

Incomplete Image Completion through GAN

Biyang Deng¹, Desheng Zheng^{1,*}, Zhifeng Liu¹, Yanling Lai¹ and Zhihong Zhang²

¹School of Computer Science, Southwest Petroleum University, Chengdu, 610000, China

²AECC Sichuan Gas Turbine Establishment, Mianyang, 621700, China

*Corresponding Author: Desheng Zheng. Email: zheng_de_sheng@163.com

Received: 03 July 2021; Accepted: 29 August 2021

Abstract: There are two difficult in the existing image restoration methods. One is that the method is difficult to repair the image with a large damaged, the other is the result of image completion is not good and the speed is slow. With the development and application of deep learning, the image repair algorithm based on generative adversarial networks can repair images by simulating the distribution of data. In the process of image completion, the first step is trained the generator to simulate data distribution and generate samples. Then a large number of falsified images are quickly generated using the generative adversarial network and search for the code of the closest damaged image. Finally, the generator generates missing content by using this code. On this basis, this paper combines the semantic loss function and the perceptual loss function. Experimental result show that the method successfully predicts the information of large areas missing in the image, and realizes the photorealism, producing clearer and more consistent results than previous methods.

Keywords: Deep learning; generative adversarial network; convolutional neural network; image completion; image repair

1 Introduction

The damage of digital image will not only affect the visual performance of the image, but also affect the identification of image information. Image completion technology means that uses the existing edge information in the image to fill in the missing area of the image according to certain repair rules [1]. Image completion has high application value in daily life which has been widely used in medicine, military, photography, film industries and so on [2]. There are three traditional image completion technologies, image completion method based on partial differential equation, image completion method based on sample matching and image completion method based on sparse representation. Later, with the development of machine learning and computer vision, there are more image completion methods based on deep learning technology [3].

1. Image completion based on partial differential equation: At the beginning, Bertalmio et al. established the image completion model called BSCB [4]. In recent years, high-order image completion model based on PDE has also been proposed by [5]. The image completion technology based on PDE mainly uses the idea of diffusion in principle, which has obvious defects. For the image with complex structure and texture, the effect of image restoration is not ideal.

2. Image completion based on sample matching: The most commonly used algorithm is Bornard's texture synthesis based algorithm, which uses general missing data to repair [6]. After that, Criminisi proposed a repair algorithm based on the sample block, which is used to remove large area objects in the image [7]. Srivastava et al. proposed an algorithm for constructing a priori, which can help image completion produce better and better results [8].

3. Image completion based on sparse representation: Fidane proposed an idea of using EM [9]



mechanism in Bayesian framework. Shen et al. proposed an image signal sparse representation algorithm based on redundant dictionary [10]. In addition, Mairal et al. use dictionary learning image completion method to repair color images for the first time, but its effect on repairing images with large damaged area is insufficient [11].

4. Image completion based on deep learning: Yang et al. used the pixel recurrent neural network to repair the image [12]. Pathak et al. first proposed the use of generative adversarial network for image completion in 2016 [13]. On the basis of generating countermeasure network, Li et al. [14] proposed a method to repair image by using deep convolution generative adversarial network (DCGAN). Li et al. also proposed a network architecture based on deep learning for face completion in 2016.

In this paper, the image completion algorithm is used to repair the missing pixels [14]. The DCGAN, a variant of the generative adversarial network, is used to build the image completion algorithm model. The CelebA face data set is selected for processing as the training and testing data set for image completion. Then, the model training and image repair are carried out on the GPU server installed with TensorFlow environment, and the final result will be carried out. After the image is compared with the original image and incomplete image, visual analysis and judgment of the experimental model in image completion repair performance. The experimental results show that the method can successfully predict the missing information in large area. It also realizes the realistic image of the image, and produces clearer and more coherent results than the previous methods.

