Flexible and Diverse Contrastive Learning for Steel Surface Defect Recognition With Few Labeled Samples

Ying Xu, Jiacong Chen, Yanyang Liang, Yikui Zhai, *Member, IEEE*, Zilu Ying, Wenlve Zhou, Angelo Genovese, *Member, IEEE*, Vincenzo Piuri, *Fellow, IEEE*, and Fabio Scotti, *Senior Member, IEEE*

Abstract*— As samples of steel defects are industrially limited, it is challenging for most deep learning methods that rely on ample labeled data to identify steel surface defects. Recently, contrastive learning has achieved good performance in natural image classification tasks with few labeled samples, yet two obstacles prevent its effective application to steel surface defect images. One is that due to the presence of inter-class and intra-class similar samples in steel surface defect, the fixed contrast strength in contrastive learning will destroy the potential semantic information of defect samples. Another is that contrastive learning requires a large amount of unlabeled data, whereas steel surface defect samples are insufficient. To overcome the above-mentioned problems, a novel framework named flexible and diverse contrastive learning (FDCL) is proposed. This framework consists of two parts, flexible contrast (FiCo) and diverse generative adversarial networks (DGANs). Diverse images generated by the DGAN and real images are fed into FiCo for representation learning. In the FiCo, the contrast strength among samples is flexibly adjusted by the proposed variable temperature discrimination and feature reconstruction (FR). In addition, the output features (OF) of FiCo will be used as input to the DGAN generator to improve image quality, thus further facilitating representation learning. The proposed FDCL is implemented on four standard steel surface defect data sets, and the experimental results demonstrated that it achieves superior performance over state-of-the-art methods. Our code is available at: https://github.com-/jiacongc/FDCL.

***Index Terms*— Contrastive learning, diverse generative adversarial network (DGAN), feature reconstruction (FR), flexible contrast (FiCo), steel surface defect recognition, variable temperature discrimination.**

Manuscript received 31 December 2022; revised 25 January 2023; accepted 5 February 2023. Date of current version 9 June 2023. This work was supported in part by the Key Research Projects for Universities of Guangdong Provincial Education Department under Grant 2020ZDZX3031; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515010716, Grant 2021A1515011576, and Grant 2017KCXTD015; in part by Guangdong, Hong Kong, Macau, and the Greater Bay Area International Science and Technology Innovation Cooperation Project under Grant 2021A050530080 and Grant 2021A0505060011; and in part by Jiangmen City Basic and Applied Basic Research Foundation under Grant 2021030103230006670. The Associate Editor coordinating the review process was Dr. Qiang Miao. *(Corresponding author: Yanyang Liang.)*

Ying Xu, Jiacong Chen, Yanyang Liang, Yikui Zhai, Zilu Ying, and Wenlve Zhou are with the Department of Intelligent Manufacturing, Wuyi University, Jiangmen 529020, China (e-mail: xuying117@163.com; jiacongchen7@163.com; liangyanyang@163.com; yikuizhai@163.com; ziluy@163.com; wenlvezhou@163.com).

Angelo Genovese, Vincenzo Piuri, and Fabio Scotti are with the Depatimento di Information, Università degli Studi di Milano, 20133 Milan, Italy (e-mail: angelo.genovess@unimi.it; vincenzo.piuri@unimi.it; fabio.scotii@unimi.it).

Digital Object Identifier 10.1109/TIM.2023.3249221

I. Introduction

STEEL strip is one of the major products of the iron and steel industry. Any surface defects will threaten the steel product quality, which can pose significant economic and reputational risks to steel manufacturers. Surface defect inspection is mainly processed through automated visual inspection (AVI) instruments [1], [2], [3]. One of the main functions of AVI, defect recognition, plays a key role in ensuring the quality of steel.

Previously, automatic defect recognition focused on traditional image processing methods [4], [5], [6]. These methods require manual design features and clean background. Once the background becomes complex, these methods will not work well. Fortunately, deep learning has a powerful automatic feature extraction capability. It has been used with good results in defect recognition in complex backgrounds in recent years [7], [8], [9]. However, these deep learning methods can only achieve high accuracy if they possess lots of data. In some actual cases, it is difficult to obtain images of steel surface defects as they rarely appear on the steel surface and have to be captured by the camera, which is time-consuming and complex. Once the defect images have been collected, it takes experienced personnel much time to screen and label them. Therefore, the steel surface defect data are always insufficient, and the labeled samples are even few. The overfitting problem will be triggered by few labeled samples due to the numerous parameters of convolutional neural networks (CNNs), which will lead to the degradation of recognition accuracy.

Based on the problem of few labeled samples, researchers are committed to exploring different methods [10], [11], [12], [13], [14], [15]. Among them, contrastive learning shows the powerful performance. It digs out supervision information from a huge amount of unlabeled data to learn good representations and uses them for classification tasks, which can significantly enhance the performance of recognition with only few labeled samples. The supervision information is mined in three main steps. First, a pair of samples is generated for each instance, with samples from the same instance considered positive and samples from different instances considered negative. Second, these samples are fed to the encoder to obtain the embeddings. Third, under the influence of contrastive loss [16], the embeddings of positive samples are pulled

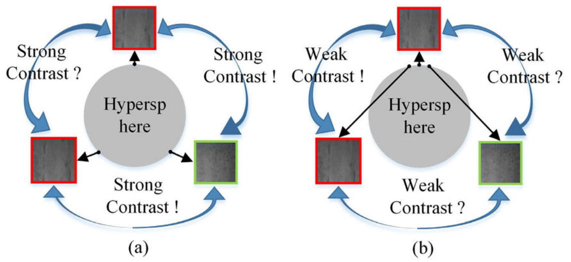


Fig. 1. (a) Strong and (b) weak contrastive embedding distributions with three instances of steel surface defects on a hypersphere. The boxes of the same color indicate intra-class similar samples, and the boxes of different colors indicate inter-class similar samples. The black dots on the hypersphere represent the embedding positions of the instance samples.

together to maximize their similarity, while the embeddings of negative samples are pushed apart to maximize their differences. These three steps guide the model to learn useful representations.

In spite of the achievement of contrastive learning in natural images, there are two obstacles that prevent its effective application to images of steel surface defect. First, the fixed contrast strength in contrast learning is not suitable for representation learning of steel surface defect samples. According to [17], contrastive learning is classified as strongly contrastive learning and weakly contrastive learning, which has strong and weak contrast among samples, respectively. Under strong contrast, it is advantageous for representation learning of inter-class similar samples of steel surface defect, but detrimental to that of intra-class similar samples of steel surface defect [see Fig. 1(a)]. The weak contrast is the opposite of the strong contrast [see Fig. 1(b)]. These two extreme contrast strengths will destroy the potential semantic information of defect samples, thus reducing the quality of the learned representations. Second, the unlabeled data of steel surface defects are inadequate, unlike ImageNet [18] that can provide lots of unlabeled data for contrastive learning to learn better representations.

Consequently, to counter the above-mentioned problems, a novel contrastive learning framework, namely flexible and diverse contrastive learning (FDCL), is proposed. It consists of flexible contrast (FiCo) and diverse generative adversarial networks (DGANs). To address the first problem, FiCo is proposed. In the FiCo, variable temperature discrimination is designed to flexibly adjust the contrast strength among steel defect samples. In addition, feature reconstruction (FR) is also designed to further adjust the contrast strength. To overcome the second problem, generative adversarial network (GAN) method [19] is adopted to assist FiCo. However, GAN relies on high volume of training data, and it is prone to suffer from mode collapse [20] with insufficient steel surface defect data. Mode collapse refers to the fact that the generator only duplicate images, which is ineffective for contrastive learning. To alleviate mode collapse with limited steel surface defect data, the DGAN is proposed. In the DGAN, the multiple generators' weights are used to generate diverse images, providing more defect data for FiCo.

In summary, the major contributions of this article are summarized as follows.

1) A novel contrastive learning framework, FDCL, is proposed to overcome two obstacles of existing contrastive learning that cannot be effectively applied to steel surface defect images: inappropriate contrast strength and insufficient unlabeled data. It can accurately identify steel surface defects with few labeled samples.
2) FiCo is proposed for representation learning of steel surface defect images. Unlike contrastive learning with fixed contrast strength, FiCo uses designed variable temperature discrimination and FR to flexibly adjust contrast strength, improving the quality of learned representations.
3) A generative method named DGAN is proposed to supplement the unlabeled data. DGAN uses multiple generator weights to alleviate mode collapse and generate diverse images for FiCo, further improving the quality of learned representations.
4) Extensive experiments are conducted on four standard steel surface defect data sets to validate the effectiveness of the proposed method. The experimental results show significant improvements over the state-of-the-art methods.

II. Related Works

In this section, deep learning in steel surface defect recognition, contrastive learning, and GANs will be briefly introduced.

A. Deep Learning in Steel Surface Defect Recognition

In recent years, deep learning methods have achieved good performance in steel surface defect recognition tasks. A work by Zhou et al. [7] utilized a CNN to identify steel surface defects and achieved a good performance. In [8], a multiscale pyramidal pooling network was introduced to improve the recognition rate for steel defects. Chen et al. [9] proposed to combine three different CNNs to identify steel surface defects. A method [10] combining semi-supervised learning and convolutional autoencoder is proposed to identify steel surface defects. He et al. [11] presented a semi-supervised learning method on the basis of GANs and residual networks to identify steel surface defects. Through multiple training processing, the above-mentioned two networks can achieve higher accuracy and better robustness. Ren et al. [12] proposed a defect identification method that first pre-trains on a CNN and then migrates the pre-trained model weights to the classifier for defect recognition. Gao et al. [21] fused multilevel information to identify steel surface defect. Wang et al. [22] suggested a graph convolutional method to identify surface defects. A work of [23] combined CNN and pseudo-label to identify steel surface defect.

Although deep learning methods have performed with excellence in steel surface defect recognition, they require lots of training samples to maintain a high recognition rate. In some industrial cases, defects are rare on the steel surface and only few samples are available after selection and cropping. This problem constrains the widespread use of deep learning methods for steel defect recognition. In recent years, contrastive learning and GANs are effective solutions to this problem.

