A Fault Diagnosis Method for Gantry Crane Trolley Reducer Bearing Under the Condition of Dataset Imbalance Based on 1DCNN-CGAN

Changjie Wu[1, a], Lihan Jin[2, *]

[1] Shanghai Maritime University, Shanghai, China
[2]Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China

[a]E-mail: will_wcj@163.com
* Correspondence: 354861797@qq.com

Abstract. In recent years, the development of deep learning has made it more and more used to solve the problem of fault diagnosis of gantry cranes and achieved good results. However, the traditional fault diagnosis methods based on deep learning rely too much on the collective quantity and specification of data, and it is difficult to obtain the ideal generalization effect in the absence or very little of standardless data. In order to solve this problem in the case of faultless data, this paper proposes a fault diagnosis method for gantry crane trolley reducer based on deep convolution-conditioned generative adversarial network (1DCNN-CGAN). Firstly, based on the conditional generative adversarial network, this paper extracts the time-domain features of the open-source fault data and generates labeled fault data to obtain a small number of initial fault datasets. Secondly, this paper constructs a feature extractor based on the deep convolutional neural network to realize the effective noise reduction of the vibration acceleration data of the shipbuilding gantry crane upward trolley reducer. Finally, based on the working data of a gantry crane, the effectiveness of the proposed method is proved by comparing the diagnostic effects of the proposed method and the two related algorithms.

Keywords: Conditionally generative adversarial networks; Deep Learning; gantry crane; Troubleshooting

1 Introduction

As an integral part of gantry crane trolley reducer, the state of bearings can directly affect the running state of equipment [1]. Bearings are widely used in gantry crane trolley reducers, which often operate in severe working conditions such as high speed and heavy load [2-5]. However, working in a harsh environment for a long time, the reducer structure is easily affected by corrosion, wear and other factors, which leads to the failure of the reducer bearing, which in turn affects the normal operation of the gantry crane. In view of this, according to international standards and related regulations, such

S. M. Ghadiri et al. (eds.), *Proceedings of the 2025 4th International Conference on Engineering Management and Information Science (EMIS 2025)*, Advances in Computer Science Research 121,
https://doi.org/10.2991/978-94-6463-736-6_15

as mechanical equipment inspection standards and gantry crane manufacturing standards, it is essential to regularly inspect and maintain the gantry crane trolley reducer. These inspections include lubrication inspections, bearing wear inspections, etc., to ensure the safe operation of the gantry crane and prolong the service life of the reducer.

Traditional fault diagnosis methods are mainly based on physical signal processing techniques, such as oil analysis [6] and so on. These methods perform fault detection by analyzing physical signals such as wear particles in the lubricating oil. However, the traditional method has the limitation of relying on expert experience for feature selection and threshold setting, and it is also not suitable for gantry cranes under different working conditions. In recent years, the fault diagnosis technology of gantry crane based on deep learning has attracted much attention. These technologies use deep neural networks to automatically learn the complex features of the data and automate fault diagnosis. Such as, Li [7] proposed a method that combines angular domain resampling technology and deep convolutional neural network to diagnose the signal. Yan [8] used EMD method to extract the features of harbor crane bearing faults, and the fault diagnosis is carried out in combination with the support vector machine. These methods achieve end-to-end fault classification by directly extracting the high-order features of the original signal, and achieve good diagnostic performance.

However, the deep learning network needs a large amount of labeled fault data for training, and, in practice, labeled samples are often too scarcity, and most are unlabeled samples since the labeling task is time-consuming and laborious [9], which seriously limits the application of deep learning network in fault diagnosis. In order to solve this problem, Scholars have proposed many methods, Matharaarachchi [10] proposed an Enhancing SMOTE for imbalanced data with abnormal minority instances and the variational autoencoders (VAE) method commonly used by many scholars. Compared to simple linear interpolation with SMOTE, CGAN generates samples that are closer to the real data distribution through adversarial training, thereby avoiding issues related to noise and invalid samples. Compared to VAE, CGAN typically produce more realistic samples and greater diversity. In addition, CGAN supports conditional generation and is able to generate samples for specific categories, which is ideal for category imbalance problems. Although difficult to train, CGAN's flexibility and ability to generate multiple data types, such as images and text, make it a powerful tool for data augmentation. the researchers proposed a method of data augmentation using conditional generative adversarial networks [13,14]. For example, Hu [11] applied the deep conditional generation adversarial network to the pipeline valve fault data for data expansion. Liu [12] combined conditional generative adversarial network and deep convolutional neural network for fault diagnosis. However, these methods still require a small amount of labeled failure data when generating data. In actual industrial production, it is possible to encounter missing fault data, which severely limits the application of these methods.

This paper proposes a new method. Firstly, the deep convolutional neural network [17] is used to convert the vibration signal of the gantry crane bearing collected under actual working conditions into feature tensors, so as to better extract the high-dimensional features in the vibration data. Secondly, the vibration time domain features of the open-source fault bearing dataset are selected, and the conditional generative adversarial network is used to generate the fault feature tensor. Finally, the multi-layer perceptron is used for fault diagnosis, and the fault diagnosis of the gantry crane is realized under the condition of no fault data. The results show that this method can effectively improve the accuracy of bearing fault diagnosis under the condition of data imbalance.

2 Gantry crane reducer bearing failure analysis

Gantry cranes are heavy manufacturing equipment, and the working scene is usually located in harsh environments such as docks and storage yards, and may be affected by storms, strong winds, extreme high temperatures, low temperatures and other bad weather, as well as seawater erosion; At the same time, the continuous operation of high intensity and variable load makes vulnerable parts such as gantry crane trolley reducer more susceptible to damage. The schematic diagram of the reducer bearing is shown in Figure 1, and the common damage of the reducer bearing is mainly divided into three types: inner ring failure, outer ring failure, and rolling element failure, and the details are shown in Table 1.

