Generative and Contrastive Combined Support Sample Synthesis Model for Few-/Zero-Shot Surface Defect Recognition

Yuran Dong, Cheng Xie, *Member, IEEE*, Luyao Xu, Hongming Cai, *Senior Member, IEEE*, Weiming Shen, *Fellow, IEEE*, and Haoyuan Tang

***Abstract*— Surface defect detection is one of the most important vision-based measurements (VBMs) for intelligent manufacturing. Existing detection methods mainly require massive numbers of defect samples to train the model to detect the defects. Nowadays, inadequate defect samples and labels are inevitably encountered in industrial data environments due to the highly automated and stable production lines escalatingly deployed, causing fewer and fewer defective products to be produced. Consequently, manual interventions are deeply required to analyze the abnormal sample once an unseen defect accidentally emerges that significantly decreases productivity. To this end, this article proposes a novel few-/zero-shot compatible surface defect detection method without requiring massive or even any defect samples to detect surface defects. First, a novel contrastive generator is proposed to use defects' text descriptions to synthesize "fake" visual features for those rare defects. Then, the synthesized visual features (for support samples) are fused with "real" visual features (for query samples) into a similarity graph to align the relationships between support samples and query samples. After, a class center optimization (CCO) method is proposed to iteratively update the similarity matrix of the graph to obtain the classification probabilities for the query samples. Eventually, the proposed method solves the problem of the lack of defect samples and the inability of few-shot learning-based methods to recognize unseen classes. Massive experiments on eight fine-grained datasets show that our method gains an average of +8.29% improvements on few-shot recognition tasks and achieves an average of +8.23% improvements on zero-shot recognition tasks compared with the state-of-the-art (SOTA) method. Moreover, the proposed method is deployed in a real-world prototype system, and the method's feasibility is finally demonstrated. The core code of the proposed method is available at: https://github.com/NDYBSNDY/AsC.**

***Index Terms*— Contrastive learning, few-shot learning, generative learning, graph embedding (GE), surface recognition, vision-based measurement (VBM), zero-shot learning.**

Manuscript received 4 July 2023; revised 25 September 2023; accepted 20 October 2023. Date of publication 3 November 2023; date of current version 26 February 2024. This work was supported in part by the National Science Foundation of China under Grant 62106216 and Grant 62162064 and in part by the Key Science and Technology Projects of Yunnan Province under Grant 202102AB080019-2 and Grant 202002AB080001-5. The Associate Editor coordinating the review process was Dr. Xianqiang Yang. *(Corresponding author: Cheng Xie.)*

Yuran Dong, Cheng Xie, and Luyao Xu are with the School of Software, Yunnan University, Kunming 650504, China (e-mail: xiecheng@ynu.edu.cn).

Hongming Cai is with the School of Software, Shanghai Jiaotong University, Shanghai 200240, China.

Weiming Shen is with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China.

Haoyuan Tang is with the Metallurgical Research Institute, Kunming Metallurgical Research Institute Co., Ltd., Kunming 650500, China.

Digital Object Identifier 10.1109/TIM.2023.3329163

I. Introduction

AUTOMATED industrial production can reduce labor costs and increase productivity. Recent research has used Bayesian techniques for manufacturing methods [1] to significantly reduce the production process cost by producing input parameters for the desired outcome. Similarly, collective robotic systems for constructing multistory buildings [2] reach state-of-the-art (SOTA) construction speeds. With the rapid development of vision computing in recent years, vision-based measurement (VBM) has become one of the most critical and influential methods for automated industrial production [3], [4]. Surface defect recognition, as an essential part of automated industrial production, has the critical role of improving production efficiency and reducing labor costs. Compared with manual defect recognition, VBM-based machine inspection methods are more objective and efficient. However, due to the environmental constraints on defect sample collection, the types of defects occurring in the production process are uncertain and random. This requires VBM-based defect recognition models to have the ability to fit a few samples and the flexibility to adapt to complex production environments.

However, the remaining surface defects, i.e., the rare-seen or unseen defects, are still hard to detect since there are not enough such defect samples that can be trained. To solve the problem, existing few-shot models [11], [12], [13], [14], [15] only require very few support samples to prompt the model to detect the query samples. The core idea of these methods is to pretrain a model from other related training samples (the samples that are relatively common and easy to collect) in advance. Then, the pretrained model extracts the features from the support samples (rare-seen samples and hard to collect) and tries to update itself to know the defects. After, the updated model extracts the features from query samples and tries to infer the defects of query samples. These methods require at least one defect sample to conduct the query inference. The more defect samples, the higher the accuracy of these detection models. However, in industrial detection practice, some defect samples are hard to collect in advance or have never appeared because the well-optimized smart manufacturing environment has further reduced the defect rate. Consequently, manual interventions are deeply required to analyze the abnormal sample once an unseen defect accidentally emerges that will significantly decrease productivity. Therefore, enabling defect inference without using any support samples becomes one of

the most critical challenges in the surface defect detection field.

To solve the nonsample problem, the zero-shot mechanism is reasonably considered. The basic idea of zero-shot learning for visual computing is to train a cross-modal network to synthesize the visual features from the corresponding semantic features [16], [17], [18], [19], [20]. Based on this cross-modal network, the model can synthesize the unseen visual features without truly learning the sample, i.e., only by describing this object in texts. Then, the synthesized visual features are matched with the visual features of the query sample to infer the category of the query sample. However, existing zero-shot learning methods are designed for general image classification tasks that do not perform well in surface defect detection. This is because these zero-shot learning methods directly match the synthesized visual features with query samples without considering the support information in the industrial data environment, which hardly guarantees detection accuracy.

To this end, this article proposes a few- and zero-shot compatible model by considering both synthesized samples and support samples. The proposed method can detect surface defects in both few-sample and nonsample data environments. First, a novel contrastive generator model is proposed to synthesize the visual features according to the semantic features. Then, the synthesized visual features are filtered and considered as support samples to augment the real support samples. After, a graph-based center feature update method is proposed to match the visual query features to the synthesized support visual features iteratively. The experimental results on massive real-world surface defect datasets show the proposed method significantly outperforms SOTA methods in both few-shot tasks and zero-shot tasks. In the highlights, compared with SOTA methods, our method has significant improvements in both few- and zero-shot surface defect detection. Moreover, the proposed method is deployed in a prototype manufacturing scenario, an automated hot-rolled steel surface detection line, to demonstrate its feasibility and applicability. In summary, the work has the following contributions.

1) Compared with deep-learning-based methods, the proposed method can be decoupled into two phases: sample generation and class inference, and only the class inference phase needs to be deployed in the application, which can significantly reduce the model complexity. Meanwhile, the graph-based class inference method has different feature space distributions and graphs when dealing with different query samples, which is more adaptable to the complex and changing industrial environment.
2) Compared with few-shot learning-based methods, we integrate zero-shot learning, where the types of defects that can be recognized are no longer limited to known classes with support samples, and support samples for unknown classes are obtained through the proposed contrast generator instead of being collected.
3) Compared with zero-shot learning-based methods, our approach uses inference rather than a fixed model for sample prediction, which allows different samples to have different spatial distributions, seen/unseen class predictions do not affect each other, and the proposed method focuses more on unlabeled query samples rather than labeled seen samples or unseen generated samples. Since there is no need to tradeoff the seen/unseen class focus, we achieve the simultaneous optimal performance of the seen/unseen class prediction instead of the tradeoff performance.
4) Compared with SOTA, the proposed method gains an average of $+8.29\%$ improvements on few-shot defect recognition tasks and an average of $+8.23\%$ improvements on zero-shot defect recognition tasks. The proposed method is deployed in a real-world prototype system to evaluate the feasibility and practical implementation.

II. Related Work

A. Different Methods of Defect Recognition

In the latest research, different methods (including methods based on Deep Learning, Few-Shot Learning and Zero-Shot Learning) are used for surface defect recognition and the advantages and disadvantages of different methods are shown in Table I.

The core idea of deep-learning-based methods is to train a fixed classifier to recognize defects through many samples. However, the lack of defect samples leads to the inability to train an accurate convolutional neural network (CNN). Some recent studies [5], [6] have utilized the relevant parameter information of defects to compensate for the wrong recognition of some defect types due to the lack of samples. However, extra information often leads to labeling noise. Yu et al. [7] dealt with labeling uncertainty through knowledge transfer and collaborative learning. Since defect datasets often suffer from data imbalance, deep stochastic chain [8] and gradient-based [9] methods can deal with the difference between defect samples of the same class. The latest research has theoretically solved some existing problems in defect recognition, but in real industrial environments, existing deep-learning-based methods inevitably have some disadvantages as follows.

1) Existing methods incorporate multiple methods (e.g., collaborative learning and deep random chains) on deep-learning networks, leading to complex structures and difficulty fine-tuning the model for nonspecialists.
2) Deep-learning methods obtain a fixed model through training, which leads to models that cannot self-optimize in the application environment, have poor generalization capabilities, and are inflexible.
3) Methods that utilize parameter information to supplement samples lead to models that are difficult to reproduce and tune due to the lack of uniform standards for different parameter representations.
4) The category of the dataset used to evaluate the model is 6, which makes it impossible to know the performance of the model in industrial scenarios where defect types are random and diverse.

The core idea of few-shot learning is to pretrain a model from other related training samples (the samples that are relatively common and easy to collect) in advance. Then, the

TABLE I
COMPARISON OF STATE-OF-THE-ART DEFECT RECOGNITION TECHNIQUES

Methodology-based	Reference	Advantages	Disadvantages
Deep Learning Applied to Surface Defects	[5]-2023 [6]-2023	Making up for sample shortages	1.Complex models 2.Inflexible, poor generalization 3.Extra information is difficult to obtain 4.Limitations in recognizing defect types
	[7]-2023	Resolving labeling uncertainty	
	[8]-2023 [9]-2023	Resolving differences between samples of the same class	
Few-Shot Learning Applied to Surface Defects	[10]-2023 [11]-2022	Reduced background interference	1.Methods do not work with shot=0 2.Recognition type limited to datasets 3.Different models need to be trained for different shots
	[12]-2023	Cross-domain few-shot learning	
	[13]-2023 [14]-2022	Extracting high-dimensional features	
	[15]-2022	Flexibility of the model	
Zero-Shot Learning Applied to Surface Defects	[16]-2022 [17]-2022 [18]-2021	Recognition of unknown novel classes	1.Difficult to balance seen/unseen class bias 2.Hyperparameter tuning is time-consuming and laborious
Zero-Shot Learning Applied to Vision	[19]-2023 [20]-2023 [21]-2022 [22]-2021	Excellent performance on benchmark datasets	1.Difficult to handle small defect datasets 2.Method not applicable to defect images 3.Poor accuracy for defect recognition

pretrained model extracts the features from the support samples (rare-seen samples and hard to collect) and tries to update itself to know the defects. After, the updated model extracts the features from query samples and tries to infer the defects of query samples. In recent studies, in order to better learn the local features of defects and reduce background interference, Zhou et al. [10] designed a feature extractor with the class agnostic mask to extract the defect features and Zhenyu et al. [11] developed a multiresolution-based cropping enhancement method to enhance the unlabeled defect images. By borrowing the idea of multiscale feature extraction, a novel backbone network, ResMSNet, was proposed [12], which realizes cross-domain few-shot learning with the training set and target defect dataset coming from different domains. Since with few support samples (e.g., shot $= 1$), few-shot learning-based methods often perform poorly, and some researchers have also attempted to solve this problem by additional information fusion. Zhao et al. [13] in fusing semantic information based on feature relationships to effectively obtain high-dimensional feature information in a few images. Song et al. [14] generated distinguishable class features by learning affine parameters from the original features, making the model more portable. Effective inference methods often play a crucial role in model performance and Xiao et al. [15] optimized the inference process through graph embedding (GE) and optimal transmission to improve model flexibility. It cannot be denied that few-shot methods have advantages under a few defect sample conditions, but some limitations seem to make them difficult to apply.