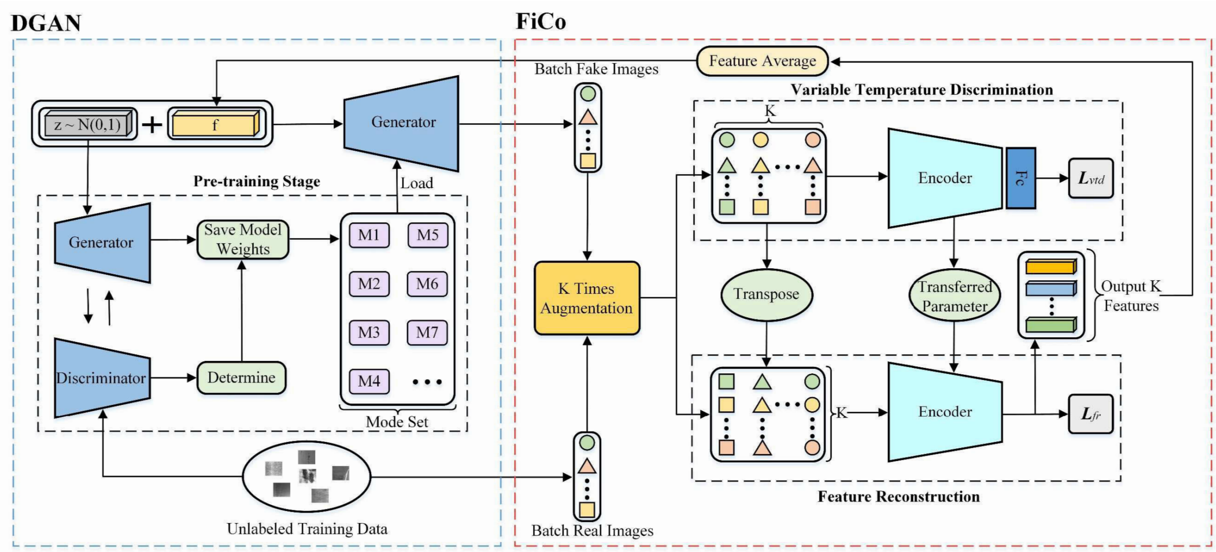


Fig. 2. Overview of the proposed FDCL method. The different shapes in the batch fake and real images indicate different instance samples; different colored samples of the same shape indicate augmentation from different times. In the pre-training stage, DGAN stores model weights in mode set via discriminator. In the training stage, first the batch fake images are generated by the generator randomly loaded with a weight from the mode set. Then, the fake and real images are fed into FlCo for representation learning. The encoder is trained by variable temperature discrimination and FR. k features output by FlCo are averaged at the end of each iteration and will be used as intake to the DGAN generator. Feature average indicates taking the mean value of k features in the width and height dimensions.

B. Contrastive Learning

Recently, contrastive learning has achieved the most advanced performance on computer vision tasks. In a study of He et al. [13], a dictionary look-up method is proposed to representation learning by using a momentum encoder. After that, a simple contrastive learning framework, named SimCLR [14], was proposed. The SimCLR has three main components, which are projection head, composition of data augmentations, and larger batch size. Chen et al. [15] established a stronger baseline called MoCov2 by introducing data augmentation and nonlinear transformation to MoCo. Zhai et al. [17] proposed a method of classifying batch instance samples to representation learning. Unlike the mechanism of the above-mentioned methods, SwAV [24] applies the clustering algorithm into the contrastive learning. SwAV clusters data by computing the cluster assignment from one augmentation and predicting it from another augmentation. Although no negative samples are involved in the learning mechanism of SwAV, the clustering centers can be viewed as negative prototypes.

Whereas the above-mentioned methods rely on negative examples for feature learning, BYOL [25] uses only positive samples. BYOL consists of an online network and a target network that predict and interact with each other. To further simplify BYOL, Chen and He [26] proposed SimSiam, a method that directly maximizes the similarity of different views from the same image, without using either a momentum encoder or negative pairs. Experiment verified that stopping the target network gradient is the key to prevent collapse, rather than the momentum encoder.

C. Generative Adversarial Networks

GAN is a deep learning method for unsupervised learning which consists of a generator and a discriminator. The goal of the generator is to trick discriminator into reaching a high discrimination value by learning generated distribution p_g to match data distribution p_{data}. Fake images are generated by feeding variables z sampled from uniform or Gaussian distribution p_z into the generator. The discriminator acts as a classifier with the aim of determining whether the image is true or false. The objective function of GAN is as follows:

$$\min_G \max_D V_{GAN}(D) = \mathbb{E}_{x\sim p_{\text{data}}}\big[\log D(x)\big] + \mathbb{E}_{z\sim p_z}\big[\log(1 - D(G(z)))\big]. \quad (1)$$

Although GAN has shown powerful potential for special tasks, reaching the better optima for GAN is still a challenge. Since GAN is vulnerable to mode collapse, especially with limited data. To improve the learning of GAN, lots of works have been proposed by modifying the GAN architecture. Deep convolutional GANs (DCGANs) [27] were proposed by introducing CNN to GAN architecture. Ghosh et al. [28] used m generators and a discriminator in order to learn the joint distribution of different domain samples. In addition to modifying the GAN architecture, lots of works have been done to improve the learning of GAN by analyzing the objective function. WGAN [29] successfully improved the stability of GAN training by using the Wassertein-1 metric as loss function. However, it does not work well for a deeper model. To address this problem, Gulrajani et al. [30] introduced gradient penalty to WGAN and successfully demonstrated a stable training performance. Mao et al. [31] proposed a least square loss for discriminator instead of the cross-entropy loss.

III. Proposed Method

The proposed FDCL mainly consists of two parts: FiCo and DGANs. Fig. 2 expounds the overall framework of the proposed FDCL. First, in the pre-training stage, DGAN stores multiple generator model weights in mode set according to different discriminant values. Next, in the training stage, the generator randomly loads a weight from mode set to generate batch fake images. After that, the batch fake and real images are enhanced k times and sent to FiCo for representation learning. Then, in the FiCo, k batch-enhanced images are encoded for variable temperature discrimination training. Finally, k batch-enhanced images are transposed and reencoded for FR training. In addition, k features' output by FiCo are averaged and used as intake to the DGAN generator to improve image quality, further facilitating representation learning.

A. Flexible Contrast

The FiCo is inspired by BIDFC [17]. There is a similarity between BIDFC and classification tasks that instance labels are assigned to each batch of samples during the model training phase. However, it is worth noting that the model can be difficult to converge during the training process, since the labels of each instance sample must be different in each epoch. To resolve this issue, k time augmentation on each sample in the batch obtains k batch augmented data for training. Cross-entropy loss is utilized as the target function to guide BIDFC, which is expressed as

$$L_{ce} = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\exp\left(w_i^T v/\tau\right)}{\sum_{j=1,\, j\neq i}^{n}\exp\left(w_j^T v/\tau\right)+\exp\left(w_i^T v/\tau\right)} \tag{2}$$

where n indicates the batch size, w_i and w_j refer to weights of positive and negative samples in the fully connected layer, respectively, v stands for sample embedding, and τ denotes the temperature coefficient.

BDIFC replaces contrastive loss [16] with cross-entropy loss to weaken the contrast strength among different samples, which is beneficial for representation learning of intra-class similar samples, whereas opposite for that of inter-class similar samples [see Fig. 1(b)]. BIDFC and strongly contrastive learning have opposite contrast strength in embeddings among samples. Due to the existence of both intra-class and inter-class similar samples in steel surface defect data at the same time, these two extreme contrast strengths are not suitable for representation learning of steel surface defect samples.

To address this problem, appropriate contrast strength needs to be found. Contrast strength is related to alignment and uniformity [32], which are two key properties of unsupervised contrastive learning, representing the degree of aggregation of positive samples and the uniformity of embeddings distribution, respectively. Excessive alignment and weak uniformity can lead to model collapse, which in turn can be detrimental to the semantic information of the features. For the representation learning of steel surface defect samples, it is important to find a balance between alignment and uniformity. According to [33], the temperature coefficient τ is able to regulate the relationship between alignment and uniformity, so we analyze the loss function for BIDFC. The gradients with respect to negative sample weights w_j is formulated as

$$\frac{\partial L_{ce}}{\partial w_j} = \frac{v}{\tau}\times\frac{1}{n}\sum_{j=1}^{n}P_j \tag{3}$$

$$P_j = \frac{\exp\left(w_j^T v/\tau\right)}{\sum_{j=1,\, j\neq i}^{n}\exp\left(w_j^{\mathrm{T}} v/\tau\right)+\exp\left(w_i^T v/\tau\right)} \tag{4}$$

where P_j denotes the probability of negative sample j being identified as positive sample i. From (3) and (4), one can observe that each negative sample has a different gradient, and their gradient values are proportional to $\exp(w_j^{\mathrm{T}} v/\tau)$. In addition, the more similar the negative samples are to the positive samples, the larger their gradient values are. The temperature coefficient τ has an amplifying and reducing effect on the gradient values. As the temperature decreases, the update of the model is concentrated in regions where the samples are highly similar, making the embedding distribution more uniform. And as the temperature increases, the model updates each sample equivalently, resulting in a more aggregated embedding distribution.