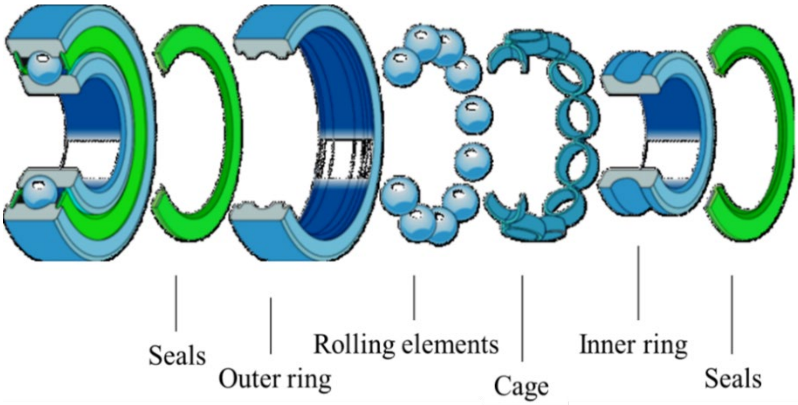
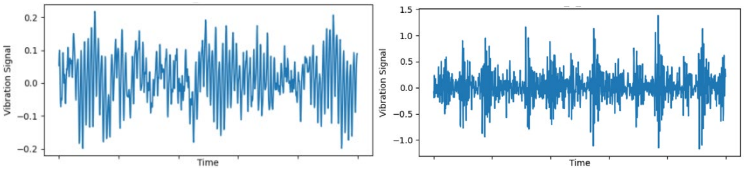


Fig. 1. Schematic diagram of bearing structure

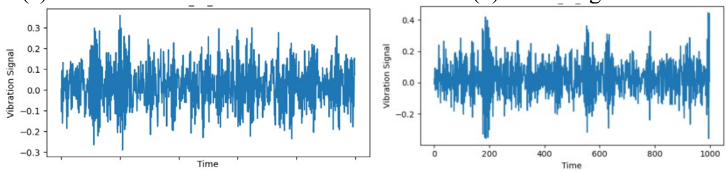
Table 1. Common damage to reducer bearings

Fault number	Fault type	Causes	Consequences
Crack_1	Inner ring failure	This can be due to overload, improper assembly insufficient lubrication, or a problem with the quality of the bearing material.	It may lead to the degradation of the performance of the reducer bearing, the increase of noise, and even the damage to the equipment, affecting the production efficiency.
Crack_2	Outer ring failure	It can be caused by improper installation, external vibration, environmental corrosion, etc.	
Crack_3	Rolling element failure	It can be caused by surface damage, fatigue, overload, etc.	

These faults can lead to abnormal vibrations during bearing operation, which are visually reflected in the vibration acceleration data, as shown in Figure 2:



(a) Normal vibration acceleration data (b) Inner ring failure



(c) Outer ring failure (d) Rolling element failure

Fig. 2. Vibration acceleration data under different fault conditions

According to China's special equipment safety regulations and related standards, the maintenance cycle of special equipment is generally based on the use of equipment and the manufacturer's recommendations, so vulnerable parts such as shipbuilding gantry crane trolley reducer need higher frequency maintenance than other equipment, which is time-consuming and laborious, and by identifying the vibration acceleration data when the bearing is running, it can accurately judge whether there is a fault, and can greatly reduce the labor cost of the enterprise. Therefore, a trouble-free data gantry crane trolley reducer fault diagnosis method based on 1DCNN-CGAN was proposed.

3 Fault diagnosis algorithm of gantry crane trolley reducer based on improved CGAN

3.1 The overall architecture of the algorithm

Due to the advantages of CGAN for supervised training according to feature conditions, this paper uses the deep convolutional network to clean the data, build a generator and discriminator, and optimize and improve the model and parameter settings according to the characteristics of the dataset, and finally realize the generation of high-quality fault data. The algorithm mainly includes three parts: (1) feature extractor: a deep convolutional neural network composed of three convolutional layers and one pooling layer is adopted; (2) Conditional Generative Adversarial Network (CGAN): It is composed of discriminant network D and generative network G. (3) Fault diagnosticator: a multi-layer perceptron neural network composed of three fully connected linear layers. The complete algorithm is shown in Figure 3.

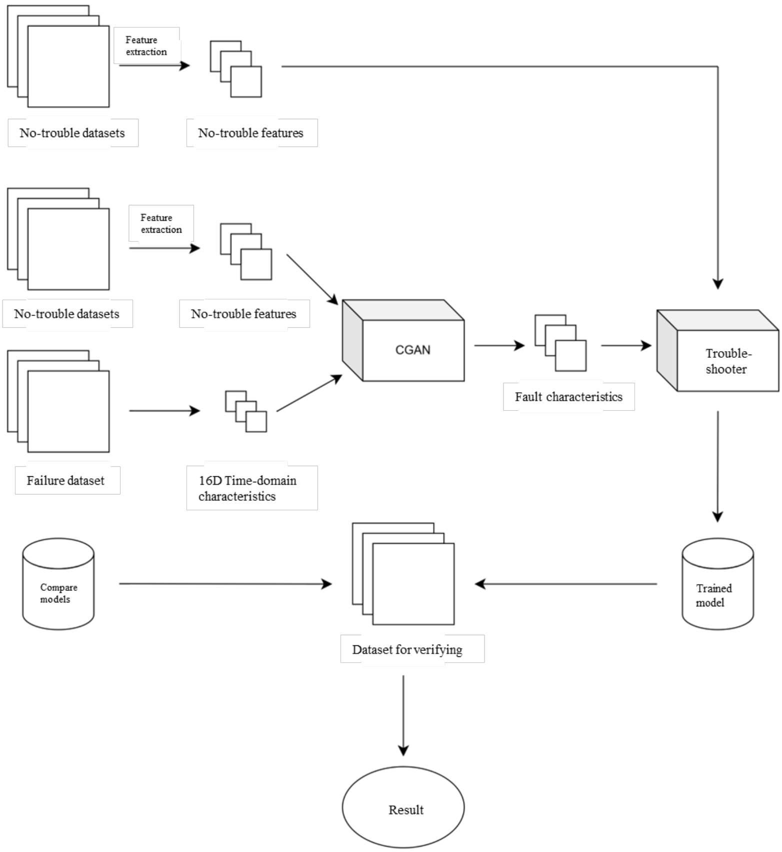


Fig. 3. Algorithm architecture diagram

3.2 Feature extractor

The feature extractor is a 1DCNN that contains three convolutional layers and three maximum pooling layers. The input layer is a 1-dimensional tensor with two channels. The first layer of convolutional layers outputs 16 channels, the second layer outputs 32 channels, and the third layer outputs 64 channels. Each convolutional layer is followed by a ReLU [18] activation function. Each convolutional layer is followed by a maximum pooling layer with a pooling kernel size of 2, as shown in Table 2 [19,20]. The output of the feature extractor is a representative, deep feature that reflects the health of the mechanical system. In this article, the input to the feature extractor is raw vibration acceleration data. Suppose the input is a normal signal x(t), consisting of many data points. Normal signals are directly put into the extractor for deep feature extraction. After feature extraction, a low-dimensional depth feature set can be obtained, as follows:

$$fs(x) = H(x(t))^{[15]} \tag{1}$$

H represents the nonlinear feature transformation in the feature extractor, and fs is the extracted feature set.

