A Deep Learning-Based Generative Adversarial Network for Digital Art Style Migration

Wenting Ou
School of Art and Design, Fuzhou University of International Studies and Trade, Fuzhou 350202, China

***Abstract*—This study introduces the ConvNeXt-CycleGAN, a novel deep learning-based Generative Adversarial Network (GAN) designed for digital art style migration. The model addresses the time-consuming and expertise-driven nature of traditional artistic creation, aiming to automate and accelerate the style transfer process using artificial intelligence. The ConvNeXt-CycleGAN integrates ConvNeXt blocks within the CycleGAN framework, enhancing convolution capabilities and leveraging self-attention mechanisms for precise and nuanced artistic style capture. The model undergoes rigorous evaluation using multiple performance metrics, including Inception Score (IS), Peak Signal-to-Noise Ratio (PSNR), and Fréchet Inception Distance (FID), ensuring its effectiveness in generating high-quality, diverse images while retaining fidelity during style transfer. The ConvNeXt-CycleGAN surpasses traditional GAN models across key metrics: it achieves an IS of 12.7004 (higher image diversity), a PSNR of 14.0211 (better preservation of original artwork integrity), and an FID of 234.1679 (closer resemblance to real artistic distributions). Additionally, its ability to efficiently train on unpaired images via unsupervised learning enhances its real-world applicability. This research presents an architectural innovation by combining ConvNeXt blocks with the CycleGAN framework, offering robust performance across diverse datasets and artistic styles. The ConvNeXt-CycleGAN represents a significant advancement in the integration of AI with creative processes, providing a powerful tool for rapid prototyping in digital art creation and innovation.**

Keywords*—*Generative Adversarial Networks (GANs); deep learning; style transfer; unsupervised learning; neural style transfer

I. Introduction

Painting is a visual art form that combines lines, colors, and abstract elements to depict real or imagined subjects [1]. It is a two-dimensional aesthetic art with a high degree of beauty, and many excellent paintings have emerged throughout history. However, traditional painting requires professional painters to invest substantial time and effort to refine their work. With the continuous development of deep learning in the fields of image processing and virtual reality, scholars have begun to employ mathematical models to integrate the artistic elements of one painting into another [2]. This progress has given rise to the style migration technique, which leverages artificial intelligence to fuse art and technology. Style migration not only drives technological reform [3] and provides robust technical support for artistic creation but also inspires the generation of art images, alleviating the laborious nature of traditional art creation.

Despite the significant advancements in style transfer techniques, key limitations remain. Traditional methods such as non-photorealistic rendering and texture transfer suffer from poor generalization and require extensive manual adjustments. Neural style transfer techniques, including VGG-based approaches and transformer-based models, have improved style fidelity but often fail to maintain fine-grained details and content consistency. GAN-based methods like CycleGAN and StarGAN have shown promise but lack robustness in handling unpaired data and diverse artistic transformations. To bridge this gap, we propose ConvNeXt-CycleGAN, which integrates ConvNeXt residual blocks into the CycleGAN framework. This novel approach enhances convolutional capabilities and self-attention mechanisms, ensuring more precise style migration, improved image quality, and efficient training on unpaired datasets. Our contributions include an architectural innovation that boosts style transfer fidelity and experimental performance improvements demonstrated through metrics such as Inception Score, PSNR, and FID. The rest of the paper is structured as follows: Section II reviews related work in style migration and neural style transfer techniques; Section III details the ConvNeXt-CycleGAN methodology, including its network architecture and training process; Section IV describes the implementation of a digital art style migration system based on the proposed and finally, Section VI concludes the paper with key findings and future work directions.

II. Related Work

Traditional style transfer techniques include non-photorealistic rendering [4, 5] and texture transfer [6,7]. While these methods can generate simple artistic re-creations, they suffer from significant limitations, such as poor generalization, an inability to extract high-level semantic features, and extended training times. The field of deep learning has accelerated advancements in computer vision, particularly after Gatys et al. [8–10] introduced neural networks into style transfer. Their VGG-based style transfer model attracted considerable attention from both academia and the art community. Subsequent improvements have been proposed, such as incorporating a Markov structure to model high-level features [11], statistical histogram loss to simulate the distribution of key image features [12], and Laplace loss, which addresses asymmetry issues in generated images while preserving low-level input details [13]. However, these approaches primarily focus on global style transfer, often leading to local style inconsistencies in the generated images. To overcome this, region-specific style transfer methods [14] emerged, aiming to establish semantic mappings between style and content image regions. Furthermore, automated image semantic segmentation techniques have been introduced to streamline the process of aligning semantic features between content and style images.

*Corresponding Author

Recent advancements have expanded the scope of style transfer beyond the reliance on a reference style image. For instance, Kwon et al. [15] proposed a framework that utilizes text descriptions to guide texture transfer in content images, leveraging the CLIP model and a novel patch-wise text-image matching loss with multiview augmentations. Meanwhile, StyTr2 [16] utilizes transformer-based architecture, improving the model's ability to capture global information and enhance style transfer effectiveness. The ArtFlow method introduces reversible neural flows and an unbiased feature transfer module to mitigate content leakage in universal style transfer, ensuring integrity across multiple stylization iterations [17]. CAST (Contrastive Arbitrary Style Transfer) employs contrastive learning to improve style representation learning from image features, yielding more consistent and high-quality style transfer results [18]. Additionally, the AdaAttN module introduces adaptive attentive normalization, allowing per-point style adaptation and enhancing visual quality, especially in video-based applications [19]. The InST method innovatively uses inversion-based style transfer, enabling efficient style adaptation from a single image without requiring complex textual descriptions [20].

