applied sciences

MDPI

Article

Welding Image Data Augmentation Method Based on LRGAN Model

Ying Wang *, Zhe Dai, Qiang Zhang and Zihao Han

School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China; daizhe18854022985@163.com (Z.D.); dqpi_zq@163.com (Q.Z.); hzhzslbq@163.com (Z.H.)
* Correspondence: wangying@nepu.edu.cn

Abstract: This study focuses on the data bottleneck issue in the training of deep learning models during the intelligent welding control process and proposes an improved model called LRGAN (loss reconstruction generative adversarial networks). First, a five-layer spectral normalization neural network was designed as the discriminator of the model. By incorporating the least squares loss function, the gradients of the model parameters were constrained within a reasonable range, which not only accelerated the convergence process but also effectively limited drastic changes in model parameters, alleviating the vanishing gradient problem. Next, a nine-layer residual structure was introduced in the generator to optimize the training of deep networks, preventing the mode collapse issue caused by the increase in the number of layers. The final experimental results show that the proposed LRGAN model outperforms other generative models in terms of evaluation metrics such as peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and Fréchet inception distance (FID). It provides an effective solution to the small sample problem in the intelligent welding control process.

Keywords: intelligent welding; data augmentation; spectral normalization discriminator; residual structure

check for updates

Academic Editor: Andrea Prati

Received: 3 April 2025
Revised: 9 May 2025
Accepted: 14 May 2025
Published: 19 June 2025

Citation: Wang, Y.; Dai, Z.; Zhang, Q.; Han, Z. Welding Image Data Augmentation Method Based on LRGAN Model. *Appl. Sci.* **2025**, *15*, 6923. https://doi.org/10.3390/app15126923

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Welding, as a core process in modern industrial manufacturing, plays a crucial role in the accuracy and reliability of quality inspection, which directly impacts product performance, production efficiency, and economic benefits [1]. In recent years, deep learning (DL) technology has demonstrated significant application potential in the field of quality inspection for complex welding components, owing to its powerful features including self-learning capabilities, complex pattern recognition performance, and excellent generalization characteristics [2]. However, the performance of deep learning models is highly dependent on large-scale, high-quality training data. In industrial welding scenarios, there is a common challenge in data acquisition: on one hand, the collection of welding samples faces significant challenges due to harsh environmental factors such as strong arc light interference, high temperatures, and high humidity; on the other hand, the distribution of welding defect samples is imbalanced (e.g., there is a scarcity of defect samples like porosity, cracks, etc.), making it difficult to meet the deep learning models' need for data diversity and representativeness. The aforementioned data bottleneck severely restricts the accuracy and robustness of welding quality detection models, impacting their detection precision [3]. Therefore, overcoming the scarcity of welding data in deep learning models and expanding the original dataset to provide a solid data foundation for the subsequent

construction of high-performance welding quality detection models has become a key issue that urgently needs to be addressed in the current field of welding automation.

Currently, the main approach to solving the issue of insufficient welding samples is based on generative adversarial networks (GANs) [4]. Although GANs have been widely applied in fields such as medicine [5] and computer vision [6], they still face several challenges due to the adversarial mechanism and the limitations of the cross-entropy loss function. These challenges include model collapse, gradient explosion, slow convergence speed, and gradient disappearance [7–9]. In reference [7], Hu Mengting et al. found that in the classification problem, the mean square error loss function suffers from gradient disappearance, resulting in a low learning rate in the early stage of model training. Therefore, it is necessary to improve the cross-entropy loss function to avoid the problem of gradient disappearance. In reference [8], Liang et al. found that one of the reasons why the original GAN is prone to model collapse during training is that the input is random noise without limit. In reference [9], Kang et al. found that during the training process of standard GANs, the discriminator converges faster than the generator, often reaching a convergence point quickly. At this point, it becomes difficult for the discriminator to continue guiding the generator correctly. Compared to the discriminator model, the generator model typically has a deeper network structure. Due to the lack of effective guidance, the generator often experiences gradient disappearance. To address the aforementioned issues and improve the image generation quality and training efficiency of the GAN across different industrial datasets, researchers have systematically improved GANs in two key dimensions: loss function optimization and network architecture innovation. By introducing a conditional constraint mechanism, the conditional generative adversarial network (CGAN) was proposed, enabling the directed generation of images belonging to specific categories [10]. StarGAN (star generative adversarial network), developed based on the concept of domain adaptation, significantly enhances the capability of cross-domain image generation [11]. ACGAN (auxiliary classifier generative adversarial network), which incorporates an auxiliary classifier, and DCGAN (deep convolutional generative adversarial network), which employs a deep convolutional architecture, have made groundbreaking advances in image feature representation and generation quality, respectively [12,13]. These improved models, through different technical approaches, have progressively narrowed the distribution gap between generated images and real samples, providing strong support for data augmentation in the field of welding quality inspection.

Arjovsky, Chintala, and other researchers have concluded the following through their study and analysis:

On the one hand, cross-entropy has limitations when measuring the distribution differences between non-overlapping data, which is a major cause of the instability in the training of the original GAN. To address this issue, Arjovsky et al. [14] proposed using the Wasserstein distance to measure the difference between the distribution of generated data and real data, tightly coupling the objective function with the quality of generated samples. Subsequently, Gulrajani et al. [15] introduced the addition of a gradient penalty term to the Wasserstein distance, further improving the stability of the discriminator's training. Later, Li Xiujun [16] conducted research on constrained optimization of the training results of WGAN. However, WGAN requires the discriminator to have Lipschitz continuity, which is typically ensured through weight clipping, and the pruning operation in WGAN can disrupt the structure of the discriminator's weight matrix, leading to slower convergence rates for WGAN [17].

On the other hand, when the pixel value distributions of the generated and real images do not overlap or the overlapping region is negligible, the discriminator model can still experience gradient vanishing during training, leading to instability. To address this issue,

Solomon Kullback and Richard Leibler [18] proposed the Kullback–Leibler divergence (KL divergence) to measure the consistency between the distributions of real and generated data. However, merely measuring the consistency between the distributions of real and generated data without enforcing a directional constraint or convergence constraint does not effectively address the gradient vanishing issue. In reference [19], a repulsion term was added to the network's loss function, which evaluates the sample overlap rate during model training, thereby alleviating the gradient vanishing problem. Inspired by this, the design of an improved loss function does not involve the Wasserstein distance and does not require weight clipping or gradient penalty. Given the decision penalty mechanism of the least squares loss function, even if fake samples are correctly classified, the least squares loss still penalizes them and pulls them toward the decision boundary. This characteristic has been thoroughly confirmed in references [20,21]. In reference [20], He Zhipeng et al. proposed a weighted least squares method for a twin support vector machine to solve the problems of noise sample sensitivity and large classification gap leading to unstable training. In reference [21], Shu Jun et al. solved the imbalance problem between normal bottle cap samples and defective bottle cap samples by semi-supervised anomaly detection (GANomaly network) when detecting defects. However, the image reconstruction effect was poor, so they proposed improving the image reconstruction of the GANomaly network based on cross-attention and least squares loss function. Therefore, this paper incorporates the least squares loss function into the KL divergence. This design not only avoids the issues caused by weight clipping, such as the destruction of the discriminator's weight matrix structure and slow convergence but also alleviates the problem of slow loss reduction.

Additionally, although the least squares loss function improves training stability to some extent, the discriminator's stability is still significantly affected by the gradient vanishing problem caused by large changes in the discriminator's parameters. To address this issue, Miyato et al. proposed applying spectral normalization to the parameter matrix during the discriminator's training process, thereby constraining the gradients of the model parameters within a reasonable range, which in turn ensures stable training of the discriminator [22]. Furthermore, Zhou et al. suggested that in order to guarantee the stability of the GAN's discriminator training, it is essential to ensure that the discriminator's function satisfies the Lipschitz condition [23]. To further improve this issue, this paper introduces spectral normalization during the discriminator's training process. By limiting the range of parameter gradients, it effectively avoids excessive changes in the discriminator's parameters, thereby enhancing the stability of the training process. Compared to traditional weight clipping methods, spectral normalization effectively controls the Lipschitz constant by normalizing the maximum singular value of the weight matrix in each layer, thereby improving the stability of the model without disrupting the structure of the convolutional kernels. Unlike gradient penalty, spectral normalization does not require the introduction of additional penalty terms, reducing computational overhead. It also avoids the training oscillations that may arise from batch normalization in adversarial training. Therefore, in this study, we adopt the spectral normalization method to impose stability constraints on the discriminator, thereby enhancing the training stability of the discriminator more effectively.

In addition, for adversarial generative models, in order for the generator to produce image features that are closer to the original images, a deeper network structure is often required. However, as the network depth increases, the generator typically faces the issue of gradient explosion during training, leading to training failure. This is primarily due to the accumulation of errors during the backpropagation process [24,25]. Residual structures, by adding shortcut connections between layers of the neural network, weaken the strong dependencies between layers, thus ensuring that the performance of deeper networks is

at least as good as that of shallower networks. This has been demonstrated in [26,27]. In summary, to address the common data bottleneck problem in industrial welding scenarios, this paper proposes a loss-reconstruction-based generative adversarial network (LRGAN), First, a five-layer spectral normalization neural network was designed as the discriminator for the model, using a least squares loss function. Next, a nine-layer residual structure was introduced in the generator. The goal is to overcome the performance bottlenecks of existing GAN models in welding image generation, accelerate the convergence process, limit drastic changes in model parameters, and alleviate issues such as gradient vanishing and mode collapse.

