Enhancing Photo Animation: Augmented Stylistic Modules and Prior Knowledge Integration

Zhanyi Lu, Yue Zhou(✉), and Ao Chen

School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
{luzhanyi,zhouyue,llykevin}@sjtu.edu.cn

Abstract. Photo-to-animation translation presents a practical and captivating task within image style transfer. However, existing methods often fall short of achieving satisfactory results in cartoonization. This inadequacy primarily stems from two key factors: the absence of dedicated network architectures tailored for anime-style transfer and the inadequate incorporation of pertinent prior knowledge specific to cartoons. In response to these limitations, this paper introduces a novel deep neural network architecture designed to optimize photo-to-animation translation. Specifically, the proposed framework consists of two pivotal modules: the SCAN module and the Ada-CTSS module, operating at the feature and image levels, respectively, to enhance the desired anime-style effects. We also leverage prior knowledge, encompassing color, texture, and surface aspects, by integrating refined color preservation loss, grayscale style loss, and region smoothness loss. Moreover, to assess the efficacy of our approach, we devise a specialized style evaluation network, circumventing the reliance on conventional evaluation metrics. Through an extensive array of experiments, we demonstrate the superior capabilities of our method in generating high-quality cartoonized images, surpassing the performance of state-of-the-art methods.

Keywords: Generative Adversarial Networks · Photo Cartoonization · Image Translation

1 Introduction

Animation is a significant and culturally important art form used in advertising, education, and entertainment[35]. Creating anime is time-consuming and requires specialized expertise. Photo cartoonization aims to automate the conversion of natural scenes into anime style, reducing the technical threshold for artistic creation and providing digital artists with more expressive freedom.

Anime exhibits unique traits, such as clear edges, smooth colors, and recognizable content. Traditional texture transfer methods are unsuitable, while neural style transfer (NST)[10] aligns input photos with reference artworks but diverges from the animation style[16]. Applying generative adversarial networks (GANs)[11] to photo cartoonization is challenging due to the need for large

paired datasets[15]. While CycleGAN[36] overcomes this limitation, it compromises content preservation and adds optimization complexity. CartoonGAN[6] specifically targets photo-to-animation translation, incorporating perceptual loss[17] and edge preservation, but accurately capturing animation characteristics remains difficult. AnimeGAN[5] improves on this approach, whereas challenges persist. White-Box[33] dissects anime style; however, it lacks expressive color and texture. CTSS[9] guides the generator using salient texture patches but produces messy textures.

The inadequacy of previous methods lies in their network structures and integration of anime-style prior knowledge. By drawing inspiration from artistic style transfer theory [16], we interpret the significance of normalization layers in photo cartoonization and introduce the SCAN (style correlation adaptive normalization) module to enhance anime stylization at the feature level. To improve the CTSS approach, the Ada-CTSS (adaptive cartoon texture saliency sampler) module is proposed, utilizing spatial attention [34] to focus on salient texture regions. Furthermore, animation characteristics are incorporated into the loss functions, encompassing region smoothness, grayscale style, and refined color preservation loss. To evaluate anime stylization, we have developed a customized feature fusion evaluation network, which serves as an alternative solution to FID[12] and alleviate the negative impact of the ImageNet-trained InceptionV3 model[29,7].

In summary, our contributions include 1) a novel network architecture for photo-to-animation translation, enhanced by the SCAN module for anime stylization and Ada-CTSS module for localized focus; 2) integration of anime knowledge, stylizing images with region smoothness, grayscale style, and refined color preservation loss; 3) a style evaluation network that learns anime-specific features, providing reliable and straightforward quantification of anime stylization. Experimental results confirm our method's generation of high-quality anime-style images, surpassing previous state-of-the-art approaches.

2 Related Work

2.1 Neural Style Transfer

Style transfer renders images in different artistic styles while preserving original content. Traditional image-processing-based methods often struggle to learn effective style and content features, limiting their stylization effects[16]. Gatys et al.[10] first introduced deep learning to style transfer, minimizing Gram loss between generated and reference VGG features[28] to imbue images with artistic style. Johnson et al.[17] and Ulyanov et al.[32] accelerated stylization using feed-forward convolutional networks, enabling real-time stylization. Researchers extended style transfer to multi-style[8,22] and arbitrary-style transfer[14] through normalization layer improvements and explored alternative loss functions[16,20,24,19]. Li et al. [21] summarized artistic stylization as a domain adaptation problem, aligning feature distributions between generated and reference images with different methods, resulting in diverse "brushstroke" styles.

