

Received 24 November 2022, accepted 2 December 2022, date of publication 12 December 2022,  
date of current version 19 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3228392

## RESEARCH ARTICLE

# Few-Shot PCB Surface Defect Detection Based on Feature Enhancement and Multi-Scale Fusion

HAODONG WANG<sup>1</sup>, JUN XIE<sup>2</sup>, XINYING XU<sup>1</sup>, AND ZIHAO ZHENG<sup>1</sup>

<sup>1</sup>College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan 030024, China

<sup>2</sup>College of Information and Computer, Taiyuan University of Technology, Jinzhong 030600, China

Corresponding author: Jun Xie (xiejun@tyut.edu.cn)

This work was supported in part by the Research Project Supported by Shanxi Scholarship Council of China under Grant 2021-046, in part by the Natural Science Foundation of Shanxi Province under Grant 202103021224056, and in part by the Shanxi Science and Technology Cooperation and Exchange Project under Grant 202104041101030.

**ABSTRACT** In printed circuit board (PCB) defect detection, it is difficult to collect defect samples, and the detection effect is poor due to the lack of data. On the basis of the few-shot learning method, a few-shot PCB defect detection model is proposed. This model introduces feature enhancement module and multi-scale fusion module. The feature enhancement module based on the improved convolution block attention module (CBAM) can highlight the key areas of the received feature maps and suppress the interference of useless information. Aiming at the small size of PCB defects, a multi-scale feature fusion strategy is proposed. It can extract multi-scale feature maps of PCB and fuse them into a high-quality feature map containing different scale information, which can improve the detection precision of the model for small object defects. A large number of experiments on PCB dataset show that our few-shot PCB defect detection model outperforms state-of-the-art methods under different shot settings ( $k=1,2,3,5,10,30$ ). Notably, the proposed model can take into account both detection efficiency and precision, which means it has high practical application value.

**INDEX TERMS** PCB defect detection, few-shot learning, feature enhancement, multi-scale fusion.

## I. INTRODUCTION

As the foundation of the modern information industry, PCB is widely used in various high-end equipment manufacturing fields such as computers, communication electronic equipment and military systems [1], [2]. As an important carrier of electrical connection and support, the quality of PCB has great influence on the stability and safety of various high-end equipment products [3]. Therefore, it is particularly important to study the high-quality detection of PCB surface defects and to eliminate them in time.

In the field of the surface defect detection, from the original manual visual method to the traditional machine learning method, the detection effect has made continuous progress [4], [5]. However, the detection precision still cannot meet the needs of modern industrial development. Deep learning has brought unprecedented advances in the surface defect detection [6], and convolutional neural network (CNN) is

widely used. Various object detection algorithms based on deep learning, such as SSD [7], YOLO [8], Region with CNN features (R-CNN) [9], Fast R-CNN [10], Faster R-CNN [11], etc., have improved the surface defect detection effect to a higher level. In recent years, there are also many excellent models to further improve the precision of object detection algorithm. For example, Zhao et al. [12] proposed an object detection method based on the Larger Scale 'You Only Look Once' Version 4 (LS-YOLOv4) algorithm for detecting the insulators and drop fuses. Zhang et al [13] proposed a novel backbone network named Deep-IRTarget to solve the problems of poor texture information, low resolution and high noise levels in infrared images. This framework superimposes features in frequency domain and spatial domain to construct Dual-domain features. They proposed a Resource Allocation model for Features (RAF) to integrate the features efficiently. This network has achieved remarkable results in infrared target detection and further expanded the application range of object detection algorithms. These algorithms have laid a solid foundation for the development of defect detection.

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval<sup>1</sup>.

These surface defect detection models based on deep learning often require a large amount of labeled data for training. However, the replacement cycle of PCB production line is short, and the number of defect samples is small. Traditional object detection models often have problems such as over-fitting or under-fitting, which cannot achieve better detection accuracy. The limited training data brings great challenges to defect detection, and also limits the promotion of defect detection methods. With the development and progress of industry, the demand for training high-precision models with limited samples is growing. The few-shot problem has received great attention in the field of defect detection. Therefore, in order to solve the problem of lack of PCB defect samples and promote the development of industrial defect detection, a few-shot PCB surface defect detection algorithm is proposed in this article.

Meta-learning has provided a very good solution for few-shot surface defect detection [14], [15]. By designing an excellent meta-learner, the meta-learning method enables the model to get a great detection effect after a few steps of updating on a very small amount of data. This model has a strong generalization ability to quickly adapt to new tasks. In order to solve the problem of few-shot object detection, Kang et al. [16] and Yan et al. [17]. designed a meta-learner to reweight the feature maps of the novel classes, which can improve the prediction precision for few-shot data. The meta-learning Few-Shot Re-Weighting model (FSRW) proposed by Kang et al. [16] is based on YOLOv2 [18]. This model is composed of a meta-feature extraction module, a reweighting module, and a detection module. The feature extractor uses a darknet-19 network to extract the meta-features of the query images. The reweighting module is composed of a lightweight CNN network. The input of reweighting module is the support images and the corresponding location mask, and the output of this module is reweighting vectors which are generated by the support feature maps. The vectors are used to reweight the features of the query images. Finally, the detection module uses the reweighted features of the query images to predict classes and bounding boxes.

Perez-Rua et al. [19] proposed a meta-learning method of center point prediction based on the structure and ideas of CenterNet [20], which can realize incremental learning. However, in the process of PCB defect detection, most methods do not have a strong ability to mine the feature information of key areas, which leads to low detection precision in few-shot defect detection. At the same time, as shown in Fig. 1, for various types of PCB defects, due to the particularity of this products, their defects characteristics are small and their sizes vary, which make it difficult for the model to capture defects and judge the types of defects in the detection process.

Aiming at the above problems, we improve few-shot detection model FSRW of Kang et al. [16] with a few-shot PCB surface defect detection model based on feature enhancement module and multi-scale fusion module (FPFM). The improved model adopts YOLOv3 [21] as the backbone network, and the feature extractor uses the DarkNet-53 network.

Experiments on PCB dataset show that when the number of samples is small, our model has higher detection precision than the original model.

The main contributions of this article are as follows:

- 1) We apply few-shot learning to the PCB surface defect detection tasks, and design a few-shot PCB surface defect detection model. The model can fully mine the information of the samples themselves, which can solve the problem of poor detection precision caused by few PCB defect samples in the actual industrial surface defect detection tasks.
- 2) The model introduces a feature enhancement module. By giving different weights to PCB feature maps, this module can highlight the important defect information in the feature maps, suppress the interference of irrelevant information and enhance the feature discrimination.
- 3) Considering the small scale of PCB defects, a multi-scale feature fusion module is proposed. This module is used to extract PCB features of multiple scales and fuse them into a feature map containing information of different scales, which improves the recognition ability of small surface defect features.

This article is organized as follows. In the Section 2, we introduced some existing research results of PCB defect detection and few-shot problems. Then, we propose our own few-shot model in the Section 3, and analyze the model specifically. We make a lot of experiments to verify the effectiveness of the model in the Section 4. In the Section 5, we make a summary for our work.

## II. RELATED WORK

In this section, we will introduce the related work about PCB surface detection and few-shot problems.

### A. PCB SURFACE DEFECT DETECTION

PCB surface defect detection has always been a challenging task, and the detection method based on deep learning has great development potential. In view of the different sizes of the defective PCB solder paste, Park et al. [23] improved the traditional convolution neural network. They proposed a double-layer defect detection point network which could detect defects at both the micro and macro semantic levels. Wu et al. [24] proposed a method to identify solder joint defects using Mask R-CNN, which could simultaneously realize the classification, location and segmentation of solder joint defects. Aiming at the problem of high false detection rate and low efficiency of PCB defect detection, Ding et al. [25] proposed a tiny defect detection network (TDD-Net) and achieved good detection results on the public PCB datasets. Then, in order to further improve the efficiency and precision of PCB detection, Adibhatla et al. [26] used YOLOv5 large to detect the defects in PCB, which optimized the detection manpower and time. Their work laid a foundation for PCB defect detection, and also promoted the follow-up research.

However, these models require a great number of samples to train. In industry, the number of PCB defect samples is insufficient. Therefore, it is very important to study the few-shot PCB surface defect detection.

