IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Multi-stage Generation of Tile Images Based on Generative Adversarial Network

JIANFENG LU[1], MENGTAO SHI[1], YUHANG LU[1], CHING-CHUN CHANG[2], LI LI[1], AND RUI BAI[3].

[1]College of Computer Science, Hangzhou Dianzi University, Hangzhou 310018, China
[2]Department of Computer Science, University of Warwick, Coventry CV4 7AL, U.K. (e-mail: c.c.chang@warwickgrad.net)
[3]Key Laboratory of Brain Machine Collaborative Intelligence of Zhejiang Province, Hangzhou 310018, China (email: bairui@hdu.edu.cn)

Corresponding author: LI LI (lili2008@hdu.edu.cn).

Funding Statement: This work was partially supported by National Natural Science Foundation of China (No. 62172132).

ABSTRACT Deep learning techniques have been recently widely used in the field of texture image generation. There are still two major problems when applying them to tile image design work. On the one hand, there is still lack of enough diverse ceramic tile images for the training process. On the other hand, the output image is difficult to control and adjust, and cannot meet the designer's requirements of interactivity. Therefore, we propose a multi-stage generation algorithm of tile images based on generative adversarial network(GAN). First, the multi-scale attention GAN is applied to generate controllable texture image. Then, the SWAG texture synthesis GAN is also applied to obtain controllable and diverse image style. And finally, through the style iteration mechanism and the multiple step magnification method based on image super-resolution reconstruction network, the final tile images can be automatically generated with larger-size and higher-precision. The relevant experiments demonstrate that our method can not only generate high-quality tile images in a relatively short period of time, but also consider human interaction to a certain extent, and maintain a certain degree of control over the main texture and style of the final generated tile images. It has good and wide application value.

INDEX TERMS Tile images, generative adversarial networks, style transfer, image super-resolution magnification.

I. INTRODUCTION

With the improvement in household living conditions, the demand for high-quality ceramic tiles continues to grow, and more attention is being paid to tile pattern texture aesthetics and personalized design. Although there are many diverse ceramic tiles, it is often difficult to meet various personalized demands.

Two traditional methods are used to design tile images. One is that a tile image is obtained by scanning marble veins and processed with some image transformation [1]. However, it is too difficult to diversify and customize tile images. Another is that multiple texture image blocks are first designed by the designer, and then stitched and fused through a stitching algorithm to generate the final tile texture image, such as Liang [2] and Wang [3]. This requires the designer to make repeated adjustments to the design texture pattern based on the generated tile image effect. In short, these methods rely heavily on manual design, and are difficult to meet varied individual needs.

In recent years, with the development of deep learning techniques, a growing number of deep-learning-based approaches for texture image generation have been proposed. For example, some approaches [4], [5], [6] can perform secondary editing or fusion of user-specified image regions by learning the correlation between pixels. Although it can generate a diversity of texture images based on a specified region, it has the drawback of being capable of performinglocal editing only.

Based on the Encoder-Decoder theory, VAE [7], UNet [8] and other networks were also proposed for the automatic generation of texture images. The input image is mapped into the feature space by an encoder, which generates an image with different texture representation by a decoder when slightly being perturbed. However it is often less controllable, and hard to edit the texture of generated images upon perturbation.

Some scholars have attempted to utilize GAN [9] to generate texture images. Random noise is input to a generator and transformed into a random output image through multiple convolution layers and up-sampling layers. Through the joint generator-discriminator training, the generated random images can gradually show the texture features of the image in the training dataset by iterations. Although it increases the diversity of the output image by changing the input random noise, it has no control over the main texture structure. For the texture image that the user wants to generate, a specific input noise vector cannot be found. In addition, limited by computational performance, the training image size is small and the output image is often blurred.

Other scholars proposed the style transfer methods [10] to generate diverse images. A pre-trained feature extraction model was used to extract high-level abstract feature representations of the original content and style image, respectively. Then given a random input noise image, a new fused image with original texture and different style is generated by iterative optimization. Although diverse images can be generated by varying the input style images, since the network parameters remain fixed during the training process, separate training is required to generate different original images, resulting in longer time to generate multiple images and making it difficult to guarantee the image quality.

As existing ceramic tiles have strict requirements for large image sizes and high-precision, the tile images generated by the above approaches are rarely satisfactory. Therefore, we propose *a multi-stage tile image generation algorithm based on generative adversarial network.* First, the multi-scale attention GAN(MSA-GAN) is applied to generate controllable texture. Next, the stylization with activation smoothing texture synthesis GAN(SWAG-TS-GAN) is also applied to achieve controlled and diverse image styles. Finally, through the style iteration mechanism and the stepwise magnification method based on super-resolution reconstruction network, the final tile image can be automatically generated with larger size and higher precision.

The main accomplishments with this paper are summarized as follows.

(1)A multi-scale GAN with attention is present for enhancing the controllability of tile textures. The attention mechanism is added to multi-scale shortcut weights, where the shortcut connections are built on the basis of the same feature map size of the discriminator and generator. Thus, compared with the traditional MSG-GAN and ResNet, whose multi-scale shortcut weights are set to 1, while the weights are updated adaptively during the network training process, which gives the different importance of the texture features at different scales in our method. Meanwhile, the convergence speed and efficiency of the network are significantly improved.

(2)A texture synthesis GAN network with SWAG is also designed to strengthen the controllability of tile image style. The traditional texture synthesis GAN is prone to produce large amplitude in the activation value. To solve the problem, the new softmax activation transform and smoothing are applied to the loss function. So the change in error back propagation will not be too drastic, avoiding the occurrence of the large amplitude and artifacts. The image quality is better improved with clear details.

(3)The style iteration mechanism and magnification method based on super-resolution reconstruction network are proposed. To resolve the low resolution of the output image produced by traditional texture synthesis GAN, the style iteration mechanism is given, which the previous output image is used as the next input. So the final output image with style-enhanced is produced. In addition, the super-resolution reconstruction network is used for magnification to achieve the high-resolution requirements.

The rest of the paper is organized as follows. Section II reviews related work. The main Section III presents the scheme of the multi-stage tile image generation algorithm, and the core model proposed by main accomplishments. Section IV presents the results of a comprehensive experimental study, and Section V concludes the paper.

II. RELATED WORKS

In this section, we describe the works related to GAN, style transfer, and super-resolution reconstruction network. The GAN network is used for the generation of tile texture in this paper, the style transfer is used for the generation of diverse tile image styles, and the image super-resolution reconstruction network is used for the magnification of the final image with high-precision.

A. GENERATIVE ADVERSARIAL NETWORK

Goodfellow creatively proposed a generative adversarial network(GAN) [9] that provides a new method for generating new images. Recently, more scholars delved into GAN and improved them them to generate relevant new images based on various datasets. For example, in the generation of texture images, Karras et al. [11] proposed ProGAN. The key idea is to grow both the generator and discriminator progressively. Starting from a low resolution, they add new layers that model increasingly fine details as training progresses, to generate higher-resolution texture images. They further proposed StyleGAN [12]. By adding the Ada-IN module to the generator, the intensity of the image features at different scales is directly controlled by adjusting the image "style" of each convolutional layer through the input latent vector. In addition, a certain amount of noise was introduced to simulate the random changes in hair textures and other features. The experimental results demonstrated that it can effectively reduce feature entanglement. Animesh and Oliver proposed MSG-GAN [13]. By allowing the flow of gradients from the discriminator to the generator at multiple scales, convergence during model training will be accelerated. In addition, there were also many distinctive networks about GAN, such as DCGAN [14], pix2pix [15], CycleGAN [16] etc.

