain the proxy. Third, the proxy is used to predict the support prediction by SA module and generate the query prediction by SSA module, respectively. After that, using Main Loss and Aux Loss to optimize the trainable parameters of CPANet.

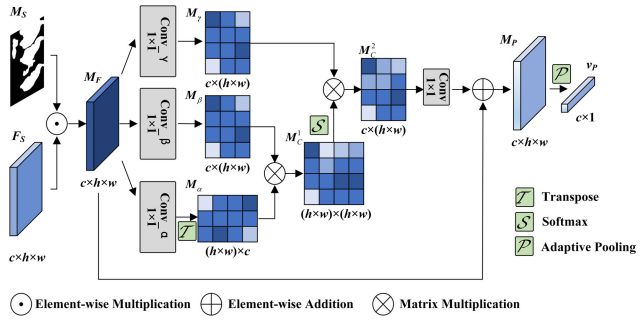

Fig. 3. Detailed illustration of the CPP module.

fused by a 1×1 convolution layer. The sequence of feature extraction is as follows:

$$F_S = \mathcal{F}_{1\times1}\big(\text{Concat}\big(\mathcal{R}_2(I^S), \mathcal{R}_3(I^S)\big)\big) \tag{1}$$

$$F_Q = \mathcal{F}_{1\times1}\big(\text{Concat}\big(\mathcal{R}_2(I^Q), \mathcal{R}_3(I^Q)\big)\big) \tag{2}$$

where $\mathcal{F}_{1\times1}(\cdot)$ denotes the 1×1 convolution operation activated by ReLU. Concat $(\cdot)$ is a concatenation operation. $\mathcal{R}_2(\cdot)$ and $\mathcal{R}_3(\cdot)$ represents the *block2* and *block3* of ResNet. F_S and F_Q are middle-level features of support and query images, respectively.

It should be noted that all of the pre-trained parameters of the backbone network are frozen during the training and testing phases.

D. Cross-Position Proxy

Unlike natural objects, S^{3}D generally distributes in different spatial positions in an identical sample, with irregular shapes and significant size variations. As shown in Fig. 3, a CPP module is proposed to capture the long-range defect information belonging to identical defect categories and generate a proxy for the following operations.

First, we down-sample the support GT M_S by *bilinear interpolation*. This operation can effectively reduce the GPU memory cost. A pixel-wise multiplication is then used to thoroughly filter out the background information of the support feature F_S. The process is as follows:

$$M_F = F_S \otimes \mathcal{I}(M_S) \tag{3}$$

where $\mathcal{I}(\cdot)$ denotes the *bilinear interpolation* operation, and M_F is the masked support foreground feature. $\otimes$ is a pixel-wise multiplication.

Second, we employ a non-local attention mechanism construct [41] to establish cross-position correlations for discrete defect features. Suppose that the size of the masked support feature M_F is $c \times h \times w$, where c is channel size, h is the height of the feature map, and w is the width of the feature map. M_F is fed to three different 1×1 convolutions (*Conv-*α, *Conv-*β, and *Conv-*γ) simultaneously. After the convolution operations, M_F is uniformly resized to $c \times (h \times w)$. The resized results are M_α(from *Conv-*α), M_β(from *Conv-*β), and M_γ(from *Conv-*γ), respectively. M_α is transposed to $M_\alpha^\top \in R^{(h\times w)\times c}$. A matrix multiplication is used to calculate the correlation map $M_C^1 \in R^{(h\times w)\times(h\times w)}$ between $M_\alpha^\top$ and M_β. Then, using a *softmax* operation, the correlation map M_C^1 is normalized. After that, the normalized result is multiplied with M_γ to calculate the correlation map $M_C^2 \in R^{c\times(h\times w)}$. With a 1×1 convolution layer, the correlation map M_C^2 is resized to the original size $c \times h \times w$. Finally, we construct a residual connection to fuse the feature information between the cross-position correlations and M_F. The process of cross-position aggregation is as follows:

$$M_P = M_F \oplus \big(\mathcal{S}\big(M_\alpha^\top \otimes M_\beta\big) \otimes M_\gamma\big) \tag{4}$$

where $\mathcal{S}(\cdot)$ represents the *softmax* layer. $\otimes$ denotes the matrix multiplication and $\oplus$ is the residual element-wise addition.

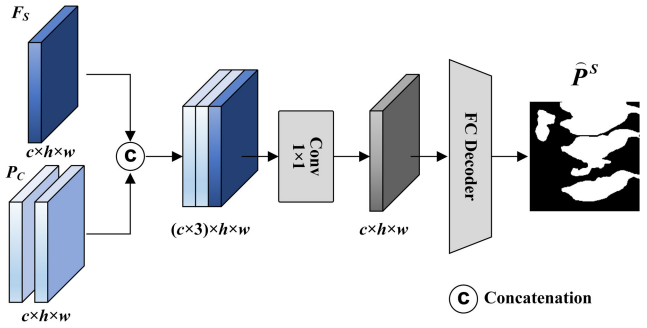

Fig. 4. Detailed illustration of the SA decoder.

