Design and Development of an Intelligent Rendering System for New Year's Paintings Color Based on B/S Architecture

Zaozao Guo
Faculty of Art, Sustainable and Creative Industry, Sultan Idris Education University, 35900, Perak Darul Ridzuan, Malaysia
Department of Computer Science, Nankai University Binhai College, Tianjin 300270, China

***Abstract*—With the arrival of the synthetic talent era, laptop technological know-how for the safety and inheritance of intangible cultural heritage has added a new way of thinking, and the range of intangible cultural heritage additionally offers greater chances for laptop technology, the utility of laptop talent science to New Year's Eve artwork of the applicable lookup there are many gaps. Training of Cyclic Generative Adversarial Network (CycleGAN) realize the task of extracting plots of different site types from planar maps and the Rendering generation from planar color block map to color texture map. This paper first introduces the B/S community architecture, Python programming technological know-how and Django framework. Then the unique approach of using pc Genius to the project of rendering Chinese New Year artwork is clarified via modeling, studying algorithms, and community architecture. Finally, a hierarchical fusion generative adversarial neural community structure is designed primarily based on generative adversarial neural networks. The structural and textural features of the image are fused by texture GAN and then rendered to generate the New Year paintings. The test results show that this kind of algorithm draws clear texture, realistic images and full color of the New Year's pictures, and the IS index reaches 3.16 in the quantitative analysis, which is higher than other comparison algorithms.**

Keywords*—*B/S architecture; intelligent rendering; adversarial neural network; Chinese New Year painting

I. Introduction

There are two essential standard approaches to instructing painting: one is the way of offline teaching, and the different is the way of online video teaching. Offline introduction is an expert instructor and college students face-to-face teaching; this offline coaching is the most common way of educating artwork education [1-3]. The online video instructing technique is that the instructor documents the video in advance, and then the college students pay to get the instructing video. The two present regular portray teaching strategies have the following disadvantages.

1) Offline teaching methods of one-on-one teaching is the teacher to meet the requirements of the students at the designated location of the class; this class is expensive; if it is a one-to-many teaching mode, although the cost can be relatively reduced, the teaching effect cannot be guaranteed;

2) The relevant video teaching method is the teacher to meet the requirements of the students to the designated location of the class.

3) Fewer relevant professional teachers, but the market demand, resulting in several students still in the school stage for unregulated training courses, resulting in uneven teaching levels.

4) Teachers and students offline teaching methods by space, time limitations, teachers and students or student's parents must be based on the designated time to the designated place of class teaching.

5) The traditional video teaching method is usually that the teacher records the video in advance, and then the students watch the video to learn. This teaching method lacks interaction between the teacher and the students and is passive.

Painting is one of the oldest types of art, and its content material has advanced all through human records and exclusive cultures. In modern times, with the popularization of electronic information technology, all kinds of electronic drawing board software and hardware have gradually become the main tools for people to create paintings, such as Adobe's Photoshop, Apple's iPad, Apple Pencil, etc., and the digital form of photographs produced by them have been widely used in various aspects of production and life, such as news dissemination, prototyping, film and television creation, etc. In general, portraying wholly demonstrates human nature, and it is the essential painting shape, focusing on realism. In general, representation wholly shows human knowledge and creativity and is a critical potential in visible conversation for humans regardless of its shape and tools. In the modern-day community era, the digital shape of portray can be very handy to disseminate and use, is one of the major types of contemporary media, and consequently, has a very realistic fee and realistic significance [4, 5].

The main manifestation of program communication, the canonical representation of the floor plan with the advancement of AI technology, the design drawing and high-quality rendering based on the design is crucial for the design presentation [4]. Therefore, machine learning (Machine Learning) of the setup data makes it possible for Artificial Intelligence Aided Design (Artificial Intelligence Aided planner often spends a lot of time to collect plan cases that can be drawn from, Design, AIAD), which, Deep Learning and rendering of the floor plan using a variety of software. At present, computer vision (Deep Learning) has been in the design of image analysis and generation of consciousness (Computer Vision) in the field of image recognition; generation technology shows great potential for application. In recent years, in architecture and graphic arts has been more mature, so that AI recognition case plane and rendering cartography design field, professional image data sets and deep learning applications do not become possible. And the realization of this automated analysis and mapping of the front break emerges, designers rely on interdisciplinary cooperation to develop AIAD field mentioning is a high-quality landscape garden floor plan training set. Unlike common face or object image databases, Landscape Architecture Flat View to improve the efficiency of design analysis and mapping. And landscape architecture discipline to carry out this type of research limitations mainly from the interdisciplinary cooperation and the number of face database has outstanding professionalism: 1) the variety of land types: the lack of data sets. 2) drawing expression both normative and artistic; 3) access to the threshold of the image is to present the design scheme is an important carrier of the Landscape Garden high, the need for professionals to carry out careful screening and data annotation Lin design floor plan is the designer case to draw upon, Concept presentation and Data Labeling.

Artificial intelligence is an important engine for future economic development, and machine learning technology is a major research field of artificial intelligence; the current machine learning technology, according to the different learning modes, is roughly divided into three categories: supervised learning, unsupervised learning and reinforcement learning. Supervised gaining knowledge is a studying technique that depends on statistics labeling, primarily the use of statistics labeling statistics to supervise the coaching model and then using the dummy to predict the new data, in general, used in classification, regression and different tasks [6]. In general, supervised and unsupervised learning focus on the perception and understanding of data; in contrast, reinforcement learning methods not only realize the perception and understanding of data but also focus on generating intelligent decisions and actions according to the task goals, which can realize more complex functions and become a hot research topic in recent years.

II. Relevant Theories and Technologies

Technology is the basis for realizing various functions. In building a platform for processing and analyzing painting images, some important theoretical techniques provide important support for the platform's design. This paper adopts the Python+Django technology framework for design and uses PyCharm for design and debugging. The B/S architecture is used for system design to facilitate subsequent upgrading and maintenance [7, 8]. Before introducing the platform, the theoretical technologies involved are first introduced and summarized.

A. Python Technology

Python is a high-level and broadly used programming language. It has the benefits of excessive efficiency, low overhead, open source, portability, sturdy interpretability, etc., and has been preferred to utilize builders quickly. Python's fusion of several algorithmic programs determines that it can play a necessary position in unique fields. The general architecture of Python's distribution of functions is shown in Fig. 1 as follows:

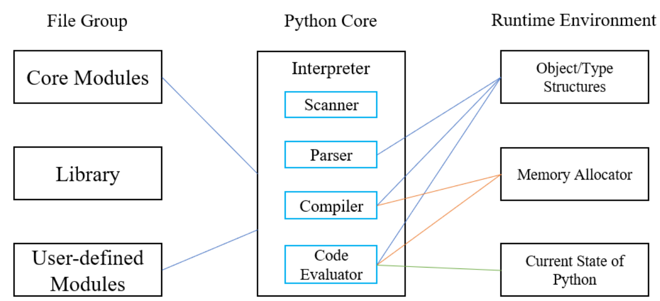

Fig. 1. Python overall architecture diagram.

As shown in Fig. 1, the overall architecture of Python can be divided into three parts: Python module libraries, commonly used standard libraries and runtime environments. For the commonly used syntax standard libraries of Python, (see Table I).