B. STYLE TRANSFER

The deep-learning-based style transfer method enables style transformation of the original texture image. Gatys and Ecker discovered that deep neural networks can extract both the underlying and high-level semantic information of an image, and introduced this technique to the field of style transfer. They pioneered a style transfer model based on VGG networks [17]. Content loss and gram loss are calculated by the pre-trained VGG network to minimize the difference between the content features and style features of the generated image and the input image. When network training converges, the final stylized image is obtained. Based on [10], Johnson proposed a faster style transfer algorithm [18]. The encoder-decoder image transformation structure is added into the basic style migration network, so as to accelerate the convergence speed of the generated images, and at the same time, achieve better style transfer results. Zhou et al. [19] made further improvements. They introduced a style transfer network into GAN, and improved the overall loss function based on the style transfer loss term. The final trained network can achieve super-resolution of images while incorporating texture image style information for various types of content images.

C. SUPER-RESOLUTION RECONSTRUCTION NETWORK

Although traditional interpolation-based algorithm [20], [21] can achieve magnification of an image, it still lacks high quality and clarity. The super-resolution reconstruction techniques provide a better solution to this problem.

Li et al. proposed an image super-resolution reconstruction network [22]. The whole network is divided into two modules, feature extraction and reconstruction. The feature extraction module mainly consists of convolutional layers and multi-scale residual blocks, to realize the extraction of multi-scale feature information from the original image. The reconstruction module is mainly composed of convolutional layers and pixel shuffle layers, where pixel shuffle arranges multiple images with similar features in a specific order to generate a feature map with a higher resolution. By combining various structures of feature map in a specific arrangement, the image can be scaled up to different scales (x2x3x4).

III. MULTI-STAGE GENERATION OF TILE IMAGES BASED ON GENERATIVE ADVERSARIAL NETWORK

A. ALGORITHM SCHEME

We propose a multi-stage tile image generation algorithm based on generative adversarial network. To be more detailed, we propose three independent networks, where the output of the previous network is used as the input for the next. First, we present MSA-GAN for generating grayscale texture image blocks. We add multi-scale attention shortcut connections to StyleGAN, to generate more controllable grayscale texture images. Then we present SWAG texture synthesis GAN for generating tile style images. Based on texture synthesis GAN, we perform softmax transform on the stylization loss activation values to be smoother. As a result, the generated tile images have better style transfer effect. Next, we divide the style image into many small image blocks with self-similar structure, and generate larger image blocks by super-resolution reconstruction network. And finally, we can get a large-size tile image with better precision. The whole process is shown in Figure 1.

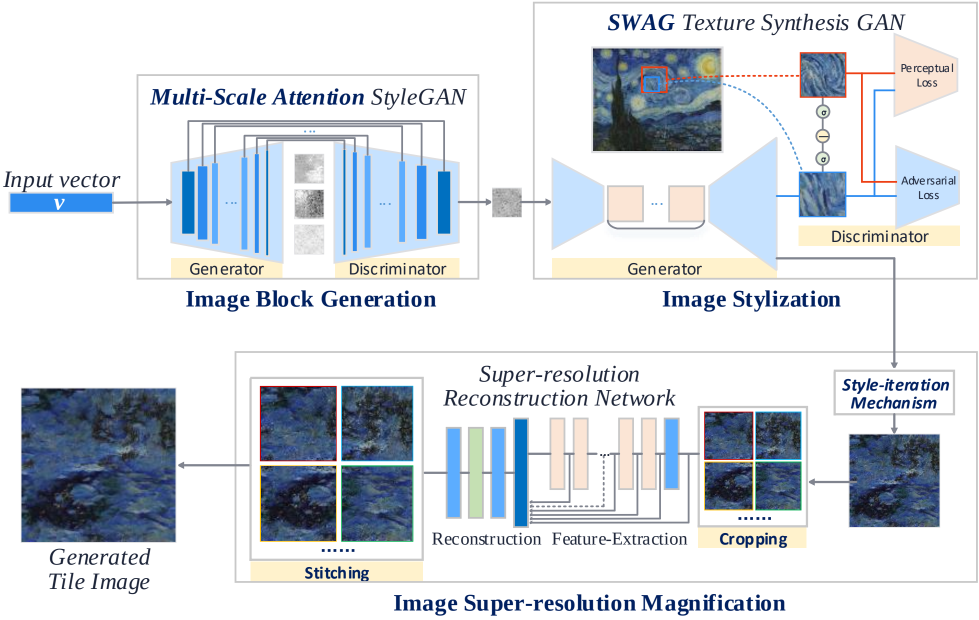


FIGURE 1. The flowchart of multi-stage generation of tile images based on generative adversarial network.

B. IMAGE BLOCK GENERATION BASED ON MULTI-SCALE ATTENTION STYLE-GAN

In this section, we introduce a multi-scale attention mechanism. We add multi-scale attention shortcut connections based on StyleGAN. By allowing the flow of self-adaptive gradients from the discriminator to the generator from multi-scale shortcut connections, the generator can be iteratively updated to learn texture features at different scales. Thus, more texture-detailed and high-quality images can be generated by the final generator. Meanwhile, we attempt to interpolate between different input vectors to generate more progressive texture image blocks. The effectiveness of this method was verified experimentally.

Network Architecture. Our generator and discriminator take StyleGAN [12] as the basic structure. In our generator, note $A_0 = \mathbb{R}^{4*4*12}$ as the initial feature map, which consists of a set of constants, and A_i as the feature map of middle layer. Let g^i be a generator module, consisting of several up-sampling layers, convolutional layers, and AdaIN [23] layers. It converts the feature map A_i into an output feature map A_{i+1}. The input vector v is transformed into an intermediate latent space through a series of linear transformations, and then fed to all AdaIN layers of the generator, which is used to control multi-scale texture of the generated image. The full generator can be defined as a sequence of compositions of the progressive generator module $g^0 \sim g^i$, and converts A_0 into an output grayscale texture image block.

In our discriminator, note D_j as the feature map of the middle layer. The discriminator module d^j consists of several convolution and down-sampling layers, and converts the feature map D_j into the output feature map D_{j+1}. The texture image block is sequentially passed through the progressive discriminator module $d^0 \sim d^j$, and output is the form of a probability value, to judge authenticity of the image.

We propose **multi-scale attention weights**. We add multi-scale attention shortcut connections, corresponding to the same-size feature map from the generator to the discriminator. As shown in Figure 2 with dotted lines, the feature map A_i is input to the 1x1 convolution module o_i, and transformed into a grayscale image O_i. Then O_i and feature map D_j are channel-wise concatenated, and input to the discriminator module d^j to generate D_{j+1}:

$$D_{j+1} = d^j(concat_{channel}[D_j, w_i * o_i(A_i)]). \tag{1}$$

Note that the generator module g^i and discriminator module d^j each have a total number of $k (0 < i, j < k)$, and satisfies $j = k - i$. For all shortcut connections, the feature map O_i needs to be multiplied by the attention weight w_i on top of itself. If w_i is larger, the corresponding shortcut connection has a more significant and positive effect on the generator during network training.