Table 2. Feature extractor network architecture

Network layer	Nucleus size	Step	Number of output channels
The first layer of convolution conv1	3	1	16
ReLU activation layer			
Pooling layer	2	2	
The second layer of convolution conv2	3	1	32
ReLU activation layer			
Pooling layer	2	2	
The third layer of convolution conv3	3	1	64
ReLU activation layer			
Pooling layer	2	2	

3.3 CGAN

CGAN is composed of a generator and a discriminator, and the 16D time domain features of the fault data need to be added as the condition of CGAN, which is the CGAN algorithm in this paper.

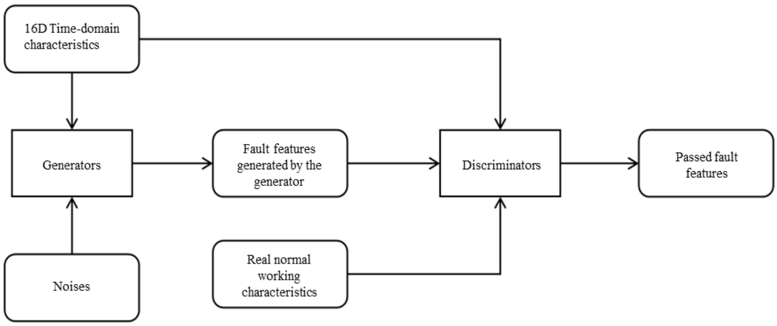


Fig. 4. CGAN architecture diagram

As shown in Figure 4 above. Generative Network G: is a deep feedforward neural network consisting of four fully connected layers and activation functions. In addition, the last layer uses the Tanh function to limit the output between -1 and 1. The function of generating network G is to receive the input noise data, and use the forward propagation algorithm of the network to generate data samples, and the data structure is a two-dimensional matrix structure. The purpose of the training to generate the network G is to learn the mapping relationship from the noisy hidden space to the sample data x.

From the loss function, it can be seen that its calculated value is related to the diagnostic result of the discriminator, which is also the inseparable place of the two processes of generating the adversaria.

$$LG = -Ex{\sim}pg[D(x)]^{[21]} \tag{2}$$

LG is the loss function of the generator G, E is the desired operator, $x{\sim}pg$ is the data sampled from the generator's generation distribution pg, and $D(x)$ is the discriminant result of the discriminator of the generated data x, i.e., $D(x)$ represents the probability that the discriminator thinks x is the real data.

The goal of generator G is to maximize the probability of the discriminator D so that $D(x)$ is as close to 1 as possible, i.e., $D(x) \to 1$. Therefore, the loss function of generator G is defined as the opposite of the probability expectation of its generated data x deception discriminator D, i.e., $-Ex{\sim}pg[D(x)]$. By maximizing this loss, it is equivalent to maximizing $D(x)$, i.e., maximizing the probability that the deceived discriminator thinks that the generated data is real data. Therefore, the design of the generator loss function makes full use of the idea of adversariality, and drives the generator to generate more realistic data by adversarial with the discriminator. This is also the core idea of generative adversarial networks. The Generator network architecture is shown in Table 3.

Table 3. Generator network architecture

Network layer	Enter size	Outer size
Input layer	Noise size (64*125) Condition size:(12)	Noise size + condition size
Fully connected layer 1 ReLU activation layer	Noise size + condition	256
Fully connected layer 2 ReLU activation layer	256	512
Fully connected layer 3 ReLU activation layer	512	1024
Fully connected layer 4 TanH activation layer	1024	(64*125)
Output layer		(64*125)

Discriminant network D: is a deep feedforward neural network consisting of four fully connected layers and activation functions. In addition, Dropout regularization is used to mitigate overfitting. The final layer uses the Sigmoid function to output a probability score between 0 and 1, which is used to determine the authenticity of the input data. The function of discriminant network D is to receive real data samples and generate data samples, and finally give the binary probability value of whether the data belongs to real data by using the forward propagation algorithm of the network, and take the label with higher probability as the final judgment value. The discriminant network D and the generative network G are trained in the state of mutual confrontation, and the loss function of the discriminant network G is expressed as:

$$L = Ex{\sim}Pg[D(x)] - Ex{\sim}\Pr[D(x)] + \lambda Ex{\sim}P \wedge \omega[(\|\nabla\text{^xD(^x)}\|2 - 1)^{[16]} \quad (3)$$

The formula for the loss function of this discriminator network D is explained in three parts as follows:

(1) $Ex{\sim}Pg[D(x)]$: where $x{\sim}Pg$ represents sampling from the false sample distribution Pg generated by generator G, and $D(x)$ represents the probability that the discriminator judges x as a real sample. This part represents the probability that the discriminator expects the generator to judge the fake sample generated by the generator as the real sample, and the discriminator should minimize this part, that is, reduce the probability of judging the fake sample as the real sample.

(2) $Ex{\sim}Pr[D(x)]$: where $x{\sim}Pr$ represents sampling from the real sample distribution pr, and $D(x)$ represents the probability that the discriminator determines that x is the real sample. This part of the term indicates the probability that the discriminator is expected to judge the real sample as the real sample, and the discriminator should maximize this part, that is, increase the probability of judging the real sample as the real sample.

(3) $\lambda E\ x{\sim}P\text{^}\omega[(\|\nabla\text{^}xD(\text{^}x)\|2 - 1)$: $\nabla\text{^}xD(\text{^}x)$ represents the gradient of the sampling input x, this term indicates the degree of deviation of the discriminator gradient from 1, and the discriminator should minimize this term so that the gradient is as close to 1 as possible to avoid the gradient disappearing or exploding.

In summary, the goal of the discriminator is to correctly distinguish between true and false samples while maintaining a stable gradient. By confronting the generator, the quality of the generated samples can be improved, bringing them closer and closer to the real sample distribution. The generative network and the discriminant network are improved together in adversarial learning, and finally the generative network G can generate sample data with probability distributions very close to the real data distribution. Discriminator network architecture is shown in Table 4.

