Intelligent Design of Ethnic Patterns in Clothing using Improved DCGAN for Real-Time Style Transfer

Yingjun Liu, Ming Wu*
School of Art, Guangxi Minzu Normal University, Chongzuo 532200, China

***Abstract*—In view of the problems that traditional real-time style transmission technology requires a large number of sample map training, low image quality, lack of realism and detail, this study combines the improved generative adversarial network (GANs) with real-time style transfer technology, and enhances the real-time style transfer calculation with adaptive instance normalization. As a result, a novel intelligent clothing ethnic pattern design model is developed. Experimental results show that the model reduces physical memory usage by 45.7%, with only 453MB, and utilizes only 26% of CPU resources in terms of CPU usage. The training time is approximately 20 minutes and 48 seconds. This model performance is obviously higher than other models. The designed intelligent clothing ethnic pattern design model in this study demonstrates higher clarity and shorter processing time, and has potential applications in the field of image generation.**

***Keywords*—Computer vision; improved DCGAN; style transfer; adaptive instance normalization; intelligent design of patterns**

I. Introduction

In today's era, clothing design plays a crucial role in personalization and innovation. With the growing demand for unique styles and cultural diversity, ethnic patterns have become a popular design element. However, manually drawing and applying these patterns require a significant amount of time and labor, limiting their application in large-scale production. To address this issue, computer vision and deep learning technologies have been widely applied in the field of fashion design [1]. In particular, style transfer methods based on Generative Adversarial Networks (GANs) have made significant progress. Among them, Deep Convolutional GAN (DCGAN) is a powerful generative model that can learn from input data and generate realistic images [2-3]. Style transfer is a technique of applying the style of one image to another image, which is widely used in artistic creation and design. AdaIN is an instance normalization technique that is adaptive to adjust the style of an image, making the result of style transfer more natural and realistic.However, there are some defects in the traditional intelligent design technology. On the one hand, traditional techniques often require a large number of sample images for training, which can be very difficult and time-consuming for specific ethnic patterns. On the other hand, traditional techniques may lead to a low quality of the generated images and a lack of realism and detail. Moreover, traditional techniques may not very well preserve the style and character of the original pattern. For the above problems, this study proposed a series of improvement measures for standard DCGAN, introduced in the decoder network, the generated false image more close to the real image, adopted the method of multi-scale feature extraction and fusion in the generator and discriminator network, makes the generated false image more real, by stacking the convolution layer and deconvolution layer to improve the quality of the generated image, and an improved algorithm of deep convolution generation against network (IDCGAN) is developed. In addition, the adaptive instance normalization (AdaIN) is also applied to the real-time style transfer technology to design an intelligent model for the clothing ethnic pattern design. The study aims to achieve image generation through adversarial learning, allowing DCGAN to learn specific ethnic patterns from a small number of samples and generate high-quality, realistic images. The article consists of four main parts. The second part provides a comprehensive review of the current research status of intelligent clothing design and style transfer systems. The third part establishes an intelligent model for ethnic pattern design in clothing based on improved DCGAN for real-time style transfer. The fourth part includes comparative experiments and efficiency verification to evaluate the optimization effects of the model.

II. Related Works

As people's pursuit of personalization and unique styles continues to grow, traditional clothing design no longer meets the demands of consumers. In order to meet this demand, researchers have been exploring intelligent design technologies that are adaptable to these needs. Ding et al. addressed the issues of low accuracy and stability in traditional manual crown design by developing an automatic crown design strategy with DCGAN, and the outcomes indicates that it had the smallest morphological differences compared to natural teeth [4]. Abd Al et al. proposed a DCGAN based algorithm and a novel Capsule Network to assist semiconductor manufacturers in identifying defect patterns in wafers, and the experimental results showed that the method achieved a training accuracy of 99.59% and a validation accuracy of 97.53% [5]. Bian et al. developed a compound screening model based on DCGAN to screen and design novel compounds with target-specificity for cannabinoid receptors, and the experimental results showed that the model had the highest accuracy compared to other models [6]. Cheng et al. proposed a Data Enhancement Communication Behavior Recognition (DECBR) scheme to address the limitations of traditional communication behavior recognition techniques in accurately analyzing communication behaviors, and the DECBR scheme significantly improved the accuracy and efficiency of behavior recognition under small sample conditions [7]. Li et al. designed an image classification model that combines DCGAN and AlexNet for rapid differentiation of multiple forms of glioblastoma images, and the experiment outcomes indicates that the model reached 0.920 accuracy and an AUC of 0.947 for distinguishing PsP and TTP after 10-fold cross-validation [8]. Ni et al. addressed the issue of large deviations in carrot quality identification using traditional visual inspection methods by designing a carrot quality identification model that combines DCGAN and Squeeze-and-Excitation Deep Networks, and the experiment outcomes showed that the model reached 98.36% accuracy [9].

Jing et al. proposed a new normalization module called Dynamic Instance Normalization (DIN) to address the deployment challenges of style transfer systems in resource-constrained environments. DIN allows for flexible and more efficient transfer of arbitrary styles. The experimental results showed that this approach reduced computational costs by more than 20% compared to existing methods [10]. Reimann et al. addressed the issue of one-shot stylization in existing style transfer computations, which mostly limit the style elements interactive adjusting. They designed a fast style transfer network which is stroke-adjustable. It can simultaneously control stroke intensity and size. The experimental results showed that the model make users achieving resolutions exceeding 20 million pixels and good output fidelity [11]. Hollandi et al. developed a deep learning-based cell nucleus segmentation framework that utilizes image style transfer to automatically generate cell nucleus segmentation masks. This framework aims to find a method for locating 2D cell nuclei in different regions. The experimental results showed that the model effectively identifies cell nuclei in different experiments without the need for expert annotations [12]. Huang et al. addressed the lack of diversity in traditional style transfer by designing a style transfer model that combines region semantics with multi-style transfer. The experimental results showed that the model seamlessly combines multiple styles together, and, with the assist of semantic matching, assigns corresponding styles to content regions [13]. Xu et al. tackled the problem of indistinguishable details between different types of objects caused by single-band imaging. They designed a target detection-oriented style transfer network for panchromatic remote sensing images. After style transfer, the target detection accuracy on panchromatic remote sensing images significantly improved [14]. Zhou et al. proposed a new approach that combines attention mechanism with style transfer models to enhance the flexibility of style transfer tasks. The experimental results showed that this approach is effective and produces high-quality images [15].

In summary, DECBR and style transfer have a solid theoretical and implementation foundation in the field of intelligent clothing design. However, there is limited research that combines the two for ethnic clothing design. Therefore, the study aims to improve DECBR and combine it with style transfer computations to develop an intelligent clothing design model, in order to further advance the clothing design industry.