Based on the above-mentioned analysis of the temperature properties and combined with the characteristics of BIDFC, FiCo is proposed. FiCo consists of variable temperature discrimination and FR. The spirit of variable temperature discrimination is the combination of temperature function and representation learning. Different from the temperature constant, the temperature function allows flexibility to adjust the contrast strength among samples. In the early stage of model training, the temperature is kept constant and set small so that the model focuses on distinguishing among embedding of different samples. In the later stages of model training, the temperature values gradually increase. The model gradually focuses on aggregating positive samples. Essentially, the variable temperature discrimination separates and aggregates different instances through temperature changes, while implicitly separating inter-class similar samples and aggregating intra-class similar samples. The objective function of the variable temperature discrimination is as follows:

$$L_{vtd} = -\frac{1}{kn}\sum_{l=1}^{k}\sum_{i=1}^{n}\log\frac{\exp\left[\left(w_i^l\right)^T v/\tau(t)\right]}{\sum_{j=1}^{n}\exp\left[\left(w_j^l\right)^T v/\tau(t)\right]} \tag{5}$$

$$\tau(t) = \begin{cases} C, & t\leq t_1 \\ C+\log_{\mathrm{C}}(t/100), & t>t_1 \end{cases} \tag{6}$$

where k represents the enhancement times, w_i^l stands for the weight of the lth enhanced view of simple i, and $\tau(t)$ denotes the temperature function. Different forms of temperature functions are also used in variable temperature discrimination, as follows:

$$\tau(t)_1 = \begin{cases} C, & t\leq t_1 \\ C+0.01\times(t-200), & t>t_1 \end{cases} \tag{7}$$

$$\tau(t)_2 = \begin{cases} C, & t\leq t_1 \\ C+C^{(t-300)}, & t>t_1 \end{cases} \tag{8}$$

$$\tau(t)_3 = C+\log_C\left[1+(t/150)\right] \tag{9}$$

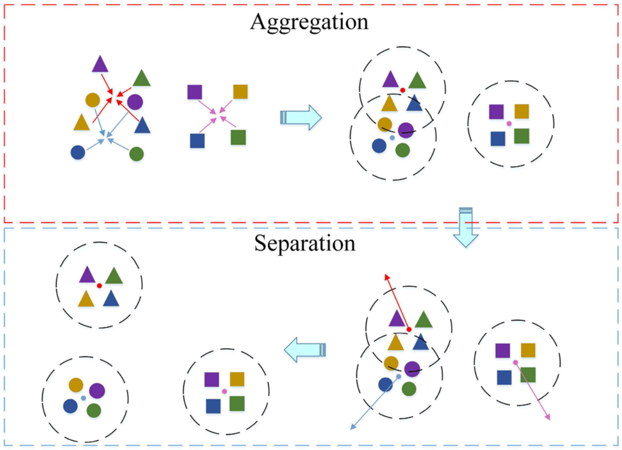


Fig. 3. Illustration of FR. Different shapes indicate different instance sample embeddings. Different colors of the same shape indicate enhanced embeddings of the same sample from different augmentation times.

where t denotes the training epochs and C represents the temperature constant. A detailed discussion of the effects of these temperature functions on model performance is shown in Section IV (see Table XIII).

In BIDFC, to further weaken the contrast strength among similar samples, a dynamic-weighted variance (DWV) loss is adopted to aggregate positive sample pairs. Although aggregating positive sample pairs implicitly brings the embedding of intra-class similar samples closer together, it also implicitly brings the embedding of inter-class similar samples closer together, which will be harmful to the semantic features. To aggregate intra-class similar samples while separating inter-class similar samples, FR is proposed. The illustration of the FR is shown in Fig. 3. The FR consists of aggregation and separation components. In the aggregation part, different enhanced views of the same sample are aggregated by minimizing the variance among their features. The objective function of the aggregation part is as follows:

$$L_a = \frac{1}{nk}\sum_{i=1}^{n}\sum_{j=1}^{k}\left(z_i^j - \frac{1}{k}\sum_{j=1}^{k} z_i^j\right)^2 \tag{10}$$

where z_i^j is the feature vector of the jth enhanced view of simple i from the global pooling layer. In the separation part, different instance samples are appropriately separated by increasing the variance among their features. The objection function of the separation part can be expressed as

$$L_s = \left[\frac{1}{p}\sum_{d=1}^{p}\left(u - \frac{1}{n}\sum_{i=1}^{n} u_i\right)^2 - \alpha\right]^2 \tag{11}$$

$$u = \frac{1}{k}\sum_{j=1}^{k} b_d^j \tag{12}$$

where p denotes the length of z_i^j, b_d^j represents a feature value from z_i^j, and α refers to the degree of separation among instance samples. These two parts serve to implicitly aggregate intra-class similar samples and separate inter-class similar samples. Based on the above-mentioned two parts of the loss, the loss is defined as

$$L_{fr} = \text{weight}(t)_1 L_a + \text{weight}(t)_2 L_s \tag{13}$$

where $\text{weight}(t)_1$ and $\text{weight}(t)_2$ indicate the slowly increasing function of training epochs t for adjusting the FR loss. As learned from [31], the representations learned by the model will be impaired if large weight is used at the beginning of model training. It is difficult for the model to learn information from images if the weight is small in the training phase. Therefore, weight is defined as a function that increases with training epochs t. The weight value of the aggregation part changes based on T_1 and T_2, and the weight value of the separation part changes based on T_3 and T_4, written as

$$\text{weight}(t)_1 = \max\left\{a_f \min\left(\frac{t - T_1}{T_2 - T_1}, 1\right), 0\right\} \tag{14}$$

$$\text{weight}(t)_2 = \max\left\{a_f \min\left(\frac{t - T_3}{T_4 - T_3}, 1\right), 0\right\} \tag{15}$$

where $\alpha_f > 0$ denotes the weight coefficient. Overall, the total loss function is formulated as

$$L = L_{vtd} + L_{fr}. \tag{16}$$

B. Diverse GAN

The performance of contrastive learning is correlated with the amount of data. Steel surface defect data are insufficient, which will limit the performance of contrastive learning. To further improve the performance of FiCo, a generative method named DGAN is proposed to generate diverse images for FiCo.

Although GAN is capable of generating large amounts of data, it is prone to mode collapse with insufficient data. Under mode collapse, lots of duplicate images are generated, which is ineffective for contrastive learning. In response to mode collapse, m generators are adopted by MAD-GAN [28] to generate images. In order for m generators to capture diverse modes, the goal of the discriminator is to distinguish not only between fake and real images but also from which generator the generated images come. However, simultaneously training multiple generators in MAD-GAN consumes a lot of computational resources, and how many generators are required to capture all modes of the data distribution is unknown.

To address the above-mentioned problem, we proposed DGAN to simply and effectively alleviate mode collapse and reduce computational cost. The goal of DGAN is to capture different modes using multiple model weights, but only with one generator. Mode collapse is characterized by the discriminator pushing the generator away whenever the generator grabs a mode during the training phase. Thus, the generator will catch different single mode instead of all modes in the case of mode collapse. Based on this property of mode collapse, we will save different generator weights in mode set according to the different discriminant values' output by discriminator. Since discriminator can be viewed as metric between generated distribution and data distribution, different outcomes of the discriminator represent different modes captured by generator. One thing to note is that the discriminator and the generator have to reach a convergent state before

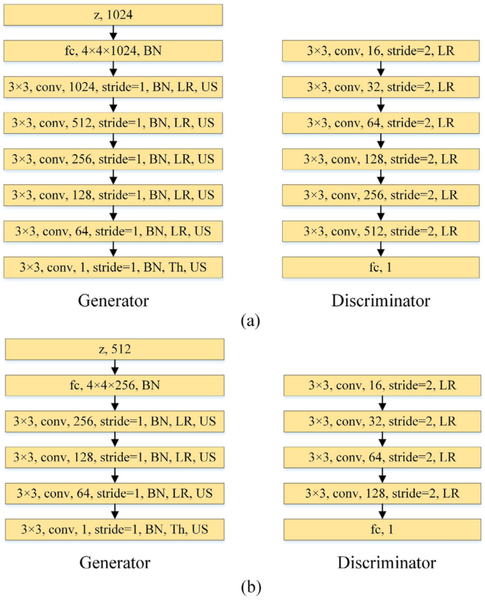

z, 1024
fc, 4×4×1024, BN
3×3, conv, 1024, stride=1, BN, LR, US
3×3, conv, 512, stride=1, BN, LR, US
3×3, conv, 256, stride=1, BN, LR, US
3×3, conv, 128, stride=1, BN, LR, US
3×3, conv, 64, stride=1, BN, LR, US
3×3, conv, 1, stride=1, BN, Th, US
Generator
3×3, conv, 16, stride=2, LR
3×3, conv, 32, stride=2, LR
3×3, conv, 64, stride=2, LR
3×3, conv, 128, stride=2, LR
3×3, conv, 256, stride=2, LR
3×3, conv, 512, stride=2, LR
fc, 1
Discriminator
(a)
z, 512
fc, 4×4×256, BN
3×3, conv, 256, stride=1, BN, LR, US
3×3, conv, 128, stride=1, BN, LR, US
3×3, conv, 64, stride=1, BN, LR, US
3×3, conv, 1, stride=1, BN, Th, US
Generator
3×3, conv, 16, stride=2, LR
3×3, conv, 32, stride=2, LR
3×3, conv, 64, stride=2, LR
3×3, conv, 128, stride=2, LR
fc, 1
Discriminator
(b)

Fig. 4. Configuration of the model for DGAN. "$K\times K$, conv, C, stride= S" means a convolutional layer with $K \times K$ kernel, C output filters, and stride $= S$. BN and US denote batch normalization and upsampling, respectively. LR and Th denote LeakyReLu activation and Tahn activation, respectively. "fc, N" indicates a fully connected layer with N output nodes. (a) DGAN architecture for 256 × 256 image. (b) DGAN architecture for 64 × 64 image.

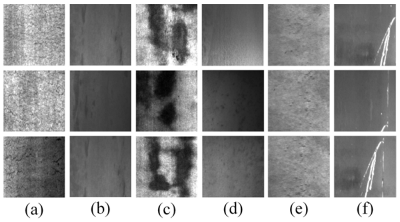


Fig. 5. Examples of defect samples in the NEU-CLS. (a) Cr. (b) In. (c) Pa. (d) Ps. (e) Rs. (f) Sc.

saving the model, which ensures that the generator learns to produce real looking images. Here, we save 400 generator weights to the mode set. The effects of number of generator weights on model performance are shown in the experimental section (see Table XVII). The configuration of the DGAN is shown in Fig. 4(a) and (b) for generating images of 256 × 256 (NEU-CLS [36], X-SDD [37], and GC10 [38]) and 64 × 64 (NEU-CLS-64 [36]), respectively. To further improve the image quality, k features generated by FiCo are averaged and fed into the DGAN generator inspired by [35].

