Data Augmentation for Cardiac Magnetic Resonance Image Using Evolutionary GAN

Ying Fu[1,2(✉)], Minxue Gong[1], Guang Yang[1], and Jiliu Zhou[1,2]

[1] School of Computer Science, Chengdu University of Information and Technology, Chengdu 610225, China
fuying@cuit.edu.cn

[2] Image and Spatial Information 2011 Collaborative Innovation Center of Sichuan Province, Chengdu 610225, China

Abstract. Generative adversarial networks (GAN) could synthesize semantically meaningful data from standard signal distribution, which make it have considerable potential to alleviate data scarcity. In this paper, based on Evolutionary GAN, cardiac magnetic resonance images enhancement method is proposed to solve over-fitting problem caused by training convolution network with small dataset. The most optimal generator which consider the quality and diversity of generated images simultaneously from many generator mutations is chosen. Meanwhile, to expand the whole training set distribution, we combine the linear interpolation of eigenvectors to synthesize new training samples and synthesize related linear interpolation labels, which can make the discrete sample space become continuous to improve the smoothness between domains. In this paper, the effectiveness of this method is verified by classification experiments, and the influence of the proportion of synthesized samples on the classification results of cardiac magnetic resonance images is explored.

Keywords: Evolutionary GAN · Cardiac magnetic resonance · Data augmentation · Linear interpolation

1 Introduction

Cardiac magnetic resonance imaging (MRI) is known as the gold standard for assessing cardiac function. Conventional cardiac MRI scanning technology has been relatively mature and has played a vital role in disease diagnosis. At present, many cardiac magnetic resonance image-assisted diagnosis tasks based on deep learning [1] have achieved good results, but cardiac magnetic resonance images not only require expensive medical equipment to obtain, but also require experienced radiologists to carry out a large number of manual data annotation, which is undoubtedly extremely time-consuming and labor-consuming. In addition, the privacy of patients in the field of medical images has always been very sensitive, so it costs a lot to obtain a large number of data sets that are balanced between positive and negative samples.

X. Wu et al. (Eds.): QShine 2020, LNICST 381, pp. 126–141, 2021.
https://doi.org/10.1007/978-3-030-77569-8_10

A great challenge in the field of medical imaging based on deep learning is how to deal with small-scale data sets and limited number of labeled data. Especially when using complex deep learning model, the data set is not sufficient or the data set sample is unbalanced, which will make the deep convolution neural network with huge parameters appear over fitting [2]. In the field of computer vision, scholars have proposed many effective methods for over fitting, such as batch regularization [3], dropout [4], early stopping method [5], weight sharing [6], weight attenuation [7], etc. The above method is to adjust the network structure. In addition, data enhancement [8] is an effective method to operate on the data itself, which alleviates the phenomenon of over fitting in image analysis and classification to a certain extent. The classical data enhancement techniques mainly include affine transformation methods such as translation, rotation, scaling, flipping and shearing [9, 10], and the original samples and new samples are mixed as training sets and input into convolutional neural network. Adjusting the color space of samples is also a data enhancement method. Wang et al. [11] used the method of changing the brightness value to expand the sample size. Although these methods have improved, only the operation on the original samples does not produce new features. The diversity of the original samples has not been substantially improved [12], and the promotion effect is weak when processing small-scale data.

Generative Adversarial Network (GAN) [13] is a generative model proposed by Ian Goodfellow and others. It consists of a generator G and a discriminator D. The generator G uses noise z sampled from uniform distribution or normal distribution as input to synthesize image G(z). The discriminator D attempts to judge the synthetic image G(z) as false as much as possible, and judges the real image x as true, and adjusts the parameters of each model through successive confrontation training. Finally, the generator obtains the distribution model of real samples and obtains the generation performance close to the real image. The specific structure of GAN is shown in Fig. 1.