1) These methods require at least one defective sample (shot ≥ 1) for inference. This leads to the fact that once an unseen defect appears unexpectedly (shot $= 0$), the few-shot learning-based recognition method breaks down outright, and manual intervention is required to analyze the abnormal sample.
2) Detectable defect types are limited to known dataset classes, which leads to the fact that to use the method in production environments with a large number of classes, it is necessary to build at least one support sample for each possible defect type. However, due to the limitations of production environments, collecting comprehensive support samples of all types is an almost impossible task.
3) Some methods dealing with different numbers of support samples (different shots) require training different models, e.g., FaNet [13], which leads to complex model deployment.

In order to detect novel defect types (classes with no support set) that arise unexpectedly in real production environments, a few studies have attempted to apply zero-shot learning to defect recognition [16], [17], [18]. The basic idea of zero-shot learning for visual computing is to train a cross-modal network to synthesize the visual features from the corresponding semantic features [19], [20], [21], [22]. Based on this cross-modal network, the model can synthesize the unseen visual features without truly learning the sample, i.e., only by describing this object in texts. Then, the synthesized visual features are matched with the visual features of the query sample to infer the category of the query sample. However, the application of zero-shot learning in the field of surface defect recognition is not emphasized, which is mainly due to as follows.

1) Zero-shot learning-based methods often train fixed models with a mixture of seen classes and generated samples of unseen classes. Since the seen/unseen classes are not differentiated, resulting in the accuracy of the two affect each other, the model needs to tradeoff the attention paid to the two to obtain a compromise performance.
2) Existing zero-shot learning models in the field of defect recognition usually have many hyperparameters that need to be selected and optimized, and manual parameter tuning is time-consuming and laborious.
3) Zero-shot learning methods in vision are only applicable to benchmark datasets (e.g., CUB, SUN, and AwA) with samples $>15\,000$, while defect datasets have no more than 1000 samples.

4) Zero-shot learning methods in the visual domain try associating local features with attributes [21], e.g., a bird includes a head, a beak, wings, and feet. This is entirely inapplicable for surface defects where it is difficult to disentangle local features.

B. Development of Few-/Zero-Shot Learning in Different Fields

One-shot learning was first proposed by Fei-Fei et al. [23]. Since the method can quickly learn new knowledge with a few training samples and generalize, it has been rapidly developed in some fields where training data is rare.

In natural language processing, relational classification tasks provide a basis for constructing structured knowledge (e.g., knowledge graphs) by judging the predefined relationship between two target entities in an utterance. However, the development has been slow due to the lack of training data. Xu et al. [24] introduced few-shot learning into the relational classification task for the first time and constructed the FewRel dataset. Many researchers explored this basis [25], [26], [27], [28], and the introduction of few-shot learning made the performance of the relationship classification task continuously improved [29], [30].

In medical image processing, due to the difficulty of biopsy label acquisition, Qinghua et al. [31] first attempted to introduce few-shot learning into the ultrasound breast tumor diagnosis system and achieved excellent performance. In recent years, the few-shot method has been widely used in the medical field, including the recognition of COVID-19 from rare chest images [32], human cell categorization in rare datasets [33], autism facial feature categorization [34], skin image categorization [35], and healthcare safety monitoring [36].

Palatucci et al. [37] proposed the concept of zero-shot learning due to the ability of this method to detect rare or unseen objects in an image. In some industrial application scenarios, the zero-shot method was introduced.

In remote sensing scene classification, satellite images are prone to new classes of objects beyond the expected scene, which leads to the collapse of deep-learning-based methods. Li et al. [38] introduced zero-shot learning into remote sensing scene classification and proposed a new method for recognizing images from unseen classes. Further studies tried to combine knowledge graphs with zero-shot learning and achieved better performance [39]. The latest methods have also continued to apply zero-shot learning to remote sensing scene classification [40], [41], remote sensing image defogging [42], and remote sensing image super-resolution [43].

In intelligent manufacturing scenarios, due to the diversity and randomness of industrial faults, some real fault samples are difficult to obtain or never occur, so zero-shot learning methods have been widely used in the field of industrial fault diagnosis in recent years [44], [45], [46], [47].

III. Methods

The general framework of the few-/zero-shot visual inspection method is shown in Fig. 1, which consists of two parts: the contrastive generator (see Section III-B) and the graph-based few-/zero-shot inference (see Section III-C).

A. Problem Formulation

Let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ denote the raw visual feature space, the corresponding image labels, and the dataset, respectively. Assume $\mathcal{D}^s = \{\mathcal{X}^s \in \mathcal{X}, \mathcal{Y}^s \in \mathcal{Y}\}$ is a training set consisting of seen classes and $\mathcal{D}^u = \{\mathcal{X}^u \in \mathcal{X}, \mathcal{Y}^u \in \mathcal{Y}\}$ is a test set consisting of unseen classes. The constraints are $\mathcal{D}^s \cap \mathcal{D}^u = \emptyset$ and $\mathcal{D}^s \cup \mathcal{D}^u = \mathcal{D}$. At the same time, the class-level text features are provided $\mathcal{A} = \mathcal{A}^s \cup \mathcal{A}^u$, where $\mathcal{A}^s$ correspond to the seen classes in $\mathcal{D}^s$, and $\mathcal{A}^u$ correspond to the unseen classes in $\mathcal{D}^u$. For the N-way K-shot task, N unseen classes are selected as the test set in $\mathcal{D}^u$, in which K with-labeled samples are reserved for each selected class as the support set $\mathcal{D}^t$, and the unlabeled samples in the test set are the query set $\mathcal{D}^q$. K is usually small or even nonexistent (i.e., $K = 0$, $K = 1$, and $K = 5$). Unlike the common task, the final support set of the proposed task is $\mathcal{D}^t \cup \mathcal{D}^a$, and $\mathcal{D}^a$ is a text prompt extracted from $\mathcal{A}^u$ corresponding to N classes.

B. Contrastive Generator

1) Visual Feature Synthesizing: Let $a^s \in \mathcal{A}^s$ be a text feature of a seen class while $x^s \in \mathcal{X}^s$ be the visual feature of the corresponding class. The input to the conditional generation network G is obtained by splicing the text features a^s and Gaussian noise $\epsilon \sim \mathcal{N}(0, 1)$. G outputs the synthetic visual samples $\bar{x}^s = G(a^s, \epsilon)$. Meanwhile, the discriminator network D is used to discriminate a real pair (x^s, a^s) from a synthetic pair $(\bar{x}^s, a^s)$. The feature generator network G and the discriminator network D can be learned by optimizing the following adversarial objective:

$$\begin{aligned}\mathcal{L}_G &= -\mathbb{E}_{\epsilon \sim p_\epsilon}\left[D(G(a^s, \epsilon))\right] + \mathcal{L}_{\text{cls}}(G(a^s, \epsilon)) \\ \mathcal{L}_D &= -\mathbb{E}_{\epsilon \sim p_\epsilon}\left[D\left(G(a^s, \epsilon)\right)\right] - \mathbb{E}_{x \sim p_d}\left[D\left(x^s\right)\right] + \mathcal{L}_{\text{cls}}\left(x^s\right).\end{aligned} \tag{1}$$

$\mathcal{L}_G$ is the loss function of generator G. It consists of a discriminator error $\mathbb{E}$ and a class classification loss $\mathcal{L}_{\text{cls}}$. $\mathcal{L}_D$ is the loss function of discriminator D that consists of a synthesized visual feature discriminating error, a real visual feature discriminating error, and a class classification loss $\mathcal{L}_{\text{cls}}$.

2) Contrastive Loss for Real Features: Let the embedding of a visual sample x^s be denoted as $f^s = E(x^s)$, E is an embedding function that maps the raw visual sample x^s into the embedding space. To learn the embedding function E, for each data point f^s embedded with real or synthetic features, try to randomly take one sample f^{s+} of the same class as the f^s sample as a positive sample and $f^{s+} \neq f^s$. And take N samples randomly as negative samples f_j^{s-} from the set of all class samples not of the same class as f^{s+} samples. Then, a positive sample f^{s+} is mixed with N negative samples f_j^{s-} into an unlabeled set of samples f_j^s, and the correlation scores between the real embedding and the other real embedding samples are obtained by calculating the dot product similarity between f^s and f_j^s. Finally, the known labeled sample f^s is used to distinguish the only positive sample in f_j^s. For example, as shown in Fig. 2, if the embedded real sample f^s class is Am, a randomly selected positive sample f^{s+} class is also Am, but f^s and f^{s+} are different pictures. Meanwhile,

samples of classes different from Am (i.e., convexity, In, blister, and bump) can be selected as negative samples f_j^{s-}.

It is worth noting that since the known labeled samples f^s need to distinguish only the positive sample among the set of $N+1$ positive and negative samples, the size of N (the number of negative samples) determines the classification difficulty. If N is small, it is not easy to learn discriminative class features, and if N is too large, it leads to long training time and high overhead. At the same time, too accurate class features may lead to the real embedded feature distribution not being compatible with the synthetic feature distribution with significant deviation, making the performance decline. Thus, by weighing the model accuracy against the training overhead, the number of negative samples (N) is set to 25% of the total number of samples (classes different from f^{s+}).

In summary, consider using a contrast loss function called InfoNCE[1] to compute the expected loss of contrast embedding $\mathcal{L}_{\mathrm{CR}}$ for the real embedding samples f^s and f_j^s. The formula is shown as follows:

$$\mathcal{L}_{\mathrm{CR}} = -\mathbb{E}_{\mathcal{F}}\left[\log \frac{\exp((f^s)^\top \cdot f^{s+}/\tau)}{\exp((f^s)^\top \cdot f^{s+}/\tau)+\sum_{j=1}^{N}\exp((f^s)^\top \cdot f_j^{s-}/\tau)}\right]. \tag{2}$$

Here, N denotes the number of negative samples f_j^{s-} (f_j^{s-} and f^s belong to different seen classes). $f^s \neq f^{s+}$ but they belong to the same seen class. $\tau > 0$ is the temperature hyperparameter, which is used to control the convergence rate of the model.

3) Contrastive Loss for Synthesized Features: Analogously, to make the synthesized samples $\bar{x}^s$ fit the real embedding space and increase the distribution distance between different classes of generated samples. The positive and negative samples of the synthetic features are shown in Fig. 2, which are selected in the same way as the real features, where the number of negative samples is also taken as 25% of the total number of samples (which are not of the same class as f^s). The positive samples are also taken randomly from among the samples of the same class as f^s. Referring to (2), let $\bar{f}^s = E(\bar{x}^s)$, the contrastive loss $\mathcal{L}_{\mathrm{CS}}$ of the synthesized features is defined as follows:

$$\mathcal{L}_{\mathrm{CS}} = -\mathbb{E}_{\mathcal{F}}\left[\log \frac{\exp((f^s)^\top \cdot \bar{f}^{s+}/\tau)}{\exp((f^s)^\top \cdot \bar{f}^{s+}/\tau)+\sum_{j=1}^{N}\exp((f^s)^\top \cdot \bar{f}_j^{s-}/\tau)}\right]. \tag{3}$$

During the contrastive generator training process, only seen visual features $\mathcal{X}^s$, seen semantic features $\mathcal{A}^s$, and seen labels $\mathcal{Y}^s$ are used. During the few-/zero-shot predicting, a generator $G(\mathcal{A}^u, \epsilon)$ is used to generate the synthesized visual features $\mathcal{X}^u$, after which the synthesized visual features are mapped to the embedding space by the embedding function $E : \bar{\mathcal{F}}^u = E(G(\mathcal{A}^u, \epsilon))$, which includes only the features of the unseen class. Raw features of unseen classes are also mapped to the embedding space $\mathcal{F}^u = E(\mathcal{X}^u)$.

