TYPE Original Research
PUBLISHED 30 May 2023
DOI 10.3389/fphy.2023.1208781

OPEN ACCESS

EDITED BY
Guanqiu Qi,
Buffalo State College, United States

REVIEWED BY
Jian Sun,
Southwest University, China
Zhiqin Zhu,
Chongqing University of Posts and Telecommunications, China
Yong Li,
Chongqing University, China

*CORRESPONDENCE
Yong Xu,
yongxu@ymail.com

RECEIVED 19 April 2023
ACCEPTED 09 May 2023
PUBLISHED 30 May 2023

CITATION
Hao L, Shen P, Pan Z and Xu Y (2023), Multi-level semantic information guided image generation for few-shot steel surface defect classification.
*Front. Phys.* 11:1208781.
doi: 10.3389/fphy.2023.1208781

# Multi-level semantic information guided image generation for few-shot steel surface defect classification

Liang Hao[1], Pei Shen[2], Zhiwei Pan[2] and Yong Xu[1]*

[1]School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China, [2]HBIS Digital Technology Co., Ltd., Shijiazhuang, China

Surface defect classification is one of key points in the field of steel manufacturing. It remains challenging primarily due to the rare occurrence of defect samples and the similarity between different defects. In this paper, a multi-level semantic method based on residual adversarial learning with Wasserstein divergence is proposed to realize sample augmentation and automatic classification of various defects simultaneously. Firstly, the residual module is introduced into model structure of adversarial learning to optimize the network structure and effectively improve the quality of samples generated by model. By substituting original classification layer with multiple convolution layers in the network framework, the feature extraction capability of model is further strengthened, enhancing the classification performance of model. Secondly, in order to better capture different semantic information, we design a multi-level semantic extractor to extract rich and diverse semantic features from real-world images to efficiently guide sample generation. In addition, the Wasserstein divergence is introduced into the loss function to effectively solve the problem of unstable network training. Finally, high-quality defect samples can be generated through adversarial learning, effectively expanding the limited training samples for defect classification. The experimental results substantiate that our proposed method can not only generate high-quality defect samples, but also accurately achieve the classification of defect detection samples.

KEYWORDS

few-shot steel surface defect classification, adversarial learning, residual module, multi-level semantic feature extractor, Wasserstein divergence

## 1 Introduction

Steel is an essential material for industrial production, with a broad range of uses in areas such as automobile, aerospace and machinery. As the demand for material fitness in various industries increases, the surface quality of steel has become increasingly important. However, during the steel manufacturing process, due to the influence of various unstable factors such as raw materials and production conditions, various types of defects may appear on the surface of steel, which affect the quality of steel to varying degrees and easily lead to serious production accidents, resulting in immeasurable losses to producer and users [1, 2]. Thus, it is of great importance to classify the defects on the surface of steel efficiently for further quality enhancement.

Generally, steel surface defects belonging to the same category meet a large intra-class difference, while those of different categories are highly similar [3], making the classification of steel surface defects more complicated. To address this problem, various approaches have been studies. For instance, Zaghdoudi et al. [4] proposed a steel surface defect classification method based on the binary Gabor pattern (BGP) algorithm and support vector machine (SVM). Hu et al. [5] extracted various visual features such as geometry, texture, and shape of the defect image and fed them to SVM for classification. Despite the fact that these methods do classify different defects, these hand-crafted features are not optimal, making a constraint on the further performance improvement. Fortunately, thanks to the development of deep learning, deep learning based methods have attracted much attention in the field of steel surface defect classification due to it powerful capability in feature extraction. Specifically, Duan et al. [6] used RGB images and gradient images as inputs to a dual-flow convolutional neural network, and fused multi-source information to recognize aluminum surface defects. Liu et al. [7] proposed an improved dual CNN model fusion framework, which uses pre-trained VGG16 and AlexNet to extract different features from the input source to classify and identify aluminum surface defects.

Although deep learning based methods enjoy superiority compared with conventional methods, they also meet the limitation on the large scale of training data. However, the number of non-defective samples in actual industrial production environments is far greater than that of defective samples. Moreover, it is difficult to identify and collect defective samples, further leading to an insufficient number of samples [8, 9, 10]. To address this issue of insufficient samples, many researchers have begun to focus on the unsupervised data enhancement algorithm: Generative Adversarial Networks (GANs). Currently, many improved GANs and adversarial learning strategies have been derived, such as Wasserstein GAN (WGAN) [11], Deep Convolutional GAN (DCGAN) [12], and ACGAN [13]. These generative models augment the original data by generating synthetic samples, thereby mitigating the effect of few-shot on the classification performance and improving the accuracy. Dosovitskiy et al. [14] showed that even with low-fidelity images, the performance can be significantly improved. If the generated images enjoy the high-quality, the over-fitting problem can further be solved [15]. However, despite the wide application of GANs and its related improved models, there are still some tough difficulties, such as insufficient model feature capture capability, gradient disappearance, and model collapse, etc.

Furthermore, generating high-quality data similar to the original data distribution can solve the over-fitting problem, and enhance the detection accuracy and generalization ability of the model [15]. Lu and Su [16] proposed a novel method to eliminate mura patterns from defect images by using conditional generation adversarial networks; Li et al. [17] studied a cross-domain fault diagnosis method based on deep neural networks, which has a good industrial application prospect; Liu et al. [18] introduced an attention mechanism into feature extraction, proposed a structural defect detection framework based on GAN-CNN, and achieved satisfactory results. Despite the wide application of GANs and its related improved models, there are still some tough difficulties, such as insufficient model feature capture capability, gradient disappearance, and model collapse, etc.

Aimed at above problems, we propose a steel surface defect classification method based on residual adversarial learning with Wasserstein divergence. First, the residual module is introduced into the network framework of adversarial learning, to enhance the feature extraction ability of the model and improve the quality of generated samples. Subsequently, to extract semantic information from defect samples at different levels, we design a multi-level semantic feature extractor (MSFE), which guides sample generation by extracting the most relevant semantic features from images. Then Wasserstein divergence is used to alleviate gradient disappearance, gradient explosion and mode collapse during model training. Finally, high-quality samples are generated, and few-shot steel surface defect classification is realized by adversarial learning. The experimental results show that the proposed method improves the accuracy of steel surface defect classification, which are superior to many state-of-the-arts.

The main contributions of this paper are as follows:

- The residual module is introduced into the network structure of adversarial learning to contribute to the feature extraction. Moreover, multiple convolutional layers are employed in the model architecture to replace the original classification layer, further boosting the classification performance of the model.
- A multi-level semantic feature extractor (MSFE) which effectively extracts features at different levels is designed, fully capturing diverse semantic information of images to guide the generator in sample generation and improve the quality of generated samples.
- The proposed method can generate high-quality samples to compensate for the deficiencies under few-shot conditions, further improving the classification performance.

