International Journal of Chemical and Biochemical Sciences
(ISSN 2226-9614)

Journal Home page: www.iscientific.org/Journal.html

Generate medical cell images that can be controlled by MCGAN

Xu Tao, Rowell Hernandez*

Batangas State University, The National Engineering University Batangas City, Philippines

Abstract

Due to hospital privacy and policy restrictions and scarcity of medical image samples for medical image development, there are few medical image databases available for deep learning training and all access to medical image databases is difficult. This study identifies a research endeavor to generate medical cell images that can be controlled by masking cell position and overlap information. The experiments concluded that the MCGAN algorithm largely improves the generation quality of generative adversarial networks and enriches the diversity of manually collected data.

Keywords: MCGAN, High-precision cell detection, Generative adversarial network model, Diversity of data

Full length article *Corresponding Author, e-mail: 21-03030@g.batstate-u.edu.ph Doi # https://doi.org/10.62877/62-IJCBS-24-25-19-62

1. Introduction

In 2014, Lan J. Goodfellow et al. from the University of Montreal proposed a Generative Adversarial Networks (GAN) model; compared with convolutional neural networks, generative adversarial networks are mainly divided into two parts of the composition: generator (Generator) and discriminator (Discriminator). Generator G is used to learn the actual probability distribution to generate the "real image," while Discriminator D distinguishes between the actual image and the "real image" generated by Generator G as much as possible, looks for the difference between the two, and feeds this difference back to Generator G. In the iterative process, Generator G generates the "real image," and Discriminator D generates the "real image." In the iterative process, the generator G generates as many "real images" as possible to deceive the discriminator D, and the discriminator D tries to distinguish the images generated by the generator G from the actual images [1-4]. Generator G and discriminator D constitute a dynamic "game process." The game between the two is to achieve an ideal state: generator G can generate a "fake" picture; discriminator D finds it difficult to distinguish whether the picture generated by generator G is true or false. The cost function is expressed as follows:

$$\min_{G}\max_{D} E_{x\in X}\left[\log\left(D(x)\right)\right]+E_{s\in Z}\left[\log\left(D(z)\right)\right]$$

2. Generate medical cell images that can be controlled by mask for cell position and overlap information (MCGAN)

Conditional one-dimensional vector containing light and color information; Mask: The mask is a manually labeled image, and the mask contains information on the location, size, and number of cells. X is the Gaussian distribution of the input, z is the image's label, M is a mask, and C is a one-dimensional condition vector [5]. Generative Adversarial Networks (GANs) are algorithms in the field of image synthesis that are capable of generating images with higher accuracy and better quality. Structurally, GANs consist of two interconnected structures: a generator network (G) and a discriminator network (D). The generator network creates realistic data, while the discriminator network evaluates the generated data against real examples. Through iterative competition and collaboration, these networks continuously refine their weights and straighten their equilibrium. That is, the generators produce highly realistic images, while the discriminators are unable to distinguish them from accurate data. However, because GANs do not generate enough diversity and do not have enough control condition functions, many scholars have investigated various GAN branch areas [6-7]. Generative Adversarial Networks (GANs) utilize a specified mathematical equation as the governing principle for the weight adjustment process during the learning phase:

$$L_{GAN}(G,D)=\min_{G}\max_{D}(E_{x\sim P_{data}(x)}\log(D(x))+E_{z\sim P_{data}(z)}\log(1-D(G(z)))) \quad (1)$$

In the equation: $\min_{G}$ the minimum value that represents the generated network error; $\max_{D}$ 表 Maximum value for discriminating network errors; $z\sim P_{data}(y)$ representing the true distribution of sample.

