electronics

Article

TOLGAN: An End-To-End Framework for Producing Traditional Orient Landscape

Booyong Kim [1], Heekyung Yang [2,*,†] and Kyungha Min [1,*,†]

[1] Department of Computer Science, Sangmyung University, Seoul 03016, Republic of Korea; hapata1120@gmail.com
[2] Department of Software, Sangmyung University, Chonan 31066, Republic of Korea
[*] Correspondence: yanghk@smu.ac.kr (H.Y.); minkh@smu.ac.kr (K.M.); Tel.: +82-41-550-5351 (H.Y.); +82-2-2287-5377 (K.M.)
[†] These authors contributed equally to this work.

Abstract: We present TOLGAN that generates traditional oriental landscape (TOL) image from a map that specifies the locations and shapes of the elements composing TOL. Users can create a TOL map by using a user interface or a segmentation scheme from a photograph. We design the generator of TOLGAN as a series of decoding layers where the map is applied between the layers. The generated TOL image is further enhanced through an AdaIN architecture. The discriminator of TOLGAN processes a generated image and its groundtruth TOL artwork image. TOLGAN is trained through a dataset composed of paired TOL artwork images and their TOL maps. We present a tool through which users can produce a TOL map by specifying and organizing the elements of TOL artworks. TOLGAN successfully generates a series of TOL images from the TOL map. We evaluate our approach using a quantitative way by estimating FID and ArtFID scores and a qualitative way by executing two user studies. Through these studies, we prove the excellence of our approach by comparing our results with those from several important existing works.

Keywords: traditional oriental landscape (TOL); GAN; SPADE; generator; discriminator

Citation: Kim, B.; Yang, H.; Min, K. TOLGAN: An End-To-End Framework for Producing Traditional Orient Landscape. *Electronics* **2024**, *13*, 4468. https://doi.org/10.3390/electronics13224468

Academic Editors: Krzysztof Szczypiorski and Ping-Feng Pai

Received: 24 October 2024
Revised: 12 November 2024
Accepted: 12 November 2024
Published: 14 November 2024

1. Introduction

Traditional orient landscape (TOL) is a fine art technique beloved in North-east Asian countries including Korea, China and Japan. The artists in those countries have created various landscapes using brushes with black ink. They express objects in a landscape such as mountains, trees, rocks, cloud, water and buildings in their own abstract shape. The strokes that express salient shape of the objects spread black ink on the paper. Since many well-known TOL artworks are created hundreds years ago, most of the artworks turn gray and yellow.

In computer graphics and computer vision society, many researchers have presented various computational models that produce stroke patterns of TOL brushes [1–10]. They build the skeleton of the strokes using controllable curve and the shape of the strokes using offset from the skeleton curves. Some of them exploit physically-based models for simulating the spread of ink on the surface of the paper. However, they encounter an important limitation in expressing the abstraction and distortion of the objects in TOL.

The progressive of deep learning has presented various schemes that produce images with various styles from the input photographs. These schemes analyze the style from the sampled artworks and apply them to the target photos. The styles are transferred through a texture structure such as Gram matrix or a set of loss functions. These schemes successfully transfer the styles from many famous artworks such as Starry Nights by Gogh, Scream by Munch and Portraits by Picasso to an arbitrary input photograph. These schemes, however, have limitations in transferring TOL style to a photograph. Since TOL artworks are expressed by strokes and spaces, the schemes that transfer texture-based styles are not effective for TOL artworks.

We present a deep learning-based approach for producing TOL artwork images from a TOL map that specifies the shape and configuration of the components of TOL images. TOL map is generated from users or input photographs. The components of a TOL map including sky, land, water, mountains or grasses, etc., is denoted as TOL components. We collect many TOL artwork images and analyze the components of the images. Then, we segment the images into the components. Afterwards, we devise a deep learning-based framework including GAN architecture and AdaIN network that can produce TOL image from a user-initiated map. The GAN architecture is trained using the dataset composed of the pairs of TOL image and its map. The result of GAN architecture can successfully mimic the style of TOL artwork, but it suffers from unclear strokes. To address this limitation, we further apply AdaIN network to enhance the strokes. Our framework can produce visually-convincing TOL artwork images from user-initiated map or an input landscape photograph.

A key advantage of our framework is its ability to generate visually pleasing TOL artwork styles using user-initiated TOL maps effectively. While generating TOL artwork styles has been challenging, our approach successfully produces high-quality TOL artwork styles.

Another advantage lies in the efficient training process achieved by leveraging appropriate components of TOL artwork images. The complex and abstract nature of TOL artwork images restricts effective training of the existing TOL generation models. To address this challenge, our framework utilizes TOL artwork components extracted from experts as training data, which enables successful training with just 253 samples.

Finally, our approach does not require users to create a detailed TOL map for generating TOL artwork images. Furthermore, the TOL map extracted from an input image is not required to show exact contour. Note that the TOL artwork technique prioritizes abstract representation by simplifying and compressing subjects, emphasizing their approximate shape and position over precise details. Therefore, verifying the accuracy of TOL maps is not a critical concern in our framework. This property eases the generation of TOL map, which is an input to our framework.

This paper is organized as follows. In Section 2, we briefly review the existing works for generative models and TOL image generation models. We explain our TOL generation model in Section 3. In Section 4, we describe the preparation of the TOL image dataset and the training process. We present the implementation details and results and evaluate our results in Section 5 and Section 6, respectively. Finally, we conclude this paper and suggest future study in Section 7.

2. Related Work

In computer graphics society, several studies have been proposed for simulating traditional oriental paintings [1–10]. Recently, the progress of deep learning techniques presents various schemes for producing traditional oriental paintings.

2.1. Deep Learning-Based General Approach

Gatys et al. [11] presented a pioneering work that transfers a texture-based style captured from a source image to the content of a target image. This scheme can be applied to transfer TOL style to a photograph. Ulyanov et al. [12] and Huang et al. [13] improved Gatys et al.'s work to present a stable and efficient style transfer. These texture-based style transfer schemes are very effective for preserving the content of the target image, but they suffer from improper transfer of the TOL stroke patterns, which are very unique characteristics of TOL artworks.

Generative adversarial network (GAN) [14] presents a very influential framework for synthesizing various styles on images. Mizra and Osindero [15] presented a conditional GAN that considers user-specified conditions for controlling styles and contents in the produced images. Isola et al. [16] presents pix2pix, one of effective schemes that produce style transfer between two domains. Later, Zhu et al. [17] presented CycleGAN that introduces cycle consistency loss to resolve the constraint of paired training dataset of pix2pix framework. Park et al. [18] continued this approach by introducing SPADE that controls the constraints of style transfer using a region map that specifies the contents to produce. Another direction is MUNIT, which was presented by Huang et al. [19], that translates images in multi domains. These schemes present a general-purpose style transfer framework between images of multiple domains. Therefore, they can be employed for producing images of TOL style. However, they have a limitation that the style they synthesize is produced only mimicking stroke patterns of TOL artworks. They cannot synthesize the deformation of objects, which is frequently observed in many TOL artworks.

The general approaches applied for producing traditional oriental paintings show a serious limitation in producing visually pleasing quality, since they does not consider the unique stroke patterns of oriental paintbrush observed in the traditional oriental paintings. They also do not consider the difference of the stroke patterns on different objects in the traditional oriental paintings.

2.2. Deep Learning-Based TOL Generation Approach

Li et al. [20] presented a framework that transfers input photograph into Chinese traditional painting style. The abstraction of Chinese traditional painting is trained and expressed through modified xDoG filters, which further process the result images of the generator. The results of the modified xDoG filters are processed through a discriminator with three loss functions including a morphological filter loss. Lin et al. [21] presented a multi-scale GAN model that transfers sketches to Chinese paintings. The generator of this model is trained with L1 and adversarial losses. By adding an edge detector, this model can operate a style generation similar to that of neural style transfer. This model successfully abstracts thick lines of the Chinese paintings, but they cannot properly express the spread of ink.

He et al. [22] presented ChipGAN that produces Chinese ink wash painting style from a photograph. They apply a brush stroke constraint on a generative model to extract an edge-based representation, which is further transferred to a Chinese ink wash painting style using an adversarial loss. They also apply a cycle consistence loss for the input photograph and the reconstructed photograph from the result of the generator. The result image of the generator is further eroded and blurred with ink wash loss for the final result. Zhou et al. [23] presented Shanshui-DaDA, an interactive and generative model that produces Chinese shanshui painting style using CycleGAN. Users of this model are presented with a web-based interface through which users draw sketches for the Shanshui-DaDA. Xue [24] presented SAPGAN that combines a couple of GAN structures including sketch GAN and paint GAN. Sketch GAN produces an edge map from an input image through either Relative Average Square GAN or StyleGAN2 model, and paint GAN executes edge-to-paint translation using either pix2pix, pix2pixHD or SPADE. The paint GAN is trained with a paired dataset of real Chinese paintings and their corresponding edge maps.