The objection function of LSGAN [31] is adopted by DGAN to balance image quality and training stability, written as

$$\begin{aligned}\min_{D} V(D) &= \frac{1}{2}\mathbb{E}_{x\sim p_{\text{data}}}\left[(D(x)-1)^2\right]\\&\quad+\frac{1}{2}\mathbb{E}_{z\sim p_z}\left[(D(G(z))+1)^2\right]\\\min_{G} V(G) &= \frac{1}{2}\mathbb{E}_{z\sim p_z}\left[(D(G(z)))^2\right]\end{aligned}\tag{17}$$

where x represents the steel surface defect image sampled from the data distribution p_{data}, z denotes the variable sampled from uniform or Gaussian distribution p_z, D stands for the discriminator, and G is the generator.

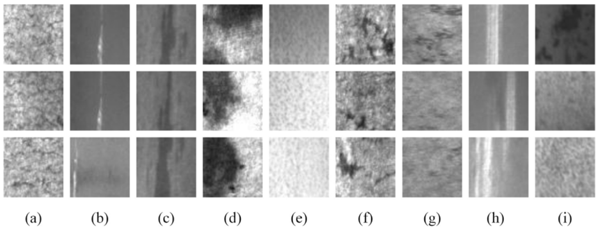


Fig. 6. Examples of defect samples in the NEU-CLS-64. (a) Cr. (b) Gg. (c) In. (d) Pa. (e) Ps. (f) Rd. (g) Rs. (h) Sc. (i) Sp.

IV. Experimental Results and Discussion

To validate and analyze our method, experiments are conducted on four benchmark data sets. First, we introduce the four steel surface defect data sets and construct few-label databases, followed by experimental settings present. Then, the comparison between the proposed method and related state-of-the-art methods is made. Thereafter, the robustness of the proposed method is tested for noise and blur inputs under few labeled samples. Finally, the proposed method is analyzed by ablation study.

A. Dataset and Experimental Settings

1) Steel Surface Defect Data Sets: NEU-CLS [36] refers to a benchmark database containing six kinds of steel surface defects, including crazing (Cr), inclusion (In), patches (Pa), pitted surface (Ps), rolled-in scale (Rs), and scratches (Sc). Each category contains 300 images with a size of 200 × 200. Sixty percent of them are randomly selected into the training set and the rest as the testing set. Some samples in this dataset are shown in Fig. 5.

Different from NEU-CLS, NEU-CLS-64 [36] has a smaller image size (64 × 64) and more defect types, including crazing (Cr), grooves and gouges (Gg), inclusion (In), patches (Pa), pitted surface (Ps), rolling dust (Rd), rolled-in scale (Rs), scratches (Sc), and spots (Sp). The number of these nine defect samples is 1589, 1210, 1148, 797, 775, 773, 438, 296, and 200. The data set is divided in the same way as NEU-CLS. Some samples in this data set are shown in Fig. 6.

X-SDD [37] is a data set with seven types of surface defects of hot rolled steel strip, including finishing roll printing, iron sheet ash, oxide scale of plate system, oxide scale of temperature system, red iron, inclusion, and scratch. The number of these seven defect samples is 203, 122, 63, 203, 397, 238, and 134. The sample size in the data set is much smaller. Following the procedure of [37], 70% of the defect samples are randomly selected into the training set and the rest as the testing set. Some samples in this data set are shown in Fig. 7.

GC10 [38] is a data set with ten types of surface defects of steel plate, including punching (Pu), welding line (Wl), crescent gap (Cg), water spot (Ws), oil spot (Os), silk spot (Ss), inclusion (In), rolled pit (Rp), crease (Cr), and waist folding (Wf). Since an image contains different types of defects, we crop the defects of the image according to bounding box. The number of these ten defects samples is 329, 513, 265, 354, 569, 884, 347, 85, 74, and 143. The data set is divided

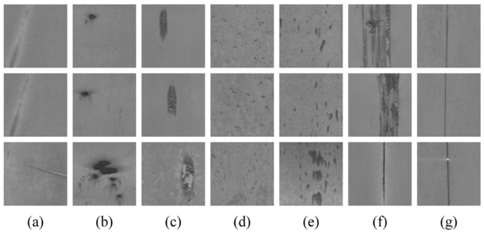


Fig. 7. Examples of defect samples in the X-SDD. (a) Finishing roll printing. (b) Iron sheet ash. (c) Oxide scale of plate system. (d) Oxide scale of temperature system. (e) Red iron. (f) Slag inclusion. (g) Surface scratch.

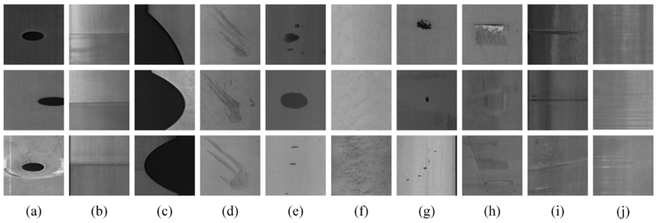


Fig. 8. Examples of defect samples in the GC10. (a) Pu. (b) Wl. (c) Cg. (d) Ws. (e) Os. (f) Ss. (g) In. (h) Rp. (i) Cr. (j) Wf.

in the same way as NEU-CLS. Some samples in this data set are shown in Fig. 8.

The label information of the above-mentioned three training sets is removed as unlabeled training data for contrastive learning.

2) Few-Label Database: To evaluate the classification performance of FDCL with few labeled samples, few-label database is constructed. It is comprised of several equivalent samples that are randomly extracted from each class in the training set. Four few-label databases are created as experimental databases, namely 1-, 2-, 3-, and 4-label.

3) Evaluation Metrics: Linear evaluation (LE) and fine-tuning (FT) are widely used for performance evaluation of contrastive learning [39]. LE committed to freezing model weights and only the fully connected layer is trained. It measures the relevance of learned features by the model during the unsupervised learning phase to downstream tasks. Different from LE, the weight of all layers will be trained during FT phase. It reflects the potential of pre-trained models for downstream task performance.

To evaluate the quality of images generated by GAN methods, Frechet inception distance (FID) [40] is employed to quantify generation quality of steel surface defect. It reflects the distance between two distributions of the real and generated images. The lower the FID, the better the results.

4) Implementation Details: Crop with resize, random vertical flip, random horizontal flip, contrast, and brightness are all adopted as the data augmentation of FiCo. The number of augmentation k is set to 5. Being as the encoder for unsupervised learning, ResNet18 is trained at batch size of 64 for 300 epochs. In the temperature function $\tau(t)$, C is set to 3.0 and t_1 is set to 200. $T_1 = 30$, $T_2 = 150$, $T_3 = 150$, $T_4 = 300$, and $\alpha_f = 1.0$ in FR loss. The separation degree α is set to 1.0. The Adam optimizer is exploited for FiCo training with learning of 0.3. DGAN is trained 4000 epochs with a batch size of 64 and learning rate of 0.0002. The Adam optimizer is adopted by DGAN. In the LE and FT stage, the model is trained for 200 epochs with learning rate of 0.03. In addition to NEU-CLS-64, the images of NEU-CLS, X-SDD, and GC10 are resized to 224 × 224 before input to train. To ensure accurate experimental results, four few-label databases are randomly created one hundred times to conduct the experiments. Then the average accuracy and standard deviations are calculated for each database. All experiments are based on Pytorch framework and implemented in a computer with an Nvidia GeForce RTX 2080Ti 11G GPU and an Intel Core of i7-9800X CPU.

TABLE I

COMPARISON WITH CONTRASTIVE LEARNING METHODS ON NEU-CLS FEW-LABEL DATABASES. FT AND LE

Method	1-label	2-label	3-label	4-label
FT:				
SimCLR	75.02±7.42	85.22±4.60	88.71±4.16	91.29±3.29
MoCov2	74.67±8.28	83.05±5.01	86.48±3.82	88.52±3.48
SimSiam	49.00±9.00	62.76±7.38	70.38±5.60	73.91±5.66
BYOL	56.37±9.87	71.83±8.39	78.58±6.54	82.44±5.21
SwAV	48.61±7.73	59.72±7.36	66.69±6.85	71.02±5.82
BIDFC	77.53±8.74	86.30±5.73	89.31±4.74	91.31±3.44
FixMatch	39.45±8.08	78.49±4.71	83.95±4.65	91.92±2.67
MPL	51.83±9.97	52.58±7.38	81.79±5.61	86.00±3.61
SimMatch	79.43±5.16	90.87±4.17	92.38±2.77	93.11±2.11
Rotation	57.59±6.76	68.40±5.95	73.83±4.97	77.78±4.59
InstFeat	56.40±7.30	70.67±6.58	78.00±5.30	82.64±4.21
ours	**91.43±6.04**	**95.35±2.90**	**96.04±1.88**	**96.48±1.42**
LE:				
SimCLR	74.33±7.53	85.46±4.83	88.83±4.13	91.24±3.03
MoCov2	74.49±7.35	84.43±4.82	88.20±3.71	90.37±2.57
SimSiam	50.35±8.80	63.75±7.51	71.34±5.55	75.30±5.03
BYOL	63.52±7.68	78.31±6.49	84.44±4.38	87.35±3.73
SwAV	48.67±7.29	58.99±6.31	64.47±5.01	68.60±4.34
BIDFC	77.51±8.72	86.28±5.73	89.28±4.75	91.29±3.43
FixMatch	30.51±6.23	75.20±4.44	82.66±4.44	92.09±2.08
MPL	52.23±9.75	50.14±7.13	85.28±3.92	89.16±2.35
SimMatch	80.05±5.44	91.38±4.14	92.77±2.12	92.48±1.86
Rotation	33.64±3.99	37.40±3.97	40.69±4.38	43.38±4.00
InstFeat	46.75±6.28	56.89±5.34	62.84±4.55	67.52±4.93
ours	**91.43±6.04**	**95.35±2.90**	**96.05±1.87**	**96.47±1.42**

B. Comparison With State-of-the-Art Methods

In this section, the proposed method is compared with state-of-the-art methods, including SimCLR [14], MoCov2 [15], BIDFC [17], SwAV [24], BYOL [25], SimSiam [26], FixMatch [41], MPL [42], SimMatch [43], Rotation [44], and InstFeat [45]. These methods are trained at the unsupervised learning stage with the data augmentations introduced earlier.