$\top$ is the matrix transposition. In addition, $M_P \in R^{c\times h\times w}$ is the output of the cross-position inference.

Finally, an adaptive averaging pooling operation is employed to build the CPP. The process is as follows:

$$v_P = \mathcal{F}_{\text{pool}}(M_P) \tag{5}$$

where $v_P \in R^{c\times 1}$ is the CPP. $\mathcal{F}_{\text{pool}}(\cdot)$ represents adaptive pooling operation.

Algorithm 1 Training and Evaluating CPANet

Input: a training set D_{train} and a testing set D_{test}
Output: Trained parameters of CPANet
for each episode training **do**:
Extract F_S and F_Q with defect feature extractor using Eqns. 1-2;
Mask support feature and project to cross-position attention, calculate cross-position proxy, using Eqns. 3-5;
Get the support prediction $\widehat{P}^S$ with support auxiliary module, using Eqns. 6;
Get the query prediction $\widehat{P}^Q$ with space-squeeze attention module, using Eqns. 7;
Compute the query main loss $\mathcal{L}_{MAIN}$, support aux loss $\mathcal{L}_{AUX}$, and model loss $\mathcal{L}$ using Eqns. 8-10;
Compute the gradient and optimize via SGD;
end
for each episode testing **do**:
Extract F_S and F_Q with defect feature extractor, using Eqns. 1-2;
Mask support feature and project to cross-position attention, calculate cross-position proxy, using Eqns. 3-5;
Get the support prediction $\widehat{P}^Q$ with space-squeeze attention module, using Eqns. 7;
end

E. Support Auxiliary

To provide a high-quality support CPP, the support GT is used as an auxiliary supervision, and the effectiveness of the SA module is explicitly evaluated in ablation experiments.

As shown in Fig. 4, at first, the size of the CPP $v_P \in R^{c\times 1}$ is expanded to match the support feature Fs. Then, the expanded proxy P_C is cloned and concatenated with the support feature Fs. After that, the fusion result is fed to the fully convolutional decoder module to predict SA probability maps $\widehat{P}^S$. The process of SA prediction generation is as follows:

$$\widehat{P}^S = \mathcal{F}_D(\mathcal{F}_{1\times 1}(\text{Concat}([P_C, P_C, F_S]))) \tag{6}$$

where $\mathcal{F}_D(\cdot)$ denotes a fully convolutional decoder. $\mathcal{F}_{1\times 1}(\cdot)$ indicates 1×1 convolution layer activated by *ReLU*.

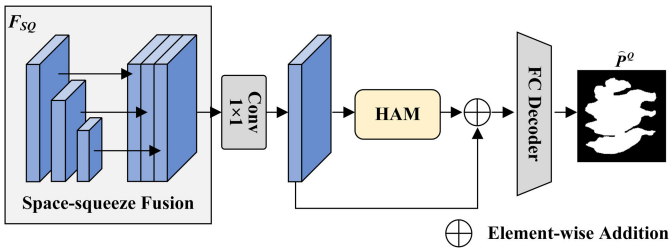

Fig. 5. Detailed illustration of the SSA module, which consists of space-squeeze fusion block, HAM block, and fully convolutional decoder block.

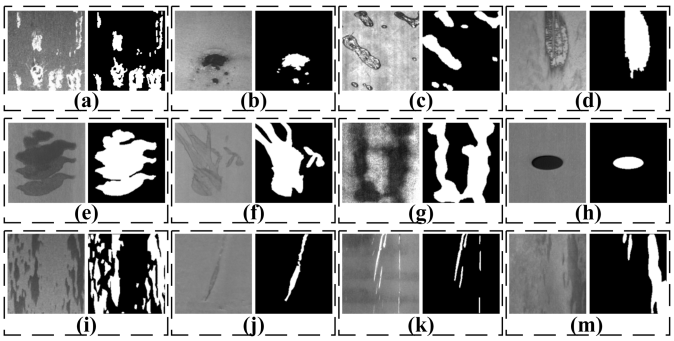

Fig. 6. Instance of S^{3}Dive images (left), corresponding mask GT (right) in FSSD-12, from (a) to (m), which are abrasion-mask, iron-sheet-ash, liquid, oxide-scale, oil-spot, water-spot, patch, punching, red-iron sheet, roll-printing, scratch, and inclusion, respectively.

