International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 2773-2784

https://doi.org/10.70517/ijhsa464230

Optimization of Recursive Algorithms in Artwork Generation and Modeling Pathways for Artistic Style Transfer

Yi Zhang[1] and Zhengyang Zhang[2,*]
[1] Department of Fine Arts, Moscow Institute of Art, Weinan Normal University, Weinan, Shaanxi, 714099, China
[2] Department of Environmental Art and Design, Hebei Art and Design Academy, Shijiazhuang, Hebei, 050034, China
Corresponding authors: (e-mail: 19829263779@163.com).

Abstract With the rapid development of technology and media in recent decades, the issue of art work generation has attracted significant attention from the academic community. To address the optimization of recursive algorithms in art work generation, a multi-scale generative adversarial network (PRMGAN) based on progressive recursion has been designed, along with the corresponding loss function. Building on this, the DualGAN network is employed to model artistic style transfer in art work generation. Finally, the model proposed in this paper is applied to conduct an in-depth analysis of art generation and artistic style transfer. Under the influence of PRMGAN and DualGAN, the metric values of the generated artworks have improved, but the effects are not significant, with corresponding SSIM and PSNR values ranging from [0.1 to 0.9] and [7 to 28], respectively. However, under the combined influence of the two, the metric values of the generated artworks have improved significantly, with SSIM and PSNR values ranging from [0.1 to 0.9] and [7 to 28], respectively. This study not only advances the field of art generation but also provides theoretical references for art style transfer modeling pathways.

Index Terms Recursive algorithm, Style transfer model, Generative adversarial network, Art generation

I. Introduction

With the development of artificial intelligence, the generation of fine art works has become an important trend in artistic creation. There are now many useful AI painting tools available, such as StableDiffusion and Midjourney [1]-[3]. These tools act like magical brushes, capable of generating a wide variety of fine art works based on the instructions input by the creator [4], [5]. However, since these are intelligently generated works, they may not fully meet the creator's requirements in terms of color, composition, and emotional expression. In such cases, optimization and style transfer in art generation have gradually gained application [6]-[8].

The primary objectives of art generation optimization are: First, to enrich the details of the artwork, such as the expressions, hairstyles, clothing styles, and colors of characters, as well as the colors of the sky, the shapes of clouds, and the types of flowers, plants, and trees [9]-[11]. Second, it is to determine style preferences. AI painting can generate various styles, such as realistic, cartoon, and abstract styles, and creators need to determine which style they prefer [12], [13]. Third, it is to enhance the visual effects of artistic works. This can be achieved by blending the style of an artistic work with that of other artistic works to enhance its visual effects [14], [15]. These optimization effects can all be achieved through intelligent algorithms. Art style transfer in generating artistic works involves applying one artistic style to another image, endowing an ordinary image with the unique charm of a specific artistic style [16]-[18]. For example, transferring the passionate and fantastical style of Van Gogh's paintings to a landscape photo taken in daily life, making the photo exhibit the unique aesthetic appeal reminiscent of Van Gogh's works [19]-[21]. Optimizing art generation and artistic style transfer not only expands the possibilities of artistic creation but also inspires creators with more creative ideas [22]-[24].

Given the suboptimal performance of recursive algorithms in art generation, a progressive recursive multi-scale generative adversarial network (PRMGAN) was constructed, and the structure of each module in the network was deeply analyzed, with corresponding loss functions determined. To ensure that art generation meets research requirements, the DualGAN network is employed on the basis of the Progressive Recursive Multi-Scale Generative Adversarial Network (PRMGAN) to model artistic style transfer in art generation. Finally, the proposed model is used to analyze art generation, aiming to provide guidance for algorithm optimization and style transfer modeling strategies in art generation.

II. Exploration of Artwork Creation

II. A.Recursive Algorithms

II. A. 1) Concept of Recursive Algorithms

A recursive algorithm refers to an algorithm that can directly or indirectly call itself, breaking down a complex problem into smaller subproblems. During execution, the algorithm repeatedly calls itself in an iterative process [25], [26]. During the recursive calling process, the algorithm only exits the recursive loop when it reaches the critical value of the recursive boundary. The recursive algorithm then returns in reverse order from back to front, meaning that the value of each recursive call is placed into a stack. When the algorithm reaches the exit of the recursive process, it performs a pop operation, returning the values from the last recursive call in reverse order until it reaches the first push operation. This completes the entire recursive algorithm operation.

II. A. 2) Characteristics of Recursive Algorithms

Recursive algorithms are an extremely important method in program design, helping to solve many complex problems. Recursive algorithms greatly simplify the amount of work involved in program development, making the program structure clearer and more readable, while also making the problem easier to understand and accept. Recursive algorithms have the following three main characteristics:

(1) Designing recursive formulas

Recursive algorithms directly or indirectly call themselves, breaking down a more complex problem into one or more subproblems for solution. The subproblems share the same algorithm as the original problem. For example, to recursively calculate the sum of n elements, we can use the formula $f(n)=n+f(n-1)$ for recursive iteration. Another classic example is the Tower of Hanoi problem, where column c serves as the support column. The formula for moving $n-1$ disks from column a to column b is $f(n-1,a,c,b)$, and the moved disks are printed as $move(n,a,c)$. Using column a as the support column, the formula for moving the disks from column b to column c is $f(n-1,b,a,c)$. Each time the recursion is iterated, the problem size decreases, and the output from the previous iteration becomes the input for the next iteration.

(2) Boundary conditions for recursive algorithms

Boundary conditions are the exit points of recursion. Recursive algorithms are not infinite; when the program reaches the last level of recursion, it must return the output. Since each level of recursion reduces the problem size, each recursion step brings the algorithm closer to the termination condition until the termination condition is met, at which point the critical value is returned. For example, in the recursive factorial problem $f(n)=n\cdot f(n-1)$, the termination condition occurs when $n=1$. Without a recursive boundary condition, the program would enter an infinite loop and produce no output. Sometimes, a recursive algorithm has more than one boundary condition. In the integer partition problem, we need to separately discuss the relationship between maximizing the fraction value and the size of the partitioned integer value. In this case, the recursive boundary conditions are also multiple.

