

DiffTransfer: A Person Portrait Style Transfer Method Based on Stable Diffusion

Jizheng Wang *

Sydney Smart Technology College, Northeastern university, Qinhuangdao, China

* Corresponding author: 202119236@stu.neu.edu.cn

Abstract. With the widespread adoption of high-performance GPUs, NPUs, and other devices in recent years, the threshold for deploying large models on the edge has gradually lowered. Meanwhile, as social networks continue to thrive and virtual reality technology advances rapidly, there is an urgent demand for individuals to possess a realistic virtual identity to meet the ever-evolving social needs. In this context, portrait style transfer has attracted growing research interest. Different from the previous works that mainly follow the pipeline of generative adversarial network or variational autoencoder, this research proposes a method for person portrait style transfer based on the stable diffusion model. By pre-training underlying models, users only need to upload a small number of photos to generate corresponding LORA models for individuals. The approach involves style fusion using textual prompts and other models to modify portrait styles and generate images of individuals in different poses. Extensive visualization results validate the effectiveness of this work, which is able to generate realistic portrait images of people with different styles.

Keywords: Style Transfer, diffusion model, multimodal.

1. Introduction

In today's society, there is a high demand for social interaction, with social platforms such as TikTok, Weibo, and Douyin dominating the landscape of internet content. Within these platforms, the emphasis on highlighting one's personal image to showcase individual charm and the exploration of innovative elements while retaining personal characteristics has become a focal point for various entities. Simultaneously, with the release and gradual popularization of devices such as Apple's VisionPro, Microsoft's HoloLens, Meta's Quest, and others, an era of metaverse, where software and hardware converge, is inevitable. Therefore, an excellent algorithm for person portrait generation and style transfer holds tremendous potential for applications in this emerging metaverse era.

Many experts and scholars have conducted in-depth research and exploration into the problem of image style transfer. In fact, even before the advent of neural networks, there were some preliminary works on image style transfer. The concept of image style can be understood as a specific texture. As early as 2000, Portilla et al. [1] proposed a method for manually modeling a Parametric Texture Model based on the Joint Statistics of Complex Wavelet Coefficients. One crucial idea in this method is that texture can be described using the local features of an image. Guided by this concept, researchers used a plethora of complex mathematical models and formulas to deduce and generate textures. However, due to the manual modeling involved, such work was extremely time and labor-intensive.

It was only a decade later, with the rapid development of GPU performance and the explosive popularity of deep learning, that renowned object recognition networks such as VGG19 emerged. Networks like VGG19, fundamentally, are layer-by-layer local feature extractors. As mentioned earlier, image style can be understood as a specific texture, and these textures can be deduced and generated through the local features of an image. Consequently, based on neural networks, methods for image style transfer were born [2]. Gatys divided the image into two components: image content and image style. In the first step, a neural network is employed to apply the Gramian matrix to calculate the correlation of local features in different parts of the image, transforming it into a statistical model. This approach eliminates the need for manual modeling to obtain the texture features of the image. In the second step, an object recognition model is used again, directly treating the extracted local features as the image content. This process effectively divides an image into content

and style components. Finally, the content of one image is combined with the style of another image (Deep Dream), achieving image style transfer.

Since Gatys, improvements by Ghiasi and others have enabled real-time style transfer under different styles [3]. While these methods efficiently and stably accomplish the task of image style transfer, they are limited to a one-to-one operation. In other words, one input image generates one corresponding output image, lacking scalability and flexibility, thus falling short of meeting the creative demands of the metaverse era.

Compared to conventional methods, the emerging Diffusion model in recent years inherently possesses powerful diversity and flexibility, finding wide applications in various fields such as object detection, semantic segmentation, and image generation, all yielding impressive results [4,5,6]. In this study, the Diffusion model is utilized to model the content dataset and extract features from the style dataset. The pre-trained model based on stable diffusion is then employed for image generation, and text-controlled image content is achieved through the Clip model. Thus, a person portrait style transfer method based on stable diffusion is established, enabling the generation of a large number of diverse and editable images with different styles by learning from a small set of images.

2. Method

2.1. Revisiting Stable Diffusion

Stable Diffusion is primarily composed of three components: the Clip model, Diffusion model, and VAE model. The most crucial component among them is the Diffusion model. The overall workflow of Stable Diffusion is outlined as follows [6] in Fig 1.