Despite substantial improvements in style transfer algorithms, particularly those leveraging pre-trained network models—challenges such as style overflow and insufficient stylization control persist. The emergence of Generative Adversarial Networks (GANs), introduced by Goodfellow et al. in 2014 [21], revolutionized style transfer by employing an adversarial process between a generator and a discriminator to refine image stylization. GAN-based style transfer methods significantly improve image quality and generation fidelity. To accommodate diverse artistic needs, researchers have designed specialized GAN architectures, including supervised Conditional Generative Adversarial Networks (CGANs) [22] and unsupervised StarGAN models [23], which enhance versatility in style transfer applications.

III. Image Style Migration Method Based on ConvNeXt-CycleGAN

Sanghyun et al. [24] referred to the idea of Swin Transformer and proposed ConvNeXt network, in which the ConvNeXt residual block uses deep convolution, similar to the weighted sum operation in self-attention, which is used to improve the performance of the network. In this paper, we propose the ConvNeXt-CycleGAN model, which incorporates ConvNeXt residual blocks into the generator to enhance artistic style migration.

A. Network Infrastructure

The network structure of ConvNeXt-CycleGAN model is improved based on the CycleGAN network, as shown in Fig. 1. The ConvNeXt-CycleGAN model consists of two generators G and F, two discriminators D_X and D_Y. Firstly, the ConvNeXt-CycleGAN model network training is unsupervised learning, i.e., the dataset training is unpaired, which enables bidirectional generation of images between domains X and Y. The ConvNeXt-CycleGAN model network is trained by the network generator. Selecting an arbitrary image x from the source domain X and inputting it into the generator G, the generated image $G(x)$ needs to be re-inputted into the generator F again. Secondly to preserve the contour features of the input image, the cyclic consistency loss [26] function is still used to constrain the reconstructed image. Again, the normalization method in the encoder and decoder is set to Layer Normalization (LN). The ResNet residual network in the converter is replaced with the ConvNeXt-block residual module in the expectation of high-quality generated results with the target style. The final discriminator is consistent with the AMS-CycleGAN model in Section IV, i.e., the attention mechanism module is introduced to prompt the generator to focus on certain key pixel locations of the image, ignoring or even directly filtering out irrelevant parts to obtain the style feature information needed for the synthesized image. In the ConvNeXt-CycleGAN model the loss function is the same as the CycleGAN model, including the generation of the adversarial loss, the cyclic consistency loss, and the constant mapping loss, which effectively regulates the content structure information, brightness, and color contrast of the generated image.

B. Generator Network Structure

The generator network structure of the ConvNeXt-CycleGAN model is shown in Fig. 2, and the internal structure information is shown in Table I. It consists of three parts: encoder, converter and decoder. The first part of the encoder: the image of 3*256*256 is transmitted to the first convolutional layer, and after the calculation of Conv-LN convolutional kernel of 7*7, the feature map of 64*256*256 is output; and then after two layers of downsampling, i.e., Conv-LN convolutional kernel of 3*3, the output of the network is the feature map of 64*64*256. The second part of the converter: after four layers of ConvNeXt Block residual network with the same architecture, the input and output are 64*64*256 feature maps, as shown in Fig. 3. Third part decoder: due to the symmetry of the encoder and decoder architectures, i.e., the decoder is set up with two layers of upsampling, i.e., De Conv-LN convolution kernel as 3*3 network layer to recover the original image size, and finally outputs 3*256*256 image by Conv-Tanh convolution kernel as 7*7 network layer. In this case, the ConvNeXt-CycleGAN model architecture contains the LN normalization method, but the ConvNeXt network by default performs the normalization process in the last dimension, i.e., (B, H, W, C), whereas the dimensions used in this experimental part are (B, C, H, W), i.e., extracting the mean (μ) and the standard deviation (σ) of the input image in the dimensions of C, H, and W. The ConvNeXt-CycleGAN model is based on the following model: (B, C, H, and W).

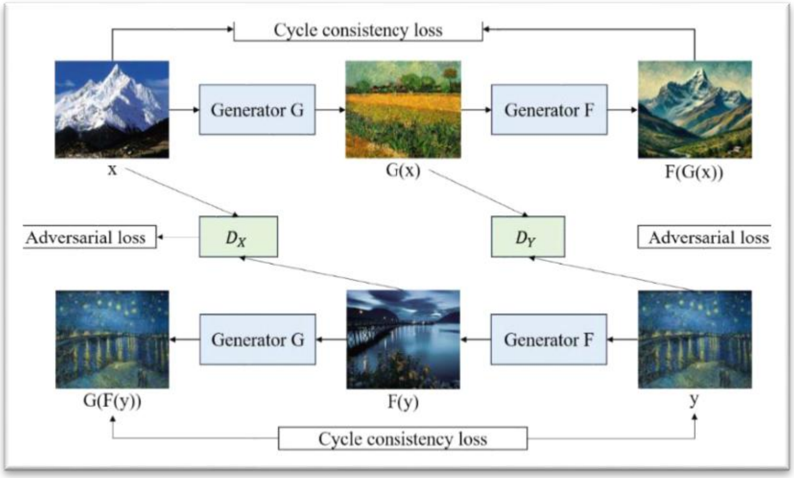


Fig. 1. ConvNeXt-CycleGAN overall network structure diagram.