(3) Reducing the scale of the problem

During the recursive process, the scale of the problem must be continuously reduced. Since the recursive algorithm directly or indirectly calls itself, during parameter passing in the recursion process, the parameter values must be gradually reduced or increased to approach the boundary conditions of the recursive algorithm. Controlling the recursion direction is essential to controlling the recursive algorithm. For example, in the Fibonacci sequence, $F(0)=0,F(1)=1,F(n)=F(n-1)+F(n-2)$, the problem size is continuously reduced during each recursive call until the problem size is reduced to the recursive exit point, at which point the recursive output condition is met.

II. B.Recursive Algorithm Optimization in Artwork Generation

To ensure the stability and effectiveness of recursive algorithms in the generation of fine art works, the network capacities of the generator and discriminator must be matched. If the generator is too powerful while the discriminator is weak, the generator may overfit the data distribution, leading to unstable training. Conversely, if the discriminator is too powerful while the generator is weak, the discriminator may easily distinguish generated samples, preventing the generator from learning effectively. Therefore, matching the network capacities of the generator and discriminator is crucial. To address this, we propose a Progressive Recursive Multi-Scale Generative Adversarial Network (PRMGAN).

II. B. 1) Asymptotic Recursion Module

Simple recursive algorithms are often ineffective for generating artwork. The progressive recursive module (PRB) proposed in this paper shares the same convolution and LSTM parameters and performs internal loop calculations, enabling it to generate artwork effectively without increasing network parameters. The progressive recursive module is a residual connection block consisting of two main parts: a multi-scale convolution module and an LSTM module.

Its formula is expressed as:

$$O = X + \sum_{i=1}^{N} (\text{ReLU}(\text{Normalize}[P(C) + P]) + L_{\text{prev}}) \tag{1}$$

In the formula: X is the input feature map. O is the output feature map. N is the number of iterations, i.e., the number of times the progressive recursive module is stacked. In this algorithm, N = 4. C is the convolution operation. P is the padding operation. L_{prev} is the output feature map of the previous iteration of LSTM. C and P are specifically represented as:

$$\begin{cases} C = C_3(P_1(O)) \\ P = \sum_{d \in 1,2,3} P(C_d(P_d(O))) \end{cases} \tag{2}$$

In the recursive module construction algorithm, X and O are the input and output feature maps, respectively, L_{prev} is the LSTM output feature map from the previous iteration, C_n denotes a convolution with a convolution sum of $n \times n$, and C_d denotes a dilated convolution with a dilation rate of d. The construction algorithm consists of three steps: initialization, iteration, and residual connection. In the initialization step, the input feature map X is assigned to the output feature map O, and the LSTM output map L_{prev} is set to empty.

Using different hole rates C_d (where d takes values of 1, 2, or 3), the input feature map O is convolved with the result B and added to it, yielding the updated B. Finally, B is normalized and passed through a ReLU activation function. In the shape transformation step, the output B is shape-transformed, concatenated with L_{prev}, and then processed through the LSTM module to obtain the updated B and the current iteration step's LSTM output L, and L_{prev} is updated to L. The shape transformation step is performed again, and the output feature map B is shape-transformed. Finally, in the residual connection step, the original feature map X is added to the output feature map O to obtain the final output feature map O.

II. B. 2) Multi-scale feature discriminator

In recursive algorithm improvements, the introduction of multi-scale feature modules allows for simultaneous attention to local details and global context, thereby generating higher-quality artwork. For:

$$\begin{aligned} F_1^{1\times1} &= Conv_{1\times1}(F_{in}; \theta_1^{1\times1}) \\ F_2^{3\times3} &= Conv_{3\times3}(F_{in}; \theta_2^{3\times3}) \\ F_3^{3\times3} &= Conv_{3\times3}(F_2^{3\times3}; \theta_3^{3\times3}) \\ F_4^{3\times3} &= Conv_{3\times3}((F_1^{1\times1} + F_3^{3\times3}); \theta_4^{3\times3}) \end{aligned} \tag{3}$$

In the equation: $F_a^{\beta\times\beta}$ denotes the α th convolution with a convolution kernel of size $\beta \times \beta$. $Conv_{n\times n}(\cdot)$ denotes the convolution operation. $\theta_a^{\beta\times\beta}$ denotes the hyperparameter formed by the α th convolution with a convolution kernel of size $\beta \times \beta$. In Conv4, the first convolution kernel performs a 1×1 and 3×3 tensor concatenation fusion. It is defined as:

$$\begin{aligned} F_5^{3\times3} &= Conv_{3\times3}(F_4^{3\times3}; \theta_5^{3\times3}) \\ F_6^{3\times3} &= Conv_{3\times3}((F_1^{1\times1} + F_3^{3\times3} + F_5^{3\times3}); \theta_6^{3\times3}) \end{aligned} \tag{4}$$

Conv6 performed a second convolution kernel 1×1 and 3×3 tensor concatenation fusion. Feature information was further fused. For:

$$F_7^{5\times5} = Conv_{5\times5}(F_6^{3\times3}; \theta_7^{5\times5}) \tag{5}$$

In Conv7, independent convolution operations are performed, introducing convolution kernels with larger receptive fields to capture broader contextual information and further extract features.

II. B. 3) Artwork Models

The models of the works of art cited in this article are as follows:

$$I = B + R \tag{6}$$

In the formula: I is the art content image. B is the background image. R is the texture. Subtracting the texture R from the art content image I yields the background image B. The model in this paper is decomposed in detail as follows:

$$I = B_{real} + R_{real} \tag{7}$$

$$\hat{R} = G_{\theta_\omega}(I) = \{Conv, Conv, Conv, PRB, ConvTranspose, ConvTranspose, Conv, tanh\} \tag{8}$$

$$B_{gen} = I - \hat{R} \tag{9}$$

$$D(B_{real}) = \sum_{a \in B_{n+d}} D_{real}(\alpha) = \{Conv, Conv, Conv, MFB, Conv, Conv, Sigmoid\} \tag{10}$$

$$D_{\theta_o}(B_{gen}) = \sum_{\beta \in \hat{B}} D_{\theta_o gen}(\beta) = \{Conv, Conv, Conv, MFB, Conv, Conv, Sigmoid\} \tag{11}$$

In the equation: $\hat{R}$ is the estimated texture. B_{gen} is the background image generated by the generator. $G_{\theta\omega}$ is the generator. $D_{\theta\upsilon}$ is the discriminator. Through the generator, the model learns how to remove textures. The discriminator evaluates both the real background image and the estimated background image, outputting the evaluation results.