TABLE I. Common Syntax Standard Library of Python

Grammar Library	Related Explanations
atexit	Functions that are allowed to be called when the program exits
argparse	A function that parses a command run
bisect	Providing a binary lookup algorithm for sorted lists
calendar	Date-related functions
codecs	Functions related to coding and decoding data
collections	Useful data structures
copy	Functions for copying data

B. Cornerstone Image Support Technology

Cornerstone.js is a medical image viewing tool written in Js script to support the display and interaction of medical images. As its name suggests, it has a “cornerstone” role in the image display field, and many image reading systems for artistic paintings are based on Cornerstone.js. Rich scripts provide powerful technical support for browsing image data. Basic operations such as zooming and panning of image data can be realized.

Currently, most home portray exhibition structures mixed with AI are also developed primarily based on Cornerstone.js. The Internet web page multi-threaded decoding used in this script quickens the show of snapshots and helps the Internet to practice compression strategies such as JPEG to transmit images. Modularity (component design) allows it to be embedded in extraordinary front-end frameworks for convenient invocation through developers. As proven in the following code, it can recognize the show of DICOM portray images [9, 10].

```
const element=document.getElementById('demo-element');
const imageId='http://example.url.com/image.dcm';
cornerstone.enable(element);
cornerstone.loadAndCacheImage(imageId).then(function(image){
cornerstone.displayImage(element,image);
cornerstoneTools.mouseInput.enable(element);
```

The flow when Cornerstone performs image display is shown in Fig. 2.

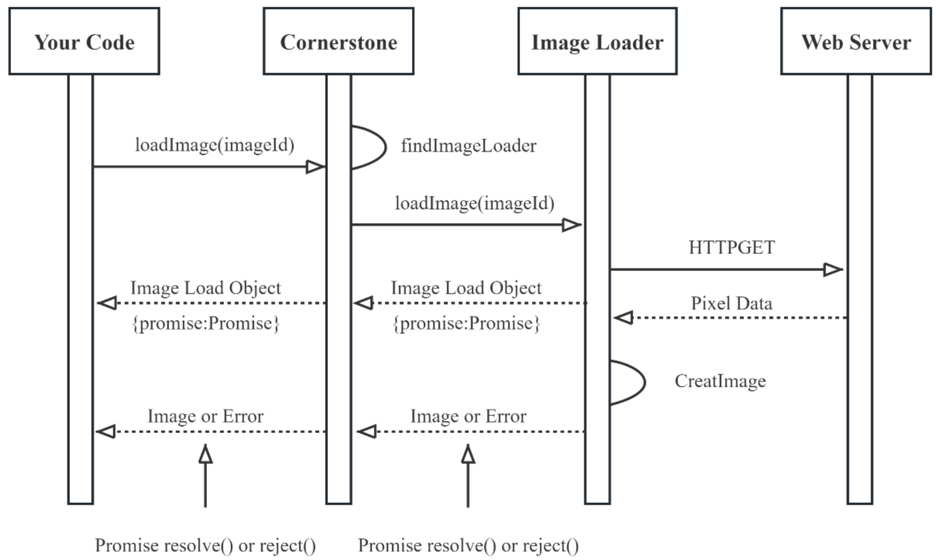

Fig. 2. Cornerstone flowchart.

C. Web Framework Technology

Web framework technological know-how is a disbursed utility software architecture, which is ordinarily divided into two foremost parts: client-side and server-side. The customer aspect frequently includes Html language, script program, CSS style, plug-in technological know-how and so on [11]. Each element on the "active" page has a variety of labels. web server development technology is also from static to dynamic gradual development, gradually being perfected. Server-side technologies include servers, CGI, ASP, Servlet and JSP technologies [12, 13]. Fig. 3 shows the Web application processing flowchart.

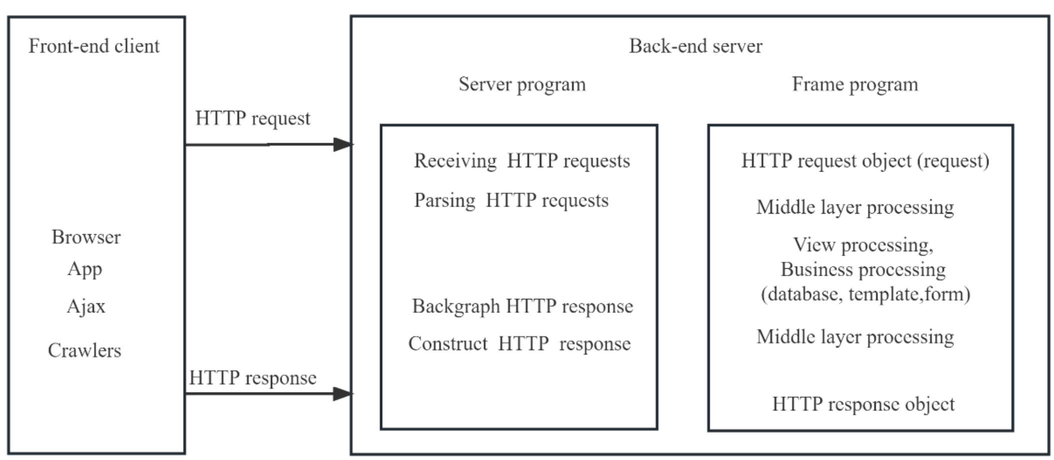

Fig. 3. Web application processing flow chart.

Web technology is more in line with the current technical requirements and is an important direction for the future development of network architecture. Therefore, this paper adopts the B/S architecture model that satisfies Web technology for the research and development of image processing systems [13].

D. MVT Framework

Generally, in improving the project, it is fundamental to first classify the functions, a massive piece of the task damaged down into many small tasks, and as a way as viable to comprehend the low coupling of the range of modules to beautify the scalability and portability.

MVC's full spelling for Model-View-Controller, after continuous development and integration, the idea of MVC has been applied to Web development. It plays an important role, known as Web MVC framework. In MVC, M is model (business logic layer), V is view (interface layer), C is controller (controller), used to schedule View layer and Model layer.

Python language Django framework uses the MVT architecture pattern. In Django for web development, this framework also follows the MVC idea. However, in Django, this architecture is called MVT, but the essence of the MVC pattern is the same.

The operational logic between the three MVTs is shown in the following Fig. 4.