However, feature alignment generates variable brushstroke patterns unsuitable for the characteristics of animation. Moreover, these methods typically learn style from a single artistic image, making it challenging to separate style from content within a single sample. Therefore, the effectiveness of style transfer heavily relies on the reference artistic image samples.

2.2 Image to Image Translation with GANs

GANs[11,25] have made impressive strides in image translation, excelling in tasks like restoration, super-resolution, and style transfer. Pix2Pix[15] achieves reliable results but requires paired data, limiting its practicality due to extensive data requirements. Addressing this issue, CycleGAN[36] introduces cycle-consistency loss, accomplished through concurrent training of two sets of generators and discriminators. Nonetheless, this dual optimization imposes heightened training complexities, and CycleGAN's deficiency in semantic preservation restricts its applicability to distinct anime styles. Conversely, CartoonGAN[6] replaces cycle-consistency loss with perceptual loss, eliminating the necessity for concurrent GAN training, and incorporates an edge adversarial term to produce well-defined edges. AnimeGAN[5] further refines this approach by lightweighting network architecture and introducing style loss functions. However, despite these advancements, their outcomes still inadequately capture anime characteristics. White-Box[33] analyzes anime image characteristics, segregating them into interpretable structural, textural, and surface features. Nevertheless, lacking an efficient network structure, White-Box[33] fails to represent anime style in color and texture faithfully. Meanwhile, CTSS [9] designs a sampler module to extract salient texture regions and enhance stylization effects. However, its methodology of selecting significant regions based on edge pixel counts results in cluttered and disorganized textures.

To overcome these limitations, we introduce the SCAN module and Ada-CTSS module, which enhance network stylization capabilities. Furthermore, we design specialized loss functions that take into account anime characteristics, aiming to improve anime style transfer.

3 Method

3.1 Network Structure

Our GAN-based style transfer framework, illustrated in Figure 1, employs a generator G tailored for photo cartoonization. G adopts a U-Net-like architecture[26], consisting of three main components: an encoder, inverted residual blocks (IRB)[27], and a decoder. Notably, the SCAN module is incorporated within the IRB. The discriminator comprises both a global discriminator D_{img} and a local discriminator D_{patch}, with the latter utilizing patches extracted by the learnable Ada-CTSS module.

We commence by discussing the significance of normalization layers in style transfer tasks, followed by an elucidation of the design principles and methodologies underlying the SCAN module. Lastly, we introduce the Ada-CTSS module.

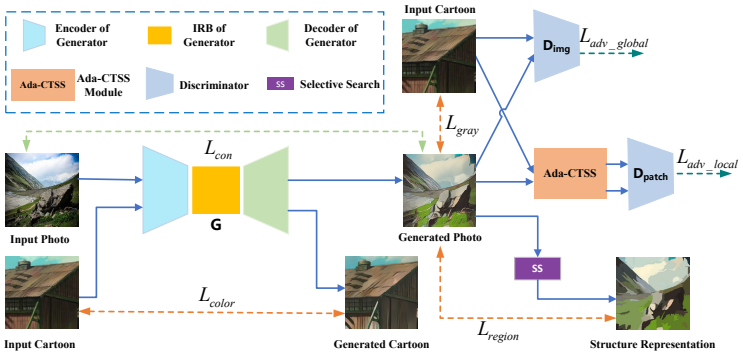

Fig. 1. Structure of the proposed method. The dashed connecting lines in different colors represent the three critical components of the loss function.

Style Correlation Adaptive Normalization Normalization expedites network training and augments generalization by standardizing input to a Gaussian distribution, thereby adjusting feature distribution via learnable parameters to counteract normalization-induced information loss[13]. However, we propose an alternative perspective on normalization layers: *The Influence of Parameters on Feature Maps*, which modify the distribution of features across entire regions of a single channel. Considering the conclusion drawn from style transfer tasks that the statistical properties of feature maps represent a style [16], it follows that the parameters of normalization layers wield significant influence on image style compared to other learnable structures in the network.

In style transfer tasks, where the primary focus lies on individual image quality, Instance Normalization (IN) [31] is frequently utilized. AnimeGAN later replaced IN with Layer Normalization (LN) [3] for more stable stylization effects without providing a clear explanation. We delve into this phenomenon and propose a superior normalization method—style correlation adaptive normalization(SCAN).