Hung et al. [25] presented UTGAN, which extends CycleGAN for transforming images from very different two image sets. They employ UTGAN for transforming two different image sets such as portraits and Shan-Shui paintings. They devised a special loss term for their UTGAN in order to transform very different image sets. However, they do not support the transformation for the landscape photos and traditional oriental landscapes.

Chung and Huang [26] presented a traditional Chinese painting using a cycle-consistent GAN with pix2pix. They presented a border enhancing scheme in order to enhance the details of border images, which improves the quality of resulting images. However, they suffer from obscure stroke patterns inside an object whose border is rendered using enhanced strokes by their scheme.

The deep learning-based TOL models preserve the stroke patterns of traditional oriental paintings to their input photographs. However, these models have a limitation in applying different stroke patterns to the objects in a scene. We resolve this limitation by sampling the stroke patterns for the objects and applying them to the appropriate objects, which presents very visually pleasing quality.

3. TOL Generation Framework

3.1. Overview

The overall structure of our TOL generation framework is presented in Figure 1. The input to our model is a TOL map, which is either initiated from users or extracted from an input photograph through an image segmentation model. The input to the generator is a latent vector Z and the TOL map, which is inserted to each SPADE layer of the architecture. Note that Z is 512 dimension, which is used in GauGAN [18]. Similar to GauGAN, Z is constructed from normal Gaussian distribution. The result of the generator mimics the style of TOL artwork image, which is further processed through AdaIN architecture to emphasize and to enhance the salient strokes of TOL brushes. The generated TOL images are processed through a discriminator. The overview of our framework is presented in Figure 1.

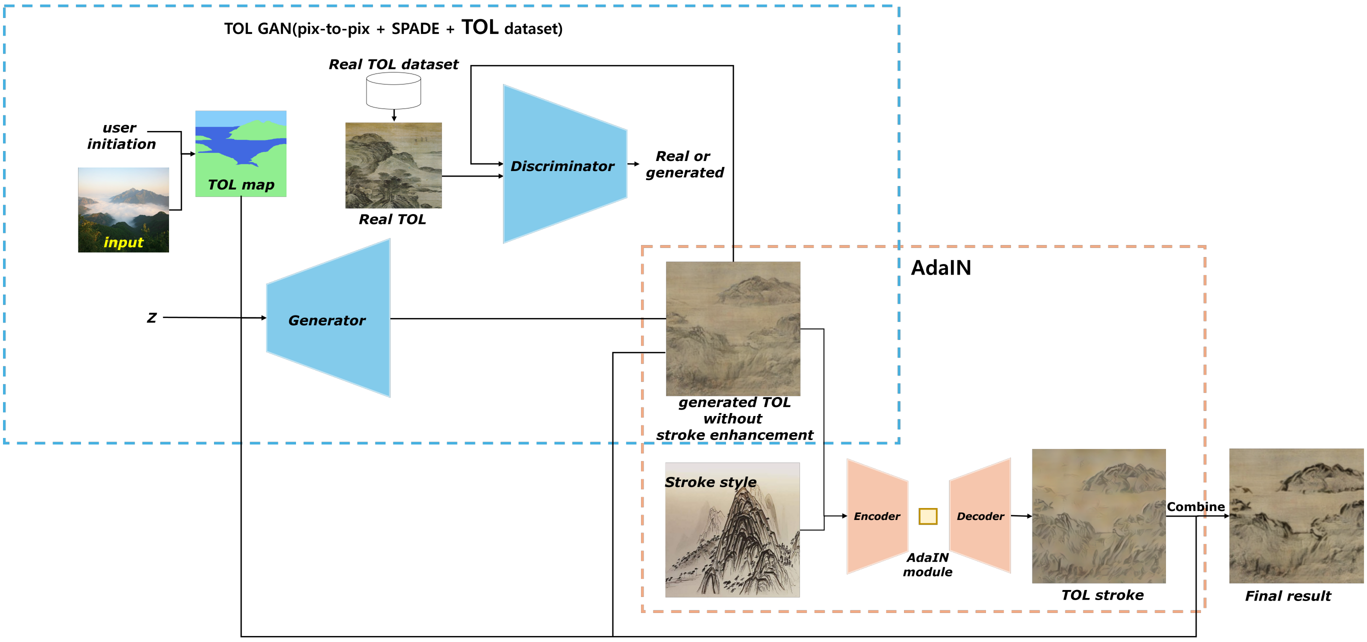


Figure 1. An overview of our TOL generation framework.

3.2. Model Architecture

3.2.1. Generator

Our generator consists of two main components: the TOL generator and the AdaIN module. The TOL Generator takes a latent vector Z and a TOL map as inputs to produce a TOL image. The latent vector enables the generation of diverse TOL images, while the TOL map specifies the configuration of TOL map components.

Our generator is constructed using a combination of convolutional layers, SPADE modules, skip connections, upsampling operations, and the LeakyReLU activation function. The SPADE module at each stage consistently takes the TOL map as input. The SPADE module executes a spatial normalization where the pixels in each TOL components are normalized independent to the pixels belong to different components. This SPADE module is very effective in generating images, which are composed of several components. Additionally, skip connections are applied at every corresponding stage. The arthitecture of our generator is described in Figure 2.

While the output of the TOL generator successfully replicates the general style of TOL artwork, it falls short in accurately expressing the distinctive stroke effects, which is a key characteristic of TOL artwork. To address this limitation, we incorporate the AdaIN module, which applies the stroke style of TOL artwork to the TOL image. This module is pre-trained to enhance image boundaries and increase scene contrast. By integrating the AdaIN module, we achieve the generation of TOL images that effectively capture both the texture and stroke effects unique to TOL artwork.

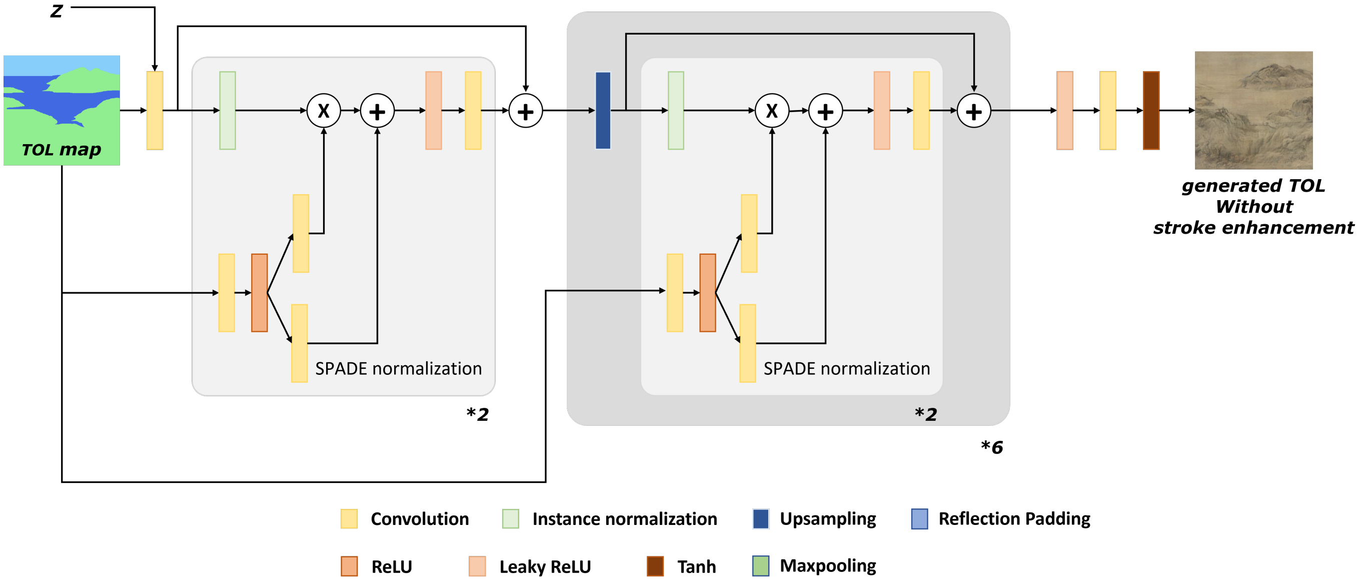


Figure 2. The architecture of our generator. *n indicates that the same structure consists of n components.

3.2.2. Discriminator

Our discriminator is designed to differentiate between the final TOL images generated by the generator and real TOL artwork images, which is collected in our TOL artwork image dataset, while a generated TOL image is the result of our generator.

The detailed architecture of our discriminator is depicted in Figure 3. The Discriminator is composed of a combination of convolutional layers, the LeakyReLU activation function, and instance normalization.

The detailed configuration of our generator and discriminator is illustrated in Figure 4.