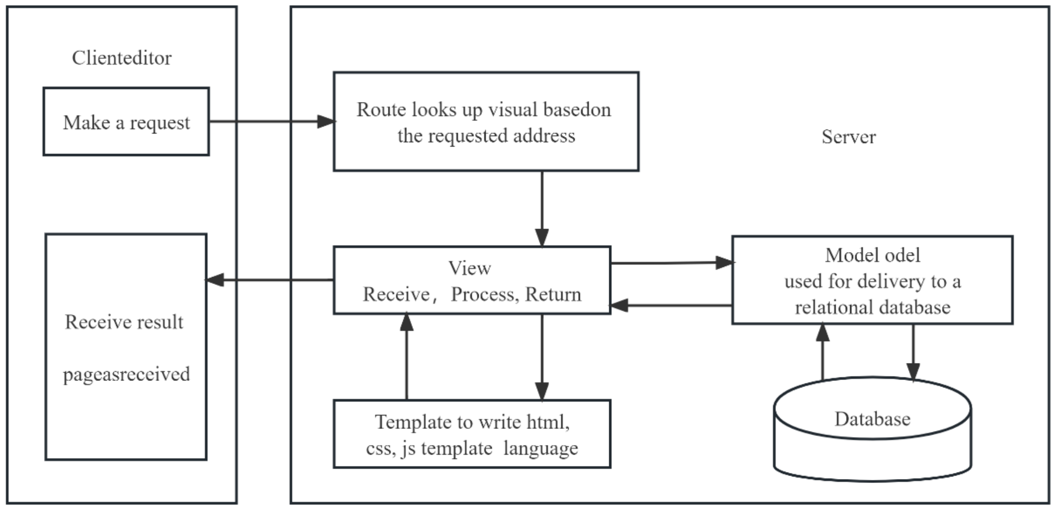

Fig. 4. MVT operation logic diagram.

Various supporting technologies are indispensable in the design and development, and the intermingling of different technologies lays a cornerstone foundation for the design and development of the platform [14]. Coordinating the different technologies to achieve the desired effect is an important and complex process. The role of different technologies are skillfully combined, in line with the needs of the current form of social development, to provide the necessary technical support for the exchange and integration of different fields.

E. Architecture Mode

The b/s structure is a famous community shape mode after the upward jab of the Web. b/s shape comprises the show, characteristic, and statistics layers. This mannequin unifies the patron and centralizes the core implementation of the device on the server side, simplifying improvement and maintenance [15,16]. The statistics layer strategies and calculates the request based on the user's conduct.

The network structure of the b/s model is shown in the following Fig. 5.

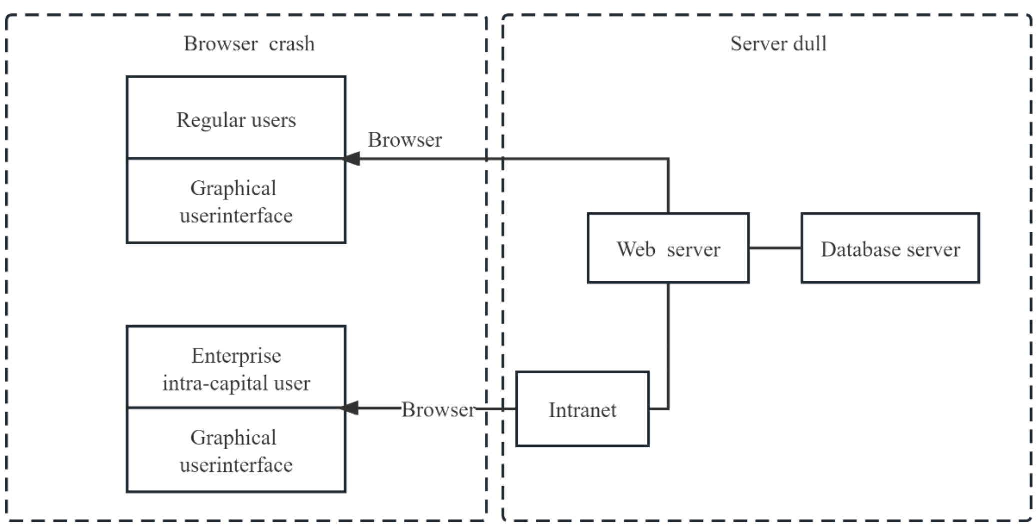

Fig. 5. Schematic diagram of the b/s architecture network structure.

The final processed page information or data is returned to the browser, and the browser displays the results to the browser by rendering the information returned by the server. The page information or data is returned to the browser, and the browser, by rendering the information returned by the server to display the results on the page [17].

Compared with the c/s model, the b/s model has more prominent advantages and is, therefore, more popular.

1) Easy maintenance and upgrading: In the common c/s model, upgrading the gadget requires upgrading each the customer and the server. After upgrading the software, every consumer desires to improve his/her purchaser to use it. If widely widespread improvements are required, the work of gadget directors in retaining the gadget will be very time-consuming and poorly maintained. But b/s structure solely wants to focus on upgrading the machine saved on the server can be, all the purchasers, that is, the browser, do no longer want to do any maintenance.

2) Superior performance: With the development of the Web, b/s architecture is widely used, but also promotes the development of Ajax technology, which makes part of the data processing in the client can be carried out, greatly reducing the burden on the Web server, and can realize the page local real-time refreshing, improve the interactive performance of the page.

3) Low development costs, more options: Software development based on b/s architecture generally only needs to be installed on a Linux server, so there are many choices of servers and high security. The Linux operating system and the database's connection are free, so this not only greatly reduce the cost of software development but also many choices.

4) High reusability: The c/s architecture of the program needs to consider the whole software design from the whole; when there is a problem, or the system needs to be upgraded and changed, it may need to be redesigned to make a completely new system. Software with b/s architecture can be largely divided into different components by functional modules to achieve high reusability.

When the software is changed from c/s architecture to b/s architecture, the system software no longer needs the developers to specialize in the development of the client software. Still, it only needs to focus on the update of the program to free up the labor force [18]. At the same time, the use of the browser as a client, the development of a more friendly interface, while the newly developed system does not require users to learn from scratch, greatly reduces the user's learning costs.

III. Reinforcement Learning based Stroke Painting Simulation

A. Model-based DDPG Algorithm

Fig. 6 shows the general framework of the algorithm, which is generally a Model-based Deep Deterministic Policy Gradient (Model-based DDPG). At the core of the framework is a drawing intelligence, whose goal is to decompose a given target image into a number of strokes, and let these strokes reconstruct the target image on the drawing board by means of a renderer, which is "model-based" in the sense that it utilizes the explicit model of a discriminator and a neural renderer [19, 20]. In order to simulate the human drawing process, this paper will use a sequential Markov decision process to model it: in the inference phase, the intelligent body decides the control parameters (i.e., actions) for the next stroke in each step based on the observed target image and the current state of the drawing board, and the renderer receives the control parameters and draws the strokes on the drawing board; in the training phase, the training samples are randomly sampled from the empirical recall cache, and then the discriminators and neural renderers are used to reconstruct the target image with the strokes [21].