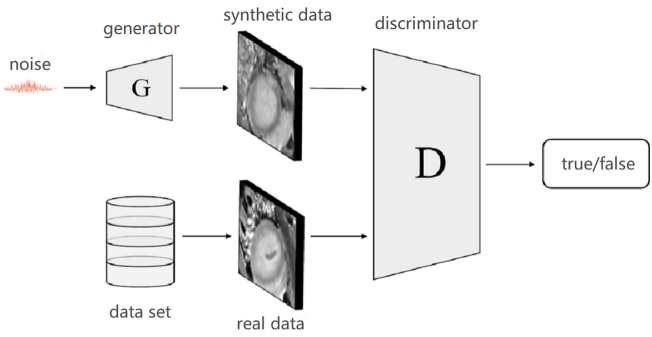

Fig. 1. The structure of GAN

The entire training process of GAN is to find the balance between the generating network and the discriminating network, which makes the discriminator unable to judge whether the samples generated by the generator are real or generated, so that the generating network can achieve the optimal performance. This process can be expressed as formula (1):

$$min_G max_D E_{xP_{data}}\left[logD(x)\right] + E_{zP_z}\left[log(1 - D(G(z)))\right] \tag{1}$$

The generative adversarial network generates new samples by fitting the original sample distribution. The new samples are generated from the distribution learned by the generative model, which makes it have new features that are different from the original samples. This feature makes it possible to use the samples generated by the generating network as new training samples to achieve data expansion. Although GAN has achieved good results in many computer vision fields, it has many problems in practical applications. On the one hand, GAN is very difficult to train. Once the data distribution and the distribution fitted by the generating network do not substantially overlap at the beginning of training, the gradient of the generating network can easily point to a random direction, resulting in the problem of gradient disappearance [14]. On the other hand, in order to make the discriminator give high scores, the generator will try to generate a relatively safe but lack of diversity of single samples, which will lead to the problem of pattern collapse [15].

In order to alleviate the gradient disappearance and model collapse, a large number of GAN variant models have been proposed. The more representative ones are: DCGAN [16] which combines convolutional neural network with GAN and Conditional GAN [17] which adds precondition control generator to the input data. There are also LSGAN [18] and WGAN [19], which have made great improvements to the loss function. Among them, WGAN uses Wasserstein distance to measure the distribution distance, which makes GAN more stable in training to a large extent. However, Ishaan et al. found that WGAN uses a forced phase method to make the parameters of the network mostly focus on $-0.01, 0.01$, which will waste the fitting ability of the convolutional neural network. Therefore, they proposed the WGAN-GP model [20], which effectively alleviated this problem, so it became a more classic model. The Evolutionary GAN [21] proposed by Zhang et al. is a variant model of a generative adversarial network based on evolutionary algorithms. It will perform mutation operations when the discriminator stops training to generate multiple generators as adversarial targets. In different environments (that is, the current discriminator), a specific evaluation method is used to evaluate the quality and diversity of the generated pictures. This series of operations can reserve one or more generators with strong performance for the next round of training. This method of overcoming the limitations of single adversarial target has been proven to be able to keep the best offspring all the time, effectively alleviate the problem of mode collapse and improve the quality of the generator.

Recently, many scholars use GAN to enhance training data samples. The article [22] uses GAN to enhance the data of human faces and handwritten fonts. Ibrahim et al. [23] used the improvement of PGGAN to expand the data set of skin injury and improved the classification accuracy. Maayan et al. [24] used DCGAN and ACGAN to expand the data of liver medical images, and proved that the classification effect of DCGAN in this data set is improved more. Compared with affine transformation, GAN can be used to generate images with new features by learning the real distribution.

Considering that Evolutionary GAN can improve the diversity and quality of generated samples, this paper uses Evolutionary GAN to enhance cardiac magnetic resonance image data. The main contributions of this paper are as follows:

1) A cardiac magnetic resonance medical image data enhancement method based on Evolutionary GAN is proposed, which generates high-quality and diverse samples to expand the training set, and finally improves the various indicators of the classification results;
2) Combining the linear interpolation of feature vectors in Evolutionary GAN to synthesize new training samples and generate related linear interpolation labels, which not only expands the distribution of the entire training set, but also makes the discrete sample space continuous and improves the smoothness between fields, so that the model can be better trained.
3) We use various indicators of downstream classification tasks to optimize the model and experimental details.