III. Improvement of DCGAN Algorithm and Establishment of Intelligent Pattern Design Model

This chapter contains two sections. The first gives an introduction on the standard Deep Convolutional Generative Adversarial Network (DCGAN) and proposes some improvement strategies to address its limitations. The second section focuses on improving traditional real-time style transfer networks and establishing an intelligent design model based on real-time style transfer networks.

A. *Improvement of DCGAN Algorithm Design*

DCGAN is a neural network model used for generating realistic images. It combines the ideas of generative models and adversarial training, mainly containing two components: the generator and the discriminator [16-17]. The former is a network that uses random noise vectors to do input and attempts to bring up images similar to the training data. The generator gradually constructs the image through multiple convolution, deconvolution, and activation function operations. The discriminator is also a convolutional neural network whose goal is distinguishing between the images generated and real ones. The discriminator extracts image features through operations such as convolution, pooling, and activation functions, and outputs a probability value between 0 and 1, indicating the likelihood that the input image is a real image [18]. In DCGAN, the two components are alternately trained. The generator manufacture images that is real enough for deceiving the discriminator, while the later strives to differentiate between the images generated by the generator and real images [19]. This process stimulates the generator to continuously enhance the generated images' quality and makes the discriminator more accurate, as shown in Fig. 1.

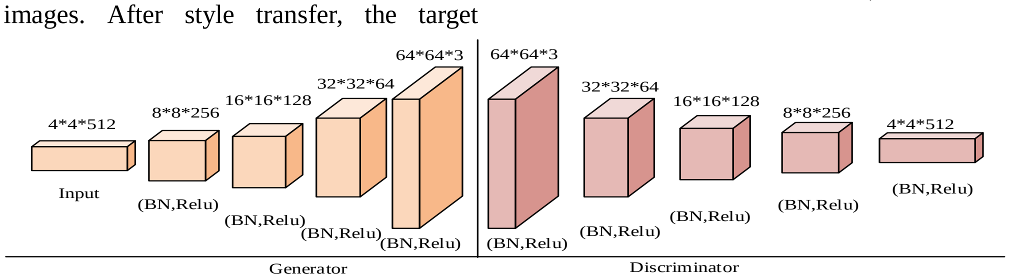


Fig. 1. Schematic diagram of DCGAN structure.

Although DCGAN is a widely used entity in the studying areas of image synthesis, editing, and super-resolution reconstruction, there are some limitations when applying DCGAN to real-time style transfer of ethnic clothing patterns' intelligence design. Firstly, DCGAN has relatively weak feature extraction capabilities. Despite using convolutional neural networks (CNN) to learn image features, it may not fully capture the complex features of ethnic patterns due to network structure and training data limitations. This can result in generated clothing that is not realistic enough and deviates significantly from the target style. Secondly, there is a lack of overall style transfer constraints. DCGAN primarily focuses on generating realistic images during the training process, with less emphasis on maintaining consistency and layout of local patterns. In clothing design, maintaining pattern consistency is crucial, but DCGAN may not fully consider the layout and details of patterns in different parts of the garment, resulting in clothing that does not resemble a normal garment and lacks coherence and integrity. To address these limitations, a modified DCGAN approach is proposed, and the network structure during the training phase is shown in Fig. 2.

Fig. 2 illustrates the network structure during the training phase of IDCGAN. At the beginning of training, random noise is input into the generator. The generator processes the noise through decoding and encoding operations to generate fake images. These fake images gradually approach real images through the generator. At the same time, real and fake images are simultaneously input into the discriminator. It is another network responsible for classifying the input images and outputting the feature space Z. This feature space represents the representation of the images in the discriminator. Next, the classification loss and the real/fake loss are calculated by comparing the classification results of real and fake images. The classification loss measures the accuracy of the discriminator in distinguishing real and fake images, while the real/fake loss reflects the adversarial training process within the two components. Throughout the training process, the two components engage in a competitive dynamic, continuously optimizing their parameters. The generator's objective is to produce increasingly realistic fake images to deceive the discriminator, while the discriminator aims to distinguish between real and fake images, improving its accuracy. The algorithm flow is shown in Fig. 3.

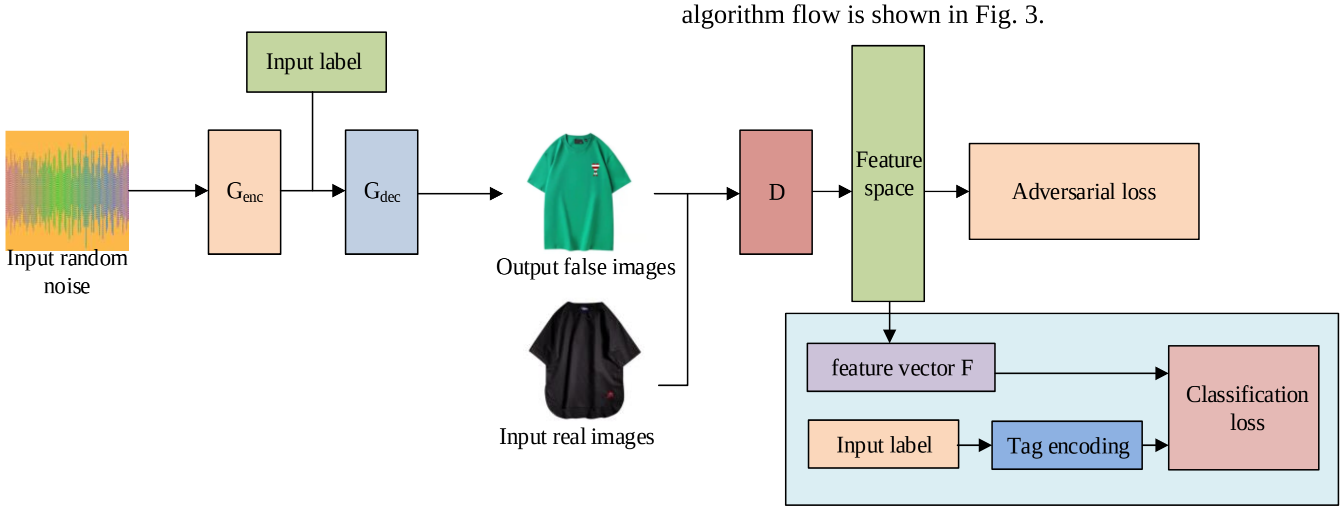


Fig. 2. Improved DCGAN network training stage structure diagram.