1) Results on NEU-CLS: The experimental results are listed in Table I, the proposed method achieves competitive performance on four few-label databases of NEU-CLS. Specifically, under FT, the proposed method achieves an average accuracy of 91.43%, 95.35%, 96.04%, and 96.48% on the 1-, 2-, 3-, and 4-label databases, respectively. These are significantly high than the state-of-the-art contrastive learning methods. The accuracy of the proposed method is 12.00%, 4.48%, 3.66%, and 3.37% higher than the best state-of-the-art methods on these four few-label databases, respectively. Under LE, the proposed method also achieves considerable accuracy. These results demonstrate that our method can learn more meaningful semantic information from steel surface defect images.

TABLE II

COMPARISON WITH CONTRASTIVE LEARNING METHODS ON NEU-CLS-64 FEW-LABEL DATABASES

Method	1-label	2-label	3-label	4-label
FT:				
SimCLR	55.04±9.11	64.36±6.03	68.96±4.71	71.96±4.24
MoCov2	77.12±5.86	80.94±3.47	83.39±2.77	84.72±2.50
SimSiam	51.75±8.12	62.57±5.80	68.69±4.38	70.96±4.13
BYOL	59.44±7.73	74.68±5.84	80.68±4.27	84.34±2.74
SwAV	36.14±7.59	46.14±5.60	53.31±5.43	57.86±4.89
BIDFC	74.77±5.27	78.38±4.33	79.63±4.55	80.38±3.77
FixMatch	22.18±5.58	49.04±7.03	56.46±6.00	73.69±3.52
MPL	30.17±8.55	50.66±6.42	53.36±6.38	72.22±4.72
SimMatch	64.91±6.93	81.49±3.63	81.96±3.30	85.96±2.20
Rotation	18.04±5.70	21.70±5.90	25.47±6.41	26.76±4.94
InstFeat	68.05±7.04	75.25±4.57	79.60±3.58	82.20±3.12
ours	**81.56±5.47**	**85.61±3.92**	**87.37±3.36**	**88.30±2.93**
LE:				
SimCLR	55.39±9.05	64.68±6.12	69.32±4.54	72.38±3.88
MoCov2	76.10±6.55	81.14±3.35	83.83±2.52	85.25±2.54
SimSiam	60.11±7.01	68.31±5.36	72.66±4.31	74.91±3.67
BYOL	61.75±6.88	76.04±5.20	82.21±3.69	85.12±2.82
SwAV	35.16±7.24	43.90±5.24	50.94±5.36	54.88±4.84
BIDFC	74.80±5.28	78.41±4.34	79.63±4.54	80.39±3.77
FixMatch	18.56±3.81	49.25±6.45	49.89±5.10	74.18±2.98
MPL	31.30±6.43	51.72±6.24	56.57±4.60	77.61±3.59
SimMatch	60.23±7.13	81.70±3.26	82.03±2.95	85.52±2.25
Rotation	12.59±2.32	12.79±2.46	13.24±2.48	13.61±1.97
InstFeat	66.56±7.18	75.81±4.51	80.84±3.16	83.63±2.73
ours	**81.55±5.56**	**85.62±3.92**	**87.37±3.42**	**88.28±2.96**

TABLE III

COMPARISON WITH CONTRASTIVE LEARNING METHODS ON X-SDD FEW-LABEL DATABASES

Method	1-label	2-label	3-label	4-label
FT:				
SimCLR	48.23±7.50	59.30±6.82	64.81±5.74	68.00±5.06
MoCov2	46.93±7.04	62.32±6.40	70.52±5.30	75.59±4.95
SimSiam	31.56±7.30	40.17±6.22	44.11±6.78	47.88±8.00
BYOL	43.00±7.04	57.28±6.70	66.64±6.66	71.76±5.87
SwAV	41.55±7.17	52.09±6.46	59.82±5.34	64.23±5.37
BIDFC	51.75±8.99	59.07±8.31	64.54±6.51	67.96±6.05
FixMatch	41.70±9.40	61.99±5.79	80.09±4.53	87.02±2.51
MPL	49.28±8.54	65.75±5.07	77.96±4.00	82.42±4.09
SimMatch	49.75±9.06	65.74±6.16	76.27±5.54	81.84±4.14
Rotation	52.85±8.50	64.78±5.53	71.32±4.55	75.96±3.87
InstFeat	44.10±6.28	57.27±5.69	65.86±4.91	71.18±4.90
ours	**71.42±8.60**	**81.59±5.55**	**86.38±4.08**	**88.46±3.70**
LE:				
SimCLR	48.20±7.62	59.07±6.85	64.94±5.92	67.63±5.23
MoCov2	45.67±7.21	61.50±6.85	68.38±5.15	72.31±5.18
SimSiam	31.13±7.39	40.24±6.67	44.39±5.21	54.38±4.47
BYOL	43.57±7.05	60.36±6.43	68.55±5.27	73.80±5.27
SwAV	40.29±7.24	50.30±6.25	55.88±5.60	60.02±5.47
BIDFC	51.76±9.01	59.04±8.28	64.58±6.49	67.93±6.05
FixMatch	38.30±9.54	59.52±5.88	79.60±4.43	87.19±2.40
MPL	46.59±8.59	67.53±5.23	79.96±3.77	80.71±3.98
SimMatch	49.19±8.48	65.80±6.49	75.48±6.12	78.08±3.81
Rotation	31.66±7.94	39.73±7.31	44.67±6.99	48.36±5.64
InstFeat	40.77±6.25	57.80±5.66	65.65±4.49	70.67±4.42
ours	**71.42±8.62**	**81.58±5.57**	**86.40±4.08**	**88.48±3.70**

2) Results on NEU-CLS-64: The experimental results are shown in Table II. From the results, it can be seen that the proposed method achieves considerable performance even with more types of defects and smaller image resolution. Under FT, the proposed method achieves an average accuracy of 81.56%, 85.61%, 87.37%, and 88.30% on the 1-, 2-, 3-, and 4-label databases, respectively. The performance of the proposed method exceeds the state-of-the-art methods by at least 4.44%, 4.67%, 4.08%, and 2.34% on these four few-label databases, respectively. Under LE, the proposed method achieves comparable performance to FT. The results suggest that the proposed method can generalize to more steel surface defect types and provide competitive performance at low resolution.

3) Results on X-SDD: The experimental results presented in Table III show that the proposed method is significantly better than the state-of-the-art contrastive learning methods. Under FT, the proposed method achieves an average accuracy of 71.42%, 81.59%, 86.38%, and 88.46% on the 1-, 2-, 3-, and 4-label databases, respectively. The performance of the proposed method exceeds the state-of-the-art methods by at least 18.57%, 15.84%, 6.29%, and 1.44% on these four few-label databases, respectively. Similar results are also observed on LE. The results suggest that the proposed method also performs well for the data set with smaller sample size and different types of defects.

TABLE IV

COMPARISON WITH CONTRASTIVE LEARNING METHODS ON GC10 FEW-LABEL DATABASES

Method	1-label	2-label	3-label	4-label
FT:				
SimCLR	43.67±6.44	53.53±5.63	59.01±3.95	61.85±3.70
MoCov2	50.73±5.52	58.72±4.75	62.81±4.60	66.17±3.99
SimSiam	21.91±2.32	24.41±2.45	25.26±2.76	25.58±2.72
BYOL	37.60±6.08	47.77±5.68	55.17±5.83	60.13±5.23
SwAV	36.06±6.70	44.13±6.07	47.88±5.42	51.50±3.85
BIDFC	53.87±6.42	61.61±5.07	66.25±3.53	67.92±3.41
FixMatch	26.46±3.63	33.85±3.60	49.15±5.08	54.83±3.98
MPL	28.56±4.52	48.85±5.07	57.15±4.03	60.47±4.49
SimMatch	44.81±6.45	57.37±5.01	61.72±3.88	71.68±4.03
Rotation	32.08±4.83	42.55±4.06	48.20±4.57	52.58±4.20
InstFeat	38.39±5.98	47.81±5.26	54.82±5.02	58.78±4.58
ours	**58.47±5.35**	**65.81±4.79**	**70.66±4.29**	**73.40±3.78**
LE:				
SimCLR	43.76±6.43	53.37±5.78	58.80±4.19	61.77±3.66
MoCov2	52.98±5.16	61.09±4.98	65.69±4.80	68.84±4.15
SimSiam	21.87±2.81	24.15±2.78	25.32±2.59	26.23±2.61
BYOL	45.31±5.93	57.83±5.34	65.43±4.53	69.71±4.32
SwAV	36.59±5.90	43.96±5.41	47.92±4.14	51.19±3.66
BIDFC	53.90±6.43	61.63±5.08	66.26±3.52	67.93±3.41
FixMatch	24.97±3.34	29.99±2.81	49.78±4.42	54.34±3.84
MPL	28.65±5.25	48.07±4.25	58.41±4.15	61.65±4.47
SimMatch	43.97±6.99	57.49±4.84	60.53±4.17	72.81±3.87
Rotation	21.85±3.99	25.54±3.68	27.40±3.15	29.52±2.66
InstFeat	36.91±5.82	48.00±4.98	55.99±4.48	59.94±3.95
ours	**58.47±5.35**	**65.82±4.80**	**70.69±4.29**	**73.43±3.78**

4) Results on GC10: The experimental results in Table IV report that the proposed method achieves competitive performance. Under FT, the proposed method achieves an average accuracy of 58.47%, 65.81%, 70.66%, and 73.40% on the 1-, 2-, 3-, and 4-label databases, respectively. The performance of the proposed method exceeds the state-of-the-art methods by at least 4.60%, 4.20%, 4.41%, and 1.72% on these four few-label databases, respectively. Under LE, the proposed method also achieves considerable accuracy. The results also demonstrate the effectiveness of the proposed method.