### B. FEW-SHOT LEARNING

The goal of few-shot learning is to train the network with a very small number of samples to obtain good performance [27], this method is conducive to solving the problem of the few PCB defect samples. In recent years, few-shot learning methods based on meta-learning have achieved remarkable results. Its focus is that the network can quickly learn when the label data are limited and can be generalized to other new tasks. Specifically, 1) based on fine-tuning, it aims to use a small number of samples to fine-tune these initialization parameters to achieve better results. Finn et al. [28] proposed Model-agnostic meta-learning (MAML) based on the optimizer. He used a brand-new weight optimization method to update the trained initial weights with a small amount of data, which could fit the new data features quickly. 2) Based on a recurrent neural network, it uses external memory to accumulate prior knowledge and then uses it in new tasks to complete classification. Santoro et al. [29] proposed a one-shot learning with memory-augmented neural networks (MANN) which skillfully applied the neural Turing machine to the few-shot learning task. He designed an external memory module to save the information of the feature maps and combined the meta-learning idea to optimize the neural Turing machine to achieve few-shot classification and regression. 3) Based on the metric, it means learning an embedding function which can map the input images into a new space. The images can be classified by similarity measurement. Li et al. [30] proposed an adaptive edge loss function for few-shot learning. He used the semantic information of the data to describe the distance of tasks or categories, so as to achieve the purpose of optimizing the boundary distance. In addition, the graph neural network has a significant effect in few-shot learning methods. Zhang et al. [31] proposed a feature distribution transformation to solve the problem of feature distribution mismatch in graph based few-shot learning. By calculating the optimal class allocation matrix, the classification precision has been further improved. This method further promotes the development of few-shot learning. However, most of these work use few-shot learning to complete classification tasks. Our work will focus on the application of few-shot learning in PCB defect detection.

### C. FEW-SHOT OBJECT DETECTION

PCB surface defect detection includes defect classification and defect object detection and we focus on the latter in this article. Therefore, learning about few-shot object detection is conducive to our research. A Low-Shot Transfer Detector (LSTD) [32] and Representative-Based Metric Learning (RepMet) [33] adopt a general migration framework to adapt to few-shot scenes through pre-training detection models. Meta R-CNN [17] and Few-Shot Object Detection and

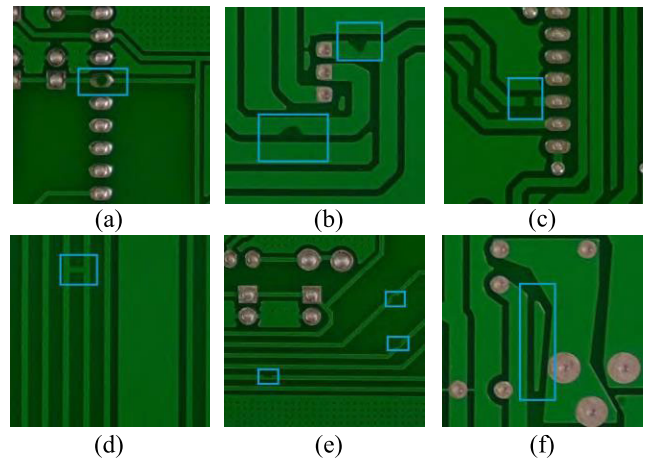


FIGURE 1. Various defect types of PCB: (a) Missing hole, (b) Mouse bite, (c) Open circuit, (d) Short, (e) Spur, (f) Spurious copper.

Viewpoint Estimation (FSDet) [38] add basic detectors to perform detection processes similar to Faster R-CNN to solve the few-shot problems. Kang et al. [16] learned the base class meta-features based on the meta-learning method and used the reweighting method to reweight the features to adapt to the novel classes. However, the above methods have not paid attention to the scale problem of samples. Therefore, Wu et al. [40] proposed multi-scale sample thinning operation to solve the scale variance problem in the model and emphasizes the necessity of dealing with the problem of scale change. However, these models lack of focus on the key areas of the images, which leads to their inability to effectively learn the important features of the images. These problems prevent models from achieving good effect when detecting the few-shot PCB surface defects. Therefore, according to the characteristics of PCB surface defects and the existing few-shot object detection model problems, we design a few-shot detection model for PCB surface defects.

### III. METHODOLOGY

In our work, the defect detection method based on meta-learning is adopted. In this method, we divide the dataset into the base classes  $C_{base}$  and the novel classes  $C_{novel}$ . The base classes contain enough training samples and annotation information, while the number of samples in the novel classes is small. When organizing data, we generally follow the *Episode* data organization method proposed by Vinyals [34]. It means that the data entered into the network each time is called a task  $T_i$  and each task consists of a query set  $Q = \{(I^q, L)\}$  and a support set  $S = \{(I_1^S, M_{I_1^S}), (I_2^S, M_{I_2^S}), \dots, (I_N^S, M_{I_N^S})\}$ . Therefore, we can get the expression for  $T_i$ :

$$T_i = Q_i \cup S_i = (I_i^q, L_i) \cup (I_{I_1^S}^S, M_{I_1^S}^S), (I_{I_2^S}^S, M_{I_2^S}^S), \dots, (I_{I_N^S}^S, M_{I_N^S}^S) \quad (1)$$

where  $I_i^q$  is the query picture in the detection task of time  $i$ , and  $L_i$  is its corresponding label.  $I_{N_i}^S$  and  $M_{I_{N_i}^S}$  is the support picture and corresponding mask in the task of time  $i$ .

The training follows the two-phase training strategy proposed by FSRW [16]. The first phase is the *base training* phase. In this phase, we use the base classes data with sufficient information. Let the network learn to use the information of the support to help the query predict. The second phase is *few-shot fine-tuning*. Following the  $k$ -shot setting, each class in the novel classes contains only  $k$  annotations to fine-tune the model.

### A. OVERVIEW OF FPFM

In this article, the structure of FPFM, a few-shot PCB surface defect detection model based on feature enhancement and multi-scale fusion, is shown in Fig. 2.

Formally, the query image  $I$  is input into the meta-feature extraction module  $D$  to obtain meta-features  $F \in \mathbb{R}^{w \times h \times c}$  with  $c$  channels, which can be expressed by:  $F = D(I)$ . The produced meta features  $F$  are enhanced by the feature enhancement module to get enhanced query features  $F_c$ .

At the same time, the support images and label information are input into the multi-scale feature fusion module to obtain a high-quality feature map  $F_m$  containing different scales. The feature reweighting module is composed of a lightweight CNN network, which shapes the high-quality feature maps  $F_m$  into reweighted feature vectors  $\omega_i \in \mathbb{R}^c$  according to classes. The model realizes the fusion of query features  $F_c$  for novel class  $i$  and support features vectors  $\omega_i$  by:

$$F_i = F_c \otimes \omega_i, \quad i = 1, 2, \dots, N, \quad (2)$$

where  $\otimes$  denotes channel-wise multiplication.

This model inputs the reweighted feature map  $F_i$  into the detection module to predict the confidence  $o$  of the object classes in the image, the position information  $(x, y, h, w)$  of the object prediction frames, and the classification score  $c$  of the object classes.

### B. FEATURE ENHANCEMENT MODULE

In the task of few-shot defect detection, the number of the labeled novel classes is small, the diversity of corresponding categories is poor, and the detection precision of the model is low. Therefore, it is very important to fully mine the information of the samples themselves. In order to fully highlight the important information in the samples and suppress the interference of useless information, a feature enhancement module is introduced in this article. This module is based on the improved convolution block attention module.

#### 1) BASIC STRUCTURE OF CBAM

The basic structure of the convolution block attention module (CBAM) [22] is shown in Fig. 3. The channel attention network filters the channels of input features, while the spatial attention network focuses on the prominent areas in the feature maps.

Specifically, the input of the CBAM module is the meta-features  $F \in \mathbb{R}^{w \times h \times c}$  extracted by the feature extractor. We use CBAM to derive a one-dimensional channel attention map  $M_c \in \mathbb{R}^{c \times 1 \times 1}$  and a two-dimensional spatial attention map  $M_s \in \mathbb{R}^{1 \times h \times w}$  in turn. The overall process can be summarized by:

$$\begin{aligned} F' &= M_c(F) \otimes F \\ F_c &= M_s(F') \otimes F', \end{aligned} \quad (3)$$

where  $\otimes$  denotes channel-wise multiplication. During multiplication, attention values are propagated accordingly.

#### 2) EL-CBAM

On the basis of the existing CBAM, this article improves the channel attention module and the spatial attention module to obtain Efficient and Lightweight CBAM (EL-CBAM), whose specific structure is shown in Fig. 4.

Specific improvement strategies are described from two aspects: channel attention and spatial attention.