[1]https://arxiv.org/abs/1807.03748

C. Graph-Based Few-/Zero-Shot Inference

1) For Zero-Shot Inference: In the industry-specific zero-shot visual inspection process, first, the similarity matrix S is obtained by calculating the feature similarities among the support features $\bar{\mathcal{F}}^t$ and the query features $\mathcal{F}^q$ synthesized from the contrastive generator (see Section III-B). Then, the similarity graph is constructed from the adjacency similarity matrix S. The class center $\bar{\mathcal{T}}_i^{(0)}$ is obtained by initializing the support node $\mathcal{T}$. Finally, the final classification probability matrix $M_{i,j}$ is obtained by continuously updating the class center $\bar{\mathcal{T}}_i^{(k+1)}$ with the classification probability matrix $M_{i,j}^{(k+1)}$. The predicted label is obtained as $\hat{Y}_i$ by selecting the maximum probability of $M_{i,j}$.

In the above process, all support embedding samples are synthesized by the proposed generator G and the embedding function E (see Section III-B), all query samples are processed by the embedding function E, and all query and support samples belong to the unseen class. Fig. 1 provides the overview process of zero-shot inference.

Let S be the adjacent similarity matrix, and $S_{i,j}$ stores a similarity value of feature i and feature j. Equation (4) provides the definitions of S

$$S_{i,j} = \begin{cases} w^\top\left(\bar{f}_i^t \parallel \bar{f}_j^t\right), & \text{if } i, j \text{ in } \bar{\mathcal{F}}^t \\ w^\top\left(f_i^q \parallel \bar{f}_j^t\right), & \text{if } i \text{ in } \mathcal{F}^q, j \text{ in } \bar{\mathcal{F}}^t \\ w^\top\left(\bar{f}_i^t \parallel f_j^q\right), & \text{if } i \text{ in } \bar{\mathcal{F}}^t, j \text{ in } \mathcal{F}^q \\ w^\top\left(f_i^q \parallel f_j^q\right), & \text{else.} \end{cases} \tag{4}$$

Here, $\bar{\mathcal{F}}^t$ is the synthesized visual feature embedding space of the support set, while $\mathcal{F}^q$ is the real visual feature embedding space of the query set, $\bar{\mathcal{F}}^t \in \bar{\mathcal{F}}^u$, $\mathcal{F}^q \in \mathcal{F}^u$. $\bar{f}^t$ denotes the synthesized visual embedding features. f^q denotes the real visual embedding features. w denotes a parameter matrix. In the experiment, for each node in S, only Top-k similar neighbors remain. The rest neighbors are marked as 0 in similarities.

Before inference on query set categories, it is crucial to construct a relational network containing support set labeling information and unlabeled query set information. The proposed method constructs interrelationships between query samples and support samples through GE to fully utilize the known label information. The graph-based inference process usually needs to initialize a center for each class and continuously optimize the class centers to achieve class differentiation during the inference process. Different methods of class center selection [48], [49], [50] often affect the quality of inference results and iteration efficiency. In order to obtain more reasonable class centers, the self-attention (SA) mechanism is introduced. By further correlating feature information between samples, the proposed method obtains class center points with rich defect feature information, which is also more global in biasing the support sample distribution.

Eventually, the SAGE module is constructed to further improve the class center optimization module (CCO) performance through sample information integration and class center initialization, which contains (5) and (6).

Given a diagonal matrix $D_{i,j} = \sum_j S_{i,j}$, the adjacency matrix S, a normalization function $\mathtt{Norm}(\cdot)$, a SA function

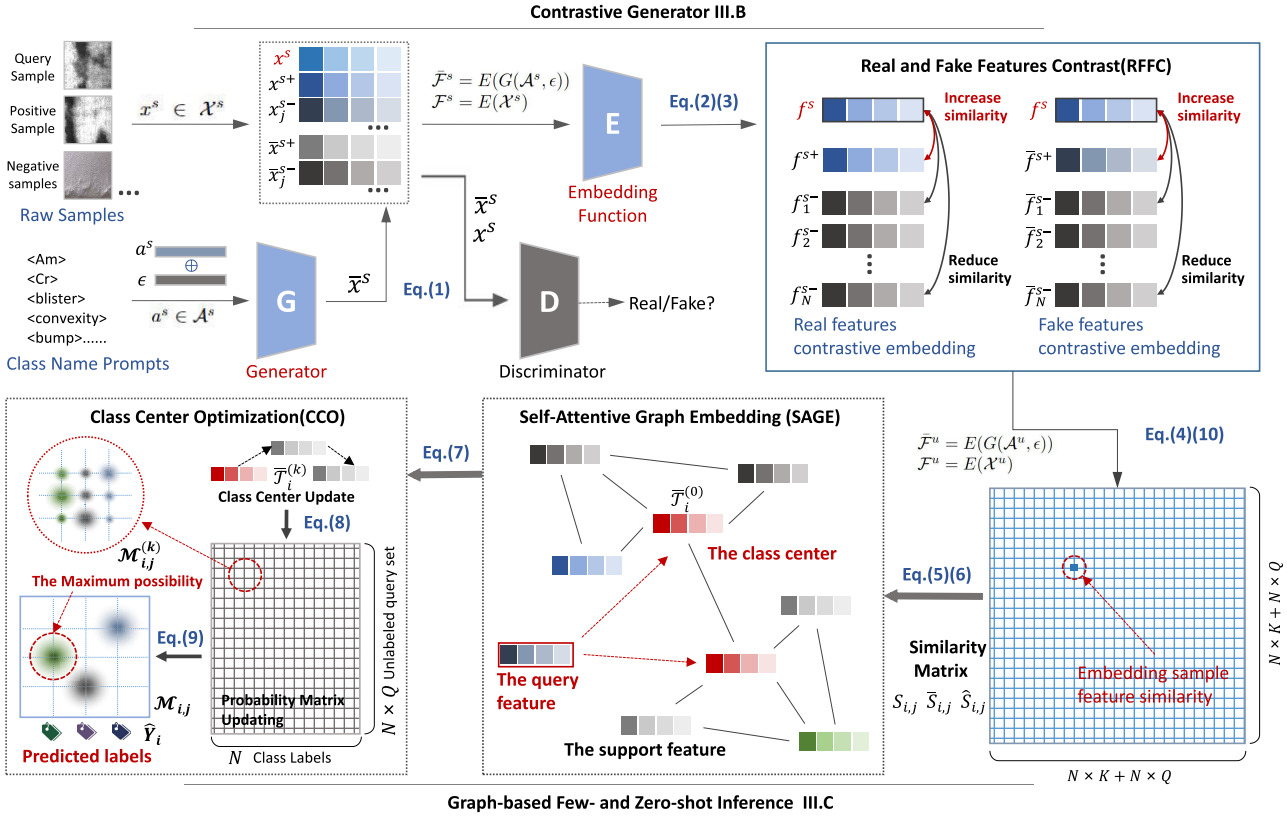

Fig. 1. Overview framework of the proposed method for few-/zero-shot visual inspection.

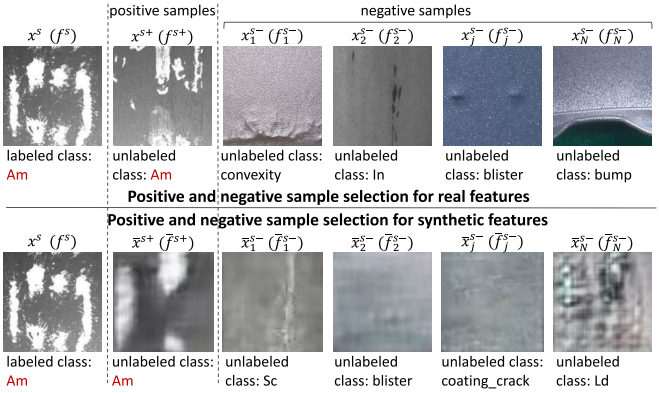

Fig. 2. Example of positive and negative samples of real and synthesized features, with synthesized sample resolution of 64 × 64 pixels.

$\texttt{Self}(\cdot)$, and a one-layer learn-able weight matrix W, the GE is defined as follows:

$$\mathcal{T} \cup \mathcal{Q} = \texttt{Self}\Big(\texttt{Norm}\Big(D^{-\frac{1}{2}}(S+\xi\cdot E)^{\theta}D^{-\frac{1}{2}}W\Big)\Big). \quad (5)$$

Here, $\mathcal{T}$ is the GE for all support samples. $\mathcal{Q}$ is the GE for all query samples. E is the node self-connection matrix and ξ is the weight parameter that balances the importance of the neighboring node and self-node information. θ is the embedding ratio parameter.

Based on the support samples feature matrix $\mathcal{T}$, the support classes' center feature matrix $\bar{\mathcal{T}}$ can be calculated by the following formula:

$$\bar{\mathcal{T}}_i^{(0)} = \frac{1}{K}\sum_{k=K\cdot(i-1)+1}^{K\cdot(i-1)+K}\mathcal{T}_k \quad (6)$$

where K is the number of support samples for a class. $\mathcal{T}_k$ denotes the kth support sample feature while $\bar{\mathcal{T}}_i$ represents the ith class' center feature. Here, $^{(0)}$ means the initial center feature

$$\bar{\mathcal{T}}_i^{(k+1)} = (1-\alpha)\cdot\bar{\mathcal{T}}_i^{(k)} + \alpha\cdot\left(\frac{K\cdot\bar{\mathcal{T}}_i^{(0)} + \sum_{j=1}^{N}(M_{i,j}^{(k)}\cdot\mathcal{Q}_j)}{K+\sum_{j=1}^{N}M_{i,j}^{(k)}}\right). \quad (7)$$

$\bar{\mathcal{T}}_i^{(k+1)}$ denotes the center feature of the ith class after $(k+1)$ iterations. α is an updating rate parameter. The updating is faster if α is bigger and vice versa. In our experiments, α is set to 0.2. Here, $M_{i,j}$ represents the classification probability of the jth query sample $\mathcal{Q}_j$ belonging to the ith class. It is calculated by measuring the distance between the class center feature $\bar{\mathcal{T}}_i^{(k)}$ and the query feature $\mathcal{Q}_j$, as defined in the following:

$$M_{i,j}^{(k+1)} = \texttt{SinkHorn}\Big(||\bar{\mathcal{T}}_i^{(k)} - \mathcal{Q}_j||^2, \lambda\Big). \quad (8)$$

Here, λ is a regularization parameter that will be discussed in the experiment. The settings of the Sinkhorn function are referred from [53]

$$\hat{Y}_i = \arg\max_i\big(M_{i,j}\big). \quad (9)$$

Finally, based on the iterated probability matrix M, the zero-shot inference can be conducted by selecting the maximum value of M_j for the given query sample j, the predicted label matrix for all classes is $\hat{Y}_i$, as shown in (9).