Assuming the feature maps of a certain network layer $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, where $\mathbf{X}_i \in \mathbb{R}^{H\times W}$ denotes the feature map of the ith channel, its output after IN and LN can be respectively represented as:

$$\tilde{\mathbf{X}}_{i-\mathbf{IN}} = \gamma\left(\frac{\mathbf{X}_i - \mu_i}{\sqrt{\sigma_i^2+\varepsilon}}\right) + \beta \quad \tilde{\mathbf{X}}_{i-\mathbf{LN}} = \gamma\left(\frac{\mathbf{X}_i - \mu_c}{\sqrt{\sigma_c^2+\varepsilon}}\right) + \beta, \tag{1}$$

where μ_i and σ_i represent the mean and standard deviation of the ith channel, while μ_c and σ_c denote the mean and standard deviation across all channels. γ and $\beta \in \mathbb{R}$ are learnable parameters.

The difference between IN and LN lies in the calculation scope of the mean and standard deviation used during normalization. In IN, the learnable parameters normalize the features of the ith channel solely using the mean and standard deviation computed within that channel. However, in LN, the statistics used for

normalization are calculated from features across all channels. Specifically,

$$\begin{aligned}\tilde{\mathbf{X}}_{i-\mathbf{LN}} &= \gamma\left(\frac{\mathbf{X}_i - \mu_c}{\sqrt{\sigma_c^2 + \varepsilon}}\right) + \beta = \gamma\left(\frac{\mathbf{X}_i - \mu(\mathbf{X}_i, \mathbf{X}_{i:c})}{\sqrt{\sigma\left(\mathbf{X}_i, \mathbf{X}_{i:c}\right)^2 + \varepsilon}}\right) + \beta \\ &= \gamma \cdot f\left(\mathbf{X}_i, \mathbf{X}\right) + \beta.\end{aligned} \tag{2}$$

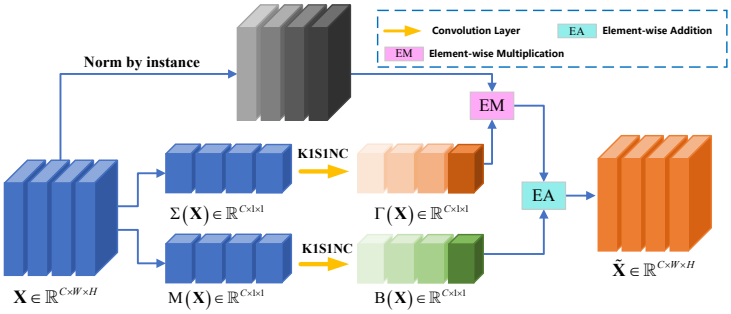

Fig. 2. SCAN Module. $\mathbf{\Sigma}(\mathbf{X})$ and $\mathrm{M}(\mathbf{X})$ are the channel-wise standard deviation and mean of $\mathbf{X}$. K, S, and N represent the kernel size, stride, and kernel count, respectively.

In Equation 2, $\mathbf{X}_{i:c}$ represents all channel features except $\mathbf{X}_i$. As mentioned earlier, the statistical properties of feature maps represent a style. When adjusting the feature distribution of the ith layer(i.e., changing the style of the ith channel), γ and β consider the information of both the ith layer features $\mathbf{X}_i$ and all other layer features $\mathbf{X}_{i:c}$. The inter-feature correlations are expressed through a function f, leading to more stable results. However, this correlation is based on a fixed function, limiting the expression of feature correlations. In cases where inappropriate correlations occur during the adjustment of feature distribution by learnable parameters, erroneous texture patterns may arise in the images. Therefore, we propose SCAN, expressed as:

$$\tilde{\mathbf{X}}_{i-\mathbf{SCAN}} = \gamma(\mathbf{X})\left(\frac{\mathbf{X}_i - \mu_i}{\sqrt{\sigma_i^2 + \varepsilon}}\right) + \beta(\mathbf{X}). \tag{3}$$

Here, $\gamma()$ and $\beta()$ are learnable network structures, which enable the network to adaptively generate control parameters γ and β based on information from all channels, allowing the normalization layer to correlate feature information across layers, to enhance stylization effects at the feature level. SCAN module is illustrated in Figure 2. Please note the correspondence: e.g. $\gamma(\mathbf{X}) = \mathbf{\Gamma}(\mathbf{X})_i \in \mathbb{R}$, $\beta(\mathbf{X}) = \mathbf{B}(\mathbf{X})_i \in \mathbb{R}$.