##### a: CHANNEL ATTENTION

In the traditional channel attention module, two fully connected layers are used to distribute the attention weights. However, this method will produce many redundant calculations and cause negative effects. Therefore, this article uses the idea of document [35] for reference and replaces the fully connected layers with a 1D-convolution to reduce the number of parameters and achieve better results. For the input features  $F$ , max pooling and average pooling are performed on the basis of channels to obtain two different spatial context descriptors  $F_{max}^c$  and  $F_{avg}^c$ . The information of  $k$  channels in the channel's neighborhood is aggregated by the 1D-convolution which replaces the original fully connected layers. The convolution kernel length of this 1D-convolution is  $k$ . The output features are subjected to element-wise weighting operation and sigmoid activation operation to generate the final  $M_c \in \mathbb{R}^{c \times 1 \times 1}$ . Finally,  $M_c$  and the meta-feature map  $F$  are multiplied element-wise to generate the intermediate feature  $F'$ . Therefore, the improved channel attention calculation formula is expressed as follows:

$$\begin{aligned} M_c(F) &= \sigma(f_{1D}^{k \times k}(AvgPool(F)) + f_{1D}^{k \times k}(MaxPool(F))) \\ &= \sigma(f_{1D}^{k \times k}(F_{avg}^c) + f_{1D}^{k \times k}(F_{max}^c)) \end{aligned} \quad (4)$$

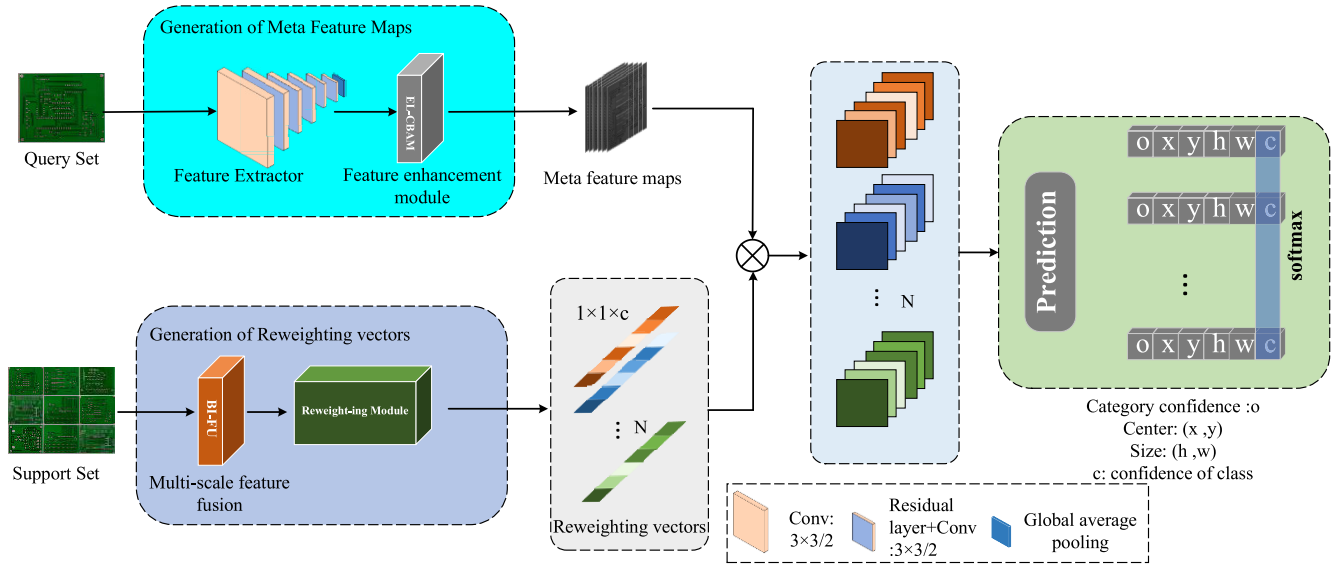
where  $\sigma$  denotes the sigmoid function, and  $f_{1D}^{k \times k}$  represents a one-dimensional convolution operation with the filter size of  $k \times k$ . Where the size of  $k$  is provided by the equation in [35]:

$$k = \left\lfloor \frac{\log_2 c}{\gamma} + \frac{b}{\gamma} \right\rfloor \quad (5)$$

where  $c$  denotes the number of feature map channels.  $\gamma$  and  $b$  are hyperparameters, which are taken as 2 and 1 in this article.

##### b: SPATIAL ATTENTION

In traditional spatial attention, the traditional convolution of the  $7 \times 7$  receptive field is used to aggregate spatial features,



**FIGURE 2.** The model mainly consists of five parts: feature extraction module, feature enhancement module, multi-scale feature fusion module, feature weighting module, and detection module. The backbone of the detection model is chosen as YOLOv3.

which has a large number of parameters and also omits a lot of information between channels. Therefore, in this article, the traditional convolution is replaced by the depth-wise separable convolution. This convolution method has fewer parameters under the same receptive field size. It can realize information exchange between channels, which can help the attention mechanism to notice rich and important information. We do max pooling and average pooling on its  $c$ -dimensional channel of the intermediate feature  $F'$  with its spatial position as the unit to obtain two maps  $F'^S_{max}$  and  $F'^S_{avg}$ , and then splice the two maps. The dimension of the spliced tensors is reduced to a channel of  $w \times h \times 1$  by convolution with the depth-wise separable convolution. After the sigmoid function,  $M_S \in \mathbb{R}^{1 \times h \times w}$  is generated. Finally,  $M_S$  and the meta-feature map  $F'$  are multiplied element-wise to generate the final feature map  $F_c$ . The improved spatial attention calculation formula is expressed as follows:

$$\begin{aligned}
 M_S(F') &= \sigma(f_{depth}^{5 \times 5}([AvgPool(F'); MaxPool(F')])) \\
 &= \sigma(f_{depth}^{5 \times 5}([F'^S_{avg}; F'^S_{max}])) \quad (6)
 \end{aligned}$$

where  $\sigma$  denotes the sigmoid function, and  $f_{depth}^{5 \times 5}$  represents depth-wise separable convolution operation with the filter size of  $5 \times 5$ .

### C. MULTI-SCALE FUSION MODULE

The deep convolution network calculates the feature hierarchy layer by layer and generates feature maps with different spatial resolutions within the network. The high-resolution feature map has weak semantics and strong structure, while the low-resolution feature map has strong semantics. PCB defects have the characteristics of small size. Therefore, the bi-directional feature pyramid network (BiFPN) [37] is used,

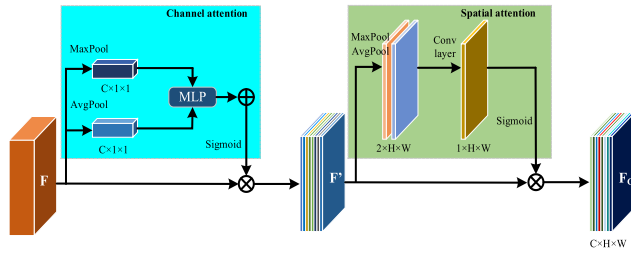
and we proposed the BiFPN and fusion strategy (BI-FU) to solve the problem of small objects. The module structure is shown in Fig. 5.

#### 1) BI

In order to extract the multi-scale features in the support more effectively, the algorithm uses VGG16 [36] and BiFPN layer [37] as the basic blocks of the multi-scale feature extraction part. VGG16 consists of 13 convolutional layers and 5 pooling layers to form 5 Blocks (Block1-Block5), and the size of the feature map in each Block becomes 1/2 of the input size. Compared with ordinary FPN, BiFPN integrates bidirectional cross-scale connections and fast normalization fusion. In the network operation, we put the  $N$  images of the support into the multi-scale feature extraction part. This part can generate  $5N$  feature maps from the images. Finally, a plurality of feature maps with different scales are input into the fusion module FU.

#### 2) FU

In the feature fusion part, the network adaptively learns a feature compression vector  $(a_1, a_2, a_3, a_4, a_5)$ . Specifically, the elements of each feature compression vector are obtained by convolving the corresponding feature map. The corresponding feature map is compressed into a tensor of size  $1 \times 1$  to obtain  $a_k, k=1,2,3,4,5$ . The BI part generates 5 feature maps from every image. We reweight the 5 feature maps through the feature compression vector, and then through element wise add, the FU part fuses those reweighted feature maps into one feature map. Therefore, the final feature map contains different scale information. This feature map can integrate low-resolution and strong semantic features with high-resolution and strong structural features perfectly, so that the network



**FIGURE 3.** The convolution block attention module consists of two parts: Channel Attention Module (CAM) and Spatial Attention Module (SAM), which carry out channel and space attention respectively.

has a better ability to extract and detect the defect features of small objects in PCB.

### D. DETECTION MODULE

The detection module consists of convolution layers and fully connected layers and is used to locate and predict the confidence for the reweighted query picture features. The classification score of the novel classes is given by this module, and the classification score is corrected by the *softmax* function. The classification score of the class  $i$  object class is  $c_i$ . The actual classification score after correction is  $\hat{c}_i$ :

$$\hat{c}_i = \frac{e^{c_i}}{\sum_{j=1}^N e^{c_j}} \quad (7)$$

The loss function of the object category is as follows:

$$\mathcal{L}_c = - \sum_{i=1}^N \mathbb{I}(\cdot, i) \log(\hat{c}_i) \quad (8)$$

where  $\mathbb{I}(\cdot, i)$  is an indicator function for whether current anchor box really belongs to class  $i$  or not.