TABLE I
DETAILS OF DATASET FSSD-12

Fold-i	Defect classes
Fold-0	abrasion-mask, iron-sheet-ash, liquid, oxide-scale
Fold-1	oil-spot, water-spot, patch, punching
Fold-2	red-iron sheet, scratch, roll-printing, inclusion

F. Space-Squeeze Attention

We propose an SSA module to aggregate multiscale foreground features and suppress disadvantageous interference from background information. As shown in Fig. 5, the module mainly consists of a space-squeeze fusion block, a hybrid attention block, and a fully convolutional decoder block. In particular, we select CBAM [45] as the hybrid attention block, which contains spatial and channel attention mechanisms. While this module is straightforward, it can effectively enhance the segmentation performance of our method.

First, the query feature is densely matched with the expanded CPP by a concatenation operation. Then the matched feature $F_{SQ} \in R^{h\times w\times c}$ is fed to the SSA module. To obtain the multiscale feature information, we use two 1×1 convolution layers *stride* is 2×2 to squeeze the size of F_{SQ} twice in a row. The size of the two feature maps are $(c \times h/2 \times w/2)$

TABLE II
CLASS MIOU RESULTS ON THREE FOLDS OF FSSD-12

Backbone	Method	MIoU (1-shot)				FB-IoU (1-shot)	MIoU (5-shot)				FB-IoU (5-shot)
		Fold-0	Fold-1	Fold-2	Mean		Fold-0	Fold-1	Fold-2	Mean	
VGG-16	$PANet_{2019}$[50]	**54.7**	45.7	40.7	47.0	**69.1**	**54.7**	50.7	44.8	50.1	**71.9**
	$PFENet_{2020}$[33]	50.9	**62.3**	43.4	**52.5**	65.9	54.1	**67.3**	46.3	55.9	68.7
	$PMMs_{2020}$[51]	46.3	54.2	42.5	47.7	63.9	46.4	54.5	43.6	48.2	64.2
	Ours	50.8	54.6	**50.7**	52.0	**69.1**	53.3	54.3	**52.4**	**55.4**	69.7
ResNet-50	$CANet_{2019}$[22]	54.4	54.2	48.2	52.3	67.8	56.1	56.0	51.2	54.4	69.2
	$PGNet_{2020}$[31]	57.9	46.4	52.3	52.2	67.5	49.3	54.2	53.9	52.5	70.1
	$PMMs_{2020}$[51]	56.7	52.7	40.9	50.1	66.7	56.9	52.9	41.2	50.4	67.2
	$PFENet_{2020}$[33]	58.9	55.3	51.6	55.3	70.3	59.9	56.0	52.2	56.0	74.0
	SCL_{2021}[36]	57.5	50.2	46.3	51.3	68.4	57.9	50.4	47.4	51.9	68.8
	$HSNet_{2021}$[38]	50.2	53.0	41.8	48.4	67.9	56.0	60.9	47.3	54.7	71.6
	$TGRNet_{2021}$[20]	61.8	62.0	48.3	57.7	73.6	62.4	59.7	47.8	58.5	75.1
	DCP_{2022}[39]	58.3	51.7	43.1	51.0	70.6	55.9	47.0	55.7	52.9	70.9
	Ours	**66.0**	**64.0**	**54.6**	**61.5**	**76.1**	**66.5**	**64.9**	**56.3**	**62.6**	**76.3**

and ($c \times h/4 \times w/4$), respectively. Then, the down-sampled feature maps are resized to $c \times h \times w$ and concatenated with F_{SQ} to aggregate the multiscale context information of defects. Then, we employ a 1×1 convolution layer to reduce the dimension of the concatenated feature. After that, the multiscale feature is fed to the hybrid attention block to represent the foreground feature and suppress interference from the background. Finally, the attention result is fed to the fully convolutional decoder block to obtain the query probability map $\widehat{P}^{Q}$. Concretely, our fully convolutional decoder block consists of two 3×3 convolution layers and a 1×1 convolution layer, all activated by *ReLU*. The process is as follows:

$$\widehat{P}^{Q} = \mathcal{F}_D\big(\text{Atten}\big(\mathcal{F}_S\big(F_{SQ}\big)\big) \oplus \mathcal{F}_S\big(F_{SQ}\big)\big) \tag{7}$$

where $\mathcal{F}_D(\cdot)$ denotes the fully convolutional decoder block. $\mathcal{F}_S(\cdot)$ presents the space-squeeze operation. Atten$(\cdot)$ indicates the hybrid attention mechanism (HAM). In addition, $\oplus$ is the residual element-wise addition.