Table 4. Discriminator network architecture

Network layer	Enter size	Outer size
Input layer	Noise size (64*125) Condition size:(12)	Noise size + condition size
Fully connected layer 1	Noise size + condition	1024
ReLU activation layer		
Dropout layer		
Fully connected layer 2	1024	512
ReLU activation layer		
Dropout layer		
Fully connected layer 3	512	256
ReLU activation layer		
Dropout layer		
Fully connected layer 4	256	1
Sigmoid activation layer		
Output layer		1(Probability Value)

The purpose of generating a network is to generate high-quality data that is close to the real data distribution, which is used to confuse the discriminant network [22]. The discriminant network is continuously optimized to accurately distinguish between generated data and real data. In such an adversarial training process, the two networks optimize together, and finally the generative network can generate samples that are very close to the real data.

3.4 Troubleshooter

The troubleshooter is an MLP (Multilayer Perceptron) and the loss function it uses is a binary cross-entropy function, which is expressed as:

$$Loss(x, class) = -\log(\sigma(x)[\text{class}]) \tag{4}$$ [23]

where x is the predicted probability of the model output, $class$ is the true class of the sample (0 or 1), and $\sigma(x)$ is the sigmoid function to compress the probability between 0 and 1. For positive examples, we expect the probability to be close to 1, and negative samples to be close to 0, and the cross-entropy loss function measures the difference between this expectation and the output probability of the model. Troubleshooter network architecture is shown in Table 5.

Table 5. Troubleshooter network architecture

Network layer	Enter size	Outer size
Fully connected layer 1	64*125=8000	512
ReLU activation layer		
Fully connected layer 2	512	128
ReLU activation layer		
Fully connected layer 3	128	2
Sigmoid activation layer		

4 Experiments and evaluations

4.1 Dataset source

The experimental data in this paper comes from the 600-ton gantry crane operation data acquisition system on the shore side of a shipyard in Shanghai, and the sensors of the system are respectively arranged on the gantry crane on the three lifting mechanisms of the pinion bearing seat near the reducer, the pinion bearing seat near the drum, the pinion bearing seat, the large gear end, the reel distal bearing seat, the reducer low-speed shaft end 1, the high-speed shaft end 1 of the reducer near the motor, and two sensors of horizontal and vertical vibration acceleration are respectively arranged at each place. The dataset used in this paper is the horizontal and vertical vibration acceleration data of the near reducer of the pinion bearing housing of the three lifting mechanisms.

The data acquisition system has a collection frequency of 1000 Hz, and the collected data is stored in the MySQL database, and the collected dataset information is shown in Table 6 and Figure 5.

	A	B	C
9934	2022-11-30 14:19:47.933	-2.877	-3.504
9935	2022-11-30 14:19:47.934	-4.821	-4.417
9936	2022-11-30 14:19:47.935	-3.436	-3.349
9937	2022-11-30 14:19:47.936	0.330	-1.951
9938	2022-11-30 14:19:47.937	-0.528	-2.450
9939	2022-11-30 14:19:47.938	-2.377	-2.284
9940	2022-11-30 14:19:47.939	0.656	-1.218
9941	2022-11-30 14:19:47.940	0.560	-0.640
9942	2022-11-30 14:19:47.941	-0.128	-1.676
9943	2022-11-30 14:19:47.942	1.558	-1.541
9944	2022-11-30 14:19:47.943	0.128	-1.284
9945	2022-11-30 14:19:47.944	-0.877	-1.431
9946	2022-11-30 14:19:47.945	-0.916	-0.883
9947	2022-11-30 14:19:47.946	0.277	-1.302
9948	2022-11-30 14:19:47.947	-1.892	-0.935
9949	2022-11-30 14:19:47.948	-3.589	-1.601
9950	2022-11-30 14:19:47.949	-2.455	-3.285
9951	2022-11-30 14:19:47.950	-4.768	-3.192
9952	2022-11-30 14:19:47.951	-3.867	-2.066
9953	2022-11-30 14:19:47.952	-2.910	-2.447
9954	2022-11-30 14:19:47.953	-4.792	-3.936
9955	2022-11-30 14:19:47.954	-3.510	-2.879
9956	2022-11-30 14:19:47.955	-3.043	-2.606
9957	2022-11-30 14:19:47.956	-2.506	-2.760
9958	2022-11-30 14:19:47.957	-0.864	-2.702
9959	2022-11-30 14:19:47.958	-1.693	-2.614
9960	2022-11-30 14:19:47.959	-0.963	0.057
9961	2022-11-30 14:19:47.960	-0.093	-0.881
9962	2022-11-30 14:19:47.961	0.967	-2.321
9963	2022-11-30 14:19:47.962	1.017	-1.540
9964	2022-11-30 14:19:47.963	-0.778	-1.421
9965	2022-11-30 14:19:47.964	0.206	-0.172
9966	2022-11-30 14:19:47.965	-0.863	-0.752
9967	2022-11-30 14:19:47.966	-1.235	-1.138
9968	2022-11-30 14:19:47.967	-0.362	-1.346
9969	2022-11-30 14:19:47.968	-2.073	-2.007
9970	2022-11-30 14:19:47.969	-2.810	-1.627
9971	2022-11-30 14:19:47.970	-3.607	-2.552
9972	2022-11-30 14:19:47.971	-4.011	-2.046
9973	2022-11-30 14:19:47.972	-3.772	-2.837
9974	2022-11-30 14:19:47.973	-4.607	-4.232
9975	2022-11-30 14:19:47.974	-3.421	-2.850
9976	2022-11-30 14:19:47.975	-1.994	-3.667
9977	2022-11-30 14:19:47.976	-2.486	-3.350
9978	2022-11-30 14:19:47.977	-1.732	-1.917
9979	2022-11-30 14:19:47.978	-2.800	-2.888
9980	2022-11-30 14:19:47.979	-1.369	-2.544
9981	2022-11-30 14:19:47.980	1.549	-0.563
9982	2022-11-30 14:19:47.981	-0.020	-0.454
9983	2022-11-30 14:19:47.982	-0.699	-3.138
9984	2022-11-30 14:19:47.983	-0.286	-2.171
9985	2022-11-30 14:19:47.984	0.678	-0.447
9986	2022-11-30 14:19:47.985	0.539	-1.127
9987	2022-11-30 14:19:47.986	-0.705	-0.063
9988	2022-11-30 14:19:47.987	-2.691	-1.205
9989	2022-11-30 14:19:47.988	-3.231	-2.788
9990	2022-11-30 14:19:47.989	-1.147	-2.725
9991	2022-11-30 14:19:47.990	-1.073	-1.503
9992	2022-11-30 14:19:47.991	-5.064	-2.531
9993	2022-11-30 14:19:47.992	-7.365	-3.995
9994	2022-11-30 14:19:47.993	-3.541	-2.788