II. B. 4) Loss Function

This paper selects a combination of content loss and adversarial loss as the loss function:

$$L_{total} = \alpha L_C + \beta L_{adv} \tag{12}$$

In the formula: L_{bul} is the total loss. α, β are weight parameters. References and experiments show that $\alpha = 100, \beta = 10$ is selected in this paper. Content loss L_c and adversarial loss L_{adv} can be expressed as:

$$L_c = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} l(\phi_{i,j}(I^N)_{x,y} - \phi_{i,j}(G_{\theta_a}(I^R))_{x,y})^2 \tag{13}$$

In the formula: $\phi_{i,j}$ is the feature map output by the i th block and j th layer in VGG19. I^N is the background image. I^R is the art image. G_θ is the generator.

$$L_{adv} = \sum_{n=1}^{N} -D_{\theta_n}(G_{\theta_n}(I^R)) \tag{14}$$

In the formula: I^R is the fine art image. $G_{\theta\omega}$ is the generator. $D_{\theta\omega}$ is the discriminator. These two loss functions are typically used together in generative adversarial networks to achieve better fine art generation results.

II. C.Art Style Transfer Modeling

As mentioned above, multi-scale generative adversarial networks based on progressive recursion can be used to generate works of art. In order to ensure that the generated works of art meet the needs of users in different fields, an art style transfer model was constructed based on this. Details are as follows:

II. C. 1) Style transfer model based on model iteration

DualGAN Model: DualGAN is a generative adversarial network model that uses dual learning modes to perform

image-to-image conversion. Most methods for converting between images require paired images. However, in certain real-world scenarios, it is difficult to obtain such paired images. The DualGAN model combines generative adversarial networks with multi-task learning. The training process is shown in Figure 1.

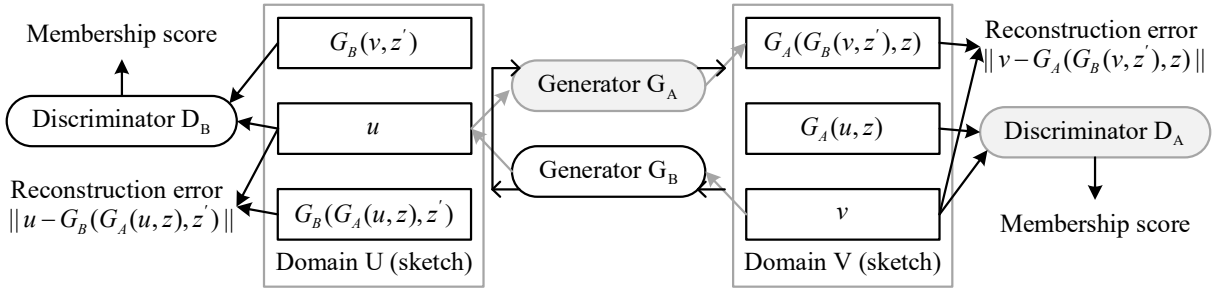


Figure 1: Dual GAN structure diagram

In Figure 1, U and V are the two image domains, and u and v are two data domain samples, respectively, then the two image domains are the target domain and the source domain of each other. The purpose of DualGAN is to train two generators, G_A and G_B. G_A can migrate data samples from the U domain to the generative network of the V domain, and G_B does the opposite of G_A. z and z' represent the noise in both generators. The discriminator D_A is the discriminant network of the U domain, and the discriminator D_B is the discriminant network of the V domain. When converting from the U domain to the V domain, the data u is converted into G_A by $G_A(u,z)$ and the real v input discriminator D_B in the V domain to obtain the discriminant loss. $G_B(G_A(u,z),z')$ is then converted back to the U field through G_B, and the result is $G_B(G_A(u,z),z')$, and the image $G_B(G_A(u,z),z')$ is similar to the image u in the original U field, so that U The data in the domain completes a loop. When converting from a V domain to a U domain, the process is the same as above. The content reconstruction loss of u in a loop is Eq. (15):

$$\| G_B(G_A(u,z),z') - u \| \tag{15}$$

Similarly, the content reconstruction loss of v is given by equation (16):

$$\| G_A(G_B(v,z'),z) - v \| \tag{16}$$

The loss function of discriminator D_A is constructed based on the judgment values of discriminator D_A for generated samples and true samples, as shown in Equation (17):

$$L_A^d(u,v) = D_A(G_A(u,v)) - D_A(v) \tag{17}$$

Similarly, we can derive the loss function of the discriminator D_B as equation (18):

$$L_B^d(u,v) = D_B(G_B(u,v')) - D_B(v) \tag{18}$$

Reconstruct the overall loss of the model based on content loss and discriminator loss:

$$L\mathbf{g} = \lambda_u \| G_B(G_A(u,z),z') - u \| + \lambda_v \| G_A(G_B(v,z'),z) - v \| + L_A + L_B \tag{19}$$

II. C. 2) Evaluation Indicators

This paper evaluates the differences between images using peak signal-to-noise ratio (PSNR) and characterizes the structural loss between the generated image and the original image using structural similarity [27]. The specific description is as follows:

(1) Peak Signal-to-Noise Ratio (PSNR)

The peak signal-to-noise ratio (PSNR) is the logarithm of the mean squared error between the original image and the new image relative to $(2^n - 1)^2$. Before calculating the PSNR, the formula for the mean squared error is given in Equation (20):

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[I(i,j)-k(i,j)]^2 \tag{20}$$

where i and k represent two images of size $m \times n$, the PSNR value can be obtained using equation (21):

$$PSNR = 10\log_{10}(\frac{MAX_i^2}{MSE}) \tag{21}$$

MAX represents the maximum pixel value of the image color. The higher the PSNR value, the lower the noise in the generated image, and the better the image quality.