From the above-mentioned experiments, it can be seen that the proposed method outperforms the state-of-the-art contrastive learning methods on four standard steel surface defect

TABLE V

COMPARISON WITH CLASSIFICATION NETWORKS ON NEU-CLS FEW-LABEL DATABASES

Method	1-label	2-label	3-label	4-label
VGG16	29.94±6.18	31.48±6.59	32.06±5.91	33.01±6.18
EffectvNet2	37.04±8.19	47.70±7.77	53.59±6.71	58.21±6.67
DenseNet	54.14±6.84	68.29±5.63	72.73±5.88	77.08±3.67
ShuffleNetv2	44.76±9.38	55.11±9.06	63.76±6.21	64.6±5.89
MobileNetv3	41.17±6.68	51.32±7.15	58.96±6.06	64.40±5.09
ResNet18	49.68±8.80	63.39±6.78	70.83±5.88	76.53±3.87
ours	**91.43±6.04**	**95.35±2.90**	**96.05±1.87**	**96.47±1.42**

TABLE VI

COMPARISON WITH CLASSIFICATION NETWORKS ON NEU-CLS-64 FEW-LABEL DATABASES

Method	1-label	2-label	3-label	4-label
VGG16	15.79±5.13	15.80±5.25	17.29±2.99	18.03±5.04
EffectvNet2	16.57±5.19	22.12±5.35	25.76±4.07	27.40±4.19
DenseNet	40.14±6.84	49.84±5.23	54.81±4.52	58.75±4.18
ShuffleNetv2	20.08±4.95	22.62±4.33	25.41±3.69	27.52±2.88
MobileNetv3	21.37±5.16	28.00±6.37	32.16±5.00	34.94±4.45
ResNet18	31.35±5.90	39.19±5.06	43.87±4.45	47.76±4.34
ours	**81.55±5.56**	**85.62±3.92**	**87.37±3.42**	**88.28±2.96**

data sets. The reason is twofold. One is that FDCL effectively handles the potential relationships among instance samples by suitable contrast strength. The second is that FDCL can provide itself with sufficient data in the unsupervised learning phase, which allows it to learn more diverse information. For comparative contrastive learning methods, their inappropriate contrast strength ignores the underlying relationships among samples and harms feature useful for downstream tasks. Moreover, the limited steel defect data prevents them from learning sufficient semantic information. For semi-supervised learning methods, the classifier tends to predict the wrong pseudo-labels to guide the model training, since few labeled data are available, resulting in impaired model performance. For other unsupervised learning methods, the low correlation between the pretext task and the defect classification task reduces the quality of the learned representations. In addition, their performance is also limited due to insufficient defect data.

C. Comparison With Classification Networks

In this part, the proposed method is compared with defect classification networks [46], [47], [48], [49], [50], [51] under four steel surface defect data sets. The comparison results are presented in Tables V–VIII, respectively. From the results, it can be seen that the proposed method outperforms these classification networks by a significant margin. The results account that the effective representation learning method can prominently improve the performance of steel surface defect recognition with few labeled samples compared to model architectures. The model architecture does not play an important role in steel surface defect recognition with few labeled samples.

D. Performance of the Proposed Method Under Noise and Blur Conditions

TABLE VII

COMPARISON WITH CLASSIFICATION NETWORKS ON X-SDD FEW-LABEL DATABASES

Method	1-label	2-label	3-label	4-label
VGG16	23.22±8.54	25.61±7.71	25.58±8.36	25.81±7.55
EffectvNet2	30.72±7.35	35.91±6.67	36.49±6.69	39.19±6.53
DenseNet	36.51±7.90	44.54±7.92	50.66±6.07	55.71±5.27
ShuffleNetv2	28.17±6.82	33.92±5.88	37.21±6.13	39.69±5.06
MobileNetv3	28.18±7.69	33.36±6.06	36.73±5.60	39.79±5.67
ResNet18	34.93±8.02	43.18±6.27	49.84±6.10	54.15±5.57
ours	**71.42±8.62**	**81.58±5.57**	**86.40±4.08**	**88.48±3.70**

TABLE VIII

COMPARISON WITH CLASSIFICATION NETWORKS ON GC10 FEW-LABEL DATABASES

Method	1-label	2-label	3-label	4-label
VGG16	15.85±4.68	17.51±4.78	17.99±4.67	19.36±4.28
EffectvNet2	16.79±3.24	19.64±4.33	21.44±4.37	22.37±4.28
DenseNet	20.22±3.59	24.85±3.03	27.74±2.99	29.35±2.76
ShuffleNetv2	18.29±3.87	21.41±3.58	23.33±3.06	24.50±2.97
MobileNetv3	16.13±3.98	19.89±3.01	22.86±3.17	24.83±3.03
ResNet18	21.32±3.09	25.18±2.90	28.13±2.65	29.78±2.74
ours	**58.47±5.35**	**65.82±4.80**	**70.69±4.29**	**73.43±3.78**

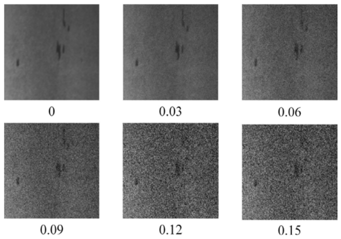


Fig. 9. Examples of defect images with different noise weights.

Industrial environments are complex and volatile. On production lines, the quality of the acquired images can be disturbed by noise and blur. These interference factors will affect the performance of recognition tasks. To evaluate the robustness of the proposed method, noise and blur are added to the test images and the few-label training images. The model is fine-tuned and linear evaluated on disturbed few-label databases and then evaluated performance on disturbed test set. The experiment is based on NEU-CLS data set.

1) Noise: Image sensor CCD camera will introduce noise when capturing images of steel surface defects, due to the influences of operating environment, electronic components, circuit structure, etc. Gaussian noise with a mean of 0 and a variance of 1 is introduced into the defect image. Fig. 9 shows the defect images perturbed by Gaussian noise with different weights. The experimental results under different noise conditions are presented in Table IX. From the experimental results, one can be observed that the proposed method is resistant to noise interference. When the noise weights are in range of 0 to 0.09, the proposed method can maintain a high recognition rate. Although the steel defect images are severely corrupted at noise weights of 0.12 and 0.15, the proposed method is able to achieve a moderate recognition rate. This result indicates that the proposed method is robust in noisy environments.

2) Blur: Blurred images of steel defects are mainly caused by improperly focused lenses, dirty lenses, and machine vibrations on the steel production lines. In this part, Gaussian blur with different radii are introduced into the defect images.

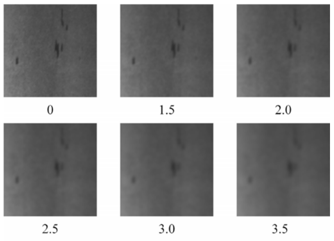


Fig. 10. Examples of defect images with different blur radii.

TABLE IX

PERFORMANCE OF THE PROPOSED METHOD ON NEU-CLS FEW-LABEL DATABASES WITH VARIOUS NOISE DISTURBANCE

Noise weight	1-label	2-label	3-label	4-label
FT:				
0	91.43±6.04	95.35±2.90	96.04±1.88	96.48±1.42
0.03	89.53±6.73	93.63±2.99	94.52±2.53	95.30±1.99
0.06	86.55±6.85	91.26±3.80	92.28±3.11	93.30±2.22
0.09	81.49±8.60	87.71±4.48	89.27±3.60	90.47±3.41
0.12	74.83±9.72	82.26±5.40	84.59±4.55	86.35±4.28
0.15	69.63±9.64	76.86±6.55	80.07±5.61	82.28±4.96
LE:				
0	91.43±6.04	95.35±2.90	96.05±1.87	96.47±1.42
0.03	89.54±6.73	93.64±2.99	94.53±2.53	95.32±1.99
0.06	86.55±6.83	91.25±2.81	92.25±3.11	93.26±2.48
0.09	81.46±8.56	87.66±4.47	89.21±3.61	90.42±3.41
0.12	74.83±9.65	82.22±5.40	84.54±4.58	86.31±4.27
0.15	69.71±9.61	76.86±6.57	80.09±5.63	82.26±5.04

TABLE X

PERFORMANCE OF THE PROPOSED METHOD ON NEU-CLS FEW-LABEL DATABASES WITH VARIOUS BLUR DISTURBANCE

Blur radius	1-label	2-label	3-label	4-label
FT:				
0	91.43±6.04	95.35±2.90	96.04±1.88	96.48±1.42
1.5	87.65±7.37	92.53±3.34	93.24±2.88	93.99±2.74
2.0	82.99±7.69	88.82±3.98	89.81±3.41	90.73±3.68
2.5	78.01±8.58	84.96±4.75	86.46±4.18	87.57±4.16
3.0	72.68±8.48	79.44±5.69	82.16±4.45	83.95±4.20
3.5	69.61±8.38	76.46±5.87	79.82±4.72	82.23±4.33
LE:				
0	91.43±6.04	95.35±2.90	96.05±1.87	96.47±1.42
1.5	87.64±7.36	92.52±3.33	93.23±2.91	93.99±2.75
2.0	82.97±7.64	88.79±3.96	89.79±3.40	90.72±3.70
2.5	78.00±8.53	84.95±4.73	86.42±4.17	87.53±4.14
3.0	72.43±8.43	79.42±5.65	82.12±4.42	83.87±4.12
3.5	69.57±8.36	76.37±5.80	79.73±4.69	82.17±4.34

Examples of blurred images and the recognition rates under different fuzzy degrees are shown in Fig. 10 and Table X. The experimental results indicate that our method can effectively resist the influences of fuzziness. Even in the case of severely blurred images, our method can maintain a considerable recognition rate in few-label databases. The results suggest that the model is also tolerant to fuzzy interference.