2 Primary Knowledge

2.1 Generative Adversarial Network

The generative adversarial network is composed of two network models, namely generation model G and discrimination model D. The function of generating model G is to learn the probability distribution of real data in training set. The purpose of the generator is to make the model can take random noise as input and output fake pictures close to real pictures. The function of discriminant model is to judge whether an image is a real picture or a picture generated by a generator. The purpose is to distinguish the image generated by the generated model from the real image from the data set. The method of generating confrontation network is to make the generative model and the discriminant model play a game. In the training process, the two models are enhanced simultaneously by competing with each other. Because of the existence of the discriminant model, the generated model can learn the distribution of real samples without a lot of prior knowledge and prior distribution, and finally make the data generated by the generation model achieve the effect of confusing the false with the true.



Figure 1: The structure model of generative adversarial network

As shown in Fig. 1, the framework of generating countermeasure model is based on multi-layer perceptron. Generation network defines a priori on the input noise variable $p_z(Z)$ in order to enable the generation network to learn the data distribution p_g from the real data x . Next, the mapping of the data space is marked as $G(z; \theta_g)$, where G represents a differentiable function that can be solved by multi-layer perceptron, and the parameter is θ_g . At the same time, we also need to define another multi-layer perceptron $D(z; \theta_d)$ whose output value is a constant. $D(x)$ is used to represent the probability that data is coming from real data rather than from generator generated data p_g . By training D repeatedly, the training samples are optimized, and the probability is identified from the samples marked by G . Therefore, the optimization objective function is defined as follows:

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}} [\log D(x|y)] + E_{z \sim p_z(z)} [\log (1 - D(G(z|y)))] \quad (1)$$

When updating the parameters of the discriminant model, for samples from real distribution P_{data} , we hope that the output of $D(x)$ is closer to 1, that is, the larger $\log D(x)$ the better. For the data generated by noise z , we hope $D(G(z))$ to be as close to 0 as possible, so the larger $\log(1-D(G(z)))$ the better, so we need to $\max D$.

When updating the parameters of the generated model, $G(z)$ is expected to be as same as the real data as possible, that is $P_g = P_{data}$. Therefore, we hope $D(G(z))$ to be as close as possible to 1, that is, the smaller the better, so we need to $\min G$.

2.2 Deep Convolutional Generative Adversarial Network

Deep convolutional generative adversarial network (DCGAN) was proposed by Pathak et al. [13] in 2016. DCGAN is combined with convolutional neural network on the basis of generative adversarial network, but it is not simply combined. The purpose of DCGAN is to solve the problems of slow convergence rate and unstable training state of GAN network in the training process. The state of deep convolutional generative adversarial network is stable in the training process, and can effectively realize the application of high-quality image generation and related generation model. In order to make GAN adapt to the architecture of convolutional neural network better, DCGAN proposed four architecture design rules:

1. The convolution layer is used to replace the original pool layer [14]. In the generation network, the upper sampling is replaced by the micro step convolution. In the discrimination network, all the samples are taken down by step convolution, so a full convolution network is formed.

2. Remove the full connection layer. Removing the full connection layer of deeper architecture, such as replacing the full connection layer with the global average pooling layer, can improve the stability of the model better.

3. Use batch normalization. Batch standardization is used to generate and identify models. Batch standardization is a very important means to accelerate convergence and slow down over fitting in deep learning. It can accelerate the speed of learning and convergence. It can convert the input of each layer into 0 mean value and unit standard deviation, and reduce the training difficulty caused by improper initialization. However, batch standardization cannot be directly applied to the whole convolution network because of sample oscillation and instability, so they only use batch standardization for the output layer of the generator and the input layer of the discriminator.

4. Use the appropriate activation function [15]. Because the function of distinguishing network is to make a true false dichotomy, we need to use Softmax function to output the confidence level of (0,1). Therefore, in the generation network, the output layer uses the Tanh activation function, and the other layers use the ReLU activation function. In the discrimination network, Softmax function is used in the output layer, and LeakyReLU activation function is used in other layers. This is the choice of activation function which makes the best performance of DCGAN.