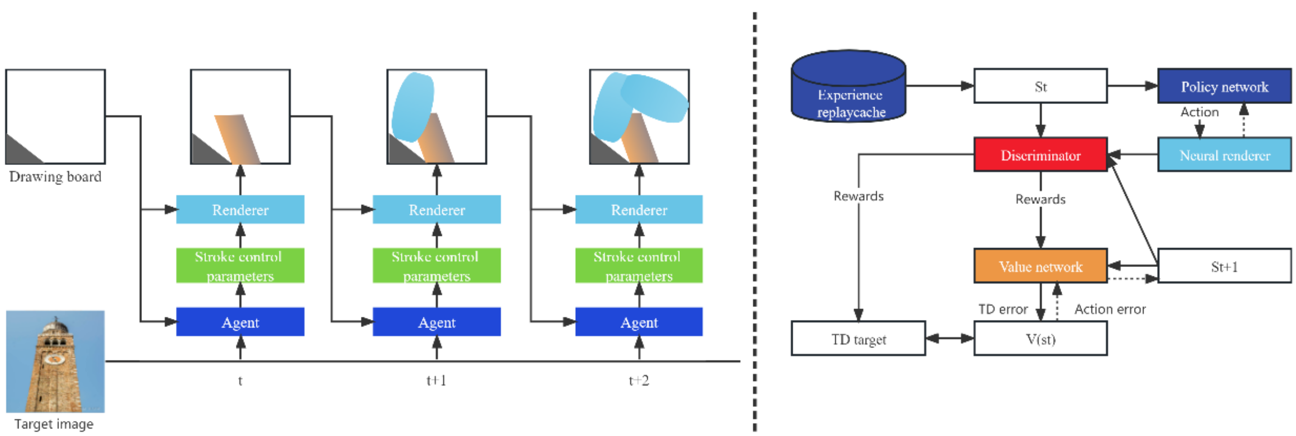

Fig. 6. The general framework of the algorithm (left, inference phase; right, training phase).

B. Neural Renderer

1) Task definition: In the reinforcement learning drawing simulation framework, the task of the neural renderer is to render the input stroke control parameters (i.e., the action produced by the drawing intelligences) into a rasterized image of the stroke, which is then added to the current drawing board. Specifically, neural renderers are used for three reasons.

a) Due to the differentiable nature of the neural renderer, the errors of the strokes can be back-propagated through the renderer, which is crucial for the model-based DDPG algorithm used in this paper;

b) The neural renderer can be trained by a supervised learning algorithm to mimic a real renderer rendering strokes. In this way, existing real renderers can be used to train the neural renderer, avoiding repetitive manual design [22, 23].

2) Learning algorithm: In this paper, we use the supervised learning method to train the neural renderer. For each stroke, suppose the corresponding stroke renderer is R_{GT}, the learning goal of the neural renderer R is to hope that for any drawing board state C and stroke control parameter a, its rendering result is as similar as possible to R_{GT}, and ideally, the two are equal, i.e. (as shown in Fig. 7).

$$R(C,a)=R_{GT}(C,a) \tag{1}$$

For the rendering process designed in this paper, the output S_t a_t of the neural renderer subject network is independent of the drawing board state, taking into account that different strokes have different degrees of prominence in different board states (e.g., black strokes are almost invisible in the black drawing board, while very obvious in the white drawing board). Therefore, during the training process, this paper defines two standard states for the board, C_B, C_W, which denote the empty boards of pure black and pure white, respectively, and then measures the gap in rendering results using the L_2 distance, and ultimately constructs the objective function of the learning as,

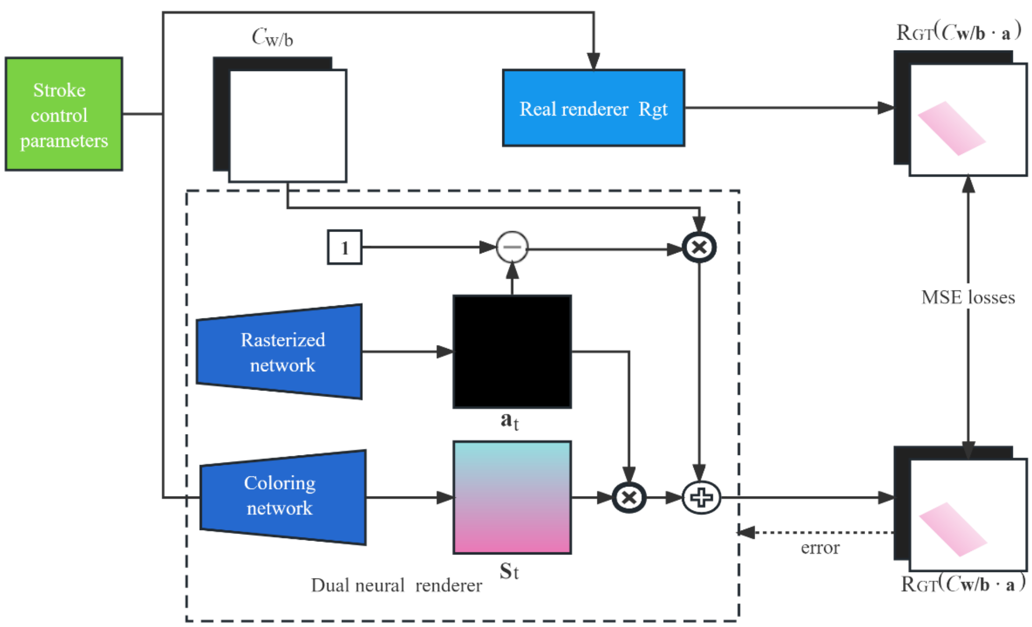

Fig. 7. Neural renderer training approach.

$$J\left(W^R\right)=E_{a\sim U(A)}\left[\begin{array}{l}\left|R\left(C_B,a\right)-R_{GT}\left(C_B,a\right)\right|^2+\\ \left|R\left(C_W,a\right)-R_{GT}\left(C_W,a\right)\right|^2\end{array}\right] \tag{2}$$

Where $\boldsymbol{W}^R$ denotes the parameters of the neural renderer, $U(A)$ denotes the uniform distribution defined on the stroke control parameter space A. The learning process of the neural renderer is to minimize the objective function.

$$\min_{W} J\left(\boldsymbol{W}^R\right) \tag{3}$$

In practice, this paper uses a stochastic gradient descent algorithm to solve this optimization problem and complete the training of the neural renderer. As in Fig. 7, the sampled stroke control parameters will be sent to the real renderer R_{GT}, which renders the stroke on the blank drawing board $C_{W/B}$ with the result $R_{GT}(C_W,a)$, and the neural renderer synthesizes the $\boldsymbol{S}_t$, $\boldsymbol{\alpha}_t$ internally according to the same control parameters, and then obtains the rendering result $R_{GT}(C_W,a)$ according to Eq. 3, and finally calculates the L_2 of the two results distance (i.e., MSE loss) to update the network parameters of the neural renderer.

3) Network structure: The neural renderer in this paper uses a two-way architecture divided into a shading network and a rasterization network. The shading network is used to output the stroke image $\boldsymbol{S}_t$, whose input is the complete stroke control parameters, and the network consists of six layers of inverse convolutional layers. The rasterization network is used to output the transparency image $\boldsymbol{\alpha}_t$, whose input is the part of the stroke control parameter that does not contain color, and the network consists of four fully-connected layers and three Shuffle convolutional layers.

C. Stroke Translator

1) *Task definition:* The undertaking of the stroke translator is to translate the managed parameters of one variety of strokes into these of any other by means of the capacity of a neural network so that the rendering consequences of the two sorts of strokes are as visually comparable as possible [24]. After the intelligent body has been trained to master one type of stroke, the control parameters of the stroke generated by the intelligent body during the painting process can be collected and then converted into the control parameters of another type of stroke using a stroke translator, and finally rendered by a renderer to obtain the painting results of another type of stroke. In this way, the painting results of multiple styles of strokes can be obtained with less time and computational resources [25, 26]. Let the control parameter of the source stroke A be $a_A \in \mathrm{A_A}$ and the control parameter of the target stroke B be $a_B \in \mathrm{A_B}$; the translator T_{AB} translates the source stroke into the target stroke, i.e.

$$a_B = \mathrm{T_{AB}}\left(a_A; \boldsymbol{W}^T\right) \tag{4}$$

Where $\boldsymbol{W}^T$ denotes the parameters of the translator.