$x\sim P_{data}(x)$ representing the distribution of random noise samples; E The mean representation of the probability of calculation. In contemporary generative adversarial networks, the generative

network G acquires noise z as its input in the latent space. As z traverses fully connected layers, G produces synthesized images. On the other hand, the discriminative network D plays the role of discerning genuine images from counterfeit ones. Nevertheless, it is crucial to note that a direct mapping between the input and output does not exist. Henceforth, conditional generative adversarial networks employ the ensuing learning principles to modify the weightage of prevailing generative adversarial networks:

$$L_{cGAN}(G,D)=\min_{G}\max_{D}[E_{x\sim P_{data}(x)}\log(D(y,x)) +E_{z\sim P_{data(z)}}\log(1-D(y,G(z,y)))] \quad (2)$$

The label information corresponding to accurate sample data is represented in mathematical notation. Adding labels to Gaussian noise can make the results of the generative adversarial network controllable (Figure 1). For example, Pix2pix expands on Conditional Generative Adversarial Networks (CGAN) by substituting noise and conditional information with mask images, thereby facilitating image-to-image translation. The adversarial network error is regulated by a regularization term error after generating the adversarial loss to maintain a strong resemblance between the input and output images [8].

$$G=\arg\min_{G}\max_{D}L_{cGAN}(G,D)+\lambda L_{L1} \quad (3)$$

To establish an error term between the detected image and the corresponding image labels L_1^I. During the training process, it is possible to perform detection simultaneously while generating.

$$L_1^I(G)=E_{x,I}[\| I-G(x)\|_1] \quad (4)$$

In the equation: I Indicates the location label corresponding to the target original image; x To express the original image, an optimizer is established on both the original image and the labels of the original image, in conjunction with the detection network.

The definition of the formula of confrontation is defined as:

$$L_{G\_adv}^{LCGAN}+L_{D\_adv}^{LCGAN}=\min_{G}\max_{D}[E_{x\sim P_{data}(x)}\log(D(y,x)) +E_{z\sim P_{data(z)}}\log(1-D(y,G(z,y)))] \quad (5)$$

$$L_1^x(G)=E_{x,z,y}[\| x-G(z,y)\|_1] \quad (6)$$

In the equation: $L_{G\_adv}^{LCGAN}$ Representation of Generative Network Items; $L_{D\_adv}^{LCGAN}$ Representation of Discriminatory Network Items. The application of the aforementioned formula guarantees the similarity between the synthesized image and the desired image. To achieve this, a regularization term is incorporated to minimize the discrepancy between the two images. In order to ensure the generation of high-quality images, the concept of structural similarity (SSIM) is introduced. SSIM is a metric for evaluating images with a clear focus on retaining some of the information. The output of SSIM ranges between 0 and 1, with a value closer to 1 indicating a higher quality of the generated image. Also, in the experiments, the data from the labelled samples are used to calculate the SSIM values and to assess the similarity between the generated image and the real image in terms of brightness, contrast and structure. This method ensures the overall quality of the generated images:

$$L_{SSIM}(z,x)=1-\frac{1}{N}\sum_{p=1}^{p}SSIM(p) \quad (7)$$

In the equation: N The number of pixels in the image; p The value of the central pixel representing a pixel block, Defined as:

$$SSIM(p)=\frac{2u_z+u_x+c_1}{u_z^2+u_x^2+c_1}\times\frac{2\sigma_{zx}+c_2}{\sigma_z^2+\sigma_x^2+c2}, \quad (8)$$

In the equation: u_z 、 u_x It denotes the average pixel value of an image; σ_z 、 σ_x Standard deviation represents the variation in pixel values of an image; σ_{zx} the covariance of two image pixels; c_1 and c_2 both are constants. By combining the above formulas, we can derive the definition of the total loss function as follows:

$$L_{LCGAN}=SSIM(p)\left(L_{G\_adv}^{LCGAN}+L_{D\_adv}^{LCGAN}+\lambda_1 L_{SSIM}+\lambda_2 L_1^x+\lambda_3 L_1^I\right) \quad (9)$$

In the equation: λ_1 、 λ_2 、 λ_3 are all constants.

3. Network architecture

3.1. Overall structure

Flow chart is given in Figure 2.

3.2. Generative Network Architecture

We have incorporated residual blocks into the intermediate architecture to improve the generative network's ability to learn from data. This allows the network to gather detection information during the training process. The generative network is made up of three modules. The first module includes three convolutional layers. The second module consists of six residual blocks. The third module includes two deconvolutional layers and one convolutional layer. In this study, we have employed a fully convolutional network for the generative network, excluding fully connected and pooling layers.

The ReLU function is used as the activation function, while the final layer employs the Tanh function as its activation function [9]. Conditional vectors in the intermediate architecture, through which the lighting of the image and information about the image are controlled [10].