2 Evolutionary GAN

The training process of Evolutionary GAN can be divided into three stages: the first stage is mutation, that is, the parent generator is mutated into multiple offspring generators; the second stage is evaluation, that is, the adaptive score of each offspring generator of the current discriminator is calculated through the fitness function; the third stage is selection, that is, the offspring generator with the highest adaptive score is selected by sorting. The basic structure of the Evolutionary GAN is shown in Fig. 2:

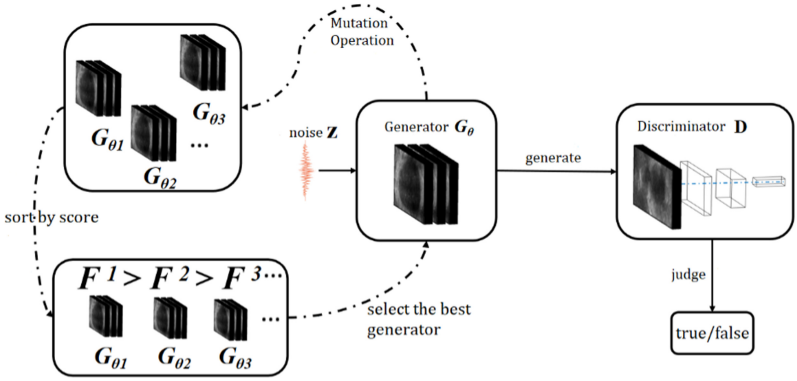

Fig. 2. The structure of Evolutionary GAN

2.1 Mutation

Evolutionary GAN uses different mutation methods to obtain offspring generators based on parent generators. These mutation operators are actually different training targets. The purpose is to reduce the distance between the generated distribution and the real data distribution through different angles. It should be noted that the best discriminator D* in formula (2) should be trained before each mutation operation.

$$D(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)} \tag{2}$$

Zhang et al. proposed three mutation methods:

1) Maximum and minimum value mutation: the mutation has little change to the original objective function, which can provide effective gradient and alleviate the phenomenon of gradient disappearance. It can be written as formula (3):

$$M_G^{minimax} = \frac{1}{2} E_{zP_z}\left[log(1 - D(G(z)))\right] \tag{3}$$

2) Heuristic mutation: heuristic mutation aims to maximize the log probability of the discriminator's error. When the discriminator judges the generated sample as false, the heuristic mutation will not be saturated, and can still provide effective gradient so that the generator can be continuously trained. It can be written as formula (4):

$$M_G^{heuristic} = \frac{1}{2} E_{zP_z} \tag{4}$$

3) Least squares mutation: inspired by ISGAN, least squares mutation can also avoid vanishing gradient. At the same time, compared with heuristic mutation, the least square mutation does not generate false samples at a very high cost, but it does not use very low cost to avoid punishment, which can avoid model collapse to a certain extent. It can be written as formula (5):

$$M_G^{least-square} = E_{zP_z}\left[D(G(z) - 1)^2\right] \tag{5}$$

2.2 Fitness Function

Evolutionary GAN uses the fitness function to evaluate the generator's performance and quantifies it to the corresponding adaptability score, which can be written as formula (6):

$$F = F_q + \gamma F_d \tag{6}$$

F_q is used to measure the quality of the generated samples, that is, whether the offspring generator can fool the discriminator, which can be written as formula (7):

$$F_q = E_z[D(G(z))] \tag{7}$$

F_q measures the diversity of the generated samples. It measures the gradient generated when the parameters of the discriminator are updated again according to the offspring generator. If the samples generated by the offspring generator are relatively concentrated (lack of diversity), it is easier to cause large gradient fluctuations when updating the discriminator parameters, which can be written as formula (8):

$$F_d = -log\left\|\nabla_D - E_x\left[logD(x)\right] - E_z\left[log(1 - D(G(z)))\right]\right\| \tag{8}$$

$\gamma(\geq 0)$ is a hyperparameter used to adjust the quality of samples generated and the weight of diversity, which can be adjusted freely in the experiment.