Since the statistics of features represent a particular style, instead of directly learning a complex mapping from input $\mathbf{X}$, we use channel-wise standard de-

viations $\mathbf{\Sigma}(\mathbf{X})$ and means $\mathrm{M}(\mathbf{X})$ as inputs to the learnable structures, thereby achieving style correlation adaptive normalization.

Adaptive Cartoon Texture Saliency Sampler CTSS utilizes an enhanced Sobel operator to extract edge maps from images and partition them into patches. Based on sorting edge pixel counts corresponding to patches, regions of significant texture are determined, and a local discriminator is trained to focus the generator on these regions. However, the model faces two issues. Firstly, CTSS's samples are based on batch rather than image, resulting in a biased sampling towards images with more edges in a batch. Secondly, counting the number of edge pixels may not be the optimal way to select texture salient regions, potentially leading to an increase in the number of edge textures in generated images. To address these issues, we propose a concise, learnable Ada-CTSS module based on spatial attention[34], as depicted in Figure 3.

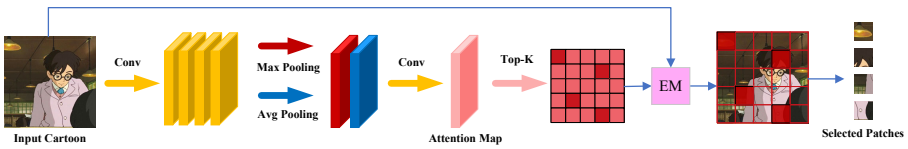

Fig. 3. Ada-CTSS Module

Attention maps are obtained by employing spatial attention at the image level. The most salient k regions in the attention maps are identified to generate a mask, which is then utilized to extract patches. These patches are fed into a local discriminator to enhance stylization effects. The advantage of this structure lies in the use of a learnable attention module, which enables the network to adaptively learn the optimal sampling method for the most significant regions and ensures that each image contributes information.

3.2 Loss Function

As illustrated in Figure 1, the loss function primarily comprises three components: adversarial loss, content loss, and anime style prior loss.

Adversarial Loss The adversarial loss is subdivided into global adversarial loss L_{adv_global} and local adversarial loss L_{adv_local}, calculated by global and local discriminator, respectively. LSGAN[23] is employed to ensure the stability of network training. For L_{adv_global}, The formula is as follows:

$$\begin{aligned} L_{adv\_global} &= L^{D}_{adv\_global} + L^{G}_{adv\_global} \\ L^{D}_{adv\_global} &= \mathbb{E}_{c_i \sim C}[(D_{img}(c_i) - 1)^2] + \mathbb{E}_{p_i \sim P}[(D_{img}(G(p_i)))^2] \\ L^{G}_{adv\_global} &= \mathbb{E}_{p_i \sim P}[(D_{img}(G(p_i)) - 1)]. \end{aligned} \tag{4}$$

For L_{adv_local}, The formula is as follows:

$$
\begin{aligned}
L_{adv\_local} &= L^{D}_{adv\_local} + L^{G}_{adv\_local} \\
L^{D}_{adv\_local} &= \mathbb{E}_{c_p \sim C_{patch}} \left[\frac{1}{K} \sum_{i=1}^{K} (D_{patch}(c_p^i) - 1)^2 \right] + \mathbb{E}_{s_p \sim S_{patch}} \left[\frac{1}{K} \sum_{i=1}^{K} (D_{patch}(s_p^i))^2 \right] \\
L^{G}_{adv\_local} &= \mathbb{E}_{s_p \sim S_{patch}} \left[\frac{1}{K} \sum_{i=1}^{K} (D_{patch}(s_p^i) - 1)^2 \right].
\end{aligned}
\tag{5}
$$

Where C and P represent the anime and the photo dataset, respectively, and S_{patch} represents the sampling set of generated images from the Ada-CTSS module, denoted as $S_{patch} = \{s_i\} = \{\text{Ada-CTSS}(G(p_i))\}$. K represents the number of patches sampled from the Ada-CTSS module.

The total adversarial loss is:

$$
L_{adv} = \omega_{global} L_{adv\_global} + \omega_{local} L_{adv\_local}. \tag{6}
$$

Content Loss Content loss ensures consistency in content between generated and input images by extracting high-level features through a pre-trained VGG model. The sparse constraint of the L1 norm is applied to local features, facilitating the presentation of a cartoon style[6]:

$$
L_{con} = \mathbb{E}_{p_i \sim P}[\|VGG_l(p_i) - VGG_l(G(p_i))\|_1]. \tag{7}
$$

Anime Style Prior Loss Vivid colors, simplified textures, and smoothed regions characterize anime images. We incorporate this prior knowledge into the description of the generator's loss function, forming the anime prior loss.