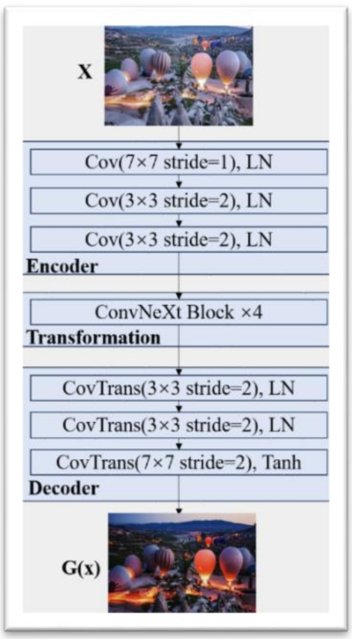


Fig. 2. ConvNeXt-CycleGAN generator network structure diagram.

TABLE I CONVNEXT-CYCLEGAN GENERATOR INTERNAL STRUCTURE INFORMATION

Components	Structural Information
Encoder (Down-sampling)	ReflectionPad2d (3) Conv2d (3,64, k=7, s=1), LN Conv2d (64, 128, k=3, s=2, p= 1), LN Conv2d (128, 256, k=3, s=2,p=1), LN
Transformation (ConvNeXt block*4)	Depthwise Conv2d (256, 256, k=7, p=3, s=1), LN Conv2d (256, 1024, k=1, s=1), GELU Conv2d (1024, 256, k=7, s=1)
Decoder (Up-sampling)	ConvTranspose2d (256, 128, k=3, s=2, p=1), LN ConvTranspose2d (128, 64, k =3, s=2, p= 1), LN ReflectionPad2d (3) Conv2d (64, 3, k = 7, s = 1) Tanh ()

The design of the ConvNeXt Block residual module mainly includes: first, the GELU activation function has the property of non-saturation, so it avoids the problem of gradient saturation in most of the time, which makes the neural network more easy to converge during the training process; second, the use of larger convolution kernel, adopting 7*7 convolution kernel in the first layer, and shifting the depth convolution module upward from 1*1 conv- >depth-wise conv->1*1 conv structure to depth-wise-conv->1*1 conv->1*1 conv structure, and change the size of the convolution kernel for depth convolution from 3*3 to 7*7; third, Layer Scale scales each channel number, and the scale is a learnable parameter (γ). The parameter γ is in the form of a vector with the same dimension as the dimension of the input channels, and for feature transformation, the parameter γ is multiplied by the feature map, i.e., x (output feature map) = $\gamma * x$ (input feature map); fourth, Drop Path is a regularization method, which mainly removes multi-branching structures randomly from the deep learning model. Fifth, less normalization is used. Borrowing the idea of Transformer, the use of normalization is reduced, so the normalization layer in the ConvNeXt Block residual network is relatively reduced, and only the normalization layer after depth-wise-convolution is retained. Sixth, the batch normalization (BN) layer is a commonly used normalization operation in convolutional neural networks, which can accelerate the convergence of the network and reduce overfitting, but a small number of samples selected in a training session can lead to poor generation, and there is also the problem that the computation of the mean and the variance in the testing phase differs from that of the training set. Liu et al. [25] borrowed the layer normalization used in Transformer. In [25], the layer normalization used in Transformer is used to calculate the mean and standard deviation of all the feature channels in turn, which is not related to the size of the batch, so the normalization layer in ConvNeXt Block is converted to layer normalization.

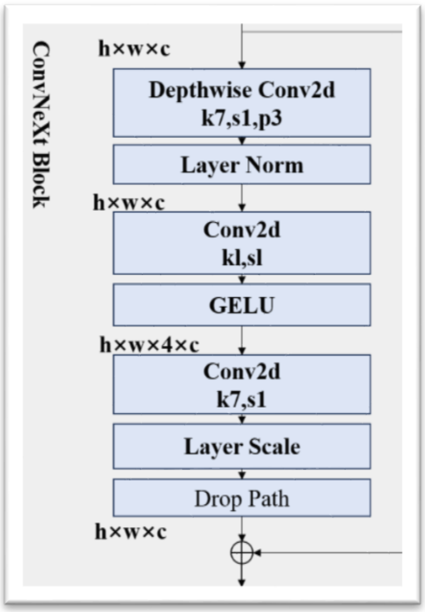


Fig. 3. Informative diagram of ConvNeXt block network structure.

C. Aggregate Loss Function

The ConvNeXt-CycleGAN model proposed in this paper is based on the CycleGAN network structure, and the total loss function loss of the entire network training includes the generative adversarial loss to compare the generated image with the data image, and constantly iterate on the generated model data; the cyclic consistency loss retains the contour features of the input image in the generated image as much as possible, and also improves the generative adversarial network training stability; the constant mapping loss reduces the possibility of the generator automatically modifying the color tone of the generated image.