3 Method

In this paper, we design a data enhancement model of cardiac magnetic resonance medical image based on Evolutionary GAN, which can generate high-quality and diverse samples to expand the training set. The linear interpolation of related labels is generated by combining the linear interpolation of feature vector, which expands the distribution of training set and makes the discrete sample space continuous, so that the model can be trained better. The specific network structure is shown in Fig. 3:

3.1 DAE GAN

Using GAN for data enhancement requires high quality and diversity of samples. Evolutionary GAN can be stably trained and can generate high-quality and diverse samples, so it is very suitable for data enhancement. By adjusting the parameters in the fitness function, you can choose to focus on diversity or quality according to your needs, which can make the data enhancement process more operative. This article improves the Evolutionary GAN and names the improved model *Data Augmentation Evolutionary GAN* (DAE GAN).

There is no difference between the input and output of Evolutionary GAN and Vanilla GAN, except that after the discriminator parameters are fixed, multiple offspring generators are mutated based on the parent generator for training. After the evaluation of the fitness function, the optimal one or more generators are selected as the parent generator in the next discriminator environment.

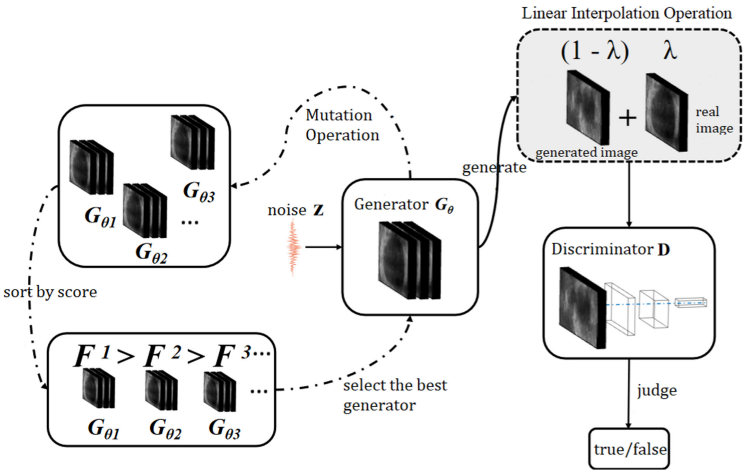

Fig. 3. The proposed network

Although Evolutionary GAN greatly improves the diversity of generated samples, a certain number of training samples are required if you want to fully train the GAN model. In the case of too few training samples, the generator and discriminator are prone to reach the equilibrium point prematurely, which will also cause the phenomenon of model collapse in the generated data. In order to alleviate this problem, this paper uses the traditional affine transformation data enhancement method before training GAN, and expands the data through horizontal flip, vertical inversion, translation, rotation and other operations. Due to the security of medical images, the original data is not added with noise, crop and other operations, and the texture and edge features of the original data are retained as far as possible. But traditional data enhancement only makes small changes to the original data, and does not generate new features, and the samples are also discrete. Thus, this article introduces the linear interpolation.

Zhang et al. proposed a data-independent data enhancement method in the article [25]. This method constructs virtual training samples from original samples, combines linear interpolation of feature vectors to synthesize new training samples and generates related linear interpolations Labels to expand the distribution of the entire training set. The specific formula is as formula (9):

$$\begin{cases} \tilde{x} = \lambda x_i + \left(1 - \lambda x_j\right) \\ \tilde{y} = \lambda y_i + \left(1 - \lambda y_j\right) \end{cases} \tag{9}$$

x_i, x_j is the original input vector, y_i, y_j is the label code, (x_i, x_j), (y_i, y_j) are two samples randomly sampled from the original sample, $\lambda \in Beta[\alpha, \alpha]$ is the weight vector, and $\alpha \in (0, +\infty)$ is the hyperparameter that controls the interpolation strength between the feature and the target vector. The linear interpolation method enables the model to behave linearly when processing the area between the original sample and the sample, so as to reduce the inadaptability of predicting test samples other than the training sample, and enhance the generalization ability. At the same time, the discrete sample space can be continuous and the smoothness between fields can be improved.