2) For Few-Shot Inference: The key challenge of few-shot inference is that the number of support samples is too small (normally only one to five samples) compared with the training and query samples, causing a serious distribution skewness problem. To handle the problem, based on the idea of zero-shot inference proposed, we try to augment the support set $\mathcal{F}^t$ by adding extra synthesized samples $\bar{\mathcal{F}}^t$ from the feature generator (see Section III-B). However, it was observed in the experiment that simply adding $\bar{\mathcal{F}}^t$ into $\mathcal{F}^t$ does not obviously improve the classification accuracy. The reason for this phenomenon is that some synthesized samples $\bar{f}^t$ might deviate from the real visual feature distribution. These deviated samples will disturb the model inference.

To guarantee the quality of the synthesized samples, we do not directly add $\bar{\mathcal{F}}^t$ into $\mathcal{F}^t$. Instead, $\bar{\mathcal{F}}^t$ is filtered in advance by a classifier based on $\mathcal{F}^t$ in which only the correct classified samples $\bar{F}^t$ are added into $\mathcal{F}^t$. The inference process is similar to the zero-shot inference. See (5)–(9). The only difference is the similarity graph construction, as defined in the following equation:

$$\bar{S}_{i,j} = \begin{cases} w^\top\left(\bar{f}_i^t \parallel \bar{f}_j^t\right), & \text{if } i, j \text{ in } \bar{\mathcal{F}}^t \\ w^\top\left(f_i^t \parallel \bar{f}_j^t\right), & \text{if } i \text{ in } \mathcal{F}^t,\ j \text{ in } \bar{\mathcal{F}}^t \\ w^\top\left(\bar{f}_i^t \parallel f_j^t\right), & \text{if } i \text{ in } \bar{\mathcal{F}}^t,\ j \text{ in } \mathcal{F}^t \\ w^\top\left(f_i^t \parallel f_j^t\right), & \text{else.} \end{cases} \tag{10}$$

Let $\bar{\mathcal{F}}^{t'}$ be the filtered synthesized sample set. We then have the augmented support set $\hat{\mathcal{F}}^t = \{\bar{\mathcal{F}}^{t'} \cup \mathcal{F}^t\}$. It is worth noting that $|\bar{\mathcal{F}}^{t'}| \gg |\mathcal{F}^t|$. Based on the support set $\hat{\mathcal{F}}^t$ and the query set $\mathcal{F}^q$ obtained by filtering, the similarity graph $\hat{S}$ for few-shot inference is re-constructed [refer to (4)].

Finally, based on the new similarity graph $\hat{S}$, few-shot inference can be conducted through (5)–(9).

IV. Experiments

A. Preliminaries

1) Datasets: To verify the effectiveness of the proposed method for surface defect recognition, we validated it on eight different datasets, which mainly include three surface defect datasets (MSD-Cls [15], FSC-20 [13], and MT-CF) and five fine-grained datasets DTD,[2] EuroSAT,[3] RESISC45,[4] MED-3 (consists of a blood cell image database,[5] multisource dermoscopic images of pigmented lesions HAM10000,[6] and optical coherence tomography (OCT) images[7]), and GTSRB.[8]

MSD-Cls [15] is a metal surface defect dataset that contains aluminum and steel with different defect types. In MSD-Cls, only a few training data are about steel defects. However, the test data are all about aluminum defects that cause a serious cross-domain problem, making it hard to detect accurately. FSC-20 MT-CF dataset consists of the oil pollution defect database,[9] the annotated road crack image database CrackForest,[10] and the magnetic tile surface defect database.[11]

Noting that MSD-Cls, MT-CF, and MED-3 are cross-domain datasets consisting of more than three different datasets from the same industrial domain, the significant data differences are extremely challenging. RESISC45, GTSRB, and DTD datasets are fine-grained multicategory datasets with insignificant class characteristics compared to conventional few-shot visual inspection datasets. Extra experiments, including ablation study, hyperparameter study, and base generator discussions, are conducted on the MSD-Cls dataset.

2) Dataset Splits: To simulate the few-sample data environment, all the above datasets are narrowed by randomly selecting 10–50 samples for each class. Then with reference to the PS-split,[12] the database is divided into the training and validation set $\mathcal{D}^s$ (seen class) and the test set $\mathcal{D}^u$ (unseen class).

3) Experimental Setups: In few-shot inference comparison experiments, we follow the different backbone network settings of the SOTA methods (i.e., ResNet-12 [62], ResNet-18 [62], and WRN [63]). In zero-shot inference comparison experiments, CLIP [64] was used to extract visual features $\mathcal{X}$ and corresponding text features $\mathcal{A}$ of the seen classes for all methods (using only class names as text cues).

4) Evaluation Metrics: For few-shot tasks, accuracy (acc) and way-shot metrics are applied. Here, the way denotes the number of classes in $\mathcal{D}^u$ while the shot means the number of support samples for each class. For example, a five-way-one-shot means five to-be-classified classes with one support sample for each class during the testing.

For the generalized zero-shot learning (GZSL) task, following the metrics,[13] Top-1 classification accuracy on seen classes (S) and unseen classes (U) are evaluated. The harmonic mean (H) of S and U is used to represent the final performance of zero-shot visual inspection where $H = 2 \times S \times U/(S + U)$.

To report stable results, 10 000 random draws with 95% confidence are conducted to obtain the average accuracy values for each evaluation.

B. Evaluations

1) Few-Shot Inference Comparison: Table II provides the way-shot results of few-shot visual inspection.

For the zero-shot comparison, the few-shot competitive methods are adjusted to zero-support samples if their source codes are available. Else, "−" in Table II denotes that zero-shot inference can not be reproduced for the corresponding method. On the dataset with mixed seen and unseen classes, our method achieves an average from +25.4% to +37.25% improvement compared with PTNET and GTnet. Notably, this is the first attempt to apply the few-/zero-shot compatible models in industry-specific visual inspection domains and achieves a significant improvement.

[2]https://paperswithcode.com/dataset/dtd
[3]https://paperswithcode.com/dataset/eurosat
[4]https://paperswithcode.com/dataset/resisc45
[5]https://github.com/Shenggan/BCCD_Dataset
[6]https://dataverse.harvard.edu
[7]https://www.kaggle.com/datasets/paultimothymooney/kermany2018
[8]https://paperswithcode.com/dataset/gtsrb
[9]http://faculty.neu.edu.cn/songkc/en/z-dylm/263267
[10]https://github.com/cuilimeng/CrackForest-dataset
[11]https://github.com/abin24/Magnetic-tile-defect-datasets
[12]https://arxiv.org/abs/1707.00600
[13]https://arxiv.org/abs/1707.00600

TABLE II
COMPARISON RESULT OF FEW-SHOT VISUAL INSPECTION

Datasets	K-shot	S2M2_R [51]	ICI-FSL [52]	PTNET [53]	Latent [54]	TRA [55]	GTnet [15]	fsl-rsvae [56]	FaNet [13]	FSL_Cls [10]	Ours
MSD-Cls (5-way)	0-shot	20.00±00.00	-	20.00±00.00	-	-	20.00±00.00	-	-	-	**46.33±0.36**
	1-shot	60.25±0.60	60.10±0.72	73.68±0.16	73.94±0.61	62.57±0.71	69.57±0.91	71.25±0.64	70.10±0.18	60.37±0.41	**79.81±0.22**
	5-shot	75.72±0.43	75.87±0.41	77.98±0.11	78.09±0.11	77.07±0.10	79.09±0.72	77.03±0.56	82.37±0.10	77.9±0.38	**89.59±0.06**
FSC-20 (5-way)	0-shot	20.00±00.00	-	20.00±00.00	-	-	20.00±00.00	-	-	-	**61.53±00.11**
	1-shot	69.84±1.47	63.50±0.66	88.72±0.19	81.03±0.62	76.31±0.59	74.59±0.22	81.22±0.12	86.21±0.22	59.88±0.28	**88.94±0.14**
	5-shot	81.51±0.79	72.86±0.51	93.65±0.06	86.40±0.29	83.34±0.25	83.60±0.09	91.05±0.53	94.24±0.05	77.38±0.11	**95.33±0.05**
MT-CF (3-way)	0-shot	33.33±00.00	-	33.33±00.00	-	-	33.33±00.00	-	-	-	**70.66±00.03**
	1-shot	77.60±1.65	59.62±0.92	89.17±0.23	82.14±0.76	80.63±0.72	90.78±0.23	72.22±0.18	80.22±0.11	92.37±0.05	**95.50±0.05**
	5-shot	86.97±1.26	76.87±0.57	93.31±0.07	89.04±0.24	85.76±0.36	95.79±0.04	92.72±0.01	91.35±0.76	95.22±0.13	**99.97±0.01**
EuroSAT (3-way)	0-shot	33.33±00.00	-	33.33±00.00	-	-	33.33±00.00	-	-	-	**77.10±00.11**
	1-shot	86.07±1.41	68.87±0.88	93.41±0.18	92.70±0.66	90.71±0.49	87.95±0.23	87.83±0.42	78.77±0.21	62.76±0.07	**97.53±0.10**
	5-shot	92.78±0.74	79.86±0.47	94.96±0.07	94.84±0.24	92.42±0.27	93.43±0.08	92.44±0.03	87.68±0.10	83.33±0.72	**98.64±0.05**
RESISC45 (5-way)	0-shot	20.00±00.00	-	20.00±00.00	-	-	20.00±00.00	-	-	-	**56.44±00.10**
	1-shot	54.91±1.93	51.93±1.06	68.77±0.29	68.38±0.90	61.49±0.77	58.00±0.25	71.27±0.03	51.24±0.16	55.86±0.14	**84.06±0.12**
	5-shot	74.31±1.24	69.21±0.74	80.36±0.15	80.18±0.48	74.05±0.47	72.77±0.15	79.01±0.37	70.80± 0.11	70.56±0.81	**91.34±0.05**
MED-3 (4-way)	0-shot	25.00±00.00	-	25.00±00.00	-	-	25.00±00.00	-	-	-	**26.72±00.01**
	1-shot	25.88±0.80	28.94±0.53	26.62±0.13	26.66±0.44	28.75±0.43	29.79±0.15	27.18±0.27	36.94±0.15	14.93±0.06	**37.72±0.19**
	5-shot	27.83±1.11	33.56±0.48	28.34±0.13	28.46±0.41	33.62±0.39	34.90±0.13	33.92±0.27	40.67±0.60	38.02±0.71	**48.55±0.14**
GTSRB (5-way)	0-shot	20.00±00.00	-	20.00±00.00	-	-	20.00±00.00	-	-	-	**62.03±00.26**
	1-shot	75.71±2.65	67.38±1.09	87.56±0.23	**87.87±0.71**	82.23±0.74	81.19±0.23	76.48±0.05	75.03±0.28	66.56±0.62	83.55±0.45
	5-shot	86.69±1.95	79.78±0.76	93.82±0.14	94.15±0.42	93.57±0.33	**95.53±0.08**	91.03±0.26	92.91±0.03	93.46±0.08	92.19±0.26
DTD (5-way)	0-shot	20.00±00.00	-	20.00±00.00	-	-	20.00±00.00	-	-	-	**61.02±00.03**
	1-shot	54.20±2.02	46.36±0.87	60.08±0.26	59.95±0.85	56.74±0.76	55.03±0.23	52.42±0.22	52.89±0.19	48.62±0.12	**71.20±0.26**
	5-shot	70.49±1.60	59.17±0.77	73.92±0.18	73.77±0.57	68.91±0.53	69.09±0.17	71.82±0.01	69.89±0.11	59.94±0.03	**83.86±0.11**

For one-shot tasks, our method obtains +5.87%, +4.72%, +4.12%, and +7.93% improvements in MSD-Cls, MT-CF, EuroSAT, and MED-3 datasets, respectively, compared to the second-best method, while 4.01 decreases in GTSRB dataset. Notably, our method obtains +12.79% and +11.12% significant improvements over the second-best method in the RESISC45 and DTD datasets, respectively, with improvements >10%. For five-shot tasks, the highlight of the comparison results is that our method obtains +10.5%, +10.98%, +13.65%, and +9.94% significant improvements in the MSD-Cls, RESISC45, MED-3, and DTD datasets, respectively, compared to the second-best method (the average improvement was >10%). On other datasets (MT-CF and EuroSAT), our method obtains +4.18% and +3.95% improvement, while there is a 3.34 decrease in the GTSRB dataset.