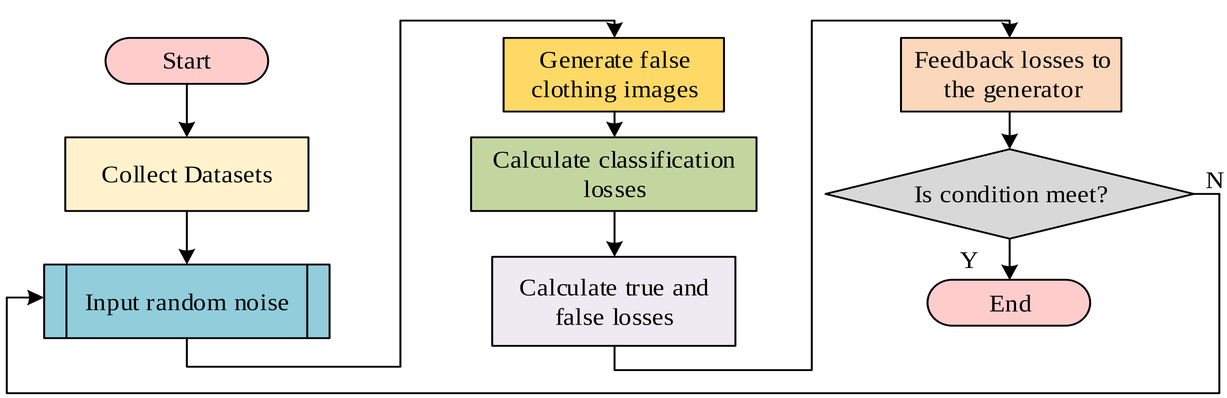


Fig. 3. IDCGN algorithm flowchart.

Fig. 3 shows the flowchart of the IDCGAN algorithm. Firstly, a dataset is collected for model training. Next, random noise is input into the encoding network of the generator to extract feature tensors. The third step is to input the extracted feature tensors and the specified style to be transformed into the decoding network of the generator, generating fake clothing images. Then, the discriminator is trained to improve its discriminative ability to distinguish the true and fake images. At the same time, the classification loss is calculated based on the feature space Z to measure the performance of the generator in generating different categories of clothing. The sixth step is to calculate the real/fake loss based on the feature space Z, which helps the generator generate more realistic clothing images. Next, the loss is fed back to the generator to adjust its strategy for generating images. Finally, steps two to seven are repeated in a loop until the total loss of the network converges, achieving the desired training effect. Through this iterative process, the IDCGAN algorithm continuously optimizes the balance between the generator and the discriminator, achieving better quality in generating fake images. The loss function used in the training process is the conditional contrastive loss. To further explain this loss function, it is necessary to first explain the NT-Xent loss function, which is expressed as Eq. (1).

$$A = \{x_1, T(x_1), ..., x_m, T(x_m)\} = \{a_1, a_2, ..., a_{2m}\} \quad (1)$$

In equation (1), $T(x_m)$ represents the random data augmentation for this loss function. After some transformations, the expression of the NT-Xent loss function is shown in Eq. (2).

$$\delta(a_i, a_j, t) = -\log\left(\frac{\exp(l(a_i)^T l(a_j)/t)}{\sum_{k=1}^{2m} \exp(l(a_i)^T l(a_j)/t)}\right) \quad (2)$$

In Eq. (2), t is the temperature that controls the push and pull forces. By incorporating the embedding equation into Eq. (2) and (3) is obtained.

$$\delta(x_i, y_j, t) = -\log\left(\frac{\exp(l(x_i)^T l(y_j)/t)}{\exp(l(x_i)^T e(y_j)/t) + \sum_{k=1}^{m} \exp(l(x_i)^T l(x_k)/t)}\right) \quad (3)$$

In equation (3), $e(y)$ represents the embedding equation. By adding the cosine similarity of negative samples in equation (3), the final loss function is shown in Eq. (4).

$$\delta(x_i, y_i, t) = -\log\left(\frac{\exp(l(x_i)^T e(y_i)/t) + \sum_{k=1}^{m} y_k = y_i \cdot \exp(l(x_i)^T l(x_k)/t)}{\exp(l(x_i)^T e(y_i)/t) + \sum_{k=1}^{m} k \neq i \cdot \exp(l(x_i)^T l(x_k)/t)}\right) \quad (4)$$

B. *Intelligent Design Model Based on Real-Time Style Transfer Network*

Real-time style transfer network is a computer vision technique used to transfer the style of an input image to another target style while preserving the content of the input image. Typically, this network combines CNN with methods for image stylization. The goal of real-time style transfer network is to perform style transformation on an image in a short period of time, making it appear as if it was drawn or rendered using the target style. By minimizing a loss function, the real-time style transfer network can generate an output image with the desired target style. The network structure is shown in Fig. 4.

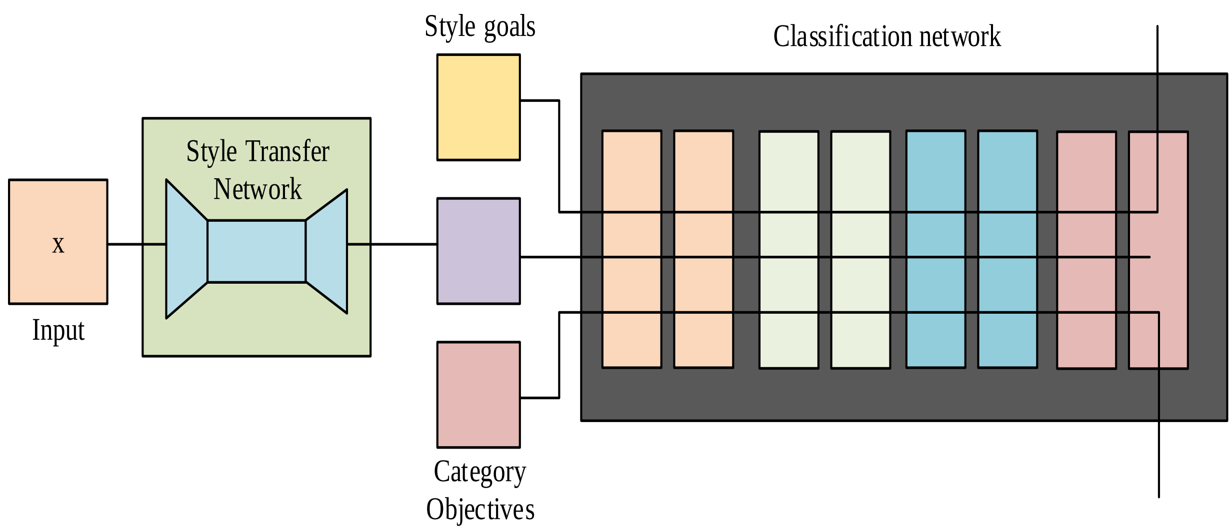


Fig. 4. Network structure of real-time style transfer network.