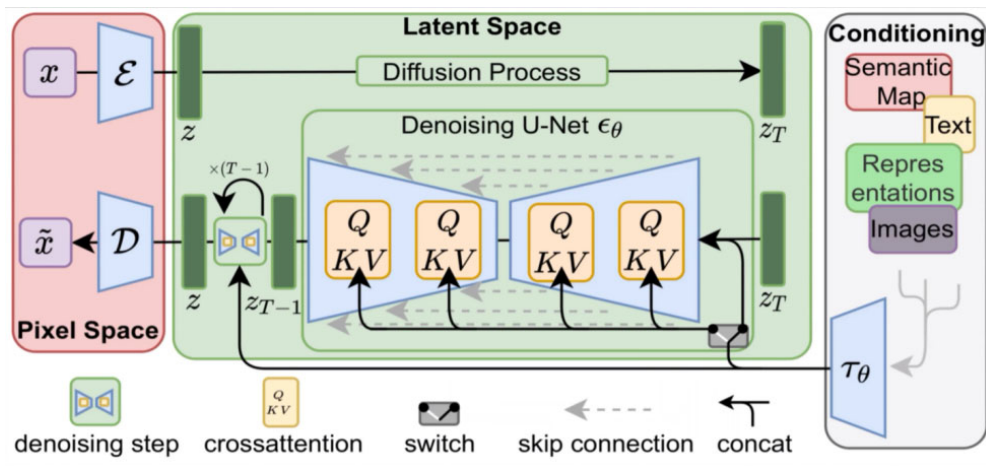


Figure 1. Workflow Diagram of Stable Diffusion

In the Diffusion model, there are primarily two processes: (1) Forward process, continuously adding noise to the original image. (2) Reverse process, continually removing noise to restore the original image. As shown in Fig 2, in the forward process, we progressively introduce Gaussian noise to the original image. At each time step t , the input is the image x_{t-1} from the previous time step $t - 1$, Gaussian noise is added to obtain the image x_t , and this image is then input into the next time step $t + 1$ for further noise addition. After iteratively applying the noise addition process multiple times, combining the reparameterization technique and the Markov chain property, we can derive the final mathematical expression for the noise distribution as follows:

$$q(x_{1:T}|x_0) = \prod_{t=1}^T q(x_t|x_{t-1}) \quad (1)$$

The reason for introducing noise instead of training a network directly in one step is mainly because directly removing pixels would lead to information loss. Adding noise allows the model to learn more image features, thereby increasing diversity during generation.

In the reverse process, i.e., image restoration, it involves iteratively denoising from random noise to ultimately obtain a clear image. As shown in Fig 2, utilizing a U-Net neural network, the input is an image with T steps of added noise. The U-Net predicts the noise at time t , and then the input image is subtracted by the noise to obtain the original image x_{t-1} . However, due to the presence of considerable noise, the network cannot predict accurately, resulting in the output image losing details and appearing very blurry. At this point, we treat this output image x_{t-1} as the original image, add slightly less noise than before ($T - 1$), and then input it into the U-Net for denoising prediction. This process is iterated continuously, ultimately yielding the clear original image x_0 .