3 Semantic Repair Based on Deep Generation Model

3.1 Principle of Image Completion

In order to fill the image with large missing area, other pixels in the image that remove the missing area are used to obtain the required information. Semantic repair based on deep generation model trains the generator G and discriminator D with undamaged data. After the training, the generator G has a good generation effect, and can use the random noise vector z to simulate the real data distribution to generate images. Under the condition of the loss function, the best image restoration algorithm is generated by using the lost content function. In this way, \hat{z} is input into the trained generation model for image generation, and the result is the image closest to the damaged part of the image, and the repair effect is the best.

Based on the assumption that a \hat{z} has been found, a reasonable reconstruction of the missing pixels $G(\hat{z})$ can be generated by the generator. \hat{z} can be obtained by the following formula:

$$\hat{z} = \operatorname{argmin}_z \{L_c(z|y, M) + L_p(z)\} \quad (2)$$

where y is the image to be repaired and M is the binary mask of the image loss part. L_c is the context loss function, which is used to constrain the region corresponding to $G(z)$ to be as similar as possible to the known pixel region of y . When the situation is not similar, $G(z)$ will be punished to make the generator more likely to generate similar context. The dissimilarity degree of L_c is obtained by $G(z)$ minus the pixels in y . L_p represents the prior loss, which is a perceptual loss function used to constrain whether the generated image is realistic or not.

3.2 Context Loss Function

Context loss function is also called importance weighted context loss function, which defines the importance of the area not missing in the image. When filling the missing area, the remaining pixel data in the image should be used. If l_2 norm is used to calculate the distance between the generated sample $G(z)$ and the non missing part of the input picture y . Under the l_2 specification, the importance of pixels in any region of the image has nothing to do with its position, which is unreasonable. Normally, pixels closer to the missing region should be more important, and pixels farther away from the missing area should play a less important role. In order to achieve this goal, it is assumed that the importance of pixels in context loss is positively related to the surrounding pixels [16]. The importance of each pixel is assumed by importance weighting, which is recorded as W .

$$W_i = \begin{cases} \sum_{j \in N(i)} \frac{(1-M_j)}{|N(i)|}, & \text{if } M_i \neq 0 \\ 0, & \text{if } M_i = 0 \end{cases} \quad (3)$$

where i is the subscript of the pixel, W_i is the importance of the element whose subscript is i , and $N(i)$ is the domain set of pixel i in the local window. $|N(i)|$ refers to the number of $N(i)$. We define context loss as the l_1 norm difference between the recovered image and the undamaged part [17].

$$L_c(z|y,M) = \left\| W \odot (G(z)-y) \right\|_1 \quad (4)$$

The context loss function is used to constrain the undamaged pixel region of the incomplete image y to be as similar as possible to the corresponding region generated by the generator.

3.3 Perceptual Loss Function

The perceptual loss function is also called prior loss [18]. Prior loss refers to the penalty based on advanced image feature representation rather than pixel by pixel difference. Prior loss encourages the restored image to be similar to the sample extracted from the training set, and penalizes the unreal image. The final purpose of discriminator is to distinguish the real image from the generated image through sufficient training [19]. Therefore, the prior loss we choose is the same as the loss function of discriminator D in the training generated countermeasure network:

$$L_p(z) = \lambda \log(1-D(G(z))) \quad (5)$$

λ is a parameter to balance the two losses. The discriminator D is cheated by updating z to make the corresponding generated image more realistic.

3.4 Image Completion

As shown in Fig. 2, the process of image restoration is as follows: Firstly, the generation network G and the discrimination network D in DCGAN network are used to train the undamaged images to learn the distribution characteristics of images. At the beginning of model training, the following variables are defined. P_z is a simple and commonly used distribution. In this algorithm, we use the uniform distribution in the closed interval of -1 to 1 to represent it. A sample taken from the distribution is denoted as $z \sim P_z$. $G(z)$ is used to represent the function sampled from the original probability distribution. The expression of $G(z)$ can be obtained by micro step convolution. Its input is a vector and its output is a 64×64 picture. P_{data} represents the probability distribution of real data. P_g represents the sampling of $G(z)$ from a

probability distribution, that is, the data distribution learned by the network after training. In the ideal state, $P_{data} = P_g$ after network training. At this time, $G(z)$ can generate sample data that obeys the real data distribution. After defining the variables, the DCGAN is trained by combining with the generation countermeasure network, so that the generated network can learn the distribution characteristics of the real data and generate pseudo images.