# 2 Related works and preliminary knowledge

## 2.1 Steel surface defect classification

Steel surface defect classification based on deep learning has gained considerable attention in recent years and achieved remarkable results. Chenon et al. [19] proposed a defect classification approach based on a single convolutional neural network, which can extract effective features for defect classification without the prior of hand-crafted features. Nakazawa et al. [20] proposed a method for surface defect classification and image retrieval using convolutional neural networks. The model was trained, validated, and tested using generated data samples, and it was demonstrated that the model trained by synthetic data can be classified efficiently. Zhu et al. [21] studied an intelligent identification algorithm based on convolutional neural networks and random forest algorithms, which enabled the intelligent identification of weld surface defects. However, obtaining effective defect samples is very challenging in the actual industrial environment, and there is the problem of insufficient samples, which leads to a low performance of the surface defect classification model based on deep learning. Therefore, data augmentation and transfer learning have been proposed by many researchers to address the few-shot problem in this field. Wan et al. [22] studied an improved VGG19 neural network based on small samples and unbalanced datasets for strip

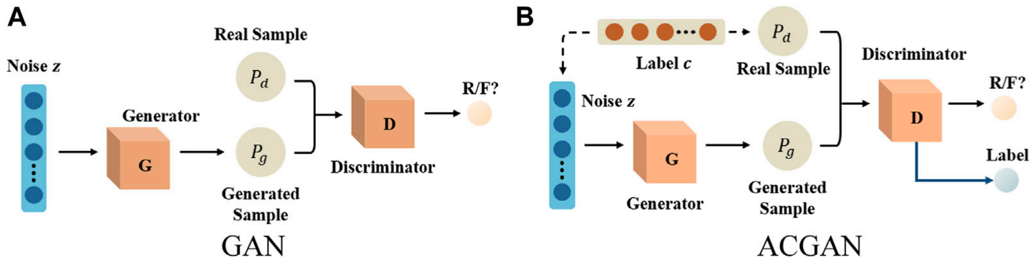

FIGURE 1
The model framework of GAN and ACGAN. **(A)** GAN. **(B)** ACGAN.

steel defect detection. Through fast image preprocessing algorithms and transfer learning theory, excellent results have been achieved on multiple datasets. Han et al. [23] proposed a new framework for intelligent fault diagnosis, namely, Deep Transfer Network (DTN), which generalized deep learning models to domain self-adaptation scenarios. By using the discriminative structure associated with the labeled data in the source domain to adapt to the unlabeled data, more accurate distribution matching is ensured. Furthermore, Liu et al. [24] designed ImDeep, a deep learning model for unbalanced multi-label surface defect classification, which combines three key technologies to improve the classification performance of the model: imbalanced sampler, Fussy-FusionNet, and transfer learning.

Apart from the domain adaptation, some scholars also utilize GAN as a data augmentation technique to address few-shot issue. Goodfellow et al. [15] first proposed the unsupervised deep learning model GAN in 2014, which was inspired by the two-player zero-sum game in game theory and consists of two components: the generator and the discriminator. The generator is mainly responsible for generating data that is as similar as possible to the original data samples, while the discriminator is tasked with distinguishing between real and fake images. Currently, GAN has been widely applied in various fields, such as image generation, data augmentation, image restoration, and image coloring. Specifically, Jain et al. [25] trained three GAN architectures to generate synthetic images for data augmentation, which significantly improved the performance of surface defect classification. He et al. [26] proposed a semi-supervised learning for defect classification based on GAN and ResNet to expand the training samples and exploit the unlabeled images. Zhao et al. [27] designed a reconstruction network to reconstruct the potential defect areas in the sample image, and determine the final defect area according to the difference between the reconstructed sample and the original sample. Lian et al. [28] proposed a novel machine vision method for automatic identification of tiny defects in a single image. To effectively achieve pixel-level defect detection on textured surfaces without manual annotation, Tsai et al. [29] introduced a two-stage deep learning scheme. Particularly, the first stage used CycleGAN to automatically synthesize and annotate the pixels of defect in images. The second stage used the synthesized defect images and their corresponding annotation results as input-output pairs for training the U-Net semantic network.

## 2.2 Preliminary knowledge

GAN consists of a generator and a discriminator [15], as shown in Figure 1A. The input of the generator is a random noise vector z, and the output is the fake sample generated by it. The discriminator uses the fake sample generated by the generator and the real data x as the input, and the output is the discrimination score of the discriminator on the fake sample. GAN's overall objective function is:

$$\min_G \max_D L(G,D) = E_{x\sim P_d}[\log D(x)] + E_{z\sim P_z}[\log(1 - D(G(z)))] \tag{1}$$

where $P_d$ is the probability density distribution of the real data $x$; $z$ is the noise vector randomly sampled from the prior distribution $P_z$; $G$ represents the generator, $D$ represents the discriminator, and $E(\cdot)$ represents the calculated expected value; $D(X)$ is a probability distribution, that is, the probability of classifying data $X$ as a real sample, and $X$ is derived from a real sample $x$ or a generated sample $G(z)$.

Formula 1 shows that the optimization problem of GAN is same as the max-min optimization problem, which includes the optimization goals of the generator and the discriminator. The main function of the discriminator is to perform binary classification on the input data to determine whether the input data comes from the distribution of the real data or the generated pseudo data. Thus, its objective function is:

$$\max_D L(G,D) = E_{x\sim P_d}[\log D(x)] + E_{z\sim P_z}[\log(1 - D(G(z)))] \tag{2}$$

It can be seen from Formula 2 that the goal of the discriminator is to maximize the discrimination accuracy for the data. In other words, we aim to maximize the discriminant result $D(x)$ for the real data $x$, and minimize the result $D(G(z))$ of the generated sample $G(z)$ (maximize $1 - D(G(z))$).

The purpose of the generator is to generate samples that the discriminator cannot distinguish as false, and its objective function is:

$$\min_G L(G,D) = E_{z\sim P_z}[\log(1 - D(G(z)))] \tag{3}$$

The generator is optimized by Eq. 3. Specifically, the probability score $D(G(z))$ of the discriminator for the generated sample $G(z)$ is maximized $(1 - D(G(z))$ is minimized). During training, the alternate optimization methods are used: fix one side and update the parameters of the other network. In other words, the model updates the discriminator's parameters firstly through the fixed generator so that the discriminator maximizes the discriminant result. Then we fix discriminator's parameters for updating the generator, which minimize the result that discriminator works. Finally, when the probability distribution $P_g$ of the samples generated by the generator $G$ is infinitely close to the probability distribution $P_d$ of the real samples (that is, $P_g = P_d$), the global optimal solution can be reached.

ACGAN is a variant of GAN [13], and its structure is illustrated in Figure 1B. By incorporating auxiliary label information c into the generator, the generated samples can be constrained to possess certain characteristics, thus allowing for more precise expression of the samples and the generation of specific samples according to it. Moreover, in order to ensure accurate classification, ACGAN adds a softmax layer to the discriminator network, thus enabling the improved model to not only judge the authenticity of the data, but also classify the input samples.

The loss function of ACGAN consists of two parts: the discriminative loss $L_s$ and the classification loss $L_c$. The role of discriminative loss is to judge the authenticity of the generated samples, thereby improving the quality of the samples generated by the generator. The role of the classification loss is to measure the accuracy of the classification of the sample category. And, the specific calculation of $L_c$ is:

$$L_c = E_{x\sim P_d}[R(x|c_x)] + E_{z\sim P_z,\, c\sim P_c}[R(G(z,c)|c)] \tag{4}$$

where $R$ is the cross-entropy loss function, $c_x$ represents the category label of the real data $x$, $c$ is the category label of the generated data $G(z,c)$, and $P_c$ is the category label distribution of the sample.

Since a classifier is added to the discriminator $D$, the network can not only distinguish the authenticity of the data, but also classify the data, so its loss function needs to calculate two parts: discriminant loss $L_s(D)$ and classification loss $L_c$. The specific calculation is as follows.

$$L_s(D) = E_{x\sim P_d}[\log D(x)] + E_{z\sim P_z,\, c\sim P_c}[\log(1 - D(G(z,c)))] \tag{5}$$

$$L(D) = L_c + L_s(D) \tag{6}$$

Similarly, the loss function of the generator $G$ also needs to consider the classification loss:

$$L_s(G) = E_{z\sim P_z,\, c\sim P_c}[\log(1 - D(G(z,c)))] \tag{7}$$

$$L(G) = L_c - L_s(G) \tag{8}$$

Formulas 6, 8 ultimately constitute the entire loss function of the ACGAN model. During the training process, the model is continually optimized to enhance the quality of the samples generated by the model and augment the classification accuracy of the model.