The loss function of bounding box regression is  $\mathcal{L}_{bbx}$ , and the loss function of objectiveness is  $\mathcal{L}_{obj}$ . These two loss functions are similar to the loss function defined by YOLOv3.

$$\mathcal{L}_{bbx} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{i,j}^{obj} (2 - w_i \times h_i) [(x_i - \hat{x}_i) + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2] \quad (9)$$

$$\mathcal{L}_{obj} = \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{i,j}^{noobj} (c_i - \hat{c}_i)^2 + \lambda_{obj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{i,j}^{obj} (c_i - \hat{c}_i) \quad (10)$$

where  $S^2$  denotes the feature map areas, and  $B$  represents the number of bounding boxes.

Thus, the overall detection loss function is  $\mathcal{L}_{det} = \mathcal{L}_c + \mathcal{L}_{bbx} + \mathcal{L}_{obj}$ .

### E. OTHER MODULES

#### 1) FEATURE EXTRACTOR

We use this module to extract the feature information in the query images. Its input is the query images, and its output is the feature maps  $F$  of the query images. We adopt DarkNet-53 network to replace the original model DarkNet-19 network. Compared with DarkNet-19, DarkNet-53 introduces residual structures and uses conv2d to replace maxpooling2d, which increase the network depth and help to extract deeper features.

#### 2) REWEIGHTING MODULE

The module is constructed by light-weight CNN. In our work, the module receives the feature maps  $F_m$  obtained from BI-FU module. We use the light-weight CNN to shape the feature maps  $F_m$  into vectors of size  $1 \times 1 \times c$ . Their feature is embedded in the vector set of specific classes. These vectors are used to reweight the query feature maps  $F_c$ .

### IV. EXPERIMENTS

In this section, we will evaluate the performance of the FPFM model in few-shot PCB defect detection through comprehensive experiments. The model is compared with several most advanced methods [16], [17], [38], [39], [40], [41]. The results are given and analyzed below to show that our model can detect few-shot PCB defects more accurately.

The experiments in this article are based on the 64-bit operating system Ubuntu 18.04 and Python 3.6 under the PyTorch deep learning framework. The CPU is an Intel Core i7-6850K. The reference frequency is 3.60GHz, the GPU is NVIDIA GeForce GTX 1080 Ti, and the memory is 11GB. CUDA 10.1 is used to accelerate the training.

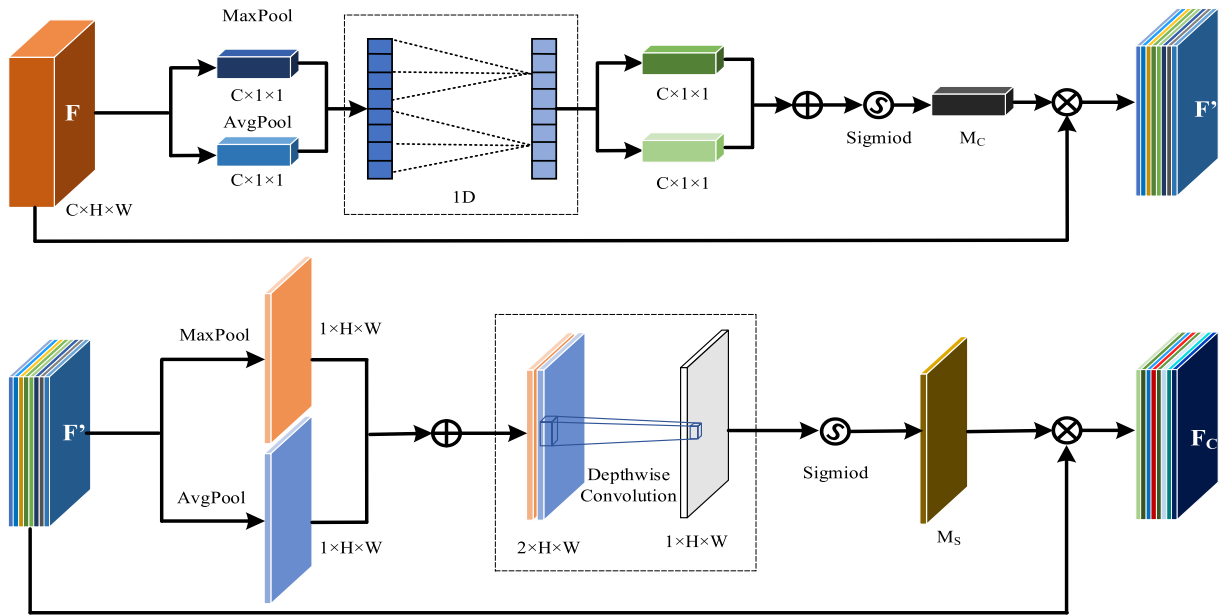
#### A. DATASETS AND TRAINING SETTINGS

##### 1) DATASETS INFORMATION

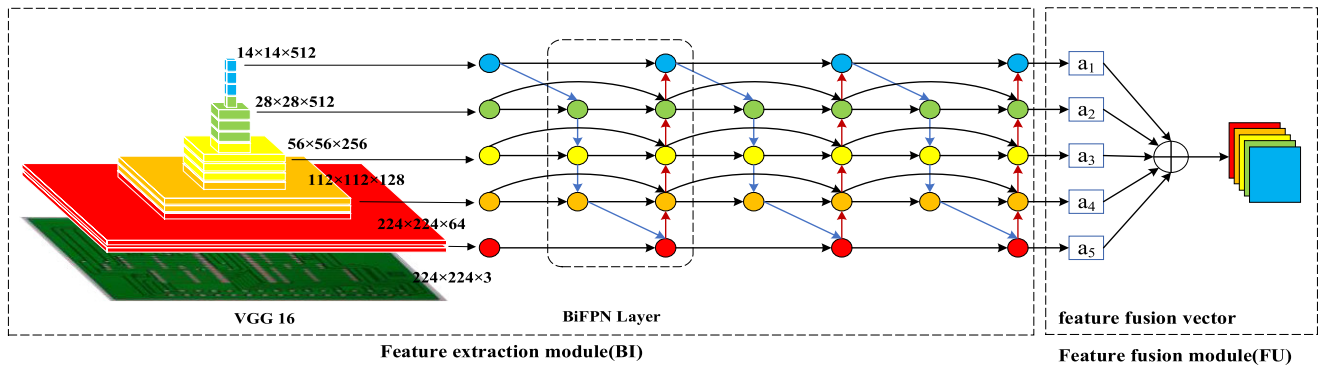
Our dataset consists of two parts, the base classes and the novel classes. In this article, the Few-Shot Object Detection Dataset (FSOD) [42] is used as our base classes. The FSOD is a professional few-shot object detection dataset created by Tencent in 2020. It is rich in category diversity, covering 1,000 categories, with a total of 66,502 images and 182,000 annotated boxes. We use FSOD dataset as the base classes, which is conducive to better meta-learning for the model, thereby improving the detection effect. The novel classes are the PCB defect image dataset from the Intelligent Robot Open Laboratory of Peking University. The dataset consists of 693 PCB defect images and corresponding annotation files in total, including 6 defect types, and the types and quantity are shown in the Table 1.

##### 2) TRAINING SETUP

The model in this article adopts a two-phase training strategy. The first phase is the *base training* phase. In this phase, the support images and query images are obtained from the base classes which have enough labeled information. We randomly



**FIGURE 4.** Schematic diagram of the proposed EL-CBAM structure. The feature enhancement module constructed by EL-CBAM is embedded in the last layer of the meta-feature extractor, and it receives the feature maps  $F \in R^{w \times h \times c}$  of the query images. This module learns the relationship between the foreground and background in the query samples, assigns different weights inside the feature maps to highlight the key feature information in the samples, which can enhance the feature expression ability of the query samples.