In the process of backward propagation, gradients can flow by all multi-scale shortcut connections from the intermediate layers of the discriminator to the intermediate layers of the generator. Thus our network can learn texture features at different scales quicker.

Improvement of loss function. We introduce the attention weights w_i in the loss function of the network as follows, which allows the generator to selectively learn texture features at different scales:

$$\min_G \max_D L(G, D) = \mathbb{E}_{x\sim p_r} + \mathbb{E}_{z\sim p_z}, \tag{2}$$

$$\mathbb{E}_{x\sim p_r} = \mathbb{E}_{x\sim p_r(x)}[\log D(x, w_1, \ldots, w_k)], \tag{3}$$

$$\mathbb{E}_{z\sim p_z} = \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)), w_1, \ldots, w_k)], \tag{4}$$

where w_i has an initial value of α, and is updated along with the training process. The subsequent experiments in this paper α take the value of 2.5 in all cases. During the training process, all network parameters and attention weights w_i are updated by Adam optimizer, with learning rate 2e-4 and parameter $\beta_1 = 0.5, \beta_2 = 0.9$.

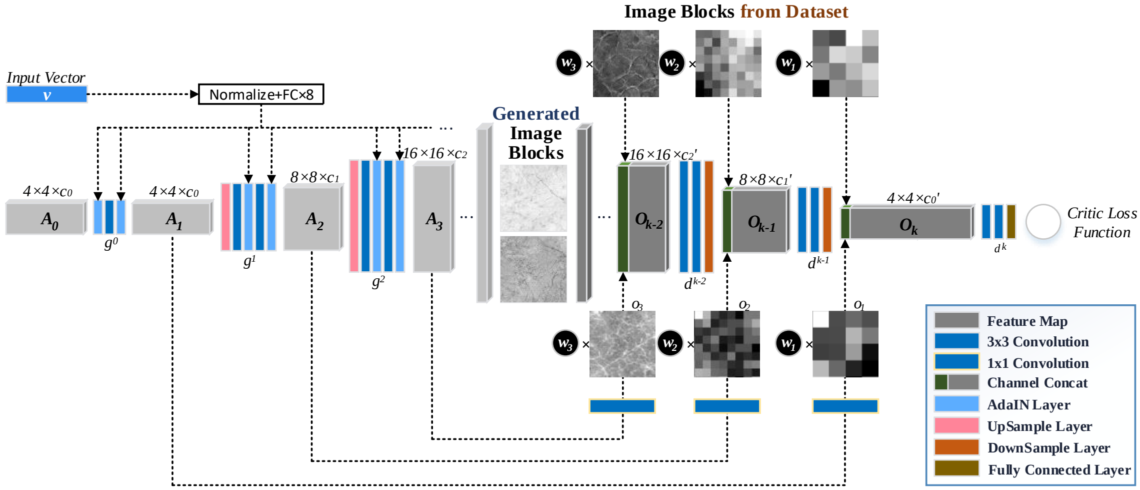


FIGURE 2. Architecture of multi-scale attention StyleGAN.

Then, to prevent an excessive gap between different attention weights, which causes image quality degradation, we add a regular term to the loss function:

$$w_i = \frac{w_i}{\alpha}, \tag{5}$$

$$w_i = \frac{\log(w_i + \max\limits_i w_i)}{\sum\limits_i \log(w_i + \max\limits_i w_i)}, \tag{6}$$

$$w_i = \frac{w_i}{\max\limits_i w_i} * \alpha. \tag{7}$$

All attention weights w_i are updated in regularization for every few batches, thus further excessive gap is reduced.

1) Comparative Study

To verify the effectiveness of our network, a set of comparison experiments are conducted. The Section IV.A dataset is fed into MSG-StyleGAN and our network(MSA-StyleGAN) for training. Among them, MSG-StyleGAN [13] keeps the attention weights at $1(w_i = 1)$ all the time, and the attention weights of our network can be updated and obtained through network training.

From Figure 3, compared with the original StyleGAN and MSG-StyleGAN, the texture generated by our network is richer with more details. And in Figure 4, (a) and (b) depict the change of loss value during network training. Our network has significantly smaller oscillation amplitude, further reflecting the faster training convergence and network stability. (c) and (d) depict the changes in the attention weights. It can be seen that the attention weights of our network remain stable during training, which further demonstrates that our network has more significant attention to the texture information of larger-scale images.

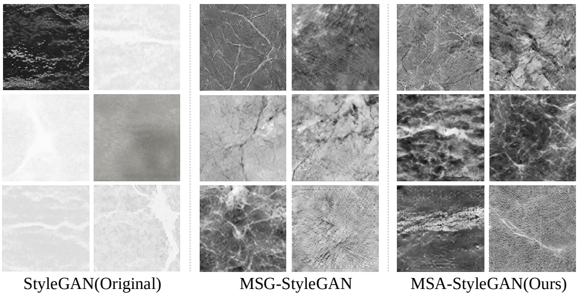


FIGURE 3. Random generated samples for comparison between different network.

2) Input Vector Interpolation Experiments

A pair of input vectors v_{image1}, v_{image2} for two gray-scale texture images with large differences are interpolated:

$$v_{**} = coef * v_{image1} + (1 - coef) * v_{image2}, \tag{8}$$

and a new vector v_{**} is obtained and inputted into the generator to obtain a series of progressive image blocks. Figure 5 demonstrates that the generated texture images are gradient.

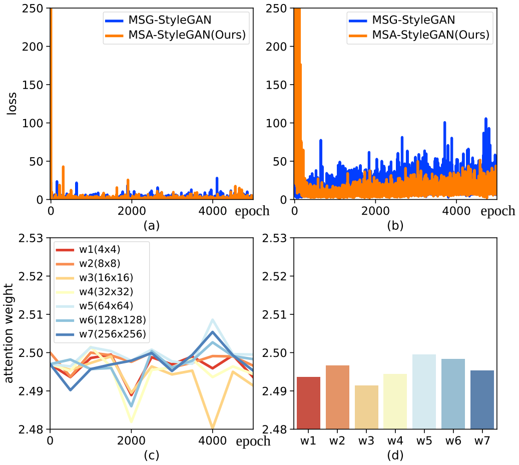


FIGURE 4. Network visualization for faster convergence. (a)Discriminator loss value changing during training. (b) Generator loss value changing during training. (c) Multi-scale attention weights changing during training. (d) Final multi-scale attention weights.

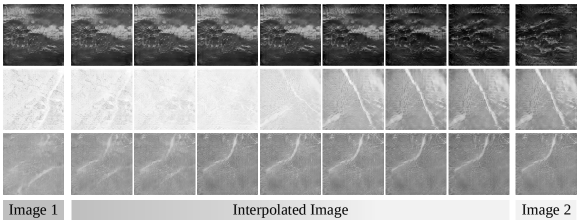


FIGURE 5. Interpolation of the input vector to obtain a series of gradient texture images.