# 3 Methods

Although Generative Adversarial Networks (GANs) and Auxiliary Classifier GANs (ACGANs) can effectively alleviate the few-shot classification problem by generating samples, they still meet the limitation on inadequate information extraction capabilities, gradient vanishing, and pattern collapse. To address these issues, we propose a novel network structure. Specifically, a residual adversarial learning model with Wasserstein divergence based on ACGAN under multi-level semantic guidance is proposed, as shown in Figure 2.

First, the random noise vector $z$ and sample label $c$ are input into the generator. The generator generates synthetic samples $I_g$, expanding the scale of the training data. By utilizing a multi-level semantic feature extractor to process original samples, semantic and contextual information can effectively be captured and used for guiding sample generation of generator. Then, the discriminator takes the generated sample $I_g$ and real sample $I$ as the inputs, and outputs the discriminant result $R/F?$ (True or Fake) and the classification result $c'$ of the generated sample. During the adversarial training of model, the Wasserstein divergence $(W\_div)$ is used as the distance measurement between the distributions of the initial data and the distributions of the generated data.

## 3.1 The modification of network

Despite the fact that ACGAN achieved significantly satisfactory results in image generation [13], it still faces the problem of insufficient feature extraction ability when it is applied to tasks within the few-shot environment, resulting in inadequate acquisition of image information and a consequent decrease in model performance. To address this issue, the overall network structure of ACGAN is optimized, as illustrated in Figure 3. The specific improvements of the network structure are detailed below.

(1) As shown in Figure 3, the residual module (Residual) is introduced into the network structure of the generator and the discriminator to optimize the feature learning ability of the model, so that the model can extract more valuable features. Meanwhile, it can ensure the quality of the samples generated by the model while optimizing the model's ability to discriminate and classify images. The specific network structure of the introduced residual module is shown in Figure 4.

(2) When the kernel size of the deconvolution layer cannot be divisible by stride in the actual calculation, uneven overlapping problems will occur. Also, the generated sample images would have some checkerboard-like artifacts [30]. Therefore, in order to avoid such problems, as shown in Figure 3A, the up-sampling layer (US) and the convolutional layer (Conv) are used to generate sample images in the generator network structure. As shown in Figure 3B, in the discriminator network structure, two convolutional layers are added before the sigmoid and softmax classification layers, which makes the classifier in the discriminator learn more image information and improve the classification performance.

(3) The generator network mainly consists of several residual modules and convolutional layers as well as operating up-sampling layers. The input of the model is the randomly generated 128-dimensional vector z and the sample label c, which undergoes a fully connected layer (FC) and the reshape (reshape) operation. Before the convolution calculation, the first two convolution layers have

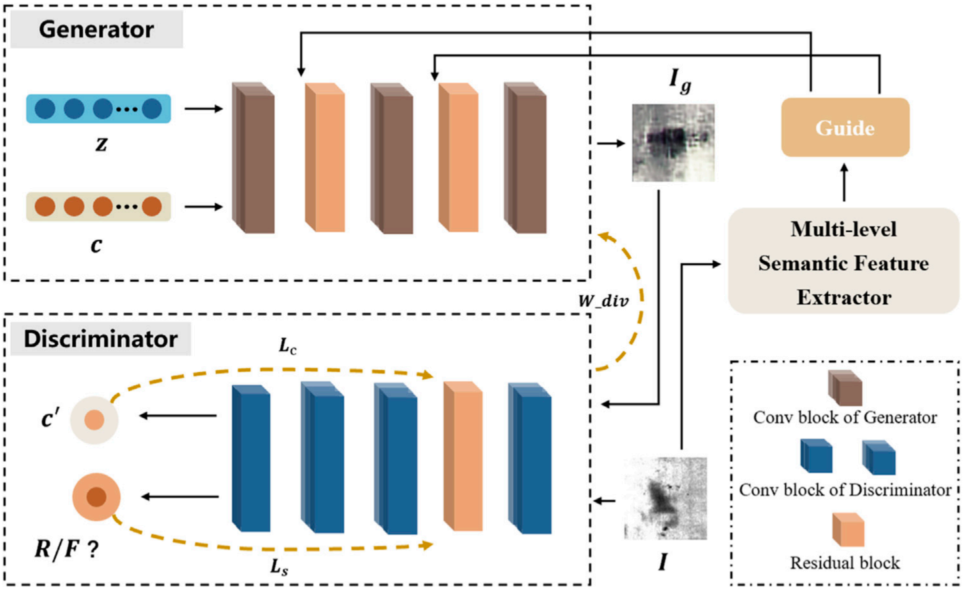

FIGURE 2
Overview of our framework. Given an image $I$ as input, our framework first extracts rich semantic information through multi-level semantic feature extractor to guide generator. After that, we deliver the noise $z$ and label $c$ to generator for generating sample $I_g$. Finally, we can obtain the classification result $c'$ and the discriminate result $R/F$? (True or Fake?) of generated sample $I_g$ by discriminator. $L_c$, $L_s$, and $W\_div$ indicates respectively the classification loss, the discriminant loss, Wasserstein divergence during training.

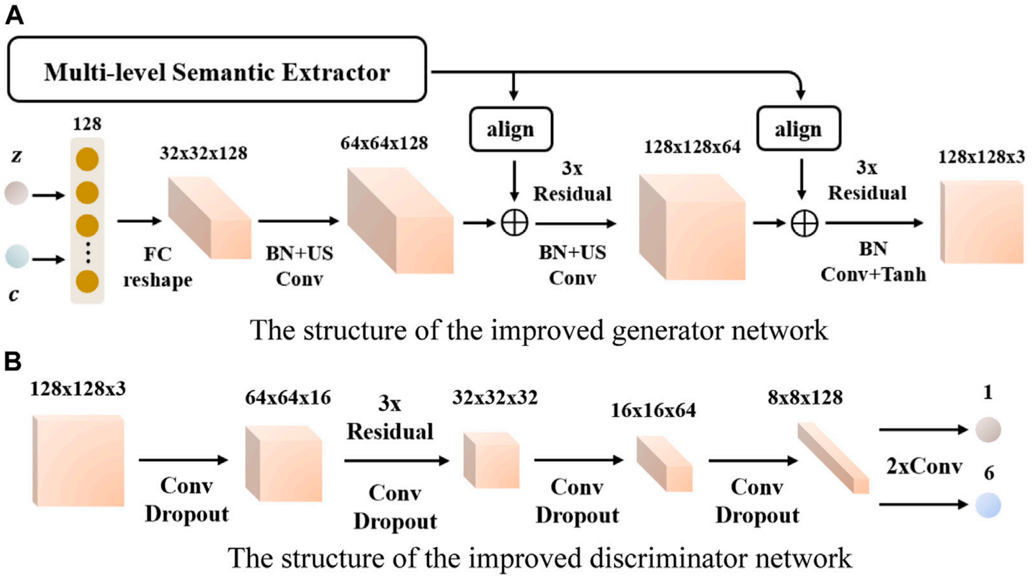

FIGURE 3
Improved generator and discriminator network structure. **(A)** The structure of the improved generator network. **(B)** The structure of the improved discriminator network.

both performed the up-sampling operation using the nearest neighbor interpolation, which increases the feature map by two times. At the same time, Batch Normalization (BN) is used to optimize the network throughput the training. There are three residual modules between each two convolution layers to improve the feature learning ability of the model, and the Leaky-ReLU activation function is used between each layer. The discriminator network also includes 6 convolutional layers and 3 residual modules. And the 3 residual modules follow the first convolutional layer. At the same time, a Dropout layer (Dropout) is further introduced to prevent overfitting problems. Furthermore, the Leaky-ReLU activation function is used between each layer.