Refined Color Preservation Loss. Color preservation loss reflects the characteristics of anime in terms of color. Traditional color preservation loss[5] is computed between the input and generated images in the YUV color space. However, this method tends to make the generated images closer to the original images, reducing the stylization effect. We propose that color preservation be explicitly applied to anime images and computed between anime images and those generated from anime images as in Figure 1. This approach ensures that the output images maintain stable anime color characteristics without compromising stylization effects. We also replace the YUV color space with the Lab color space, which offers a more precise range of color expression.

$$
L_{color} = E_{c_i \sim C} \left\| L\left(G\left(c_i\right)\right) - L\left(c_i\right) \right\|_1 + \left\| a\left(G\left(c_i\right)\right) - a\left(c_i\right) \right\|_H + \left\| b\left(G(c_i)\right) - b(c_i) \right\|_H \tag{8}
$$

Grayscale Style Loss. The grayscale style loss is employed to capture the texture characteristics of anime. We convert the output and the anime image to grayscale before computing the style perceptual loss[17]. This approach aims to mitigate the negative impact of vibrant color "brushstroke" effects in artistic

styles on the anime style.

$$L_{gray} = \mathbb{E}_{p_i \sim P, c_i \sim C} \left\| Gram\left(VGG_l\left(f_{gray}\left(G(p_i)\right)\right)\right) - Gram\left(VGG_l\left(f_{gray}\left(c_i\right)\right)\right) \right\|_1, \tag{9}$$

where f_{gray} represents converting a color image to grayscale.

Region Smoothness Loss. The region smoothness loss is employed to capture the surface characteristics of anime. The output image is initially processed using selective research algorithm f_{st}[30] to obtain region structures. Subsequently, the distance between the structure and the output image is minimized using content perceptual loss[17], ensuring that the generated image exhibits the smooth surface characteristics of anime.

$$L_{region} = \left\| VGG_l(G(p_i)) - VGG_l(f_{st}(G(p_i))) \right\|_1 \tag{10}$$

The expression for the anime style prior knowledge to loss is as follows:

$$L_{anime\_prior} = \omega_{gray} L_{gray} + \omega_{color} L_{color} + \omega_{region} L_{region}. \tag{11}$$

The total loss of the network is defined as the weighted sum of all individual losses used in the training process:

$$L_{total} = L_{anime\_prior} + L_{adv\_total} + \omega_{con} L_{content} + \omega_{tv} L_{tv}. \tag{12}$$

The variable L_{tv} represents TV regularization[2], employed to suppress noise in generated images.

3.3 Style Evaluation Network

Previous methods have relied on the Fréchet Inception Distance (FID) to evaluate anime style transfer quantitatively. This method uses a pre-trained InceptionV3 model to extract features and measure the distance between the distributions of generated and anime images. However, FID has limitations when evaluating anime styles. Firstly, the Inception model is trained on ImageNet and lacks expressiveness for anime-specific features. Secondly, FID evaluates image sets and cannot assess individual images directly. Additionally, computing FID with InceptionV3 is time-consuming due to the large number of images involved and the complex network architecture.

To investigate the expressive capability of FID, we collected a dataset of photos and anime scenes with similar content and computed the cosine distances using Inceptionv3. The results, shown on the left side of Figure 4, indicate that the features utilized by FID strongly correlate with the content of images and do not effectively differentiate between the natural and anime styles. Specifically, we observed that $D3 > D2 > D1$, contrary to the expected pattern where ideal features should have small intra-class differences (D3) and large inter-class differences (D1 and D2), with $D1 \approx D2 > D3$.

To address the issues above, a compact and efficient style evaluation network has been devised to evaluate the extent of anime style in images. As shown in

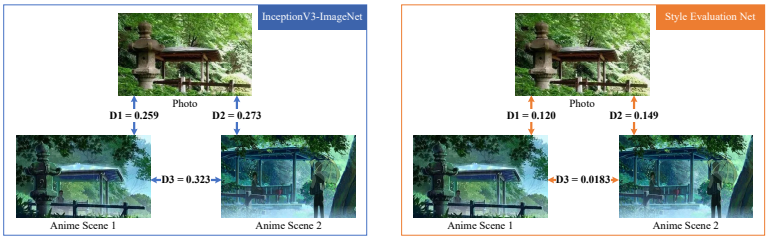

Fig. 4. Feature distances between photos and anime scenes. Left: FID. Right: Style evaluation network.