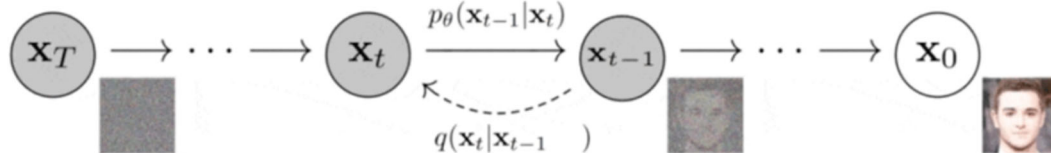


Figure 2. Forward and Reverse Processes in the Diffusion Model

However, the aforementioned process is not the actual workflow of Stable Diffusion. In fact, any manipulation of images in Stable Diffusion occurs in the latent space because current home computers are practically incapable of handling such massive data volumes in pixel space. Stable Diffusion introduces the VAE model, which uses an encoder to compress the original image to a size of $4 \times 64 \times 64$ [7]. This significantly improves processing speed and reduces computational load. After completing the forward noise addition and reverse denoising processes, the image is restored to its original size through the decoder. The Clip model [8] is primarily used to convert text into text features, which are then added to the U-Net network. By employing the Class-Free Guidance method, it predicts two types of noise: one guided by text features and the other without guidance. Subtracting the two yields the changes influenced by the text guidance. This signal of change is then amplified, combined with the noise without text feature guidance, resulting in enhanced noise guided by text. This process facilitates the generation of images that align with the textual content. The overall process is as follows: the input image is compressed into the latent space through the VAE encoder, then noise is added. The image with noise added for T steps is input into the U-Net. Simultaneously, the text guidance, converted into text features through the Clip model, is also input into the U-Net. The U-Net iteratively restores the original image at $t=0$, and finally, it is restored to its original size through the VAE decoder.

2.2. Dataset Construction

2.2.1. Collection of Content Images

Testing images were captured on October 20, 2023, indoors, using a mobile phone (iPhone 13 Pro) with the front camera (23mm f2.2). The images were sampled by handholding the phone and capturing various directions of the user's face. A total of 57 original images were collected with a resolution of 2316x3088. Among them, 18 images were taken from a frontal angle, 27 from a side angle, and 12 were full-body shots. The image format is HEIF. A subset of the original image set is shown in Fig 3.



Figure 3. Thumbnail of a Subset of the Original Image Set

2.2.2. Preprocessing of Content Dataset

For the convenience of subsequent training stages, preprocessing is applied to the content dataset. Initially, the resolution of the original images is reduced, cropping them from the original size of 2316×3088 to 512×512 . Images with overly similar angles are manually filtered, resulting in a final set of 41 images with suitable angles and a size of 512×512 . This step reduces the total number of steps in each training round, thereby accelerating the process and lowering computational load. Subsequently, these 41 images are labeled. The WD 1.4 labeler, integrated into Stable Diffusion, is utilized for automated labeling, employing the built-in `milingWolf/wd-v1-4-convnextv2-tagger-v2` model for the content dataset. After the automated labeling by the WD 1.4 labeler, the `BooruDatasetTagManager` is employed for manual label management, ensuring that each image's label aligns with its content. Once verified, the `BooruDatasetTagManager` is used for batch management of all labels, adding character names for subsequent keyword invocation of this Lora model in text prompts [9].

2.2.3. Preparation of Style Dataset

To achieve style transfer, similar to traditional style transfer tasks, it is necessary to prepare a dataset for the target style. Just as with preparing the content dataset, you can collect images for the target style and train them into a usable model. For ease of training, it is recommended to use a space-efficient Lora model. The preprocessing steps for the dataset are the same as those described for the content dataset above. Once the model training for the style dataset is complete, you can adjust the weight ratios of the style dataset and the content dataset in influencing the generated images through text prompts. By blending the features of both datasets, you can achieve the desired style transfer effect. Additionally, since target styles often represent well-known artistic styles or characters, you may choose to download pre-trained models available on the internet. Common websites for this purpose include <https://civitai.com>, among others.

2.3. Model Training

After completing the preprocessing of the dataset, the next step is model training. The computing platform for this work is a laptop with an AMD Ryzen 5800H processor, 16GB RAM, an Nvidia RTX 3060 Laptop GPU, and the Windows 11 23H2 operating system.

To facilitate training and reduce computational overhead, it is necessary to opt for fine-tuning models rather than large models that train the entire U-Net structure. Common fine-tuning models for Stable Diffusion include the Lora model, Dreambooth model [10], and Textual Inversion model [11]. As shown in Fig 4, the Dreambooth model directly adjusts the parameters in the U-Net based on new images and keywords, so saving the model requires storing the entire U-Net model, which occupies a large space. The Lora model, on the other hand, adds some side branches to the U-Net. The parameters obtained from the training dataset are used to adjust the parameters of these branches, thereby tuning the output of the U-Net through these branches. When saving, only these layers need to be saved, occupying less space. However, since the Lora model itself does not include the U-Net structure, a base model is needed as the underlying structure when using it. Textual Inversion adjusts the Clip model to output text features that match the features of the images in the training dataset, resulting in even smaller space usage.