2022-11-30 1	-1.40548	-1.40429
2022-11-30 1	-0.27299	-2.09808
2022-11-30 1	-1.62959	-1.99795
2022-11-30 1	-0.24319	0.18954
2022-11-30 1	0.43988	-1.19328
2022-11-30 1	-0.16332	-1.42694
2022-11-30 1	-0.21219	-0.71645
2022-11-30 1	0.00477	-0.38266
2022-11-30 1	-0.16212	-0.6783
2022-11-30 1	-1.91092	-0.71287
2022-11-30 1	-2.20656	-0.6485
2022-11-30 1	-3.32832	-1.77383
2022-11-30 1	-3.41654	-1.81079
2022-11-30 1	-2.7144	-3.00407
2022-11-30 1	-3.97921	-1.72496
2022-11-30 1	-5.03659	-2.03371
2022-11-30 1	-5.19514	-3.3617
2022-11-30 1	-3.04341	-2.85506
2022-11-30 1	-2.48671	-2.34723
2022-11-30 1	-2.82526	-1.64509
2022-11-30 1	-1.82152	-3.52025
2022-11-30 1	-3.02911	-2.352
2022-11-30 1	-0.30994	-1.50204
2022-11-30 1	2.91705	-0.15736
2022-11-30 1	-0.88811	-0.06318
2022-11-30 1	-2.10524	-3.21746
2022-11-30 1	0.92864	-1.75238
2022-11-30 1	1.43409	-0.29683
2022-11-30 1	-0.7987	0.12517
2022-11-30 1	-1.64747	-1.50681
2022-11-30 1	-1.30296	-2.1565
2022-11-30 1	-2.89202	-0.82612
2022-11-30 1	-2.13742	-0.82374
2022-11-30 1	-3.24011	-1.63436
2022-11-30 1	-5.85794	-3.75628
2022-11-30 1	-3.81231	-2.35677
2022-11-30 1	-5.13077	-2.36154
2022-11-30 1	-3.91603	-3.33309
2022-11-30 1	-1.40429	-1.74046
2022-11-30 1	-5.06759	-2.83003
2022-11-30 1	-3.68118	-3.36289
2022-11-30 1	-0.15497	-3.01838
2022-11-30 1	-0.88692	-1.62125
2022-11-30 1	0.09656	0.55075

Fig. 5. Partial bearing vibration acceleration dataset

Table 6. Dataset information

Dataset category	Dataset size
Normal datasets	3*800000*2 (Horizontal and vertical vibration acceleration of 800 seconds of normal operation of three different lifting mechanisms)
Failure dataset	600000*2+100000*2 (Horizontal and vertical vibration acceleration for 600 seconds under the fault state of the No. 1 hoisting mechanism and 1000 seconds of horizontal and vertical vibration acceleration under the fault state of the No. 3 hoisting mechanism.)

4.2 Model training

Training environment.

The algorithm is programmed using Python 3.9, and the development environment uses Pycharm 2021 version. The deep learning framework uses Pytorch 2.0.1, an open-source Python-based deep learning framework developed by Facebook AI Research Labs and released in 2016. Pytorch provides flexible, easy-to-use deep neural network building blocks with dynamic computational graphs for efficient model customization and debugging. At the same time, Pytorch provides highly optimized GPU support, making model training and deployment very efficient.

The hardware environment is a mobile workstation, 13th Gen Intel (R) Core (TM) i9-13900HX, the GPU is NVIDIA GeForce RTX 4060, the workstation memory is 16 GB, the operating system is Windows 11 x64, and the training algorithm runs on the GPU.

CGAN training.

In the experiment, 500 normal operation vertical vibration acceleration samples and 500 normal operation horizontal vibration acceleration samples collected every 0.5 seconds are regarded as a normal operation unit, and 1600 normal operation units are used to form a training set, which are first put into the feature extractor and converted into 1600 sets of feature tensor data. Since fault data needs to be generated based on condition data, the condition data required in this paper is fault condition data and needs to be extracted from the fault data set. In this paper, the conditions for generators and discriminators consist of 16 commonly used time-domain indexes, extracted from the rolling bearing dataset in the CRWU Bearing Data Center. The time-domain metrics manually extracted from the fault signal are the mean, standard, absolute, square root, kurtosis, maximum, minimum, variance, skewness, peak-to-peak, pulse index, margin index, kurtosis index, waveform index, peak index, and skewness. Table 7 lists the index names and formulas.

Table 7. Time-domain indexes

Index name	Calculation formula
average value	$f_1 = \frac{1}{N}\sum_{n=1}^{N} x(n)$
The absolute value of the average	$f_3 = \frac{1}{N}\sum_{n=1}^{N} \lvert x(n) \rvert$
kurtosis	$f_5 = \frac{1}{N}\sum_{n=1}^{N} (x(n))^{-4}$
minimum	$f_7 = \min \lvert x(n) \rvert$
Skewness	$f_9 = \frac{1}{N}\sum_{n=1}^{N} (x(n))^{3}$
Pulse index	$f_{11} = \frac{f_6}{f_3}$

Kurtosis index	$f_{13}=\frac{f_5}{(\sqrt{f_8})^2}$
Peak Index	$f_{15}=\frac{f_6}{f_2}$
Standard value	$f_2=\sqrt{\frac{1}{N-1}\sum_{n=1}^{N}[x(n)-p_1]^2}$
Square root value	$f_4=(\frac{1}{N}\sum_{n=1}^{N}\sqrt{\lvert \mathrm{X(n)} \rvert})^2$
maximum	$f_6=\max\lvert x(n)\rvert$
variance	$f_8=\frac{1}{N}\sum_{n=1}^{N}(x(n))^2$
Peak-to-peak	$f_{10}=f_6-f_7$
Margin index	$f_{12}=\frac{f_6}{f_4}$
Wave exponent	$f_{14}=\frac{f_2}{f_3}$
Skewness index	$f_{16}=\frac{f_9}{(f_8)^3}$

Put it into CGAN, set the batchsize to 10, train a total of 1000 rounds, generate fault data, label different types of faults, and save the generated data. The training results are shown in Figure 6.