C. IMAGE STYLIZATION BASED ON IMPROVED TEXTURE SYNTHESIS GAN

In the traditional style transfer network, although the subject texture features of the input image are well preserved owing to the introduction of residual blocks [24], the stylization quality is significantly reduced [25], with a single tone and artifacts. In this section, we improve texture synthesis GAN(TS-GAN) [19] using activation smoothing. Experiments prove that it has a better style enhancement effect.

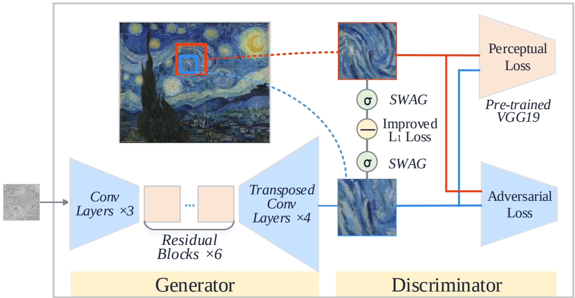


FIGURE 6. Architecture of improved TS-GAN.

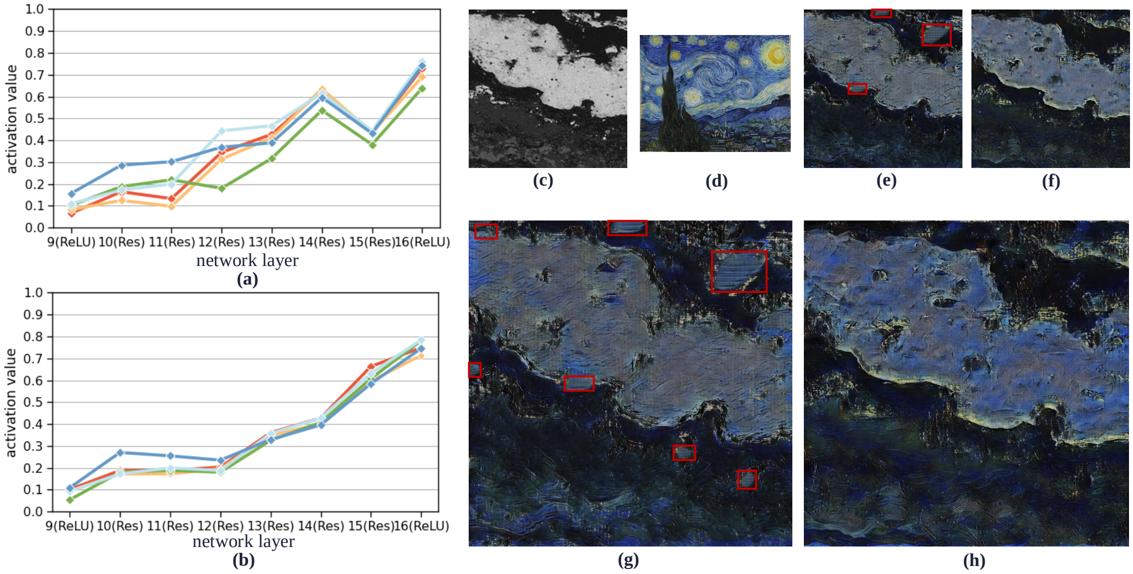


FIGURE 7. Experimental results of the improved TS-GAN, compared with the original network. (a) Activation tracks from original network. (b) Activation tracks from improved network. (c) Input grayscale texture image block. (d) Training style image. (e) Output image from original network. (f) Output image from Improved network. (g) Second iteration output image from original network (h) Second iteration output image from improved network.

Network Architecture. The structure of the network follows the original TS-GAN [19]. There is a down-sampler, a residual module, and an up-sampler in the generator, for converting the input grayscale texture image into a stylized image. The down-sampler is composed of 3 convolution layers, two of which use stride-2 convolutions that reduce the spatial dimensions of the input. The residual module is composed of 6 residual blocks, which can realize the enhancement of texture features [19]. The up-sampler is composed of 4 transposed convolution layers, where the first three layers use stride-2 transposed convolutions, thus ensuring that the final output image is twice as long and wider as the input image.

There is also a GAN discriminator and a VGG discriminator. The GAN discriminator adopts PatchGAN as its basic structure to determine whether the generated image is consistent with the original style image. The VGG discriminator adopts a pre-trained VGG19 [17] to evaluate the authenticity of stylistic features by calculating the perceptual loss of the gram matrix [10].

Training process. The network is trained based on a specific style exemplar image. During each training iteration, we randomly crop a $2k \times 2k$ target block T from the style exemplar as the ground truth, and a $k \times k$ source block t^* from T, which is input to the generator to generate the style image T'. The difference between T' and T is calculated using the loss function:

$$L = L_{adv} + \lambda_1 L_{content} + \lambda_2 L_{style}, \tag{9}$$

where L_{adv} is the adversarial loss of GAN, $L_{content}$ is the content loss, L_{style} is the perceptual loss [10]. λ_1, λ_2 are the hyperparameters, where the specific values are determined by the style image size and its style information. Then, the network parameters are updated based on the Adam optimizer, with learning rate 2e-4 and parameter $\beta_1 = 0.5, \beta_2 = 0.9$. For maximum utilization of the available data we choose not to set aside a validation or a test set. The loss function of the network in the training process is as follows.

Stylization With Activation smoothinG(SWAG). Owing to the introduction of residual blocks, bumps and jittering of activation values (the coefficients of the successive feature maps output from network middle layers) occur, which affects the image stylization process. Considering that the softmax transform can suppress large activation values, and can map activation values with different scales to the 0-1 range, the distance calculation becomes more accurate and smooth [25]. We introduce the softmax activation transform into the calculation of the loss item. Thus, the probability of jitter is effectively mitigated during network training, and the image stylization process becomes more stable.

The whole calculation process is described as follows. We perform a global softmax transform on the target block T and the output image T' to obtain $\sigma(T)$ and $\sigma(T')$. Then we calculate the L2 distance for each pixel softmax activation value, and take the average value as the final $L_{content}$:

$$\sigma(T_{i,j}) = \frac{e^{T_{i,j}}}{\sum_{m,n} e^{T_{m,n}}}, \tag{10}$$

$$L_{content} = \frac{1}{2mn} \sum_{i,j} (\sigma(T_{i,j}) - \sigma(T'_{i,j}))^2, \tag{11}$$

where σ denotes the global softmax transform of the image T. Additionally, there is no residual block in the feature extraction VGG19 network, so the softmax transform is not considered to be incorporated into the process of calculating the loss term L_{style}:

$$L_{style} = \sum_{h} \frac{1}{4N_{G^h}^2 M_{G^h}^2} \sum_{i,j} (G_{i,j}^h(T) - G_{i,j}^h(T'))^2, \quad (12)$$

where $G_{i,j}^h(T)$ denotes the feature map of T output through the $h-th$ layer of VGG19 discriminator, and $N_{G^h}^2$,$M_{G^h}^2$ denote the number of channels of feature map T, and the total number of pixel points per channel, respectively.