2. LRGAN Model

To improve the training stability of the network and the quality of welding image generation, this paper proposes the LRGAN model, whose architecture is shown in Figure 1. In the discriminator network, the least squares loss function and spectral normalization are applied to restrict the gradients of each layer's network parameters within a specified range. This approach helps slow down the convergence speed of the discriminator while ensuring the network's training stability without compromising the structure of the parameter matrices. In the generator network, a residual module is introduced to prevent the model from experiencing mode collapse caused by the increase in the number of network layers, making the generated images more similar to the original images.

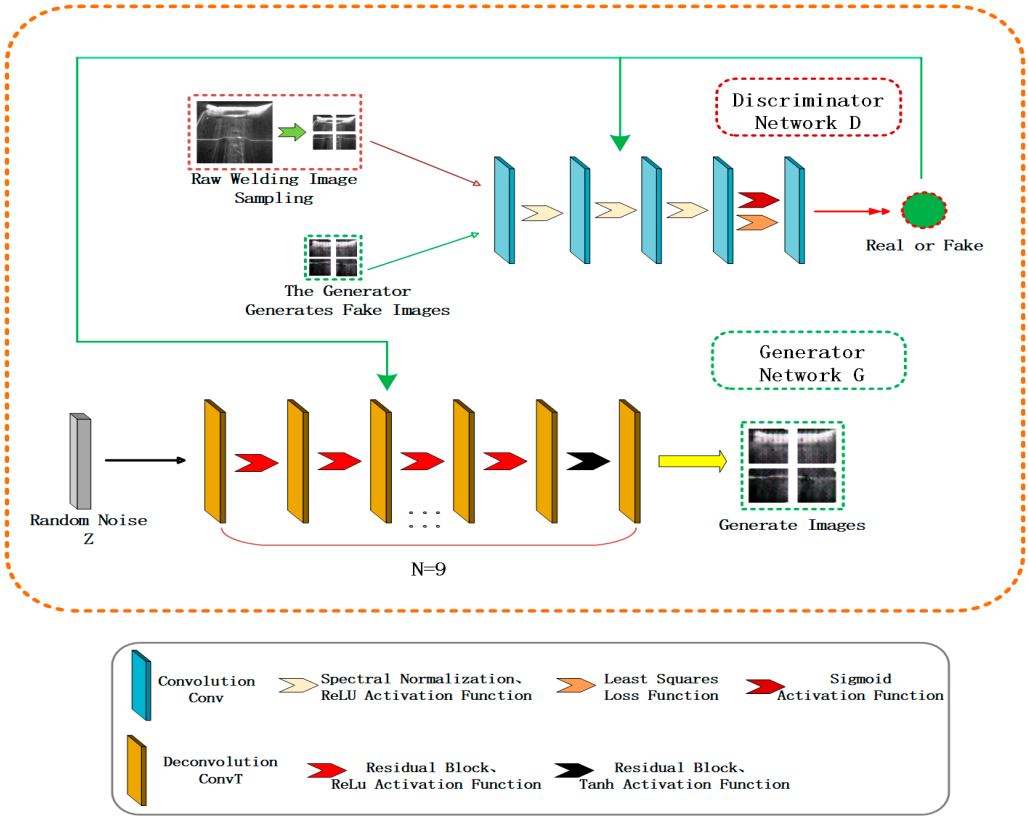


Figure 1. LRGAN model diagram.

2.1. The Discriminator Part of the LRGAN Model

Considering the balance between network capacity and overfitting, a discriminator with too few layers (such as one to three layers) may lead to insufficient model capacity, making it difficult to capture complex image patterns. On the other hand, too many layers (such as nine or more) may enhance expressive power but also risk overfitting the training

data. Therefore, in this paper, a five-layer spectral normalization neural network was designed as the discriminator using a least squares loss function. The five-layer architecture is widely used in many image generation tasks (such as DCGAN and SRGAN) and is an empirically validated choice for a good trade-off. Furthermore, the LRGAN generator in this study uses a nine-layer residual module. To match the feature learning capability during the adversarial process and align with the generator's layer count, the discriminator should not be too shallow, and five layers provide an appropriate level of complexity.

2.1.1. The Discriminator Structure of the LRGAN Model

The discriminator structure in the LRGAN model is shown in Figure 2.

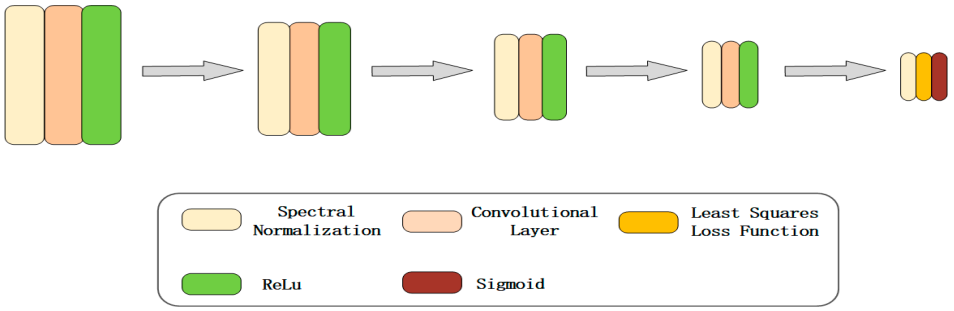


Figure 2. LRGAN discriminator structure diagram.

2.1.2. Loss Function

The least squares method is an optimization technique used to find the best function fit for the data by minimizing the sum of the squared errors. By applying the least squares method, we can easily estimate unknown data while minimizing the sum of squared errors between the estimated data and the actual data. The instability of the original GAN training process largely arises from using the cross-entropy loss function as the discriminator's loss function. Replacing the cross-entropy loss function with the least squares loss function can effectively address this issue. According to Legendre's principle of best approximation, the least squares method is represented by the following Equation (1).

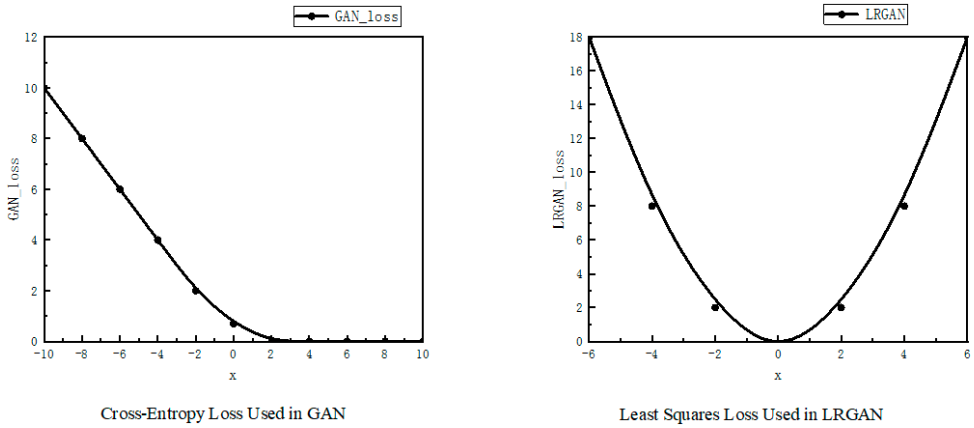
$$E = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - y)^2 \tag{1}$$

where y_i represents the observed multiple sample data, y is the hypothesized fitting model function, and E is the loss function (the sum of squared residuals). The goal of the least squares method is to minimize the loss function. After introducing the least squares method into the improved LRGAN model, the final objective functions for the generator and discriminator networks are as follows:

$$\min V_{LRGAN}(D) = \frac{1}{2} E_{x-p_{\text{data}}(x)} \left[(D(x) - b)^2 \right] + \frac{1}{2} E_{z-p_z(z)} \left[(D(G(z,y)) - a)^2 \right] \tag{2}$$

$$\min V_{LRGAN}(G) = \frac{1}{2} E_{z-p_z(z)} \left[(D(G(z,y)) - c)^2 \right] + \frac{1}{2} E_{x-p_{\text{data}}(x)} \left[(D(x) - b)^2 \right] \tag{3}$$

Figure 3 illustrates the difference between the cross-entropy loss function and the least squares loss function. Even when fake samples are correctly classified, the least squares loss function still imposes a penalty on them, pulling them toward the decision boundary. The cross-entropy loss function has a large saturated region on the right side where the gradient is zero. When samples fall within this region, the generator's update becomes significantly weakened. In contrast, the least squares loss function only saturates at one point, allowing the LRGAN training process to continue without interruption.



Cross-Entropy Loss Used in GAN Least Squares Loss Used in LRGAN

Figure 3. Loss function comparison diagram.