Fig. 4 illustrates the network structure of the real-time style transfer network. The left half represents the image transformation network, which consists of a series of CNN layers and deconvolution layers. These layers are used to gradually transform the input image into an output image with the target style. Each CNN layer can extract different features from the input image, while the deconvolution layers are used to synthesize these features into the final output image. The right half represents the loss network, which is used to calculate the content loss and style loss. The content loss ensures that the output image preserves the content information of the input image by comparing the feature representations of the input and generated images. The style loss captures the target style features by comparing the feature statistics of the input image, generated image, and target style image. The loss function is shown in Eq. (5).

$$L = \gamma_1 L_1 + \gamma_2 L_2 \quad (5)$$

In Eq. (5), L_1 represents the content loss function, and L_2 represents the style loss function. The formula for the content loss function is shown in Eq. (6).

$$L_1^{\varphi,j}(\hat{y}, y) = \frac{1}{C_j H_j W_j} \| \phi_j(\hat{y}) - \phi_j(y) \|_2^2 \quad (6)$$

In Eq. (6), y represents the original image, and $\hat{y}$ represents the generated image. The formula for the style loss function is shown in Eq. (7).

$$L_2^{\varphi,j}(\hat{y}, y) = \| G_j^{\varphi}(\hat{y}) - G_j^{\varphi}(y) \|_F^2 \quad (7)$$

In Eq. (7), G_j^{φ} represents the Gram matrix, which is a matrix that describes the correlations between features by taking the inner product between different channels in the feature map of the j th layer. The formula for the Gram matrix is shown in Eq. (8).

$$G_j^{\varphi}(x)c, c' = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'} \quad (8)$$

In Eq. (8), ϕ represents the pre-trained network model, and $\phi_j(x)_{h,w,c}$ represents the values of the feature map in the image network with a height of h, width of w, and c channels. The detailed structure size of the generator network is shown in Fig. 5.

Fig. 5 shows a detailed schematic diagram of the network structure size for real-time style transfer. The network includes one reflection padding layer, six convolutional layers, and five residual blocks. With this network, style transfer between images can be achieved, and the time required to generate images is significantly reduced. However, although this network has achieved certain results, there is still room for improvement. For example, there are still areas that can be optimized in terms of image quality, detail preservation, and style restoration. In addition, the current network structure needs to be trained for each specific style, which limits its applicability. To address these issues, some improvements have been made to the network, as shown in Fig. 6.

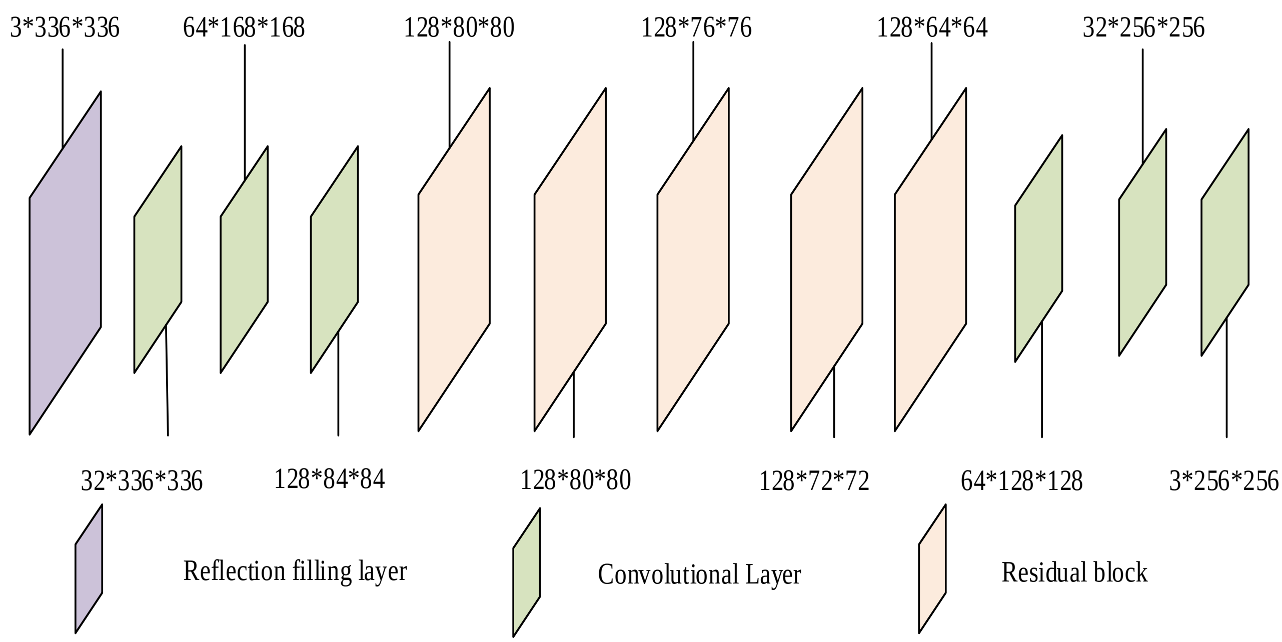


Fig. 5. Generate a detailed schematic diagram of the network structure size.

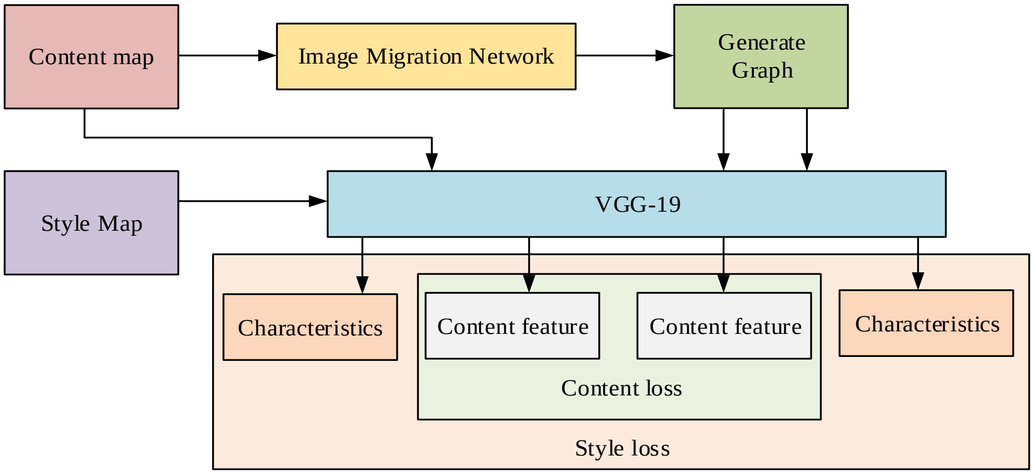


Fig. 6. Improved real-time style migration network structure diagram.