After training, DCGAN was used for image completion. Firstly, z is initialized randomly. Assuming that one z^* can be found, a $G(z^*)$ can be generated to reconstruct the missing values in the image. Finally, through the same steps of training DCGAN, we use the loss function of formula (2) to carry out the back propagation s update to find z^* , and finally obtain the repaired image.

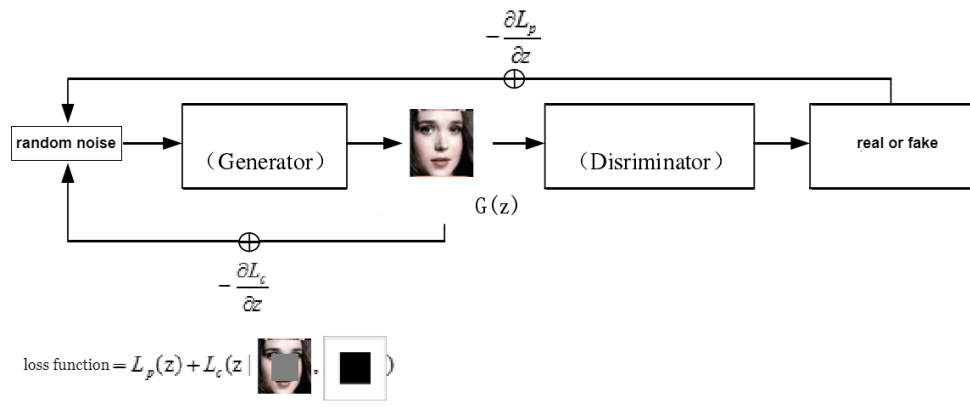


Figure 2: The framework of image completion algorithm

4 Experiment and Analysis

4.1 Experimental Environment

This experiment is based on the Linux kernel of Ubuntu 14.4 operating system, GPU version for GTX 1080ti, programming environment using Python 3.6, based on Tensorflow framework.

4.2 Data Set Processing

Before being used in the experiment, the image needs to be cropped to 64×64 pixel size in the center, which needs to contain faces with various viewpoints and expressions. Firstly, preprocess CelebA face data set and download image from the network `img_align_celeba.zip`. And decompress, randomly select 300 of them as the data to be repaired and put them into the files to be repaired for standby.

In order to successfully apply the data image to the training of the model, we need to further process the image. Because all the current images are original human images, our model needs $64 \times 64 \times 3$ facial images, so we need to use Python face in data set processing. The recognition library locates the facial position and cuts it to the same height and width, and finally changes it into a 64×64 size picture. The face recognition library mainly encapsulates the C++ graphics library `dlib`, and encapsulates it into an API library that can realize face recognition by Python language, which shields the algorithm details of face recognition and greatly reduces the difficulty of developing face recognition function.

The algorithm traverses all the images in the folder and uses the installed face-Recognition library to identify the upper, lower, left and right coordinates of the face position, and finally cut it into a 64×64 pixel image as shown in Fig. 3.