**FIGURE 5.** The multi-scale fusion module consists of a multi-scale feature extraction module BI and an adaptive feature fusion module FU. Blue line means the down sampling, red line means the up sampling, and the black line represents the scale invariant convolution operation in the BiFPN part.

sample images from the dataset to form the query set and the support set. A query set has only one image, and the support set has  $N$  images per class. Every image has only one corresponding class's object, and the other object pixels in the image are set to 0. In this process, we use the momentum parameter of 0.9 and the parameter attenuation of 0.0001 to run the SGD optimizer, and the batch size is set to 8. The model uses FSOD as the base classes for meta-learning, and adopts end-to-end training methods. we train the model for 14k, 12k, 10k iterations at the learning rate of  $5e-3$ ,  $5e-4$  and  $5e-5$  respectively. In *few-shot fine-tuning* phase, we use the PCB defect dataset as the novel classes for fine-tuning, and we get the data in the same ways as first stage. The difference is that each class in the novel classes only has  $k$  annotations

( $k$ -shot) for training. In this phase, we train the model for 8k, 6k, 4k iterations at the learning rate of  $5e-3$ ,  $5e-4$  and  $5e-5$  respectively. In order to balance the difference in the number of samples at this stage, only  $k$  annotations are also selected from each base class for fine-tuning. In this article, experiments are carried out under the conditions of  $k = 1, 2, 3, 5, 10, 30$  respectively.

### B. COMPARISON OF TESTING RESULTS

In order to verify that the FPFM model can detect PCB surface defects with higher detection precision under the condition of fewer training samples, the FPFM model and several excellent few-shot object detection models are analyzed under different shot settings. The experimental results

are shown in the following Table 2 and Table 3. For fair comparison, the training strategies and settings of the comparison models are the same as our model.

### 1) COMPARISON OF FEW-SHOT RESULTS

In the above table, AP represents the average precision obtained by testing under different settings of IoU (from 0.5 to 0.95, step size 0.05).  $AP_{50}$  and  $AP_{75}$  refer to the precision of IoU of 0.5 and 0.75.  $AP_L$  means the objects' average precision whose bounding box area is larger than  $96 \times 96$ ,  $AP_S$  represents the average precision in which the bounding box area is less than  $32 \times 32$ , and  $AP_M$  refers to the average precision whose bounding box area is between the above.

We compare our FPFM model with the recent state-of-the-art methods. In order to evaluate the performance of the model more comprehensively, we also report the time required for the model tuning process. It can be seen from Table 2 that our method FPFM outperforms recent state-of-the-art (SOTA) methods in most indicators. For AP, compared with the SOTA methods, our FPFM has achieved about 2.57% and 3.39% performance improvement under 10-shot and 30-shot respectively. When  $k = 10$  and  $k = 30$ ,  $AP_{50}$  reached 69.52% and 78.86%, respectively, 7.33% and 5.45% higher than the SOTA methods. In addition, due to the addition of the BI-FU multi-scale feature fusion module, the small object defects precision  $AP_S$  which is difficult to be optimized is improved by 3.36% in 10-shot and 4.89% in 30-shot, which shows that this module has a great role in the detection of PCB small object defects.

Besides, we also compare the tuning time of our FPFM model and other few-shot object detection models. When  $k = 10$ , the time required for the proposed model is not the shortest, but it is very close to SOTA time. When  $k = 30$ , the model tuning time can reach the level of the SOTA methods. We analyze the following two main reasons: First, the feature enhancement module improves the adaptation speed of the model, so that the model can achieve better detection results in fewer iterations. Second, the addition of the feature enhancement module and the BI-FU module does not increase the computational complexity of the model too much. The feature enhancement module is built on the basis of the CBAM module. This module has the characteristics of high lightweight and strong versatility, and it itself brings less calculation increment to the model. In addition, we further improve the CBAM module. The inner fully connected layers are replaced by 1D-convolution, and the traditional convolution is replaced by depth-wise separable convolution, which further reduce the parameters generated by the model. For the BI-FU module, the module first extracts the multi-scale feature maps of the PCB images, which increases the computational complexity. However, after that, through feature fusion, the multi-scale information of the feature maps is retained, and the calculation increment of the module is reduced. These improvements not only improve the detection precision of

**TABLE 1. Number of various defects in PCB defect dataset.**

Type of defects	Number of images	Number of defects
Missing hole	115	497
Mouse bite	115	492
Open circuit	116	482
Short	116	491
Spur	115	488
Spurious copper	116	503
Total	693	2953

PCB defects, but also ensure the detection efficiency of the model.

Then, further comparative experiments are conducted to verify the few-shot detection effect of our FPFM model under different shot settings ( $k=1, 2, 3, 5$ ). As shown in Table 3, the  $AP_{50}$  of our FPFM model can be 3.95%-8.42% better than the state-of-the-art methods under different shot settings. In most other indicators, the FPFM model can also outperform other methods, which shows the general effectiveness of the model under various few-shot settings in PCB defect detection.

### 2) ADAPTATION SPEED

In order to further verify the detection effect of the model, we further compare the adaptation speed and precision of different methods under the 5-shot setting, as shown in Figure 7. We use the number of iterations required for model convergence to express the adaptation speed of the model. If the current novel class AP no longer exceeds the best recorded AP for consecutive 2000 iterations, we consider that the model has converged, and determine the iteration of the best recorded as the model's adaptation speed. In Figure 7, we show the visualization results of the model training process and the number of iterations when the model converges under the 5-shot setting. The curve shows the fine-tuning results in all periods of the novel class AP, and our proposed method achieves better detection precision. Besides, because of the addition of the feature enhancement module, the key features of the feature maps can be highlighted, which helps the model grasp the key feature information of the images faster, and suppresses the interference of useless information. Therefore, our model exhibits a faster adaptation speed. It takes only 3100 iterations to reach the peak, which is still 400 fewer iterations than the FSRW model with the second fastest convergence speed. More importantly, the FPFM model only needs 2000 iterations to achieve 95% peak performance. This represents that 33% training time can be saved, and the performance is only reduced by 5%. This shows that our model has a better adaptation speed and reflects a strong generalization ability. These properties are very helpful for few-shot PCB defect detection.

### 3) COMPARISON OF DIFFERENT TYPES OF DEFECTS

In Fig. 8, we compare the FPFM algorithm under 10-shot with four other few-shot object detection algorithms that are also

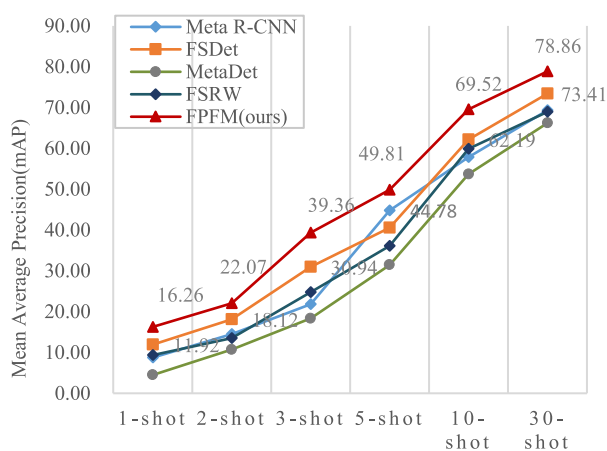


**TABLE 2.** Few-shot detection performance on PCB dataset under 10-shot and 30-shot. “Time” means the tuning time. The best results are shown as black bold.

shot	Methods	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>	Time
k=10	Meta R-CNN [17]	42.81	57.83	41.20	12.96	17.10	40.26	2.5 <sup>+</sup> h
	FSDet [38]	47.34	62.19	44.15	16.51	21.09	42.23	1.8 <sup>+</sup> h
	MetaDet [39]	40.76	53.67	39.81	8.94	17.83	36.41	1.4 <sup>+</sup> h
	MPSR [40]	44.86	59.45	42.26	14.79	17.34	40.07	<b>45<sup>+</sup> min</b>
	DCNet [41]	36.06	52.03	33.25	8.76	17.33	<b>49.98</b>	55 <sup>+</sup> min
	FSRW [16]	41.45	59.91	39.98	14.62	15.87	43.16	1.1 <sup>+</sup> h
	FPFM (ours)	<b>49.91</b>	<b>69.52</b>	<b>45.34</b>	<b>19.87</b>	<b>29.41</b>	49.65	50 <sup>+</sup> min
k=30	Meta R-CNN [17]	45.81	69.34	43.72	16.07	27.43	55.68	3.8 <sup>+</sup> h
	FSDet [38]	50.24	73.41	47.23	21.94	31.32	59.07	3.2 <sup>+</sup> h
	MetaDet [39]	45.14	66.20	44.27	11.56	28.43	57.13	2.6 <sup>+</sup> h
	MPSR [40]	49.25	71.86	46.08	16.92	27.13	57.86	2.0 <sup>+</sup> h
	DCNet [41]	43.61	60.83	40.75	14.15	22.54	51.72	<b>1.5<sup>+</sup> h</b>
	FSRW [16]	44.69	68.94	46.15	20.38	25.04	58.73	1.7 <sup>+</sup> h
	FPFM (ours)	<b>54.17</b>	<b>78.86</b>	<b>50.79</b>	<b>26.83</b>	<b>36.71</b>	<b>63.24</b>	<b>1.5<sup>+</sup> h</b>

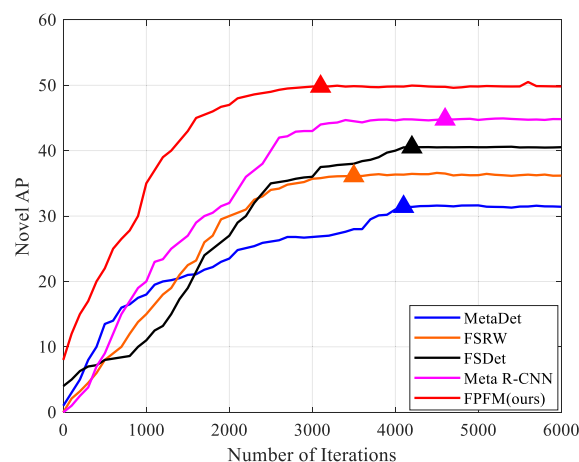
**TABLE 3.** Few-shot detection performance on PCB dataset under different k-shot settings (k=1, 2, 3, 5). The best results are shown as black bold.