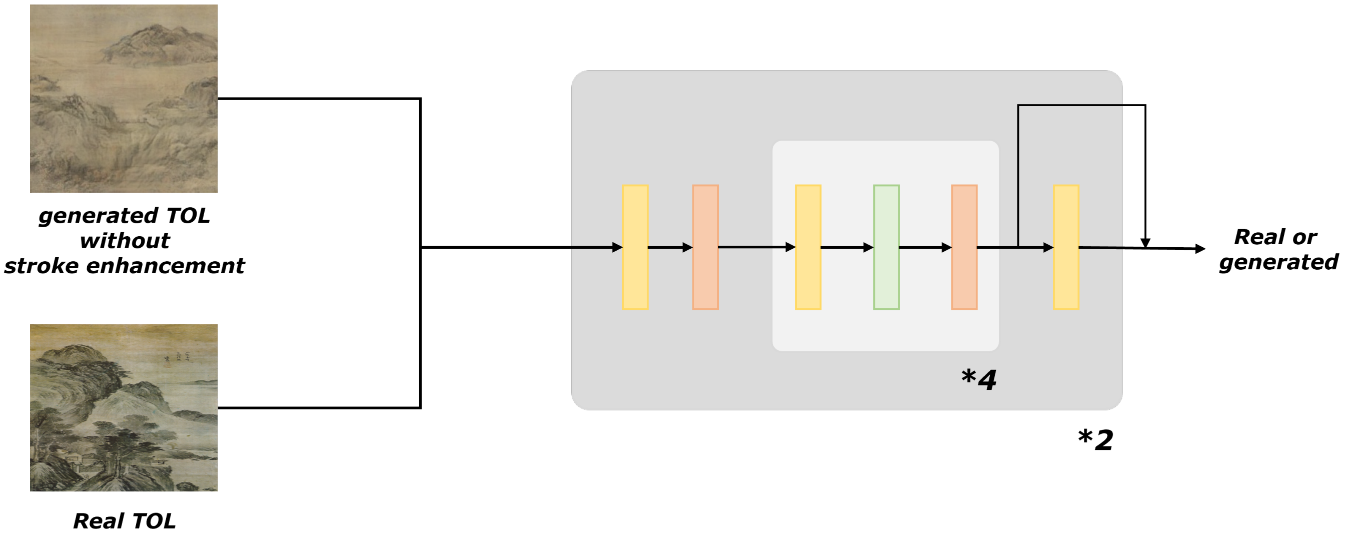
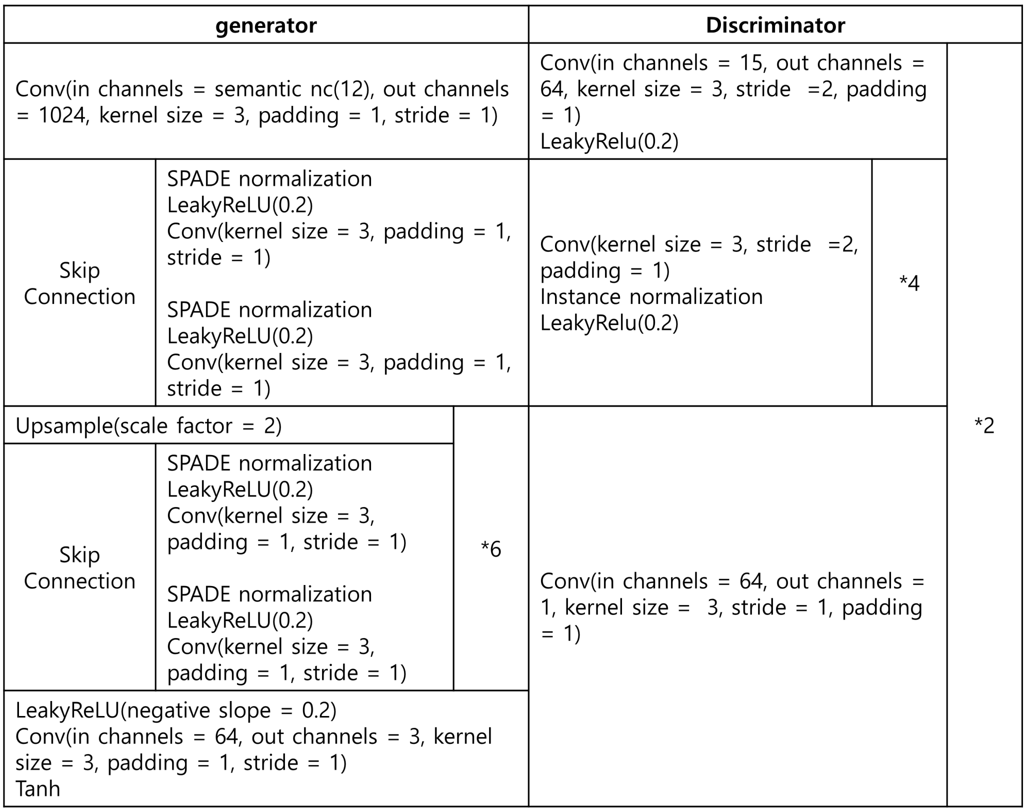


Figure 3. The structure of our discriminator. *n indicates that the same structure consists of n components.



<table>
<tr><th colspan="3">generator</th><th colspan="3">Discriminator</th></tr>
<tr><td colspan="3">Conv(in channels = semantic nc(12), out channels = 1024, kernel size = 3, padding = 1, stride = 1)</td><td colspan="2">Conv(in channels = 15, out channels = 64, kernel size = 3, stride =2, padding = 1)
LeakyRelu(0.2)</td><td rowspan="5">*2</td></tr>
<tr><td>Skip Connection</td><td colspan="2">SPADE normalization
LeakyReLU(0.2)
Conv(kernel size = 3, padding = 1, stride = 1)
SPADE normalization
LeakyReLU(0.2)
Conv(kernel size = 3, padding = 1, stride = 1)</td><td>Conv(kernel size = 3, stride =2, padding = 1)
Instance normalization
LeakyRelu(0.2)</td><td>*4</td></tr>
<tr><td colspan="2">Upsample(scale factor = 2)</td><td rowspan="2">*6</td><td colspan="2" rowspan="3">Conv(in channels = 64, out channels = 1, kernel size = 3, stride = 1, padding = 1)</td></tr>
<tr><td>Skip Connection</td><td>SPADE normalization
LeakyReLU(0.2)
Conv(kernel size = 3, padding = 1, stride = 1)
SPADE normalization
LeakyReLU(0.2)
Conv(kernel size = 3, padding = 1, stride = 1)</td></tr>
<tr><td colspan="3">LeakyReLU(negative slope = 0.2)
Conv(in channels = 64, out channels = 3, kernel size = 3, padding = 1, stride = 1)
Tanh</td></tr>
</table>

Figure 4. The structure of our model. *n indicates that the same structure consists of n components.

3.2.3. Loss Function

The loss functions we employed for TOLGAN are feature matching loss and hinge loss. The feature matching loss L_{FM} is defined as follows:

$$L_{FM} = \sum_i w_i ||VGG19_i(y) - VGG19_i(G(z))||^2,$$

where y belongs to TOL artwork images and $G(z)$ belongs to generated TOL images. i denotes the i-th layer of the VGG19 network.

The hinge loss L_H is defined as follows:

$$L_H = -E_{z \sim p_z, y \sim p_{data}}[min(0, -1 + D(D(z), y))] - E_{z\ p_z, y \sim p_{data}}[D(G(z), y)].$$

4. Training

4.1. Building TOL Image Dataset

The first step of our building dataset is to collect TOL artwork images. First of all, we collect traditional oriental artwork images from various websites and museum archives as many as possible. As a result, we have collected 18K TOL artwork images including landscape, portrait, and still-life, etc. At the end of the first step, we select landscape artwork images by excluding portrait, still-life and other inadequate low-quality artwork images. Finally, we select 4627 TOL artwork images for our purpose. In this step, we do not include cultural or stylistic considerations during selection. We only consider the quality and resolution of TOL artwork images.

The second step is preprocessing where we apply color correction and resizing for the selected TOL artwork images. The images in our dataset are organized to have 256 × 256 resolution.

We illustrate some of the collected TOL artwork images in Figure 5.