(2) Structural Similarity

In the generation of artistic works, there is a strong correlation between images, and there is also a strong connection between pixels within the same image. By measuring various types of information within the image (such as grayscale range), we can determine the degree of similarity between two images, known as structural similarity (SSIM). Structural similarity is composed of three comparison modules: brightness, contrast, and structure. The average grayscale value of an image is used as its brightness, so the brightness calculation formula is given by Equation (22):

$$ux = \frac{1}{H \times M}\sum_{i=1}^{H}\sum_{j=1}^{M}X(i,j) \tag{22}$$

Construct a brightness contrast function based on the brightness of the two images. As can be seen from equation (23), the brightness contrast function takes values in the range (0, 1):

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{23}$$

When constructing the contrast function, it is necessary to calculate the average gray value of the image and then calculate the standard deviation of the image. The formula for constructing the standard deviation is shown in (24):

$$\sigma x = (\frac{1}{H+W-1}\sum_{i=1}^{n}\sum_{j=1}^{M}(X(i,j)-\mu x)^{\frac{1}{2}})^2 \tag{24}$$

The contrast and brightness comparison functions of the two images are similar, as shown in formula (25):

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{25}$$

The structural comparison function of two images is constructed from the covariance and variance between the images, as shown in formula (26):

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{26}$$

The comparison functions of the above three parts constitute the SSIM function:

$$\begin{aligned} SSIM(x,y) &= f(l(x,y),c(x,y),s(x,y)) \\ &= [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \end{aligned} \tag{27}$$

Among them, α, β, γ are all greater than zero and are mainly used to adjust the importance of the modules. When α, β, γ are all equal to 1, $C_3 = C_2/2$, then we have:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{28}$$

The range of the SSIM function is [0, 1]. The higher the value, the less distortion in the image and the more similar the two images are.

III. Analysis of examples of artworks generated

III. A. Recursive Algorithm Optimization Analysis

III. A. 1) Experimental Environment

Python is one of the most popular computer programming languages today. Its Chinese name is “Jùmàng” (meaning “giant python”), which was originally chosen for educational purposes targeting non-professional programmers with an entertaining twist. Python is free to use, and it can significantly enhance matrix computation speed through the NumPy library, giving it a distinct advantage over other languages. When handling the same computer vision tasks, Python also has excellent scalability compared to languages like C/C++, Java, and MATLAB. It supports a wide range of matrix operations and dimensional array operations in its application libraries, addressing issues such as code complexity and matrix operation efficiency. As a result, Python enables faster, more convenient, efficient, and high-speed processing in the fields of computer vision and artificial intelligence. Additionally, Python integrates interfaces with high-level programming languages such as Java, C, C++, and Fortran, further enhancing its universality across various fields. In deep learning within artificial intelligence, PyTorch is a Python package and end-to-end machine learning framework. PyTorch demonstrates powerful capabilities in the field of computer vision, achieving desired results through efficient matrix computations and numerical analysis in areas such as image clarification, image segmentation, image visualization, and image recognition. This enables Python programming to achieve core functional tasks in a more concise manner. Given Python's efficient computational capabilities and PyTorch's powerful learning framework, this paper uses Python's PyTorch as the experimental development environment. The experimental hardware environment used for the proposed algorithm in this study is an Intel(R) Xeon(R) Gold 5118CPU@2.60GHz with 64GB of memory. Additionally, in the first method, six NVIDIA RTX 4080Ti graphics cards were used, with the experimental software environment consisting of Ubuntu 20.00, CUDA 6.0, cuDNN 9.0, Python 4.2, and PyTorch 0.6.0.

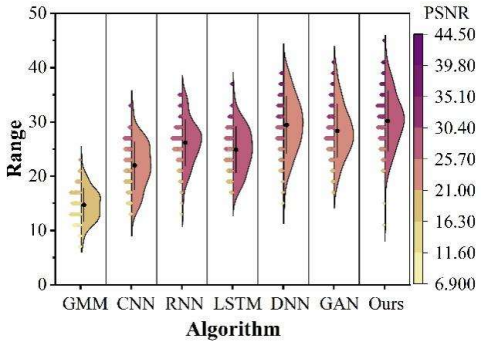
III. A. 2) Experimental Data

To validate the practical application value of the improved recursive algorithm (multi-scale generative adversarial network based on progressive recursion), a numerical quantitative analysis method was used to conduct algorithm validation analysis. The experimental data for the algorithm validation analysis in this section was obtained from a certain art work website (datasets A, B, and C). The next section will provide a detailed analysis and summary of the validation analysis results of the improved recursive algorithm (multi-scale generative adversarial network based on progressive recursion).

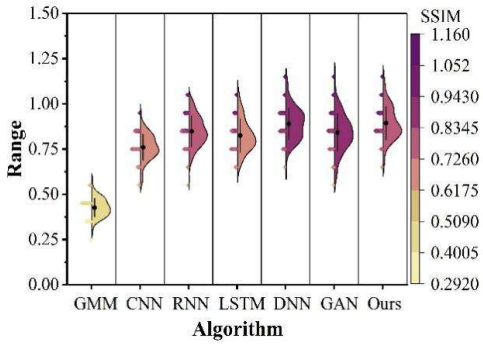
III. A. 3) Analysis of Results

(1) Algorithm Comparison Analysis

To validate the improved recursive algorithm (multi-scale generative adversarial network based on progressive recursion), six comparison algorithms were set up, namely GAN (generative adversarial network), DNN (deep neural network), CNN (convolutional neural network), GMM (method based on sparse coding dictionary learning), RNN (recurrent neural network), LSTM (Long Short-Term Memory Network). The results of the algorithm comparison analysis on datasets A–C are shown in Figures 2–4, where (a) and (b) represent PSNR and SSIM, respectively. As shown in Figures 2–4, the comprehensive PSNR and SSIM values of the proposed method are higher than those of GAN (Generative Adversarial Network), DNN (Deep Belief Network), CNN (Convolutional Neural Network), GMM (Method Based on Sparse Coding Dictionary Learning), RNN (Recurrent Neural Network), and LSTM (Long Short-Term Memory Network), demonstrating the effectiveness of the improved recursive algorithm (Progressive Recursive Multi-Scale Generative Adversarial Network) in generating fine art works.