The highlights also show from Table II that the proposed method has significant improvements in MSD-Cls, MED-3, RESISC45, and DTD. In detail, from +5.9% to +13.4% improvements are achieved in MSD-Cls and MED-3 datasets (the training and testing classes are not intersected) on the few-shot inference. From +10.1% to +13.2% improvements are obtained in RESISC45 and DTD datasets (relatively larger numbers of classes for the few-shot task) on the few-shot inference. This indicates the proposed method can obtain more critical class differentiation in nontrivial datasets.

On the large-scale few-shot classification dataset FSC-20, the proposed method improves +2.73% and +1.09% on one- and five-shot, respectively, compared to FaNet, the method applied to the FSC-20 dataset. It is worth mentioning that the proposed method obtained a significant improvement of +41.53% on zero-shot.

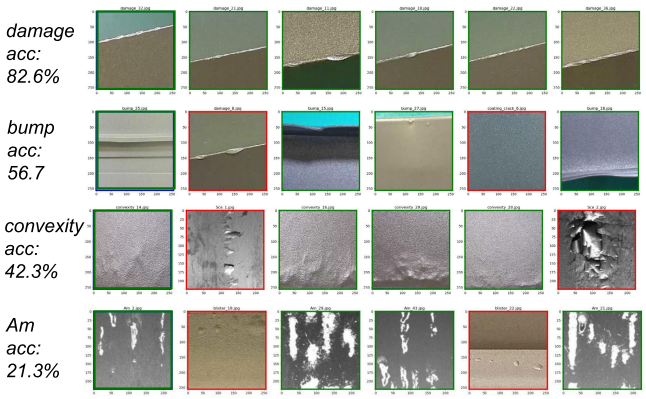

Fig. 3. Qualitative results of one-shot retrieval. Correct and incorrect retrieved instances are shown in green and red, respectively.

Furthermore, the qualitative result of one-shot retrieval is provided in Fig. 3. In Fig. 3, each row represents a class such as "Am," "bump," and "damage." Each cell in each row is the to-be-retrieval sample. The green frames denote the correct retrieval, while the wrong retrieval for red frames. The last row indicates steel surface defects, and the other four rows indicate aluminum surface defects. It can be seen relatively high acc is obtained for aluminum damage, bump, and convexity defect retrieval. However, relatively high acc is observed on aluminum defect retrieval. This is because of the unbalanced data distribution problem, very few steel samples

TABLE III
COMPARISON RESULT OF ZERO-SHOT VISUAL INSPECTION

Datasets	Metrics	CvcZSL [57]	ZSL_ABP [58]	CADA-VAE [59]	LisGAN [60]	CE-GZSL [22]	MSDN [21]	Ours
MSD-Cls	U	18.56%	18.75%	10.80%	3.14%	16.13%	18.22%	**28.29%**
	S	42.22%	24.63%	29.50%	18.15%	26.73%	32.18%	**51.09%**
	H	25.78%	21.29%	15.81%	5.36%	20.12%	23.26%	**36.42%**
FSC-20	U	12.40%	34.00%	9.30%	21.60%	11.60%	18.72%	**34.61%**
	S	69.73%	61.63%	59.91%	20.00%	59.20%	40.19%	**71.03%**
	H	20.41%	43.82%	16.10%	20.77%	19.40%	25.54%	**46.54%**
MT-CF	U	11.33%	25.00%	8.20%	32.27%	23.99%	35.02%	**51.85%**
	S	73.41%	47.89%	52.18%	54.94%	34.51%	41.22%	**76.85%**
	H	19.63%	32.85%	14.17%	40.00%	28.30%	37.87%	**61.92%**
EuroSAT	U	8.27%	21.71%	2.33%	17.25%	9.10%	29.28%	**35.49%**
	S	87.86%	36.14%	70.43%	55.58%	61.65%	61.02%	**89.62%**
	H	15.12%	27.13%	4.52%	22.92%	12.25%	39.57%	**50.85%**
RESISC45	U	1.50%	18.24%	1.17%	33.71%	41.35%	31.92%	**43.00%**
	S	73.39%	47.49%	75.27%	71.52%	69.48%	52.01%	**82.10%**
	H	2.94%	26.36%	2.30%	45.44%	51.85%	39.56%	**56.44%**
MED-3	U	8.02%	28.96%	9.00%	25.69%	24.85%	21.11%	**29.13%**
	S	**82.10%**	48.20%	60.44%	45.37%	49.14%	51.92%	64.29%
	H	14.61%	36.18%	15.67%	32.32%	32.77%	30.02%	**40.09%**
GTSRB	U	6.21%	13.91%	10.10%	17.47%	14.29%	21.01%	**32.63%**
	S	88.79%	80.11%	83.90%	42.78%	31.24%	41.79%	**91.65%**
	H	11.62%	23.70%	18.03%	23.77%	18.92%	27.96%	**48.13%**
DTD	U	1.82%	17.28%	1.45%	29.95%	30.52%	31.22%	**42.78%**
	S	**71.00%**	45.41%	63.11%	54.36%	46.70%	47.03%	59.72%
	H	3.55%	25.04%	2.84%	38.30%	36.84%	37.53%	**49.85%**

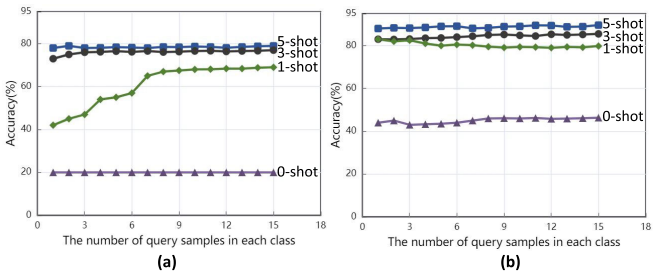

Fig. 4. Influence of the initial number of query samples q of our method and compares with previous SOTA method GTnet. (a) GTnet. (b) Ours.

in the training set, of the dataset that need to be considered in future research.

To evaluate the sensibility of the model on the initial number of query samples, 1–15 query samples are applied to the proposed method with the competitor GTnet, as shown in Fig. 4. It is observed that the proposed method always keeps stable no matter the initial number of query samples. On the contrary, GTnet requires a larger initial number (greater than 9) of query samples to get a fair performance. This demonstrates the proposed method is insensitive to the initial query samples. This is mainly because the proposed method uses synthesized samples to augment the query samples, which reduces the dependencies on the initial number of query samples.

2) Zero-Shot Inference Comparison: Table III provides the comparison results of zero-shot visual inspection. It is observed that our method significantly outperforms all competitors in all datasets. In highlights, on the H metric, our method achieves +10.64%, +21.92%, +11.28%, +20.17%, +11.55%, and +2.72% improvements in MSD-Cls, MT-CF, EuroSAT, GTSRB, DTD, and FSC-20, respectively, compared with the highest records. On the H metric, our method significantly surpasses all SOTA methods. This demonstrates that the proposed method can effectively balance the performance between unseen and seen visual inspection. On the U metric, our method obtains +9.54%, +16.83%, +6.21%, +1.65%, +0.17%, +11.62%, and +11.56% improvement over SOTA methods on the MSD-Cls, MT-CF, EuroSAT, RESISC45, MED-3, GTSRB, and DTD datasets, respectively. This denotes that the proposed method can synthesize "fake" features that are very similar to the real features, and the synthesized features can represent the real sample space distribution. On the S metric, our method obtains +8.87%, +3.44%, +1.76%, +6.83%, and +2.86% improvements in MSD-Cls, MT-CF, EuroSAT, RESISC45, and GTSRB datasets, respectively, while 17.81, and 11.28 decreases in MED-3, and DTD dataset. Interestingly, existing zero-shot methods have lower U scores than S in industry-specific data environments. This is because, in industry-specific data environments, there are not enough training samples for the existing zero-shot methods to learn a stable network for predicting unseen samples. On the contrary, instead of using augmented samples for model training, our method uses synthesized samples for model inference. This significantly decreases the requirements for the number of training samples.

To further reveal the performance of the method, three representative methods, LisGAN, CE-GZSL, and CvcZSL, are selected to construct the confusion heat maps, as shown in Fig. 5. The x-axis represents the predicted defect classes, while the y-axis refers to the real defect classes. The dark color represents the high probability given by the model for

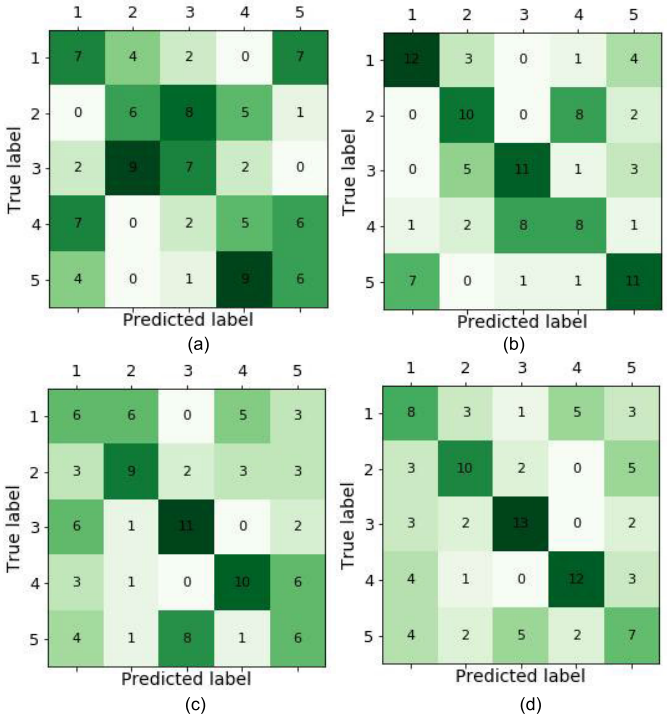

Fig. 5. Confusion heat maps of the representative methods. The x-axis represents the predicted defect classes, while the y-axis refers to the real defect classes. (a) LisGAN. (b) CE-GZSL. (c) CvcZSL. (d) Ours.

predicting the class label. The more dark colors close to the diagonal of the heat map, the more accurate the model is.

Obviously, the color distribution of Fig. 5(a) is chaotic which means the corresponding method fails to conduct the task. Fig. 5(b) and (c) has similar color distributions that are close to the diagonal of the heat map. However, there are still many dark colors that deviate from the diagonal of the heat map which represents the wrong predictions. Fig. 5(d) has the clearest color distribution that is close to the diagonal. It has the best prediction performance. This further demonstrates the comprehensive superiority of the proposed method on unseen defect prediction compared with the existing methods.