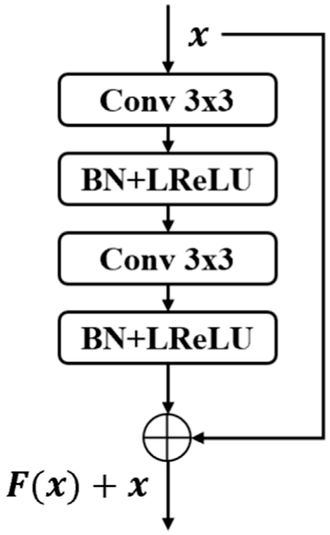

FIGURE 4
The specific network structure diagram of the residual block.

## 3.2 Multi-level semantic feature extractor

Recently, ACGAN has achieved remarkable progress in the field of image generation. Given a category label, ACGAN can map random noise into high-resolution images with abundant texture features and comprehensive shape details. However, satisfactory results depend on training ACGAN with sufficient quantity of samples. When there is an inadequate number of samples, the effectiveness of ACGAN in generating samples close to reality is compromised due to its inability to obtain enough semantic information, which motivates us to design a multi-level semantic feature extractor to facilitate sample generation tasks, as shown in Figure 2. As illustrated above, the role of the multi-level semantic feature extractor is to extract the semantic and contextual information of defects at different levels in the image. Therefore, the original samples are input into the multi-level semantic feature extractor to obtain the learned hierarchical features, such as texture and shape, which are then incorporated into the generator to serve as guidance for sample generation. Specifically, the sample image I corresponding to the true sample label c is processed by the multi-level semantic feature extractor to obtain rich semantic information, which is then aligned with the different convolutional layers in the generator, facilitating better integration of multi-level semantic features into the process of sample generation. The core of alignment operation mainly relies on a convolutional layer, which adjusts semantic features extracted by MSFE to the size corresponding to different layers of the generator, and adds the adjusted features to the original ones to obtain new features under semantic guidance. The aligned features are added to the convolutional layers of the generator, leveraging diverse levels of semantic features to facilitate the generation of specific defective samples, as depicted in Figure 3A. We use VGG19 pretrained on the ImageNet dataset as a multi-level semantic feature extractor, and use the features extracted from layers 7 to 23 in it to guide the generator.

## 3.3 Objective function

The Kullback-Leibler (KL) divergence [15] is prone to gradient instability in the Generative Adversarial Networks (GANs) training phase, and can also lead to mode collapse. To address these issues, the Wasserstein GAN (WGAN) uses the Wasserstein distance to ensure that the gradient of the model is continuous during the training process [11]. However, WGAN utilizes weight clipping to restrict the weights within a fixed range strictly, which greatly limits the expressiveness of the network. Consequently, WGAN-GP [31] adopts gradient penalty to enhance the stability of the network training. According to the research conducted by [32], in experiments, WGAN-GP typically employs the technique of interpolating between real and fake data to simulate a uniform distribution across the whole space. This approach is somewhat mechanistic and empirical, which makes it challenging to simulate the full spatial distribution using limited sampling.

In order to solve this problem, Wu et al. [32] proposed Wasserstein divergence to reduce the distance loss function properly between two distributions, as shown in Formula 9. It removes the K-Lipschitz conditional restriction, and changes the penalty term added to the loss function.

$$\begin{aligned} W_{k,p}(P_d, P_z) = \max_{D}\ & E_{x\sim P_d}[D(x)] - E_{z\sim P_z}[D(z)] \\ & -kE_{u\sim P_u}\left[\|\nabla D(u)\|^p\right] \end{aligned} \tag{9}$$

where k and $p$ are selected empirically. Generally, k = 2, $p = 6$. $\nabla$ represents the gradient. $x$ comes from the distribution $P_d$ of the real data; similarly, $z$ comes from the generated sample distribution $P_z$. $P_u$ is a distribution derived from the real data distribution $P_d$ and the generated data distribution $P_z$. $D$ represents the discriminator, and $E(\cdot)$ represents the calculated expected value. Experiments in [32] prove that all different distributions have improved performance.

Based on the loss function of ACGAN [13], we use Wasserstein divergence to address the potential gradient explosion issue in the training process. Hence, the loss function of our method consists of two parts: the loss function $L(D)$ of discriminator and the loss function $L(G)$ of generator, with each loss function consisting of two components: the adversarial loss function $L_s$ and the conditional loss function $L_c$.

The purpose of $L(D)$ is to ensure that the discriminator can distinguish between real and generated samples and accurately classify them based on their respective conditions, as shown below:

$$L_s(D) = E_{x\sim P_d}[D(x)] - E_{z\sim P_z}[D(z)] - kE_{u\sim P_u}\left[\|\nabla D(u)\|^p\right] \tag{10}$$

$$L_c = E_{x\sim P_d}[R(x|c_x)] + E_{z\sim P_z,\, c\sim P_c}[R(G(z,c)|c)] \tag{11}$$

$$L(D) = L_c + L_s(D) \tag{12}$$

where $L_s(D)$ represents the adversarial loss function that is modified with Wasserstein divergence; $L_c$ is the conditional loss function; $R(\cdot)$ denotes the cross-entropy loss function; $c_x$ indicates the category label of real data sample $x$, and $c$ denotes the category label of generated data $G(z,c)$. $P_c$ represents the distribution of sample class labels. During the training process of discriminator, our objective is to maximize its loss function $L(D)$.

Likewise, the purpose of $L(G)$ is to generate high-quality data samples such that the discriminator cannot distinguish whether the sample is real or fake, as illustrated below:

$$L_s(G) = E_{z\sim P_z}[D(z)] \tag{13}$$

$$L_c = E_{x\sim P_d}[R(x|c_x)] + E_{z\sim P_z,\, c\sim P_c}[R(G(z,c)|c)] \tag{14}$$

$$L(G) = L_c - L_s(G) \tag{15}$$

where $L_s(G)$ represents the adversarial loss function for the generator. Similarly, we aim to maximize its loss function $L(G)$ in the training process.

## 3.4 Network training

During the training process of the model, the discriminator continuously enhances its capability to distinguish between real samples and generated samples, while the generator continuously improves its ability to generate realistic samples. The discriminator updates its weights by utilizing both real and generated samples, and the generator updates its weights through the error feedback from the discriminator. The training process of the model is a maximization and minimization process. In the adversarial training of the discriminator and the generator, the discriminator minimizes the probability of misclassification, and the generator maximizes the error probability of the discriminator. The iterative training method of the generator and the discriminator is employed to prevent the over-fitting of the generator network. The specific training steps of the model are illustrated in Algorithm 1.

```
Data: image dataset
Output: trained Discriminator and Generator, Training Accuracy
for epoch=0 to n do
  randomly sample from real samples and get (real_images, labels), and randomly sample from a uniform distribution to obtain noise $z$
  input ($z$, labels) into Generator to generate sample fake_images
  generated sample fake_images and real sample real_images are fed into discriminator
    calculate the gradient of the real sample space, calculate the gradient of the generated sample space, and calculate the Wasserstein divergence according to Formula 9
  for D_epoch=0 to m do
    calculate Discriminator's loss by Formulas 10, 11, 12
    update Discriminator parameters
  end for
    calculate Generator's loss according to Formulas 13, 14, 15
    update Generator parameters
  end for
```

**Algorithm 1. Residual Adversarial Learning Model with Wasserstein Divergence.**

# 4 Experiments

In order to verify the effectiveness of the proposed method, experiments are conducted on the NEU-CLS dataset using a Windows 10 system with 16 GB of memory, an AMD Ryzen 7 4800HS processor, and an NVIDIA GTX 1660 Ti graphics card. The model is constructed using the PyTorch platform.