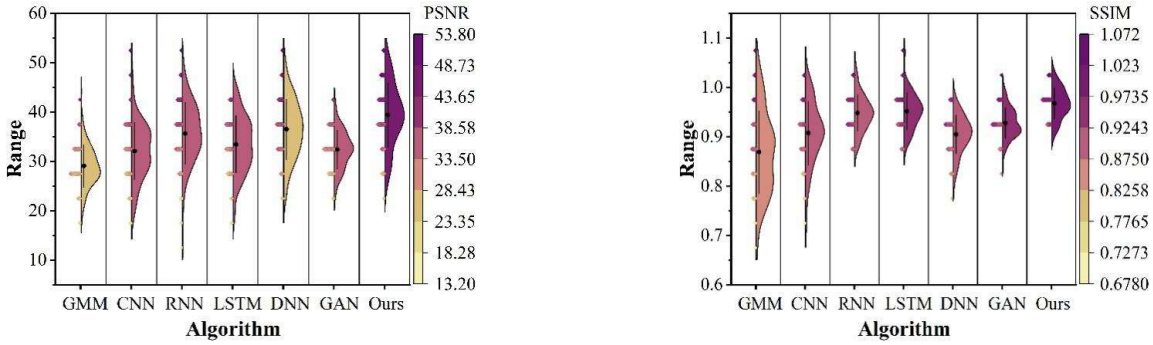


(a)PSNR



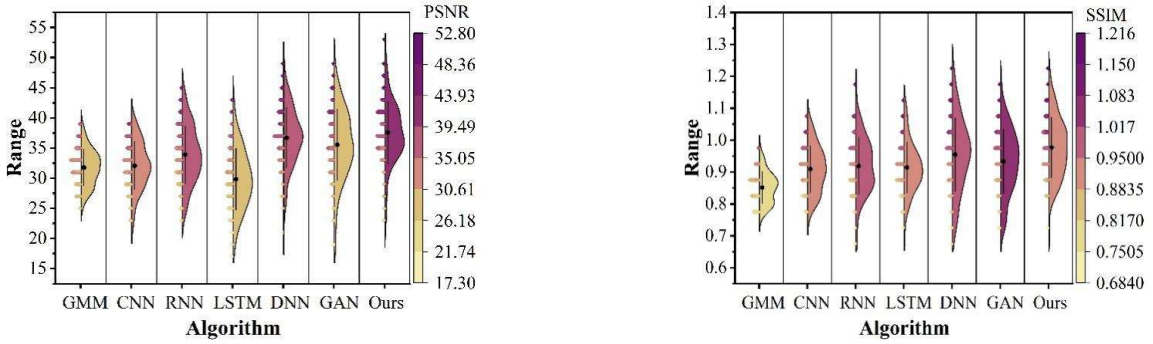
(b)SSIM

Figure 2: Compare and analyze the results of the algorithm on dataset A



(a)PSNR (b)SSIM

Figure 3: Compare and analyze the results of the algorithm on dataset B



(a)PSNR (b)SSIM

Figure 4: Compare and analyze the results of the algorithm on dataset C

(2) Algorithm Application Analysis

Next, we verify and analyze the application performance of the improved recursive algorithm (multi-scale generative adversarial network based on progressive recursion). The results of the algorithm application analysis are shown in Figure 5. As the number of recursions increases, the two indicator values of the generated artwork are significantly improved.

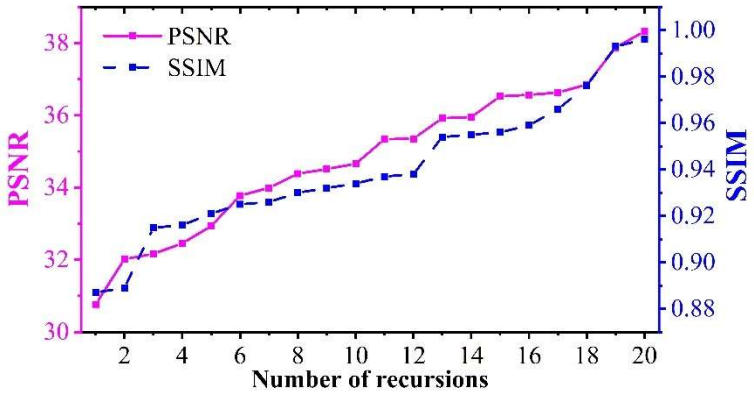


Figure 5: Analysis results of algorithm application

III. B. Analysis of Art Style Transfer Models

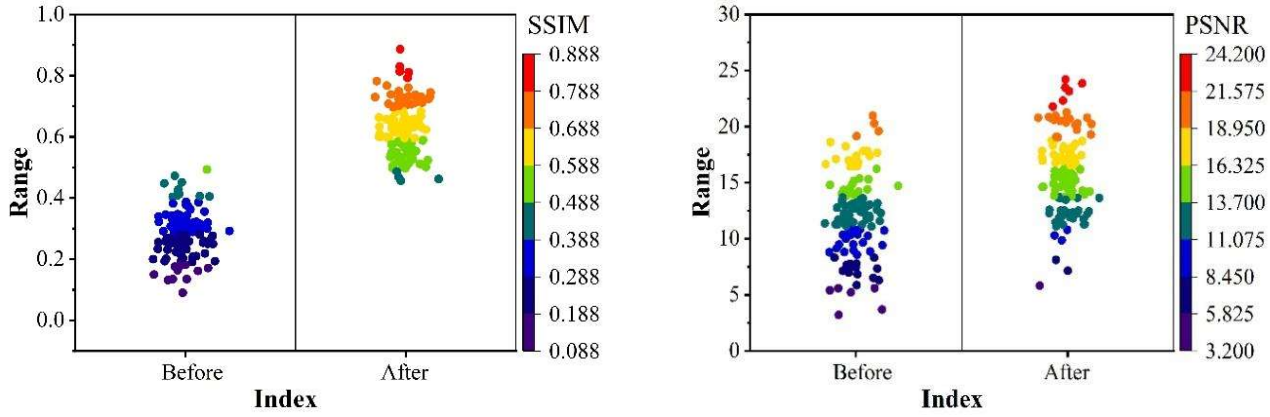
III. B. 1) Qualitative Analysis

In this section, to demonstrate that the general text model can assist art style transfer methods in achieving realistic style transfer tasks for fine art works and to illustrate the versatility of this plugin, we conduct a qualitative analysis

of the art style transfer model. Before incorporating the proposed model, the stylized results exhibited significant distortion and distortion, failing to meet the requirements for realistic style transfer in fine art works. However, after incorporating the proposed model, the stylized results became more realistic, achieving photo-realistic effects. With the assistance of the proposed model, these art style transfer methods successfully completed the task of realistic style transfer in fine art works, significantly enhancing the practicality and versatility of the methods across diverse scenarios.