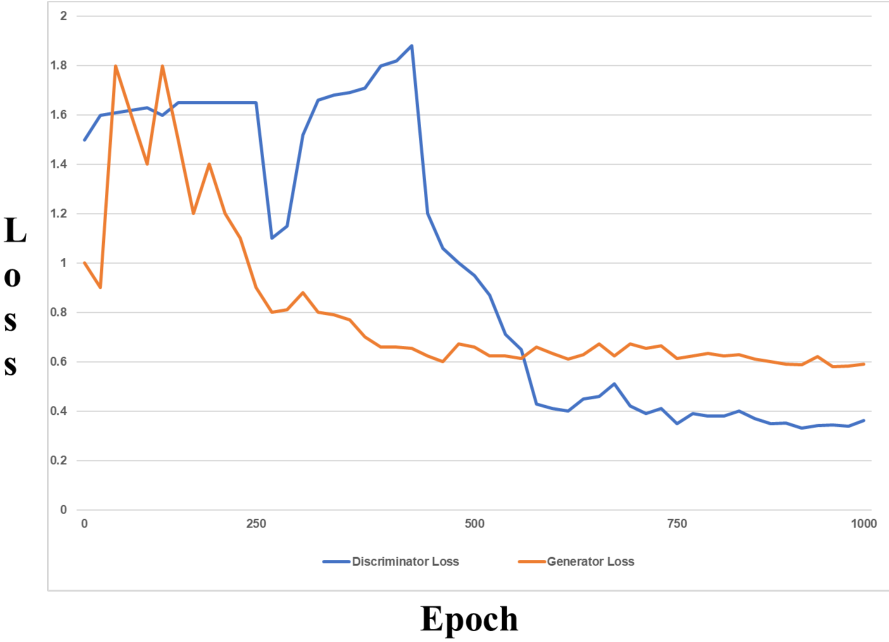


Fig. 6. Generator, discriminator loss value

In order to further demonstrate the quality of the data generated by CGAN, this paper uses RMSE (Root Mean Square Error, which represents the sample standard deviation of the differences between predicted values and observed values, known as residuals. RMSE is used to illustrate the degree of dispersion of the samples. When performing nonlinear fitting, a smaller RMSE is preferable) and MAE (Mean Absolute Error, which indicates the average of the absolute errors between predicted values and observed values) to evaluate the quality of the generated data.

From the perspective of generator and discriminator loss values, the performance of the CGAN model is relatively good: firstly, the generator and discriminator loss values remain stable during the training process, which indicates that the performance of the model is consistent in different training rounds. Secondly, the generator and discriminator loss values are low, which indicates that the model can produce realistic data to deceive the discriminator. Finally, the difference between the generator and discriminator loss values is small, indicating that the performance of the generator and discriminator is relatively balanced. Overall, the CGAN model performed well during training and was able to produce high-quality synthetic data, with Figure 7,8,9 showing some of the failure data generated by the CGAN model.

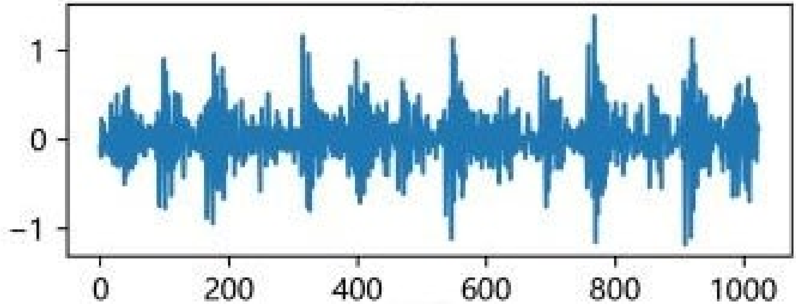


Fig. 7. Inner ring failure data generated

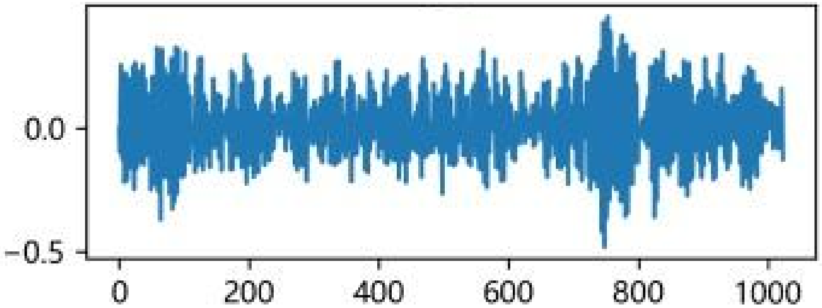


Fig. 8. Rolling element failure data generated

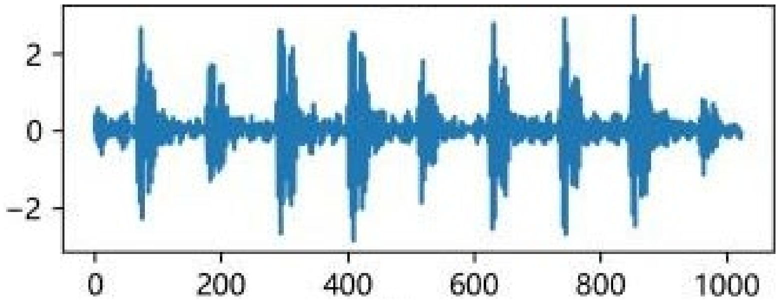


Fig. 9. The resulting outer ring failure data

Troubleshooting training.

After CGAN successfully generates the fault dataset, the feature tensor of the normal operation data is combined with it to form a training set. Put it into the troubleshooter for training, and save the trained neural network. Troubleshooter training loss value is shown in Figure 10.

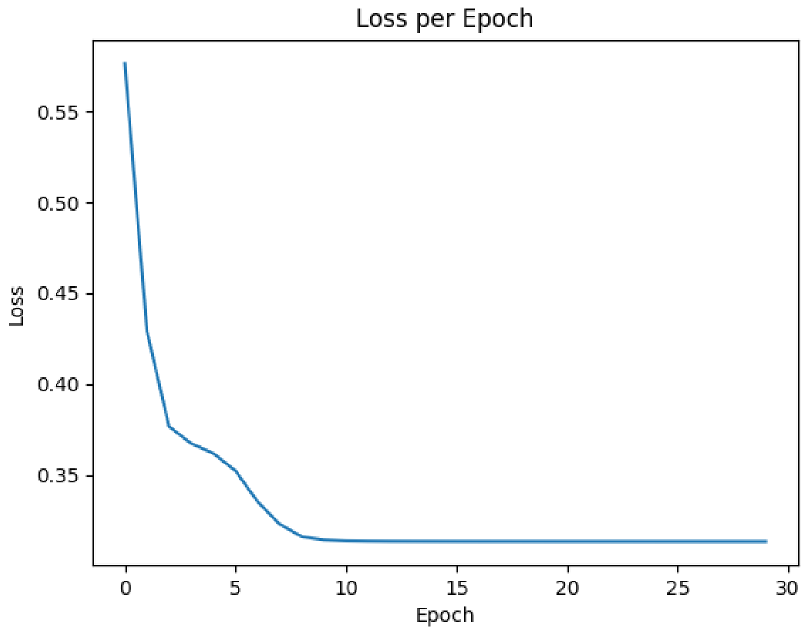


Fig. 10. Troubleshoot the training loss value

Based on the loss values of each generation, we can conclude that the troubleshooter exhibits good convergence, i.e., the loss function gradually decreases during training. In addition, the test loss is also low, indicating that the model performs well on the test data. Therefore, the troubleshooter can be considered to perform well in training.