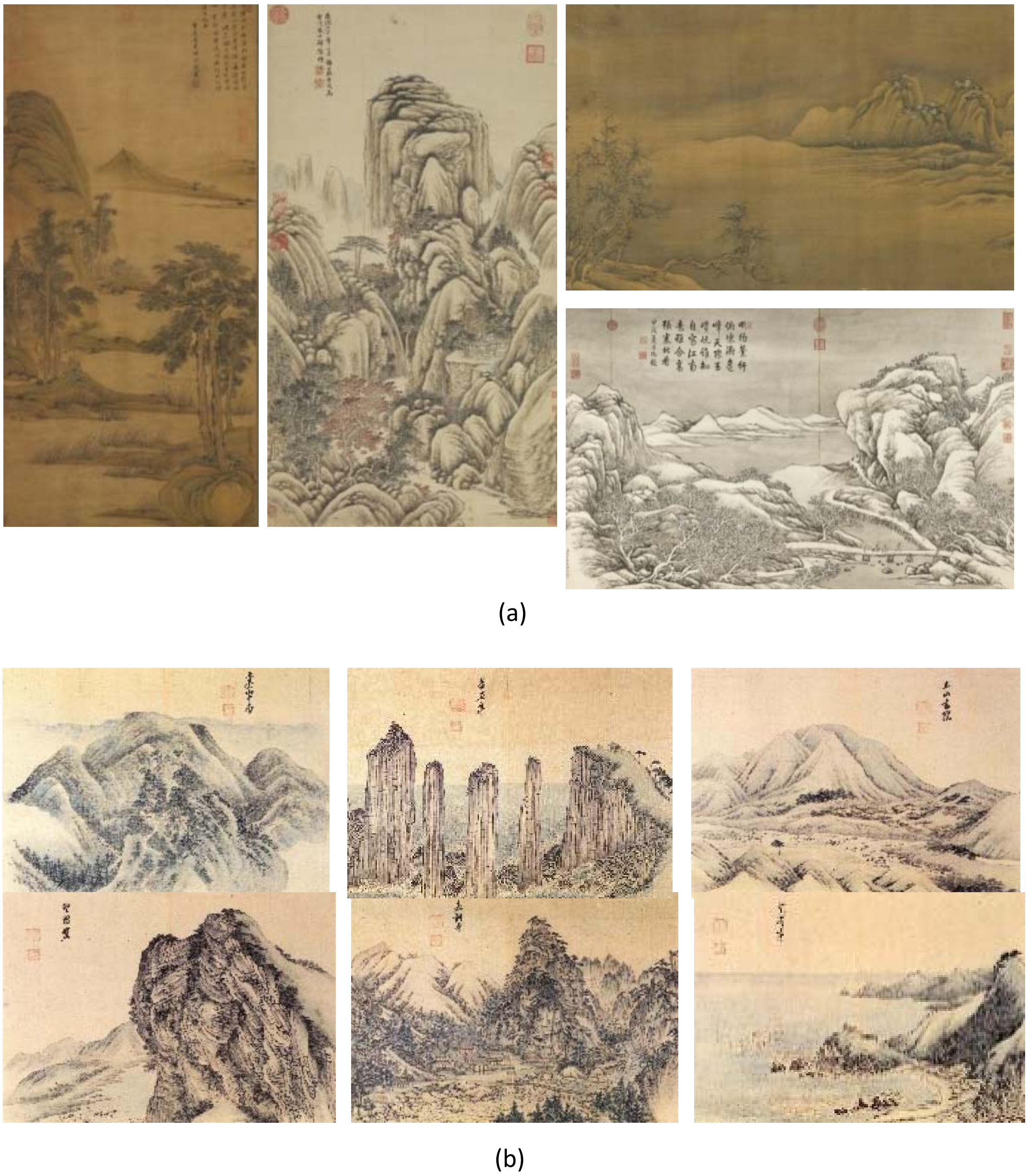


Figure 5. TOL image dataset: (**a**) The images belong to 18K TOL artwork images, (**b**) The selected TOL images.

The third step is TOL map generation. We select the most frequently observed objects in TOL artwork images as sky, close mountain, far mountain, ground, rock, grass, water, close tree and far tree. In many TOL artwork images, characters or artists' seal are observed. These components are excluded. Persons and animals are also excluded, since they are observed very rarely. These objects are denoted as TOL components. A TOL map is composed of these TOL components segmented from a TOL artwork image. We select 253 images among the TOL artwork images and build TOL maps. For this purpose, we implement a TOL map generation tool that segment regions corresponding to TOL components of TOL artwork.

Finally, the TOL artwork images with TOL maps are augmented by flipping, tone control and three-way rotation. Finally, we train our model with 11,418 images.

The process of building TOL image dataset is illustrated in Figure 6.

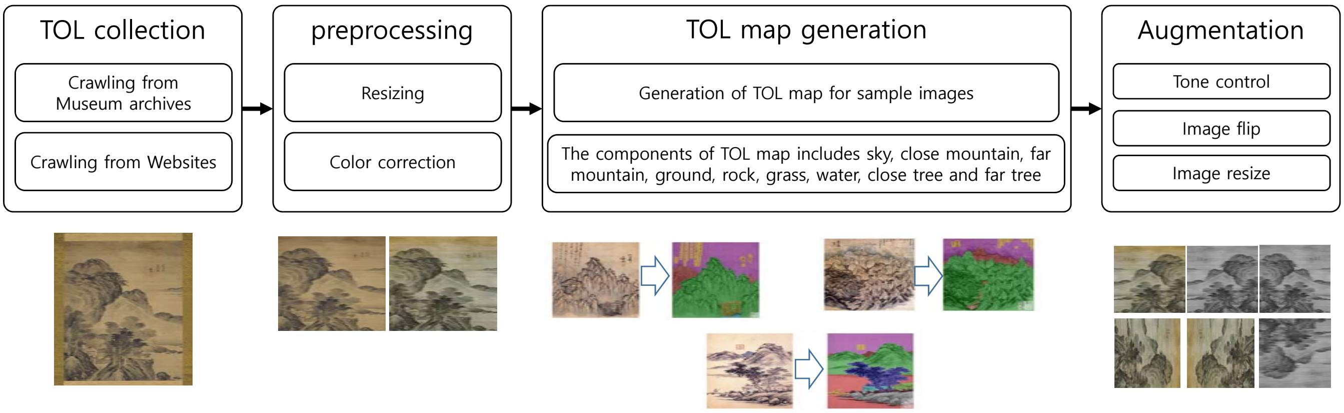

TOL collection
Crawling from Museum archives
Crawling from Websites
preprocessing
Resizing
Color correction
TOL map generation
Generation of TOL map for sample images
The components of TOL map includes sky, close mountain, far mountain, ground, rock, grass, water, close tree and far tree
Augmentation
Tone control
Image flip
Image resize

Figure 6. The process of building dataset.

4.2. Training Our Model

The paired set of real TOL artwork image with their matching TOL map is employed for the training. The discriminator compares the generated TOL image from the TOL map with its matching real TOL artwork image. This training process is executed for 120 epochs. The training process of 120 epochs takes 100 h. We present the graphs of loss values for the loss functions of our model in Figure 7.

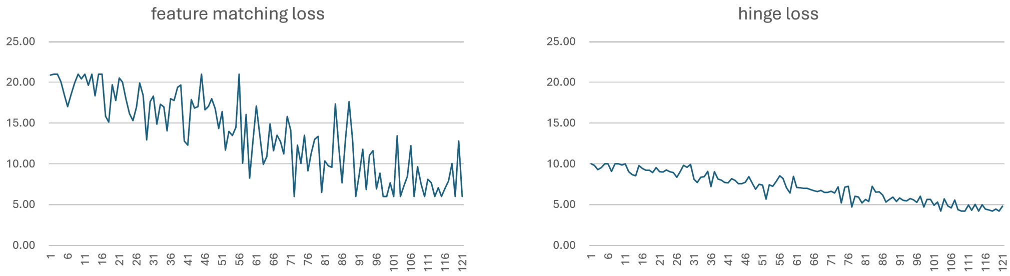


Figure 7. The loss curves of our model.

5. Implementation and Results

We implemented our TOL generation framework on a personal computer with a Pentium i7 CPU, 128 GB main memory and double nVidia RTX3090 GPUs. The software environment is Python 3.9 with PyTorch 1.12.1 and torchvision 0.13.1. We employed Adam optimizer whose β_1 is 0.1 and β_2 is 0.9. The learning rate is 0.0002. Further specific hyperparameters are suggested in Table 1. In compared to many studies that employ high-end GPUs such as A100 or H100, our framework is trained on RTX3090 GPU. Therefore, we claim that our framework shows an efficient traing record.

After the training we implement a tool with a user interface for generating TOL images (See Figure 8). This interface presents nine components of TOL artwork images. The components cover sky, close mountain, far mountain, ground, rock, grass, water, close tree and far tree. Users select the component to insert and draw the shape of components in the left canvas of the interface. After finishing the drawing, they select the resolution of the image to generate using the job shuttle and generate TOL image.

We present twelve TOL map images in Figure 9 and their corresponding TOL images in Figure 10. For the comparison, some of them are generated from photographs.

Table 1. The hyperparameters of our model.