$$\mathrm{L}_{\mathrm{Generator}} = \lambda_1 \mathrm{L}_{\mathrm{lsgan_Generator}} + \lambda_2 \mathrm{L}_{\mathrm{identity}}(\mathrm{G},\mathrm{F}) + \lambda_3 \mathrm{L}_{\mathrm{cycle}}(\mathrm{G},\mathrm{F},\mathrm{X},\mathrm{Y}), \quad (1)$$

$$\mathrm{L}_{\mathrm{discriminators}} = \min_{\mathrm{D_Y}} \mathrm{L}_{\mathrm{lsgan}}(\mathrm{G},\mathrm{D_y},\mathrm{X},\mathrm{Y}) + \min_{\mathrm{D_X}} \mathrm{L}_{\mathrm{lsgan}}(\mathrm{F},\mathrm{D_x},\mathrm{Y},\mathrm{X}), \quad (2)$$

$$L(G,F,D) = arg \min_{G,F,D_Y,D_X}(L_{Generator}, L_{\mathrm{discriminators}}) \quad (3)$$

where the parameters $\lambda_1, \lambda_2, \lambda_3$ are used to control the linear combination of the generator and discriminator loss functions. The values of $\lambda_1, \lambda_2, \lambda_3$ weights are set to 1.0, 0.5, and 10.0 respectively in the experiment.

The training flow of the ConvNeXt-CycleGAN model for the data bi-directional generation experiment is shown in Table II. An image x is randomly extracted from the natural image domain (X) and inputted into the generator G. Similar to the previously proposed CycleDPN-GAN model and AMS-CycleGAN model, firstly, layer normalization is used in the convolutional layer to compute all the feature channel components. Second, in the residual module, the ConvNeXt Block is used, firstly, the feature map 64*64*256 is used as the input, and the number of groups is equal to the number of input channels, i.e., the number of channels is 256, and each channel corresponds to a convolutional kernel, and the spatial information is mixed-weighted within a single channel. Secondly, the number of channels will be expanded to 4 times of the original, imitating the idea of Swin Transformer network, at this time, the size of the feature map is 1024, and finally, after the network layer, the channels of the feature map will be restored to 256, and at this time, the size of the feature map will be 64*64*256, and then re-input the image $G(x)$ generated by the generator to the generator F, and the discriminator will judge the authenticity of the image.

TABLE II OVERALL FLOW CHART OF THE CONVNEXT-CYCLEGAN MODEL

Conv NeXt-Cycle GAN-based training process for art style image migration
Input: natural image domain (X), artistic image domain (Y), number of iterations T, initial learning rate α_0, weights $\lambda_1, \lambda_2, \lambda_3$ Parameters θ_G, θ_F of initialized generator mapping function G, F Parameters W_Y, W_X of initialized discriminator D_Y, D_X Output: generated images x and y
for t=1,2,…, $T_{max} = 200$: 1: Randomly draw an image x from the natural image domain (X) and enter it into the generator G to output $G(x)$. On the other hand, an image y is randomly selected from the artistic image domain (Y) and entered into the generator F to output $F(y)$. 2: The generated image $G(x)$ and the art image y are sent to the discriminator D_Y, and the performance of the network is improved by the Attention Mechanism module, i.e., by the interdependence between the feature channels, i.e., the importance weights of the different channels are obtained and then applied to the corresponding channels of the previous intermediate feature map F. The following is an example of how to minimizemin(D_Y). Minimize $\min_{D_Y} L_{\mathrm{lsgan}}(G, D_y, X, Y)$, optimize the discriminator D_Y according to the associated error, optimize according to Adam's algorithm, and update W_Y. And the generated image $F(x)$ and the natural image x are fed to the discriminator D_X, minimize $\min_{D_{D_v}} L_{\mathrm{lsgan}}(F, D_x, Y, X)$, discriminator D_X and update W_X. X and update W_X. 3: Send $G(x)$ to generator F and output reconstructed image $F(G(x))$. And send $F(y)$ to generator G, output the reconstructed image $G(F(x))$, compute $L_c ycle(G, F, X, Y)$; then using the second step, the resulting $\min_{D_Y} L_{\mathrm{lsgan}}(G, D_y, X, Y)$ and $\min_{D_X} L_{\mathrm{lsgan}}(F, D_x, Y, X)$, compute the generative antagonistic loss, i.e., $L_{\mathrm{lsgan\_Generator}}$. Optimize the generators G and F according to Adam's algorithm, update θ_G, θ_F. Were, if $t > t_1$,the learning rate linearly decays $\alpha = \alpha_0(T - t)/(T - t_1)$) end

IV. Design of a Digital Art Style Migration System Based on Generative Adversarial Networks

The proposed system architecture integrates advanced Generative Adversarial Network (GAN) technologies to automate the style transfer from target artistic images to source images, preserving the structural integrity of the source content while creatively transforming its aesthetic style. This system is built on a modular architecture that enhances scalability, maintainability, and performance. The system architecture is shown in Fig. 4.