The goal of the least squares method is to minimize the loss function. In traditional GAN, when the discriminator network is trained optimally, the objective function of the generative adversarial network $V(D_G^*, G)$ becomes a Jensen–Shannon divergence problem, where $D_G^* = \frac{P_{\text{data}}(x)}{P_{\text{data}}(x)+P_g(x)}$ is obtained:

$$\begin{aligned} C(G) &= V(D_G, G) = -\log 4 + KL\left(p_{\text{data}}(x) \,||\, \tfrac{p_{\text{data}}(x)+p_g(x)}{2}\right) + KL\left(p_g(x) \,||\, \tfrac{p_{\text{data}}(x)+p_g(x)}{2}\right) \\ &= -\log 4 + 2JSD\left(p_{\text{data}}(x) \,||\, p_g(x)\right) \end{aligned} \tag{4}$$

Here, D is a binary classifier that distinguishes between real and fake, x is the input real data, p(x) is the distribution of the real training samples, G is the generative network, and C(G) represents the global minimum of the objective function of the generative network G. The formulas for the Kullback–Leibler (KL) divergence and the Jensen–Shannon (JS) divergence are as follows:

$$KL(p \parallel q) = -\int p(x) \log \frac{q(x)}{p(x)} dx \tag{5}$$

$$JSD(p \parallel q) = \frac{1}{2}KL\left(p \parallel \frac{p+q}{2}\right) + \frac{1}{2}KL\left(q \parallel \frac{p+q}{2}\right) \tag{6}$$

For the KL divergence, if the two distributions p and q are far apart and have no overlap, the JS divergence in the objective function of the generative network will tend to a constant value of log4, causing the objective function value to approach 0 and resulting in a vanishing gradient problem. LRGAN addresses this issue by using the least squares loss function. The objective functions of the generative network and the discriminator network are as follows:

$$\min V_{LRGAN}(D) = \frac{1}{2}E_{x-p_{\text{data}}(x)}\left[(D(x)-b)^2\right] + \frac{1}{2}E_{z-p_z(z)}\left[(D(G(z,y))-a)^2\right] \tag{7}$$

$$\min V_{LRGAN}(G) = \frac{1}{2}E_{z-p_z(z)}\left[(D(G(z,y))-c)^2\right] \tag{8}$$

Here, E represents the expectation; a and b represent the labels of the generated data and real data, respectively, and c indicates that the generative network tries to make the discriminator believe the input has the label of the generated data; z is a random noise, drawn from the distribution p(z); $V_{LRGAN}(D)$ represents the least squares loss function for the discriminator; and the $V_{LRGAN}(G)$ represents the least squares loss function for the generator. The meanings of the other symbols are the same as in the previous formula and are not repeated. To overcome the gradient vanishing problem and make the training

process more stable, the $\frac{1}{2}E_{x-p_{data}(x)}\left[(D(X)-d)^2\right]$ will be added to Formula (3), as it does not contain any generator parameters, so it does not affect the optimal solution. The following formula is obtained:

$$\min V_{LRGAN}(G) = \frac{1}{2}E_{z-p_z(z)}\left[(D(G(z,y))-c)^2\right] + \frac{1}{2}E_{x-p_{data}(x)}\left[(D(x)-d)^2\right] \quad (9)$$

By fixing the generative network as constant, the optimal discriminator network can be derived:

$$D^*(x) = \frac{bp_{data}(x) + ap_g(x)}{p_{data}(x) + p_g(x)} \quad (10)$$

Then, the generative network formula becomes:

$$\begin{aligned} 2C(G) &= \mathrm{E}_{x\sim p_{\mathrm{date}}(x)}\left[(D^*(x)-c)^2\right] + E_{x\sim p_g(x)}\left[(D^*(x)-c)^2\right] \\ &= \mathrm{E}_{x\sim p_{\mathrm{data}}(x)}\left[\left(\frac{bp_{\mathrm{data}}(x)+ap_{\mathrm{g}}(x)}{p_{\mathrm{data}}(x)+p_{\mathrm{g}}(x)}-c\right)^2\right] + \mathrm{E}_{x\sim p_{\mathrm{g}}(x)}\left[\left(\frac{bp_{\mathrm{data}}(x)+ap_{\mathrm{g}}(x)}{p_{\mathrm{data}}(x)+p_{\mathrm{g}}(x)}-c\right)^2\right] \\ &= \int_x p_{\mathrm{data}}(x)\left[\frac{(b-c)p_{\mathrm{data}}(x)+(a-c)p_{\mathrm{g}}(x)}{p_{\mathrm{data}}(x)+p_{\mathrm{g}}(x)}\right]^2 dx + \int_x p_g(x)\left[\frac{(b-c)p_{\mathrm{data}}(x)+(a-c)p_{\mathrm{g}}(x)}{p_{\mathrm{data}}(x)+p_{\mathrm{g}}(x)}\right]^2 dx \\ &= \int_x \frac{\left((b-c)p_{\mathrm{data}}(x)+(a-c)p_g(x)\right)^2}{p_{\mathrm{data}}(x)+p_g(x)}\,dx \\ &= \int_x \frac{\left((b-c)\left(p_{\mathrm{data}}(x)+p_g(x)\right)-(b-a)p_g(x)\right)^2}{p_{\mathrm{data}}(x)+p_g(x)}\,dx \end{aligned} \quad (11)$$

Set b − c=1, b − a=2; then:

$$\begin{aligned} 2C(G) &= \int_x \frac{\left(2p_g(x)-\left(p_{data}(x)+p_g(x)\right)\right)^2}{p_{data}(x)+p_g(x)}\,dx \\ &= \chi^2_{Pearson}\left(p_{data}(x)+p_g(x)\|2p_g(x)\right) \end{aligned} \quad (12)$$

Here, $x^2_{Pearson}$ is the Pearson x^2 distance, which is zero only when $P_{\mathrm{data}}(x) = P_{\mathrm{g}}(x)$ are the same. In practice, the distribution of data generated by the generative network cannot exactly match the distribution of the real data. Therefore, it can be said that the least squares loss function, when replacing cross-entropy, significantly addresses the gradient vanishing problem in the training of generative adversarial networks at a theoretical level.

2.1.3. The Activation Function in the Discriminator

The discriminator in this paper is designed as a simple five-layer nested convolutional neural network, where the first four layers use the ReLU activation function, and the output layer uses the sigmoid activation function. The formulas and curves for the ReLU and sigmoid functions are shown in Equations (13) and (14) and Figure 4:

$$\mathrm{f(x)} = \max(0, \mathrm{x}) \quad (13)$$

$$\partial(\mathrm{x}) = \frac{1}{1+e^{-x}} \quad (14)$$

In the ReLU function, the input and output only have a linear relationship, and there is no gradient vanishing issue when the input is greater than 0. In deep networks, ReLU causes some neurons' outputs to be zero, reducing the connections between network parameters and ensuring the sparsity of the network. While this effectively prevents overfitting and gradient vanishing problems, ReLU does not effectively control the output range. Compared to other activation functions, the sigmoid activation function has a

smooth curve and is easy to differentiate. Therefore, in this paper, the sigmoid function with a Lipschitz constant of 1/4 is used in the output layer of the discriminator, ensuring smoother changes in the function. This typically helps improve training stability and reduce problems such as slow weight updates and gradient vanishing.

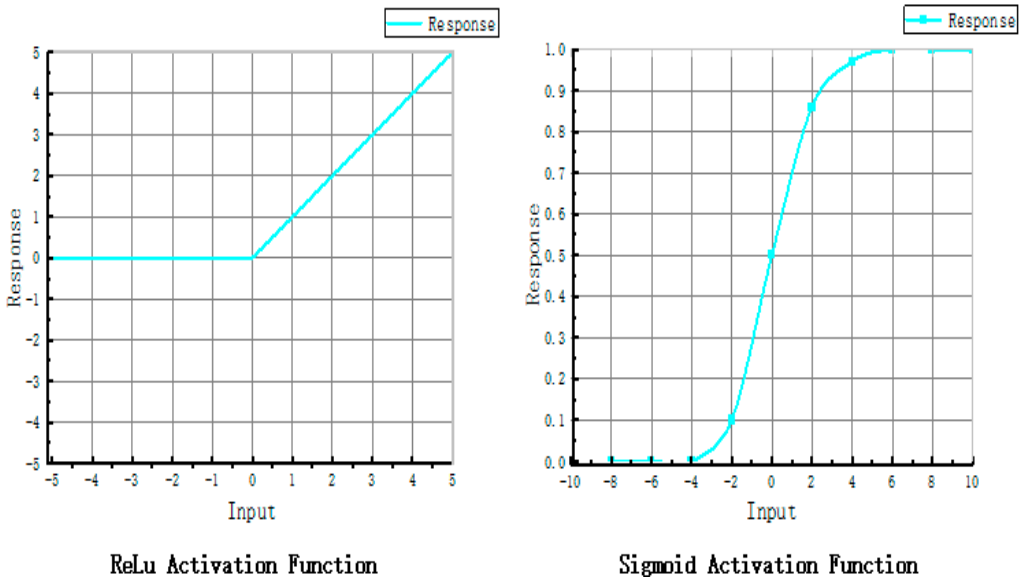


Figure 4. Discriminator activation function.

2.1.4. Spectral Normalization and Lipschitz Constant

Although the least squares loss function can accelerate convergence and, to some extent, ensure the stability of training, the issue of gradient vanishing caused by drastic changes in the discriminator's parameters has not been fully resolved. Therefore, this paper introduces spectral normalization during the training of the discriminator model to constrain the parameter matrix, ensuring that the gradients during backpropagation are limited within a certain range. This prevents excessive changes in the discriminator's parameters while preserving the structure of the convolutional kernels. Spectral normalization strictly constrains the spectral norm of the weight matrix at each layer, thereby controlling the Lipschitz constant of the discriminator, which enhances the stability of the generative adversarial network during training. Compared to other normalization techniques, spectral normalization only requires adjusting the Lipschitz constant, a conclusion that has been confirmed in [23]. According to the definition of the Lipschitz constant, we can obtain Equation (15):

$$\frac{\|f(x) - f(x')\|}{\|x - x'\|} \leq M \tag{15}$$

Here, M is a constant greater than 0, f(x) is a function defined on the real number set, and let $||f||_{Lip} = \| f(x) - f(x') \| / ||x - x'||$; then, $\| f \|_{Lip} = \sup_x \sigma(\nabla f(x))$, $\sigma(A)$ is the spectral norm of the matrix A, as shown in Equation (16):

$$\sigma(\mathbf{A}) = \max \frac{\| \mathbf{A}x \|_2}{\| x \|_2} = \max_{\|x\|_2 \leqslant 1} \| \mathbf{A}x \|_2 \tag{16}$$