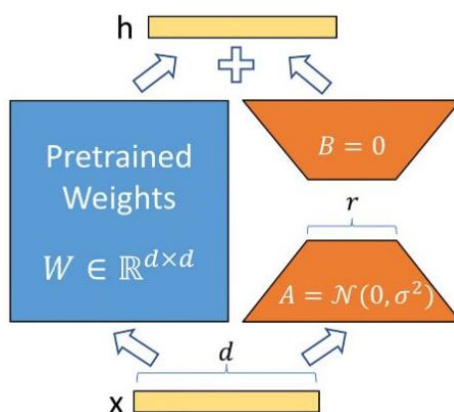


Figure 4. Lora Model Weighted Fusion with Pre-trained Model through Side Branches

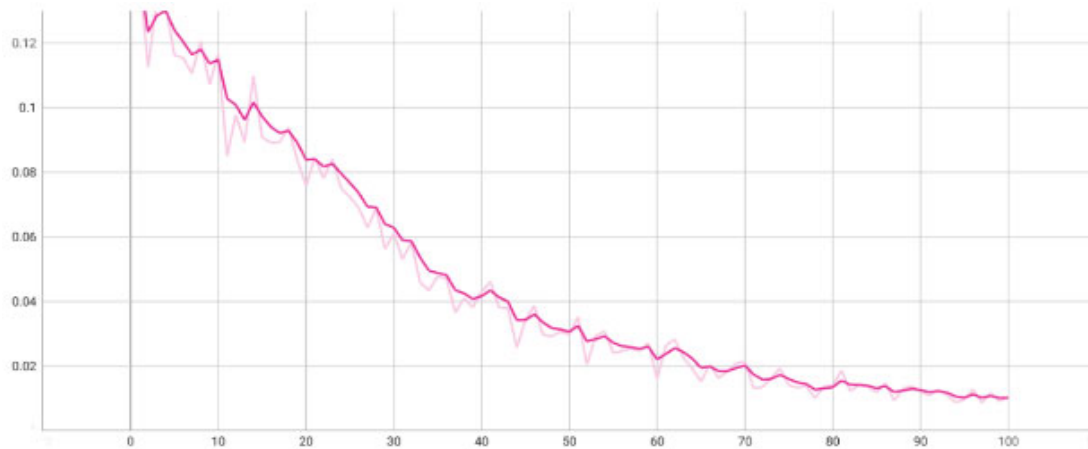
Taking into account the characteristics of various fine-tuning models, the Lora model is a relatively moderate choice. It occupies a small space, has modest performance requirements, and exhibits good generalization. Therefore, in this work, the Lora model is chosen for training with the following specific parameter settings: the pretrained_model is set to v1-5-pruned.ckpt. This model is a parent model officially released by the developers and serves as the foundation for many subsequent sub-models of different styles. Choosing this model as the base model for Lora training is beneficial for improving the model's generalization. A total of 100 rounds of training will be conducted, saving the model after each round. The subsequent results section of this document will discuss the descent of the loss function at different rounds and the practical effectiveness of the model.

3. Result and Analysis

3.1. Model convergence analysis

To find the optimal learning rate, the optimizer is first adjusted to DAdaptation[12], the LR Scheduler is set to constant, Text Encoder Learning rate is set to 0.5, and Unet Learning rate is set to 1. Since the trained model in this work is focused on human portrait, both Network dim and Network alpha are adjusted to 64 to enhance precision. During the initial stages of training, the learning rate change curve is observed using Tensor Board. It is noted that the learning rate sharply increases at the beginning and then stabilizes at 1.06e-4. This value is identified as the optimal learning rate under the AdamW8bit setting.

With the optimizer set to AdamW8bit, U-net Learning rate at $1.06\text{e-}4$, and Text Encoder Learning rate at $1.06\text{e-}5$, training begins with each image repeated 5 times. After 100 epochs of iteration, the curve depicting the change in loss is shown in Fig 5.