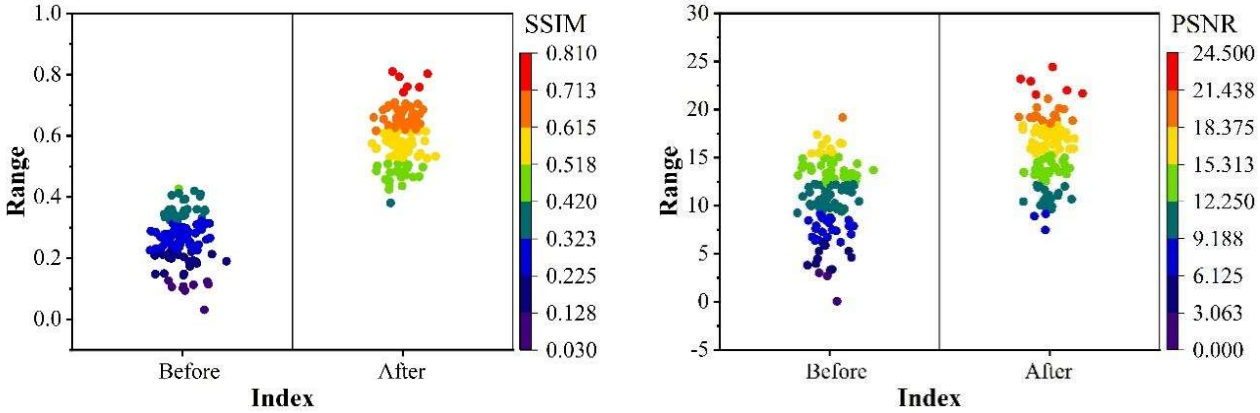
III. B. 2) Quantitative analysis

This section further evaluates the effectiveness of the art style transfer model proposed in this paper using the SSIM and PSNR evaluation metrics. The following models are set as comparison models: AdaIN (Adaptive Instance Normalization Style Transfer Model), SANet (Attention Mechanism Style Transfer Model), MCCNet (Multi-Channel Correlation Style Transfer Model), and AdaAttN (Adaptive Attention Neural Style Transfer Model). We use the aforementioned fine art dataset as the test dataset, which contains 120 content-style image pairs, with images primarily consisting of faces, buildings, or landscapes. We then calculate the SSIM and PSNR values of the transferred images relative to the content images before and after applying these methods. The quantitative analysis results of the art style transfer model are shown in Figures 6–10, where (a) and (b) represent SSIM and PSNR, respectively. Based on the SSIM and PSNR values in Figures 6–10, compared to AdaIN (Adaptive Instance Normalization Style Transfer Model), SANet (Attention Mechanism Style Transfer Model), MCCNet (Multi-Channel Correlation Style Transfer Model), and AdaAttN (Adaptive Attention Neural Style Transfer Model), the style transfer model proposed in this paper demonstrates a more significant improvement in SSIM and PSNR for generated artworks, effectively validating the effectiveness of the proposed style transfer model.