C. Discussions

1) Ablation Study: The proposed method consists of RFFC, SAGE, and CCO modules (see Section III-B). To ensure a fair comparison and more clearly demonstrate the performance of the proposed module, the baseline combines a traditional few-shot learning model and a traditional generative zero-shot learning model, similar to S2M2_R [51] and GAZSL [61]. The result of the ablation study is provided in Table IV. Interestingly, the addition of the synthesized feature contrast module (RFFC) resulted in a significant improvement (+13.67%, +5.2%, and +3.04%) in accuracy on fewer shots (zero-shot, one-shot, and five-shot). This indicates the RFFC can effectively generate unseen features for few-/zero-shot predicting.

Furthermore, compared with the baseline using the SAGE module alone, the accuracy was optimized on different shots (+1.31%, +3.07%, and +1.47%). This demonstrates that the SAGE module can optimize model performance through

TABLE IV
ABLATION STUDY

Baseline	RFFC	SAGE	CCO	5-way 0-shot	5-way 1-shot	5-way 5-shot
√				23.62%	62.35%	76.92%
√	√			37.29%	67.55%	79.33%
√		√		24.93%	65.39%	78.39%
√			√	26.44%	71.62%	79.42%
√	√	√	√	**46.33%**	**79.81%**	**89.59%**

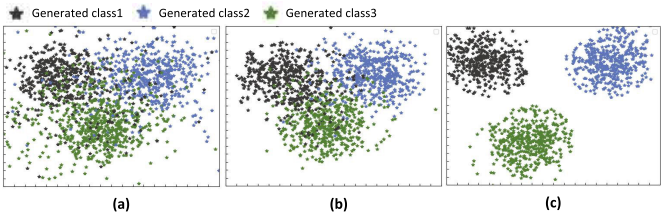

Fig. 6. Comparison of RFFC module effects. The visual analysis for different classes of features synthesized by different zero-shot learning models, with different colors representing that the synthesized features belong to different classes. (a) GAZSL. (b) CE-GZSL. (c) Proposed.

sample information fusion. Using the combination of RFFC, SAGE, and CCO modules compared with the RFFC module alone, the accuracy was significantly improved on different shots (+9.04%, +12.26%, and +10.26%). This shows that SAGE as a feature preprocessing for the CCO module and the combination of the two is more outstanding.

As shown in Fig. 6, we visualize the different classes of features generated by the proposed model (contrastive generator RFFC module), with different colors representing different classes of the MSD-Cls dataset. It can be found that, compared with GAZSL (the milestone model for zero-shot learning) and CE-GZSL (the representative model for zero-shot learning), after the first embedding space optimization based on contrast learning, our method has a significant distance between different classes of synthesized features, a clear boundary between the generated different classes, and a significant decrease of biased samples.

2) Hyperprameters Discussion: The hyperparameters used in our method are k, θ, and λ. For the maximum similarity retention k [$k \geq 1$, see (4) and (10)] and the embedding graph ratio θ [see (5)], the metric evaluated is the accuracy of one-shot [Fig. 7(a)] and five-shot [Fig. 7(b)] classifications. The dataset for the evaluations is MDS-Cls.

It is observed from Fig. 7(a) that the accuracy decreases when θ increases and the maximum accuracy is obtained when $\theta = 1$. The accuracy first increases when k increases from 2 to 4 but then gradually decreases when $k \geq 6$. Thus, all things considered, for θ and k, the best settings at one-shot are $\theta = 1$ and $k = 6$. Similarly, according to Fig. 7(b), for θ and k, the best settings at five-shot are $\theta = 1$ and $k = 4$.

For the regularization parameter λ [see (8)], one- to five-shot classification experiments are conducted. As shown in Fig. 7(c), the accuracy increases quickly when λ starts from 0 to 5. After, the accuracy of one-shot, three-shot, and five-shot classifications tend to be stable when λ continuously

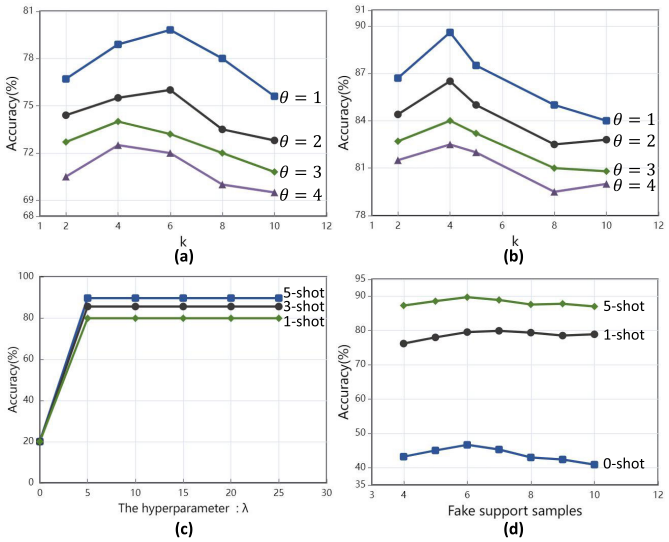

Fig. 7. Influence of hyperparameters. (a) 1-shot classification accuracies from different hyperparameter values (k and θ). (b) 5-shot classification accuracies from different hyperparameter values (k and θ). (c) Graph of classification accuracy versus hyperparameter λ. (d) Graph of classification accuracy versus number of fake support samples.

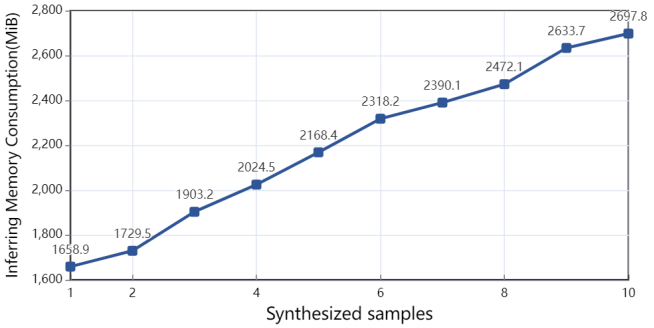

Fig. 8. Memory consumption of the inference process with different synthesized support samples.

increases. Thus, the reasonable setting of the regularization parameter is $\lambda = 5$.

Since the proposed method generates samples from the RFFC module as synthesized support samples for inference on the query samples category, this leads to a different number of synthesized samples with different accuracy rates. As shown in Fig. 7(d), in zero-shot condition, the model performance is optimal when the number of synthesized samples $= 6$, and then it gradually decreases, which may be because some biased synthesized samples cause the inference process to be misguided, leading to the decrease of the accuracy rate.

Similarly, the model performance is optimal in the one- and five-shot conditions when the number of synthesized samples is 7 and 6.

The memory consumption of the model's inference process on the MSD-Cls dataset with different numbers of synthesized support samples is shown in Fig. 8. A clear trend is that the larger the number of synthesized support samples, the larger the memory consumption of the model inference process, while the query samples remain constant.

In summary, by weighing the relationship between model size and accuracy, the number of synthesized support samples of the proposed method is uniformly set to 6 in practical deployment. To ensure that the model obtains the maximum performance while consuming as little memory as possible.

TABLE V
EVALUATION OF INFERENCE TIME WITH DIFFERENT SUPPORT SAMPLES

Method	5-way-0-shot	5-way-1-shot	5-way-5-shot
S2M2_R [51]	-	0.7344	2.1298
ICI-FSL [52]	-	1.2703	1.4182
PTNET [53]	-	3.7299	4.1307
Latent [54]	-	2.8341	4.3862
TRA [55]	-	14.5392	16.0262
GTnet [15]	-	6.0329	11.8432
fsl-rsvae [56]	-	7.1127	10.4728
FaNet [13]	-	5.1127	9.7328
Ours(Filtering + Inference)	4.4637	8.1528	12.7347
Ours(Inference)	4.4637	5.4831	9.8305

TABLE VI
PERFORMANCE EVALUATION OF DIFFERENT BACKBONE NETWORKS

Method	5-way-0-shot	5-way-1-shot	5-way-5-shot
WRN-28–10	**46.69 ± 0.36**	**78.92 ± 0.31**	88.74 ± 0.11
ResNet-12	45.92 ± 0.44	78.02 ± 0.82	87.25 ± 0.27
ResNet-18	43.01 ± 0.08	77.06 ± 0.66	**89.91 ± 0.11**

3) Inference Time Evaluation: To further understand the performance of the proposed method, the model inference time is evaluated on the MSD-Cls dataset, as shown in Table V. Since the filtering phase of the few-shot inference process can be completed before inference about the query sample categories, the inference process of the proposed method is decoupled into two phases (filtering + inference). The filtering and inference time before decoupling and the inference time without the filtering phase are validated here, respectively.

A clear trend is that the inference time becomes longer as the support sample increases. Meanwhile, the inference time of the proposed method alone is much smaller than the time of both inference and filtering phases. Therefore, filtering operations are performed before model deployment to improve real-time prediction.

4) Backbone Network Discussion: In order to evaluate the impact of different backbone networks on the proposed method, it is evaluated on the MSD-Cls dataset using several mainstream backbone networks (i.e., WRN-28-10, ResNet-12, and ResNet-18). The test results are shown in Table VI, where the accuracy of the proposed method is 46.69% and 78.92% on WRN-28-10 for shot $= 0$ and shot $= 1$. The highest accuracy is achieved on ResNet-18 for shot $= 5$. Overall, WRN-28-10 is more suitable to handle the MSD-Cls dataset with fewer sample sizes.

D. Prototype Scenario

1) Hot-Rolled Steel Sample Collection: In order to evaluate the performance of the method in practical applications, we collected 15 types of hot-rolled steel surface defects from our partner manufacturers. Some of the defect samples are shown in Fig. 9(a), and the types of defects include contaminants (Co), inclusions (In), scratches (Sc), oxides (Ox), and so on. This included 150 defect samples (10 for each defect class) and 210 normal samples, and a hot-rolled steel surface

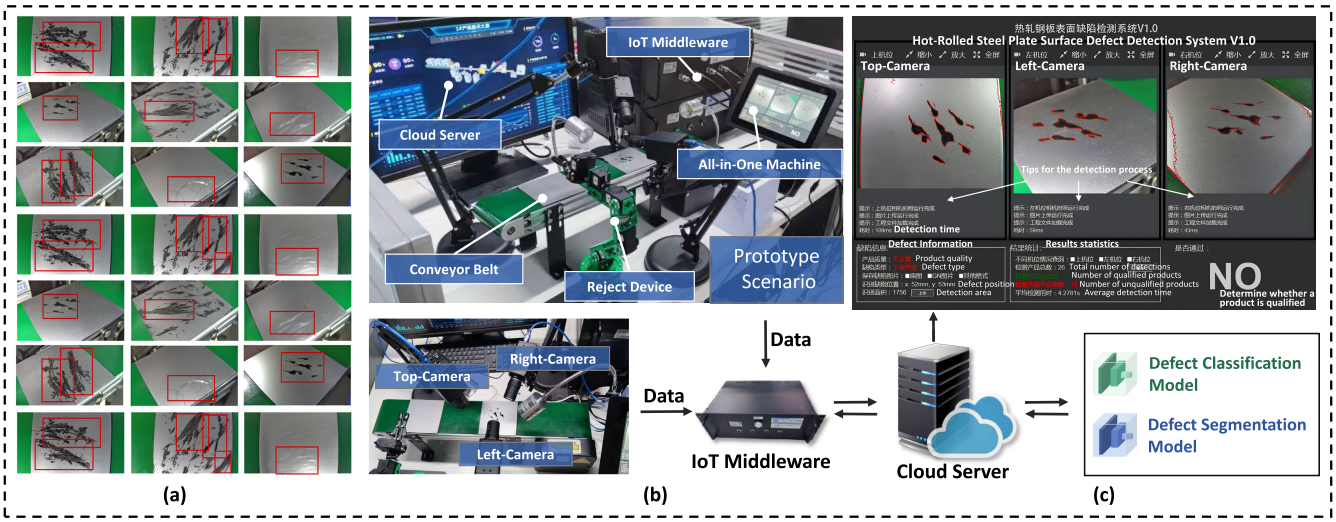

Fig. 9. Prototype manufacturing scenario for hot-rolled steel defect classification based on the proposed method. (a) Hot-rolled steel defect samples. (b) Production environment. (c) Cloud server results in feedback.

defect dataset (HRS-SD) was created. Ten defect classes and their samples and all normal samples are split as the training set. The five defect classes (50 defect samples) are split into a test set for a few-/zero-shot defect classification.