Model	1-shot			2-shot			3-shot			5-shot		
	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP	AP <sub>50</sub>	AP <sub>75</sub>
Meta R-CNN [17]	5.35	8.74	4.12	7.19	14.53	6.30	12.34	21.85	10.49	26.79	44.78	<b>25.85</b>
FSDet [38]	6.82	11.92	<b>5.93</b>	10.45	18.12	9.87	16.78	30.94	14.08	21.69	40.52	20.23
MetaDet [39]	2.96	4.48	1.87	6.82	10.69	6.24	10.22	18.34	9.34	16.84	31.43	15.26
FSRW [16]	5.86	9.35	4.75	7.21	13.51	6.54	14.35	24.75	13.76	19.07	36.12	18.65
FPFM (ours)	<b>7.42</b>	<b>16.26</b>	5.63	<b>11.03</b>	<b>22.07</b>	<b>11.96</b>	<b>22.54</b>	<b>39.36</b>	<b>21.30</b>	<b>27.53</b>	<b>49.81</b>	24.87



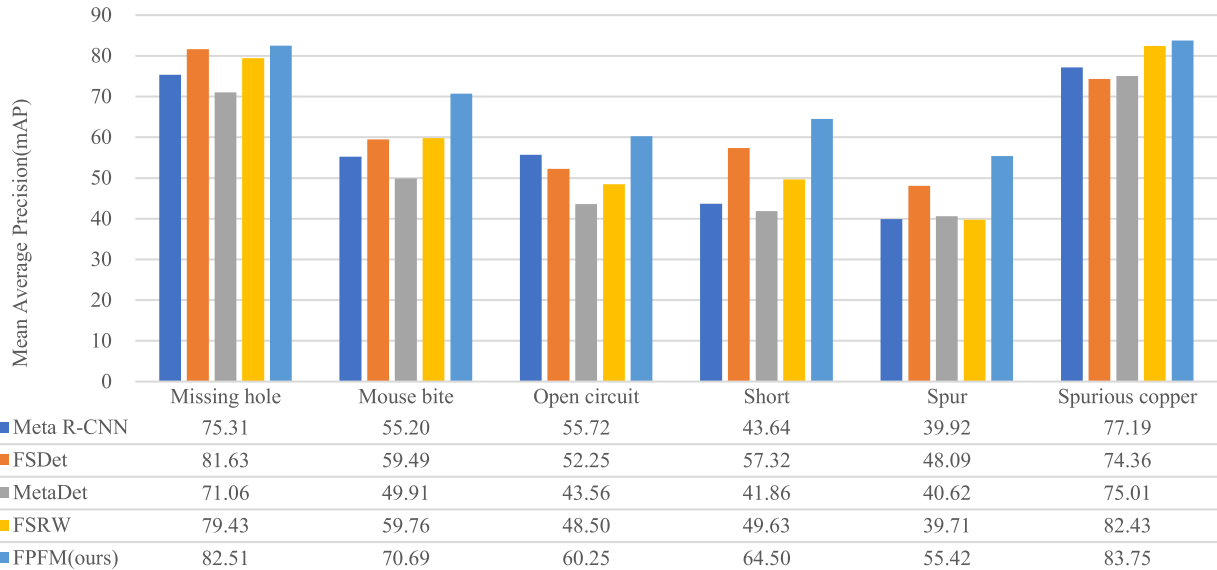
**FIGURE 6.** Few-shot detection performance on PCB dataset under different k-shot settings. The line graph shows that the proposed method is obviously superior to other state-of-the-art methods, where the IoU threshold is 0.5.

based on meta-learning. These four models are Meta R-CNN [17], FSDet [38], MetaDet [39], FSRW [16]. We can find that among the 6 types of defects, the detection precision of the



**FIGURE 7.** Comparison of iterative process of different few-shot methods under 5-shot. The triangle in the figure represents the model convergent point, and the IoU threshold is 0.5.

missing hole and spurious copper defects is relatively high on all models, while the precision of mouse bite, open circuit, short and spur is relatively lower. The main reason is that the defect features of the missing hole and spurious copper are relatively obvious, the defect size is relatively large, and it



**FIGURE 8.** Detection precision of various types of PCB defects by different meta-learning algorithms. Several models are trained using the same few-shot learning strategy, where the IoU threshold is 0.5.

**TABLE 4.** Comparison of experimental results of different base class datasets.

Dataset	Novel mAP					
	1	2	3	5	10	30
VOC07-12	9.57	15.26	33.58	41.75	60.32	69.92
MS-COCO	13.32	19.41	36.94	45.12	64.87	73.60
FSOD	<b>16.26</b>	<b>22.07</b>	<b>39.36</b>	<b>49.81</b>	<b>69.52</b>	<b>78.86</b>

is easy to be detected by the model. Open circuit, short, and spur are generally small in size, so they are easily ignored by ordinary models. For mouse bite defects, the defect characteristics vary greatly, and the sizes of mouse bite defects at different positions are also different, so it is difficult for general models to accurately judge the defect type.

From Fig 8, due to the introduction of the BI-FU multi-scale fusion module and feature enhancement module, the detection precision of small object defects such as open circuit, short, and spur has been greatly improved by the model, and the detection precision of mouse bite defects with different scales has also been significantly improved. For the detection of the missing hole and spurious copper, although the improvement is small, it is still better than other detection models, indicating that FPFM has the highest comprehensive performance for PCB defect detection. When  $k = 10$ , some visual detection results are shown in Fig. 9.

### C. ABLATION EXPERIMENT

In this section, we conduct comprehensive ablation experiments to analyze the effect of various modules in the model.

#### 1) EFFECTS OF DIFFERENT BASE CLASSES

When training the model, we use the FSOD dataset as the base classes for model pre-training. As a control experiment,

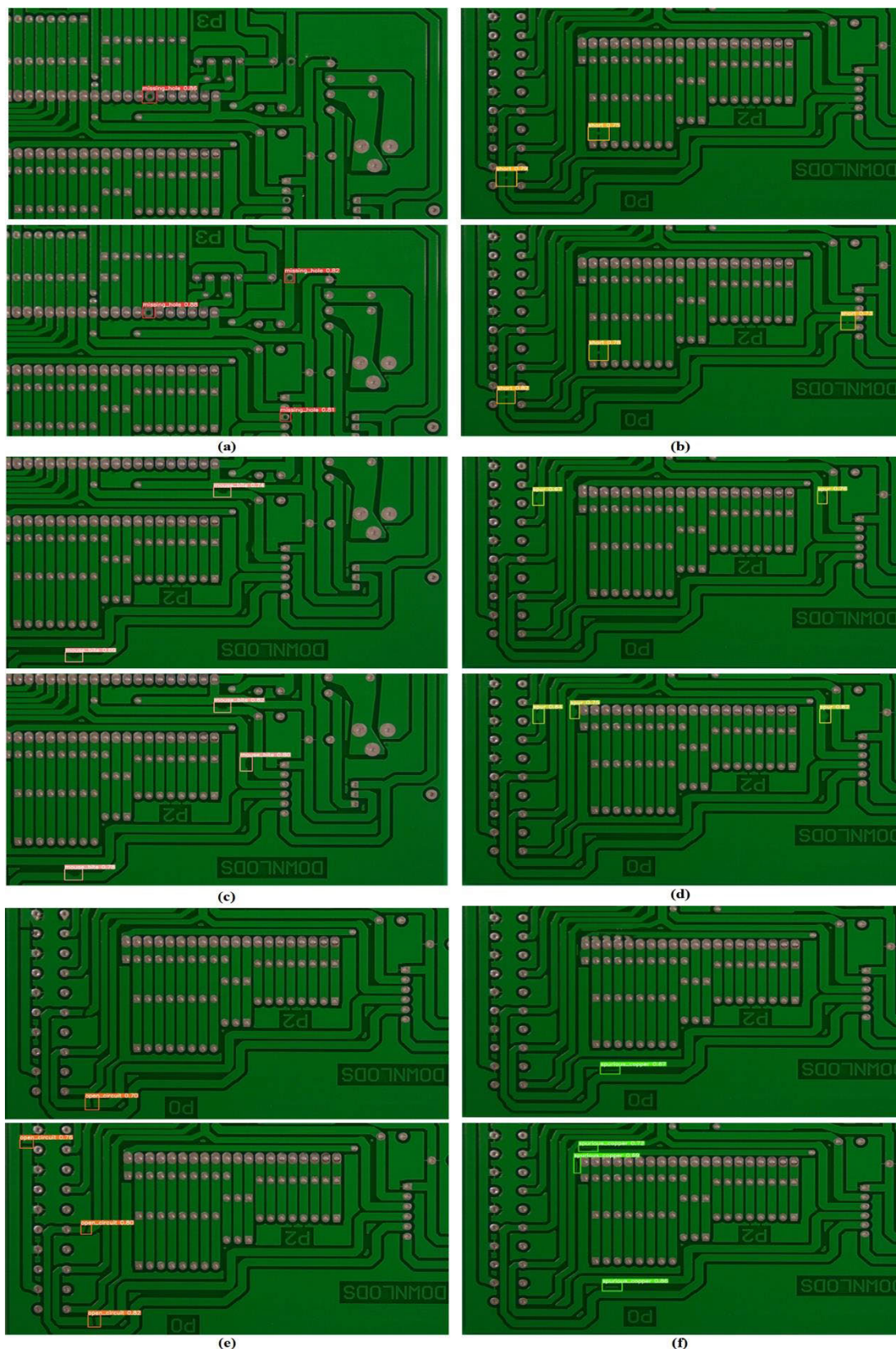
**TABLE 5.** AP and mAP on different backbone networks.