For each layer g of the network, with h as the input, the bias term is not discussed in this paper, and g(h) = Wh; then:

$$\| g \|_{Lip} = \sup_k \sigma(\nabla g(h)) = \sup_h \sigma(W) = \sigma(W) \tag{17}$$

In the entire network, $f(x) = W^{L+1}a_L(W^L(a_{L-1}(W^{L-1}(\cdots_{a_1}(W^1x)\cdots))))$. Because of $\| g_1 \circ g_2 \| \leq \leq \| g_1 \|_{\text{Lip}} \bullet \| g_2 \|_{\text{Lip}}$, there is:

$$\| f \|_{Lip} \leq \prod_{l=1}^{L+1} \sigma\left(W^l\right) \tag{18}$$

Here, L is the number of network layers, and Wl is the weight of the L-th layer. To limit the Lipschitz constant of each layer to 1, $\| g \|_{\text{Lip}} = 1, \sigma(W) = 1$ is required. Therefore, by applying spectral normalization to the weight matrix W as shown in Equation (19), $\sigma(W) = 1$ is obtained.

$$W\text{sn} = \frac{W}{\sigma(W)} \tag{19}$$

In the LRGAN model, the ReLU activation function, the sigmoid activation function, and the spectral normalization operation in the discriminator all satisfy the Lipschitz constraint. To ensure that the entire discriminator meets the Lipschitz constraint, it is sufficient to divide the parameters of each convolutional kernel by the spectral norm of the corresponding convolutional kernel matrix. As shown in Figure 5, the discriminator in this paper is a simple five-layer nested convolutional neural network, where all layers use the ReLU activation function except for the output layer, which uses the sigmoid activation function. Using bounded activations allows the network to quickly learn to reach saturation, especially for high-resolution images, where using the ReLU activation function in the discriminator works very well.

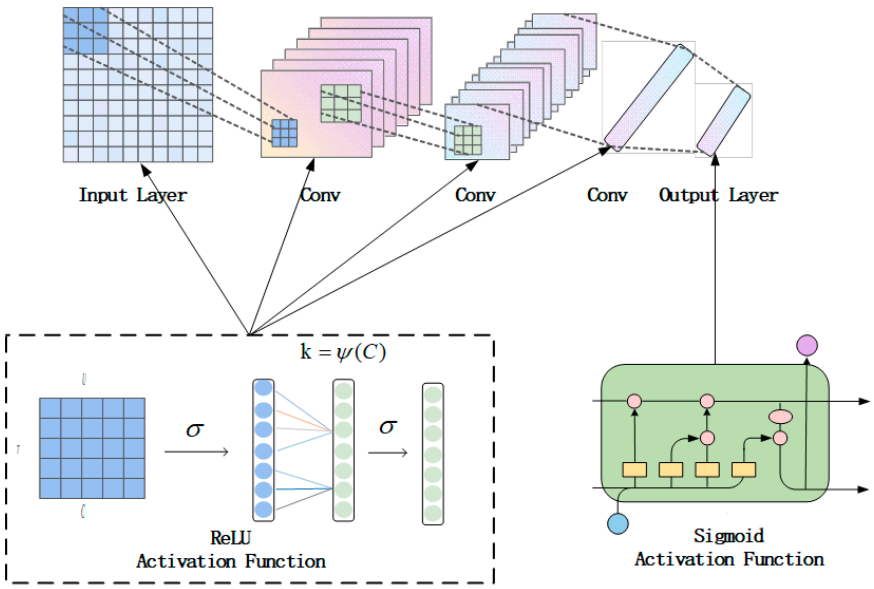


Figure 5. Activation function layer diagram.

To ensure that the five-layer discriminator network satisfies the Lipschitz condition, the maximum singular value of each layer's weight matrix must be equal to 1. The discriminator can use spectral normalization to make the maximum singular value of each layer's weight matrix equal to 1, thereby limiting the gradient size of the discriminator to within 1 without altering the structure of the weight matrix. This helps make the adversarial training process more stable and easier to converge. Therefore, introducing spectral normalization [22] in the discriminator network can further constrain the discriminator's performance, preventing it from being trained too quickly and thereby improving the network's overall performance.

2.2. The Generator Part of the LRGAN Model

A residual module was introduced in the generator network, with the number of residual network layers set to 9. ResNet commonly uses standard residual depths such as 9, 18, 34, and 50 layers in image processing tasks. For medium-resolution generation tasks (such as 128 × 128 or 256 × 256 images), the ResNet-9 structure is a widely validated lightweight configuration that can effectively learn features without causing gradient explosion or overfitting. The use of a nine-layer residual structure in the generator is primarily inspired by the widespread application of lightweight ResNet architectures in image generation tasks. The nine layers enhance the expressive power of the generator without significantly increasing the training burden, and effectively mitigate the mode collapse issue caused by overly deep networks.

2.2.1. The Generator Structure of the LRGAN Model

The generator structure in the LRGAN model is shown in Figure 6.

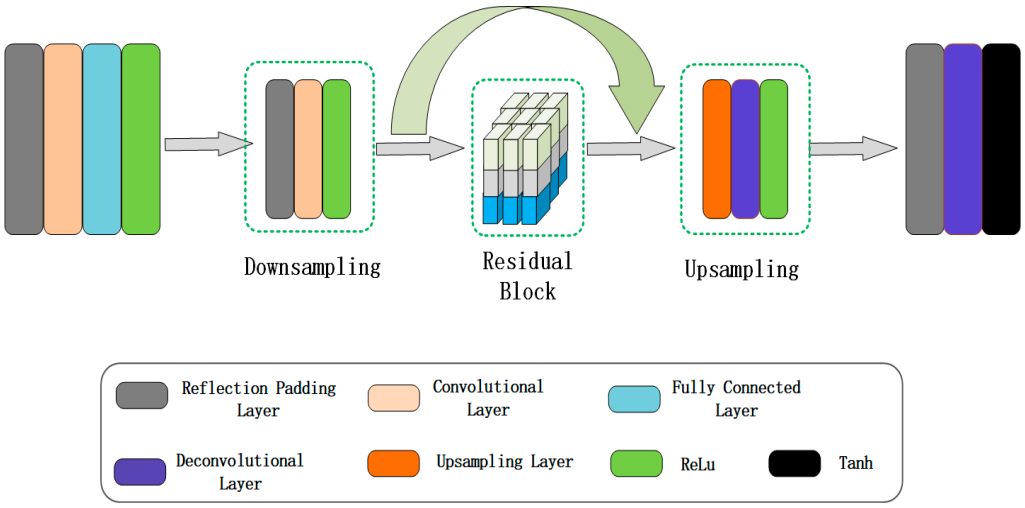


Figure 6. LRGAN generator structure diagram.

2.2.2. Generator Activation Functions

In the backpropagation process of neural networks, division is required to compute the derivative of the activation function in order to obtain the error gradients. This can result in high computational complexity and is prone to the vanishing gradient problem. The sigmoid function, when used in deep neural networks during backpropagation, exhibits a low rate of change in the saturated region, causing its derivative to approach zero. As a result, this can lead to the loss of information during propagation, resulting in information saturation, which hinders normal training of deep networks. Therefore, in the output layer of the generator, this paper abandons the sigmoid activation function with a Lipschitz constant of 1/4 and instead opts for the tanh activation function with a Lipschitz constant of 1 to better control the output range of the generator and avoid information saturation. The expression of the tanh function is shown in Equation (20), and the function curve is depicted in Figure 7.

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{20}$$

As shown in Figure 7, the Tanh function is centrally symmetric with a mean of 0, and it can map a 0–1 Gaussian distribution to values near 0, maintaining the zero mean property. Additionally, because it has both positive and negative values, it allows for faster discrimination of inputs from the previous layer. This leads to faster convergence and

helps mitigate the issue of high computational complexity caused by backpropagation. It reduces the computational overhead in multi-layer networks, making it more suitable for the generator's requirement of multi-layer network backpropagation.

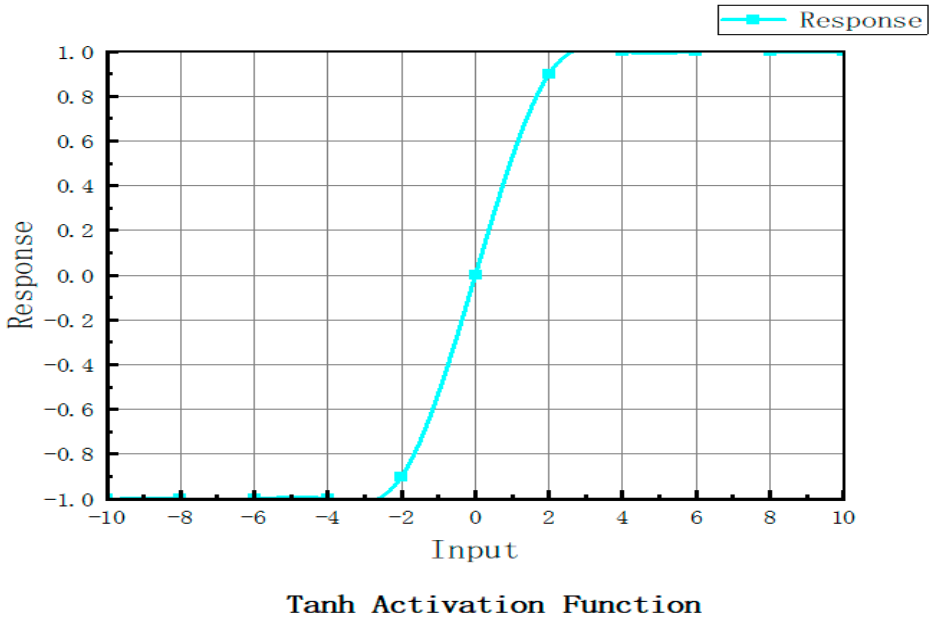


Figure 7. Generator activation function.