3.3. Discriminative Network Architecture

The imbalance between the generative and discriminative network capability occurs during the training process, resulting in the generative Network being unable to compete with the discriminative Network. So, limiting the network depth of the discriminative Network limits the ability of the discriminative Network to a certain extent. The structure of the discriminative Network is shown in Figure 3. It consists of five convolutional layers without using fully connected and Pooling layers.

4. Experimental Analysis

4.1. Dataset

The study used a dataset of manually collected images of human blood red blood cells obtained by the University. We divided the images into a training set and a test set where 80% of the images were used for training and 20% for testing. The model was trained using the training set and its performance was evaluated using the test set.

4.2. Experimental Comparison

The experiment was performed on a Linux operating system, utilizing the Tensorflow 1.20 deep learning framework. The computer setup consisted of an NVIDIA 4080Ti-12G graphics card and two DDR4-3000-32G memory sticks. To ensure unbiased results, the dataset was randomly split into a training set and a test set, with a ratio of 8:2. Each training model underwent an equal number of iterations. Furthermore, a comparative approach was employed by using the original image dataset with labels to ensure a fair quantitative comparison of the generated quality. After randomly selecting three sets of experiments, I observed and recorded the convergence data of the loss function. As training iterations increased, the loss function consistently decreased in the convergent region. This decrease in the loss function is driven by the mechanism of generating an adversarial loss, which continuously opposes the loss during the iteration process. To evaluate the quality of the generated images, we can also perform quantitative analysis using metrics such as PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), and IS (Inception Score). PSNR is a metric that assesses the quality of images by directly measuring the differences between pixel values. A higher PSNR value indicates less distortion. It is calculated by comparing the grayscale difference between pixels in the generated image and the real sample. The formula for calculating PSNR is as follows:

$$L_{PSNR} = 10\log_{10}\frac{255^2 \times W \times H}{\sum_{i=1}^{W}\sum_{j=1}^{H}[z(i,j)-x(i,j)]^2} \quad (12)$$

In the equation: W and H are used to represent the width and height of an image. The SSIM metric, or Structural Similarity Index, is utilized during the training process to assess the similarity between two images. It generates a value ranging from 0 to 1, where a value closer to 1 signifies a smaller disparity between the image and the actual image, thus denoting higher image quality. A higher SSIM value implies that there is less loss of detailed information from the original image. On the other hand, the Inception Score (IS) metric is employed to compare and evaluate image quality. A higher IS value indicates better image quality. Table 1 showcases the image quality of different models using three different methods of measuring image quality. The loss function of the generative adversarial network was selected above for comparison, and it can be seen that as the number of training iterations deepens, the loss function decreases in the convergence region, and the generative adversarial loss mechanism is a process of constant confrontation that prompts the loss function to decrease as the number of iterations increases (Fig. 4). The results generated by the independent generator, with random input masks to the generator. Figure 5 (b) shows the results generated without the SSIM loss term, and Figure 5 (c) shows the results generated with the SSIM loss term. It is clear that the edges of the cells are much sharper and meet the requirements of the cell detection database. Figure 5 (d) and Figure 5 (e), on the other hand, are both overlapping cell images generated with the Generative Adversarial Networks algorithm in this chapter, and the image overlap in Figure 5 (e) is particularly severe, which largely augments the cell database. Table 1, the experiments found that the addition of SSIM term on the generator can improve the overall image generation quality, for small scales on the addition of MS-SSIM term to a certain extent can increase the local generation quality and improve the generation effect. The red mask label (Fig. 6), Gaussian kernel label and the trained cellular dataset, each figure corresponds to each other. It belongs to the experimental validation part, mainly to verify the experimental accuracy and the original figure for comparison. The table 2 shows the comparison of image generation of GAN and MCGAN, the quantitative analysis shows that the generation quality of MCGAN is better than that of the original GAN, and also significantly better than that of the original GAN in terms of PSRN, SSIM metrics, but this paper's improved GAN is more suitable for image generation for this database.

Figure 7, the above three tables are the image generation comparison of GAN, DCGAN and CGAN, quantitative analysis can be seen that the generation quality of DCGAN and CGAN is better than the original GAN generation quality, in the PSRN, SSIM indexes are significantly better than the original GAN generation quality, but in this paper's improved GAN is more suitable for the image generation of this database. The subjective analysis is also given in the following figure, which shows the quality of images generated by the original GAN, DCGAN and CGAN compared to the improved GAN in this paper.