Fig. 6 shows the structure diagram of the improved real-time style transfer network. The network consists of two main components: the upper part and the lower part. In the upper part, the input content image is passed through an image encoder network to generate the generated image. This network can be a CNN that encodes the content image into an initial version of the generated image. The generated image and the style image are then passed through the VGG-19 model for feature extraction [20]. In the lower part, the content loss and style loss are propagated back through the process of backpropagation, and the pixel values of the generated image are updated using the gradient descent optimization algorithm. The optimization objective is to minimize the content loss and style loss, thereby preserving the content and matching the target style in the generated image. Additionally, the research addresses the issue of traditional normalization methods struggling to learn highly nonlinear features by introducing Adaptive Instance Normalization (AdaIN), which is formulated as Eq. (9).

$$Z_{bcwh} = \frac{x_{bcwh}}{\sum_{w=1}^{W}\sum_{h=1}^{H} x_{bcwh}} \quad (9)$$

In Eq. (9), $x \in R^{B\times C\times W\times H}$. W and H represents the width and height of the image, respectively. To allow Eq. (9) to be fitted by the ReLU activation function, a transformation is applied to the equation, as shown in Eq. (10).

$$Z_{bcwh} = \frac{x_{bcwh} - \eta}{\sqrt{\sigma^2 + \varepsilon}} \quad (10)$$

In equation (10), b represents the image index in the batch, σ^2 represents the variance, and ε represents the mean. The calculation of the variance is approximated as shown in Eq. (11).

$$\sigma_{bc}^2 = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(x_{bcwh} - h\eta_{bc}\right)^2 \quad (11)$$

In Eq. (11), c represents the number of channels in the image. Compared to traditional instance normalization (IN), AdaIN only requires one forward pass, as shown in Eq. (12).

$$Z = \sqrt{\sigma^2 + \varepsilon} \cdot \frac{x - \eta_x}{\sqrt{\sigma_x^2 + \varepsilon}} + \eta_y \quad (12)$$

In Eq. (12), y represents the input values for style, and x represents the input values for content. In addition, the style loss needs to be optimized by removing the Gram matrix from the loss function of AdaIN, as shown in Eq. (13).

$$L_2^{\varphi,j}(\hat{y}, y) = \| \eta\phi_j^{\varphi}(\hat{y}) - \eta\phi_j^{\varphi}(y) \|_2 + \| \lambda\phi_j^{\varphi}(\hat{y}) - \lambda\phi_j^{\varphi}(y) \|_2 \quad (13)$$

In Eq. (13), $\|.\|_2$ represents the L2 norm.

IV. Performance Testing and Application Analysis of Pattern Intelligent Design System

This chapter is divided into two sections. The first section mainly verifies the improvement effect of the IDCGAN algorithm by comparing it with the standard DCGAN algorithm. The second section focuses on the application analysis of the pattern intelligent design system and its application in practical clothing design.

C. Comparative Experiment of IDCGAN Algorithm

In order to address some limitations of the standard DCGAN algorithm in the field of clothing design, the study made a series of improvements and finally formed the IDCGAN algorithm. To verify whether this improved algorithm is superior compared with standard DCGAN algorithm, the study used PyTorch 1.4 software on Ubuntu64-bit platform, the learning rate starting value can be set to 0.0002, batch size using batch size such as 16,32 or 64, noise dimension set to 100, loss function as binary cross-entropy loss function, IDCGAN and DCGAN using the CV-PTON dataset for 40,000 iterations. The PR curves were used as an assessment criterion. The results are shown in Fig. 7.

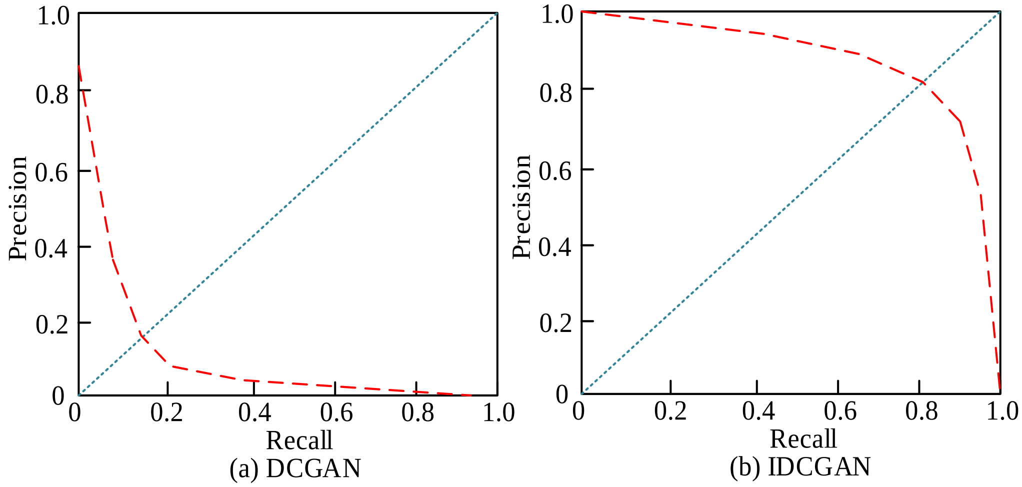


Fig. 7. Comparison of PR curves before and after DCGAN improvement.

Fig. 7 shows the changes in the PR curves before and after the improvement of the DCGAN algorithm. From Fig. 7(a), after the model convergence, the recall rate of the DCGAN algorithm has improved, but the precision has significantly decreased. This indicates that the original DCGAN algorithm may have some noise or errors in generating samples. From Fig. 7(b), it can be seen that by using a deep feature extraction network to improve the algorithm, the feature extraction capability of DCGAN is significantly enhanced. This allows the improved algorithm to generate samples with better style transfer effects while maintaining high recall rate and precision. To verify the improvement effect of the loss function comparison in this study, a similarity heatmap of the IDCGAN during the experimental process was visualized, as shown in Fig. 8.

Fig. 8 shows the similarity heatmap of IDCGAN. By observing the results in the figure, it can be seen that the contrastive loss used in this algorithm is very effective in distinguishing between input and generated patterns of the same category, with a similarity score of 1. This means that the improved algorithm can accurately identify and generate samples that are similar to the input pattern. Additionally, for different categories of style patterns, the similarity score is close to 0. This indicates that the improved algorithm can differentiate between samples of different categories and will not mistake them for similar patterns. Furthermore, a series of experiments were conducted to evaluate the resource consumption of IDCGAN and DCGAN, and the results are shown in Fig. 9.

Based on the results in Fig. 9, it can be observed that IDCGAN and DCGAN differ in terms of CPU resources, memory resources, and training time. By comparing Fig. 9(a) and Fig. 9(b), IDCGAN reduces the physical memory usage by 45.7% compared to DCGAN, with only 453MB compared to DCGAN's 834MB. Additionally, in terms of CPU resource utilization, IDCGAN only utilizes 26% of the CPU resources, indicating its relatively low computational demand. This is advantageous for devices or environments with limited resources. Furthermore, it is worth noting that the training time of IDCGAN is approximately 20 minutes and 48 seconds, while DCGAN takes about 34 minutes and 49 seconds. This comparison shows that the training time of IDCGAN is reduced by approximately 70%. This means that IDCGAN is more efficient in terms of training speed and can complete training tasks faster.