D steps per G	1
batch size	1
optimizer	Adam Adam ($\beta_1 = 0.1$, $\beta_2 = 0.9$)
preprocessing	resize, crop, flip
crop size	256
number of labels	12
parameter initialization	Xavier
GAN mode	Hinge
training scheme	TTUR
discriminator ine epoch	multiscale (no. of subnet = 2, no. of layer = 4) 120
normalization	instance normalization

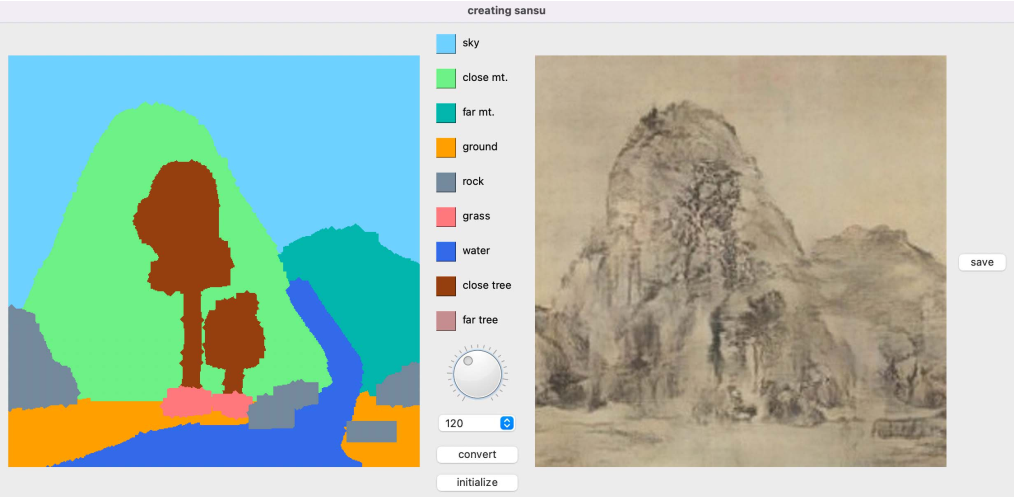


Figure 8. The user interface of our model.

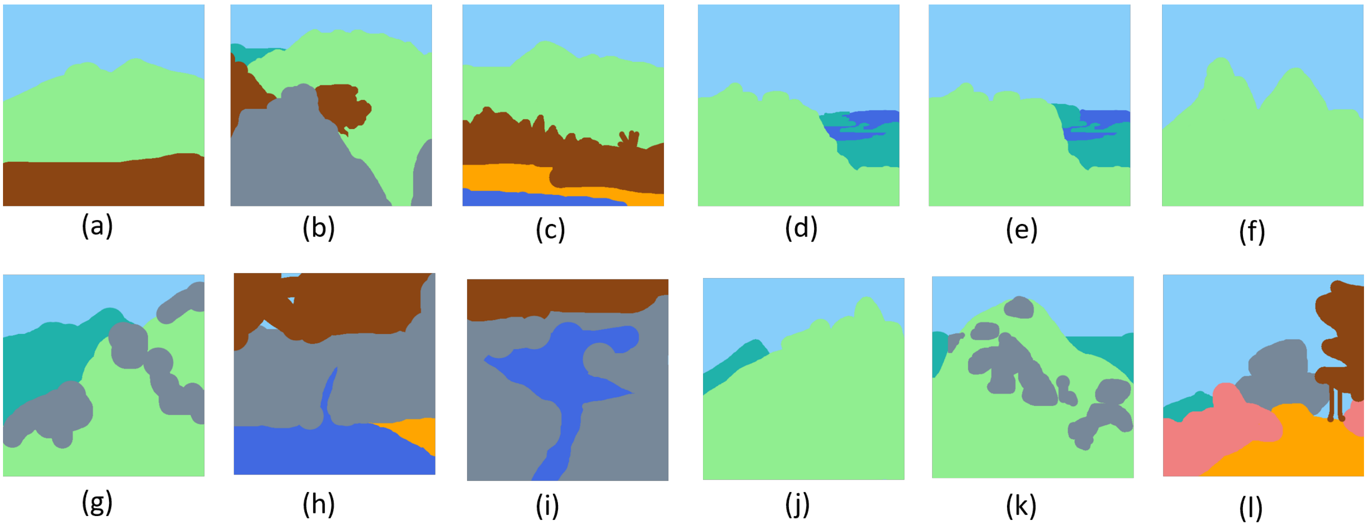


Figure 9. Twelve different input TOL maps (**a**–**l**) drawn by the user for TOL generation.

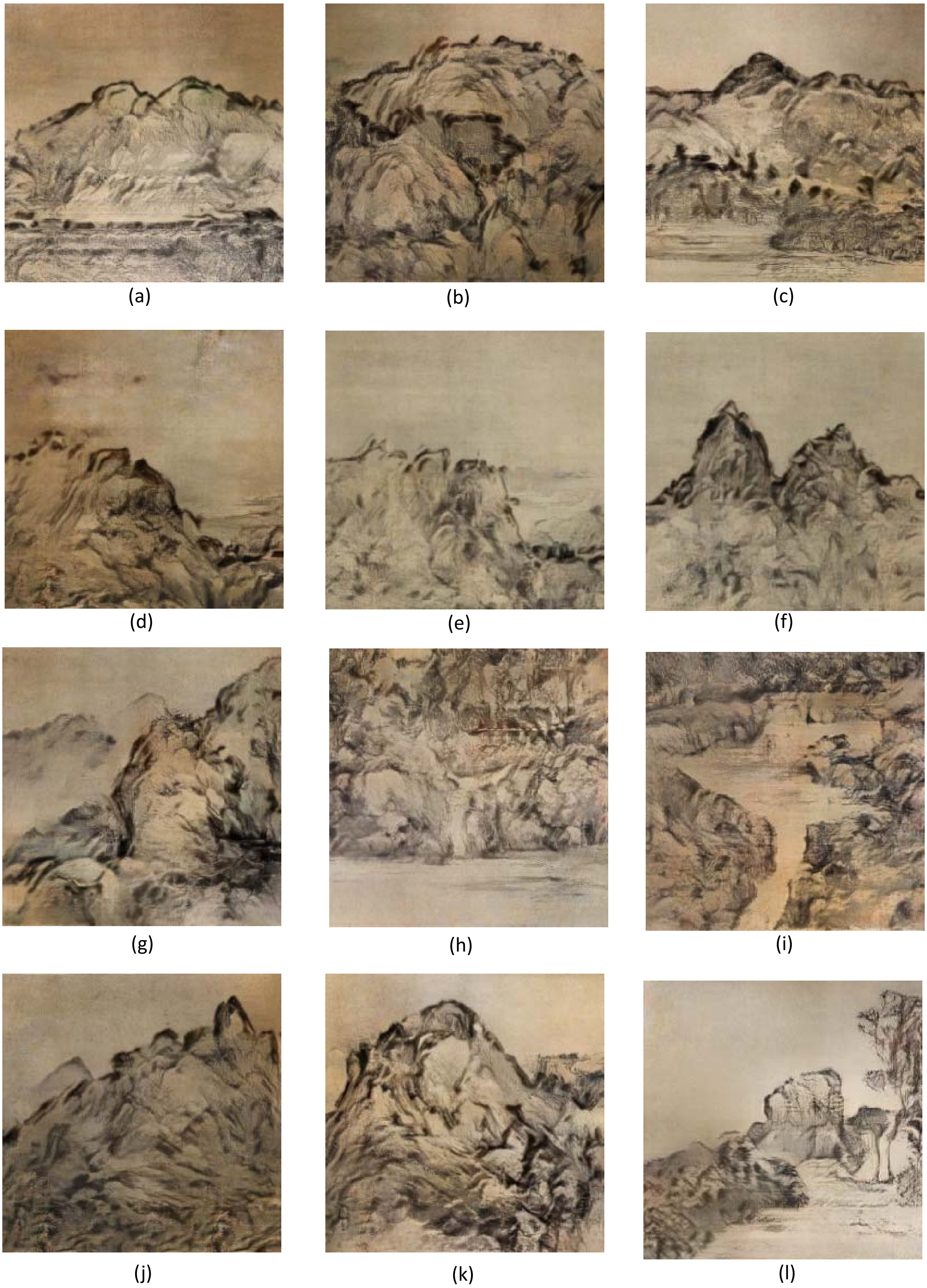


Figure 10. Twelve generated TOL images (**a**–**l**), corresponding to each of the twelve input TOL maps in Figure 9.

6. Analysis

6.1. Comparison

We compare the results from our model with those from several important existing works including MUNIT [19], ChipGAN [22], CycleGAN [17] and neural style transfer (NST) [11]. For a comprehensive comparison, we selected the compared studies from two areas: one is GAN-based study and the other is non GAN-based study. Furthermore, the GAN-based studies can be further categories into general purpose GAN and TOL-specific GAN. MUNIT and CycleGAN are general-purposed GAN-based model that transfers styles between various domains of images. We re-train MUNIT and CycleGAN using the collected TOL images and landscape photographs. ChipGAN is a designated GAN-based model for TOL image generation. Neural style transfer is one of the most pioneering works for applying the styles captured from TOL to a photograph. The results are presented in Figure 11 and are compared in Tables 2–4.