Type of defects	DarkNet-19	DarkNet-53
Missing hole	82.26	82.51
Mouse bite	66.75	70.69
Open circuit	57.42	60.25
Short	61.93	64.50
Spur	49.80	55.42
Spurious copper	81.31	83.75
mAP	66.58	69.52

we select PASVAL VOC [44], [45] and MS-COCO [46] as the base classes for the same pattern of pre-training.

#### a: FSOD

This dataset is a professional few-shot learning dataset with high category diversity, including 83 parent semantics, which are further divided into 1000 leaf categories with a total of 66502 images. In addition, the dataset contains objects with large differences in object size and aspect ratio, which are very conducive to few-shot learning for the model. We use 1000 subcategories as the base classes to pre-train the FPFM model. **PASVAL VOC:** The dataset is a small-scale object detection dataset containing 20 object classes with a total of



**FIGURE 9.** The detection results of the same defect image on the baseline algorithm FSRW and the improved algorithm PPFM are given. Every two lines in the figure represent a type of defect, from top to bottom: (a) Missing hole, (b) Short, (c) Mouse bite, (d) Spur, (e) Open circuit, and (f) Spurious copper. In every two lines, the top is based on the FSRW algorithm and the bottom is based on the PPFM algorithm. All results are obtained under the 10-shot setting.

**TABLE 6.** Ablation studies to verify the impact of different modules on model performance.

Model	DN-53	EL	BI-FU	Novel mAP					
				1	2	3	5	10	30
PPFM	✓			9.87	14.25	26.93	40.24	62.43	70.81
	✓	✓		11.34	17.97	34.89	45.75	64.10	74.95
	✓		✓	12.51	18.26	33.50	44.28	65.36	75.02
	✓	✓	✓	<b>16.26</b>	<b>22.07</b>	<b>39.36</b>	<b>49.81</b>	<b>69.52</b>	<b>78.86</b>

16,551 images. We use VOC 07-12 as our base classes to pre-train the model for comparison. **MS-COCO:** The dataset is a large-scale object detection dataset containing 80 categories, with a total of 123287 images. We selected the 80 category images as our other group of comparison base classes and pre-trained the model. For fair comparison, the novel classes select the same images in the PCB dataset and fine-tune the model under the same shot settings to compare the experimental results.

It can be seen from Table 4 that when FSOD dataset is used as the base classes for pre-training, the model achieves the highest detection precision under all shot settings. With the increase of shots, the precision gap becomes more and more obvious. When PASVAL VOC is used for pre-training model, the detection precision is the worst. Compared with PASVAL VOC, MS-COCO has larger scale and more pictures, which is more helpful for the model to learn low-level semantic information in the images, so it can achieve higher detection precision. However, the key of few-shot learning is to improve the generalization ability of the model. Therefore, the FSOD dataset with a large number of objects and high diversity is more conducive to the meta-learning for the model and can help the model achieve better detection results.

## 2) BACKBONE NETWORK

In this article, the feature extractor uses DarkNet-53 as the backbone network to extract query features, instead of the original DarkNet-19. The experimental results of the two networks under 10-shot are shown in Table 5.

The detection precision of DarkNet-53 is 69.52%, which is better than 66.58% of DarkNet-19. DarkNet-53 draws on the idea of feature pyramid network and introduces residual mechanisms, so it has stronger ability to extract small defects in PCB and has higher detection precision. Therefore, the DarkNet-53 network is adopted as the feature extractor

## 3) EFFECTS OF DIFFERENT MODULES

As shown in Table 6, in order to verify the effect of different modules on model performance, we conduct ablation experiments under different shot settings ( $k = 1, 2, 3, 5, 10, 30$ ). In the table, DN-53 indicates that the feature extractor adopts the DarkNet-53 network, EL refers to the addition of EL-CBAM feature enhancement module, and BI-FU represents the addition of multi-scale fusion module.

**TABLE 7.** Comparative experiment of different fusion methods. The table records the mAP of three different feature fusion methods under  $k$ -shot ( $k=5, 10$ ), where the IoU threshold is 0.5.

	5-shot	10-shot
Fusion Method 1	<b>47.53</b>	<b>68.24</b>
Fusion Method 2	42.76	63.85
Fusion Method 3	44.28	65.36

**TABLE 8.** Ablation experiment of feature enhancement module based on EL-CBAM. The table records the influence of the addition of various attention mechanisms on the model detection precision when the IoU threshold is 0.5 under  $k$ -shot ( $k = 5$  and  $10$ ).

	k=5	k=10
w/o Attention	40.24	62.43
With SENet [43]	42.31	62.98
With ECANet [35]	43.60	63.35
With CBAM [22]	45.37	63.70
With EL-CBAM	<b>45.75</b>	<b>64.10</b>

When only the feature extractor network is replaced, the model detection precision improvement is small due to the lack of enhancement of key feature information and the lack of multi-scale feature information for small object defects. When the feature enhancement module or multi-scale fusion module is added separately, the performance of the model is further improved, which proves the effectiveness of the two modules respectively. When the two modules are joined at the same time, the model detection precision reaches the highest level, which shows that the combination of the two modules is very useful for improving the model performance.

## 4) COMPARISON OF FEATURE FUSION STRATEGIES

The multi-scale feature fusion module is composed of multi-scale feature extraction part BI and adaptive feature fusion part FU to realize image multi-scale feature extraction and fusion. In Table 7, we record the effects of three different feature fusion methods under 5-shot and 10-shot settings. (1) The  $N$  feature maps extracted from the feature extraction part BI are directly input into the reweighted module and converted into reweighting vectors. (2) We connect  $N$  feature

maps according to the channel dimension. Then, we use convolution operation to directly compress the feature map by channel to obtain the fused feature maps. (3) We adopt the method in this paper, that is, we use feature compression vectors to achieve feature maps fusion.

From Table 7, we can find that method 1 achieves the highest detection precision. However, Method 1 directly inputs  $N$  feature maps into the feature reweighting module, which increases the computational complexity of the model and affects the detection efficiency. Method 2 adopts direct convolution compression. It is easy to operate, but too much feature information is lost, which leads to poor detection precision. In order to further balance the model precision and calculation speed, Method 3 is selected as our final feature fusion method.

## 5) COMPARISON OF FEATURE ENHANCEMENT MODULE

In order to fully mine the key information of samples and suppress the interference of useless information, a feature enhancement module based on EL-CBAM is introduced in this article.

In this article, the feature enhancement module is constructed based on SENet [43], ECANet [35], CBAM [22], and EL-CBAM. As can be seen from Table 8, after adding the feature enhancement module, the detection precision of the model has been greatly enhanced. CBAM, which combines both spatial and channel attention, has a better effect on feature enhancement than single dimensional attention of SENet and ECANet. Compared with CBAM, the EL-CBAM module increases the mAP by 0.38% and 0.40% at  $k = 5$  and  $k = 10$ , respectively, which reflects the effectiveness of EL-CBAM. It can further highlight the key feature information of the feature map and help the network detect PCB.