2) Prototype Scene Building: A prototype manufacturing defect detection scenario was established to evaluate the model's performance in a realistic scenario. The prototype scenario consists of a production environment, an IoT middleware (our previous work [65]), and a cloud server (Huawei kAi1s accelerated cloud server).

The production environment is shown in Fig. 9(b), where three industrial cameras with different angles (top-camera, left-camera, and right-camera) were used to parallel obtain defect samples of hot-rolled steel on the conveyor belt (running at 10 m/s and the length of 600 mm), with the top-camera at 230 mm distance from the samples, and cameras with resolution dimensions of 2594 $\times$ 1944 pixels. The images were resized to 64 $\times$ 64 pixels to be passed to the server to improve the speed of the model run.

Considering that in the realistic application environment, multiple production lines may be monitored in real-time with multiple cameras, which will lead to a large number of product images being captured at the same time, it is not easy to expand the devices and transmit image data quickly by connecting the cameras directly to the server. Thus, to integrate a large amount of image data quickly, the obtained image data and the control signals of other devices (e.g., reject devices) are integrated into a cloud server via the IoT middleware. The reject device uses a programmable vision robot arm with five degrees of freedom and a vision resolution of 640 $\times$ 480, and the microprocessor is a Quad-core ARM A57 + 128-core NVIDIA Maxwell.

The proposed method is deployed to perform defect detection response (controlling the reject device and conveyor belt) and result visualization (displaying on an all-in-one machine) on a cloud server with a Kunpeng 920 2.6 GHz processor.

It is worth mentioning that to perform real-time defect detection, the proposed method is decoupled into three phases in the deployment, including feature generation based on contrast learning (see Section III-B), support sample filtering (see Section III-C2), and defect class inference (see Section III-C1). Only the inference phase is deployed on the server, and the feature generation and sample filtering phases are preprocessed before deployment. First, the synthesized features of seen/unseen classes are generated using the class prompts provided by the experts. Then, the generated seen class features are filtered. Finally, the unseen class synthesized features, the filtered seen class synthesized features, and the real support features are combined to form a feature support library for use in the defect class inference stage.

TABLE VII
PROTOTYPE SCENARIO EXPERIMENTAL RESULTS

K-shot	Recognition	Send	Receive	Average Accuracy
0-shot	3600ms	1200ms	150ms	72.30%
1-shot	5700ms	1500ms	150ms	91.30%
5-shot	8200ms	2100ms	150ms	100.00%

In practical applications, to realize defect detection (including classification and segmentation), a defect segmentation model with segmentation and object detection functions is introduced, which crops out the detected defects and then passes them to our proposed classification model. More accurate defect locations and smaller image sizes help improve the proposed method's accuracy and speed. The defect segmentation model segments the image when the proposed method finishes classifying the defects. The final visualization is shown in Fig. 9(c).

3) Evaluation Results: For each experiment, we repeated ten times to obtain the average value. Combining the parameter analyses and inference time evaluations from Section IV-C, all experimental parameters were fixed to $\theta = 1$, $k = 4$, $\lambda = 5$, and the number of synthesized support samples was four by weighing model size, classification accuracy, and run time.

The experimental results are shown in Table VII, which includes the average recognition time for K-shot classification,

the time to send visual features on the network, and the time to receive the results. Based on the experiment results, it can be observed that the proposed method achieves an average accuracy of 100% for five-shot classification. This result indicates that for the classification of hot-rolled steel surface defects in real manufacturing, the model performance of our method is relatively high for both zero- and few-shot. The time cost of classification is also acceptable for manufacturing applications.

This work will be further collaborated with Metallurgical Research Institute Company Ltd., and promoted in industrial production lines.

V. Conclusion

In the field of surface defect recognition, our work focuses on solving three problems: lack of training samples and model complexity in deep-learning-based methods, recognition of defect types limited to known classes in few-shot learning-based methods, and inability to tradeoff attention to seen and unseen classes in zero-shot learning-based methods. A novel few-/zero-shot compatible surface defect classification method is proposed. Extensive experiments on eight fine-grained datasets show that our method improves by an average of 8.29% on the few-shot recognition task and 8.23% on the zero-shot recognition task compared to SOTA methods. The prototype scenario evaluation demonstrates that the proposed method can recognize defect types in real-time. Meanwhile, the average accuracy of the five-shot classification of hot-rolled steel defects reaches 100%, proving the adaptability of the proposed method in industrial environments.

The limitations of this method are as follows. 1) Compared with the existing zero-shot learning methods, the accuracy of the proposed method has significantly improved, but it has not yet reached the expected accuracy for industrial applications. The next step will explore associating the seen class sample information with the unseen class and further optimizing the recognition of the unseen class using the few-shot learning idea. 2) Through experiments, it is found that although the classification performance of the proposed method outperforms the methods based on few-shot learning on surface defect datasets (i.e., MSD-Cls, FSC-20, and MT-CF), the model size and inference time are not optimal. The next step is introducing model compression methods, such as knowledge distillation, to make the model more adaptable to real-time industrial production environments.

Acknowledgment

The authors would like to thank China Copper (Kunming) Company Ltd., Kunming, China, for providing the manufacturing equipment used in this work. They also thank Kunming Metallurgical Research Institute Company Ltd., Kunming, for providing hot-rolled steel samples and expert descriptions of unseen defects.

References

[1] X. Guidetti, A. Rupenyan, L. Fassl, M. Nabavi, and J. Lygeros, "Advanced manufacturing configuration by sample-efficient batch Bayesian optimization," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 11886–11893, Oct. 2022, doi: 10.1109/LRA.2022.3208370.

[2] Q. Xu et al., "RECCraft system: Towards reliable and efficient collective robotic construction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 8979–8986, doi: 10.1109/IROS47612.2022.9982068.

[3] A. Salazar-Gomez, M. Darbyshire, J. Gao, E. I. Sklar, and S. Parsons, "Beyond mAP: Towards practical object detection for weed spraying in precision agriculture," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 9232–9238, doi: 10.1109/IROS47612.2022.9982139.

[4] N. Hanson, M. Shaham, D. Erdomuş, and T. Padir, "VAST: Visual and spectral terrain classification in unstructured multi-class environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Kyoto, Japan, Oct. 2022, pp. 3956–3963, doi: 10.1109/IROS47612.2022.9982078.

[5] V. Sampath, I. Maurtua, J. J. A. Martín, A. Rivera, J. Molina, and A. Gutierrez, "Attention-guided multitask learning for surface defect identification," *IEEE Trans. Ind. Informat.*, vol. 19, no. 9, pp. 9713–9721, Sep. 2023, doi: 10.1109/TII.2023.3234030.

[6] Z. Jiang et al., "A deep convolutional network combining layerwise images and defect parameter vectors for laser powder bed fusion process anomalies classification," *J. Intell. Manuf.*, vol. 37, pp. 1–31, Aug. 2023, doi: 10.1007/s10845-023-02183-4.

[7] X. Yu, L. Han-Xiong, and H. Yang, "Collaborative learning classification model for PCBs defect detection against image and label uncertainty," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–8, 2023, doi: 10.1109/TIM.2023.3235461.

[8] S. Xiao, Z. Liu, Z. Yan, and M. Wang, "Grad-MobileNet: A gradient-based unsupervised learning method for laser welding surface defect classification," *Sensors*, vol. 23, no. 9, p. 4563, May 2023.

[9] T. Zhang et al., "Automatic detection of surface defects based on deep random chains," *Expert Syst. Appl.*, vol. 229, Nov. 2023, Art. no. 120472, doi: 10.1016/j.eswa.2023.120472.

[10] C. Zhou, M. Liu, S. Zhang, P. Wei, and B. Chen, "Few-shot classification of screen defects with class-agnostic mask and context-based classifier," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–16, 2023, doi: 10.1109/TIM.2023.3280532.

[11] Z. Liu, Y. Song, R. Tang, G. Duan, and J. Tan, "Few-shot defect recognition of metal surfaces via attention-embedding and self-supervised learning," *J. Intell. Manuf.*, vol. 34, no. 8, pp. 3507–3521, Dec. 2023.

[12] J. Zhao et al., "A knowledge distillation-based multi-scale relation-prototypical network for cross-domain few-shot defect classification," *J. Intell. Manuf.*, vol. 35, pp. 841–857, Feb. 2023, doi: 10.1007/s10845-023-02080-w.

[13] W. Zhao, K. Song, Y. Wang, S. Liang, and Y. Yan, "FaNet: Feature-aware network for few shot classification of strip steel surface defects," *Measurement*, vol. 208, Feb. 2023, Art. no. 112446.

[14] Y. Song, Z. Liu, S. Ling, R. Tang, G. Duan, and J. Tan, "Coarse-to-fine few-shot defect recognition with dynamic weighting and joint metric," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022, doi: 10.1109/TIM.2022.3193204.

[15] 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, doi: 10.1109/TIM.2022.3169547.

[16] Z. Li, L. Gao, Y. Gao, X. Li, and H. Li, "Zero-shot surface defect recognition with class knowledge graph," *Adv. Eng. Informat.*, vol. 54, Oct. 2022, Art. no. 101813.

[17] A. M. A. N. Abdo and L. Czúni, "Zero-shot learning and classification of steel surface defects," in *Proc. 14th Int. Conf. Mach. Vis. (ICMV)*, Mar. 2022, pp. 386–394.

[18] X. Sun, J. Gu, M. Wang, Y. Meng, and H. Shi, "Wheel hub defects image recognition base on zero-shot learning," *Appl. Sci.*, vol. 11, no. 4, p. 1529, Feb. 2021.

[19] Z. Jia, Z. Zhang, C. Shan, L. Wang, and T. Tan, "Dual-focus transfer network for zero-shot learning," *Neurocomputing*, vol. 541, Jul. 2023, Art. no. 126264.

[20] J. Huang, Z. Li, and Z. Zhou, "A simple framework to generalized zero-shot learning for fault diagnosis of industrial processes," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 6, pp. 1–3, Jun. 2023, doi: 10.1109/JAS.2023.123426.

[21] S. Chen et al., "MSDN: Mutually semantic distillation network for zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7602–7611.

[22] Z. Han, Z. Fu, S. Chen, and J. Yang, "Contrastive embedding for generalized zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2371–2381.

[23] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006, doi: 10.1109/TPAMI.2006.79.

[24] X. Han et al., "FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1–7.

[25] Y. Wang and D. V. Anderson, "Hybrid attention-based prototypical networks for few-shot sound classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 651–655, doi: 10.1109/ICASSP43922.2022.9746118.

[26] T. Gao et al., "FewRel 2.0: Towards more challenging few-shot relation classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1–6.