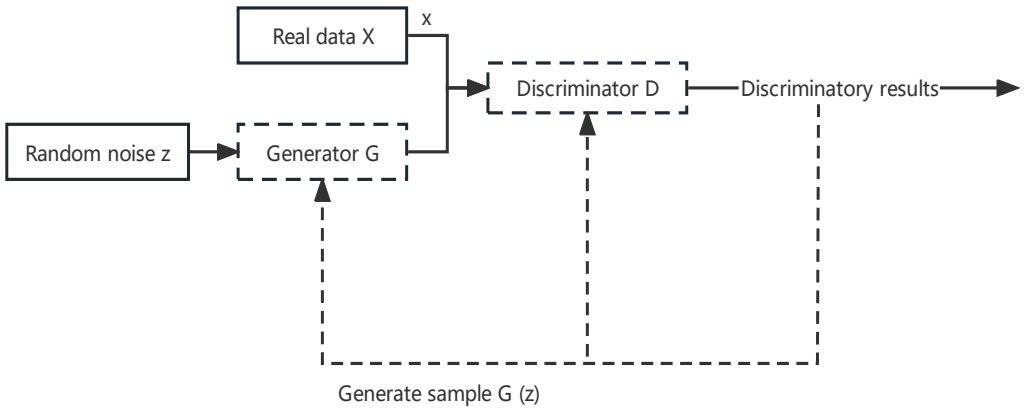

Figure 1: Flowchart of the model for generating adversarial networks.

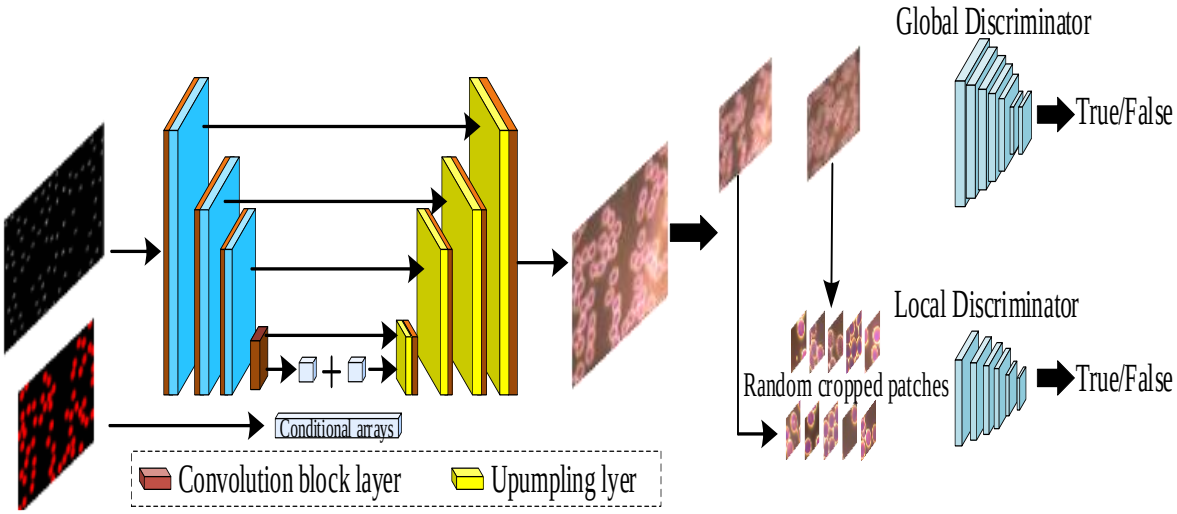

Figure 2: Flow Chart.