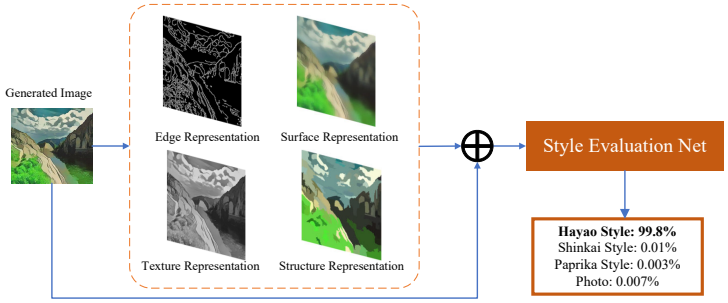

Fig. 5. The evaluation framework. Visualization representations correlated with anime style are concatenated with images as input.

Figure 5, the network is a shallow CNN that integrates representations related to anime styles, such as edges[4], textures, surfaces, and structures[33]. By concatenating these representations as inputs, the network autonomously assesses their correlation, improving the accuracy of style evaluation. When evaluating a generated image, the network assesses its accuracy across various styles, indicating a close resemblance to the corresponding anime style if the tested image is correctly classified with high confidence.

The network's performance in evaluating images is visualized in Figure 6. The network exhibits a high confidence level in classifying photos with minor color alterations but shows low confidence when more changes in color and texture occur. As alterations become more pronounced, the network exhibits low confidence in classifying images as anime. High confidence in anime discrimination is achieved when surface attributes, colors, and textures align with anime characteristics.

Our approach improves reliability compared to FID by directly learning discriminative features from anime and photo images, as in Figure 4. It also allows for direct assessment of individual images and offers a faster evaluation with a more straightforward structure than InceptionV3. The supplementary material further validates the style evaluation network's superiority over FID by con-

Fig. 6. The results of the evaluation network.

ducting photo-anime discrimination experiments. It also examines the impact of feature fusion on the evaluation network's performance.

4 Experiment

4.1 Experimental Setup

Implementation Our method utilizes TensorFlow[1] and trains the network with the Adam algorithm[18]. For photo cartoonization, the generator has a learning rate of 0.0001, the discriminator has a learning rate of 0.00005, and the batch size is 16. Pre-training involves 10 epochs with only content loss, followed by 90 epochs of formal training. The model is trained on one NVIDIA Tesla A100 GPU.

Hyper-parameters In the adversarial loss, ω_{adv_global} and ω_{adv_local} are both set to 300. Within the Ada-CTSS module, we extract $K = 4$ patches per image, each with a 64×64 pixels resolution. Regarding the anime prior loss, we assign weights $\omega_{gray} = 2.5$, $\omega_{col} = 100$, and $\omega_{region} = 1.2$. For content loss, ω_{con} is 1.5, and for TV regularization, ω_{tv} is 1.0.

Dataset The photo cartoonization model is trained on unpaired datasets. The real-world photo dataset[36] consists of 6656 training and 750 testing images. The animation images dataset consists of around 1700 images per anime, randomly cropped from "The Wind Rises" by Makoto Shinkai and "Your Name" by Hayao Miyazaki. Images from [5] are used for qualitative experiments. All images are resized to 256×256 pixels during training, with no resolution constraints during testing.

Compared Methods The proposed model will be compared to existing approaches: Neural Style Transfer (NST), image-to-image translation methods CycleGAN, and various photo cartoonization methods, including CartoonGAN, AnimeGAN, White Box, and CTSS.

Evaluation Metrics The effectiveness of style transfer will be quantitatively assessed using both Fréchet Inception Distance (FID) and our style evaluation network. The stylization effect will be evaluated by assessing the accuracy of the test set using our evaluation network.

4.2 Comparative Experiments

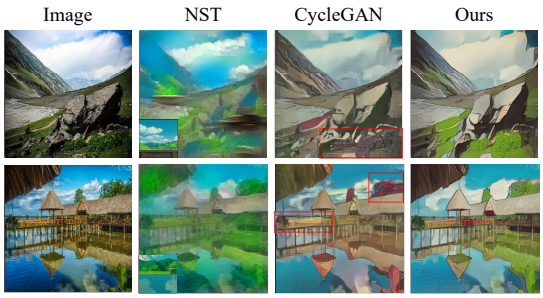

Fig. 7. Comparison with image translation methods, with red boxes indicating content mismatches.