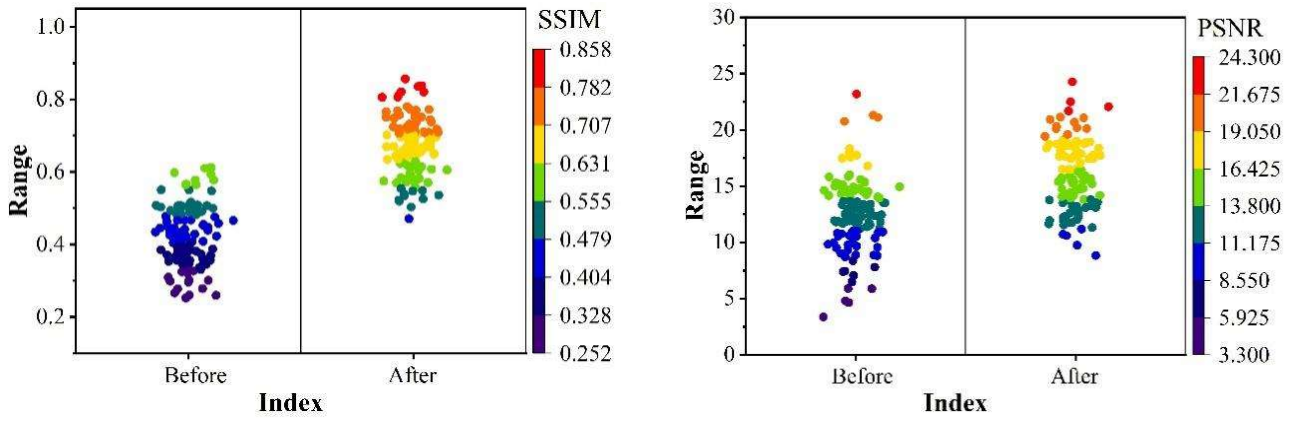
(a)SSIM (b)PSNR

Figure 6: Quantitative analysis results of the AdaIN model



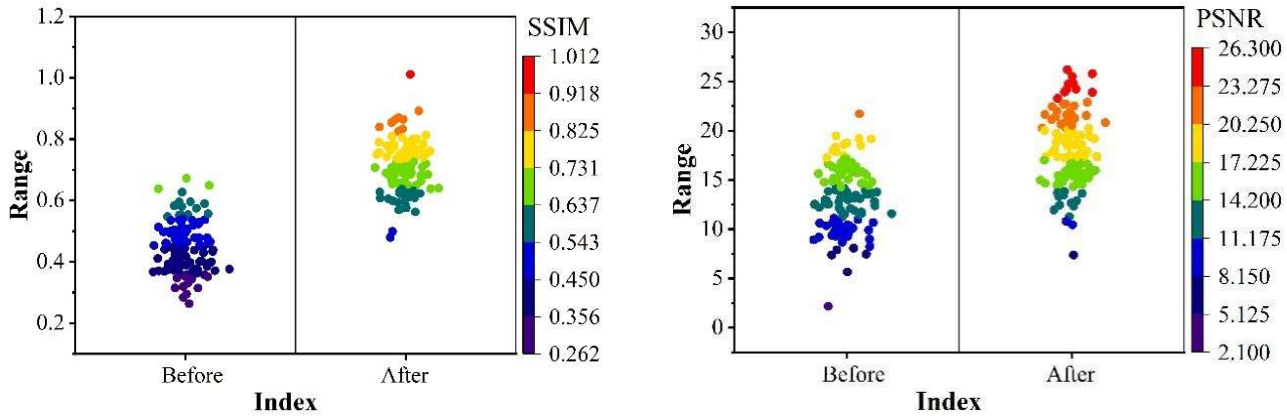
(a)SSIM (b)PSNR

Figure 7: Quantitative analysis results of the SANet model



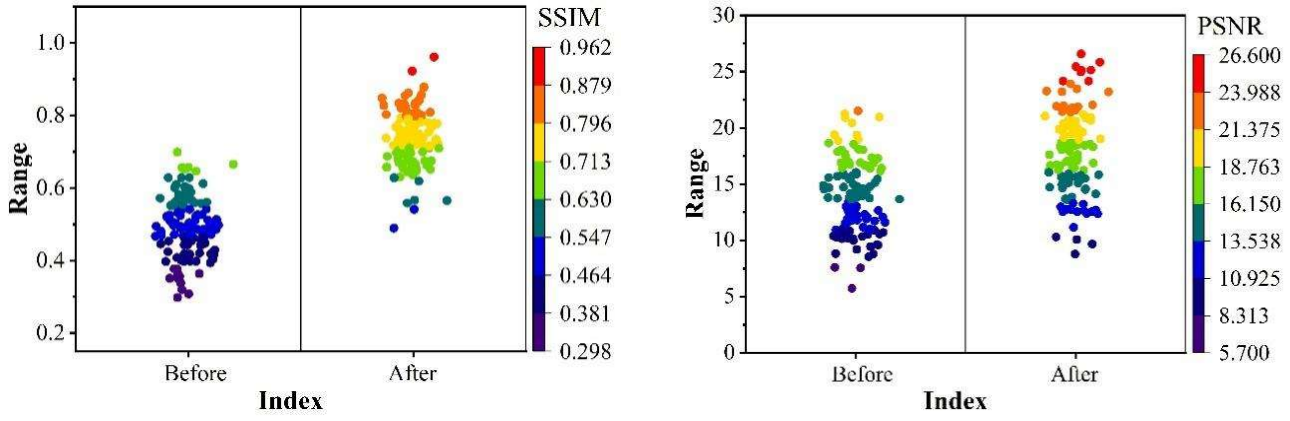
(a)SSIM (b)PSNR

Figure 8: Quantitative analysis results of the MCCNet model



(a)SSIM (b)PSNR

Figure 9: Quantitative analysis results of the AdaAttN model



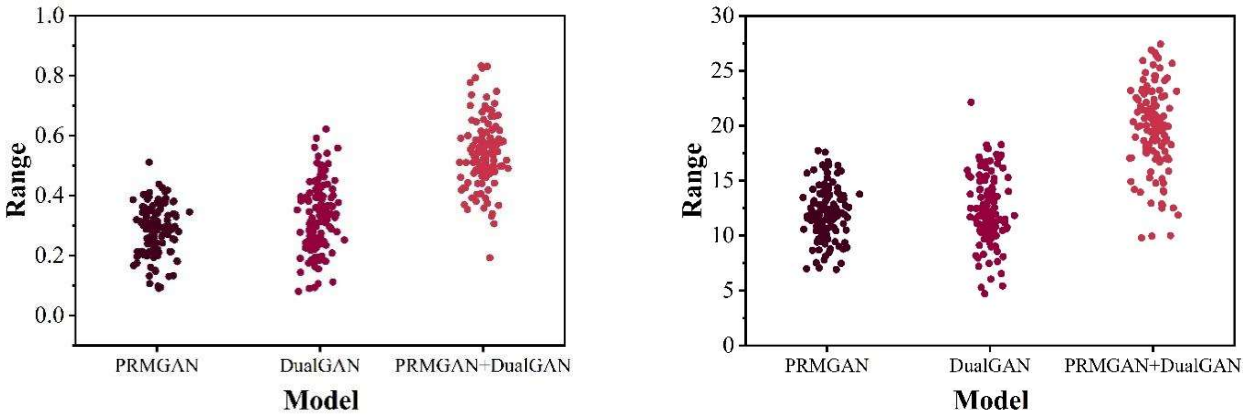
(a)SSIM (b)PSNR

Figure 10: Quantitative analysis results of the DualGAN model

III. B. 3) Analysis of ablation experiments

As mentioned earlier, to ensure that the generated artworks meet the needs of users across different fields, this study optimized the recursive algorithm in art generation (Progressive Recursive Multi-Scale Generative Adversarial Network: PRMGAN). To demonstrate the effectiveness of each module in generating artistic works, ablation experiments were conducted. The results of the ablation experiments are shown in Figure 11, where (a) and (b)

represent SSIM and PSNR, respectively. The data shows that the metrics of the generated artworks improved to some extent when using PRMGAN and DualGAN separately, but the effect was not significant. However, when both were used together, the metrics of the generated artworks improved significantly, making the generated artworks more aligned with user needs and aesthetic standards. This also provides guidance for algorithm optimization and style transfer modeling in art generation.