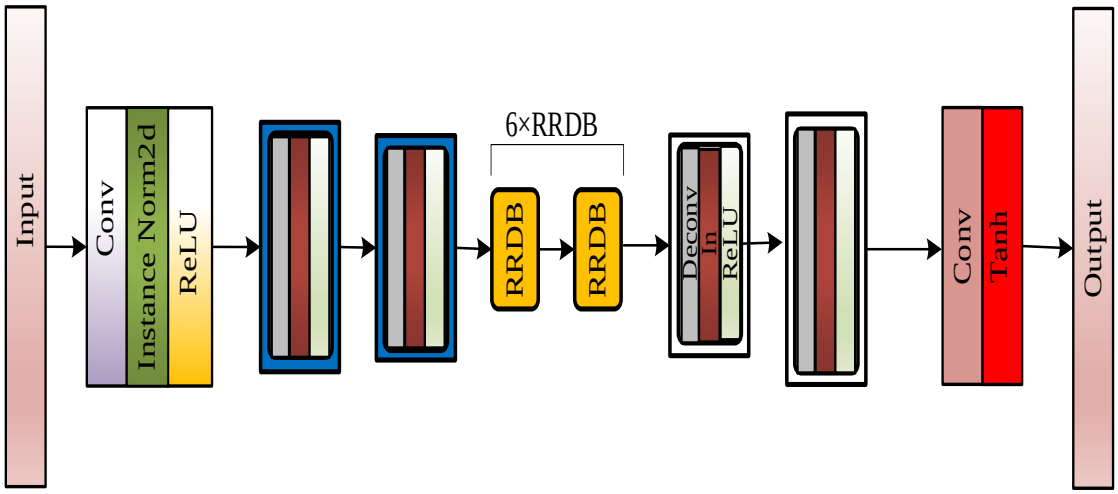

Figure 3: Structure of discriminator network.

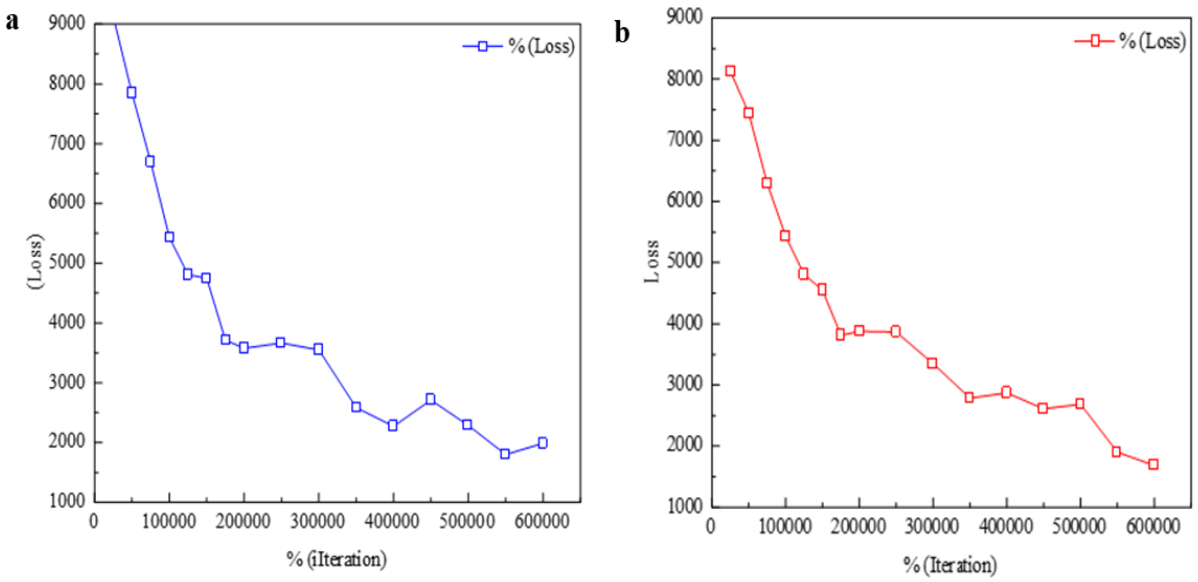

Figure 4: Loss Function.