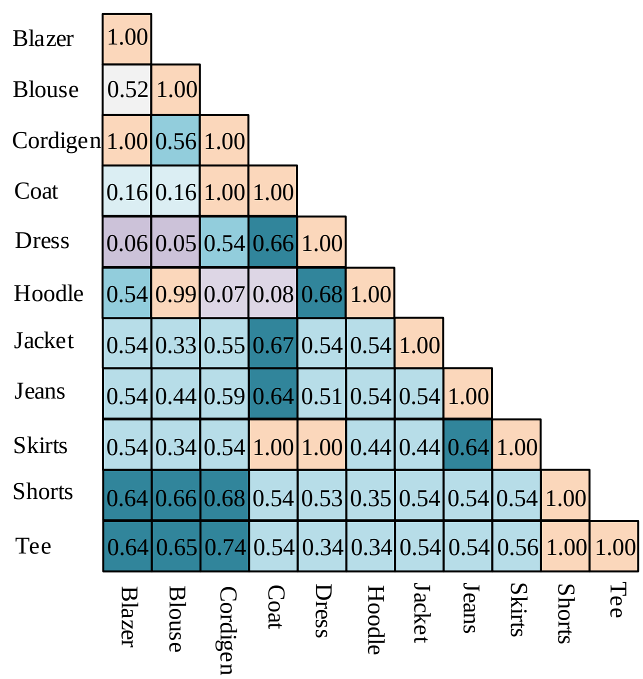
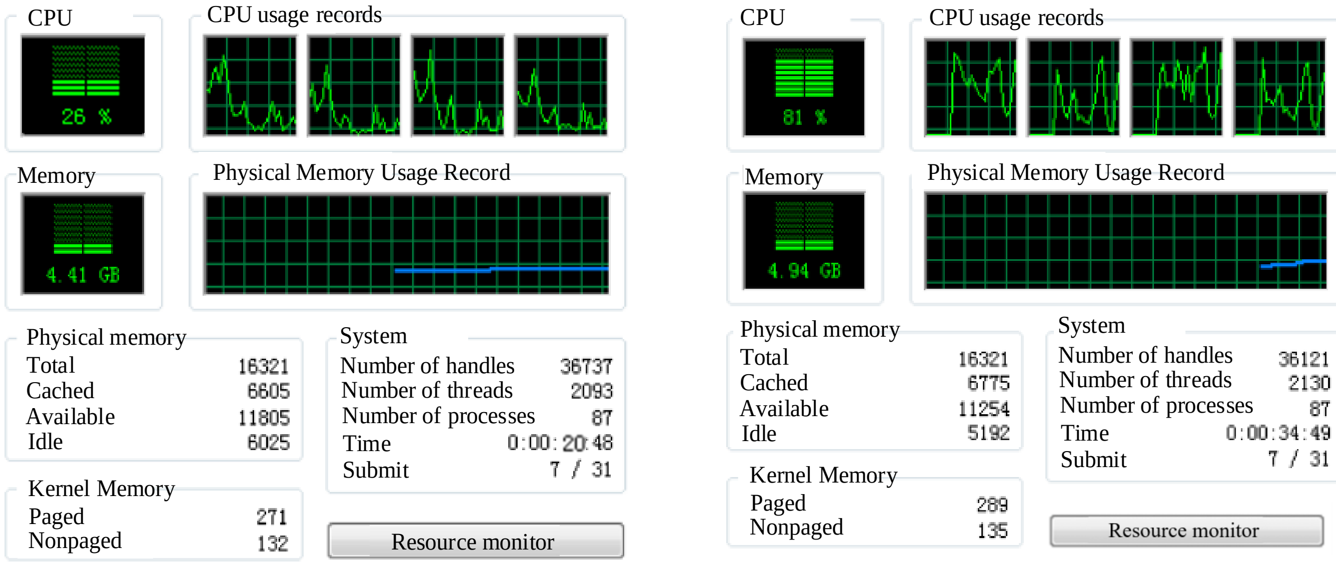


Fig. 8. Improving DCGAN similarity thermal map.



(a) IDCGAN (b) DCGAN

Fig. 9. Resource consumption of DCGAN and IDCGAN in various aspects.

D. Application Analysis of Style Transfer Network and Intelligent Design System

To address the limitations of real-time style transfer networks in terms of generated image quality, detail preservation, and style restoration, an improved real-time style transfer network was designed. The experiment first verified the optimization effect of the style transfer network based on AdaIN. For this purpose, batch normalization (BN) and instance normalization (IN) were introduced as control groups. The experiment was conducted using PyTorch 1.8 software on the Windows 10 platform, and the three models were trained for 62,500 iterations each. The results are shown in Fig. 10.

Fig. 10 displays the correlation of the loss and the iterations in the training based on three different normalization methods. From the figure, the loss values of the three models rapidly decrease in the first 5,000 iterations, then stabilize in the range of 50,000 to 45,000 iterations, and then decrease rapidly again. The final loss values of the models based on BN, IN, and AdaIN normalization converge to 1.02, 0.83, and 0.52, respectively. The real-time style transfer network based on AdaIN achieves the lowest loss value. To analyze the application of the proposed improved transfer model in this experiment, the trained model was applied to actual clothing design and evaluated by 36 professional fashion designers. Aesthetic quality scores and visual realism were used as evaluation criteria. The experimental results are shown in Fig. 11.

Fig. 11 provides a detailed display of the evaluation results of the images generated on AdaIN by 36 professional fashion designers. This chart clearly reflects that the proposed style transfer network has received widespread acclaim among the fashion designer community. Designers evaluated the images generated by the model rigorously and comprehensively from their professional perspectives. The results show that the 36 designers gave high aesthetic quality ratings to the images, with an average score of 96, indicating the excellent performance of the model in terms of aesthetic representation. Additionally, the designers highly recognized the visual realism of the images generated by the model, with an average score of 94.7. This score demonstrates the model's ability to successfully transfer the target style while preserving the original image content. To validate the superiority of the proposed intelligent design system in this study, comparative experiments were conducted with classical style transfer networks and standard real-time style transfer networks, as Fig. 12.

Fig. 12 displays the application effects of the three different style transfer networks. By observing Fig. 12(a), the images generated by the classical style transfer network are relatively blurry and lack detail. However, Fig. 12(b) shows that the real-time style transfer network has made significant progress compared to the classical style transfer network, but there is still room for improvement in terms of detail. In contrast, Fig. 12(c) demonstrates that the style transfer network proposed in this study preserves more details and generates style transfer results that are clearer. In conclusion, the style transfer network in this study has achieved significant improvements in image quality and detail expression.