When the generator parameters are fixed, the original input of the Evolutionary GAN discriminator is two samples: one is the generated sample, the discriminator tries to minimize the distance between the predicted label of the sample and "0"; the other is the real sample, the discriminator is minimized as much as possible the distance between the predicted label of this sample and "1". The discriminator loss function of the original Evolutionary GAN is as formula (10):

$$L_D = L_{Real} + L_{Fake} \tag{10}$$

The discriminator loss function is expanded into formula (11) as follow:

$$\underset{x,z}{E} L(D(x), 1) + \underset{x,z}{E} L(D(G(z)), 0) \tag{11}$$

This paper uses linear interpolation operation in Evolutionary GAN to modify the discriminator input from the original two pictures to one picture, and the discriminator task is changed to minimize the distance between the predicted label of the fusion sample and "λ". The loss function of discriminator is modified as formula (12):

$$\underset{x,z,\lambda}{E} L(D(\lambda x + (1-\lambda)G(z)), \lambda) \tag{12}$$

3.2 Algorithm

Usually GAN will use the noise z that obeys the multivariate uniform distribution or multivariate normal distribution as the input of the model. Matan et al. [26] believe that multiple Gaussian distributions can better adapt to the inherent multi-modality of the real training data distribution, so a multi-modal distribution is used as input in GAN and it is proved that this method can improve the quality and variety of generated images. The algorithm combined with Gaussian mixture model in this paper is as follows:

Input: **N, K, D, BS, M** and β_1, β_2. The total number of iterations **N**, the number of Gaussian distributions **K**, latent spatial dimension **D**, batch size **BS**, the number of mutation operations **M**, hyper-parameters β_1, β_2.

Output: the updated weight of discriminator and generator parameters.

Step1. Initialize the discriminator parameter ω_0 and generator parameter θ_0

Step2. Gaussian distribution operation

cycle 1 start: for k = 1 : **K**

Step2.1. Sample the initial mean of the Gaussian distribution **K**

Step2.2. Initialize the covariance matrix of Gaussian distribution **K**

cycle 1 end

cycle 2 start: for n = 1 : **N**

Step3. Train discriminator

cycle 3 start: for j = 1 : **BS**

Step3.1. Sample a real image

Step3.2. Sample Gaussian index

Step3.3. Sample noise from the k-th Gaussian distribution

Step3.4. Input the noise into the generator to synthesize a sample

Step3.5. Perform linear interpolation on real samples and synthetic samples, interpolate new samples and labels

Step3.6. Calculate the loss of discriminator

cycle 3 end

Step3.7. Calculate the average loss of discriminator

Step3.8. Update the discriminator parameter

Step4. Train generator

cycle 4 start: for m = 1 : **M**

cycle 5 start: for j = 1 : **BS**

Step4.1. Sample Gaussian index

Step4.2. Sample noise from the k-th Gaussian distribution

Step4.3. Input the noise into the generator to synthesize a sample

Step4.5. Perform a mutation operation on the parent generator and use the offspring generator to generate samples

Step4.6. Calculate the loss of the offspring generator

cycle 5 end

Step4.7. Calculate the average loss of the offspring generator

Step4.8. Update the parameter of the offspring generator

Step4.9. Calculate the adaptive score of the offspring generator

cycle 4 end

Step4.10. Sort the offspring generators in descending order according to the adaptive score

Step4.11. Leave the offspring generator with the highest adaptive score in the current environment and use it as the parent generator for the next iteration

cycle 2 end

4 Results Analysis

4.1 Data Set and Preprocessing

The cardiovascular magnetic resonance data in this experiment comes from a partner hospital. All samples are 2D short-axis primary T1 mapped images. The spatial distance of these cardiac magnetic resonance images ranges from $1.172 \times 1.172 \times 1.0$ mm^3 to $1.406 \times 1.406 \times 1.0$ mm^3, and the original pixel size was $256 \times 218 \times 1$. The benign and malignant labeling and segmentation areas of the image are manually labelled and drawn by senior experts. The original image data is in the ".mha" format. The original image data was in ". MHA" format. After preprocessing, such as resampling, selection of regions of interest, normalization and final selection of interest, a total of 298 images were obtained, including 221 cardiomyopathy images and 77 non-diseased images. The image size after preprocessing was $80 \times 80 \times 1$. The pretreated cardiac magnetic resonance image is shown in Fig. 4.