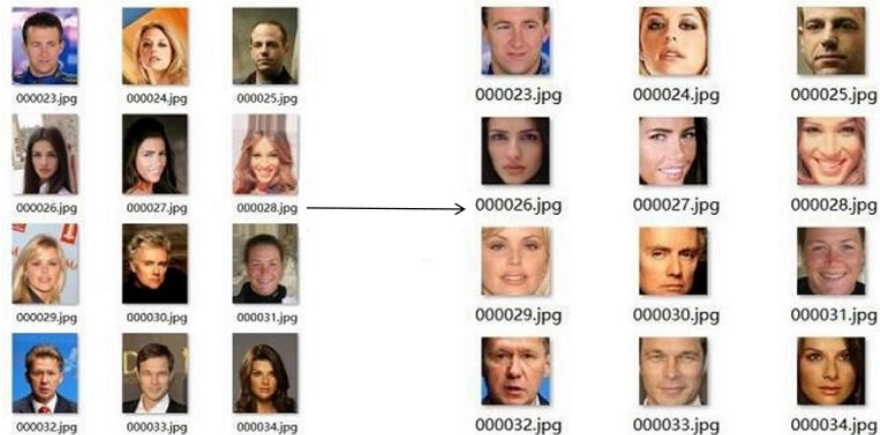


Figure 3: The comparison before and after image pretreatment

4.3 Visual Comparison

In this experiment, image completion is based on the principle of semantic repair of deep generation model. Based on the analysis and research of the original generative countermeasure network, we choose the deep convolutional generative adversarial network as the basic model of image completion. Then, we combine the importance weighted context loss and prior loss in the loss function of image completion, and then iterate the depth neural network 5000 times. The comparison of repair results after 5000 times is shown in Fig. 4.

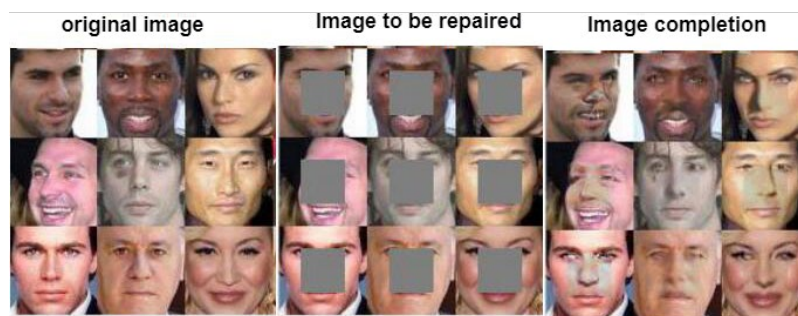


Figure 4: comparison of image completion results

From Fig. 4, we can see that the experimental model basically has the ability to generate missing region pixels on the occluded face image, and the image restoration results are relatively good.

Compared with the original generative adversarial network model, the contrast results of image restoration based on deep convolutional generative adversarial network model are shown in Fig. 5. The experiment is carried out in CelebA face data set, and the repair situation is compared after 1000 iterations.

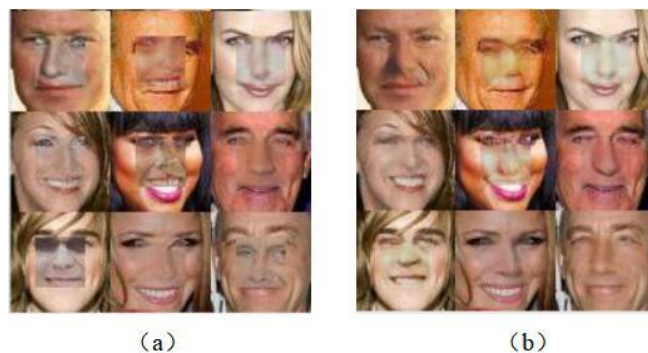


Figure 5: comparison of image completion results based on GAN and DCGAN.

In Fig. 5, (a) is the repair result based on the original generative adversarial network model, and (b) is the repair result based on the deep convolutional generative adversarial network. As can be seen from the results shown in Fig. 5, the image completion model based on DCGAN has a better restoration effect, and is more coherent and realistic on the whole. The image completion model based on the original GAN has some problems, such as incoherent pixels, unreal texture and structure.