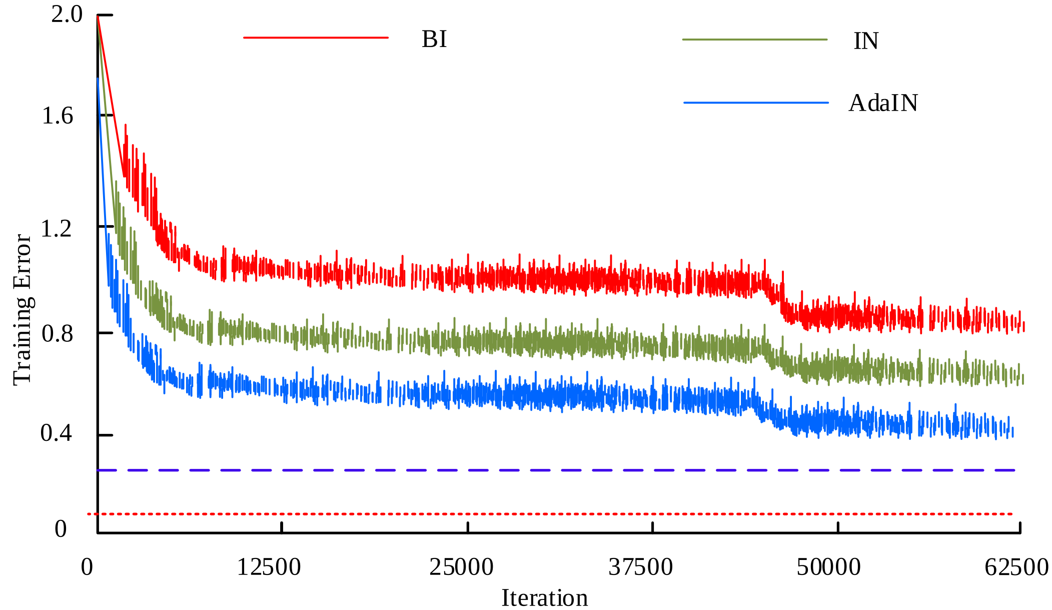


Fig. 10. Loss value variation curves of three models.

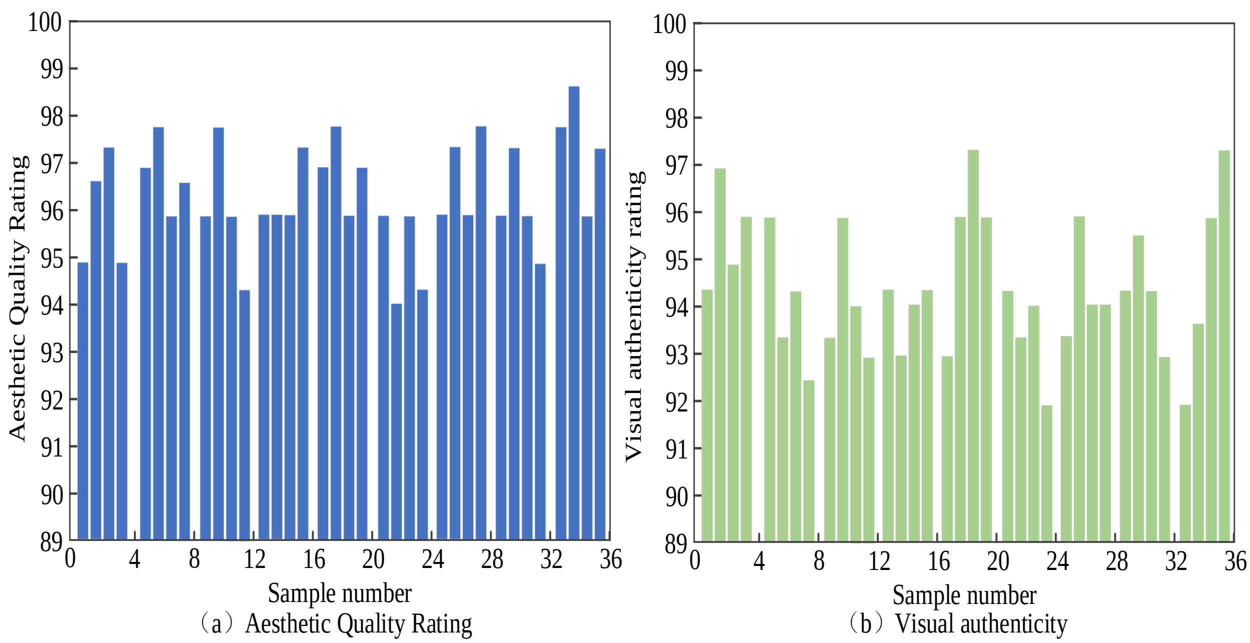
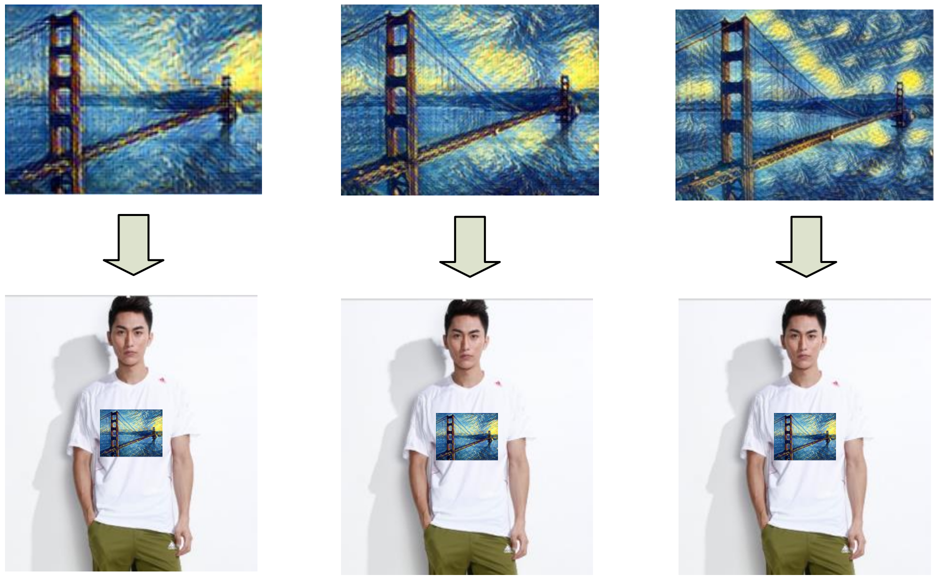


Fig. 11. Model aesthetic quality rating and visual authenticity rating.