2) *Learning algorithm:* In this paper, we use a supervised learning method to train a stroke translator. Assuming that the renderer of the source stroke A is R_{GT}^A and the renderer of the target stroke B is R^B, where R^B is a neural renderer, the learning goal of the translator is to hope that for any source stroke $a_A \in \mathrm{A_A}$ and any drawing-board state C, the renderer is able to find a corresponding target stroke $\mathrm{T_{AB}}\left(a_A; \boldsymbol{W}^T\right) \in \mathrm{A_B}$ so that the rendering results of the two strokes $R_{GT}^A\left(C, a_A\right)$ and $R^B\left(C, T_{AB}\left(a_A\right)\right)$ are as similar as possible visually. For this, this paper defines two standard drawing board states, C_B, *and* C_W, which denote pure black and pure white empty drawing boards, respectively, and then measures the gap between the rendering results using the L_2 distance and finally constructs the learned objective function as,

$$J\left(\boldsymbol{W}^T\right) = \mathrm{E}_{a_A \sim \mathrm{U}(\mathrm{A}_A)} \left[\begin{array}{l} \left| R^B\left(C_B, T_{AB}\left(a_A; \boldsymbol{W}^T\right)\right) - R_{GT}^A\left(C_B, a_A\right) \right|^2 \\ + \left| R^B\left(C_W, T_{AB}\left(a_A; \boldsymbol{W}^T\right)\right) - R_{GT}^A\left(C_W, a_A\right) \right|^2 \end{array} \right] \tag{5}$$

Where $\mathrm{U}\left(\mathrm{A}_A\right)$ denotes the uniform distribution defined on the stroke control parameter space A_A, and the learning process of the translator is to minimize the objective function, i.e.

$$\min_{\boldsymbol{W}} J\left(\boldsymbol{W}^T\right) \tag{6}$$

Due to the differentiable property of the neural renderer R^B, this optimization problem is solved by stochastic gradient descent algorithm in this paper to complete the training of the stroke translator. For the sampled control parameter a_A, the corresponding neural renderer R_{GT}^A is first used to render the stroke $R_{GT}^A\left(C_{W/B}, a_A\right)$ on a blank drawing board $C_{B/W}$, then the control parameter a_A is converted to a_B by the translator T_{AB}, and then the renderer R^B is used to obtain the rendering result $R^B\left(C_{W/B}, T_{AB}\left(a_A; \boldsymbol{W}^T\right)\right)$, and finally calculate the L_2 distance (i.e., MSE loss) of the two results to update the network parameters of the translator [27].

3) *Network structure:* Since the translation process of the two-stroke control parameters is equivalent to completing a kind of mapping, this paper adopts a four-layer fully-connected network to construct the stroke translator, the size of the input layer is the number of control parameters of the source stroke, and the size of the output layer is the number of control parameters of the target stroke.

IV. Design of Intelligent Rendering Generation Algorithm based on Image Texture

At present, although it is possible to render images, the processing of image details and textures is not satisfactory. In order to solve this problem, an intelligent painting generation algorithm based on image texture drawing technique is proposed and a hierarchical fusion generation countermeasure neural network is constructed. It renders the image using structural GAN and texture GAN, which further generates the texture details of the image and makes the image clearer and more realistic.

A. Generative Adversarial Neural Network

Generative Adversarial Networks (GAN) is a new kind of deep generative mannequin proposed by using Goodfellow et al. It has been effectively utilized to picture generation, video generation, picture fashion migration and picture complementation and different scenarios.

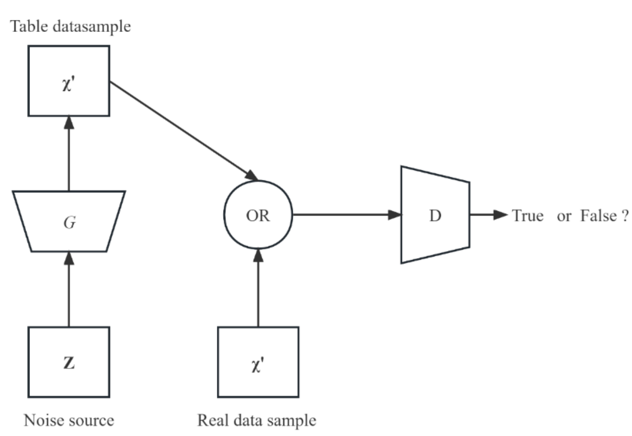

Fig. 8. Schematic diagram of GAN structure.

The generative adversarial community is stimulated via the zero-sum sport in recreation theory, which regards the trouble of producing statistics as a disagreement and sport between two networks, the discriminator and the generator. The position of the generator is to synthesize statistics in a given uniformly dispensed noise or generally dispensed noise [28, 29]. The two networks are trained, improved in the confrontation, and then continue the confrontation after repeated confrontations to obtain progress so that the generated data is constantly close to the real data until it cannot be distinguished from the real data so that the desired data content can be obtained. The principle of the GAN structure is shown in Fig. 8.

Definition G is a network to generate images, which can be called a generator, and z represents a random noise through the noise generated by the image can be recorded as G(z). D is a discriminative network, also known as a discriminator, whose role is to distinguish whether the noise generated by the image is real. If x represents an image, when the input parameter value x, the output D(x) represents the probability value that this image is a real image. When the value is equal to 1, it means that x must be a real image; and when the value is 0, it means that the image x is not a real image. During the training process, the generator and the discriminator play with each other; the mathematical relationship between the two can be expressed as follows:

$$\min_G \max_D V(D,G) = E_{x\sim p_{\text{data}}(x)}\left[\ln D(x)\right] + E_{z\sim p_z(z)}\left[\ln\left(1-D\left(G(z)\right)\right)\right] \tag{7}$$

Where E is the probability function, x is the real data, z is the noise, $_{\text{data}}(x)$ denotes the distribution of the real data set, $D()$ denotes the discriminator, $p_z(z)$ denotes the defined a priori noise, and $G()$ denotes the generator. $D\left(G(z)\right)$ characterizes the probability of the D network to judge whether the image generated by G is real or not, and G, in order to make the image it generates converge as much as possible to the real image, needs to deceive the D network. G, in order to make the image generated by itself as close as possible to the real image, it needs to deceive the D network. Given any function D and G, there is a unique solution, and when G can generate a false image $G(z)$, the value of $D\left(G(z)\right)$ is always 0.5. As a result, the overall model can reach the global optimal state. In practice, it is found that maximizing $\ln D\left(G(z)\right)$ is better than minimizing $\ln\left(1-D\left(G(z)\right)\right)$.