G. Training Loss

Our model is trained using binary cross entropy (BCE) loss. The total loss function of our method consists of two components: model loss $\mathcal{L}_{\text{MAIN}}$ and auxiliary loss $\mathcal{L}_{\text{AUX}}$. Concretely, $\mathcal{L}_{\text{MAIN}}$ is computed between the final prediction mask of the query image and its corresponding GT. Moreover, the $\mathcal{L}_{\text{AUX}}$ is proposed to improve the ability of the CPP module to aggregate rich discrete defect information. $\mathcal{L}_{\text{AUX}}$ is calculated between the prediction mask of the SA module and its corresponding GT. In short, they are defined as follows:

$$\mathcal{L}_{\text{MAIN}} = -\frac{1}{N}\sum_{i=1}^{N} M_i^{Q} \log \widehat{P}_i^{Q} \tag{8}$$

$$\mathcal{L}_{\text{AUX}} = -\frac{1}{N}\sum_{i=1}^{N} M_i^{S} \log \widehat{P}_i^{S} \tag{9}$$

where M_i^{Q} and M_i^{S} denote the query and support GT. $\widehat{P}_i^{Q}$ and $\widehat{P}_i^{S}$ are the query and support prediction results, respectively. And the total loss is represented as follows:

$$\mathcal{L} = \mathcal{L}_{\text{MAIN}} + k\mathcal{L}_{\text{AUX}} \tag{10}$$

where k represents a hyperparameter, set to 0.4 in our work, according to the superiority result in the ablation experiment.

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

IV. EXPERIMENTS

A. Dataset

We construct a novel few-shot segmentation dataset, FSSD-12, to address the severely insufficient pixel-wise labeled S^3D samples in existing works. As shown in Fig. 6, there are twelve S^3D classes in FSSD-12, including abrasion-mask, iron-sheet ash, liquid, oxide-scale, oil-spot, water-spot, patch, punching, red-iron sheet, roll-printing, scratch, and inclusion.

All raw defective images are integrated from DET GC-10 [10], X-SDD [11], SD-900 [5], and Surface Defects-4^i [20], which are taken in the production and manufacturing stages. Subsequently, we meticulously annotate the overall defective images with pixel-wise labels. In order to

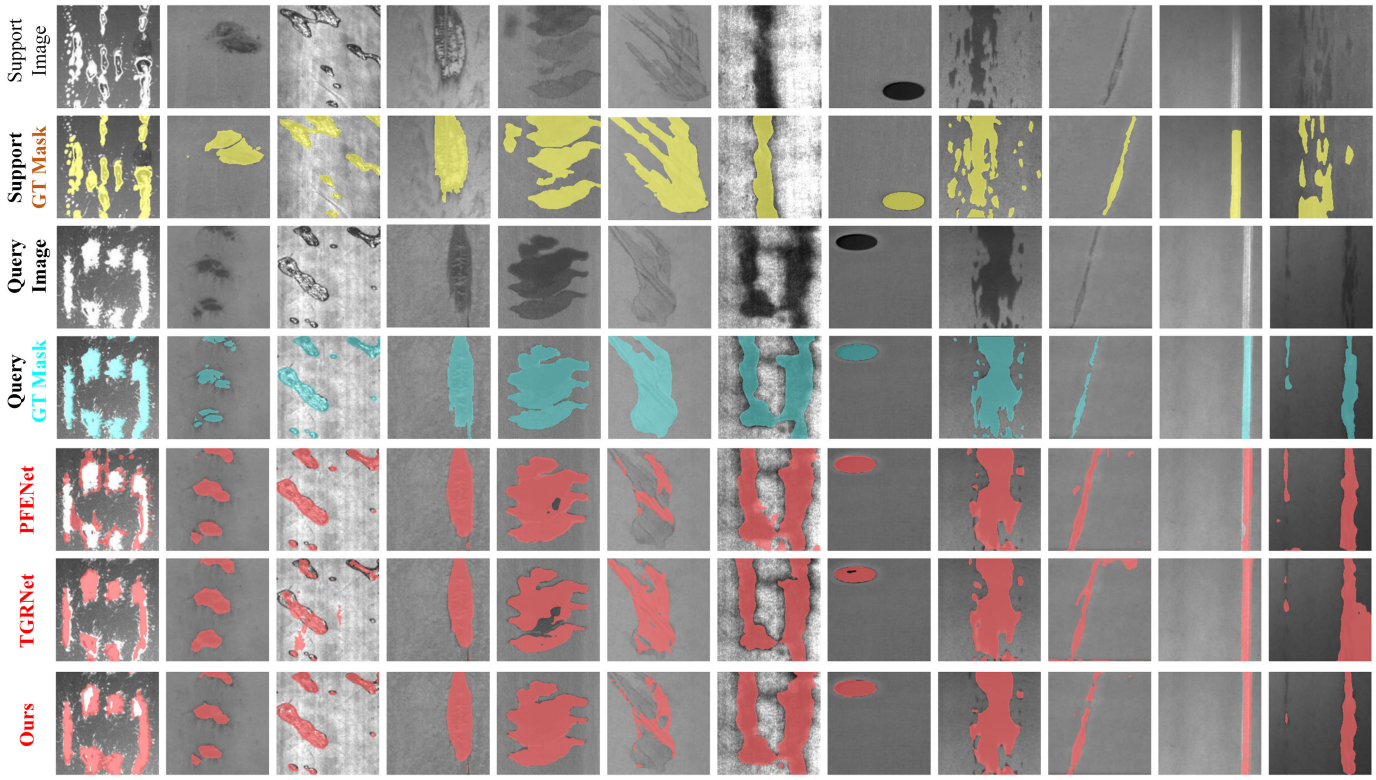

Fig. 7. Visualize the comparative experiments results of CPANet with PFENet and TGRNet. From top to bottom, each row represents support images and corresponding mask GT (yellow), query images and corresponding mask GT masks (blue), and prediction of different methods (red).