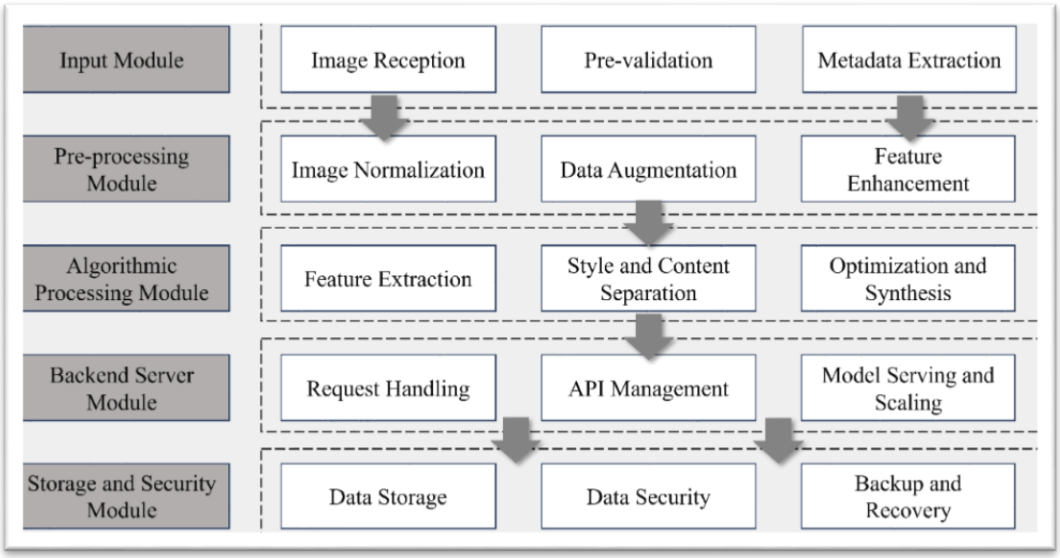


Fig. 4. System architecture of digital art style migration system.

A. System Modules

Input Module: This module is the entry point for the system, accepting diverse image formats including JPEG and PNG. It validates and processes images to ensure compatibility with the style migration model. This module also handles initial image adjustments such as resolution normalization and color space conversion to prepare images for subsequent processing steps.

Pre-processing Module: Critical for standardizing input data, this module applies a series of transformations to the input images. These include resizing the images to a uniform scale, applying normalization to adjust pixel values for neural network processing, and potentially augmenting the data to increase the robustness of the style transfer model. This step ensures that the style migration process operates under optimal conditions by providing consistently formatted input data.

Style Migration Model: At the core of the architecture is the style migration model powered by the ConvNeXt-CycleGAN, which utilizes advanced neural network techniques for deep style learning. This model leverages the unique properties of ConvNeXt blocks within a CycleGAN framework to apply high-quality artistic style transfers. The model operates under an unsupervised learning paradigm, allowing for bidirectional image style translation between distinct domains, facilitated by a dual generator and discriminator setup.

Post-processing Module: After the style transfer, this module refines the output images to enhance visual quality. Adjustments made here include tuning the color balance, enhancing contrast and sharpness, and applying final cropping or padding as necessary. This step ensures that the final styled images are visually appealing and maintain a high degree of fidelity to the artistic intent.

Output Module: This module manages the storage and distribution of the final styled images. It supports functionalities such as saving the images in various formats, preparing them for download, or embedding them into digital galleries. The output module ensures that users can easily access and utilize the generated artworks in their desired manner.

B. Backend Server

1) Architecture and technology stack: The backend server architecture is designed to efficiently handle computational loads and multiple user requests simultaneously, ensuring robustness and scalability. The server employs a microservices architecture, which allows for the modular deployment of the application's components. This modularity facilitates independent updating and scaling of services, enhancing the system's flexibility and maintenance efficiency.

For the technology stack, the system utilizes Python due to its extensive support for scientific computing and machine learning libraries. Python's Flask framework is selected for handling HTTP requests and responses, owing to its lightweight nature and its ability to scale up to accommodate growing user demand. Flask provides the flexibility necessary for rapid development and deployment of web applications, which is crucial for iterative testing and enhancement in response to user feedback.

2) Model deployment: The style migration model, a key component of this architecture, is deployed as a Docker container. This approach ensures that the model runs in an isolated environment, where dependencies are managed consistently, thus eliminating conflicts between different running applications. Docker also simplifies the deployment process across different development and production environments, ensuring consistency and reducing setup times.

Kubernetes is employed to orchestrate these containers, managing their lifecycle, scaling them up or down based on traffic demands, and maintaining system availability through load balancing strategies. Kubernetes also facilitates the rollout of new updates with minimal downtime, enabling continuous integration and continuous deployment (CI/CD) practices that are essential for maintaining the operational efficacy of the system.

C. Storage and Security

Image storage is managed through integrated solutions that prioritize security and efficiency. Both original and styled images are stored in a manner that supports quick retrieval and guarantees data integrity and confidentiality.

V. Results and Discussion

A. Experimental Setup and Environment

In our experiments with the ConvNeXt-CycleGAN model, we conFig.d the batch size to a single instance per training iteration, covering a total of 200 epochs. Both the input and output resolutions were maintained at 256*256 pixels. Network optimization was conducted using the Adam algorithm, starting with a learning rate of 0.0002. This rate was maintained steady for the initial 200 epochs, followed by a gradual reduction to zero towards the end of the training period. An NVIDIA RTX 3090 GPU powered the computations.

B. Introduction to the Dataset

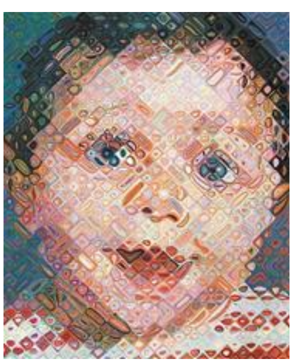
The experiments in this chapter are important to apply the model on the art style dataset, the real images in the dataset used are animal images, the animal dataset is 3600 randomly selected animal images downloaded from Chapter 3 as the training set of this chapter, and 200 animal images are randomly selected as the test set of this chapter.