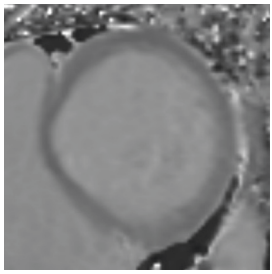

Fig. 4. Cardiac magnetic resonance image

In order to ensure the consistency of training data, all samples are normalized in this experiment. Before training GAN, this experiment performed affine transformation data enhancement on the training set, including: horizontal flip, vertical flip, 0°–20° random amplification and rotation, 90°, 180°, 270° rotation, 0–2% random amplification and translation of vertical and horizontal axes, small and specific amplitude amplification and rotation, and amplification translation, so as to make the data not lose the original image information. After the training set is enhanced once, it is divided into two types of operations: one is to put it into the classifier for training directly, and then use the test set to get the classification results; the other is to put it into different GAN for training, and finally generate new samples to train the classifier again.

4.2 Training DAE GAN

The original evolutionary GAN uses the structure of DCGAN. In this paper, we consider that the residual structure [27] can alleviate the gradient vanishing problem and accelerate the convergence speed of the model, so as to train the high-performance generator more quickly in the same training time. The residual structure as shown in Fig. 5 is used in the generator and discriminator in this article.

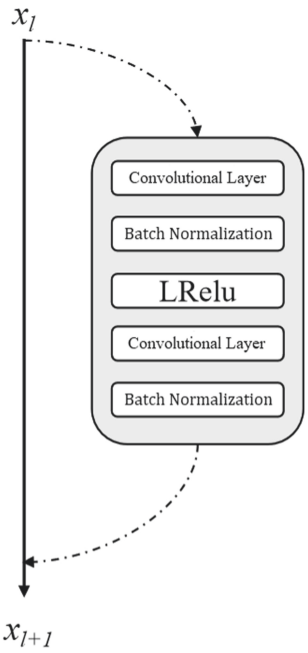

Fig. 5. Residual block structure

Combined with the self-attention module [28], the detailed structure of the generator and discriminator and the output size of each layer are shown in Table 1.

DAE GAN experimental environment: Ubuntu 16.04.1 TLS, Tensorflow 1.14.0, two Nvidia Tesla M40 GPU with 12 GB video memory used to train the generative models of diseased and non-diseased samples respectively. The maximum storage capacity of the model is set to 4, taking into account the space occupation and preventing accidental interruption.

4.3 The Generation Results of DAE GAN

This experiment uses 5-fold cross validation to dynamically divide the heart magnetic resonance image into a training set and a test set at a ratio of 0.8:0.2. Training DAE Gan only uses the training set. Due to the uncertainty of deep convolution model in the training process, each model was trained several times ($\geq$5), and the specific effect of data enhancement method was verified by average classification results.

After normalization and affine transformation, the training set of the cardiac magnetic resonance image data is expanded. We train DAE GAN model following the steps of Algorithm 1. In order to intuitively show the training process of the generative model, Fig. 6 shows the changing process of the samples generated in the training process of the model.

The comparison between the samples generated by the trained model generator and the real samples is shown in Fig. 7.

Table 1. The structure of DAE GAN

Generator	Kernel size
Noise z	—
Fully Connected Mapping	—
Residual Structure	3 × 3
Residual Structure	3 × 3
Self-attention Module	—
Residual Structure	3 × 3
Residual Structure	3 × 3
Convolutional Layer & tanh	3 × 3
Discriminator	Kernel size
Input Image	—
Convolutional Layer	3 × 3
Residual Structure	3 × 3
Self-attention Module	—
Residual Structure	3 × 3
Residual Structure	3 × 3
Residual Structure	3 × 3
Fully Connected Layer	—

4.4 Classification Experiment and Analysis of Experimental Results

Observation method has strong subjectivity. In this experiment, data enhancement is performed on small sample medical images, as a result, the observation method can only be used as a reference evaluation standard. In order to evaluate the effect of data enhancement, this article uses the ResNet50 model and the Xception model [29] as a classifier, the classification results are used to uniformly evaluate the effects of various data enhancement methods.

In addition to the conventional accuracy index, the sensitivity and specificity of two medical image classification indexes are also calculated. These indicators are briefly explained below.