apply our method to process strip steel production, we only crop defective samples from the original photographs and do not perform other complicated processing. Besides, all defective images are uniform to 200 × 200 to ensure consistency and decrease the calculating cost. Furthermore, the number of samples in each defect class is limited to 50 to avoid a long-tailed distribution. Following the few-shot dataset setting in [16], all defect classes are randomly divided into three folds for cross-evaluation. Note that the classes of defects do not overlap in the different folds. The details of the fold splitting are given in Table I.

B. Evaluation Metric

Following prior few-shot segmentation works [30], [33], [38], Mean Intersection-over-Union (MIoU) is used as the priority indicator, owing to its objectivity and comprehensiveness. Given a specific defect class C, MIoU is calculated as follows:

$$\text{MIoU} = \frac{1}{C}\sum_{c=1}^{C} \text{IoU}_c \tag{11}$$

where IoU_c represents the IoU of defect class c.

Foreground-and-Background IoU (FBIoU) ignores the class information, and we only presented it for a fair comparison. FBIoU is calculated as follows:

$$\text{FBIoU} = \frac{1}{2}\left(\text{IoU}_f + \text{IoU}_b\right) \tag{12}$$

where IoU_f and IoU_b denote foreground and background IoU in the target fold, respectively.

C. Experimental Setup

We used Resnet-50 [50] and VGG-16 [49], pre-trained on ImageNet [48], as the backbone network. Following [22], our CPANet used the dilated convolution version of the Resnet-50 and the original version of the VGG-16. SGD is used as the optimizer. We set the momentum to 0.9 and the weight decay to 0.001. Our network was trained on FSSD-12 for 200 epochs with a learning rate of 0.025 and a batch size of 2. During training, we froze the overall pre-trained weights of the backbone. Data augmentation was performed with random mountings from $-10°$ to $10°$ and mirroring operation. All defective images were resized to 200 × 200. Both comparison and ablation experiments were performed under the PyTorch 1.70 framework with NVIDIA GeForce RTX 3060 (12 G) GPU and Intel Core-i5 11400F @ 2.60 Ghz CPU, Ubuntu 20.04 system. In addition, our method used an end-to-end training pattern and cross-entropy loss for backpropagation.

TABLE III
COMPARISON OF METHOD PARAMETERS, GPU LOAD, AND GPU TIME

Method	Parameters	GPU Load	GPU Time
PFENet	34.5M	2324MB	36.6ms
TGRNet	31.2M	3533MB	89.0ms
Ours	30.3M	2823MB	40.6ms

D. Comparison Experiment

1) Quantitative Results: Table II illustrates the segmentation performance of our CPANet along with other existing FSS approaches on FSSD-12. Our CPANet, with the backbone ResNet50, outperforms other advanced FSS methods by a considerable margin in all settings. Our method achieves 6.2%p (1-shot) and 6.6%p (5-shot) MIoU improvements over the previous best general few-shot segmentation method PFENet [33]

TABLE IV
ABLATION STUDY OF 1-SHOT AND 5-SHOT MIOU AND FB-IOU

CPP	SA	SSA	MIoU (1-shot)				FB-IoU (1-shot)	MIoU (5-shot)				FB-IoU (5-shot)
			Fold-0	Fold-1	Fold-2	Mean		Fold-0	Fold-1	Fold-2	Mean	
	baseline		57.7	55.1	48.9	53.9	71.4	58.2	55.6	49.3	54.4	72.2
✓			58.3	60.9	49.9	56.4	72.6	59.6	61.5	52.3	57.8	73.0
✓	✓		65.8	63.3	54.4	61.2	75.9	66.2	63.8	55.0	61.7	76.4
		✓	59.3	58.6	50.6	56.2	71.9	61.5	59.7	52.4	57.9	72.5
✓	✓	✓	**66.0**	**64.0**	**54.6**	**61.5**	**76.1**	**66.5**	**64.9**	**56.3**	**62.6**	**76.6**