Table 2. The FID and ArtFID scores for the four existing studies and ours. The red figures denote the smallest score.

	MUNIT	CycleGAN	ChipGAN	NST	Ours
FID	213.5	198.4	235.7	187.6	154.5
ArtFID	185.7	156.3	176.3	163.3	131.5

Table 3. The results of a user study for two questions: The red figure denotes best result.

Question	Models	01	02	03	04	05	06	07	08	Sum
1	MUNIT	0	0	0	0	0	0	0	0	0
	ChipGAN	4	6	0	9	3	12	10	12	56
	CycleGAN	0	0	5	5	0	0	4	0	14
	NST	0	4	4	1	2	4	3	2	20
	ours	26	20	21	15	25	14	13	16	150
2	MUNIT	0	0	0	0	0	0	0	0	0
	ChipGAN	3	2	0	8	1	10	7	9	40
	CycleGAN	0	5	5	8	2	3	5	0	28
	NST	6	12	11	3	14	8	8	10	72
	ours	21	11	14	11	13	9	10	16	150

Table 4. The results of a focus group evaluation: The red figure denotes best result.

Models	01	02	03	04	05	06	07	08	Average
MUNIT	2.5	1.8	2.2	2.1	2.2	2.1	2.2	1.9	2.13
ChipGAN	4.2	4.7	3.7	7.2	6.3	7.2	7.0	6.4	5.84
CycleGAN	3.3	4.0	5.8	5.7	2.9	5.4	5.7	1.7	4.31
NST	2.9	5.8	5.2	4.2	6.1	6.3	6.9	4.5	5.24
ours	6.9	7.4	7.6	7.7	7.6	6.5	7.1	6.3	7.14

6.2. Quantitative Evaluation

For the quantitative evaluation of our study, we estimate Frechet Inception Distance (FID) and ArtFID [27] for the TOL images in Figure 11. FID is a widely used metric for evaluating the quality and diversity of images generated by generative models, particularly

GAN. It compares the distributions of real images and generated images by measuring the distance between their feature representations. FID is calculated in the following formula:

$$FID(X, G) = ||\mu_x - \mu_g||^2 + Tr(\Sigma_x + \Sigma_g - 2(\Sigma_x \Sigma_g)^{1/2}),$$

where X and G are real and generated images. We set X as the TOL artwork images and G as the generated TOL images.

ArtFID, which is originally devised to evaluate neural style transfer models, is also employed to estimate our results, since ArtFID is very effective for evaluating images with artistic elements such as texture and brush strokes. FID is calculated in the following formula:

$$ArtFID = \alpha FID(C, G) + \beta FID(S, G),$$

where C is a content image, S, a style image, and G, a generated image. In our study, C is the input photograph, S is a TOL artwork image, and G is the generated TOL image. We set α and β as 0.5, respectively.

We estimate both FID and ArtFID for the results from four existing schemes including MUNIT, CycleGAN, ChipGAN and NST are compared with ours for the eight input landscape images. The FID and ArtFID values from the results of four existing schemes and ours are suggested in Table 2.

In Table 2, our results show lower FID and ArtFID scores than those from the existing studies. Since our results preserve the TOL texture and stroke patterns better than those from the existing studies, ours show lower FID and ArtFID scores.

6.3. Qualitative Evaluation

We evaluate our results in comparisons of two viewpoints: one comparison is with the results from the existing works and the other comparison is with real TOL artwork images. For this purpose, we design two tests: a user study and a focus group evaluation. The participants for our user study are selected from ordinary people who do not have special backgrounds on TOL artworks, and the participants for our focus group evaluation is selected from those who have backgrounds on TOL artworks.

6.3.1. User Study

We hire thirty volunteer participants for a user study. nineteen of them are in their twenties and eleven of them are in their thirties. seventeen of them are female and thirteen are male. They are required to have not special backgrounds on TOL painting artworks.

We have two questions for the images in Figure 11.

1. Question 1 (Quality of the generated image): Among the five result images, select one image that most resembles TOL style.
2. Question 2 (Preservation of input photograph): Among the five result images, select one image that most resembles the input image.

Each participant marks one model for the eight images in Figure 11. We summarize the marks from the participants and present the results in Table 3. For question 1, the scores of our model are higher than those of the other models for all sample images. For question 2, our model records higher scores for five images among eight sample images. Even though our model outperforms for generating TOL style, some of the models show better results for preserving shapes of the objects.

We further execute Chi-square test to prove that our results are preferred over other results in a statistic way. Null hypothesis (H_0) is that there is no evidence that our model is preferred over other models. and Alternative hypothesis (H_1) is that our model is preferred over ours. After executing the Chi-square test, the p-value for Question 1 is 0.0000169 and for Question 2 is 0.0000941. Since the p-values are less than 0.05, we can reject H_0. This indicates that our model is statistically and significantly preferred over other models.

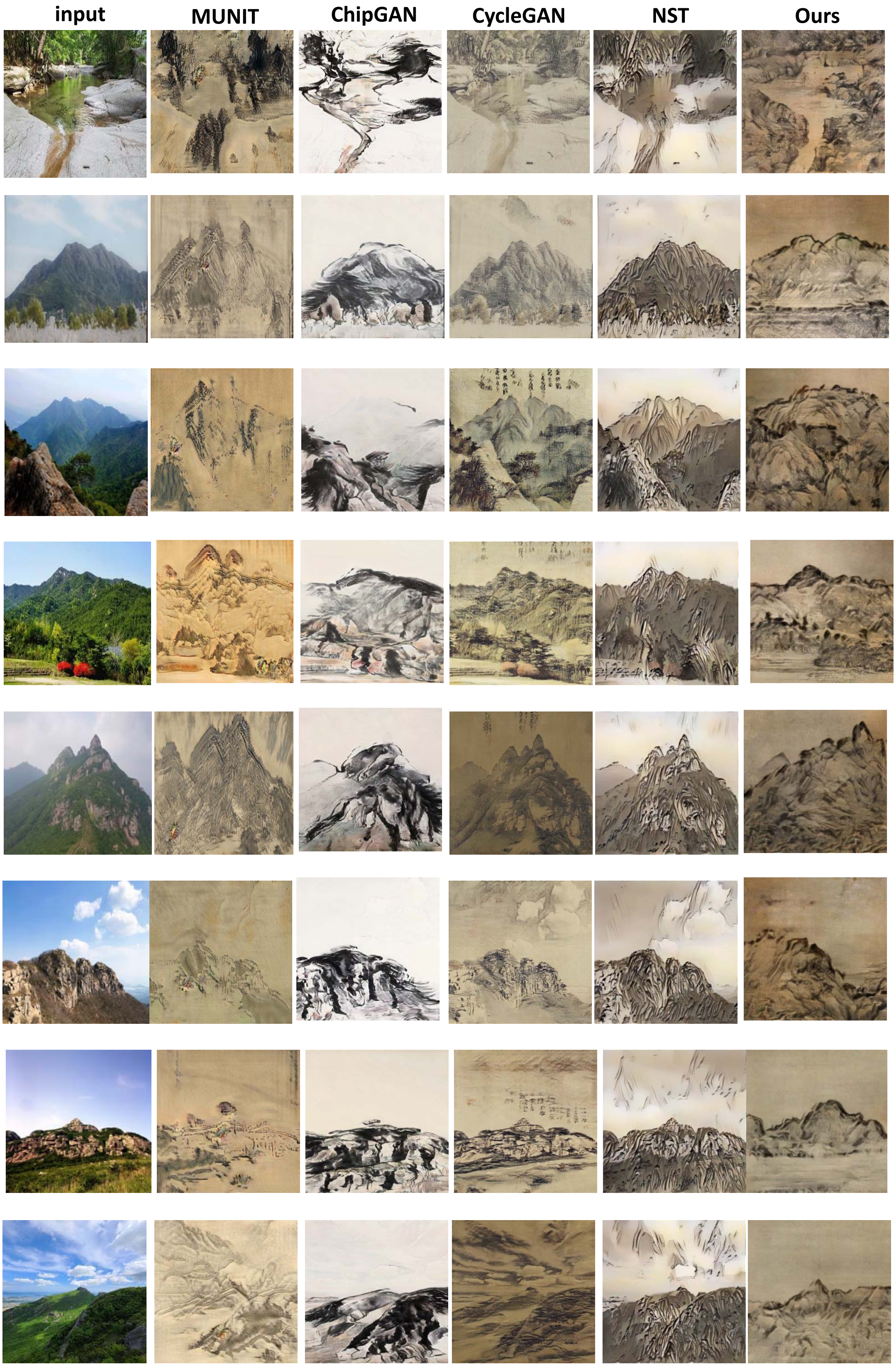


Figure 11. Comparison of our results with those from existing works.