The accuracy rate, that is, the probability that the diseased sample and the non-diseased sample are judged correctly. The calculation method is as formula (13):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

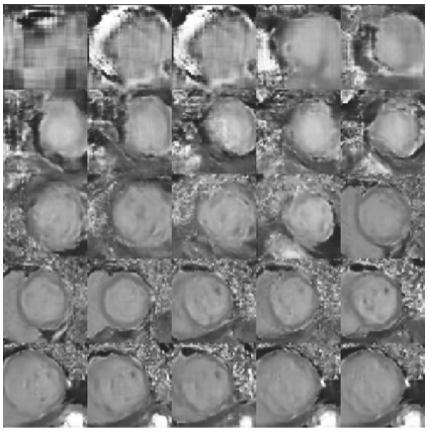

Fig. 6. The changing process of generated sample

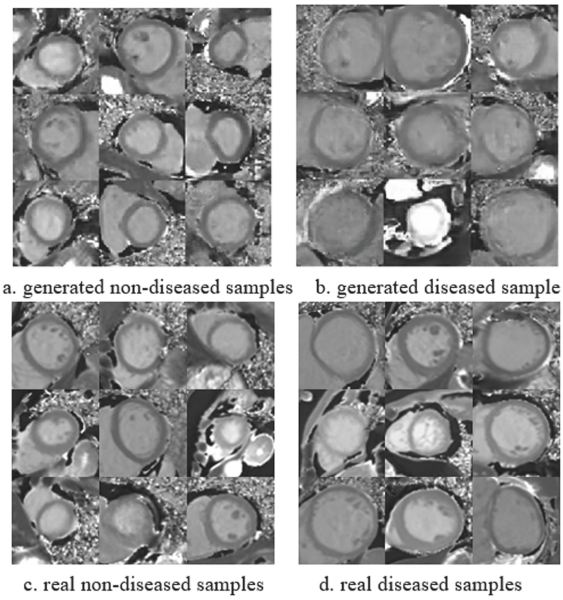

Fig. 7. Comparison of generated samples with real samples

Sensitivity, namely the probability that a diseased sample is judged to be diseased. The calculation method is as formula (14):

$$Sensitivity = \frac{TP}{TP + FN} \tag{14}$$

Specificity, namely the probability of judging a non-diseased sample as non-diseased. The calculation method is as formula (15):

$$Specificity = \frac{TP}{TP + FN} \tag{15}$$

TP represents True Positive, namely the classifier judges it to be a diseased sample, which is in fact also a diseased sample; **TN** represents True Negative, that is, the classifier judges it to be a non-diseased sample, which is in fact not a diseased sample. **FP** is short for False Positive, namely the classifier judges it to be a diseased sample, which is in fact a non-diseased sample; **FN** is short for False Negative, that is, the classifier judges that the sample is not diseased, but is actually a diseased sample.

In this article, the classification experiment uses the Keras framework under the Ubuntu 16.04.1 TLS system environment, the version number is 2.24; the training process uses a Tesla M40. The learning rate is set to 1e−4, and we use the RMSprop optimizer, setting early stopping method to prevent over-fitting, and the 5-fold cross-validation method is used to find the average classification result of the classifier. Table 2 details the average classification results of each enhancement methods in the ResNet50 and Xception classification models.

Table 2. The classification results of enhancement methods

Enhancement method	Accuracy	Sensitivity	Specificity
Classification Network 1: ResNet50			
No Enhancement	0.7767	0.9674	0.6964
Affine Transformation	0.8093	**0.9806**	0.7300
DCGAN	0.8140	0.9760	0.7436
ACGAN	0.7915	0.9728	0.7127
Evolutionary GAN	0.8288	0.9726	0.7591
Our Method	**0.8478**	0.9772	**0.7822**
Classification Network 2: Xception			
No Enhancement	0.7953	0.9765	0.6833
Affine Transformation	0.8279	0.9700	0.7696
DCGAN	0.8326	0.9672	0.7239
ACGAN	0.8054	0.9731	0.7082
Evolutionary GAN	0.8514	0.9780	0.7821
Our Method	**0.8698**	**0.9798**	**0.8116**

Through the experiments, we found that compared with the classification results without any data enhancement method, in ResNet50 model, the classification accuracy increased from 0.7767 to 0.8478, the sensitivity from 0.9674 to 0.9772, the specificity from 0.6964 to 0.7822; in Xception model, the classification accuracy increased from

0.7953 to 0.8698, the sensitivity from 0.9765 to 0.9798, and the specificity from 0.6833 to 0.8116.