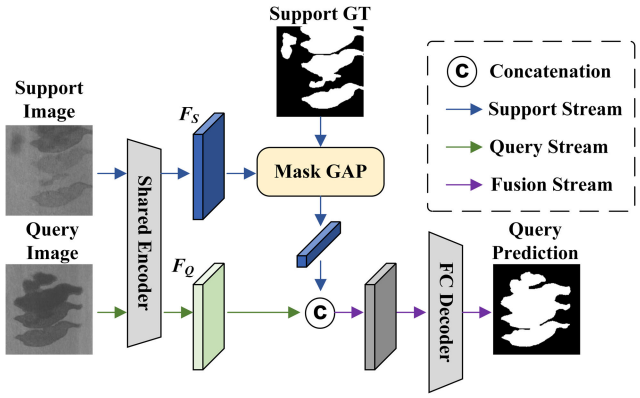

Fig. 8. Detailed illustration of the Baseline.

on FSSD-12. Besides, CPANet performs 3.8%p (1-shot) and 4.1%p (5-shot) of MIoU improvements over surface defect few-shot segmentation method TGRNet [20] on FSSD-12. Moreover, we also compare our CPANet with existing methods for FB-IoU on FSSD-12. Table III also evaluates our CPANet with PFENet and TGRNet through parameters, GPU load, and GPU time.

2) Qualitative Results: As shown in Fig. 7, we visualize the segmentation results to analyze better and evaluate the effectiveness of our CPANet. From top to bottom, each row represents the support image, the corresponding support GT mask (yellow), the query image, the corresponding query GT mask (blue), and the segmentation results (red). Our CPANet can achieve better segmentation performance than TGRNet and PFENet in most defect classes. However, the segmentation performance of our work is inferior to TGRNet in column 6. This phenomenon may be caused by the low contrast and ambiguous boundary between the defective foreground and defect-free background. Besides, our CPANet is more focused on aggregating discriminative foreground information from discrete defects. Since few-shot S^3D defect segmentation has been poorly studied, we hope that our method will catalyze future research to address these issues.

E. Ablation Study

We conduct a series of ablation experiments on FSSD-12 with ResNet-50 backbone network in 1-shot and 5-shot settings. These experiments allow the impact of each component to be evaluated.

1) Baseline Method Versus CPANet: As shown in Fig. 8, a baseline method is established to evaluate the effectiveness of each module proposed in our approach. Instead of using a CPP module, the baseline method uses a mask global average pooling to extract foreground information of defects. Moreover, the SA module is removed, and a fully convolutional decoder replaces the squeeze-space attention decoder module. Compared to the performance of CPANet, the baseline network performs 7.6%d (1-shot) and 8.2%d (5-shot). These results demonstrate the impact of the proposed modules on segmentation performance.

2) Ablation Experiment of CPP Module and SA Module: Our CPP module can better capture the detailed information of defects and aggregate the long-range discrete defect features. As shown in the second row of Table IV, by leveraging the CPP module, our method can achieve 2.5%p (1-shot) and 3.4%p (5-shot) MIoU improvements. In addition, the SA module can effectively improve the feature aggregation ability of CPP. As shown in the third row of Table IV, compared to without SA, our method can achieve 4.8%p (1-shot) and 3.9% (5-shot) MIoU improvements. In general, CPP + SA can better aggregate long-range defect features at different discrete positions.

3) Ablation Experiment of SSA Module: The SSA decoder module can widely aggregate the multiscale context feature of the defect feature and improve the segmentation performance. The fourth row of Table IV shows that the segmentation performance will perform 2.3%d (1-shot) and 3.5%d (5-shot) MIoU without the SSA decoder module. It has been analyzed that the HAM can further aggregate the multiscale defect features and suppress the disadvantageous interference of background information.

4) Ablation Experiment of Backbone Network: As shown in Table V, we conduct ablation experiments about ResNet to evaluate the effect of different backbones on CPANet. Each middle layer (*block2* and *block3*) feature is extracted for a fair comparison. Also, for the best performance, a full CPANet is used. Taking 1-shot as an example, with the ResNet-50 backbone, our CPANet can achieve better segmentation performance. Specifically, it is assumed that the depth of the backbone is insufficient. In that case, the segmentation performance will decrease, such as 4.4%d (ResNet-18) and 1.4%d (ResNet-34), because these backbone networks are too shallow to extract the discriminative defect feature. However, ResNet-101 and ResNet-152 have too many parameters, which makes them prone to overfitting in the presence of insufficient

TABLE V
1-SHOT MIOU AND FB-IOU OF ABLATION STUDY FOR *BACKBONE*

Backbone	MIoU				FB-IoU
	Fold -0	Fold -1	Fold -2	Mean	
Resnet-18	59.8	60.3	51.2	57.1	72.3
Resnet-34	64.5	62.7	53.2	60.1	72.5
Resnet-50	**66.0**	**64.0**	**54.6**	**61.5**	**74.8**
Resnet-101	64.7	56.9	54.4	58.7	74.3
Resnet-152	62.5	53.5	52.1	56.0	73.8

TABLE VI
1-SHOT MIOU AND FB-IOU OF ABLATION STUDY FOR *Hyperparameter k*

k	MIoU				FB-IoU
	Fold-0	Fold-1	Fold-2	Mean	
0	61.8	60.3	50.4	57.5	72.9
0.2	65.2	61.0	53.0	59.7	74.8
0.4	**66.0**	**64.0**	54.6	**61.5**	**76.1**
0.6	63.4	62.9	52.9	59.7	74.8
0.8	62.8	61.4	**55.2**	59.8	74.5
1.0	63.8	62.4	50.6	58.9	74.7

defect samples. In summary, ResNet-50 is used as a backbone network.