E. Ablation Study

To examine and analyze the effectiveness of the proposed method, ablation studies are conducted on NEU-CLS data set, including augmentation times k, degree of separation α, temperature function $\tau(t)$, DGAN, output features (OF), FR loss, number of generator weights, weight coefficient α_f, and iterations of FDCL. Evaluation metrics for ablation studies are based on LE.

TABLE XI

LE WITH DIFFERENT K VALUES UNDER NEU-CLS FEW-LABEL DATABASES

k	1-label	2-label	3-label	4-label
1	*fail*	*fail*	*fail*	*fail*
2	*fail*	*fail*	*fail*	*fail*
3	*fail*	*fail*	*fail*	*fail*
4	65.83±7.03	71.42±4.91	73.48±3.23	74.30±2.76
5	**91.43±6.04**	**95.35±2.90**	**96.05±1.87**	**96.47±1.42**
6	89.34±5.82	93.60±3.74	94.97±2.88	96.01±1.64

TABLE XII

LE WITH DIFFERENT α VALUES UNDER NEU-CLS FEW-LABEL DATABASES

α	1-label	2-lable	3-lable	4-label
0.6	86.32±6.03	91.70±4.22	93.33±3.57	94.35±2.65
0.8	89.57±6.50	93.92±3.11	94.85±2.30	95.57±1.67
1.0	**91.43±6.04**	**95.35±2.90**	**96.05±1.87**	**96.47±1.42**
1.2	89.50±6.30	92.95±3.90	94.19±3.36	95.08±2.52

TABLE XIII

LE WITH DIFFERENT TEMPERATURE FORMS UNDER NEU-CLS FEW-LABEL DATABASES

Method	1-label	2-label	3-label	4-label
1.0	75.03±7.43	83.99±4.53	87.16±3.17	88.07±3.56
2.0	78.41±7.89	87.65±4.58	90.31±3.67	90.81±3.13
3.0	87.79±6.84	93.26±3.46	94.78±2.38	95.60±1.80
4.0	85.71±6.07	91.93±4.18	93.59±3.11	94.39±2.31
5.0	84.53±5.16	89.64±3.65	91.37±3.10	92.53±2.16
$\tau(t)_1$	89.95±7.03	94.38±3.34	95.60±2.28	96.21±1.57
$\tau(t)_2$	89.09±6.91	94.23±3.02	95.36±2.11	95.87±1.70
$\tau(t)_3$	88.44±4.66	91.78±3.07	92.92±2.78	93.69±2.30
$\tau(t)$(ours)	**91.43±6.04**	**95.35±2.90**	**96.05±1.87**	**96.47±1.42**

TABLE XIV

COMPARISON OF FID SCORES WITH DIFFERENT GENERATION METHODS OF OUTPUT FEATURES

Method	FID
LSGAN	199.75
DCGAN	208.52
WGAN-GP	296.06
MAD-GAN	191.99
DGAN	162.16
DGAN + OF(ours)	**152.36**

1) Effects of Augmentation Times k: Augmentation times k is an important parameter that helps the model converge. As illustrated in Table XI, the augmentation times k is set to 1, 2, 3, 4, 5, and 6, respectively. The model fails to train effectively when $k = 1$, 2, and 3, because k is too small for the model to extract feature information from a batch of images. The recognition rate increases rapidly when k changes from 4 to 5, but decreases slightly when k reaches 6, which is due to overfitting. Therefore, it is reasonable to choose 5 for k.

2) Effects of Degree of Separation α: The role of the degree of separation α is to adjust the embedding distribution of samples. To test the effect of separation in the proposed method, the performance corresponding to different α values is presented in Table XII. It can be seen that when α is set to 1.0, the recognition accuracy achieves the best performance.

TABLE XV
LE for FiCo With Different Generation Methods Under NEU-CLS Few-Label Databases

Method	1-label	2-label	3-label	4-label
FiCo	89.37±5.91	93.00±2.96	93.47±2.64	94.65±1.66
FiCo + LSGAN	86.47±6.81	92.18±3.49	93.94±2.90	94.99±2.05
FiCo + WGAN-GP	84.74±5.88	90.54±3.61	92.58±2.55	93.86±2.15
FiCo + DCGAN	86.83±6.23	91.63±3.74	93.33±2.98	94.46±1.98
FiCo + MAD-GAN	88.78±4.90	92.36±3.59	93.68±2.72	94.70±2.05
FiCo + DGAN	90.43±6.04	93.91±3.31	94.67±2.94	95.31±2.08
FiCo + DGAN + OF(ours)	**91.43±6.04**	**95.35±2.90**	**96.05±1.87**	**96.47±1.42**

When smaller α is selected, the embedding distribution tends to be more aligned, which makes it difficult for the model to distinguish inter-class similar samples. When it gets bigger, the embedding distribution tends to be more uniform, which will strongly push the intra-class similar samples to break the semantic information. Therefore, setting α to 1.0 is the best choice to balance the uniformity and alignment of the embedding distribution.

3) Effects of Temperature Function $\tau(t)$: Temperature function $\tau(t)$ plays an important role in the unsupervised representation learning of the proposed method. To verify and analyze the effectiveness of temperature function, it is compared with temperature constants. The temperature constants are set to 1.0, 2.0, 3.0, 4.0, and 5.0, respectively. In addition to the temperature function used in the proposed method, there are the linear [see (7)], exponential [see (8)], and logarithmic [see (9)] temperature functions for comparison. The experimental results are shown in Table XIII. In the comparison among temperature constants, it can be observed that the performance increases with the rise of τ. When τ is larger than 3, the performance starts to decrease. This is because, for larger or smaller τ, the contrast strength becomes more extreme. As a result, the semantic information of the defect samples is damaged. Therefore, τ is set to 3.0 as the initial temperature of the temperature function and the temperature function is varied from 3.0 to 4.0. From this table, it can be observed that the performance of the temperature function is better than the constant temperature except for $\tau(t)_3$, indicating that the appropriate temperature variation is more conducive to the representation learning of steel defects. In the comparison of $\tau(t)$ and $\tau(t)_3$, the conclusion can be obtained that the continuous temperature changes in the early stages of training will hurt the performance. Thus, the temperature is set as a constant for the first 200 training epochs, and then the temperature varies with the training epochs. In the comparison of temperature functions, the proposed method achieves the best performance, which indicates that the logarithmic temperature function is more favorable for learning the representation of steel defects.

4) Effects of DGAN: DGAN is an essential component of the proposed method. In this section, the influence of different GAN methods is estimated in terms of image quality and representation learning of FiCo. The image quality results are shown in Table XIV, and DGAN achieves lowest FID score. It reflects that compared to other GAN methods, the images generated by DGAN have the best performance in terms of diversity and quality. The recognition results are displayed in Table XV; it can be seen that DGAN outperforms other GAN methods. The reason is that diverse images promote the quality of representation learning and thus classification accuracy is improved.

TABLE XVI
LE With DWV Loss and FR Loss Under NEU-CLS Few-Label Databases

Method	1-label	2-label	3-label	4-label
DWV loss	86.22±6.95	92.01±4.14	93.51±3.26	94.61±2.65
FR loss(ours)	**91.43±6.04**	**95.35±2.90**	**96.05±1.87**	**96.47±1.42**

5) Effects of OFs: In this part, we conduct experiments to verity the effectiveness of OF. As can be seen from Tables XIV and XV, OFs have a positive contribution to image quality and representation learning of FiCo. The reason is that OF contain semantic information, which can guide DGAN to generate better images and thus facilitate representation learning.

6) Effects of FR Loss: In our method, the proposed FR loss replaces the DWV loss. To verify the merits of the proposed FR loss, we compare it with DWV loss. As can be seen from Table XVI, FR loss achieves better performance. This indicates that the model can learn more useful semantic information from the steel surface defect images under the guidance of FR loss.

7) Effects of Number of Generator Weights: In the proposed method, multiple generator weights are used by DGAN to provide diverse images for FiCo. For testing the effects of the number of generator weights, the performance of proposed method with different numbers of generator weights is shown in Table XVII. From this table, it can be observed that as the number of generator weights increases from 100 to 400, the recognition rate of model gradually increases. When the number of generator weights is over 400, the recognition rate decreases. The reason is that the generated defect data size is much larger than the real defect data size, making the generated defect data occupy a major part of the representation learning. When the model with learned representation is classified on a real defect test set, the recognition rate will decrease to some extent. Therefore, the number of generator weights is set to 400.

8) Effects of Weight Coefficient α_f: The weight coefficient α_f is an important parameter in FR loss. To test the effect of weight coefficient, the performance of proposed method with different α_f values is presented in Table XVIII. It can be seen that the recognition accuracy sharply increases when α_f increases from 0.6 to 1.0. When α_f grows larger than 1.0,

TABLE XVII
LE WITH DIFFERENT NUMBERS OF GENERATOR WEIGHTS UNDER NEU-CLS 4-LABEL DATABASE

Number of generator weights	Accuracy
100	94.96±1.50
200	95.33±1.67
300	95.84±1.44
400	**96.47±1.42**
500	95.85±1.71
600	95.40±1.62

TABLE XVIII
LE WITH DIFFERENT α_f VALUES UNDER NEU-CLS FEW-LABEL DATABASES

α_f	1-label	2-lable	3-lable	4-label
0.6	87.69±6.03	92.04±3.40	93.69±2.65	94.73±2.10
0.8	89.61±6.86	93.70±2.89	94.56±2.49	95.36±2.02
1.0	**91.43±6.04**	**95.35±2.90**	**96.05±1.87**	**96.47±1.42**
1.2	91.07±6.02	94.89±2.98	95.79±2.19	96.35±1.78

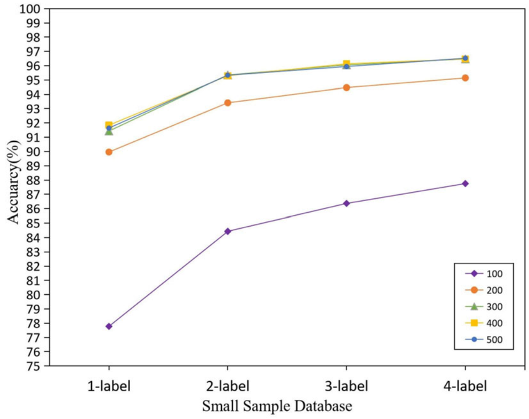


Fig. 11. LE with different training epochs under NEU-CLS few-label databases.

the recognition accuracy decreases slightly. Therefore, α_f is chosen as 1.0 in the proposed method.