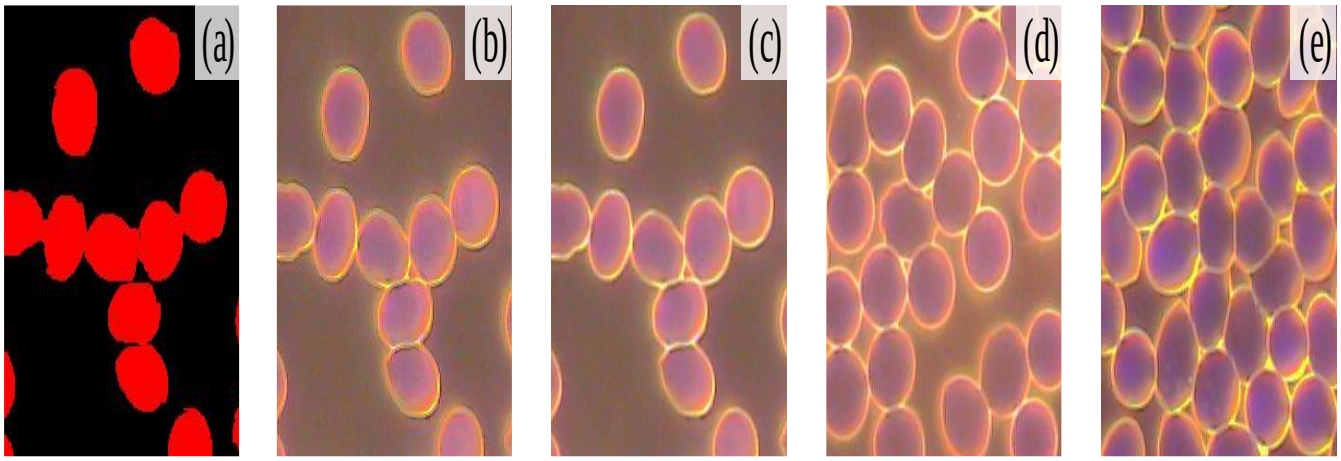

Figure 5: Experimental data.

Table 1: Experiment contrast result of various test methods

Method name	Number of iterations	PSRN	SSIM	MSE
GAN	50000	18.23	0.875	232.4
CGAN	50000	20.87	0.905	198.2
MCGAN	50000	20.64	0.907	202.3