The art style dataset is a public dataset downloaded from wikiart, and some images of the art style dataset are shown in Fig. 5. The downloaded dataset is cropped by Python to 256×256 size images, and the art style training set mainly contains 637 Van Gogh works, 511 Ukiyo-e images, 419 Monet works, and 309 Paul Cézanne works. The collection is organized in the following ways.

(a) Abstract Expressionism

(b) Analytical Cubism

(c) Baroque



(d)Contemporary Realism

(e) Early Renaissance

(f) Fauvism

Fig. 5. Art dataset image.

C. Test Metrics

In this section, the effectiveness and generalization of the existing network model and the improved model in the previous chapter as well as the proposed model in this chapter on the art style migration task are verified by conducting multiple sets of experiments among four art styles, and it can be found that the existing network have learned the style It can be found that the existing network learns the overall style characteristics of the style image, but the effect is not good enough in the content learning and detail migration. The proposed model in this paper is improved compared with the existing network and the improved network in the previous chapter, and it can maintain the content characteristics of the content image as well as realize the migration of the art styles in the details and as a whole in the visual effect, which verifies the validity and versatility of the proposed model in the data domain of multiple styles. In order to further verify the superiority of the proposed model in this chapter compared with the improved model in the previous chapter and other networks, the proposed model in this chapter is further compared with the existing models using the four metrics of IS, SSIM, PSNR and FID.

Inception Score is used to evaluate the quality of generated images by a model, particularly in the context of generative adversarial networks (GANs):

$$\mathrm{IS} = \exp\left(\mathbb{E}_{x\sim p_g}[\mathrm{KL}(p(y|x) \parallel p(y))]\right) \tag{4}$$

Where $p(y|x)$ is the conditional probability distribution of the label y given the generated image x as predicted by an Inception network. $p(y)$ is the marginal probability distribution of the labels. $\mathrm{KL}(\cdot\|\cdot)$ is the Kullback-Leibler divergence.

SSIM is used to measure the similarity between two images:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{5}$$

Where μ_x and μ_y are the means of images x and y, σ_x^2 and σ_y^2 are the variances of images x and y, σ_{xy} is the covariance between x and y, C_1 and C_2 are constants to stabilize the division.

PSNR is used to measure the quality of a reconstructed image compared to its original version (6):

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}}\right) \tag{6}$$

Where MAX_I is the maximum possible pixel value of the image (e.g., 255 for an 8-bit image), MSE is the mean squared error between the original image and the reconstructed image (7).

$$MSE = \frac{1}{m\cdot n}\sum_{i=1}^{m}\sum_{j=1}^{n}[I(i,j) - K(i,j)]^2 \tag{7}$$

Where $I(i,j)$ and $K(i,j)$ are the pixel values of the original and reconstructed images, respectively.

FID measures the distance between feature distributions of real and generated images (8):

$$\mathrm{FID} = \| \mu_r - \mu_g \|_2^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}) \tag{8}$$

Where μ_r and μ_g are the means of the real and generated image feature vectors, respectively, Σ_r and Σ_g are the covariance matrices of the real and generated image feature vectors, respectively. Tr denotes the trace of a matrix.

D. Test Results

The proposed model was rigorously evaluated against established style migration algorithms. We utilized three metrics for this comparative analysis: Inception Score (IS), Peak Signal-to-Noise Ratio (PSNR), and Fréchet Inception Distance (FID). The results, detailed below, illustrate the efficacy of our model in relation to its counterparts.

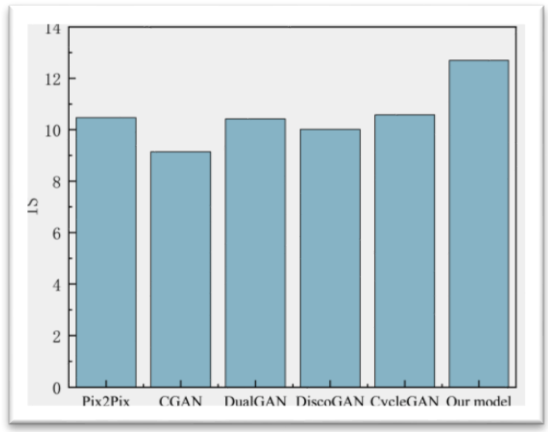


Fig. 6. IS metric of all the models.

From Fig. 6, our model achieved the highest IS value at 12.7004, indicating its superior capability in generating images that are both meaningful and diversified compared to the other models tested. This score is significantly higher than that of the CycleGAN, which scored next highest at 10.5812, and substantially outperforms the CGAN model, which had the lowest score at 9.1411.

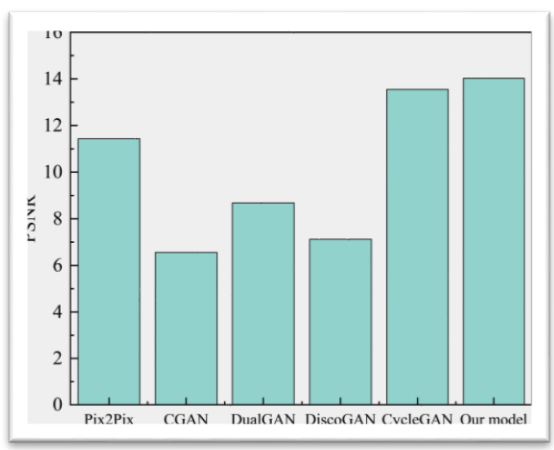


Fig. 7. PSNR result of all the models.