## V. CONCLUSION

In this article, we propose a PCB defect detection method FPFM based on few-shot learning. Based on FSRW, the proposed method improves the network in the feature extractor to darknet-53. What's more, we introduce a feature enhancement module based on the improved CBAM to highlight key regional features, which effectively suppress useless interference information and improve the feature extraction ability of query samples. Meanwhile, the few-shot learning method combined with multi-scale feature fusion is used for PCB defect detection for the first time. This module can extract multi-scale features and fuse them into a high-quality feature map, which improves the detection precision of small-scale defects. Finally, we use the FSOD dataset as the base classes and the PCB dataset as the novel classes to conduct a large number of experiments. We can find that the proposed model outperforms the recent state-of-the-art results under different shot settings. At the same time, we take into account the efficiency and precision of detection, which is conducive to the application of industrial production. However, there is still some work to continue to be researched. At present, the defect types studied in this article are relatively fixed,

and there are still more defect types on PCB that need to be experimentally analyzed. In addition, this article does not consider the multi-label problem and the correlation between labels. In the future work, we consider expanding the model to more PCB defect types, even to other industrial products. At the same time, we will consider the study of multi-label problem, which means that we predict both the types and locations of PCB defects and other attributes such as defect severity.

## REFERENCES

- [1] Y. Shi, Z. Xin, P. C. Loh, and F. Blaabjerg, "A review of traditional helical to recent miniaturized printed circuit board Rogowski coils for power-electronic applications," *IEEE Trans. Power Electron.*, vol. 35, no. 11, pp. 12207–12222, Nov. 2020.
- [2] H.-Y. Lin, C.-J. Lin, and M.-L. Huang, "Optimization of printed circuit board component placement using an efficient hybrid genetic algorithm," *Int. J. Speech Technol.*, vol. 45, no. 3, pp. 622–637, Oct. 2016.
- [3] K. Shringarpure, S. Pan, J. Kim, J. Fan, B. Achkir, and B. Archambeault, "Sensitivity analysis of a circuit model for power distribution network in a multilayered printed circuit board," *IEEE Trans. Electromagn. Compat.*, vol. 59, no. 6, pp. 1993–2001, Dec. 2017.
- [4] Z. Li, J. Zhang, T. Zhuang, and Q. Wang, "Metal surface defect detection based on MATLAB," in *Proc. IEEE 3rd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Oct. 2018, pp. 2365–2371.
- [5] M. X. Li, Q. X. Liu, and C. Y. He, "Research and achievement on cigarette label printing defect detection algorithm," *Appl. Mech. Mater.*, vol. 200, pp. 689–693, Oct. 2012.
- [6] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang, "Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges," *Materials*, vol. 13, no. 24, p. 5755, Dec. 2020.
- [7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV*, vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [10] R. Girshick, "Fast R-CNN," 2015, *arXiv:1504.08083*.
- [11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [12] H. Zhao, C. Wang, R. Guo, X. Rong, J. Guo, Q. Yang, L. Yang, Y. Zhao, and Y. Li, "Autonomous live working robot navigation with real-time detection and motion planning system on distribution line," *High Voltage*, pp. 1–13, Jun. 2022, doi: [10.1049/hve2.12221](https://doi.org/10.1049/hve2.12221).
- [13] R. Zhang, L. Xu, Z. Yu, Y. Shi, C. Mu, and M. Xu, "Deep-IRTarget: An automatic target detector in infrared imagery using dual-domain feature extraction and allocation," *IEEE Trans. Multimedia*, vol. 24, pp. 1735–1749, 2022.
- [14] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," 2017, *arXiv:1703.05175*.
- [15] T. Wang, X. Zhang, L. Yuan, and J. Feng, "Few-shot adaptive faster R-CNN," 2019, *arXiv:1903.09372*.
- [16] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8419–8428.
- [17] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin, "Meta R-CNN: Towards general solver for instance-level low-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9576–9585.
- [18] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6517–6525.
- [19] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, and T. Xiang, "Incremental few-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13843–13852.

- [20] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2019, pp. 5192–5201.
- [21] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [22] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Computer Vision—ECCV*, vol. 11211, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 3–19.
- [23] J.-M. Park, Y.-H. Yoo, U.-H. Kim, D. Lee, and J.-H. Kim, "D3PointNet: Dual-level defect detection PointNet for solder paste printer in surface mount technology," *IEEE Access*, vol. 8, pp. 140310–140322, 2020.
- [24] H. Wu, W. Gao, and X. Xu, "Solder joint recognition using mask R-CNN method," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 10, no. 3, pp. 525–530, Mar. 2020.
- [25] R. Ding, L. Dai, G. Li, and H. Liu, "TDD-Net: A tiny defect detection network for printed circuit boards," *CAAI Trans. Intell. Technol.*, vol. 4, no. 2, pp. 110–116, Jun. 2019.
- [26] V. A. Adibhatla, H.-C. Chih, C.-C. Hsu, J. Cheng, M. F. Abbod, and J.-S. Shieh, "Applying deep learning to defect detection in printed circuit boards via a newest model of you-only-look-once," *Math. Biosciences Eng.*, vol. 18, no. 4, pp. 4411–4428, 2021.
- [27] S. Thrun and L. Pratt, *Learning to Learn: Introduction and Overview*, S. Thrun and L. Pratt, Eds. Boston, MA, USA: Springer, 1998, pp. 3–17.
- [28] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 3, Mar. 2017, pp. 1856–1868.
- [29] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," 2016, *arXiv:1605.06065*.
- [30] A. Li, W. Huang, X. Lan, J. Feng, Z. Li, and L. Wang, "Boosting few-shot learning with adaptive margin loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12573–12581.
- [31] R. Zhang, S. Yang, Q. Zhang, L. Xu, Y. He, and F. Zhang, "Graph-based few-shot learning with transformed feature propagation and optimal class allocation," *Neurocomputing*, vol. 470, pp. 247–256, Jan. 2022.
- [32] H. Chen, Y. Wang, G. Wang, and Y. Qiao, "LSTD: A low-shot transfer detector for object detection," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2018, pp. 2836–2843.
- [33] L. Karlinsky, J. Shtok, S. Harary, E. Schwartz, A. Aides, R. Feris, R. Giryes, and A. M. Bronstein, "RepMet: Representative-based metric learning for classification and few-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5192–5201.
- [34] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3637–3645.
- [35] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.
- [36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [37] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.
- [38] Y. Xiao and R. Marlet, "Few-shot object detection and viewpoint estimation for objects in the wild," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Nov. 2020, pp. 192–210.
- [39] Y.-X. Wang, D. Ramanan, and M. Hebert, "Meta-learning to detect rare objects," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9924–9933.
- [40] J. Wu, S. Liu, D. Huang, and Y. Wang, "Multi-scale positive sample refinement for few-shot object detection," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 456–472.
- [41] H. Hu, S. Bai, A. Li, J. Cui, and L. Wang, "Dense relation distillation with context-aware aggregation for few-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10185–10194.
- [42] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with attention-RPN and multi-relation detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4012–4021.
- [43] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.
- [44] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, Jul. 2010.
- [45] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
- [46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV*, vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 740–755.



**HAODONG WANG** was born in Suqian, Jiangsu, China. He received the B.S. degree from the Taiyuan University of Technology, Taiyuan, China, in 2019, where he is currently pursuing the M.S. degree with the College of Electrical and Power Engineering. He was mainly responsible for the completion of the Fund Project related to defect detection. His research interests include deep learning and computer vision.



**JUN XIE** was born in Taiyuan, China, in 1979. She received the M.S. degree in intelligent autonomous system from Aalborg University, Denmark, in 2005, and the Ph.D. degree in circuit and system from the Taiyuan University of Technology, Taiyuan, in 2009. Since 2011, she has been an Associate Professor with the College of Information Engineering, Taiyuan University of Technology. She has authored over 30 articles. Her research interests include granular computing, computer vision, text mining, and intelligent information processing.



**XINYING XU** was born in Dingxiang, China, in 1979. He received the Ph.D. degree in circuit and system from the Taiyuan University of Technology, Taiyuan, China, in 2009. Since 2018, he has been a Professor with the College of Electrical and Power Engineering, Taiyuan University of Technology. He has authored over 40 articles. His research interests include computer vision, intelligent control, and intelligent information processing.



**ZIHAO ZHENG** received the B.S. degree from the Xiamen University of Technology, Xiamen, China, in 2020. He is currently pursuing the M.S. degree with the College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan, China. His research interests include deep learning and target detection.

...