2.2.3. Nine-Layer Residual Network

To prevent the model from collapsing due to the accumulation of network layers, this paper introduces a residual module into the generator, with the number of residual network layers set to nine. The last layer uses a tanh activation function, while the other layers use the ReLU activation function. With this nine-layer residual network design, the module can avoid introducing additional computational complexity or increasing the risk of overfitting when expanding the network's depth and width. At the same time, the residual connections ensure the continuity and effectiveness of deep features, allowing information to flow freely without the need for complex nonlinear transformations. The structure of the nine-layer residual generator is shown in Table 1.

Table 1. Input and output of each layer in the residual module.

Network Layer	Input Feature Map (H × W × C)	Output Feature Map (H × W × C)
ReLU, Conv1, BN	128 × 128 × 1	64 × 64 × 64
ReLU, Conv2, BN	64 × 64 × 64	32 × 32 × 128
ReLU, Conv3, BN	32 × 32 × 128	16 × 16 × 256
ReLU, Conv4, BN	16 × 16 × 256	8 × 8 × 512
ReLU, Conv5, BN	8 × 8 × 256	16 × 16 × 512
ReLU, DeConv1, BN	8 × 8 × 1024	16 × 16 × 256
ReLU, DeConv2, BN	16 × 16 × 512	32 × 32 × 128
ReLU, DeConv3, BN	32 × 32 × 256	64 × 64 × 64
Tanh, DeConv4	64 × 64 × 128	128 × 128 × 1

In the generator, the residual block allows the neural network to skip one or more layers without adding extra parameters or increasing computational complexity. The residual module increases the network's depth, enhancing its expressive power, while effectively preventing overfitting and addressing training instability caused by gradient explosion.

3. Experimental Process

3.1. Data Preparation and Experimental Procedures

The industrial camera used in the experiment is the MV-CA004-10UC advanced area scan camera from Hikrobot. The camera uses the Song IM287CMOS chip, and the soldering site is shown in Figure 8.



Figure 8. Welding site pictures.

During the weld seam tracking process, 36 video sets were captured. Due to the high similarity between consecutive frames, one image was selected every 20 frames when capturing images from the video. In total, 2513 laser welding stripe images of 1280 × 700 size were obtained. Although the similarity of noise such as arcs and spatter within the same welding path is low, the similarity of the laser stripes remains high. Therefore, after manually reviewing the 2513 welding seam images, 1468 images with lower similarity were selected for the dataset, as shown in Figure 9.

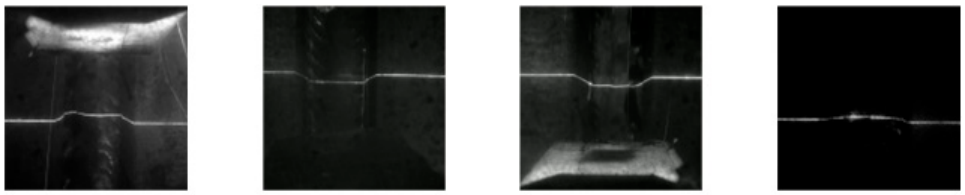
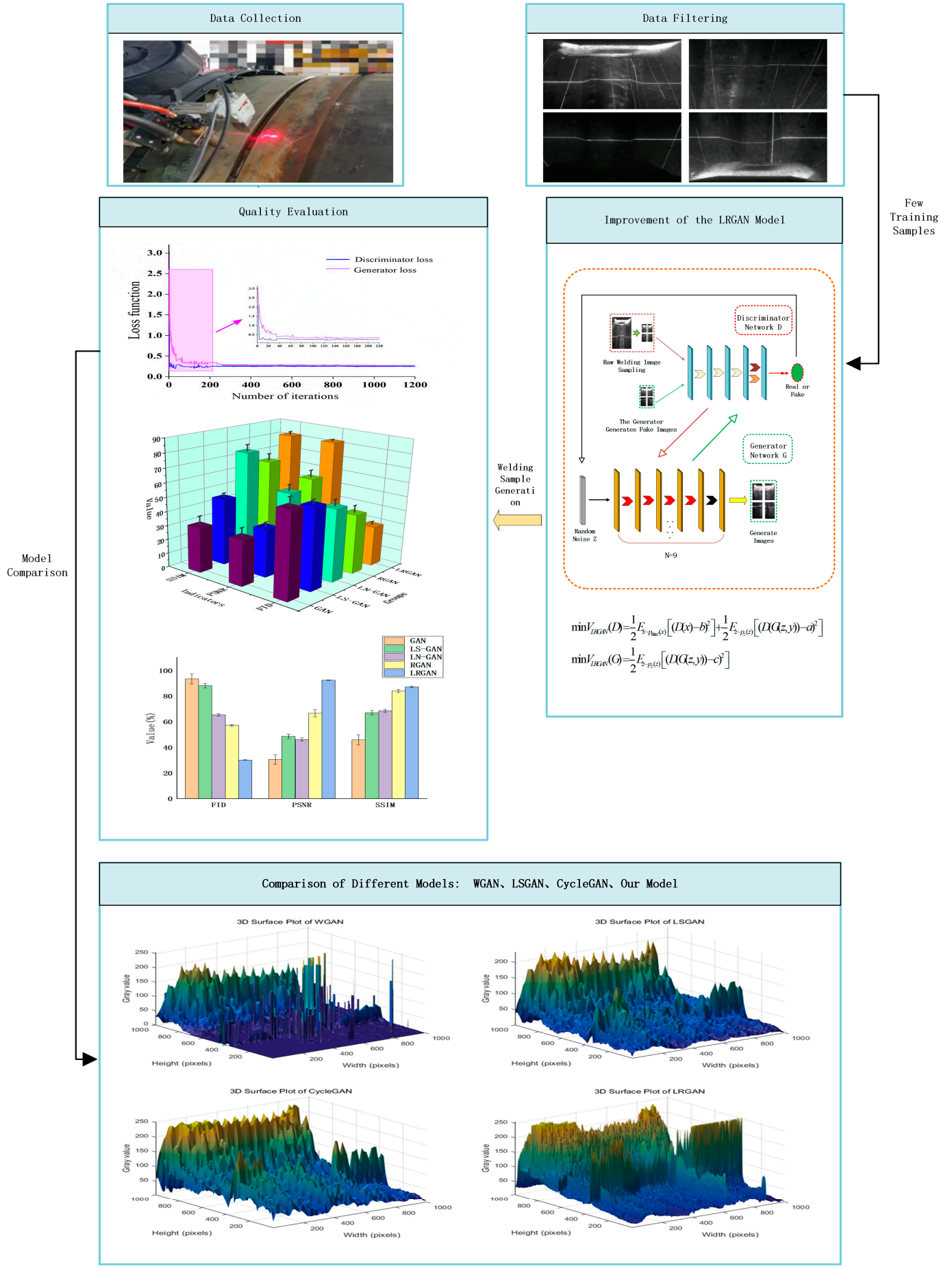


Figure 9. Original collection welding diagrams.

To validate the effectiveness of the proposed model, an experimental process was designed as shown in Figure 10.

3.2. Experimental Environment

The experimental environment in this study consists of the following system setup: Windows 10. The hardware and parameters used for training include an NVIDIA Tesla P100 GPU and 60 GB of memory. The software environment includes Pytorch 3.7.3 and CUDA 11.4, while the programming environment is Python 3.8. Programming was conducted in Python using PyCharm 2023 software. The experiment is based on the Pytorch 3.7.3 deep learning framework for network model construction, training, and testing.



$$\min V_{LRGAN}(D) = \frac{1}{2}E_{x\sim p_{data}(x)}\left[(D(x)-b)^2\right] + \frac{1}{2}E_{z\sim p_z(z)}\left[(D(G(z,y))-a)^2\right]$$

$$\min V_{LRGAN}(G) = \frac{1}{2}E_{z\sim p_z(z)}\left[(D(G(z,y))-c)^2\right]$$

Figure 10. Overall experimental diagram.

3.3. LRGAN Model Training

Using the alignment tool of OpenFace, the 1468 images were processed and standardized to a size of 768 × 480 pixels. These images were then divided into a training set and a testing set, with 900 images in the training set and 568 images in the testing set. The model was run for 50 epochs, with 2000 iterations per epoch, and the variations in the discriminator and generator loss functions during the image generation process were monitored. To prevent multiple batches from having the same data during iterations, the dataset was randomly shuffled before each training session, which helped accelerate the model's convergence speed. The training set was used as input for the LRGAN model, and the loss function variations during the training process are shown in Figure 11.