(a) Classic style transfer

(b) Real time style migration

(c) Style transfer in this article

Fig. 12. Three styles of transfer application effects.

V. Conclusion

To address the issues of long manual design pattern creation time and low flexibility, an intelligent image design model based on the IDCGAN real-time style transfer algorithm was designed. IDCGAN The innovative content of the algorithm mainly includes the introduction of encoder and decoder network, the introduction of style vector, the method of multi-scale feature extraction and fusion, the use of conditional constraints and multi-task learning. These innovations enable the IDCGAN algorithm to generate more realistic and constrained fake images, and improve the efficiency and effect of the algorithm. For actually verifying the superiority of this improved algorithm compared to the standard DCGAN, experiments were conducted using PyTorch 1.4 software on a 64-bit Ubuntu system platform for 40,000 iterations. The results showed that after the model iterations converged, the recall rate of the DCGAN algorithm improved, but the precision significantly decreased. This indicates that the original DCGAN algorithm may have some noise or errors when generating samples. However, by using a deep feature extraction network to improve the algorithm, IDCGAN significantly enhanced the feature extraction capability of DCGAN. Additionally, the final loss values of the models based on BN, IN, and AdaIN normalization converged to 1.02, 0.83, and 0.52, respectively. The real-time style transfer network based on AdaIN achieved the lowest loss value. By comparison, IDCGAN can reduce physical memory usage by 45.7% to only 453MB, compared with DCGAN by 834MB. Moreover, in terms of CPU resource utilization, IDCGAN occupies only about 26% of the CPU resources, showing relatively low computational requirements. This is a very important advantage for scenarios where devices or environments are limited. Thus, this result indicates that IDCGAN not only generates more realistic false images, but also improves the efficiency of the algorithm. Finally, the proposed improved transfer model was subjected to application analysis, and the outcomes tells that the images generated by the classical style transfer network were relatively blurry and lacked detail. Furthermore, the aesthetic quality ratings given by the 36 designers were high, with an average score of 96, indicating that the model was well-received by the designers. On the other hand, the style transfer network proposed in this study preserved more details and generated clearer style transfer results. However, it should be noted that the model still required at least 45,000 iterations to stabilize during training, which is an aspect that needs improvement in future research. Future research will try to apply this method to more areas of image generation, such as art creation, interior design, game development, film and television special effects, etc.

References

[1] Oslund S, Washington C, So A. Multiview Robust Adversarial Stickers for Arbitrary Objects in the Physical World. Journal of Computational and Cognitive Engineering, 2022, 1(4): 152-158.

[2] Wang X, Cheng M, Eaton J. Fake node attacks on graph convolutional networks. Journal of Computational and Cognitive Engineering, 2022, 1(4): 165-173.

[3] Nimrah S, Saifullah S. Context-Free Word Importance Scores for Attacking Neural Networks. Journal of Computational and Cognitive Engineering, 2022, 1(4): 187-192

[4] Ding H, Cui Z, Maghami E, Chen, Y., Matinlinna, J. P., Pow, E. H. N., ... & Tsoi, J. K. H. Morphology and mechanical performance of dental crown designed by 3D-DCGAN. Dental Materials, 2023, 39(3): 320-332.

[5] Abd Al Rahman M, Danishvar S, Mousavi A. An improved capsule network (WaferCaps) for wafer bin map classification based on DCGAN data upsampling. IEEE Transactions on Semiconductor Manufacturing, 2021, 35(1): 50-59.

[6] Bian Y, Wang J, Jun J J,, & Xie, X. Q. Deep convolutional generative adversarial network (dcGAN) models for screening and design of small

molecules targeting cannabinoid receptors. Molecular pharmaceutics, 2019, 16(11): 4451-4460.

[7] Cheng K, Zhu L, Yao C,Yu, L., Wu, X., Zheng, X., ... & Lin, F.. DCGAN based spectrum sensing data enhancement for behavior recognition in self-organized communication network. China Communications, 2021, 18(11): 182-196.

[8] Li M, Tang H, Chan M D,Zhou, X., & Qian, X. DC-AL GAN: Pseudoprogression and true tumor progression of glioblastoma multiform image classification based on DCGAN and AlexNet. Medical physics, 2020, 47(3): 1139-1150.

[9] Ni J, Liu B, Li J, Gao, J., Yang, H., & Han, Z. Detection of carrot quality using DCGAN and deep network with squeeze-and-excitation. Food Analytical Methods, 2022, 15(5): 1432-1444.

[10] Jing Y, Liu X, Ding Y, Wang, X., Ding, E., Song, M., & Wen, S. Dynamic instance normalization for arbitrary style transfer[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(4): 4369-4376.

[11] Reimann M, Buchheim B, Semmo A,Döllner, J., & Trapp, M. Controlling strokes in fast neural style transfer using content transforms. The Visual Computer, 2022, 38(12): 4019-4033.

[12] Hollandi R, Szkalisity A, Toth T,Tasnadi, E., Molnar, C., Mathe, B., ... & Horvath, P. nucleAIzer: a parameter-free deep learning framework for nucleus segmentation using image style transfer. Cell Systems, 2020, 10(5): 453-458.

[13] Huang Z, Zhang J, Liao J. Style Mixer: Semantic-aware Multi-Style Transfer Network.Computer Graphics Forum. 2019, 38(7): 469-480.

[14] Xu K, Wang S, Jin Y, Che, Q., & Zhou, B. Object detection-oriented style transfer network for panchromatic remote sensing image. Journal of Applied Remote Sensing, 2023, 17(2): 026503-026503.

[15] Zhou L, Zhang T. AttCST: attention improves style transfer via contrastive learning. Journal of Electronic Imaging, 2023, 32(3): 033018-033018.

[16] Zhang L, Duan L, Hong X,Liu, X., & Zhang, X. Imbalanced data enhancement method based on improved DCGAN and its application. Journal of Intelligent & Fuzzy Systems, 2021, 41(2): 3485-3498.

[17] Zhang F, Wang X, Sun T,Xu, X. SE-DCGAN: a new method of semantic image restoration. Cognitive Computation, 2021, 13(4): 981-991.

[18] Li Q, Qu H, Liu Z, N., Sun, W., Sigg, S., & Li, J. AF-DCGAN: Amplitude feature deep convolutional GAN for fingerprint construction in indoor localization systems. IEEE Transactions on Emerging Topics in Computational Intelligence, 2019, 5(3): 468-480.

[19] Cho J, Moon N. Design of image generation system for DCGAN-based kids' book text . Journal of Information Processing Systems, 2020, 16(6): 1437-1446.

[20] Ikechukwu A V, Murali S, Deepu R, et al. ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images. Global Transitions Proceedings, 2021, 2(2): 375-381.