[27] R. Geng, B. Li, Y. Li, J. Sun, and X. Zhu, "Dynamic memory induction networks for few-shot text classification," 2020, empharXiv:2005.05727.

[28] X. Geng, X. Chen, and K. Q. Zhu, "MICK: A meta-learning framework for few-shot relation classification with little training data," 2020, *arXiv:2004.14164*.

[29] J. Han, B. Cheng, Z. Wan, and W. Lu, "Towards hard few-shot relation classification," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 1–14, Sep. 2023, doi: 10.1109/TKDE.2023.3240851.

[30] J. Wang, Y. Gao, and Z. Fang, "An angular shrinkage BERT model for few-shot relation extraction with none-of-the-above detection," *Pattern Recognit. Lett.*, vol. 166, pp. 151–158, Feb. 2023.

[31] Q. Huang, F. Zhang, and X. Li, "Few-shot decision tree for diagnosis of ultrasound breast tumor using BI-RADS features," *Multimedia Tools Appl.*, vol. 77, no. 22, pp. 29905–29918, Nov. 2018, doi: 10.1007/s11042-018-6026-1.

[32] S. Jadon, "COVID-19 detection from scarce chest X-ray image data using few-shot deep learning approach," *Proc. SPIE*, vol. 11601, Feb. 2021, Art. no. 116010X.

[33] R. Walsh, M. H. Abdelpakey, M. S. Shehata, and M. M. Mohamed, "Automated human cell classification in sparse datasets using few-shot learning," *Sci. Rep.*, vol. 12, no. 1, p. 2924, Feb. 2022, doi: 10.1038/s41598-022-06718-2.

[34] N. Zhang, M. Ruan, S. Wang, L. Paul, and X. Li, "Discriminative few shot learning of facial dynamics in interview videos for autism trait classification," *IEEE Trans. Affect. Comput.*, vol. 14, no. 2, pp. 1110–1124, Apr./Jun. 2022, doi: 10.1109/TAFFC.2022.3178946.

[35] Moxuan, Y. QianG, Z. Xiyi, C. Juan, and W. Quan, "Contrastive representation for dermoscopy image few-shot classification," in *Proc. 17th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, Dec. 2020, pp. 134–137, doi: 10.1109/ICCWAMTIP51612.2020.9317490.

[36] Q.-H. Nguyen, C. Q. Nguyen, D. D. Le, and H. H. Pham, "Enhancing few-shot image classification with cosine transformer," *IEEE Access*, vol. 11, pp. 79659–79672, 2023, doi: 10.1109/ACCESS.2023.3298299.

[37] M. Palatucci, D. Pomerleau, G. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran, 2009, pp. 1410–1418.

[38] A. Li, Z. Lu, L. Wang, T. Xiang, and J.-R. Wen, "Zero-shot scene classification for high spatial resolution remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 4157–4167, Jul. 2017, doi: 10.1109/TGRS.2017.2689071.

[39] Y. Li, D. Kong, Y. Zhang, Y. Tan, and L. Chen, "Robust deep alignment network with remote sensing knowledge graph for zero-shot and generalized zero-shot remote sensing image scene classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 179, pp. 145–158, Sep. 2021.

[40] T. Toizumi, K. Sagi, and Y. Senda, "Automatic association between SAR and optical images based on zero-shot learning," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2018, pp. 17–20, doi: 10.1109/IGARSS.2018.8517299.

[41] W. Xu, J. Wang, Z. Wei, M. Peng, and Y. Wu, "Deep semantic-visual alignment for zero-shot remote sensing image scene classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 198, pp. 140–152, Apr. 2023, doi: 10.1016/j.isprsjprs.2023.02.012.

[42] J. Wei, Y. Cao, K. Yang, L. Chen, and Y. Wu, "Self-supervised remote sensing image dehazing network based on zero-shot learning," *Remote Sens.*, vol. 15, no. 11, p. 2732, May 2023, doi: 10.3390/rs15112732.

[43] Z. Cha, D. Xu, Y. Tang, and Z. Jiang, "Meta-learning for zero-shot remote sensing image super-resolution," *Mathematics*, vol. 11, no. 7, p. 1653, Mar. 2023.

[44] P. Pan, Y. Li, and D. Zhao, "Generalized zero-shot learning fault diagnosis framework based on anomaly detection and contractive stacked autoencoder," in *Proc. China Autom. Congr. (CAC)*, Nov. 2022, pp. 2427–2432, doi: 10.1109/CAC57257.2022.10055111.

[45] Z. Hu, H. Zhao, L. Yao, and J. Peng, "Semantic-consistent embedding for zero-shot fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 19, no. 5, pp. 7022–7031, May 2023, doi: 10.1109/TII.2022.3210215.

[46] B. Li and C. Zhao, "Federated zero-shot industrial fault diagnosis with cloud-shared semantic knowledge base," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11619–11630, Jul. 2023, doi: 10.1109/JIOT.2023.3243401.

[47] Z. Y. Ding, J. Y. Loo, S. G. Nurzaman, C. P. Tan, and V. M. Baskaran, "A zero-shot soft sensor modeling approach using adversarial learning for robustness against sensor fault," *IEEE Trans. Ind. Informat.*, vol. 19, no. 4, pp. 5891–5901, Apr. 2023, doi: 10.1109/TII.2022.3187708.

[48] H. Y. Zhou, "A new kind of based on the graph K-Means clustering initial center selection algorithm," *Appl. Mech. Mater.*, vols. 241–244, pp. 2845–2848, Dec. 2012, doi: 10.4028/www.scientific.net/AMM.241-244.2845.

[49] A. Hassani, A. Iranmanesh, M. Eftekhari, and A. Salemi, "DISCERN: Diversity-based selection of centroids for k-estimation and rapid non-stochastic clustering," *Int. J. Mach. Learn. Cybern.*, vol. 12, pp. 635–649, Mar. 2019.

[50] W. Tong, Y. Wang, and D. Liu, "An adaptive clustering algorithm based on local-density peaks for imbalanced data without parameters," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3419–3432, Apr. 2023, doi: 10.1109/TKDE.2021.3138962.

[51] P. Mangla, M. Singh, A. Sinha, N. Kumari, V. N. Balasubramanian, and B. Krishnamurthy, "Charting the right manifold: Manifold mixup for few-shot learning," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 2207–2216.

[52] Y. Wang, C. Xu, C. Liu, L. Zhang, and Y. Fu, "Instance credibility inference for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12833–12842.

[53] Y. Hu, V. Gripon, and S. Pateux, "Leveraging the feature distribution in transfer-based few-shot learning," in *Proc. Int. Conf. Artif. Neural Netw.*, 2020, pp. 487–499.

[54] T. Chobola, D. Vasata, and P. Kordík, "Transfer learning based few-shot classification using optimal transport mapping from preprocessed latent space of backbone neural network," 2021, *arXiv:2102.05176*.

[55] Y. Hu, V. Gripon, and S. Pateux, "Exploiting unsupervised inputs for accurate few-shot classification," 2020, *arXiv:2001.09849*.

[56] J. Xu and H. Le, "Generating representative samples for few-shot classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8993–9003.

[57] K. Li, M. R. Min, and Y. Fu, "Rethinking zero-shot learning: A conditional visual classification perspective," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3582–3591.

[58] Y. Zhu, J. Xie, B. Liu, and A. Elgammal, "Learning feature-to-feature translator by alternating back-propagation for generative zero-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9843–9853.

[59] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero- and few-shot learning via aligned variational autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8239–8247.

[60] J. Li, M. Jing, K. Lu, Z. Ding, L. Zhu, and Z. Huang, "Leveraging the invariant side of generative zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7394–7403.

[61] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1004–1013.

[62] 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.

[63] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–12.

[64] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 8748–8763.

[65] C. Xie, B. Yu, Z. Zeng, Y. Yang, and Q. Liu, "Multilayer Internet-of-Things middleware based on knowledge graph," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2635–2648, Feb. 2021.

Yuran Dong received the B.S. degree in software engineering from the Southwest University of Nationalities, Chengdu, China, in 2021. She is currently pursuing the master's degree with the School of Software, Yunnan University, Kunming, China.

Her current research interests include Few-shot learning and Zero-shot learning.

Cheng Xie (Member, IEEE) received the B.S. degree in software engineering from the Minzu University of China, Beijing, China, in 2009, and the M.S. and Ph.D. degrees in software engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012 and 2017, respectively.

From 2015 to 2016, he was a Visiting Scholar with the Data and Web Science Group, University of Mannheim, Mannheim, Germany. He is currently an Associate Professor and a Professor with the School of Software, Yunnan University, Kunming, China. His research interests include industrial informatics and knowledge graph.

Dr. Xie is a recipient of the Shanghai Science and Technology Progress Award (Second Prize) and Yunnan Science Award (Second Prize) in 2014 and 2022, respectively. The Youth Talent Project of China Association for Science and Technology and the High-level Talent Program of Yunnan in 2018 and Donglu Young Scholars in 2021.

Luyao Xu received the B.S. degree in electrical engineering and automation from the Hefei University of Economics, Hefei, China, in 2020. He is currently pursuing the master's degree with the School of Software, Yunnan University, Kunming, China.

His current research interests include deep learning and surface defect detection.

Hongming Cai (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Northwestern Polytechnical University, Xi'an, China, in 1996, 1999, and 2002, respectively.

He was a Postdoctoral Research Fellow with the Computer Science and Technology Department, Shanghai Jiao Tong University, Shanghai, China, during 2002 to 2004. He was a Visiting Professor with the Business Information Technology Institute, University of Mannheim, Germany, during 2008 to 2009. He is currently a Professor with the School of Software, Shanghai Jiao Tong University.

Dr. Cai is a standing Director of China Graphics Society, and a Senior Member of ACM and China Computer Federation. The visiting scholarship was appointed and sponsored by Alfried Krupp von Bohlen und Halbach Foundation, Germany. He is rewarded as a "National outstanding scientific and technological workers" by China Association for Science and Technology in 2012.

Weiming Shen (Fellow, IEEE) received the B.Sc. and M.S. degrees in mechanical engineering from Beijing Jiaotong University, Beijing, China, in 1983 and 1986, and the Ph.D. degree in artificial intelligence in systems control from the Université Technique de Compiègne, France, in 1996.

He is a Senior Research Scientist with the National Research Council Canada, Ottawa, ON, Canada, and an Adjunct Professor with Tongji University, Shanghai, China, and the University of Western Ontario, London, ON, Canada. He is currently a Professor with the School of Software, Huazhong University of Science and Technology, Wuhan, China. He has been invited to provide over 80 invited lectures/seminars at different academic and research institutions over the world and keynote presentations/tutorials at various international conferences. He has published several books and over 450 papers in scientific journals and conferences. His work has been cited over 10 000 times with an H-index of 47. His current research interests include agent-based collaboration technology and applications, Internet of Things, and big data analytics.

Haoyuan Tang received the master's degree from the Kunming University of Technology, Kunming, China in 2008, focusing on the research and development of Aluminum and Aluminum Alloys, Copper and Copper Alloys. He is currently a Senior Engineer with the Kunming Institute of Metallurgy, China. He has presided over and participated in a number of major science and technology projects in Yunnan Province, presided over the key new product development special projects in Yunnan Province, "grain refiner aluminum titanium boron wire new product development", major science and technology special projects in Yunnan Province, "Yunnan Province, rare and precious metal materials genetic engineering (Phase I 2021) - copper materials, special database research, development and construction and engineering demonstration". Key technology research and other projects. He has published more than 10 papers and applied for more than 10 patents.