(a) Loss Curve with Iteration Rounds



(b) Loss Curve with Steps

Figure 5. Visualization of loss during the training procedure

3.2. Visualization and Performance Analysis

3.2.1. Analysis of Model Fitting Ability at Different Rounds

To analyze the model fitting ability at different rounds, an additional experiment is conducted. By selecting Lora models generated at iteration rounds 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100, with corresponding loss values of 0.1416, 0.1165, 0.0757, 0.0610, 0.0428, 0.0295, 0.0159, 0.0212, 0.0137, 0.01197, and 0.0101, the generative effects of the images were observed. Each model at every iteration round generated 7 images with weights of 0.2, 0.4, 0.6, 0.7, 0.8, 0.9, and 1. An xy graph was established with the epoch as the x-axis and weights as the y-axis, resulting in a total of 77 images. To test the model's fitting ability to the original characters, positive prompts were utilized from labeled data in the training set, ensuring the generated images closely match the originals. Specific prompts of Style 1 include: wangjizheng, 1boy, solo, male focus, glasses, realistic, shirt, black hair, tile wall, looking at viewer, blue shirt, closed mouth, short hair. The iteration steps were set at 30, using the DPM++SDE Karas sampling method, with a resolution of 512×512 . The prompt guiding coefficient was set to 7, and the SD base model used for testing was majicmixRealistic_V7, representing a realistic style that aligns with the original image style. The results are shown in the following Fig 6.



Figure 6. Visualization of generation results with Style 1

Based on the analysis of the images, it can be observed that when the weight is less than 0.4, the fitting of generated images is poor. When using the prompt "tile wall," the generated images do not replicate the same wall as in the original images, and the character representation differs significantly from the originals. When the weight exceeds 0.9, there is a tendency for overfitting. Models beyond epoch 1 exhibit facial distortions, strong smearing in the images, and the presence of many artifacts.

Focusing on images with weights between 0.6 and 0.9, let's analyze the generative effects of the models at different rounds. At epoch 1 (loss 0.1416), there is evident underfitting, with significant differences in facial details, character movements, clothing details, and scene details compared to the original image. At epoch 10 (loss 0.1148) and epoch 20 (loss 0.0838), the fitting degree of facial features significantly improves. Glasses texture and hairstyle show notable restoration to the original image. Simultaneously, the fitting degree of scene details increases, with wall color and tile size resembling the original image. However, character movements and clothing details do not match the original image. From epoch 30 (loss 0.0627) to epoch 100 (0.0101), the fitting effect continues to improve. Apart from facial details and scenes, character movements and clothing details are essentially consistent with the original image, demonstrating good fitting characteristics.

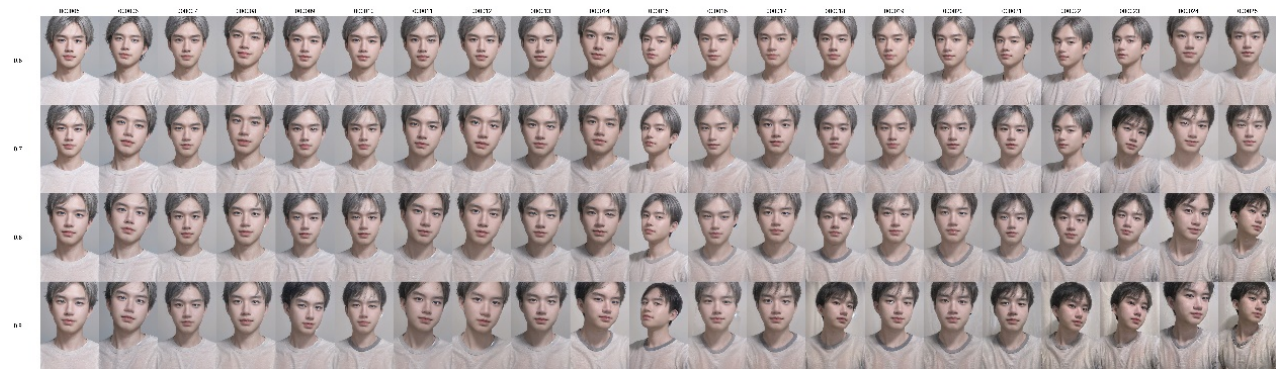
3.2.2. Generalization Analysis of the Model at Different Iteration Rounds