In Fig. 7, as measured by PSNR, our model again outperformed all others with a score of 14.0211. This indicates that our model can produce images with higher fidelity to the original content. CycleGAN followed with a PSNR of 13.5478, while the CGAN model lagged behind at 6.5543, highlighting significant differences in output image quality among the models.

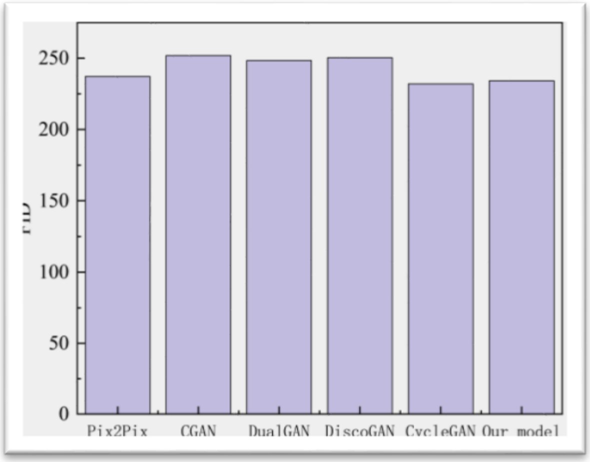


Fig. 8. FID result of all the models.

Fig. 8 illustrates the FID comparison among different models. Our model achieved an FID score of 234.1679, which is close to the best-performing CycleGAN (231.9711). A lower FID score indicates that the generated images closely resemble real images in terms of feature distribution. While our model slightly lags behind CycleGAN in FID, it offers a better trade-off between style diversity (IS) and content preservation (PSNR).

E. Test Discussion

While numerical metrics provide an objective evaluation of style migration performance, a qualitative analysis reveals the perceptual advantages of the ConvNeXt-CycleGAN model. Compared to baseline models, it consistently produced more visually appealing and natural artistic images. A key strength of our approach is its ability to retain fine-grained details while minimizing artifacts and distortions commonly found in CGAN-based methods. This ensures that the stylized images maintain structural coherence without sacrificing artistic expression.

Furthermore, ConvNeXt-CycleGAN demonstrated improved texture consistency and color adaptation over CycleGAN, resulting in more harmonious and refined stylization. The model effectively balances content preservation with artistic transformation, producing high-quality outputs that closely resemble real artworks. These qualitative observations align with the quantitative results, reinforcing the effectiveness of our approach in generating diverse, high-fidelity images suitable for digital art applications.

F. Algorithm Efficiency Analysis

Less efficient algorithms make it difficult to generate a large number of creative designs for images. Hence, assessing the performance of the style migration algorithm is crucial. This study utilizes the control variable technique to measure the efficiency of the algorithm, examining whether training is required and if the algorithm can handle various style transformations effectively and the conversion speed, as shown in Table III.

TABLE III ALGORITHM EFFICIENCY ANALYSIS

Name		CGAN	DiscoGAN	CycleGAN	Ours
Whether training is required		Yes	Yes	104min	131min
Arbitrary style or not		trainable	trainable	trainable	trainable
Need style image		One	One	One	One
Ink style effect		not good	good	good	good
Conversion speed	256*256	3.515s	>1h	0.762s	1.044s
	512*512	16.952s	>1h	2.003s	3.180s
	1024*1024	>1min	>1h	6.855s	11.232s

This investigation delves into the practicality and effectiveness of using generative adversarial networks, particularly CGANs, for the task of transferring styles across a vast array of images. Detailed evaluations indicate that while CGANs are adept at handling complex and vibrant patterns, their performance is noticeably less efficient when applied to simpler, monochromatic styles. The exhaustive training regimen for CGANs necessitates a comprehensive collection of style images, which serves as a critical foundation for achieving satisfactory results. Moreover, DiscoGAN's methodology, which circumvents traditional training protocols, entails a lengthy process of iterative image adjustments. This method, despite its ability to process images with diverse color schemes without prior training, significantly extends the duration required to stylize images—often taking upwards of an hour to refine a single standard 256x256 pixel image under typical CPU processing conditions.

Contrastingly, the innovative style migration technique developed in this study, markedly reduces the time required for style conversion when compared to methods reliant on instance normalization (IN). This efficiency gain is not only reflected in faster processing times but is also quantitatively supported by enhanced PSNR and SSIM values, indicating superior image quality post-stylization.

In summary, the style migration framework proposed herein offers significant advantages for digital image design. It not only expedites the creative process but also supports a broad spectrum of styling tasks. This capability substantially augments the versatility and richness of the digital image database, empowering artists and designers to explore new creative horizons with greater efficiency and effectiveness.

VI. CONCLUSION

The research presented in this paper marks a significant advance in the field of digital art creation through the development and deployment of the ConvNeXt-CycleGAN model. This model not only champions the cause of integrating deep learning into artistic processes but also sets a new benchmark in style migration effectiveness and efficiency, leveraging the cutting-edge capabilities of Generative Adversarial Networks (GANs).

The ConvNeXt-CycleGAN model has demonstrated superior performance over existing GAN models such as Pix2Pix, CGAN, and others, as evidenced by its exceptional scores on several key metrics. Achieving an Inception Score (IS) of 12.7004, it has proven its superior capability in generating images that are not only diverse but also retain a high degree of semantic meaning relative to the style domains being targeted. This indicates a substantial improvement in the model's ability to handle complex style migrations without losing the essence of the original artworks. Moreover, with a Peak Signal-to-Noise Ratio (PSNR) of 14.0211, the model confirms its efficacy in producing high-fidelity images, which is critical for applications where detail preservation is paramount.