## 4.1 Dataset

This paper performs experiments on the NEU-CLS hot-rolled steel surface defect dataset from Northeastern University [34]. The dataset consists of 6 types of defects, and each category contains 300 grayscale images (200 × 200 pixels). These six types of defects are: crazing (Cr), inclusion (In), patches (Pa), pitted surface (PS), rolled-in scale (RS) and scratches (Sc), as illustrated in Figure 5.

In the experiment, the NEU-CLS dataset is divided according to a 2:1 ratio, with 1,200 images used as the training set and 600 images used as the test set. It takes 10,000 epochs to train our network with Adam optimizer and a batch of 64 images. The parameter settings of the model are as follows: learning rate of $\alpha = 0.0002$, random noise vector dimension of z = 128, and Adam optimization parameters of β1 = 0.5 and β2 = 0.999. In addition, we use VGG19 pretrained on the ImageNet dataset as a multi-level semantic feature extractor, and use the features extracted from layers 7 to 23 in it to guide the generator.

## 4.2 Few-shot classification of steel surface defects

Considering the restricted size of the dataset, we conduct experiments with six different training sample sets (200, 150, 100, 50, 30, 10) to evaluate the few-shot classification performance enhancement of the proposed method after training, and to comparatively analyze the impact of the data size on the model. The numbers of test sets are kept constant. The results of the comparison between ACGAN and the method proposed in this paper under different training sample sizes are presented in Table 1.

According to Table 1, it can be observed that the classification performance of ACGAN and the proposed model decreases as the training sample size decreases. It is evident that insufficient samples reduce the generalization capability of the model, resulting in a poorer performance on the test set. Furthermore, the decline of our model is more gradual than that of ACGAN, indicating that the method proposed in this paper is more stable and robust when dealing with few-shot issues. As illustrated in Figure 6, the trend of classification results of ACGAN and our model under different training sample sizes can be observed.

Observing Figures 5, 6, it can be seen that the accuracy of our model has a distinct advantage over ACGAN under different training sample sizes. When the sample size is 200, the average accuracy of our model reaches 98.67%. At the same time, when the training sample size is 10, the average accuracy of the model in this paper is 89.67%, while the accuracy of ACGAN drops to 66.5%. This indicates that ACGAN is more reliant on data. Furthermore, as the training sample size decreases, the classification accuracy gap between ACGAN and the model proposed in this paper increases. When the sample size is 10, the accuracy of ACGAN is 23.17% lower than that of the method proposed in this paper, making it evident that ACGAN is far less

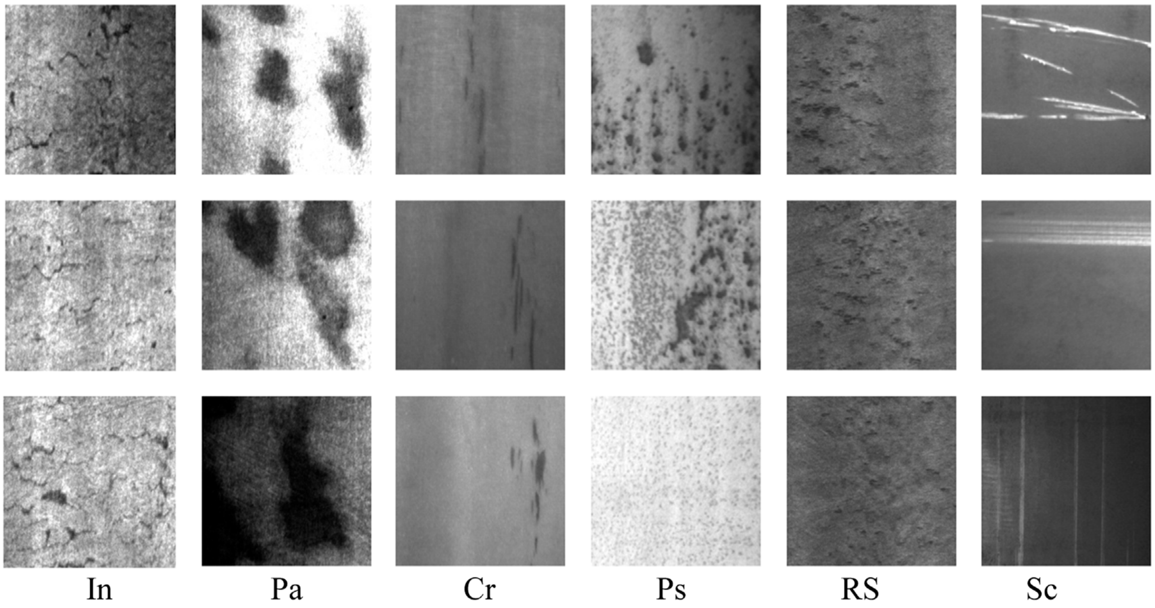

FIGURE 5
Six types of steel surface defects.

**TABLE 1 Average classification accuracy of different sample sizes.**

| Sample size for each category (total sample size) | Average accuracy (%) | | Increase (%) |
|---|---|---|---|
| | ACGAN | Ours | |
| 200 (1,200) | 94.83 | **98.67** | 4.04 |
| 150 (900) | 93.67 | **98.33** | 4.66 |
| 100 (600) | 90.00 | **95.50** | 5.5 |
| 50 (300) | 83.83 | **94.67** | 10.84 |
| 30 (180) | 77.83 | **94.00** | 16.17 |
| 10 (60) | 66.50 | **89.67** | 23.17 |

Bold values mean the best results.

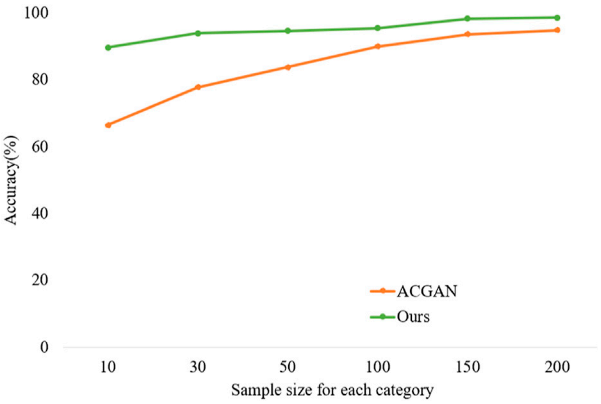

FIGURE 6
Trend chart of classification results under different sample sizes.

effective than the model in this paper when dealing with few-shot problems.

To illustrate the classification ability of the proposed model for each type of defect, Figure 7 shows the confusion matrix of our model under different sample sizes, where the numbers 0–5 in the abscissa and ordinate represent defect types, respectively: Cr, In, Pa, PS, RS, and Sc. It is evident that our method can train an ideal model under different training sample sizes and can accurately classify most of the defects. Moreover, when the sample size is 200, the model can accurately classify all Pa defects. Under different training sample sizes, the cases of classifying Cr as RS and RS as Cr occupy a large proportion in the wrong classification cases. The high similarity between Cr and RS defects and the lack of distinct inter-class features lead to misjudgment of the model. The overall results demonstrate that the method proposed in this paper only misjudges a few fault types under different sample sizes, and the overall accuracy remains high as the sample size decreases.