Fig. 8. Comparison with photo cartoonization methods: red indicates texture errors, orange denotes color deviations, and blue represents weak stylization. Please zoom in for details.

Figure 7 shows a comparison with image translation methods. NST's image stylization is limited by the reference image, leading to blurred textures and content. CycleGAN lacks semantic preservation, introducing new content that is not present in the original image. Our approach, however, maintains the original content while incorporating anime-like color and texture.

Figure 8 compares our method with photo cartoonization methods. CartoonGAN shows disorganized color and texture, while AnimeGAN exhibits weak stylization with local color deviations due to limited integration of anime-specific

prior knowledge. White-box excels in surface smoothing but resembles real images with occasional flickering spots because of the lack of an effective structure to control the anime style. CTSS generates cluttered textures by focusing excessively on edge-rich regions. In contrast, our method combines SCAN and Ada-CTSS modules, leveraging anime-related prior knowledge to produce vivid colors, smoother regions, and clearer edges, thereby enhancing anime-style characteristics.

Table 1 shows the experimental results of our method on Hayao and Shinkai styles. Our approach outperforms previous photo stylization methods regarding FID and evaluation network metrics, indicating its ability to generate images closer to the anime style.

	FID to Cartoon Images↓		Evaluation Confidence↑	
	Hayao	Shinkai	Hayao	Shinkai
Photo Image	200.42	183.29	0.53	0.79
CycleGAN	141.47	139.45	84.32	71.90
CartoonGAN	167.88	157.49	86.28	78.70
AnimeGAN	165.13	146.77	88.15	76.70
White-Box	168.07	168.05	78.16	53.00
CTSS	154.99	158.66	90.01	86.02
Ours	**126.64**	**124.02**	**94.27**	**92.14**

Table 1. Quantitative results of the comparative experiment.

4.3 Ablation Study

Fig. 9. Ablation experiment of normalization methods: The red box denotes texture errors, and the blue represents weak stylization.

The results of the ablation experiments are depicted in Figures 9, 10, and 11. Figure 9 illustrates the influence of normalization layers. Removing the normalization layers will reduce the stylization effect, leading to a decrease in stylization or the generation of local texture errors. Instance Normalization (IN), due

Fig. 10. Ablation experiment of salient region extraction module.

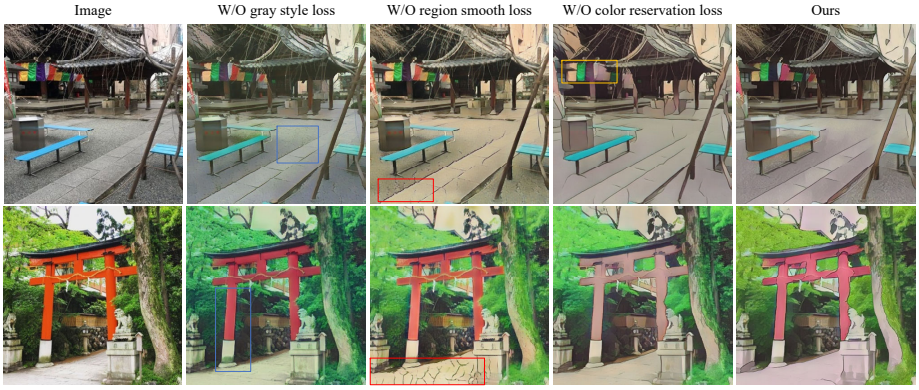

Fig. 11. Ablation experiment of loss functions. The red box represents texture errors, orange denotes color deviations, and blue indicates weak stylization.

to its lack of consideration for feature correlations, tends to produce dark images and unnatural colors. Layer Normalization (LN) fixedly associates features, which also decreases overall stylization or generates incorrect texture patterns, thereby making the effect of stylization unstable. In contrast, our approach can adaptively consider feature correlations, resulting in enhanced and stable anime stylization effects in texture and color.