Unlike typical coaching models, GAN implements two exclusive networks and an adversarial education method. It makes use of the returned propagation mechanism, in which a clearer and greater sensible pattern can be synthesized besides the complicated Markov Chain, and the computation is surprisingly simple Model Design

1) Model architecture: The paper proposes to generate an adversarial neural network based on the hierarchical fusion of image structure and texture to realize the intelligent generation of painting images so as to assist the creators to create better works. Its main model architecture is shown in Fig. 9

The sketches drawn are first fed into the system model for normalization preprocessing to ensure that the image conforms to the model.

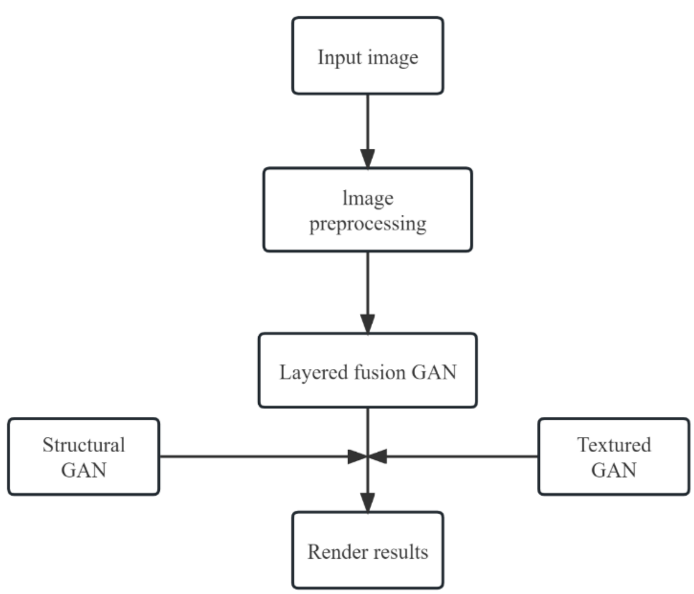

Fig. 9. The model architecture of the paper.

2) Structural GAN: The input of the generative network G is collected from uniformly distributed noise, and the output is a structural line drawing. A 100-dimensional vector is used to represent the input noise z, and the size of the output image is

72×72×3. The discriminative network D will classify the generated structural line drawings based on the real images obtained from deep learning. The generative network adopts a 10-layer model and performs convolutional operations after passing through the fully connected layers to finally generate the structural line drawings. In the setup, batch normalization is used in each layer, and the ReLU activation function is used except for the last layer, which uses the Tanh activation function [30].

3) Texture GAN: The generated network is modified as a conditional GAN, that is, the conditional information is used as additional input for generator G and discriminator D. As an addition to discriminator D, the structure line map not only makes the generated image more real after input, but also requires the generated image to match the structure line map to ensure its controllability. When training the discriminator, only the real image and its corresponding structural line drawings are considered as positive examples, so that a higher resolution 128×128×3 image can be generated by texture GAN.

For example, if the image $x=\left(x_1,\cdots,x_M\right)$, its corresponding structural line drawing is $C=\left(C_1,\cdots,C_M\right)$, and the uniformly distributed noise is $\hat{z}=\left(\hat{z}_1,\cdots,\hat{z}_M\right)$, the generating function can be changed from $G\left(\tilde{z}_i\right)$ to $G\left(C_i,\tilde{z}_i\right)$, and the discriminative function can be changed from $D\left(x_i\right)$ to $D\left(C_i,x_i\right)$ losses of the discriminative network can be rewritten as follows, respectively.

$$L_{\text{cond}}^{G}\left(C,\tilde{z}\right)=\sum_{i=M/2+1}^{M}L\left(D\left(C_i,G\left(C_i,\tilde{z}_i\right)\right),1\right) \tag{8}$$

$$L_{\text{cond}}^{D}\left(X,C,\tilde{z}\right)=\sum_{i=1}^{M/2}L\left(D\left(C_i,x_i\right),1\right)+\sum_{i=M/2+1}^{M}L\left(D\left(C_i,G\left(C_i,\tilde{z}_i\right)\right),0\right) \tag{9}$$

In the generator network architecture, a 128×128×3 textured line graph and 100-dimensional vector noise $\tilde{z}$ are taken as inputs to the network. The inputs to the network are the structured lines and the concatenation of images. The two will first go into the convolutional layer and the inverse convolutional layer, respectively, and then merge to form the 32×32×192 feature maps and further perform seven layers of convolution and inverse convolution on the top of these feature maps. The final output is a 128×128×3 image.

4) Joint model: The discriminator in the texture GAN treats the generated pattern line drawings and images as negative samples and the real structure line drawings and real images as positive samples [31]. As a result, the generator network loss function of the structural GAN can be expressed as follows:

$$L_{\text{joint}}^{G}\left(\hat{z},\tilde{z}\right)=L^{G}\left(\hat{z}\right)+\lambda\cdot L_{\text{cond}}^{G}\left(G\left(\hat{z}\right),\tilde{z}\right) \tag{10}$$

Where $\hat{z}$ and $\tilde{z}$ represent two sets of uniformly distributed samples in structural GAN and texture GAN, respectively. The first term on the right side of the equation represents the adversarial loss of the structural GAN discriminator, and the second term on the right side of the equation represents the loss of the texture GAN. λ is a hyperparameter, and its value is set to 0.1 in this experiment, which is smaller than the learning rate of the texture GAN, so as to avoid the occurrence of the overfitting situation.

B. Test Results and Analysis

1) Experimental environment: The experimental environment is Python 3.6, the processor is i7-6800k, the memory is 32 GB, the graphics card is GTX2080Ti, the operating system is Linux, and the experiment is based on the open-source deep learning framework PyTorch for simulation [32].

The experiments were optimized using Adam optimizer, where the momentum term β=0.5, β_2=0.999, and the batch size M is 128, and the inputs and outputs of all the networks are scaled to [-1, 1]. The learning rate of both structural GAN and texture GAN is set to 0.0002. In joint learning, the learning rate of texture GAN is set to 10^{-6}, and the learning rate of structural GAN is set to 10-7. In addition, the rest of the parameter settings are designed according to the parameters of the experiments in DCGANI.

2) Experimental parameter settings: All algorithms involving stochastic gradient descent in this paper were trained using the Adam optimizer. The neural renderer and translator learning rates are updated using an exponential update strategy every 100 Epochs, and the reinforcement learning uses a manual method to update the learning rates. The dimensions of both the drawing board and the target image are set to H×W×3= 128×128×3. Noting that the limited number of steps for the drawing intelligences is 40 and the size of the action combination is 5, which corresponds to a total number of strokes of 40×5=200.

3) Experimental results: It can be found that after the fusion of the generative adversarial neural network, the final image is more realistic in texture and does not change the original object that the painter is trying to depict, so the model works well.

In Fig. 10, as can be seen from the figure, according to the training method proposed in this paper, the model can converge to a stable value within 140 Epochs, which indicates the feasibility of the training method proposed in this paper. For different strokes, the neural renderer converges to different values: the point-like texture of chalk strokes is more complex, which is difficult for the neural renderer to learn. Watercolor strokes and oil strokes are relatively simple, and thus converge to higher precision, but due to the difficulty in learning the texture of the details of the oil strokes, the convergence precision is relatively low. This may be due to the fact that the strokes are relatively "light," and thus, the absolute value of the noise is small, resulting in a high PSNR value.