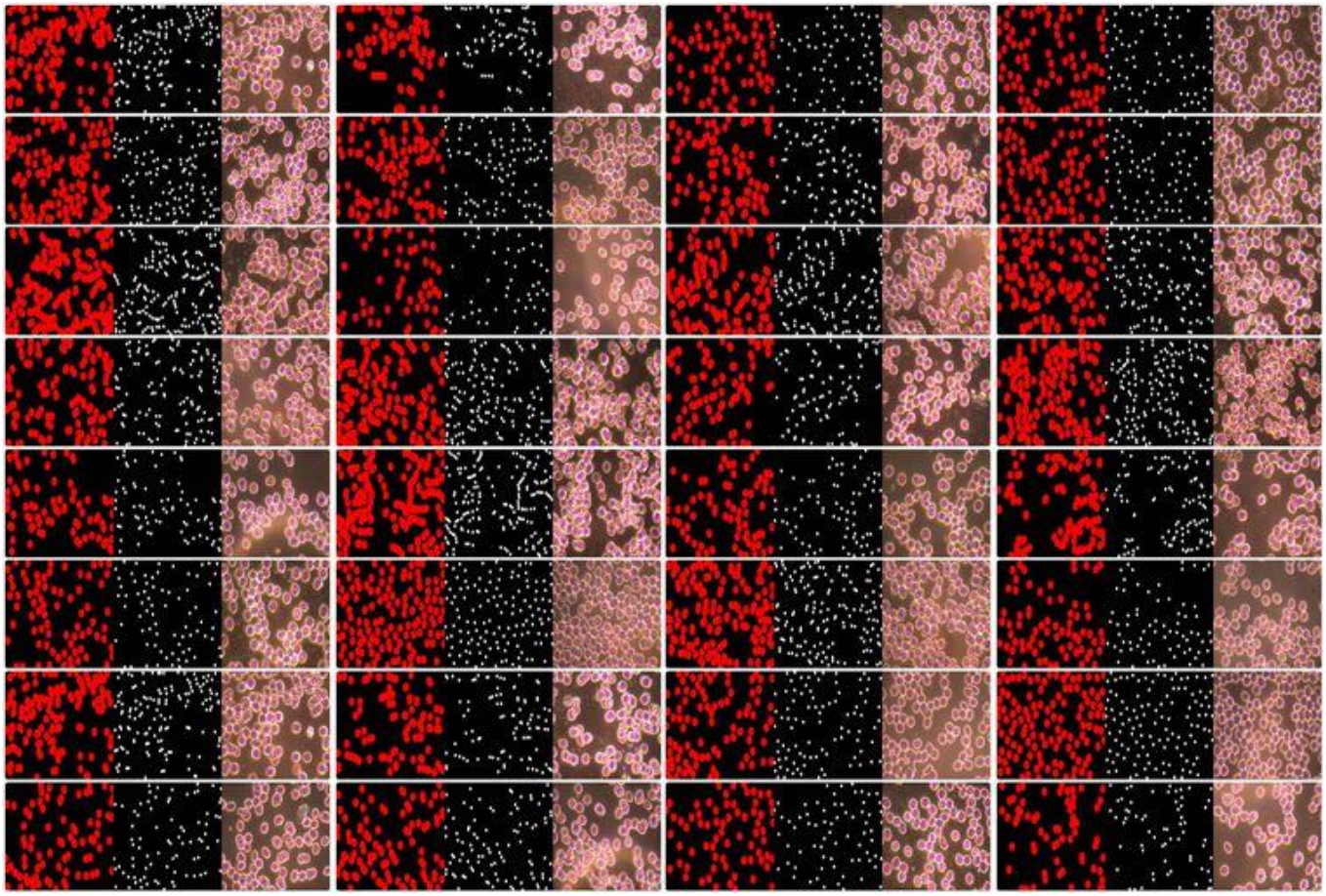

Figure 6: Experimental data.

Table 2: Comparison of quantitative results of PSNR, SSIM and IS.