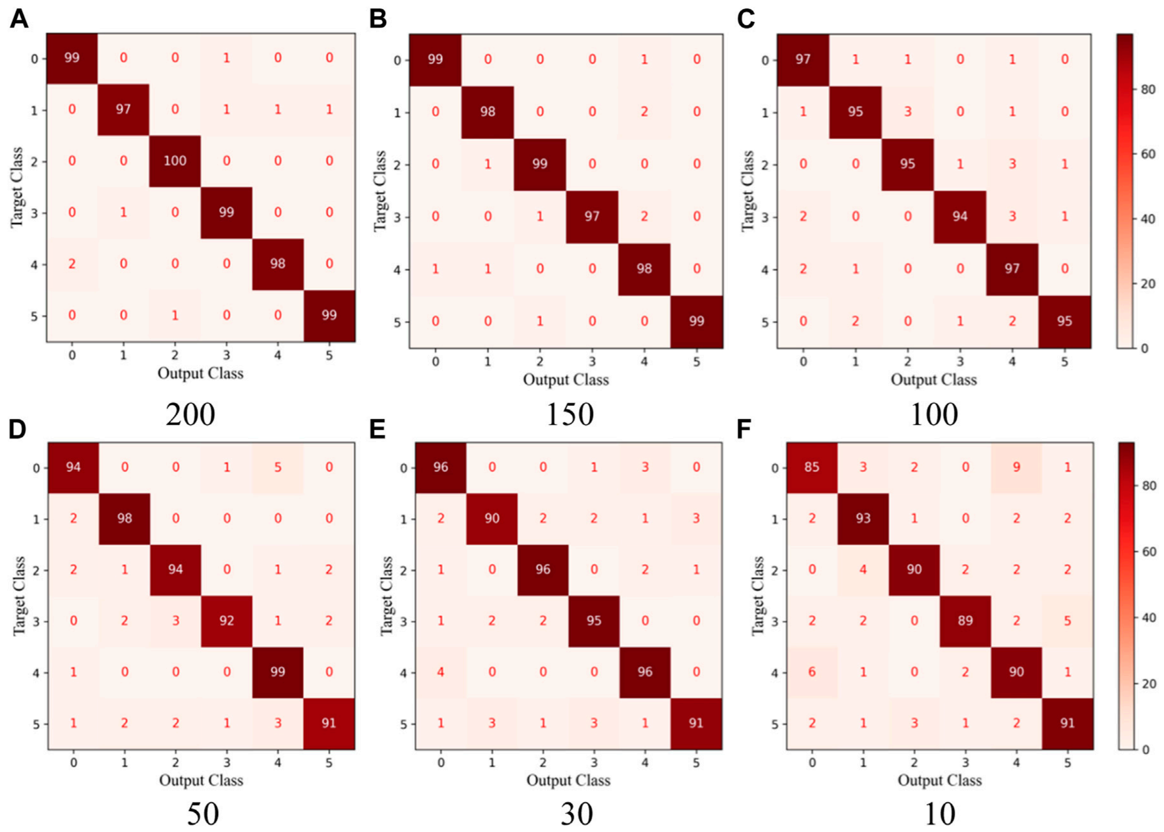

FIGURE 7
Confusion matrix of our model under different training sample sizes. **(A)** 200, **(B)** 150, **(C)** 100, **(D)** 50, **(E)** 30, and **(F)** 10.

**TABLE 2 Results of steel surface defects under different methods and sample sizes.**

| Methods | Average accuracy (%) | | | | | |
|---|---|---|---|---|---|---|
| | 200 | 150 | 100 | 50 | 30 | 10 |
| ResNet18 | 92.33 | 90.67 | 85.00 | 83.33 | 76.33 | 59.5 |
| ResNet50 | 93.00 | 92.33 | 85.33 | 83.00 | 77.00 | 63.17 |
| Res-ACGAN | 96.17 | 95.00 | 91.00 | 84.67 | 79.00 | 70.50 |
| [28] | 96.50 | 95.50 | 91.00 | 89.50 | 87.50 | 76.33 |
| [35] | 96.67 | 94.67 | 90.50 | 85.33 | 84.83 | 71.33 |
| Ours (lack MSFE) | 97.00 | 96.00 | 94.33 | 93.67 | 91.33 | 86.00 |
| Ours | **98.67** | **98.33** | **95.50** | **94.67** | **94.00** | **89.67** |

Bold values mean the best results.

In order to further validate the classification performance of our model, we compare it with the classic ResNet18 and ResNet50 classification methods. To ensure the efficient classification performance of the classic classification models, the ResNet18 and ResNet50 models are pre-trained using the ImageNet dataset. Additionally, we also compared with the latest few-shot deep learning classification models, including: the model proposed by Lian et al. [28], which combine generative adversarial networks and convolutional neural networks to generate exaggerated defect image samples to ensure the accuracy of micro-surface defect detection; and the model proposed by Li et al. [35], which replace the fully connected classification layer with an orthogonal SoftMax layer, significantly reducing the complexity of the model and making it suitable for few-shot classification. Moreover, in order to fully demonstrate the impact of MSFE on the final classification results, MSFE is deliberately excluded in the original framework and a corresponding experiment is conducted. The experimental results are presented in Table 2, and it can be seen that, in the case of different sample sizes, the methods proposed in this paper have achieved the best results and achieved the highest classification accuracy.

In order to further verify the importance of introducing various parts in our model, we compare the classification performance of the model after introducing the residual module, Wasserstein distance and penalty weight GP [30], Wasserstein divergence, and MSFE into the ACGAN model respectively. At the same time, in order to verify the important role played by the residual module in the discriminator network, we replace the residual module in the discriminator of our model with the CBAM attention mechanism module proposed by [35], and introduce the SENet module to conduct comparative experiments. The training sample size is 200, and the experimental results are shown in Table 3. It can be found that after adding the residual module to the original model, the classification accuracy of ACGAN increases by 1.34%, the

**TABLE 3 Classification accuracy of introducing different modules.**

| Method | Accuracy (%) |
|---|---|
| ACGAN | 94.83 |
| ACGAN + Res | 96.17 |
| ACGAN + Wasserstein + GP | 95.33 |
| ACGAN + Wasserstein-div | 95.83 |
| ACGAN + Res + Wasserstein + GP | 96.50 |
| ACGAN + SENet + Wasserstein + GP | 95.83 |
| ACGAN + Res + SENet + Wasserstein + GP | 96.67 |
| ACGAN + Res + CBAM + Wasserstein + GP | 96.83 |
| ACGAN + Res + Wasserstein-div | 97.00 |
| ACGAN + Res + Wasserstein-div + MSFM (Ours) | **98.67** |

Bold values mean the best results.

classification accuracy of ACGAN + Wasserstein + GP increases by 1.17%, and the classification accuracy of ACGAN + SENet + Wasserstein + GP increases by 0.84%. After adding Wasserstein divergence to the original model, the classification accuracy of ACGAN increases by 1%, and the classification accuracy of ACGAN + Res increases by 0.83%, which is higher than that of using Wasserstein distance and penalty weight, showing that introducing the residual module and Wasserstein divergence into the model can improve the feature extraction ability of the model and further improve the model's ability to discriminate and classify sample images. In addition, the introduction of attention mechanism modules SENet and CBAM in the discriminator network can improve the classification ability of the model, but the discriminator network structure proposed in this paper has achieved the best results in experiments. The incorporation of MSFE in the original framework results in a 1.67% increase in classification accuracy. This implies that employing semantic features at varying levels to guide the generator can enhance its efficiency, thereby advancing the classification abilities of the discriminator.