Furthermore, the competitive Fréchet Inception Distance (FID) score of 234.1679 underscores the model's capacity to generate stylized outputs that closely mimic the distribution of real-world artistic images. The architectural innovations—such as the integration of ConvNeXt blocks within the CycleGAN framework—play a pivotal role in capturing intricate artistic details and facilitating effective style translation. By employing an unsupervised learning approach with unpaired images, our method significantly reduces the reliance on extensive paired datasets.

Future Work

By explicitly addressing the gap between existing and proposed work, we have identified key areas requiring further research. Current style migration models struggle with real-time performance, precise detail retention, and consistency across diverse datasets. To overcome these challenges, we propose the following strategies for future improvement: (1) optimizing the ConvNeXt-CycleGAN model with lightweight network architectures and quantization techniques to enhance computational efficiency; (2) incorporating advanced perceptual loss functions and attention mechanisms to refine fine-grained detail preservation; (3) expanding the dataset diversity and utilizing semi-supervised learning techniques to improve training consistency and reduce artifacts. These strategies will contribute to a more robust and scalable digital art style migration framework, making AI-powered artistic creation more accessible and efficient.

In future work, we plan to refine the ConvNeXt-CycleGAN model by developing adaptive style control mechanisms that mitigate style overflow, thereby ensuring a more balanced integration of artistic style with the original content. We also aim to optimize the model for higher-resolution images and more complex compositions, which will enable it to handle intricate details and diverse artistic elements more effectively. Furthermore, integrating interactive, user-guided features will allow artists to have greater control over the stylization process, making the model more versatile and user-friendly. Additionally, we intend to conduct comprehensive perceptual evaluations through user studies to better align the generated outputs with artistic standards and industry expectations. These enhancements will not only improve the overall quality and flexibility of the style migration process but also further bridge the gap between advanced AI techniques and practical digital art applications.

References

[1] Andrew, Nell. Moving Modernism: The Urge to Abstraction in Painting, Dance, Cinema. Oxford University Press, USA, 2020.

[2] Santos, Iria, et al. "Artificial neural networks and deep learning in the visual arts: A review." Neural Computing and Applications 33 (2021): 121-157.

[3] Deng, Yingying, et al. "Stytr2: Image style transfer with transformers." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

[4] Rosin, Paul L., et al. "NPRportrait 1.0: A three-level benchmark for non-photorealistic rendering of portraits." Computational Visual Media 8.3 (2022): 445-465.

[5] Karimov, Artur, et al. "Comparing neural style transfer and gradient-based algorithms in brushstroke rendering tasks." Mathematics 11.10 (2023): 2255.

[6] Han, Xinying, Yang Wu, and Rui Wan. "A method for style transfer from artistic images based on depth extraction generative adversarial network." Applied Sciences 13.2 (2023): 867.

[7] Liu, Yuan. "Improved generative adversarial network and its application in image oil painting style transfer." Image and Vision Computing 105 (2021): 104087.

[8] Gatys, Leon A. "A neural algorithm of artistic style." arXiv preprint arXiv:1508.06576 (2015).

[9] Gatys, Leon, Alexander S. Ecker, and Matthias Bethge. "Texture synthesis using convolutional neural networks." Advances in neural information processing systems 28 (2015).

[10] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[11] Svoboda, Jan, et al. "Two-stage peer-regularized feature recombination for arbitrary image style transfer." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[12] Yeh, Mao-Chuang, et al. "Improving style transfer with calibrated metrics." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2020.

[13] Lin, Tianwei, et al. "Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[14] Liao, Yi-Sheng, and Chun-Rong Huang. "Semantic context-aware image style transfer." IEEE Transactions on Image Processing 31 (2022): 1911-1923.

[15] Kwon, Gihyun, and Jong Chul Ye. "Clipstyler: Image style transfer with a single text condition." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[16] Deng, Yingying, et al. "Stytr2: Image style transfer with transformers." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

[17] An, Jie, et al. "Artflow: Unbiased image style transfer via reversible neural flows." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[18] Zhang, Yuxin, et al. "Domain enhanced arbitrary image style transfer via contrastive learning." ACM SIGGRAPH 2022 conference proceedings. 2022.

[19] Liu, Songhua, et al. "Adaattn: Revisit attention mechanism in arbitrary neural style transfer." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

[20] Zhang, Yuxin, et al. "Inversion-based style transfer with diffusion models." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023.

[21] Goodfellow, Ian, et al. "Generative adversarial networks." Communications of the ACM 63.11 (2020): 139-144.

[22] Loey, Mohamed, Gunasekaran Manogaran, and Nour Eldeen M. Khalifa. "A deep transfer learning model with classical data augmentation and CGAN to detect COVID-19 from chest CT radiography digital images." Neural Computing and Applications (2020): 1-13.

[23] Choi, Yunjey, et al. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[24] Woo, Sanghyun, et al. "Convnext v2: Co-designing and scaling convnets with masked autoencoders." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[25] Liu, Junchi, Xiang Zhang, and Zhigang Luo. "TransConv: Transformer Meets Contextual Convolution for Unsupervised Domain Adaptation." Entropy 26.6 (2024): 469.

[26] Dwibedi, Debidatta, et al. "Temporal cycle-consistency learning." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.