In order to test the model's generalization, some elements that do not exist in the original training set were modified in the prompt, such as changing facial details to gray hair, blue eyes, and altering the scene to a white solid background. The specific prompt of Style 2 is as follows: (1 boy: 1.4), (white background: 1), (white T-shirt: 1.2), (masterpiece, best quality), intricate details, (grey hair: 1.2), (blue eyes: 1.2), handsome boy, white skin, sharp jawline, thin, ((slim)), messy hair, lips, upper body, close up, smirk. The iteration step is 30, the sampling method is DPM++SDE Karas, the resolution is 512x768, the prompt guidance coefficient is 7, and the SD base model is majicmixRealistic_V7. After the previous verification, the Lora model performed well with weights between 0.6 and 0.9. Therefore, in this test, the Lora models generated at iterations 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 are taken as the x-axis, and weights 0.6, 0.7, 0.8, 0.9 are taken as the y-axis to generate an xy chart to analyze the model's generative effects, as shown in Fig 7.



Figure 7. Generalization Effects at Different Weights and Iteration Rounds

According to the analysis of the figures, the model exhibits good generalization performance from epoch 1 (loss 0.1416) to epoch 20 (loss 0.0627), accurately reproducing the described elements in the text prompts, such as gray hair, white background, and blue eyes. However, the generalization performance declines after epoch 30, failing to reproduce elements like gray hair from the prompts and retaining the characteristics of black hair from the original images. At weight 0.9, there is background contamination, and the painting effect is strong, indicating lower model usability. In epoch 1, although the generalization performance is good, the fitting degree is too low, resulting in a significant difference from the appearance of characters in the training set. To find the optimal balance between fitting and generalization, experiments were conducted for epochs 5-25, and the results are in Fig 8.



(a) Generation Results from Epochs 5 to 15



(b) Generation Results without Using Lora Model with the Same Prompts

Figure 8. Visualization of generation results with Style 2

According to the analysis of the images, it can be observed that the model performs well between epoch 9 (loss 0.1137) and epoch 21 (loss 0.08408). The model demonstrates excellent generalization ability by accurately reproducing gray hair, blue eyes, and a white background. Before epoch 9, the model lacks fitting ability, resulting in significant differences from the original character. Beyond epoch 21, the model's generalization ability diminishes, failing to correctly generate gray hair and blue eyes, and exhibiting background contamination. Considering both generalization and fitting performance in this training round, epoch 13 (loss 0.08927) stands out as the optimal choice.

3.3. Style Transfer Effect Analysis

Continuing with the prompts used in 3.2.2 ((1 boy:1.4), (white background:1), (white T-shirt:1.2), (masterpiece, best quality), intricate details, (grey hair:1.2), (blue eyes:1.2), handsome boy, white skin, sharp jawline, thin, ((slim))), messy hair, lips, upper body, close up, smirk), generating images under different styles using the Lora model trained on epoch 13. First, using the cuteyukimixAdorable model to generate images in a cute style. The results are in Fig 9.