1) Comparative Study

To verify the effect of the improved TS-GAN, a set of experiments for comparison conducted. A gray-scale tile image block with high contrast is selected, and input into the original network and the improved TS-GAN, respectively. Five random image positions are selected and tracked the corresponding activation values across the network layers, using nearest-neighbor interpolation. And the output style images are also recorded.

As shown in Figure 7, the activation values of the improved network are smoother, with no bumps or jitter. The generated images are more colorful, and do not show any artifacts (compared with (g), there are less meshed and lined artifacts marked by red boxes in (h)), which is more obvious from the second iteration output style images.

D. IMAGE SUPER-RESOLUTION MAGNIFICATION

In this section, we first enlarge the image using the style-iteration mechanism. And limited by the GPU computing speed, we then use the super-resolution reconstruction network to further enlarge the image to meet large-size and high-precision requirements. The image super-resolution reconstruction network mainly learns and memorizes the super-resolution and continuity features, and does not blur and lose image details after magnification, which is unique different from traditional interpolation.

1) Style-iteration Mechanism

Based on the network described in Section III.C, we propose the style-iteration mechanism to generate larger images. The original grayscale texture image block is denoted by I_0, and the style image output in one iteration is I_1. Next, input I_1 into the same network again to obtain the second iteration output style image I_2. By repeating $t-1$ times according to the same operation steps, we can obtain the final style image I_t.

Figures 7, 10 and 11 demonstrate that as the number of iterations increases, the image becomes enlarged and the texture and style features are further enhanced.

2) Magnification Algorithm for Tile Images Based on Super-resolution Reconstruction Network

We further propose a complementary algorithm for image enlargement, based on a super-resolution reconstruction network [22]. The generated style image is cropped into many small image blocks with self-similar structure, and each small image block is amplified by the super-resolution reconstruction network, and these enlarged small image blocks are stitched together to finally obtain a large-size tile image with high-precision.

Image block cropping. Denote the tile style image generated by III.D.1 as I_t, and divide it into a number of equal-sized, non-overlapping tile image blocks $I_{t(k)}$.

Image block magnification. All image blocks $I_{t(k)}$ are input into the pre-trained super-resolution reconstruction network RN, to generate $I_{t(k)}^*$ as follows:

$$I_{t(k)}^* = RN(I_{t(k)}, sz), \quad (13)$$

where sz is the amplification ratio, which is supported by 2,3 and 4, in the current pre-trained network.

Image block stitching. Arrange and stitch $I_{t(k)}^*$, according to the order of $I_{t(k)}$, to form the final large tiled image I^*.

IV. EXPERIMENTS

A. DATASET PRODUCTION

Establishment of grayscale texture image dataset. We first select 48 large tile images, with image size greater than $4096px \times 4096px$. For each large tile image, a number of $256px \times 256px$ image blocks are randomly cropped. Subsequently, those with higher complexity blocks are selected for grayscale processing. The final selected grayscale image blocks, with the number around 2500, will constitute the dataset for the training of MSA-StyleGAN.

Establishment of style samplars. In this paper, we also select 15 images with higher resolution and more prominent styles. They are partly derived from natural texture pictures and partly from oil and watercolor paintings suitable for tile production. All of them will be fed into the improved TS-GAN network for training. Figure 8 and Figure 9 show some of the grayscale texture image blocks and style sample images.

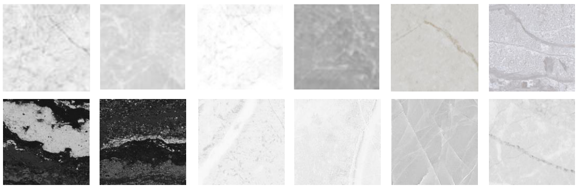


FIGURE 8. Grayscale Texture Sample Images for MSA-StyleGAN.

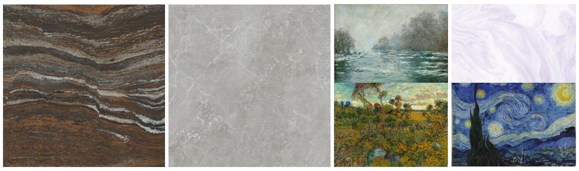


FIGURE 9. Style Exemplars for SWAG texture synthesis GAN.

B. EXPERIMENTAL ENVIRONMENT

We use the following experimental environment for testing: an Intel Core I7-7800X server with 6 cores at 3.5GHz, 64G of RAM, and 2 RTX2080TI-11GB graphics cards. The training and most of the image generation process are carried out in the GPU environment, while CPU environment is only required for testing and the generation of higher-resolution tile images. This is because at this time, a large amount of GPU resource is employed due to the presence of intermediate layer higher-resolution feature maps, which far exceeds the established memory of the GPU.

C. EXPERIMENTAL RESULTS

In order to verify the quality and effectiveness of the tile images generated by our method, this paper conducts a series of experiments based on the existing tile dataset and the experimental environment. First, all of the grayscale texture blocks and style exemplars are fed into the corresponding network for training, and the tile images are randomly generated based on trained network. As shown in Figure 10,11, (b) is a grayscale image block randomly generated by MSA-GAN, and (a) is the style exemplar with a corresponding pre-trained TS-GAN. (c)(4096px × 4096px) is the intermediate generated tile image, which is further amplified to obtain the final result (d)(8192px×8192px). The experiment proves that the texture features and style features are continuously

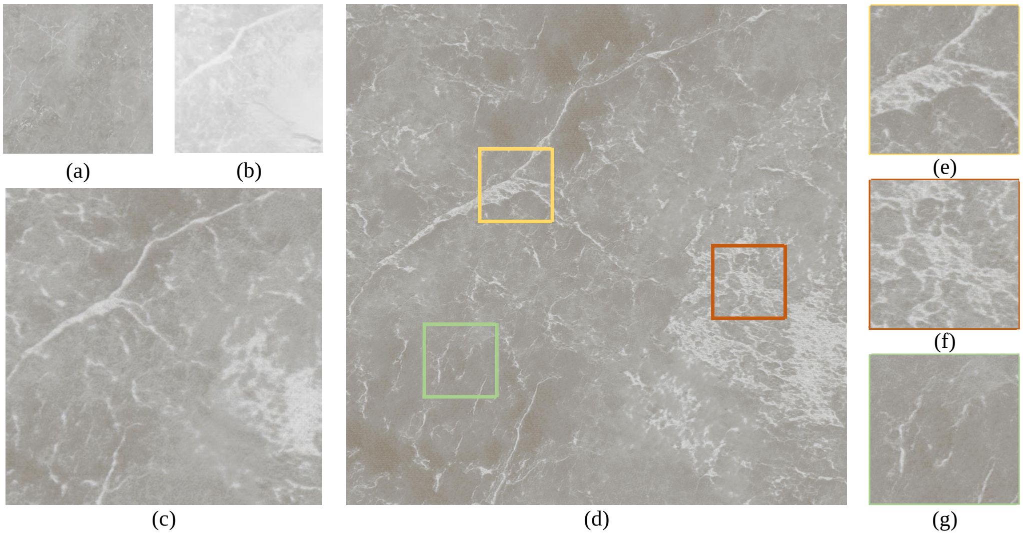


FIGURE 10. Generated sample image. (a)Original style image. (b)Input grayscale tile texture block(256px×256px). (c)Intermediate tile image generation results(4096px×4096px). (d)Output tile image(8192px×8192px). (e)(f)(g)Image blocks after zooming in on some areas of (d).