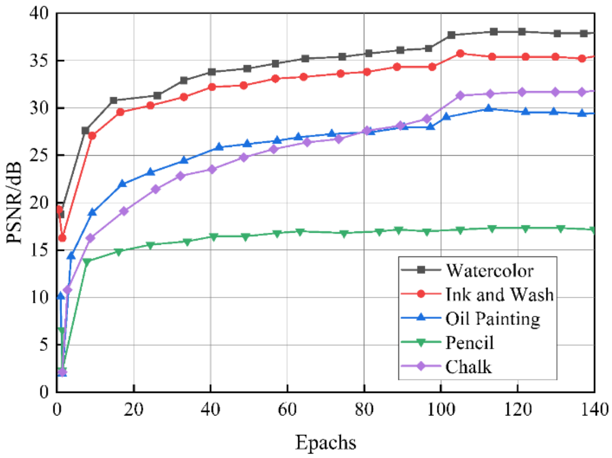

Fig. 10. Stroke renderer training content.

The translator is further applied to the reinforcement learning drawing simulation framework to translate the stroke control parameters generated by the intelligent body, and the drawing results of different stroke styles can be obtained. Overall, the translated drawing results can basically maintain the content of the original results, but the details may be lost, and the differences between different strokes are huge, as shown in Fig. 11.

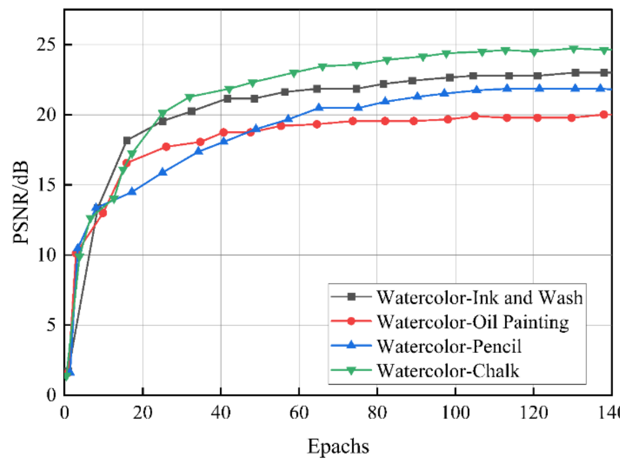

Fig. 11. Stroke translator training process.

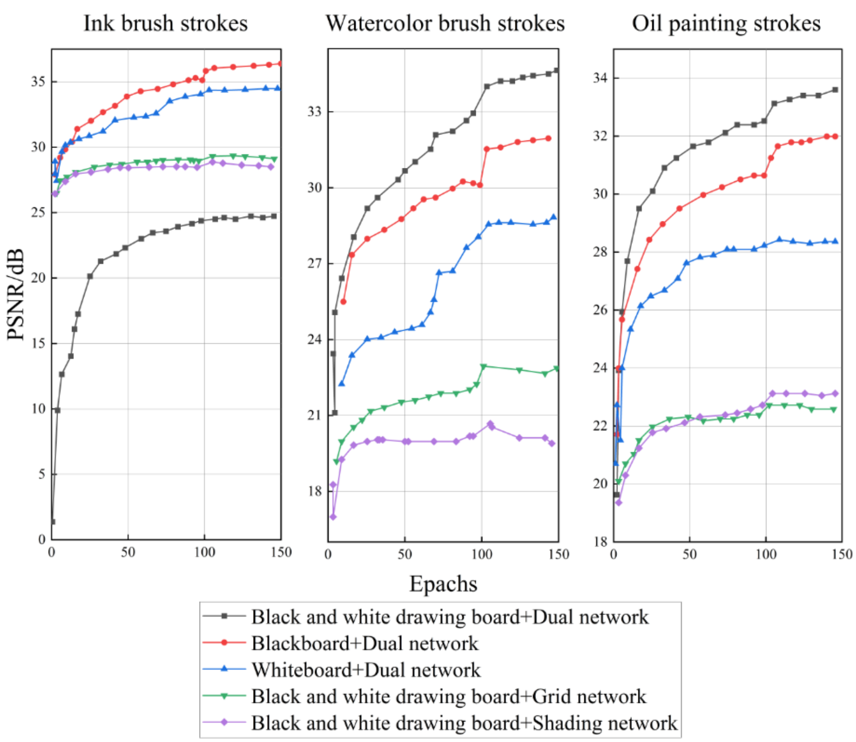

Fig. 12. Neural renderer ablation experiments.

In the case of training using black and white panels, neither the rasterization network nor the coloring network alone can achieve the best results, illustrating the effectiveness of the two-way network. In the case of using the two-way network, neither the black drawing board alone nor the white drawing board alone can achieve the best results, illustrating the effectiveness of the training method proposed in this paper (as shown in Fig. 12).

4) Experimental analysis and comparison: The paper also quantitatively analyzes the generated images and evaluates the quality of the generated images by the international common evaluation index IS. The larger the value of IS, the higher the quality of the generated images and the more realistic the effect, which mainly examines two aspects of the image performance: 1) whether the generated images are clear or not; 2) whether the generated images are varied or not.

In GAN, the conditional probability $p(y|\, x)$ is usually expected to be highly predictable. Where x denotes a given image, and y denotes a specific object contained in the image.

IS uses a fixed classification network, Inception Network, to realize the classification of the generated image and then predicts $p(y|\ x)$. In terms of diversity, the edge probabilities are computed using the following formula.

$$\int_z p(y|\ x=G(z))\mathrm{dz} \tag{11}$$

In summary, the formula for IS is as follows:

$$\mathrm{IS}(G)=e^{E_{x\in p}D_{\mathrm{kl}}\left(p(y|x)\|p(y)\right)} \tag{12}$$

where, Dx is the discriminant function containing the KL-Divergence constraint. In order to better validate the effect of this experiment, 1000 images were also generated using DCGAN and ProGAN respectively, and 10 evaluations were done using MNIST samples to take the mean value. The experimental results obtained are shown in Table II.

TABLE II. IS ASSESSMENT RESULTS

arithmetic	IS
GAN	0.8
DCGAN	1.44
ProGAN	1.87
Methodology of this paper	2.30

As can be viewed from Table I, the IS fee of the paper's approach is the highest, which shows that the photo generated the usage of the hierarchical fusion facets is of greater nice and has clearer and extra practical important points.

In this section, the portray consequences generated by way of the techniques in this paper are in contrast (the goal photograph is from the take a look at dataset), the quantitative assessment is completed via calculating the similarity index between the portray consequences generated by way of every approach and the goal photograph (taking the common price of the complete dataset), and the qualitative evaluation is finished through examining the true portray effects of one of a kind methods.

The parameters of several comparison methods were set as follows:

Using ink strokes, in order to make the size of the output image 128×128, the grid is set to 3x3, 23 strokes per grid totaling 207 strokes, and the number of iteration steps is 50;

Using watercolor brushstrokes, the size of the board is 128×128; using the pre-training model and parameters provided in the original article, the number of painting steps is 40, corresponding to the number of strokes is 40×5=200;

Using oil brush strokes, the board size is 128×128, and the number of strokes is limited to 200.