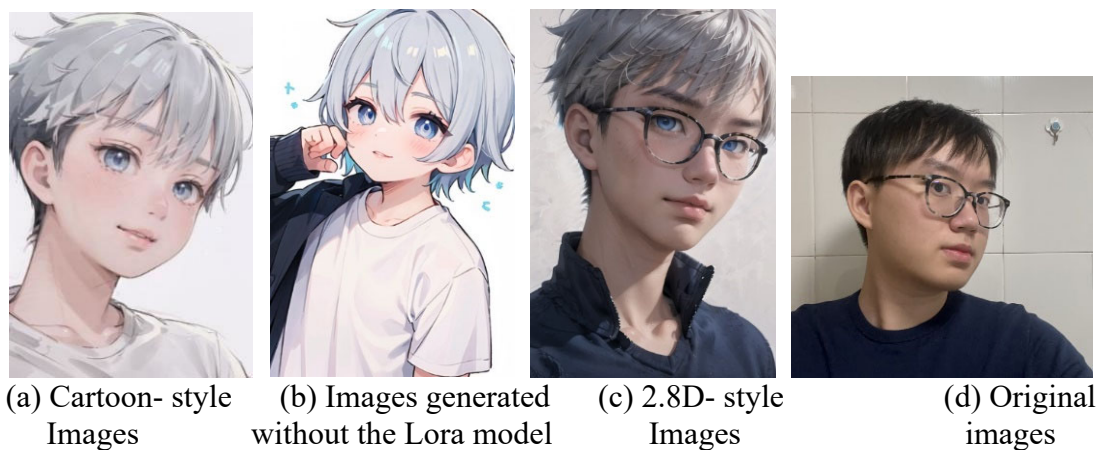


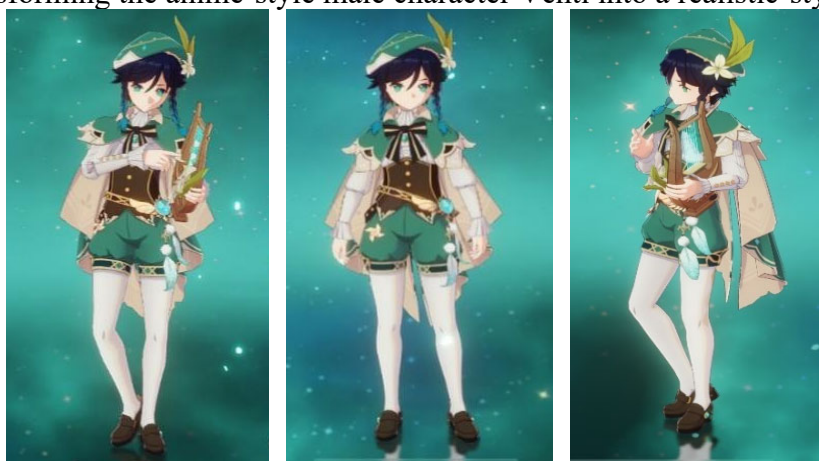
Figure 9. Visualization of generation results in different styles

By comparing Fig 9 (a) and 9 (b), it can be observed that there is a significant difference in the generated results between the two under the influence of the trained model. Examining the hairstyle, lips, nose, and facial features in Fig 9 (a), a clear correlation with the original training set is evident. Subsequently, an attempt was made to transition to the 2.8D style using the aniverse model, incorporating more features from the training set into the prompts for a better comparative assessment of the transfer effects. In Fig 9 (c), by adding the prominent prompt word "glasses" to test the fitting ability, the analysis of the images reveals that the details of the glasses in the generated images are almost identical to those in the original images. Moreover, the evident style transfer effect and changes in hair color, pupil color, etc., indicate a strong generalization ability of the model.

In the next set of tests, the character Venti from the game Genshin Impact is used as the content dataset, and the corresponding Lora model is trained to test the style transfer effect. In this set of experiments, as shown in Fig 10, not only did the images transition from an anime style to a realistic style, but Venti was also transformed from a male to a female figure. The specific prompt is as follows: masterpiece, best quality, 1 girl, female focus, plump chest, Venti (Genshin Impact), lyre, black hair, hat, smile, braid, long sleeves, twin braids, gradient hair, multicolored hair, instrument, pantyhose, vision (Genshin Impact), bow, shorts, blue hair, green headwear, beret, white pantyhose, green shorts, collared cape, bangs, shirt, jewelry, looking at the viewer, white shirt, brooch, cape, androgynous, frilled sleeves, frills, corset, short hair with long locks, (grey background: 1.4), (kbxll: 0.6), <lora: venti: 1>. The base model used is majicmixRealistic_V7, with 20 iterations, DPM++SDE Karras sampler, and a resolution of 576×832 . Analyzing the generated results, it is observed that the fitting is good, and details of clothing and hairstyle perfectly reproduce the original character. Additionally, facial details and body proportions are consistent with female characteristics, demonstrating good generalization.



(a) Transforming the anime-style male character Venti into a realistic-style female



(b) Original appearance of Venti in Genshin Impact

Figure 10. Transfer results of Venti in Genshin Impact

In the final set of experiments, the style transfer effect is tested by combining different weights of multiple Lora models, targeting the Iron Man style, with settings `<lora: ironman: 1>`, `<lora: JZ-000013: 0.7>`. The generated results are shown in Fig 11.