## 4.3 Quality assessment of generated samples

Figure 8 presents a comparison of steel surface defect samples generated by different models, including ACGAN, the model augmented with SENet module, the model augmented with CBAM module [35], the proposed method while lacking MSFE, and our model. The training process utilizes 200 samples of each type of defect, with 10,000 iterations and other parameters hold constant.

It can be observed that, compared to the original sample in Figure 5, the samples generated by the method proposed in this

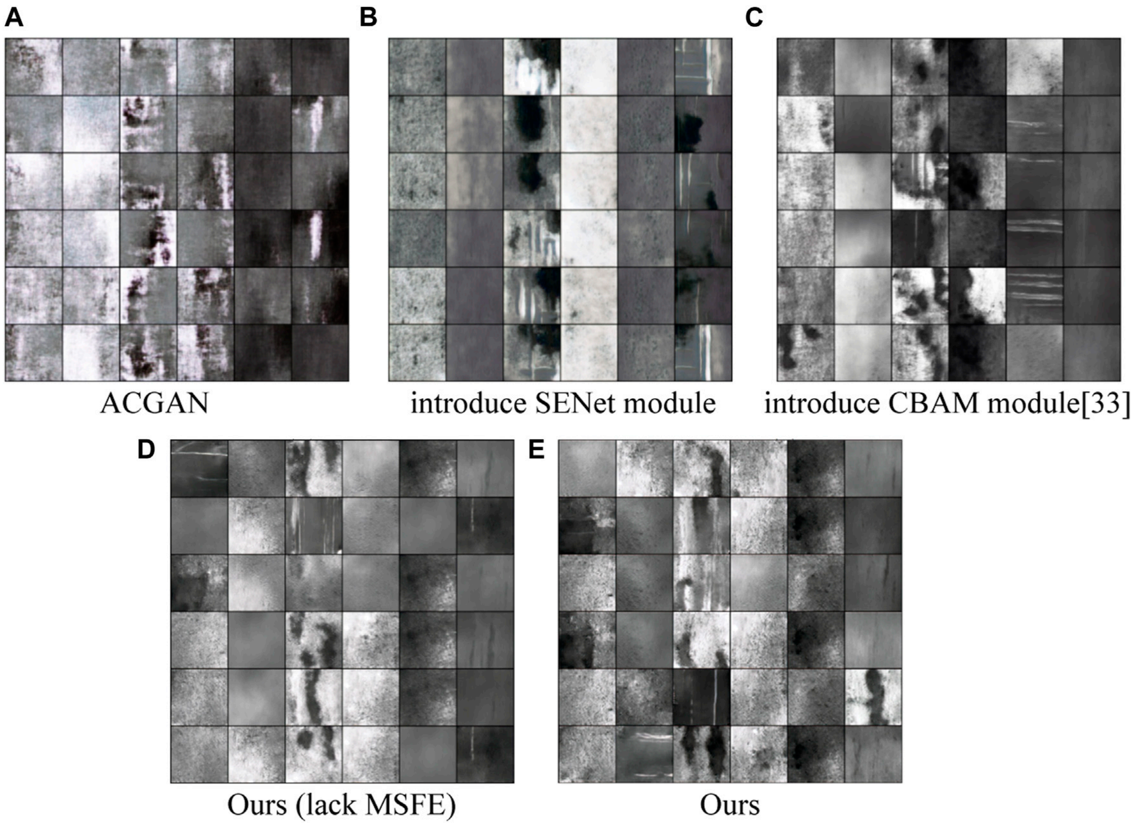

FIGURE 8
Sample images generated by different methods. **(A)** ACGAN. **(B)** introduce SENet module. **(C)** introduce CBAM module [34]. **(D)** Ours (lack MSFE). **(E)** Ours.

**TABLE 4 Comparison of MSE and SSIM values of different models.**

| Methods | MSE | SSIM |
|---|---|---|
| ACGAN | 347.8543 | 0.6935 |
| SENet-ACGAN | 272.5821 | 0.7074 |
| CBAM-ACGAN | 222.4989 | 0.7583 |
| Ours (lack MSFE) | 193.8484 | 0.7828 |
| Ours | **184.2617** | **0.7912** |

Bold values mean the best results.

paper are more distinct and the quality of the samples are also much better. For instance, for the defect of scratch, such as the third one in the fifth row and the second one in the sixth row in Figure 8E generated by the method proposed in this paper, when compared to the last one in the second row in (a), the last one in the first row in (b), the third one in the fourth row in (c), and the last one in the last row in (d), its defect features are more discernible, the defect is sharper, and it is also more similar to the original sample image. Although the version of lacking MSFE can also generate high-quality sample images, it is evident that its feature extraction ability is inadequate, leading to blurred images and unclear semantic information, as demonstrated in Figure 8D, specifically in the fifth one of the second row and the second item of the fourth row.

In order to assess the quality of samples generated by different models, the MSE (Mean Square Error) and SSIM (Structural Similarity) metrics are employed to evaluate the sample quality. MSE is a metric that reflects the degree of discrepancy between the estimator and the estimated quantity; SSIM is used to measure the similarity between two images. The results of different models are presented in Table 4. The smaller the value of MSE, or the larger the value of SSIM, the larger the similarity between original image and generated image. It can be seen from Table 4 that the MSE and SSIM of our model are more proximate to the original images than other methods, which demonstrates that the sample data distribution generated by our model is more similar to the original sample distribution, and also shows that MSFE and Wasserstein divergence can improve the quality of samples generated by the model.

## 5 Conclusion

Aiming at the difficulties of steel surface defect few-shot classification, this paper introduces multi-level semantic feature extractor under the residual adversarial learning network framework to generate high-quality samples and achieves promising steel surface defect classification. First, we modify the network structure of the adversarial learning model by the residual module, so that the model can obtain more information during training and generate synthetic data to the original sample. To overcome the challenge of inadequate feature extraction in generator networks which may lead to suboptimal sample quality in small-sample environments, we design a multi-level semantic feature extractor for obtaining diverse semantic information at various levels. By leveraging this comprehensive semantic information, we directed sample generation. At the same time, the Wasserstein divergence is introduced into the loss function to solve the problem of unstable model training and to improve the generation efficiency and classification performance of the model. Experiments are conducted on the steel surface defect dataset NEU-CLS from Northeastern University. The results demonstrate that, under the condition of the restricted number of training samples, the method proposed in this paper achieves the highest classification accuracy. Moreover, when the number of training data is reduced, our method exhibits better stability and robustness than classical classification models and state-of-the-art of deep learning models. Additionally, in terms of the quality of generated samples, the MSE value and SSIM value of the samples generated by the model proposed in this paper are the closest to the original samples, further showing the effectiveness of our proposed method. With the popularization of sensors and lightweight devices, the demand for model compression and lightweight models is becoming increasingly important. Improving the real-time performance of defect detection systems is the main trend for deploying online detection systems in actual industrial production in the future.

## Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

## Author contributions

LH: Ideas, methodology, experimental design, formal analysis. PS: Software, validation, data curation. ZP: Supervision, Writing—review and editing. YX: Supervision, writing—review and editing. All authors contributed to the article and approved the submitted version.

## Funding

This work was supported by the Establishment of Key Laboratory of Shenzhen Science and Technology Innovation Committee under Grant ZDSYS20190902093015527, the Shenzhen Science and Technology Innovation Committee under Grant JSGG20220831104402004, and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005).