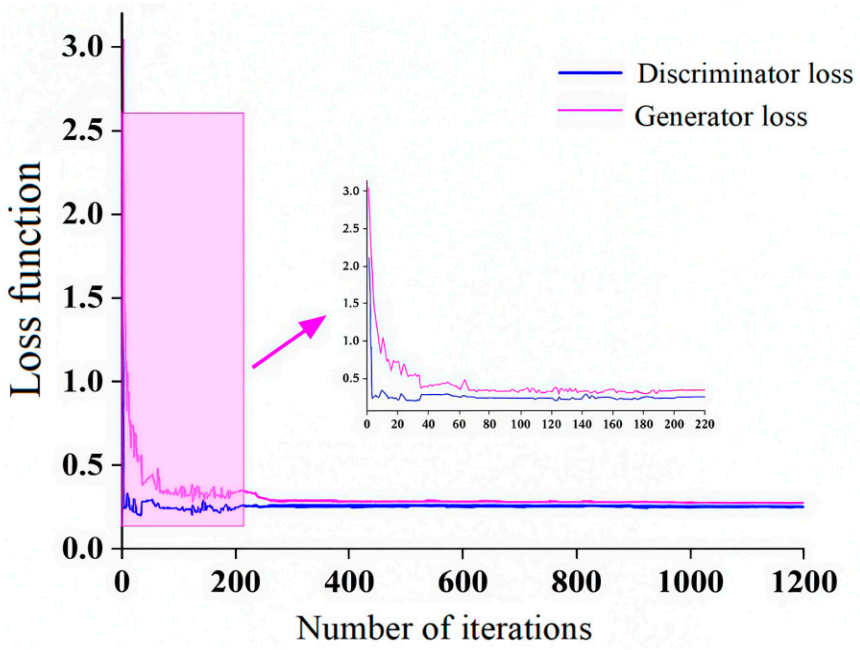


Figure 11. LRGAN loss function diagram.

As shown in Figure 11, at the beginning of the training, the generator has a higher initial loss value compared to the discriminator. However, after 15 iterations, both losses begin to decrease, with the discriminator's loss decreasing more significantly. After 40 iterations, the generator's loss starts to slightly increase, but the fluctuation is not large, remaining within the range of 0.3. This indicates that the generator initially learned some features and patterns from the data, which led to a reduction in the loss function. However, as the model continues to train, due to the complexity of the data and the inadequacy of the model structure, the loss function begins to rise. After 90 iterations, both the discriminator and generator loss values drop below 0.5 and remain stable with small fluctuations, indicating that the model is approaching a Nash equilibrium state. This suggests that the original and generated images fed to the discriminator have gradually become more similar, with the discriminator's loss stabilizing and the generated images becoming increasingly realistic.

3.4. Generate Image Evaluation Metrics

Peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and Fréchet inception distance (FID) can be used to quantitatively evaluate generated images. PSNR is an objective evaluation method that reflects the error between corresponding pixel points in images. SSIM, on the other hand, measures the quality of generated images from three aspects: luminance, contrast, and structure, which aligns more closely with human visual perception. FID, as one of the metrics for measuring the feature distance between real

and generated images, can represent both the diversity of images and the quality of the generated images. The detailed indicator information is shown in Table 2.

Table 2. Evaluation metrics.

Evaluation Metrics	Evaluation Method	Numerical Representation
PSNR	Reflecting the error between corresponding pixel points in the image	A larger value indicates better quality of the image being evaluated
SSIM	Measuring the generated image from three aspects: luminance, contrast, and structure	The range of structural similarity (SSIM) is from 0 to 1; the closer the value is to 1, the more similar the image is
FID	Measuring the feature distance between real and generated images, by calculating the distribution distance between the generated image and the real image to evaluate the quality of the image	The smaller the FID value, the closer the generated image is to the real image

4. Experimental Results and Discussion

4.1. Ablation Experiment

This paper uses ablation experiments to validate the performance improvement of different modifications in the LRGAN model. Here, GAN refers to the original generative adversarial network; LS-GAN represents the model where the least squares loss function is introduced on top of GAN; LN-GAN refers to the model where spectral normalization is introduced into LS-GAN; RGAN refers to the model where residual networks are incorporated into GAN; and LRGAN is the model proposed in this paper. The experimental results of these five methods on the training set are shown in Table 3 and Figure 12.

Table 3. Ablation experiment.

Experiment	Methods	FID	PSNR	SSIM (%)
1	GAN	60.01	32.07	32.00
2	LS-GAN	57.07	33.98	47.00
3	LN-GAN	51.01	54.26	76.00
4	RGAN	41.95	60.03	66.00
5	LRGAN (Our)	28.31	82.33	82.00

As shown in Figure 12, compared to GAN, LS-GAN improves PSNR by 5.9%, reduces FID by 4.9%, and the SSIM value is 0.47. Furthermore, after adding spectral normalization, the PSNR of the LN-GAN model increases by 59.7%, FID decreases by 10.6%, and the SSIM value is 0.76. Therefore, it can be seen that spectral normalization has a more significant effect on improving PSNR. Compared to LN-GAN, RGAN improves PSNR by 10.6%, reduces FID by 17.8%, and the SSIM value is 0.66. The FID value is smaller and decreases more quickly, indicating that the residual structure is more helpful in reducing the FID value. In comparison to RGAN, LRGAN improves PSNR by 37.1%, and reduces FID by 32.5%, with the smallest error bars among all groups. Additionally, in the five comparison experiments, LRGAN achieves the highest PSNR, the lowest FID, and an SSIM value of 0.82, which is closest to 1, indicating that the generated images in this group are most similar to the original welding images.

To further validate the effectiveness of the improvement methods, experiments were also conducted on the test set, and the results are shown in Figure 13. LRGAN has the smallest FID value, and the highest PSNR and SSIM, with the smallest error in this group.

A comparative analysis shows that LRGAN performs the best. From the error bars in the figure above, it can be seen that the overall error from GAN to LRGAN generally shows a gradual downward trend, with the three evaluation metrics in LRGAN having the smallest errors. The experimental results on the test set further confirm that using the least squares loss function as the adversarial network loss function, as well as adopting spectral normalization and residual structures, all contribute to improving the quality of images generated by LRGAN. Spectral normalization has a more significant effect on improving PSNR, while the residual structure is more helpful in reducing the FID value. The least squares loss function plays the largest role in controlling SSIM, making its value closest to 1 (as shown in the figure above, which represents the value closest to 100%).

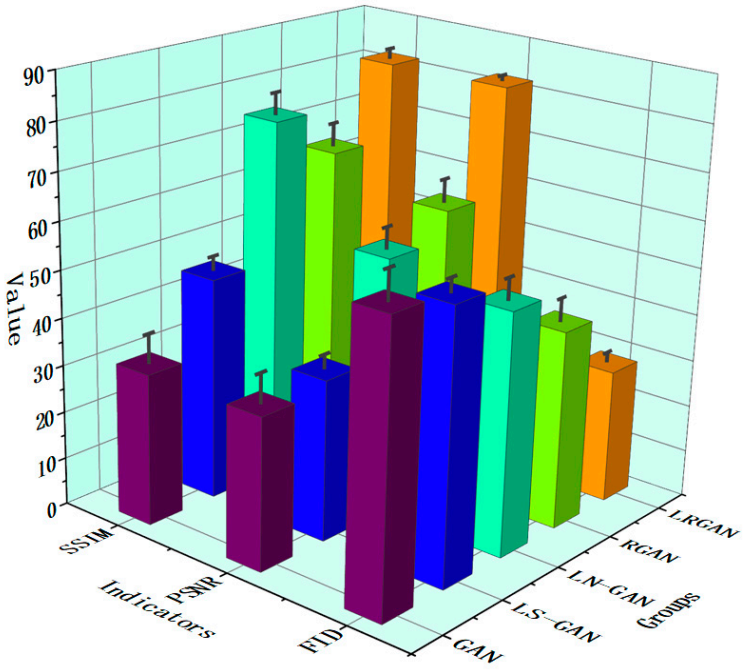


Figure 12. Training set ablation experiment 3D effectiveness visualization.

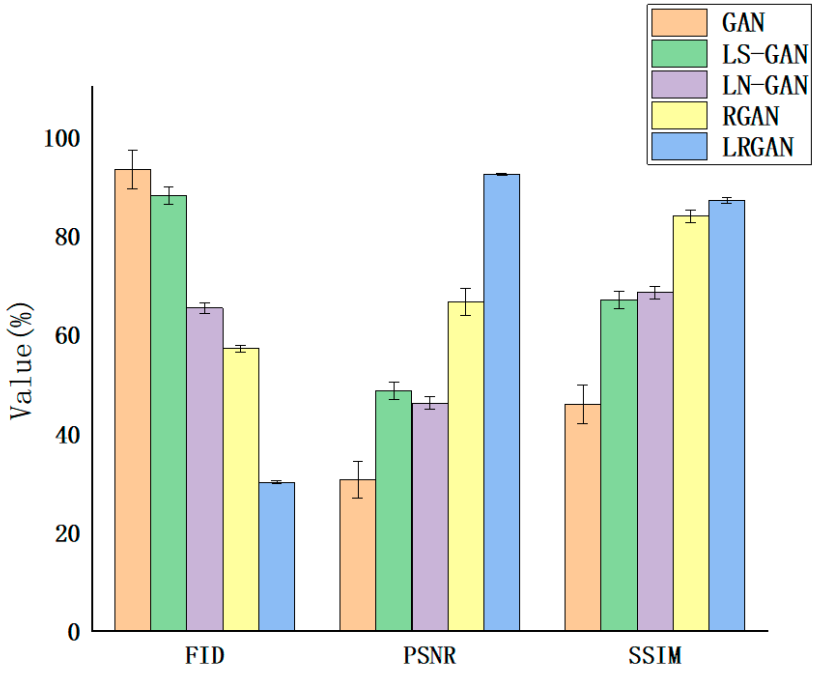


Figure 13. Testing set ablation experiment effectiveness comparison diagram.

4.2. Comparison of Images Generated by Different Models

4.2.1. Subjective Evaluation of the Results

To further prove the effectiveness of the proposed LRGAN model, a comparative analysis was conducted with three adversarial generation models: WGAN, LSGAN, and CycleGAN. Simulations were carried out while ensuring that the model parameters remained essentially the same. The number of iterations was set to 110, and the images generated by each model are shown in Figure 14.