5) Ablation Experiment of Hyperparameter k: Our total model loss is a linear combination of the two independent loss functions. Concretely, it consists of $\mathcal{L}_{\text{MAIN}}$ calculated by query prediction and $\mathcal{L}_{\text{AUX}}$ calculated by support prediction. The hyperparameter k controls the ratio of $\mathcal{L}_{\text{MAIN}}$ and $\mathcal{L}_{\text{AUX}}$ in the total loss, which significantly affects the segmentation performance of our method.

As the results in Table VI, hyperparameter k is set from $\{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ to conduct ablation experiments. When $k = 0$, the SA is inoperative. Taking 1-shot as an example, it can be found explicitly that our CPANet achieves the best segmentation performance when $k = 0.4$.

V. CONCLUSION

In this article, we propose a simple but effective FSS method, CPANet, to address the existing challenges in S^3D inspection. Our CPANet consists of a CPP module, an SSA decoder module, and a SA module. In addition, we construct a novel S^3D segmentation dataset, FSSD-12. We performed extensive comparison experiments and ablation experiments on FSSD-12 and our CPANet achieves state-of-the-art results. However, there are some failure cases for our CPANet to segment complex defects, such as low contrast and ambiguous defect boundaries. In the future, we plan to introduce multiple sensors to perceive multidimensional defect information, such as depth information [52] and thermal infrared information [53], [54], to alleviate the defective information loss problem in existing two-dimensional RGB images. We hope that our work may shed some positive enlightenment on existing related challenges for future works.

REFERENCES

[1] X. Wen, J. Shan, Y. He, and K. Song, "Steel surface defect recognition: A survey," *Coatings*, vol. 13, no. 1, p. 17, Dec. 2022.

[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] 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.

[4] Q. Luo, J. Su, C. Yang, W. Gui, O. Silven, and L. Liu, "CAT-EDNet: Cross-attention transformer-based encoder–decoder network for salient defect detection of strip steel surface," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.

[5] G. Song, K. Song, and Y. Yan, "EDRNet: Encoder–decoder residual network for salient object detection of strip steel surface defects," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9709–9719, Dec. 2020.

[6] J. Xing and M. Jia, "A convolutional neural network-based method for workpiece surface defect detection," *Measurement*, vol. 176, May 2021, Art. no. 109185.

[7] S. Ghorai, A. Mukherjee, M. Gangadaran, and P. K. Dutta, "Automatic defect detection on hot-rolled flat steel products," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 3, pp. 612–621, Mar. 2013.

[8] Q. Li and S. Ren, "A real-time visual inspection system for discrete surface defects of rail heads," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 8, pp. 2189–2199, Aug. 2012.

[9] R. Tian and M. Jia, "DCC-CenterNet: A rapid detection method for steel surface defects," *Measurement*, vol. 187, Jan. 2022, Art. no. 110211.

[10] X. Lv, F. Duan, J.-J. Jiang, X. Fu, and L. Gan, "Deep metallic surface defect detection: The new benchmark and detection network," *Sensors*, vol. 20, no. 6, p. 1562, Mar. 2020.

[11] X. Feng, X. Gao, and L. Luo, "X-SDD: A new benchmark for hot rolled steel strip surface defects detection," *Symmetry*, vol. 13, no. 4, p. 706, Apr. 2021.

[12] Q. Luo et al., "Surface defect classification for hot-rolled steel strips by selectively dominant local binary patterns," *IEEE Access*, vol. 7, pp. 23488–23499, 2019.

[13] W. Xiao, K. Song, J. Liu, and Y. Yan, "Graph embedding and optimal transport for few-shot classification of metal surface defect," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[14] G. K. Nand and N. Neogi, "Defect detection of steel surface using entropy segmentation," in *Proc. Annu. IEEE India Conf. (INDICON)*, Dec. 2014, pp. 1–6.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[16] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–13.

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[18] T.-Y. Lin et al., "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 740–755.

[19] 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, p. 2020.

[20] Y. Bao et al., "Triplet-graph reasoning network for few-shot metal generic surface defect segmentation," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[21] R. Yu, B. Guo, and K. Yang, "Selective prototype network for few-shot metal surface defect segmentation," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[22] 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.