6.3.2. Focus Group Evaluation

We hire ten experts who have an experience of painting TOL artworks. All of them major in oriental paintings. Three of them are senior undergraduate students, four of them are graduate students, and three of them are professional oriental painting artists. Four of them are female and six of them are male.

We ask the participants to evaluate the completeness of the TOL images generated by our model in ten-point metric. The completeness includes both the authenticity and the aesthetics. The authenticity estimates the similarity of our results with real TOL artwork images. They have long been engaged in both the appreciation and creation of TOL artwork, which has established their high standards for TOL artwork images. Their evaluation standards include color fidelity, level of detail, and stroke effect. Based on these standards, they are instructed to evaluate the similarity between the generated TOL images and real TOL artwork images. The aesthetics metric assesses how well the generated TOL images align with their aesthetic criteria. The participants are guided to consider these two metrics in conjunction. The points range in $(1 \sim 10)$. They are instructed to mark the generated TOL images by assuming the real TOL artworks have 10 points. The results are presented in Table 4. Among the eight sample images, our results record highest scores for six images.

6.4. Ablation Study

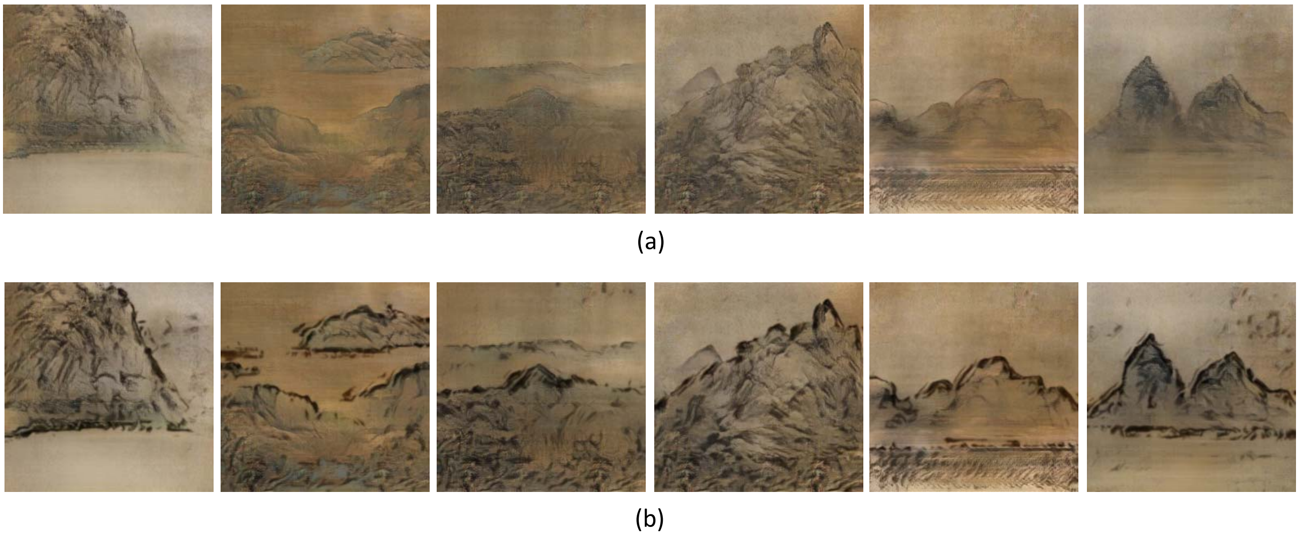
For an ablation study of our model, we compare two sets of generated TOL images in Figure 12. The images in the upper row are generated without AdaIN module that enhances the generated TOL images. The images in the lower row are produced by processing the images in the upper row through the AdaIN module. In comparing the generated TOL images in both upper and lower row, we can conclude that TOL generator using SPADE module has a limitation in producing thick stroke patterns of TOL artwork images. By adding AdaIN module for the enhancement, we can generate TOL images that mimic the thick stroke patterns of TOL artwork. We estimate FID and ArtFID for the generated TOL images in Figure 12 and present the average FID and ArtFID values in Table 5. The average FID and ArtFID values in Table 5 show that the TOL images generated with AdaIN module is superior to those without AdaIN module. From the values in Table 5, we conclude that AdaIN module plays an important role in generating TOL artwork images. Similar to the FID and ArtFID scores in Table 2 that compares ours with those from the existing studies, ours show better FID and ArtFID scores.

Table 5. The FID and ArtFID values for the images in Figure 12.

	FID	**ArtFID**
(a) without AdaIN	194.7	178.6
(b) with AdaIN	157.4	140.8

6.5. Diversity

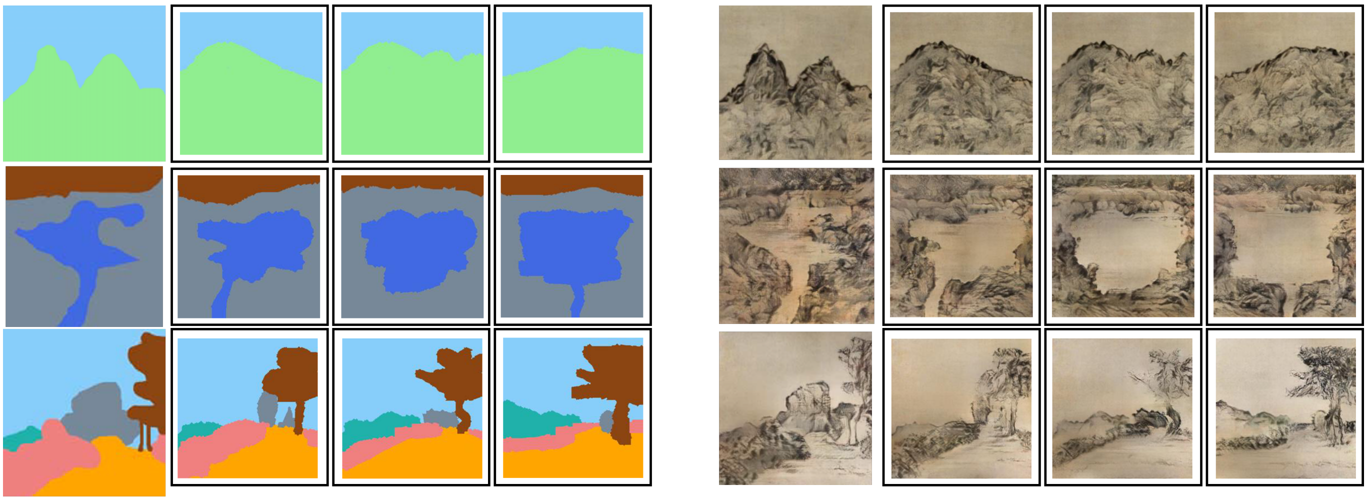
The diversity of TOL images comes from both shape and texture. However, since the shape is determined by the user-initiated TOL map, the diversity controlled by our framework is related to texture. Figure 13 illustrates TOL images generated from similar TOL maps. These images demonstrate that our framework is capable of generating TOL images with diversity in texture.



(a)

(b)

Figure 12. Ablation study of our results: generated TOL images based on six different TOL maps. (**a**) The images in the upper row are generated without the AdaIN module, which enhances the generated TOL images. (**b**) The images in the lower row are produced by processing the images in the upper row through the AdaIN module.



(a) Variation of user-initiated TOL maps

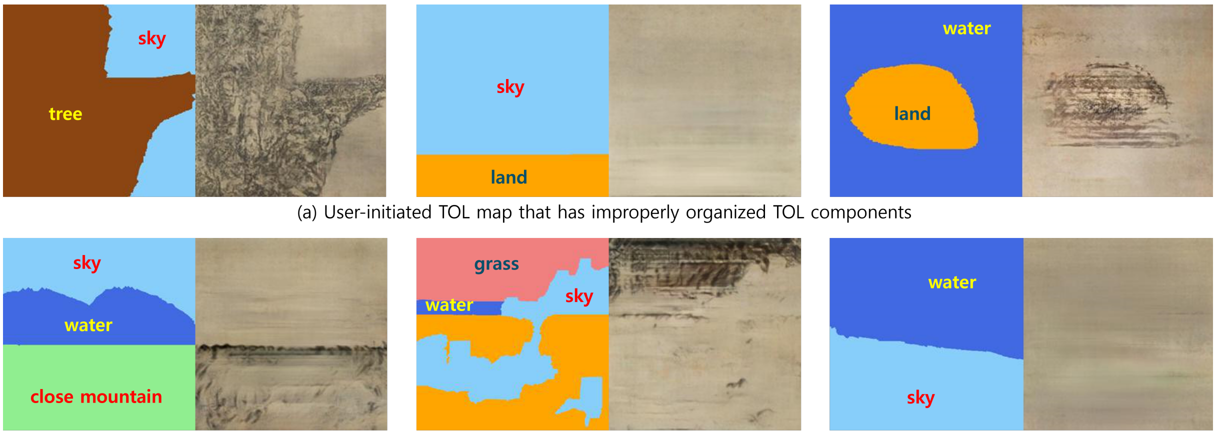
(b) Their corresponding TOL images

Figure 13. Various TOL images generated from similar TOL maps.