(a)SSIM (b)PSNR

Figure 11: Analysis results of the ablation experiment

IV. Conclusion

Given the limitations of recursive algorithms in the generation of artistic works, we designed a progressive recursive multi-scale generative adversarial network (PRMGAN). Additionally, to ensure that the generated artistic works better align with user preferences, we established an art style transfer model using the DualGAN network on the existing framework and conducted experiments on the proposed model. The results show that when PRMGAN and DualGAN are used independently, the metrics of the artworks are slightly improved, with SSIM values ranging from 0 to 0.8, and PSNR values ranging from 3 to 23. When both models are used together, the improvement in artistic work metrics is particularly significant, with SSIM and PSNR values ranging from [0.1 to 0.9] and [7 to 28], respectively. This not only demonstrates the practical value of the proposed model in the generation of artistic works but also provides a reference for algorithmic approaches to artistic style transfer modeling in the generation of artistic works.

References

[1] Ting, T. T., Ling, L. Y., Azam, A. I. B. A., & Palaniappan, R. (2023, August). Artificial intelligence art: Attitudes and perceptions toward human versus artificial intelligence artworks. In AIP Conference Proceedings (Vol. 2823, No. 1). AIP Publishing.

[2] Bozard, Z. (2023). What does it mean to create art? Intellectual Property rights for Artificial Intelligence generated artworks. SCJ Int'l L. & Bus., 20, 83.

[3] Hong, J. W., & Curran, N. M. (2019). Artificial intelligence, artists, and art: attitudes toward artwork produced by humans vs. artificial intelligence. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 15(2s), 1-16.

[4] Messer, U. (2024). Co-creating art with generative artificial intelligence: Implications for artworks and artists. Computers in human behavior: artificial humans, 2(1), 100056.

[5] Yusa, I. M. M., Yu, Y., & Sovhyra, T. (2022). Reflections on the use of artificial intelligence in works of art. Journal of Aesthetics, Design, and Art Management, 2(2), 152-167.

[6] Harsanto, K., Zaidan, M. T., Lucky, H., & Suhartono, D. (2024, November). Detecting AI-Generated Artworks With Deep Learning. In 2024 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS) (pp. 472-476). IEEE.

[7] Samo, A., & Highhouse, S. (2023). Artificial intelligence and art: Identifying the aesthetic judgment factors that distinguish human-and machine-generated artwork. Psychology of Aesthetics, Creativity, and the Arts.

[8] Wang, Y., & Sun, L. (2025). The “Aura” of Artworks in the era of Artificial Intelligence. Leonardo, 352-360.

[9] Yu, Z., Wang, H., Katsaggelos, A. K., & Ren, J. (2023). A novel automatic content generation and optimization framework. IEEE Internet of Things Journal, 10(14), 12338-12351.

[10] Durmus, D., Abdalla, D., Duis, A., & Davis, W. (2020). Spectral optimization to minimize light absorbed by artwork. Leukos.

[11] Xu, L., Ye, C., & Tang, J. (2024, November). Intelligent Generation and Optimization of New Media Art Driven by AIGC. In Proceedings of the 2024 4th International Conference on Big Data, Artificial Intelligence and Risk Management (pp. 521-527).

[12] Papia, E. M., Kondi, A., & Constantoudis, V. (2023). Entropy and complexity analysis of AI-generated and human-made paintings. Chaos, Solitons & Fractals, 170, 113385.

[13] Chen, J., An, J., Lyu, H., Kanan, C., & Luo, J. (2024). Learning to evaluate the artness of AI-generated images. IEEE Transactions on Multimedia.

[14] Kostelnick, C. (2020). The art of visual design: The rhetoric of aesthetics in technical communication. Technical Communication, 67(4), 6-27.
[15] Welke, D., Purton, I., & Vessel, E. A. (2023). Inspired by art: Higher aesthetic appeal elicits increased felt inspiration in a creative writing task. Psychology of Aesthetics, Creativity, and the Arts, 17(3), 261.
[16] Wang, X., Lyu, Y., Huang, J., Wang, Z., & Qin, J. (2021). Interactive artistic multi-style transfer. International Journal of Computational Intelligence Systems, 14, 1-13.
[17] Chen, H., Wang, Z., Zhang, H., Zuo, Z., Li, A., Xing, W., & Lu, D. (2021). Artistic style transfer with internal-external learning and contrastive learning. Advances in Neural Information Processing Systems, 34, 26561-26573.
[18] Wang, X., Oxholm, G., Zhang, D., & Wang, Y. F. (2017). Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5239-5247).
[19] Zheng, L. (2025). Artistic style image migration model based on cycle-consistent generative adversarial networks. International Journal of Information and Communication Technology, 26(16), 53-68.
[20] Respini, E., & Erickson, R. (Eds.). (2019). When Home Won't Let You Stay: Migration Through Contemporary Art. Yale University Press.
[21] Wang, W., Shen, W. G., Guo, S. M., Zhu, R., Chen, B., & Sun, Y. X. (2018, November). Image artistic style migration based on convolutional neural network. In 2018 5th International Conference on Systems and informatics (ICSAI) (pp. 967-972). IEEE.
[22] Safta, R. (2023). ARTISTIC INSPIRATION AND DIVINE INSPIRATION. LIMBA ȘI LITERATURA–REPERE IDENTITARE ÎN CONTEXT EUROPEAN, (32), 175-186.
[23] Władyka-Łuczak, Z. (2019). Artistic Inspiration: Initiation of an Artistic Communication. Acta Universitatis Lodziensis. Folia Litteraria Polonica, 54(3), 223-236.
[24] Mihaila, O. (2023). Artistic Creations. Originality, Inspiration, Imitation and Artificial Intelligence. Rom. J. Intell. Prop. L., 9.
[25] Sadiq H. Abdulhussain & Basheera M. Mahmmod. (2021). Fast and efficient recursive algorithm of Meixner polynomials. Journal of Real-Time Image Processing,18(6),1-13.
[26] Jean-Guillaume Dumas,Thierry Gautier,Clément Pernet,Jean-Louis Roch & Ziad Sultan. (2016). Recursion based parallelization of exact dense linear algebra routines for Gaussian elimination. Parallel Computing,57,235-249.
[27] Feng Wang &Zihan Zhang. (2025). Pseudo-coordinates graph convolutional generative adversarial network for art style transfer. International Journal of Information and Communication Technology,26(6),45-61.