To summarize the floor plan rendering capabilities of the model: 1) the quality of CyceGAN rendered floor plan meets the need for clear presentation; 2) Epoch50 rendering is flatter, Epoch300 is rich in detail but has a certain chance of producing pixelated details, 3) the rendering of the nature of the land is accurately expressed, but the rendering of the structures is not prominent enough and needs to be manually further emphasized, and 4) the middle of the rendering colors is not as good as the middle of the rendering, and there is a certain lack of transition tones which are somewhat missing.

V. CONCLUSION

With the rapid development of Web technology, basic application systems based on b/s architecture have been widely developed and applied, and software based on b/s architecture has become the trend of software development. As a common form of artistic creation, painting is an important means of visual communication in the fields of news dissemination, prototype design, movie and television creation. The content of digital painting is easier to disseminate in the current network era and has important research and application value. In this paper, based on the previous research work, an image generation algorithm is proposed based on the intelligent rendering algorithm of image texture, which generates the structural line drawing through the structural GAN, and then through the rendering of the texture GAN to generate the ideal effect of the painting. The main conclusions are:

1) This chapter mainly introduces the related technical theories involved in the art image processing platform, the B/S network architecture which is popular among developers nowadays, the Python programming technology which has a great momentum, and the Django framework which is convenient for researchers in Python technology.

2) This chapter first introduces the model-based DDPG reinforcement learning approach as the main framework and clarifies in detail how reinforcement learning is applied to the task of painting simulation through three aspects: modeling, learning algorithm, and network architecture. Then, we introduce a two-way neural renderer for constructing the drawing simulation environment and a stroke translator that can realize the conversion of stroke control parameters.

3) Based on the existing open-source drawing board and renderer technologies, this paper defines a variety of stroke rendering methods and integrates them into a unified renderer framework to construct various drawing simulation environments. On this basis, this paper uses a unified two-way neural network structure and training method to realize a neural renderer that can be used for different stroke rendering. From the experimental results, it can be seen that the two-way neural renderer used in this paper combined with the DDPG reinforcement learning framework can effectively generate the drawing content of multiple stroke styles and realize the simulation of the drawing process; the stroke translator proposed in this paper can complete the expected translation of stroke control parameters and realize the migration of multiple stroke drawing styles.

REFERENCES

[1] Jiang Y. Computer Vision Object Detection Algorithm based on Convolutional Neural Network. Journal of Shenyang University of Technology, 2021, 43(5): 557-562.

[2] Chen J, Du M, Zheng J, et al. Double Level stroke line simplification

Method based on drawing time sequence. Journal of Computer-Aided Design and Graphics, 2019, 9.

[3] Kong S, Yin J. Design of real-time rendering system for 3D animation image texture. Modern Electronic Technique, 2018, 41(5): 102-105.

[4] Xu Q, Zhong S, Chen K, et al. Optimal selection method of CycleGAN Cycle consistent loss coefficient in image generation with different texture complexity. Computer Science, 2019, 46(1): 100-106.

[5] Ma Y, Xu X, Zhang R, et al. Research progress of generative adversarial networks and their applications in image generation. Journal of Computer Science and Exploration, 2021, 15(10): 1795.

[6] Chen F, Zhu F, Wu Q, et al. Review of generative adversarial networks and their applications in image generation. Chinese Journal of Computers, 2021, 44(2): 347-369.

[7] Liang J, Wei J, Jiang Z. A review of generative adversarial networks. Exploration of Computer Science and Technology, 2020, 14(1): 1-17.

[8] Wang K, Gou C, Duan Y, et al. Research progress and prospects of generative adversarial networks. Acta Automatica Sinica, 2017, 43(3): 321-332.

[9] Ye W, Gao H, Weng S, et al. A two-stage art font rendering method based on CGAN network. Journal of Guangdong University of Technology, 2019, 36(03): 47-55.

[10] Zhu Danni, Xu Xiao-Hua, He Jing-Jing, et al. Non-reference super resolution image quality evaluation using multi-layer perceptron regression. Journal of Xi'an Polytechnic University, 2022, 36(5).

[11] Popa T, Ibanez L, Levy E, et al. Tumor volume measurement and volume measurement comparison plug-ins for VolView using ITK. Medical Imaging 2006: Visualization, Image-Guided Procedures, and Display. SPIE, 2006, 6141: 395-402.

[12] Tang C, Yuan J, Xia C. Research and design of medical image reading system based on B/S structure. Computer and Digital Engineering, 2014, 42(2): 311-314.

[13] He J, Bao Y, Zhang J, et al. Design and implementation of Medical Laboratory Information Platform (LIs) based on B/s model. Computer Application and Software, 2016, 33(3): 83-86.

[14] Mukai Y U, Tokoi K. Watercolor Rendering with Consideration of Motion.

[15] Frans K, Soros L, Witkowski O. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. Advances in Neural Information Processing Systems, 2022, 35: 5207-5218.

[16] Huang X, Belongie S. Arbitrary style transfer in real-time with adaptive instance normalization. Proceedings of the IEEE international conference on computer vision. 2017: 1501-1510.

[17] Kotovenko D, Sanakoyeu A, Ma P, et al. A content transformation block for image style transfer. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 10032-10041.

[18] Lee J, Hwangbo J, Wellhausen L, et al. Learning quadrupedal locomotion over challenging terrain. Science Robotics, 2020, 5(47): eabc5986.

[19] Li G. Application of 3ds Max Rendering Technology in Virtual Scene Design Experimental Teaching. Art Education Research, 2015 (1): 170-173.

[20] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 2015, 28.

[21] Wang T, Gao X. A rendering method of multi-stroke anisotropic Van Gogh style oil painting. Computational Technology and Automation, 2017, 36(2): 125-128.

[22] Sanakoyeu A, Kotovenko D, Lang S, et al. A style-aware content loss for real-time hd style transfer. proceedings of the European conference on computer vision (ECCV). 2018: 698-714.

[23] Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE international conference on computer vision. 2017: 2223-2232.

[24] Kim T, Cha M, Kim H, et al. Learning to discover cross-domain relations with generative adversarial networks. International conference on machine learning. PMLR, 2017: 1857-1865.

[25] Levine S, Pastor P, Krizhevsky A, et al. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. The International journal of robotics research, 2018, 37(4-5): 421-436.

[26] Zhang L, Ping X, Zhang T. Image information camouflage algorithm with first-order statistical feature preserving. Journal of Computer-Aided Design and Graphics, 2005, 17(1): 99-104.

[27] Guan Q, Zhu J, Zhao X, et al. An image steganography method based on linear programming feature selection and integrated classifier. Journal of Cyber Security, 2018, 3(1).

[28] Zhang R, Isola P, Efros A A, et al. The unreasonable effectiveness of deep features as a perceptual metric. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 586-595.

[29] Isola P, Zhu J Y, Zhou T, et al. Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1125-1134.

[30] Sekhon A. Synthesizing Programs for Images using Reinforced Adversarial Learning.

[31] Kosugi S, Yamasaki T. Unpaired image enhancement featuring reinforcement-learning-controlled image editing software. Proceedings of the AAAI conference on artificial intelligence. 2020, 34(7): 11296-11303.

[32] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search. nature, 2016, 529(7587): 484-489.