9) Effects of Iterations of FDCL: To test the effect of iterations of FDCL on the performance of the model, FDCL is trained 500 epochs. The experimental results are shown in Fig. 11; it can be seen that the model performance improves significantly when the epoch ranges from 100 to 300. When epoch is over 300, the model performance tends to stabilize. Hence, the iterations of FDCL are set to 300.

V. CONCLUSION

In this article, FDCL is presented. It refers to a novel contrastive learning framework for steel surface defect recognition with few labeled samples. This method solves the following two problems of contrastive learning in steel surface defect images: inappropriate contrast strength among defect samples and insufficient steel surface defect data for unsupervised learning. In the proposed method, diverse images are generated by DGAN and provided to FiCo for representation learning. The contrast strength among samples is flexibly adjusted by FiCo to improve the quality of representation learning. Through experiments, it can be proved that FDCL can maintain high accuracy even provided with few labeled samples. Moreover, it is robust to noise and blur disturbance. Therefore, it is meaningful for efficient industrial production to deploy this framework in AVI instruments.

REFERENCES

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

[2] X. Cheng and J. Yu, "RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

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

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

[5] M. Chu, R. Gong, S. Gao, and J. Zhao, "Steel surface defects recognition based on multi-type statistical features and enhanced twin support vector machine," *Chemom. Intell. Lab. Syst.*, vol. 171, pp. 140–150, Sep. 2017.

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

[7] S. Zhou, Y. Chen, D. Zhang, J. Xie, and Y. Zhou, "Classification of surface defects on steel sheet using convolutional neural networks," *Mater. Tehnol.*, vol. 51, no. 1, pp. 123–131, Jan. 2017.

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

[9] W. Chen, Y. Gao, L. Gao, and X. Li, "A new ensemble approach based on deep convolutional neural networks for steel surface defect classification," *Proc. CIRP*, vol. 72, pp. 1069–1072, Jan. 2018.

[10] D. He, K. Xu, P. Zhou, and D. Zhou, "Surface defect classification of steels with a new semi-supervised learning method," *Opt. Lasers Eng.*, vol. 117, pp. 40–48, Jun. 2019.

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

[12] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 929–940, Mar. 2018.

[13] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 9729–9738.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, Jul. 2020, pp. 1597–1607.

[15] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[16] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jun. 2006, pp. 1735–1742.

[17] Y. Zhai et al., "Weakly contrastive learning via batch instance discrimination and feature clustering for small sample SAR ATR," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5204317.

[18] J. Deng et al., "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 20–25.

[19] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 2672–2680.

[20] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proc. Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017, pp. 1–17.

[21] Y. Gao, L. Gao, X. Li, and X. V. Wang, "A multilevel information fusion-based deep learning method for vision-based defect recognition," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 3980–3991, Jul. 2020.

[22] Y. Wang, L. Gao, X. Li, Y. Gao, and X. Xie, "A new graph-based method for class imbalance in surface defect recognition," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–16, 2021.

[23] Y. Gao, L. Gao, X. Li, and X. Yan, "A semi-supervised convolutional neural network-based method for steel surface defect recognition," *Robot. Comput.-Integr. Manuf.*, vol. 61, Feb. 2020, Art. no. 101825.

[24] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–13.

[25] J. B. Grill et al., "Bootstrap your own latent—A new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–14.

[26] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15745–15753.

[27] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[28] A. Ghosh, V. Kulharia, V. Namboodiri, P. H. S. Torr, and P. K. Dokania, "Multi-agent diverse generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 8513–8521.

[29] I. M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, 2017, pp. 214–223.

[30] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, Sydney, NSW, Australia, 2017, pp. 5767–5777.

[31] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Honolulu, HI, USA, Oct. 2017, pp. 2813–2821.

[32] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 9929–9939.

[33] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2495–2504.

[34] Z. D. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 1–6.

[35] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4401–4410.

[36] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," *Appl. Surf. Sci.*, vol. 285, no. 21, pp. 858–864, Nov. 2013.

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

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

[39] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 3733–3742.

[40] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 6626–6637.

[41] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–13.

[42] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11557–11568.

[43] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, "SimMatch: Semi-supervised learning with similarity matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 14451–14461.

[44] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, May 2018, pp. 1–16.

[45] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 6210–6219.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–14.

[47] M. Tan and Q. V. Le, "EfficientNetV2: Smaller models and faster training," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2021, pp. 10096–10106.

[48] G. Huang et al., "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269.

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

[50] N. Ma, X. Zhang, H. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 649–666.

[51] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, Oct. 2019, pp. 1314–1324.

Ying Xu received the B.S. and M.S. degrees in automation, and control theory and control engineering from the Wuhan University of Science and Technology, Wuhan, China, in 2004 and 2008, respectively, and the Ph.D. degree in control theory and control engineering from the South China University of Science and Engineering, Guangzhou, China, in 2013.

She is currently an Associate Professor with the School of Department of Intelligence Manufacturing, Wuyi University, Jiangmen, Guangdong, China. Her current research interests include image processing, deep learning, and biometric extraction.

Jiacong Chen received the B.S. degree in mechanical engineering from Wuyi University, Jiangmen, China, in 2019, where he is currently pursuing the M.S. degree in mechanical engineering with the Department of Intelligence Manufacturing.

His current research interests include image processing and pattern recognition.

Yanyang Liang received the B.S. degree in automation from the Southwest University of Science and Technology, Mianyang, China, in 2002, and the Ph.D. degree in control theory and control engineering from the University of Science and Technology of China, Hefei, China, in 2008.

He is currently an Associate Professor with the School of Department of Intelligence Manufacturing, Wuyi University, Jiangmen, Guangdong, China. His current research interests include computer vision, motion controls, and robust control for nonlinear systems.

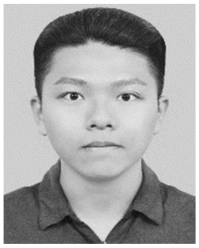
Yikui Zhai (Member, IEEE) received the bachelor's degree in optical electronics information and communication engineering from Shantou University, Shantou, China, in 2004, the master's degree in signal and information processing from Shantou University, in 2007, and the Ph.D. degree in signal and information processing from Beihang University, Beijing, China, in June 2013.

Since October 2007, he has been working with the Department of Intelligence Manufacturing, Wuyi University, Jiangmen, China, where he is an Associate Professor. He has been a Visiting Scholar with the Department of Computer Science, University of Milan, Milan, Italy, since 2016. His research interests include image processing, deep learning, and pattern recognition.

Zilu Ying received the B.S., M.S., and Ph.D. degrees in electrical information engineering from Beihang University, Beijing, China, in 1985, 1988, and 2009, respectively.

He is a Full Professor with Wuyi University, Jiangmen, China. He is an Executive Director of the Guangdong Society of Image and Graphics. His research interests include biometric extraction and pattern recognition.

Mr. Ying is a member of the Signal Processing Branch of the Chinese Institute of Electronics.



Wenlve Zhou received the B.S. degree from the Beijing Institute of Technology, Zhuhai, China, in 2019. He is currently pursuing the master's degree with the Department of Intelligence Manufacturing, Wuyi University, Jiangmen, China.

His research interests include biometric extraction and pattern recognition.

Angelo Genovese (Member, IEEE) received the Ph.D. degree in computer science from the Università degli Studi di Milano, Milan, Italy, in 2014.

He has been a Post-Doctoral Research Fellow in computer science with the Università degli Studi di Milano, since 2014. He has been a Visiting Researcher with the University of Toronto, Toronto, ON, Canada. Original results have been published in over 30 articles in international journals, proceedings of international conferences, books, and book chapters. His current research interests include signal and image processing, three-dimensional reconstruction, computational intelligence technologies for biometric systems, industrial and environmental monitoring systems, and design methodologies and algorithms for self-adapting systems.

Dr. Genovese is an Associate Editor of the *Journal of Ambient Intelligence and Humanized Computing* (Springer).

Vincenzo Piuri (Fellow, IEEE) received the M.S. and Ph.D. degrees in computer engineering from Politecnico di Milano, Milan, Italy, in 1984 and 1988, respectively.

He was the Department Chair at the University of Milan, Milan, from 2007 to 2012, where he has been a Full Professor, since 2000. He was an Associate Professor with Politecnico di Milan, from 1992 to 2000, a Visiting Professor with The University of Texas at Austin, Austin, TX, USA, from 1996 to 1999, and a Visiting Researcher with George Mason University, Fairfax, VA, USA, from 2012 to 2016. He founded a startup company, SENSURE SRL, Bergamo, Italy, in the area of intelligent systems for industrial applications (leading it from 2007 to 2010) and was active in industrial research projects with several companies. His main research and industrial application interests include intelligent systems, computational intelligence, pattern analysis and recognition, machine learning, signal and image processing, biometrics, intelligent measurement systems, industrial applications, distributed processing systems, Internet-of-Things, cloud computing, fault tolerance, application-specific digital processing architectures, and arithmetic architectures.

Dr. Piuri is an ACM Fellow.

Fabio Scotti (Senior Member, IEEE) received the Ph.D. degree in computer engineering from the Politecnico di Milano, Milan, Italy, in 2003.

He has been an Associate Professor of computer science with the Universitá degli Studi di Milano, Milan, since 2015. His original results have been published in more than 100 articles in international journals, proceedings of international conferences, books, book chapters, and patents. His research interests include biometric systems, machine learning and computational intelligence, signal and image processing, theory and applications of neural networks, 3-D reconstruction, industrial applications, intelligent measurement systems, and high-level system designs.

Dr. Scotti is an Associate Editor with the IEEE TRANSACTIONS ON HUMAN–MACHINE SYSTEMS and *Soft Computing* (Springer). He has been an Associate Editor with the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and a Guest Coeditor for the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.