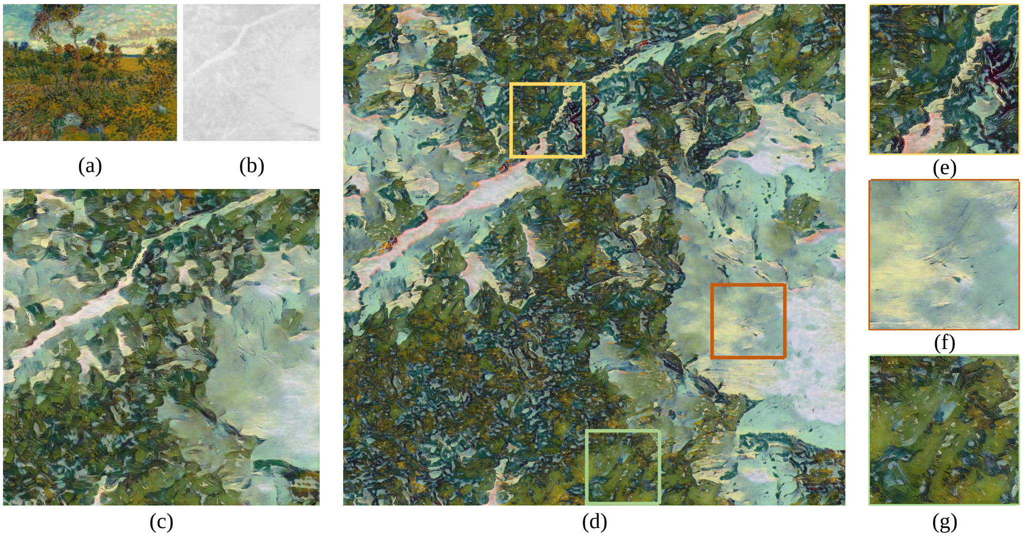


FIGURE 11. Generated sample image. (a)Original style image. (b)Input grayscale tile texture block(256px×256px). (c)Intermediate tile image generation results(4096px×4096px). (d)Output tile image(8192px×8192px). (e)(f)(g)Image blocks after zooming in on some areas of (d).

enhanced in the process of image super-resolution. Three areas of (d) are randomly selected with richer texture features for enlargement, to obtain (e)(f)(g), and it is easy to see that these image blocks are clearer without obvious blurring.

Qualitative Validation. A series of grayscale texture images and style exemplar are input into different networks, and the differences are compared through the network output results. For the basic style transfer algorithm [10] and DCGAN [14], the generated tile images retain most of the grayscale texture image block information, but lack the presentation of style features, and the images are not natural enough. As an example, in the last row of the generated images in Figure 12, the blue patches are embedded in the darker areas of the input grayscale block, which increases the overall sense of violation. For images generated by MGANs [26], although the images are colorful, the main texture features of the images are still missing. For example, in the last three rows in Figure 12, the generated image no longer has distinct texture features compared to the input texture block, and is more susceptible to interference from the oil painting style. The images generated by original TS-GAN [19] are relatively monochromatic in tone and exhibit artifacts. Our method preserves texture regions better compared to TS-GAN, and has stronger color differences with background regions, especially verified in the third row in Figure 12. In conclusion, our method generates images that highlight strong image style features and produces a more intuitive visual effect than other algorithms.

This paper also evaluates the subjective quality of the images generated by the different methods through a set of questionnaires [27]. All methods and models use the same training tile images, and for each method to be compared,

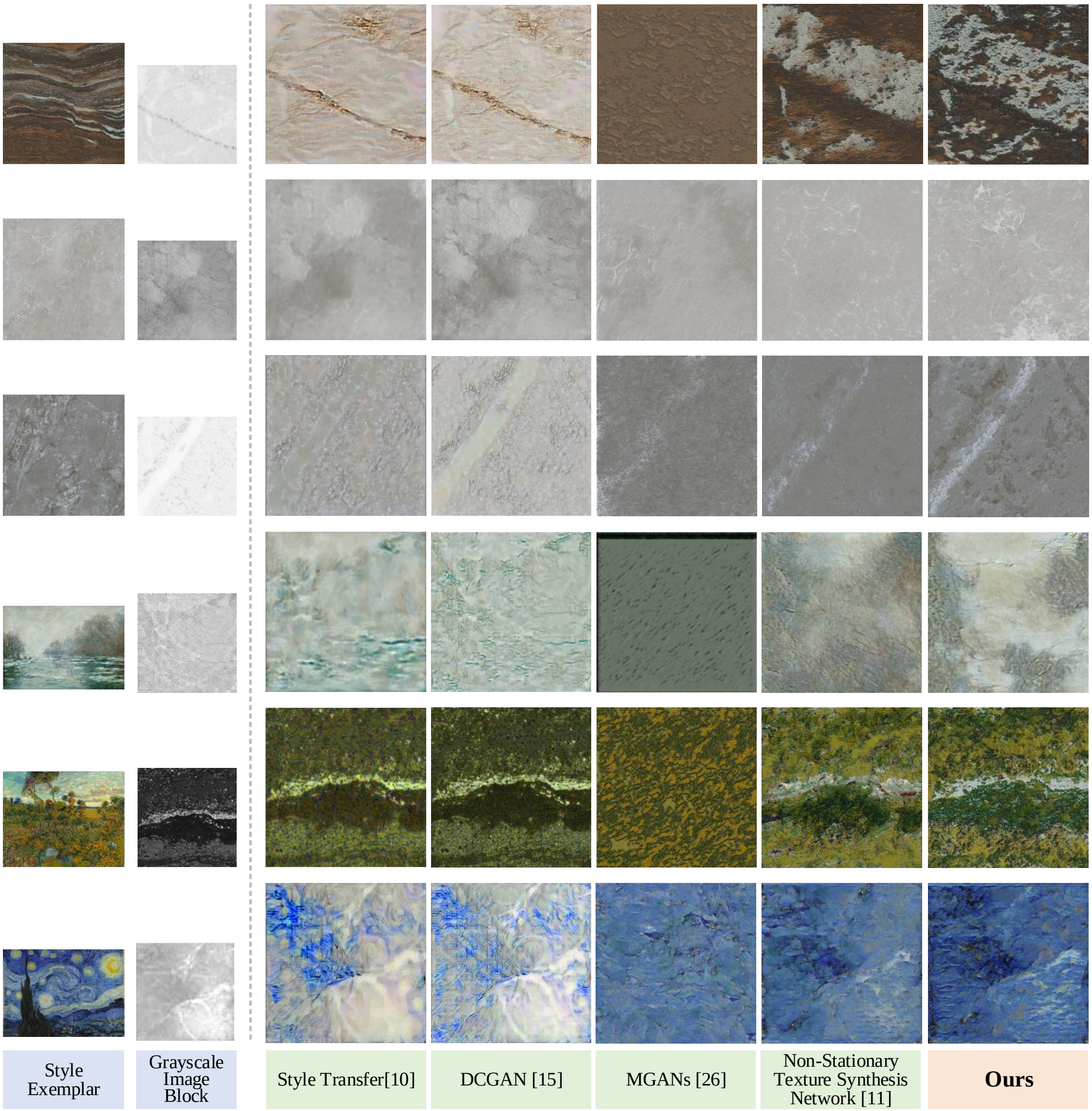


FIGURE 12. Comparison of experimental results between the traditional and the recent style transfer methods.