5 Conclusion

The DAE GAN model proposed in this paper can effectively expand the amount of cardiac magnetic resonance image data, and effectively alleviate the problem that the classification network cannot be fully trained due to the small amount of medical image data and the uneven data. Compared with any data enhancement method, the classification accuracy of DAE GAN in ResNet50 and Xception models has been improved by 7.11% and 7.45% respectively; compared with affine transformation data enhancement, the method proposed in this paper has been improved by 3.85% and 4.19% respectively, and the experimental results show that the method is effective in different classification models.

Funding Statement. This work was supported in part by the Sichuan Science and Technology Program under Grant 2019ZDZX0005 and the Chinese Scholarship Council under Grant 201908515022.

References

1. Hinton, G.E., Srivastava, N., Krizhevsky, A., et al.: Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. arXiv preprint arXiv:1207.0580 (2012)
2. Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. Neural Comput. **4**(1), 1–58 (1992). https://doi.org/10.1162/neco.1992.4.1.1
3. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv:1502.03167
4. Srivastava, N., Hinton, G., Krizhevsky, A., et al.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
5. Morgan, N., Bourlard, H.: Generalization and parameter estimation in feedforward nets: some experiments. In: [22], pp. 630–637 (1990)
6. Nowlan, S.J., Hinton, G.E.: Simplifying neural networks by soft weight-sharing. Neural Comput. **4**(4), 473–493 (1992)
7. Krogh, A., Hertz, J.A.: A simple weight decay can improve generalization. In: [16], pp. 950–957 (1992)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
9. Roth, H.R., Lu, L., Liu, J., et al.: Improving computer-aided detection using convolutional neural networks and random view aggregation. IEEE Trans. Med. Imaging **35**(5), 1170–1181 (2015)
10. Setio, A.A.A., Ciompi, F., Litjens, G., et al.: Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks. IEEE Trans. Med. Imaging **35**(5), 1160–1169 (2016)
11. Wang, S.: Facial Affect Detection Using Convolutional Neural Networks. Stanford University (2016)

12. Medical Image Synthesis for Data Augmentation and Anonymization using Generative Adversarial Networks
13. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NIPS) (2014)
14. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2017)
15. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2016)
16. Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv preprint arXiv:1511.06434 (2015)
17. Mirza, M., Osindero, S.: Conditional Generative Adversarial Nets. arXiv:1411.1784
18. Mao, X., Li, Q., Xie, H., et al.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2794–2802 (2017)
19. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. arXiv preprint arXiv:1701.07875
20. Gulrajani, I., Ahmed, F., Arjovsky, M., et al.: Improved Training of Wasserstein GANs. In: Advances in Neural Information Processing Systems, pp. 5767–5777 (2017)
21. Wang, C., Xu, C., Yao, X., et al.: Evolutionary generative adversarial networks. IEEE Trans. Evol. Comput. **23**(6), 921–934 (2019)
22. Antoniou, A., Storkey, A., Edwards, H.: Data Augmentation Generative Adversarial Networks. arXiv preprint arXiv:1711.04340 (2017)
23. Alia, I.S., Mohameda, M.F., Mahdya, Y.B.: Data Augmentation for Skin Lesion Using Self-Attention Based Progressive Generative Adversarial Network. arXiv:1910.11960
24. Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing **321**, 321–331 (2018)
25. Zhang, H., Cisse, M., Dauphin, Y.N., et al.: mixup: Beyond Empirical Risk Minimization. arXiv preprint arXiv:1710.09412 (2017)
26. Ben-Yosef, M., Weinshall, D.: Gaussian Mixture Generative Adversarial Networks for Diverse Datasets, and the Unsupervised Clustering of Images. arXiv preprint arXiv:1808.10356 (2018)
27. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
28. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-Attention Generative Adversarial Networks. arXiv:1805.08318
29. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)