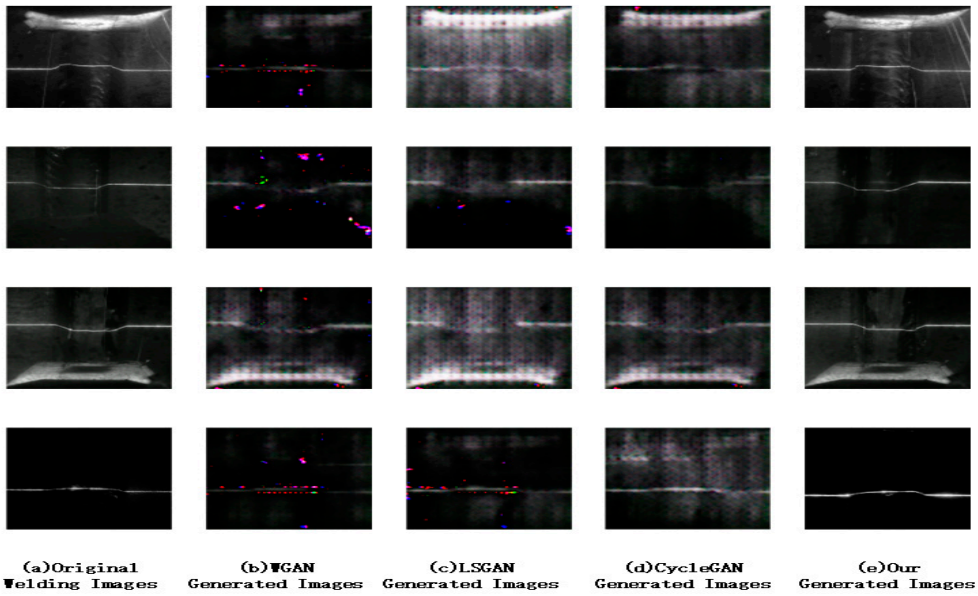


Figure 14. Images generated by different models.

From Figure 14, it can be seen that from a visual perception perspective, the WGAN model still exhibits a large amount of noise distributed along the welding centerline and its vicinity. The LSGAN model, due to the decision penalty mechanism of the least squares loss function, significantly reduces explicit noise during the training process, but its resolution still needs improvement. The images generated by the CycleGAN model are quite similar to the original welding images, with a significant reduction in blurriness. The images generated by the LRGAN model are clearly superior to the previous three models, more closely resembling the original welding images and showing the best results.

4.2.2. Grayscale 3D Surface Analysis

To objectively evaluate the performance of each model, this paper uses grayscale surface plots to display the characteristics and trends of the data. The grayscale surface plots provide a more intuitive way to demonstrate the improvements in brightness and contrast of the images generated by each model. Taking the images in the third row of Figure 14 as an example, the corresponding grayscale surface plot is shown in Figure 15.

In the grayscale surface plot, the x-axis represents the image height, the y-axis represents the image width, and the z-axis represents the grayscale value (ranging from 0 to 255). The lower the grayscale value, the darker the color on the surface plot. By comparing the enhancement results obtained from different methods, we can observe that: the grayscale value difference of the generated image using the WGAN method is too large, indicating an uneven brightness distribution with overly bright and dark areas; the grayscale value difference of the generated image using the LSGAN and CycleGAN methods is somewhat reduced, but there are still dark areas, and abnormal dark regions appear at both the image edges and center, which is an important cause of the blurriness of the generated images; the

grayscale value difference of the welding images generated by the LRGAN is the smallest, indicating that the pixel distribution of the generated image is more balanced, effectively improving the quality of the generated welding images. Therefore, the proposed method performs better.

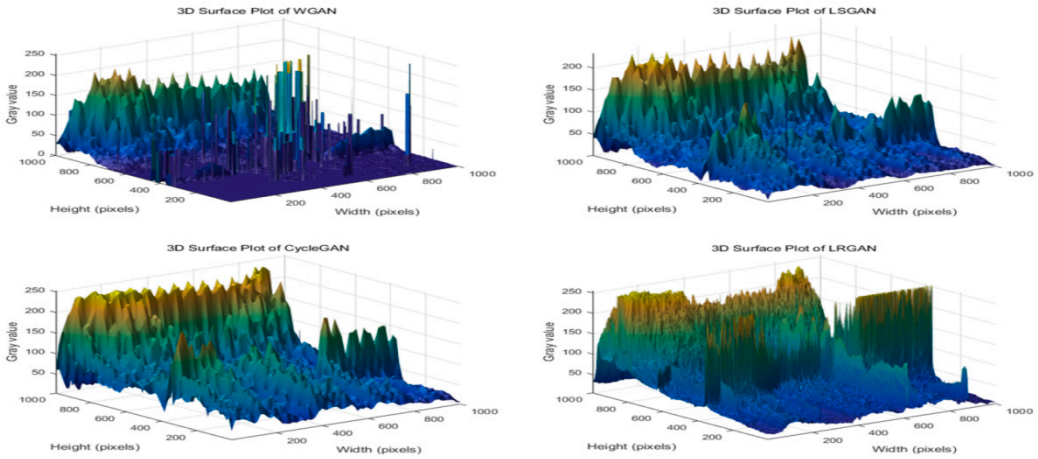


Figure 15. 3D surface plot visualization of different models.

4.2.3. Quantitative Analysis

In addition to analyzing the model's generation performance using the PSNR, SSIM, and FID evaluation metrics, we can also assess the LRGAN model's performance from the perspective of large data volumes and manifold structure evaluation by introducing the GAN_train and GAN_test metrics. These metrics, obtained by training on the VGG19 classification network, are mainly used to evaluate the manifold structure similarity between datasets. The higher the GAN_train and GAN_test values, the more similar the manifold structures of the datasets. The results are shown in Table 4 and Figure 16.

Table 4. Quantitative analysis of different models.

Methods	FID	PSNR	SSIM	GAN_Train	GAN_Test
WGAN	87.93	76.03	52.01	57.67	67.25
LSGAN	56.52	86.54	74.50	88.00	43.33
CycleGAN	48.32	60.02	80.33	66.71	57.30
LRGAN	42.11	90.31	89.72	92.00	70.61

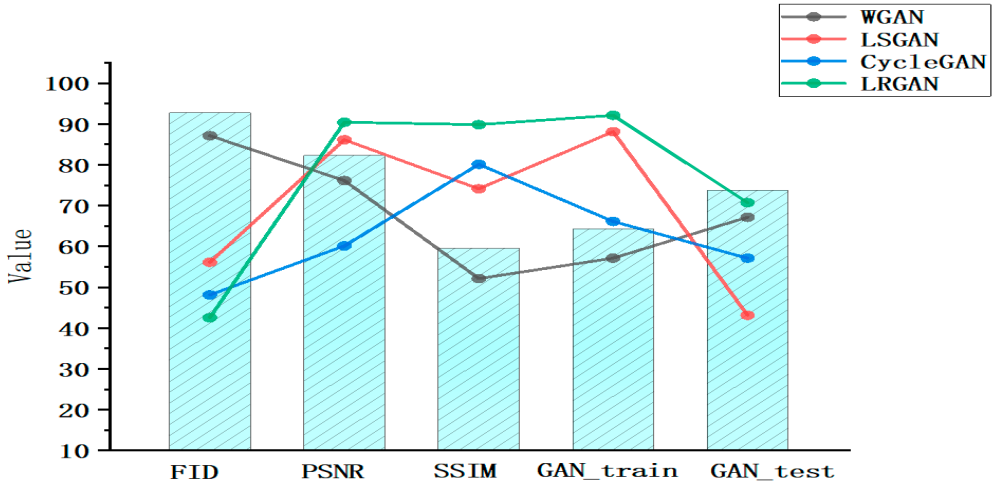


Figure 16. Comparison of model effectiveness by different models.

Through quantitative analysis, it can be seen that the proposed model outperforms the other three models in terms of having the smallest FID, the highest PSNR, and SSIM closest to 1 (100%), as well as achieving the highest manifold structure similarity evaluation (GAN_train and GAN_test). Based on these five objective evaluation metrics, it can be concluded that the LRGAN model better preserves the feature distribution and structural similarity between the generated welding image dataset and the original welding image dataset, resulting in the best image generation effect, which is closest to the original welding images.

Furthermore, based on the 3D area plot in Figure 17, it can be concluded that the area blocks corresponding to the conditions of small FID values and large PSNR, SSIM, as well as GAN_train and GAN_test values, are primarily concentrated around LRGAN (i.e., the proposed model in this paper). This indicates that the number of area blocks meeting these conditions is highest for the proposed model. Therefore, this further confirms that, among the four models, the LRGAN model proposed in this paper generates welding images that are the most similar to the original welding samples, demonstrating the best performance.