4.3 Controlled trials

Introduction to Comparison Algorithms.

CNN-BiLSTM-CrossAttention Model

The CNN-BiLSTM-CrossAttention model extracts the global features of the frequency domain features of the one-dimensional fault signal through FFT transformation and the time domain features of the signal itself through CNN convolutional pooling operation, and then extracts the time series features through BiLSTM, and uses the cross-attention mechanism to fuse the features of the time domain and frequency domain, and calculates the attention weights to make the model pay more attention to the important features and then carry out feature enhancement fusion. Finally, the classification results are output through the fully connected layer and softmax.

One-dimensional fault signal recognition model based on TCN-CNN parallelism

The input data dimension of the one-dimensional fault signal recognition model based on TCN-CNN parallel is [32, 1, 1024], and its principle is to first send the data into the CNN network for 1d convolutional pooling, and at the same time, the signal is sent to the TCN layer to extract the time series features, and finally the spatial features and time series features are fused, and finally the fully connected layer and softmax are sent to the fully connected layer and softmax for classification.

CNN Model

Convolutional Neural Networks (CNN) are a type of deep learning model, with the core structure of CNN including convolutional layers, pooling layers, and fully connected layers. The convolutional layer performs local feature extraction on the input signal through filters (convolution kernels), capturing key patterns within the signal; the pooling layer reduces data dimensions through down-sampling, enhancing the robustness of the model; the fully connected layer maps the extracted features to fault classification results. In bearing fault diagnosis, CNN can effectively identify the health status and fault types of bearings by extracting features from vibration or acoustic signals.

Algorithm comparison

Table 8 shows the number of training samples used by each model.

Table 8. The number of training samples for each model

Diagnostic methods	Normal sample size	Number of fault samples
CNN-BiLSTM-CrossAttention	100	400
One-dimensional fault signal identification model based on TCN-CNN parallelism	100	400
Method of this article	100	0
CNN Model	100	400

As shown in Figure 11, the average diagnostic accuracy of the CNN-BiLSTM-CrossAttention model is 99.7%, and the diagnostic training duration is 754 seconds. The diagnostic success rate of the proposed method is 100%, and the diagnostic training duration is 425 seconds. The diagnosis success rate of the one-dimensional fault signal recognition model based on TCN-CNN parallel is 97.9%, and the diagnostic training time is 863 seconds. the average diagnostic accuracy of the CNN Model is 48%, and the diagnostic training time is 1057 seconds.

In terms of the accuracy of fault diagnosis of different methods, the diagnostic accuracy of the three methods is very high, and the diagnostic accuracy of the method proposed in this paper is the highest 100%. Secondly, the method proposed in this paper requires the shortest training time, which is 425 seconds. At the same time, considering that the method proposed in this paper can be diagnosed without fault data, it is concluded that the method proposed in this paper has better performance than the other three methods and is the most suitable for the application of industrial scenarios.

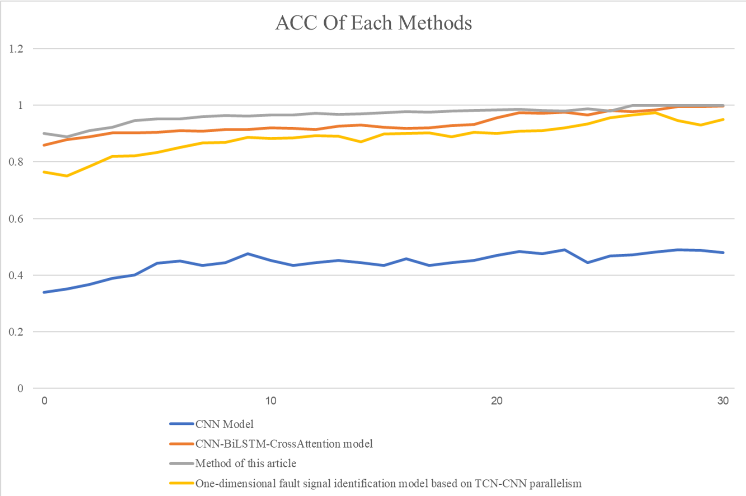


Fig. 11. ACC Of Each Methods

4.4 Results discussed

Based on the conditional generative adversarial network, this paper proposes a new feature generation network to try to solve the problem of trouble-free data. The advantages of the proposed method can be summarized as follows:

First, due to the high complexity of the vibration signal and the instability of the CGAN model, training such a model for fault diagnosis using raw vibration data is time-consuming, difficult, and ineffective. To this end, an additional feature extractor is constructed to reduce the dimension and learn representative features, so the noise in the original data is eliminated, thereby improving the success rate of fault diagnosis.

Secondly, in order to solve the practical problem of trouble-free data in gantry crane operation in industrial production, this paper tries to collect the time-domain features of the fault dataset in the open-source data as a condition to generate simulated fault samples, and selects the most appropriate time-domain feature number after comparison, so as to efficiently generate the fault feature tensor and solve the problem of fault-free data.

Despite the advantages, there are some disadvantages. Firstly, the proposed method requires the time-domain features of the fault data as the condition, and the acquisition of these time-domain features needs to rely on public datasets, and it is difficult to obtain time-domain features if there is a lack of public datasets in a certain field. In addition, debugging critical parameters that have a significant impact on network performance is time-consuming and difficult.