Figure 11. Different style base models after combining Iron Man and JZ Lora models

4. Discussion

The above results indicate that using Stable Diffusion, training a generative Lora model for the content dataset and applying it to various base models with different styles or blending it with other Lora models in certain proportions can achieve effective style transfer. With the evolution of technology in social media platforms and the increasing demand for personalized and customized content from users, the application prospects of this technology in related fields are undoubtedly vast.

As observed earlier, the feasibility of implementing style transfer based on Stable Diffusion is relatively high, and the actual results are promising. However, under the current technological

conditions, there are still some issues. For instance, in Fig 10, the portrayal of the left hand's form appears abnormal and does not conform to the correct human body structure. This is caused by various factors. Firstly, in the workflow of Stable Diffusion, the first step involves compressing the image to latent space through the VAE encoder. While this reduces the computational load, it unavoidably decreases accuracy, leading to poorer learning outcomes for certain details in the images. In addition to the issue with hand details, poor handling of text (as seen in the upper-left wall in Fig 11) further supports this point. The second reason is the limited targeted training for hands. In most existing models focusing on characters, the training dataset primarily consists of facial features and the overall structure of the body. Due to the lack of abundant hand-related materials, the model's generation effect for faces and bodies is more accurate, while the generation effect for hands is subpar. In fact, there are already optimization methods for details such as hands, such as ControlNet [13], which uses techniques like Canny, depth, OpenPose, etc., to recognize the target pose map. By introducing an external network, it controls the parameters of the decoder in the Unet model to precisely control details like hand posture and body movements. In the future, with the emergence of more methods like ControlNet, addressing the issues of poor image detail effects and high image randomness in diffusion models is only a matter of time.

5. Conclusion

In response to the demand for style transfer, this paper proposes a method for portrait style transfer based on the stable diffusion model. The paper briefly introduces the development history of style transfer, the basic principles of stable diffusion, and then focuses on showcasing the effects of style transfer using the stable diffusion model.

The paper tested the fitting and generalization capabilities of models at different epochs during 100 epochs. The results indicate that the fitting capability of the portrait model reaches a usable level when the loss value is below 0.12. As the loss value decreases, the fitting capability gradually improves. However, it was observed that the model's generalization capability significantly decreases when the loss value is below 0.8. Taking both fitting and generalization capabilities into account, the model performs well when the loss value is around 0.9.

Different styles, including 2D, 2.8D, and realistic styles, were tested for their generation effects. The results show excellent style transfer effects in various styles, demonstrating the feasibility of this method for style transfer tasks. It successfully achieves the goal of generating a large number of different styles and editable images with minimal training images.

Finally, as mentioned in this paper, methods based on stable diffusion perform poorly in generating details such as hands, and the images are relatively uncontrollable. In the future, combining methods like ControlNet can enhance the model's control over details and improve the controllability of the images.

References

- [1] Portilla, J., Simoncelli, E.P. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. *International Journal of Computer Vision* 40, 49 – 70 (2000).
- [2] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." *arXiv preprint arXiv: 1508.06576* (2015).
- [3] Ghiasi, Golnaz, et al. "Exploring the structure of a real-time, arbitrary neural artistic stylization network." *arXiv preprint arXiv: 1705. 06830* (2017).
- [4] Chen, Shoufa, et al. "Diffusiondet: Diffusion model for object detection." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.
- [5] Baranchuk, Dmitry, et al. "Label-efficient semantic segmentation with diffusion models." *arXiv preprint arXiv: 2112. 03126* (2021).

- [6] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.
- [7] Andrew. How does Stable Diffusion work? Retrieved on November 18, 2023. Retrieved from <https://stable-diffusion-art.com/how-stable-diffusion-work/>.
- [8] Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PMLR, 2021.
- [9] Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv: 2106.09685 (2021).
- [10] Ruiz, Nataniel, et al. "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.
- [11] Gal, Rinon, et al. "An image is worth one word: Personalizing text-to-image generation using textual inversion." arXiv preprint arXiv: 2208.01618 (2022).
- [12] Defazio, Aaron, and Konstantin Mishchenko. "Learning-rate-free learning by D-adaptation." arXiv preprint arXiv: 2301.07733 (2023).
- [13] Zhang, Lvmin, Anyi Rao, and Maneesh Agrawala. "Adding conditional control to text-to-image diffusion models." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023.