5 Conclusion

In this experiment, we use the network model based on deep convolutional generative adversarial network, and combine the context loss function and perception loss function to repair the authenticity and rationality of the image. CelebA face data is selected as the training data set, after a certain number of iterative training, it has the ability to repair large area of incomplete images. The image completion algorithm can get acceptable restoration results visually, and the restoration effect has corresponding coherence and authenticity. From the visual effect of image completion and the quality of the repaired image, it has a better performance effect.

Acknowledgement: We are grateful to the peoples for the support and encouragement.

Funding Statement: This work was supported by Scientific Research Starting Project of SWPU (No. 0202002131604); Major Science and Technology Project of Sichuan Province (No. 8ZDZX0143); Ministry of Education Collaborative Education Project of China (No. 952); Fundamental Research Project (Nos. 549, 550).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] X. Peng, "Image inpainting research based on TV model and texture synthesis," Master dissertation, South China University of Technology, Guangzhou, 2011.
- [2] H. Y. Zhang and Q. C. Peng, "A survey on digital image inpainting," *Journal of Image and Graphics*, vol. 22, no. 4, pp. 587–597, 2007.
- [3] Z. Ran, D. Zheng, Y. Lai and L. Tian, "Applying stack bidirectional LSTM model to intrusion detection," *Computer, Materials & Continua*, vol. 65, no. 1, pp. 309–320, 2020.
- [4] M. Bertalmio, G. Sapiro and V. Caselle, "Image Inpainting," in *Proc. ACM*, New Orleans, LU, France, pp. 417–424, 2000.
- [5] R. L. Biradar and V. V. Kohir, "A novel image inpainting technique based on median diffusion," *Sadhana*, vol. 38, no. 1, pp. 621–644, 2013.
- [6] W. Brendt, "Inpainting by joint optimization of linear combinations of exemplars," *Signal Processing Letters*, vol. 18, no. 1, pp. 75–78, 2011.
- [7] A. Criminisi, P. Perez and K. Toyama, "Object removal by exemplar-based inpainting," in *Proc. CVPR*, Wisconsin, WI, USA, pp. 2124–2246, 2003.
- [8] N. Srivastava, G. Hinton and A. Krizhevsky, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 22, no 4, pp. 1929–1958, 2014.
- [9] R. I. Fidane, "A survey on variational image inpainting texture synthesis and image completion," Ph.D. dissertation, Bogazici University, Istanbul, 2008.
- [10] B. Shen, W. Hu and Y. Zhang, "Image Inpainting via Sparse Representation," in *Proc. ICASSP*, Taipei, Taiwan, pp. 621–632, 2009.
- [11] J. Mairal, F. Bach and J. Ponce, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, no. 1, pp. 45–53, 2010.
- [12] C. Yang, X. Lu and Z. Lin, "High-resolution image inpainting using multi-scale neural patch synthesis," in *30th IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, no. 4, pp. 4076–4084, 2017.

- [13] D. Pathak, P. Krahenbuhl and J. Donahue, "Context encoders: Feature learning by inpainting," *Computer Vision and Pattern Recognition*, vol. 1, no 1, pp. 2536–2544, 2016.
- [14] X. Y. Li, Q. S. Zhu, M. Z. Zhu and Y. M. Huang, "Machine learning study of the relationship between the geometric and entropy discord," *Europhysics Letters*, vol. 6, no. 7, pp. 542–551, 2019.
- [15] A. Radford, L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *Computer Science*, vol. 1, no. 4, pp. 89–96, 2015.
- [16] L S. Meng, K. Ren, C. Q. Fan and L. Huang, "Dense convolution generative adversarial networks based image inpainting," *Computer Science*, vol. 47, no. 8, pp. 202–207, 2020.
- [17] P. L. He, Y. X. Shi and J. Cheng, "Face image translation based on generative adversarial text," *Computing Technology and Automation*, vol. 37, no. 4, pp. 77–82, 2019.
- [18] R. Yeh, C. Chen and T. Y. Lim, "Semantic image inpainting with deep generative models," in *Proc. CVPR*, Honolulu, HI, pp. 6882–6890, 2017.
- [19] S. Lizuka, S. S. Rae and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–14, 2017.