5 images are randomly generated to form the respective image set. Then 50 people are randomly invited to score the respective image quality evaluation factors of each the image set, with the average value taken as the final result. The image quality evaluation factors are image diversity, degree of retention of major texture features, style transfer effect, image size, and image quality. All features are rated in the range of 0 to 5, with a rating range falling in the interval [0, 1.5] corresponding to a low degree of evaluation factors, (1.5, 3] medium, and (3, 5] high.

The results of the questionnaires are summarized in Table 1. The images generated by the image-editing based method [4], [6] have fewer missing parts of the image, in other words, the missing parts of the image are edited based on a large number of a priori and known features. Although the texture features are better preserved, it is not as good as our method in terms of image diversity, and still needs to be improved in terms of generated image size and quality. In the methods based on AutoEncoder and GAN [8], [12], [14], [28], although the output images have more obvious texture features and the image diversity increases with the change of the input vector, these methods cannot generate tile images with specified styles as our method. In the methods based on style transfer [10], [19], [26],although the output image diversity can be increased by changing the input style image, it is far inferior to our method in terms of image size and image quality, and the texture features are severely distorted. In comparison, the images generated by our method are better evaluated for all the subjective quality features.

Quantitative Validation. Four metrics, namely, maximum image resolution, average image generation time, inception score (IS), and blind/referenceless image spatial quality evaluator (BRISQUE), are introduced for further quantitative comparison. The inception score [29] is used to evaluate the diversity and authenticity of the images generated by the network, and the more varied and distinct the generated images are, the higher the score is. The blind/referenceless image spatial quality evaluator [30] is a no-reference spatial domain image quality assessment metric that scores lower if the generated image is smooth and less noisy.

The experimental results are summarized in Table 2. Although our method is slightly higher than other networks and algorithms in terms of average image generation time, the maximum image resolution is greatly increased. In terms of IS, our method is significantly higher than the methods based on image editing, AutoEncoder, and GAN, which proves that the introduction of image stylization operations can increase the diversity of the generated images. Additionally, our method is superior to the rest of style transfer algorithms in IS, which further proves that our method can generate more diverse and natural-looking tile images. As for BRISQUE, the accuracy of our method is lower than [4], [6] and other style transfer methods [10], [19], [26], but slightly higher than [8], [12], [14], [28], and the image quality needs to be further improved.

In conclusion, the above experiments prove that our method is not only capable of controlling the main texture of the output tile image, but also able to achieve image

TABLE 1. Comparison of the image quality evaluation factors to different networks/algorithms.

Category	Algorithm/ Network	Generate image diversity	Degree of retention of major texture features	Style transfer effect	Generated image size	Image qualityafter superscaling
Image editing	pixel-cnn [4]	Medium	High	——	Low	Low
	DIP [6]	Medium	High	——	Low	Medium
AutoEncoder/GAN	UNet [8]	High	Medium	——	Medium	Low
	Style-GAN [12]	High	Medium	——	Medium	Low
	DC-GAN [14]	High	Medium	Low	Medium	Low
	GAN-Sketching [26]	High	High	——	Medium	Low
Style Transfer	Style Transfer [10]	High	Low	Low	Low	Low
	MGANs [28]	High	Low	Low	Low	Low
	Texture Synthesis [19]	High	Low	Medium	Medium	Medium
Ours		High	High	High	High	High

TABLE 2. Comparison of the quantitative metrics among different networks/algorithms.

Category	Algorithm/Network	Maximum image resolution(px×px)	Average image generation time(s)				IS	BRISQUE
			256	1024	4096	16384		
Image editing	pixel-cnn [4]	512×512	0.85	——	——	——	2.34±0.09	29.83
	DIP [6]	512×512	0.73	——	——	——	2.32±0.10	30.11
AutoEncoder/GAN	UNet [8]	1024×1024	0.54	2.16	——	——	2.25±0.11	24.79
	Style-GAN [12]	1024×1024	0.49	1.72	——	——	2.28±0.13	25.63
	DC-GAN [14]	1024×1024	0.22	**1.08**	——	——	2.22±0.09	**23.27**
	GAN-Sketching [28]	1024×1024	**0.13**	1.69	——	——	1.92±0.12	33.71
Style Transfer	Style Transfer [10]	512×512	150.13	——	——	——	2.98±0.11	42.21
	MGANs [26]	512×512	122.06	——	——	——	3.12±0.13	31.99
	Texture Synthesis [19]	4096×4096	0.67	2.46	**34.7**	——	3.09±0.12	31.63
Ours		**16384×16384**	1.15	2.98	34.94	**47.72**	**3.45±0.23**	29.54

stylization and super-resolution. It can be applied to practical designs, and provides a good reference for designers to design customized tiles.

V. CONCLUSION

In this paper, we have proposed a multi-stage tile generation algorithm based on generative adversarial network, and designed the corresponding networks to cleverly divide tile generation into three stages: image block generation, image stylization, and image super-resolution and magnification, to realize the generation of high-quality tile image styles. The relevant experiments further demonstrate that our method can not only ensure the generation of high-quality tile images in a relatively short period of time, but also consider human interaction factors to a certain extent, to maintain a certain degree of controllability in the main texture and style content of the generated tile images, which is in line with people's psychological expectations.

There is still room for further improvement in our method. On the one hand, the speed and efficiency of the model can be further improved to meet the real-time interaction with the designer. On the other hand, the controllability of texture and style can still be further enhanced. For example, semantic information can be added to our network, so in the process of tile images generation, designers can freely choose the image style and carve out the main texture structure of the image. Finally, in the future research, we hope to improve the versatility of our method and extend it to other deep learning network architectures [31], [32], [33].

REFERENCES

[1] C. Sterken and J. Manfroid, "Color transformation," in Astronomical Photometry: A Guide (C. Sterken and J. Manfroid, eds.), Astronomical Photometry: A Guide, pp. 119–125, Dordrecht: Springer Netherlands, 1992.

[2] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, "Real-time texture synthesis by patch-based sampling," ACM Transactions on Graphics (ToG), vol. 20, no. 3, pp. 127–150, 2001.

[3] C. Wang, C. Mo, and W. Wu, "Textile pattern design using wang's tile algorithm," Journal of Engineering Graphics, vol. 29, no. 5, pp. 56–61, 2008.

[4] A. v. d. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with pixelcnn decoders," in Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, (Red Hook, NY, USA), p. 4797–4805, Curran Associates Inc., 2016.

[5] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2536–2544, 2016.