[23] 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.

[24] S. Mei, H. Yang, and Z. Yin, "An unsupervised-learning-based approach for automated defect inspection on textured surfaces," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 8, pp. 1266–1277, Jun. 2018.

[25] K. Liu, H. Wang, H. Chen, E. Qu, Y. Tian, and H. Sun, "Steel surface defect detection using a new Haar–Weibull-variance model in unsupervised manner," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 10, pp. 2585–2596, Oct. 2017.

[26] R. Neven and T. Goedemé, "A multi-branch U-Net for steel surface defect type and severity segmentation," *Metals*, vol. 11, no. 6, p. 870, May 2021.

[27] Z. Huang, J. Wu, and F. Xie, "Automatic surface defect segmentation for hot-rolled steel strip using depth-wise separable U-shape network," *Mater. Lett.*, vol. 301, Oct. 2021, Art. no. 130271.

[28] G. Song, K. Song, and Y. Yan, "Saliency detection for strip steel surface defects using multiple constraints and improved texture features," *Opt. Lasers Eng.*, vol. 128, May 2020, Art. no. 106000.

[29] X. Zhou et al., "Dense attention-guided cascaded network for salient object detection of strip steel surface defects," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.

[30] H. Wang, Z. Li, and H. Wang, "Few-shot steel surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[31] 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.

[32] W. Liu, C. Zhang, G. Lin, and F. Liu, "CRNet: Cross-reference networks for few-shot segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4165–4173.

[33] 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.*, vol. 44, no. 2, pp. 1050–1065, Feb. 2022.

[34] G.-S. Xie, J. Liu, H. Xiong, and L. Shao, "Scale-aware graph neural network for few-shot semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5471–5480.

[35] X. Zhang, Y. Wei, Z. Li, C. Yan, and Y. Yang, "Rich embedding features for one-shot semantic segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6484–6493, Nov. 2022.

[36] B. Zhang, J. Xiao, and T. Qin, "Self-guided and cross-guided learning for few-shot segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8312–8321.

[37] Z. Lu, S. He, X. Zhu, L. Zhang, Y.-Z. Song, and T. Xiang, "Simpler is better: Few-shot semantic segmentation with classifier weight transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8721–8730.

[38] J. Min, D. Kang, and M. Cho, "Hypercorrelation squeeze for few-shot segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6921–6932.

[39] C. Lang, B. Tu, G. Cheng, and J. Han, "Beyond the prototype: Divide-and-conquer proxies for few-shot segmentation," 2022, *arXiv:2204.09903*.

[40] A. Vaswani, N. Shazeer, and N. Parmar, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 6000–6010.

[41] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7794–7803.

[42] M. Jaderberg et al., "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015, pp. 2017–2025.

[43] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2011–2023.

[44] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.

[45] S. Woo, J. Park, and J. Lee, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2018, pp. 3–9.

[46] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3141–3149.

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

[48] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[50] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "PANet: Few-shot image semantic segmentation with prototype alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9197–9206.

[51] B. Yang, C. Liu, B. Li, J. Jiao, and Q. Ye, "Prototype mixture models for few-shot semantic segmentation," in *Eur. Conf. Comput. Vis. (ECCV)*, Nov. 2020, pp. 763–778.

[52] J. Wang, K. Song, D. Zhang, M. Niu, and Y. Yan, "Collaborative learning attention network based on RGB image and depth image for surface defect inspection of no-service rail," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 6, pp. 4874–4884, Dec. 2022.

[53] K. Song, J. Wang, Y. Bao, L. Huang, and Y. Yan, "A novel visible-depth-thermal image dataset of salient object detection for robotic visual perception," *IEEE/ASME Trans. Mechatronics*, early access, Oct. 27, 2022, doi: 10.1109/TMECH.2022.3215909.

[54] K. Song, Y. Bao, H. Wang, L. Huang, and Y. Yan, "A potential vision-based measurements technology: Information flow fusion detection method using RGB-thermal infrared images," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

Hu Feng received the B.S. degree from the School of Mechanical Engineering, Yanshan University, Qinhuangdao, China, in 2021. He is currently pursuing the M.S. degree with the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China.

His current research interests include few-shot semantic segmentation, anomaly detection, and metal surface defect detection.

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.

Wenqi Cui received the B.S. degree from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 2022, where he is currently pursuing the Ph.D. degree.

His current research interests include semantic segmentation, salient object detection, and domain generalization.

Yiming Zhang received the B.S. degree from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 2016, where he is currently pursuing the M.S. degree.

His current research interests include multimodal few-shot semantic segmentation.

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 with Northeastern University, and became as a Professor in 1997. From 1993 to 1994, he stayed as a Visiting Scholar at Tohoku National Industrial Research Institute, Sendai, Japan.