## Conflict of interest

Authors PS and ZP were employed by HBIS Digital Technology Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

1. Ayarkwa J. Influence of wood defects on some mechanical properties of two tropical Ghanaian hardwoods. *J Ghana Sci Assoc* (1999) 1:131–47. doi:10.4314/jgsa.v1i2.17813

2. Yu Z, Wu X, Gu X. *Fully convolutional networks for surface defect inspection in industrial environment*. Cham: Springer (2017).

3. Song G, Song K, Yan Y. EDRNet: Encoder–Decoder residual network for salient object detection of strip steel surface defects. *IEEE Trans Instrumentation Meas*, 2020, 69:1–. doi:10.1109/TIM.2020.3002277

4. Zaghdoudi R, Seridi H, Ziani S. Binary Gabor pattern (BGP) descriptor and principal component analysis (PCA) for steel surface defects classification[C]. In: Proceeding of the 2020 International Conference on Advanced Aspects of Software Engineering (ICAASE); November 2020. IEEE (2020). p. 1–7.

5. Hu H, Li Y, Liu M, Liang W. Classification of defects in steel strip surface based on multiclass support vector machine. *Multimedia tools Appl* (2014) 69(1):199–216. doi:10.1007/s11042-012-1248-0

6. Duan C, Zhang T. Two-stream convolutional neural network based on gradient image for aluminum profile surface defects classification and recognition. *IEEE Access* (2020) 8:172152–65. doi:10.1109/access.2020.3025165

7. Liu X, He W, Zhang Y, Yao S, Cui Z. Effect of dual-convolutional neural network model fusion for Aluminum profile surface defects classification and recognition. *Math Biosciences Eng* (2022) 19(1):997–1025. doi:10.3934/mbe.2022046

8. Mayr M, Hoffmann M, Maier A, Christlein V. Weakly supervised segmentation of cracks on solar cells using normalized L p norm[C]. In: Proceeding of the 2019 IEEE International Conference on Image Processing (ICIP); September 2019. IEEE (2019). p. 1885–9.

9. Huang Y, Qiu C, Yuan K. Surface defect saliency of magnetic tile. *Vis Comp* (2020) 36(1):85–96. doi:10.1007/s00371-018-1588-5

10. Li K, Qi Y, Su L, Gu J, Su W. Visual inspection of steel surface defects based on improved auxiliary classification generation adversarial network[J/OL]. *Chin J Mech Eng* (2023) 1–9. Available at: http://kns.cnki.net/kcms/detail/11.2187.TH.20220526.1827.106.html.

11. Panaretos VM, Zemel Y. Statistical aspects of Wasserstein distances. *Annu Rev Stat its Appl* (2019) 6:405–31. doi:10.1146/annurev-statistics-030718-104938

12. Radford A, Metz L, Chintala S. *Unsupervised representation learning with deep convolutional generative adversarial networks[J]*. arXiv preprint (2015).

13. Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier gans[C]. In: *International conference on machine learning*. New York: PMLR (2017). p. 2642–51.

14. Dosovitskiy A, Brox T. Generating images with perceptual similarity metrics based on deep networks. *Adv Neural Inf Process Syst* (2016) 29.

15. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. *Commun ACM* (2020) 63(11):139–44. doi:10.1145/3422622

16. Lu HP, Su CT. CNNs combined with a conditional GAN for mura defect classification in TFT-LCDs. *IEEE Trans Semiconductor Manufacturing* (2021) 34(1):25–33. doi:10.1109/tsm.2020.3048631

17. Li X, Zhang W, Ding Q. Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks. *IEEE Trans Ind Elect* (2018) 66(7):5525–34. doi:10.1109/TIE.2018.2868023

18. Liu J, Zhang BG, Li L. Defect detection of fabrics with generative adversarial network based flaws modeling[C]. In: Proceeding of the 2020 Chinese Automation Congress (CAC); November 2020; Shanghai, China. IEEE (2020). p. 3334–8.

19. Cheon S, Lee H, Kim CO, Lee S. Convolutional neural network for wafer surface defect classification and the detection of unknown defect class. *IEEE Trans Semiconductor Manufacturing* (2019) 32(2):163–70. doi:10.1109/tsm.2019.2902657

20. Nakazawa T, Kulkarni DV. Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Trans Semiconductor Manufacturing* (2018) 31(2):309–14. doi:10.1109/tsm.2018.2795466

21. Zhu H, Ge W, Liu Z. Deep learning-based classification of weld surface defects. *Appl Sci* (2019) 9(16):3312. doi:10.3390/app9163312

22. Wan X, Zhang X, Liu L. An improved VGG19 transfer learning strip steel surface defect recognition deep neural network based on few samples and imbalanced datasets. *Appl Sci* (2021) 11(6):2606. doi:10.3390/app11062606

23. Han T, Liu C, Yang W, Jiang D. Deep transfer network with joint distribution adaptation: A new intelligent fault diagnosis framework for industry application. *ISA Trans* (2020) 97:269–81. doi:10.1016/j.isatra.2019.08.012

24. Liu Y, Yuan Y, Liu J. Deep learning model for imbalanced multi-label surface defect classification. *Meas Sci Tech* (2021) 33(3):035601. doi:10.1088/1361-6501/ac41a6

25. Jain S, Seth G, Paruthi A, Yang EWR, Lwin S, Yeo TT, et al. Pseudoaneurysm resulting in rebleeding after evacuation of spontaneous intracerebral hemorrhage. *J Intell Manufacturing* (2020) 143:1–6. doi:10.1016/j.wneu.2020.07.088

26. He Y, Song K, Dong H, Yan Y. Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network. *Opt Lasers Eng* (2019) 122:294–302. doi:10.1016/j.optlaseng.2019.06.020

27. Zhao Z, Li B, Dong R, Zhao P. A surface defect detection method based on positive samples[C]. In: *Pacific rim international conference on artificial intelligence*. Cham: Springer (2018). p. 473–81.

28. Lian J, Jia W, Zareapoor M, Zheng Y, Luo R, Kumar D. Deep-learning-based small surface defect detection via an exaggerated local variation-based generative adversarial network. *IEEE Trans Ind Inform* (2019) 16(2):1343–51. doi:10.1109/TII.2019.2945403

29. Tsai DM, Fan SKS, Chou YH. Auto-annotated deep segmentation for surface defect detection. *IEEE Trans Instrumentation Meas* (2021) 70:1–10. doi:10.1109/tim.2021.3087826

30. Dumoulin V, Visin F. *A guide to convolution arithmetic for deep learning* (2016). arXiv preprint arXiv:1603.07285.

31. Arjovsky M, Chintala S, Bottou L. *Wasserstein GAN* (2017).

32. Wu J, Huang Z, Thoma J, Acharya D, Gool L. Wasserstein divergence for gans[C]. In: *Proceedings of the European conference on computer vision*. Switzerland: ECCV (2018). p. 653–68.

33. Dong H, Song K, He Y, Xu J, Yan Y, Meng Q. PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection. *IEEE Trans Ind Inform* (2019) 16(12):7448–58. doi:10.1109/TII.2019.2958826

34. Meng Z, Li Q, Sun D, Cao W, Fan F. An intelligent fault diagnosis method of small sample bearing based on improved auxiliary classification generative adversarial network. *IEEE Sensors J* (2022) 22(20):19543–55. doi:10.1109/jsen.2022.3200691

35. Li X, Chang D, Ma Z, Tan ZH, Xue JH, Cao J, et al. OSLNet: Deep small-sample classification with an orthogonal softmax layer. *IEEE Trans Image Process* (2020) 29:6482–95. doi:10.1109/tip.2020.2990277