[6] X. Pan, X. Zhan, B. Dai, D. Lin, C. C. Loy, and P. Luo, "Exploiting deep generative prior for versatile image restoration and manipulation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[7] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," Journal of The American Statistical Association, vol. 112, no. 518, pp. 859–877, 2017.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 234–241, Springer, 2015.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, (Cambridge, MA, USA), p. 2672–2680, MIT Press, 2014.

[10] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," arXiv preprint arXiv:1508.06576, 2015.

[11] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," arXiv preprint arXiv:1710.10196, 2017.

[12] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4401–4410, 2019.

[13] A. Karnewar and O. Wang, "Msg-gan: Multi-scale gradients for generative adversarial networks," in Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7799–7808, 2020.

[14] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1125–1134, 2017.

[16] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2223–2232, 2017.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[18] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in European Conference on Computer Vision, pp. 694–711, Springer, 2016.

[19] Y. Zhou, Z. Zhu, X. Bai, D. Lischinski, D. Cohen-Or, and H. Huang, "Non-stationary texture synthesis by adversarial expansion," ACM Trans. Graph., vol. 37, jul 2018.

[20] K. Turkowski, "Filters for common resampling tasks," in Graphics Gems (A. S. Glassner, ed.), Graphics Gems, pp. 147–165, San Diego: Morgan Kaufmann, 1990.

[21] P. Getreuer, "Linear methods for image interpolation," Image Processing On Line, vol. 1, pp. 238–259, 2011.

[22] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in Proceedings of The European Conference on Computer Vision (ECCV), pp. 517–532, 2018.

[23] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in Proceedings of The IEEE International Conference on Computer Vision, pp. 1501–1510, 2017.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.

[25] P. Wang, Y. Li, and N. Vasconcelos, "Rethinking and improving the robustness of image style transfer," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 124–133, 2021.

[26] S.-Y. Wang, D. Bau, and J.-Y. Zhu, "Sketch your own gan," in Proceedings of the IEEE/CVF International Conference on Computer Vision(ECCV), pp. 14050–14060, 2021.

[27] J. Yu and L. Zhao, "A novel deep cnn method based on aesthetic rule for user preferential images recommendation," Journal of Applied Science and Engineering (Taiwan), vol. 24, pp. 49–55, 01 2021.

[28] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in Computer Vision – ECCV 2016 (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 702–716", Springer International Publishing, 2016.

[29] J. Sun and B. Zhang, "Mca-gan: Text-to-image generation adversarial network based on multi-channel attention," in 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), vol. 1, pp. 1845–1849, 2019.

[30] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," IEEE Transactions on Image Processing, vol. 21, no. 12, pp. 4695–4708, 2012.

[31] S. Yin, H. Li, and L. Teng, "Airport detection based on improved faster rcnn in large scale remote sensing images," Sensing and Imaging, vol. 21, no. 1, p. 49, 2020.

[32] S. Yin, H. Li, L. Teng, M. Jiang, and S. Karim, "An optimised multi-scale fusion method for airport detection in large-scale optical remote sensing

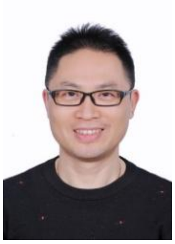
images," International Journal of Image and Data Fusion, vol. 11, no. 2, pp. 201–214, 2020.

[33] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in Advances in Neural Information Processing Systems (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 6840–6851, Curran Associates, Inc., 2020.

ABBREVIATIONS AND ACRONYMS

TABLE 3. Abbreviations and acronyms

Abbreviations	Detailed Definition
GAN	Generative Adversarial Network
VGG	Visual Geometry Group Convlutional Network
VGG19	Visual Geometry Group with 19 Network Layers
AdaIN	Adaptive Instance Normalization
ProGAN	Progressive Growing of Generative Adversarial Network
StyleGAN	A Style-Based Generator Architecture for Generative Adversarial Network
MSA	Multi-Scale Attention
MSA-GAN	Multi-Scale Attention Generative Adversarial Network
MSA-StyleGAN	StyleGAN with Multi-Scale Attention
SWAG	Stylization With Activation smoothinG
TS-GAN	Texture Synthesis Generative Adversarial Network
SWAG-TS-GAN	Stylization With Activation smoothinG Texture Synthesis Generative Adversarial Network



JIANFENG LU received his B.S. and M.S. degrees from the department of Mechanical Engineering in 1998 and 2001 respectively, and his Ph.D. degree in computer science department from the Zhejiang University in 2005. He works in the department of computer science in Hangzhou DIANZI University from 2005. His main research interests are in the areas of digital video and image processing and scientific visualization.

MENGTAO SHI was born in Linhai, Zhejiang, China. He is currently pursuing his M.S. degree at Hangzhou Dianzi University. During his master's degree, he is mainly responsible for the completion of National Natural Science Foundation Project. His current research interests include image generation, deep learning, and digital watermarking.

YUHANG LU was born in Honghu, Hubei, China. He is an undergraduate student at Hangzhou Dianzi University. His current research interests include image processing, deep learning.

CHING-CHUN CHANG received the Ph.D. degree in computer science from the University of War wick, Coventry, U.K., in 2019. He participated in a short-term scientific mission supported by European Cooperation in Science and Technology Actions with the Faculty of Computer Science, Otto von Guericke University Magdeburg, Germany, in 2016. He was granted the Marie-Curie Fellowship and participated in a research and innovation staff exchange scheme supported by Marie Skłodowska Curie Actions with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, USA, in 2017. He was a Visiting Scholar with the School of Computer and Mathematics, Charles Sturt University, Australia, in 2018, and the School of Information Technology, Deakin University, Australia, in 2019. He was a Research Fellow with the Department of Electronic Engineering, Tsinghua University, China, in 2020. He has been a Post-Doctoral Fellow with the National Institute of Informatics, Japan, since 2021. His research interests include steganography, watermarking, forensics, biometrics, cybersecurity, applied cryptography, image processing, computer vision, natural language processing, computational linguistics, machine learning, and artificial intelligence

PROF.LI LI come from Department of Computer Science of Hangzhou Dianzi University. She got PhD in Computer Science of Zhejiang University, CN, 2004; MSc in Applied Mathematics of Dalian University of Technology, CN, 1997; BSc in Applied Mathematics, Dalian University of Technology, CN, 1994.Researchon Information Forensics Security, Multimedia watermarking, Pattern Recognition, Machine Learning; In recent ten years, Li Li has engaged in the research of digital watermarking technology, and has obtained innovative and original outcomes, which have been protected through 10 patents and widely used in the industry. The series of achievements have led to the inauguration of of Hangzhou Dianzi University Shangyu science and Engineering Research Institute Co., Ltd in 2018, which is a joint commercial venture between Hangzhou Dianzi University and Shaoxing government. She has more than 10 invention patents, more than 20 software registration rights, more than 30 provincial and local enterprise projects. She also published extensively with more than 135 widely cited papers in her credit.

RUI BAI received the B.S. degree in computer science and technology from the Ankang University, Ankang, China, in 2014, the M.S. degree in computer application technology from North Minzu University, Yinchuan, China, in 2017. She is currently pursuing the Ph.D. degree in computer science and technology with Hangzhou Dianzi University. Her research interests include image processing, deep learning, image watermarking, and data hiding.