6.6. Limitation

We analyze the most serious limitation of our model is that our model depends on the TOL map produced by users. Users are required to initiate a reliable TOL map for a successful TOL image generation. Figure 14 illustrates some of the failed TOL images generated from improper TOL map, that has TOL components either improperly organized or unreal. This limitations comes from the training dataset of TOL artwork images. The TOL artwork images have a lot of common TOL components such as land, sky, water, and grass, etc. and these TOL components have located in a similar style. Therefore, user-initiated TOL map whose components do not follow the components of TOL artwork images may lead to failed TOL images.

Another limitation of our model is that the generated TOL images from our model depends on the components of the TOL map. Since we have selected components of frequently appearing components of real TOL artwork images, some components such as person or animals are neglected.



(a) User-initiated TOL map that has improperly organized TOL components

(b) User-initiated TOL map that has unreal TOL components

Figure 14. Failed TOL images generated: (**a**) shows TOL images generated from improper user-initiated TOL map, (**b**) shows TOL images generated from unreal TOL map.

The other limitation of our model is that the result images show yellow-paper effect, which is frequently observed for papers more than hundred years. We have collected real TOL artwork images from many museums. Therefore, the collected TOL artwork images are painted on yellow papers. Some contemporary TOL artwork images are required to be collected for produce diverse TOL images.

7. Conclusions and Future Work

We have presented TOLGAN that generates TOL images from TOL map that specifies the locations and shapes of the elements composing TOL. TOL map is produced by a user interface or a segmentation scheme from a photograph. The generator of TOLGAN is developed as a decoder structure where SPADE module is inserted between the layers. The SPADE module processes TOL map to control the result of TOL image. The generated TOL image is further enhanced through AdaIN structure to complete a visually-pleasing TOL image.

The results of our approach are evaluated through both quantitative and qualitative evaluations. For the quantitative evaluation, we estimate FID and ArtFID scores for our results and the results from important existing studies and show that ours show smallest FID and ArtFID scores. For the qualitative evaluation, we execute two user studies including volunteers and focus group. These user studies show our framework produces better TOL images than the compared studies.

For a further work, we aim to enhance our model that can produce TOL artwork images without TOL maps. In order to produce scholastic TOL artwork images, TOL map should be produced automatically. We aim to design a model that can produce TOL maps from scratches. Another direction is to extent our TOL generation model for various applications such as TOL animation. TOLGAN can automatically generate background images of TOL animation. TOL style can be extended to different applications such as virtual environment, game and metaverse. Many researchers or artists have rarely tried to apply TOL style to these applications. Our scheme, which effectively generates TOL images from simple user-initiated TOL maps, can present a helpful tool for these applications.

Author Contributions: Methodology, B.K.; Investigation, H.Y.; Data curation, B.K.; Writing—original draft, K.M.; Writing—review & editing, H.Y.; Supervision, H.Y. and K.M.; Project administration, K.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Sangmyung University at 2022.

Data Availability Statement: The datasets presented in this article are not readily available because they are part of an ongoing study. Requests to access the datasets should be directed to yanghk@smu.ac.kr. The copyright of the photographs used in our manuscript falls under "Fair Use".

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Strassmann, S. Hairy Brushes. *ACM Comput. Graph.* **1986**, *20*, 225–232. [CrossRef]
2. Guo, Q.; Kunii, T. Modeling the Diffuse Paintings of 'Sumie'. *Proc. Model. Comput. Graph.* **1991**, *1991*, 329–338.
3. Lee, J. Simulating Oriental Black-ink Painting. *IEEE Comput. Graph. Appl.* **1999**, *19*, 74–81. [CrossRef]
4. Lee, J. Diffusion Rendering of Black Ink Paintings using new Paper and Ink Models. *Comput. Graph.* **2001**, *25*, 295–308. [CrossRef]
5. Way, D.; Lin, Y.; Shin, Z. The Synthesis of Trees in Chinese Landscape Painting using Silhoutte and Texture Strokes. *J. WASC* **2002**, *10*, 499–506.
6. Huang, S.; Way, D.; Shih, Z. Physical-based Model of Ink Diffusion in Chinese Ink Paintings. *Proc. WSCG* **2003**, *2003*, 33–40.
7. Yu, J.; Luo, G.; Peng, Q. Image-based Synthesis of Chinese Landscape Painting. *J. Comput. Sci. Technol.* **2003**, *18*, 22–28. [CrossRef]
8. Xu, S.; Xu, Y.; Kang, S.; Salesin, D.; Pan, Y.; Shum, H. Animating Chinese Paintings through Stroke-based Decomposition. *ACM Trans. Graph.* **2006**, *25*, 239–267. [CrossRef]
9. Zhang, S.; Chen, T.; Zhang, Y.; Hu, S.; Martin, R. Video-based Running Water Animation in Chinese Painting Style. *Sci. China Ser. F Inf. Sci.* **2009**, *52*, 162–171. [CrossRef]
10. Shi, W. Shan Shui in the World: A Generative Approach to Traditional Chinese Landscape Painting. In Proceedings of the IEEE VIS 2016 Arts Program, Baltimore, MD, USA, 23–28 October 2016; pp. 41–47.
11. Gatys, L.; Ecker, A.; Bethge, M. Image Style Transfer Using Convolutional Neural Networks. In Proceedings of the CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 2414–2423.
12. Ulyanov, D.; Lebedev, V.; Vedaldi, A.; Lempitsky, V. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. In Proceedings of the ICML 2016, New York, NY, USA, 19–24 June 2016; pp. 1349–1357.
13. Huang, X.; Belongie, S. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In Proceedings of the ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 1501–1510.
14. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the NIPS 2014, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
15. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784.
16. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A. Image-to-image Translation with Conditional Adversarial Networks. In Proceedings of the CVPR 2017, Honolulu, HI, USA, 17–26 June 2017; pp. 1125–1134.
17. Zhu, J.; Park, T.; Isola, P.; Efros, A. Unpaired Image-to-image Translation using Cycle-Consistent Adversarial Networks. In Proceedings of the ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
18. Park, T.; Liu, M.; Wang, T.; Zhu, J. Semantic Image Synthesis with Spatially-Adaptive Normalization. In Proceedings of the CVPR 2019, Long Beach, CA, USA, 15–20 June 2019; pp. 2337–2346.
19. Huang, X.; Liu, M.-Y.; Belongie, S.; Kautz, J. Multimodal Unsupervised Image-to-Image Translation. In Proceedings of the ECCV 2018, Munich, Germany, 8–14 September 2018; pp. 172–189.
20. Li, B.; Xiong, C.; Wu, T.; Zhou, Y.; Zhang, L.; Chu, R. Neural Abstract Style Transfer for Chinese Traditional Painting. In Proceedings of the ACCV 2018, Perth, Australia, 2–6 December 2018; pp. 212–227.
21. Lin, D.; Wang, Y.; Xu, G.; Li, J.; Fu, K. Transform a Simple Sketch to a Chinese Painting by a Multiscale Deep Neural Network. *Algorithms* **2018**, *11*, 4. [CrossRef]
22. He, B.; Gao, F.; Ma, D.; Shi, B.; Duan, L. Chipgan: A Generative Adversarial Network for Chinese Ink Wash Painting Style Transfer. In Proceedings of the ACM Multimedia 2018, Seoul, Republic of Korea, 22–26 October 2018; pp. 1172–1180.
23. Zhou, L.; Wang, Q.-F.; Huang, K.; Lo, C.-H.Q. ShanshuiDaDA: An Interactive and Generative Approach to Chinese Shanshui Painting Document. In Proceedings of the International Conference on Document Analysis and Recognition 2019, Sydney, Australia, 20–25 September 2019; pp. 819–824.
24. Xue, A. End-to-End Chinese Landscape Painting Creation Using Generative Adversarial Networks. In Proceedings of the WACV 2021, Online, 5–9 January 2021; pp. 3863–3871.
25. Hung, M.; Trang, M.; Nakatsu, R.; Tosa, N. Unusual Transformation: A Deep Learning Approach to Create Art. In *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*; Springer: Cham, Switzerland, 2022; Volume 422, pp. 309–320.
26. Chung, C.; Huang, H. Interactively transforming chinese ink paintings into realistic images using a border enhance generative adversarial network. *Multimed. Tools Appl.* **2023**, *82*, 11663–11696.
27. Wright, M.; Ommer, B. ArtFID: Quantitative Evaluation of Neural Style Transfer. In Proceedings of the German Conference on Pattern Recognition 2022, Konstanz, Germnay, 27–30 September 2022; pp. 560–576.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.