Method name	Number of iterations	PSRN	SSIM	MSE
GAN	50000	12.39	0.77	439.96

Method name	Number of iterations	PSRN	SSIM	MSE
MCGAN	50000	15.13	0.86	387.43

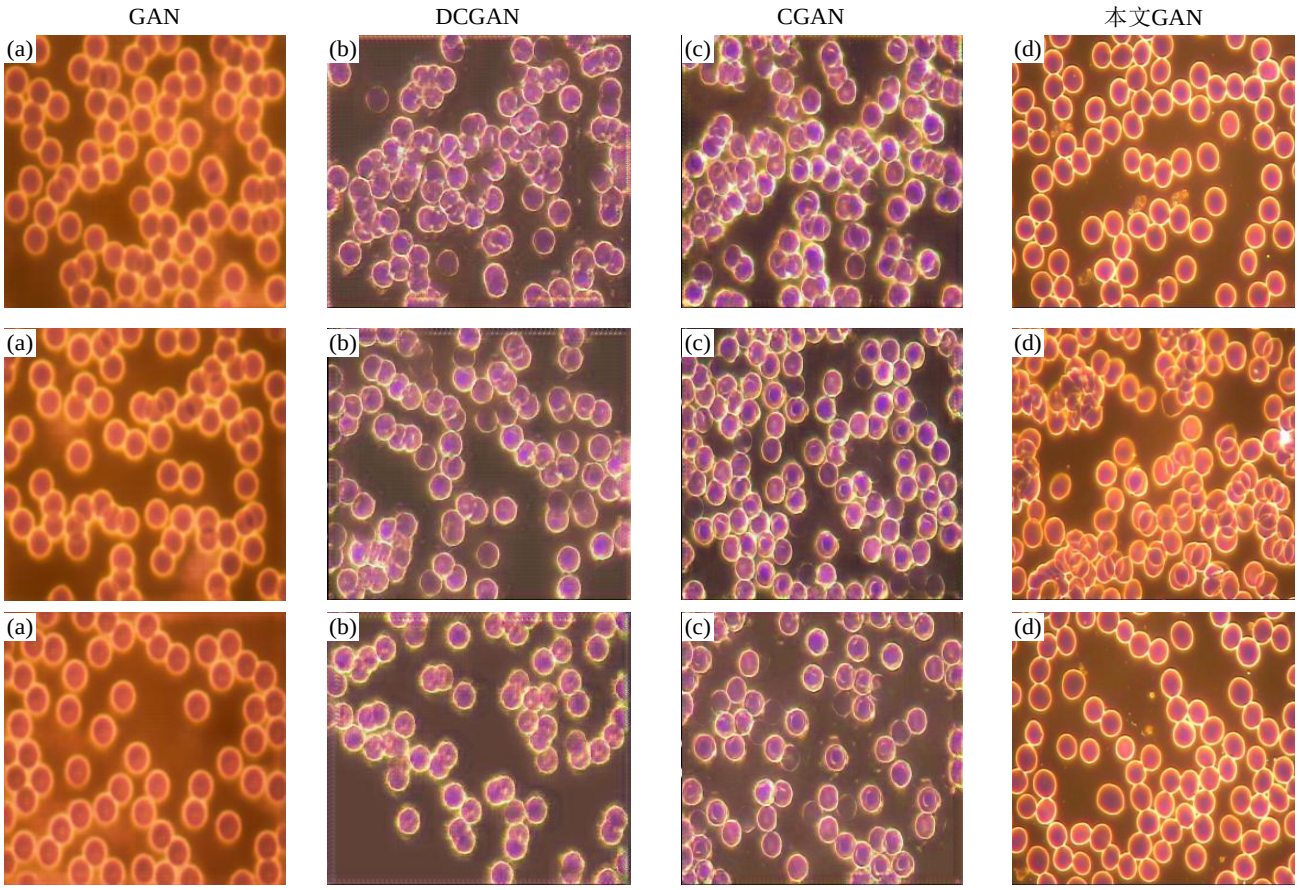

Figure 7: Generative result of GAN

5. Conclusions

This paper realizes high-precision cell detection with a generative adversarial network model, which should be able to detect target images with adhering and overlapping complex backgrounds and less contrast with the surrounding background. The experiments concluded that the MCGAN algorithm largely improves the generation quality of generative adversarial networks and enriches the diversity of manually collected data

References

[1] X. Lv, X. Zhang. (2019). Generating chinese classical landscape paintings based on cycle-consistent adversarial networks. In 2019 6th International Conference on systems and Informatics (ICSAI). 1 (1): e1265-e1269.

[2] H. Hammouch, S. Mohapatra, M. El-Yacoubi, H. Qin, H. Berbia, P. Mäder, M. Chikhaoui. (2022). GANSet-Generating annnotated datasets using Generative Adversarial Networks. In 2022 International Conference on Cyber-Physical Social Intelligence (ICCSI). 1 (1): e615-e620.

[3] S. A. Shah, S. H. Drabu, M. M. Khan, D. B. Q. Kirmani, S. N. Singh. (2023). Generative Adversarial Networks (GANs) in Computer-Generated Imagery. In 2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI). 1 (1): e171-e174.

[4] H. Kapalavai, S. Mondal. (2023). Generating New Human Faces and Improving the Quality of Images Using Generative Adversarial Networks (GAN). In 2023 2nd International Conference on Edge Computing and Applications (ICECAA). 1 (1): e1647-e1652.

[5] Y. Quan, C. Liu, Z. Yuan, B. Yan. (2024). Hybrid Data Augmentation Combining Screening Based MCGAN and Manual Transformation for Few-Shot Tool Wear State Recognition. IEEE Sensors Journal.

[6] D. K. Vishwakarma. (2020). Comparative analysis of deep convolutional generative adversarial network and conditional generative adversarial network using hand written digits. In 2020 4th international conference on intelligent computing and control systems (ICICCS). 1 (1): e1072-e1075.

[7] W. Cheng, S. Zhang, Y. Lin. (2023). Study on the Adversarial Sample Generation Algorithm Based on Adversarial Quantum Generation Adversarial Network. In 2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS). 1 (1): e238-e243.

[8] G. Ikhsan, N. Suciati. (2022). The Comparative Study of Adding Edge Information to Pix2pix Architecture for Face Image Generation. In 2022 6th

International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE). 1 (1): e160-e165.

[9] C. L. Li, C. Y. Su. (2022). Multi-connection of double residual block for yolov5 object detection. In 2022 8th International Conference on Applied System Innovation (ICASI). 1 (1): e13-e16.

[10] S. N. Farsani, J. Fisher, J. R. Kim, M. Jafaritadi, C. S. Levin. (2023). Achieving BSREM image quality at the speed of OSEM reconstruction using the cGAN framework. In 2023 IEEE Nuclear Science Symposium, Medical Imaging Conference and International Symposium on Room-Temperature Semiconductor Detectors (NSS MIC RTSD). 1 (1): e1.