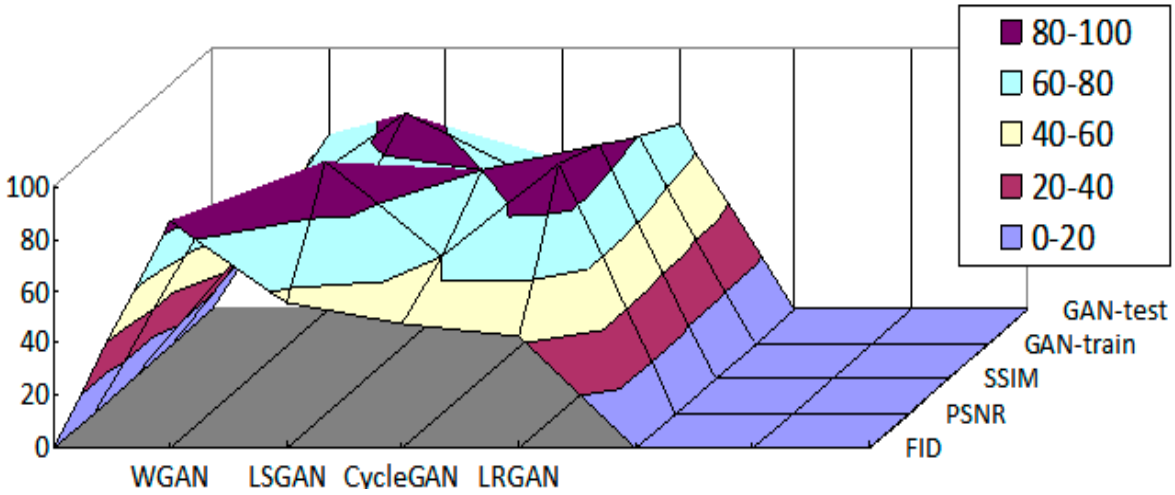


Figure 17. Different models' effectiveness 3D area visualization.

4.3. The Optimal Number of Samples for LRGAN in Few-Shot Learning

Based on the final selection of 1468 original welding images with lower similarity, 400, 300, and 200 original samples were respectively chosen for the training of LRGAN, with the same test set used for validation. The experimental results comparing different sample sizes are shown in Table 5.

Table 5. Optimal sample size.

Original Sample Quantity	PSNR	SSIM	FID
400	63.30	59.98	72.20
300	82.22	88.13	46.32
200	84.03	82.33	41.90

From Table 5, it can be concluded that as the number of original samples input into LRGAN decreases, each evaluation metric either increases or decreases to some extent. From 400 original samples to 200 original samples, PSNR and SSIM increased by 32 and 37 percentage points, respectively, while FID decreased by 42 percentage points. This indicates that the quantity of original input data directly affects the quality of the generated LRGAN samples and determines the similarity between the generated samples and the original samples. When comparing the 300-sample case with the 200-sample case, the

values of PSNR, SSIM, and FID show minimal fluctuation. However, the SSIM value for the 300-sample case is closer to 1 (100%). Therefore, based on this generative model (LRGAN), the optimal small sample size is 300.

5. Conclusions

This paper proposes a welding image generation method based on the LRGAN model. This method utilizes an adversarial learning mechanism to deeply mine the features from a small number of real samples, and then generates similar welding data, effectively addressing the problem of insufficient samples in the welding industry. The results of various comparative experiments show the following:

(1) The least squares loss function is used to replace the original cross-entropy loss function, leading to a more stable training process. The discriminator becomes stricter and more precise in classification, making the generated images from the generator more closely resemble the original images in terms of structure and details. The least squares loss function contributes the most to controlling the SSIM and bringing its value closer to 1.
(2) Spectral normalization has a more significant effect on improving PSNR. By constraining the discriminator's Lipschitz continuity, it increases the PSNR value and enhances the model's ability to capture the overall clarity of the image.
(3) The residual structure is more helpful in reducing the FID value. It helps retain the complex texture structures of the image and reduces mode collapse.
(4) The number of real data samples directly impacts the quality of the data generated by the LRGAN model. Comparative experiments with different quantities of real samples show that when the number of real data is 300, the LRGAN model achieves optimal overall performance, and the SSIM value is closer to 1. This indicates that, under this data quantity, the LRGAN model performs optimally and demonstrates good adaptability and generalization ability for small sample sizes.

However, this study still has certain limitations. Firstly, the current model is primarily validated in the specific industrial scenario of welding images, and its adaptability in other types of industrial images or general image generation tasks still requires further exploration. Secondly, while spectral normalization and residual structures improve model performance, they also increase training complexity and computational cost, which may pose challenges for deployment on resource-constrained edge devices.

Future research can be pursued in the following directions:

(1) Explore more lightweight network architectures to balance performance and deployment efficiency.
(2) Introduce self-supervised learning or contrastive learning mechanisms to further enhance the model's learning ability under ultra-small sample conditions.
(3) Extend this method to the generation of multi-modal or 3D welding data to meet a broader range of industrial needs.

Author Contributions: Conceptualization, Y.W. and Z.D.; methodology, Z.D.; software, Z.D.; validation, Z.D.; investigation, Z.D., Q.Z. and Z.H.; writing—review and editing, Y.W. and Z.D.; supervision, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, G. Welding Quality Inspection and Management of Low-Temperature Pressure Vessels. *Pop. Stand.* **2024**, *14*, 25–27.
2. Zhou, X.; Zhao, K.; Liu, J.; Chen, C. Quality Detection of Short-Cycle Arc Stud Welding Based on Deep Learning. *Mod. Manuf. Eng.* **2025**, *1*, 87–93. [CrossRef]
3. Qu, J. Research on Welding Defect Detection of New Energy Vehicles Based on Improved Generative Adversarial Networks. *Weld. Technol.* **2024**, *53*, 120–125. [CrossRef]
4. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Commun. ACM* **2020**, *63*, 139–144.
5. He, Y.; Zhou, X.; Jin, J.; Song, T. PE-CycleGAN network based CBCT-sCT generation for nasopharyngeal carsinoma adaptive radiotherapy. *J. South. Med. Univ.* **2025**, *45*, 179–186. [CrossRef] [PubMed] [PubMed Central]
6. Du, H.; Yuan, X.; Liu, X.; Zhu, L. Generative adversarial network image restoration algorithm based on diffusion process. *J. Nanjing Univ. Inf. Sci. Technol.* **2024**, 1–11. [CrossRef]
7. Hu, M.; Luo, C. Bearing Fault Diagnosis Based on MCNN-LSTM and Cross-Entropy Loss Function. *Manuf. Technol. Mach. Tools* **2024**, *9*, 16–22. [CrossRef]
8. Liang, X.; Xing, H.; Gu, W.; Hou, T.; Ni, Z.; Wang, X. Hybrid Gaussian Network Intrusion Detection Method Based on CGAN and E-GraphSAGE. *Instrumentation* **2024**, *11*, 24–35. [CrossRef]
9. Kang, M.; Shim, W.J.; Cho, M.; Park, J. Rebooting ACGAN: Auxiliary classifier GANs with stable training. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 1–13.
10. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
11. Choi, Y.; Uh, Y.J.; Yoo, J.; Ha, J.-W. StarGAN v2:diverse image synthesis for multiple domains. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 8188–8197.
12. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier GANs. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2642–2651.
13. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutionalgenerative adversarlal net-works EB. *arXiv* **2015**, arXiv:1511.06434.2015.
14. Martin, A.; Soumith, C.; Léon, B. Wasserstein GAN. In Proceedings of the 34th International Conference on Machine Learning Representations, Sydney, Australia, 6–11 August 2017.
15. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of Wasserstein GANs. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; MIT Press: Cambridge, MA, USA, 2017; pp. 5767–5777.
16. Li, X.; Ge, X.; Yang, J. Bayesian Optimization-Based WGAN-GP for fNIRS Data Augmentation and Emotion Recognition. *J. Zhengzhou Univ.* **2025**, 1–8. [CrossRef]
17. Jiang, Y.; Chang, S.; Wang, Z. TransGAN: Two pure transformers can make one strong GAN, and that can scale up. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 9 December 2022; MIT Press: Cambridge, MA, USA, 2022; pp. 14745–14758.
18. Pan, T.; Xiong, W. Tiny Fault Detection and Diagnosis Based on Minimizing KL Divergence. In Proceedings of the 35th China Process Control Conference, Chinese Society of Automation, Yichang, China, 20–22 May 2023. [CrossRef]
19. Liu, S.W.; Jiang, H.K.; Wu, Z.H.; Li, X. Data synthesis using deep feature enhanced generative adversarial networks for rolling bearing imbalanced fault diagnosis. *Mech. Syst. Signal Process.* **2022**, *163*, 108139. [CrossRef]
20. He, Z.; Lv, L.; Chen, J.; Kang, P. Critic Feature Weighting Multi-Kernel Least Squares Twin Support Vector Machine. *Inf. Control* **2025**, *54*, 123–136. [CrossRef]
21. Shu, J.; Wang, X.; Li, L.; Lei, J.; He, J. Rotary Lid Defect Detection Method Based on Improved GANomaly Network. *J. South-Cent. Univ. Natl.* **2023**, *42*, 788–798. [CrossRef]
22. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. Spectral normalization for generative adversarial networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; NetPress: Ithaca, NY, USA, 2018.
23. Zhou, Z.; Liang, J.; Song, Y.; Yu, L.; Wang, H.; Zhang, W.; Yu, Y.; Zhang, Z. Lipschitz generative adversarial nets. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; ACM Press: New York, NY, USA, 2019; pp. 7584–7593.
24. Qin, L.; Feng, N. Sparse Data Feature Extraction Based on Deep Learning Backpropagation. *Comput. Simul.* **2022**, *39*, 333–336.

25. Tang, B.; Zhu, J.; Hu, A.; Zhu, M. Prediction Model for the Amount of Decoction Made by Automatic Herbal Decoction Machine Based on Backpropagation Artificial Neural Network for Fruits and Seeds. *Tradit. Chin. Med.* **2025**, *47*, 1386–1390.
26. Xiang, Y.; Zhao, X.; Huang, J. A Residual Connection-Based Swin Transformer Enhanced Joint Encoding Architecture Design. *Radio Eng.* **2025**, *55*, 905–912.
27. He, J.; Wang, C.; Wang, T. Multi-Layer Multi-Pass Weld Seam Recognition Based on Deep Residual Networks. *J. Tianjin Univ. Technol.* **2025**, *44*, 91–96.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.