5 Conclusion

In recent years, the rapid development of deep learning technology has led to its widespread application in the field of fault diagnosis for gantry cranes. However, traditional deep learning fault diagnosis methods rely on a large amount of labeled fault data, which is often scarce or difficult to obtain in actual industrial scenarios, limiting their application. This study proposes an intelligent algorithm based on Conditional Generative Adversarial Networks (CGAN) to generate operational fault data for gantry cranes, addressing the issues of insufficient training data and dataset imbalance. The main contributions and industrial application value of this study are summarized as follows:

(1) Improve the accuracy and efficiency of equipment fault detection

In view of the large amount of noise in the vibration acceleration data of gantry cranes, this study proposes a feature extraction method based on Deep Convolutional Neural Network (1DCNN), which converts the original signal into feature tensors, so as to reduce the impact of noise and improve the accuracy of fault diagnosis. This improvement can effectively reduce the false positive rate and reduce the risk of false or missed detections of equipment.

(2) Enhance the applicability of data-driven approaches to industrial applications

In this study, the supervised training characteristics of CGAN were used to extract the time-domain features and generate realistic fault data by combining the open-source bearing fault dataset. Experiments show that the proposed method can achieve high-precision fault diagnosis without fault data. This is critical for real-world industrial scenarios, especially in the absence of historical failure data, and provides an effective solution for data-driven equipment health management.

(3) Promote the integration and application of intelligent O&M systems

This method can be integrated into intelligent O&M systems for real-time health monitoring and predictive maintenance of gantry cranes. Combined with existing Industrial Internet of Things (IIoT) platforms, the generated fault data can be used to train and optimize online monitoring systems, improving fault identification capabilities. In addition, this method can also be combined with other AI-based O&M platforms (such as CMMS or intelligent diagnostic software) to achieve automatic fault detection and early warning, reduce manual intervention, and improve equipment maintenance efficiency.

(4) Reduce maintenance costs and increase equipment reliability

Traditional maintenance methods rely on regular inspections and empirical judgments, which have high maintenance costs and downtime risks. The fault diagnosis method proposed in this study can effectively reduce unnecessary maintenance man-hours, optimize maintenance plans by predicting possible failures in advance, thereby reducing unplanned downtime and improving the reliability and service life of equipment.

Although this method achieves high accuracy in fault diagnosis without fault data, there are still some limitations. For example, the method relies on open-source datasets to extract time-domain features, and there may be a lack of applicable public data in specific industrial environments. In addition, the adjustment of key parameters of

CGAN is still complex and needs to be further optimized. Future research may consider extending to different types of crane equipment and combining with other generative models such as diffusion models or variational autoencoders to further improve the quality of generated data and the adaptability of the models. In conclusion, this study provides an innovative solution for the intelligent fault diagnosis of gantry cranes, which can achieve efficient and accurate fault detection and predictive maintenance in the case of unbalanced or even missing data, which is of great practical significance for intelligent manufacturing and industrial equipment management.

References

1. P. Liang, Z. Yu, B. Wang, X. Xu, J. Tian, Fault transfer diagnosis of rolling bearings across multiple working conditions via subdomain adaptation and improved vision transformer network, Adv. Eng. Inf. 57 (2023) 102075.
2. J. Lu, R. Cui, S. Li, An imbalanced sample intelligent fault diagnosis method using data enhancement and improved broad learning system, Meas. Sci. Technol. 34 (7) (2023) 075106.
3. J. Li, R. Huang, Z. Chen, G. He, K.C. Gryllias, W. Li, Deep continual transfer learning with dynamic weight aggregation for fault diagnosis of industrial streaming data under varying working conditions, Adv. Eng. Inf. 55 (2023) 101883.
4. J. Lu, B. Jia, S. Li, S. Gong, A noise reduction method of rolling bearing based on empirical wavelet transform and adaptive time frequency peak filtering, Meas. Sci. Technol. 34 (2023) 125146.
5. Y. Jin, G. Xin, J. Antoni, Towards automated, integrated and unsupervised diagnosis of rolling element bearings, Mech. Syst. Sig. Process. 203 (2023) 110691.
6. H. Wei, D. Guan. Research on friction and wear of piston rings of marine diesel engine based on oil monitoring [J]. Journal of Dalian Maritime University,2002(04):55-56+63. DOI:10.16411/j.cnki.issn1006-7736.2002.04.016.
7. S. LI, L. WU, Y. DAI. Fault diagnosis of harbor crane bearing under unstable condition based on ADR-SDP-DCNN algorithm [J]. Journal of Shanghai Maritime University, 2022, 43(03):102-110. DOI:10.13340/j.jsmu.2022.03.016.
8. Shen Keyu, Yan Hua. Fault diagnosis of port crane reduction box based on fusion feature support vector machine [J]. China Water Transport (2nd half).),2020,20(05):100-103+182.
9. J. Zhang, X. Kong, X. Li, et al., Fault diagnosis of bearings based on deep separable convolutional neural network and spatial dropout, Chin. J. Aeronaut. 35 (10) (2022) 301–312.
10. Surani Matharaarachchi, Mike Domaratzki, Saman Muthukumarana, Enhancing SMOTE for imbalanced data with abnormal minority instances, Machine Learning with Applications, Volume18, 2024, 100597, ISSN26668270,https://doi.org/10.1016/j.mlwa.2024.100597.
11. L. HU, X. WANG, Z. LIN, et al. Research on internal leakage fault diagnosis of pipeline valve based on DCGAN-1DCNN[J/OL]. Industrial safety and environmental protection:1-9[2023-10-24].http://kns.cnki.net/kcms/detail/42.1640.X.20230529.1725.022.html.
12. Y. Liu, H. Cai, W. Li, et al. DCGAN Diagnostic Method for Bearing Failure under Small Sample Condition [J]. Vibration. Testing & Diagnostics,2023,43(04):817-823+836. DOI:10.16450/j.cnki.issn.1004-6801.2023.04.027.
13. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

14. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).
15. Odena, A., Olah, C., & Shlens, J. (2017). Conditional image synthesis with auxiliary classifier gans. In International conference on machine learning (pp. 2642-2651).
16. Miyato, T., & Koyama, M. (2018). cgan with projection discriminator. arXiv preprint arXiv:1802.05637.
17. Fukushima, Kunihiko, and Sei Miyake. "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition." Competition and cooperation in neural nets. Springer, Berlin, Heidelberg, 1982. 267-285.
18. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012): 1097-1105.
19. Dumoulin, Vincent, and Francesco Visin. "A guide to convolution arithmetic for deep learning." arXiv preprint arXiv:1603.07285 (2016).
20. Lecun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.
21. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
22. He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
23. Yosinski, Jason, et al. "How transferable are features in deep neural networks?." Advances in neural information processing systems 27 (2014).