Figure 10 illustrates the results of the improved Ada-CTSS module. The original CTSS module determines salient regions based on the statistical count of edge pixels, which tends to introduce disorganized textures into the images. Our approach, employing spatial attention, adaptively selects regions favorable for anime style transfer, yielding smoother surfaces that align with the distinctive style of anime. Figure 11 demonstrates the influence of the anime prior loss functions. Without the grayscale style loss, the generated images exhibit anime characteristics, albeit weakly. Omission of the region smoothness loss leads to the emergence of unnatural cluttered textures, and neglecting the color preservation loss results in erroneous coloration. By incorporating these three anime-related

losses, our method achieves results closer to the anime style. The supplementary material provides further qualitative research on the weights of the global-local discriminator and the refined color preservation loss.

Table 2 presents the quantitative results of the ablation experiments. Replacing the SCAN module with IN/LN or replacing Ada-CTSS with CTSS both decrease style evaluation metrics, indicating that our structural improvements effectively enhance the anime stylization effect. Similarly, removing the three loss functions results in varying degrees of metric decline, demonstrating our success in incorporating anime prior knowledge in terms of color, texture, and surface, thus achieving better stylization effects in transforming photos into anime style.

	FID to Cartoon Images↓		Evaluation CNN↑	
	Hayao	Shinkai	Hayao	Shinkai
W/O SCAN (With IN)	138.96	139.68	86.15	86.55
W/O SCAN (With LN)	144.14	139.14	87.22	83.49
W/O Ada-CTSS (With CTSS)	136.64	136.62	90.41	88.55
W/O L_{color}	132.95	134.59	93.74	89.34
W/O L_{gray}	139.15	135.64	89.34	85.08
W/O L_{region}	140.34	139.45	90.95	91.47
Ours	**126.64**	**124.02**	**94.27**	**92.14**

Table 2. Quantitative experimental results of the ablation study.

5 Conclusion

In this paper, we devised a deep neural network architecture tailored for photo cartoonization tasks and incorporated prior knowledge of anime style into loss functions. Our proposed SCAN module enhances inter-feature correlations to strengthen anime style effects at the feature level, while the Ada-CTSS module focuses on local image regions to enhance anime style effects at the image level. By integrating anime characteristics, we introduced loss functions in three aspects: color, texture, and surface, thereby reinforcing the anime effects in these aspects individually. Additionally, we designed a style evaluation network specifically for evaluating anime style, which is more reliable than generic metrics. Our network architecture outperforms existing photo cartoonization methods across these metrics, achieving superior anime style effects.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: {TensorFlow}: a system for {Large-Scale}

machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16). pp. 265–283 (2016) 10
2. Aly, H.A., Dubois, E.: Image up-sampling using total-variation regularization with a new observation model. IEEE Transactions on Image Processing **14**(10), 1647–1659 (2005) 8
3. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) 4
4. Canny, J.: A computational approach to edge detection. IEEE transactions on pattern analysis and machine intelligence **PAMI-8**(6), 679–698 (1986) 9
5. Chen, J., Liu, G., Chen, X.: Animegan: A novel lightweight gan for photo animation. In: Artificial Intelligence Algorithms and Applications: 11th International Symposium, ISICA 2019, Guangzhou, China, November 16–17, 2019, Revised Selected Papers 11. pp. 242–256. Springer (2020) 2, 3, 7, 10
6. Chen, Y., Lai, Y.K., Liu, Y.J.: Cartoongan: Generative adversarial networks for photo cartoonization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9465–9474 (2018) 2, 3, 7
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 2
8. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. arXiv preprint arXiv:1610.07629 (2016) 2
9. Gao, X., Zhang, Y., Tian, Y.: Learning to incorporate texture saliency adaptive attention to image cartoonization. In: ICML. vol. 2, p. 6 (2022) 2, 3
10. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016) 1, 2
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020) 1, 3
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017) 2
13. Huang, L., Qin, J., Zhou, Y., Zhu, F., Liu, L., Shao, L.: Normalization techniques in training dnns: Methodology, analysis and application. IEEE transactions on pattern analysis and machine intelligence **45**(8), 10173–10196 (2023) 4
14. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE international conference on computer vision. pp. 1501–1510 (2017) 2
15. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 2, 3
16. Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., Song, M.: Neural style transfer: A review. IEEE transactions on visualization and computer graphics **26**(11), 3365–3385 (2019) 1, 2, 4
17. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. pp. 694–711. Springer (2016) 2, 7, 8
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 10

19. Kolkin, N., Salavon, J., Shakhnarovich, G.: Style transfer by relaxed optimal transport and self-similarity. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10051–10060 (2019) 2
20. Li, C., Wand, M.: Combining markov random fields and convolutional neural networks for image synthesis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2479–2486 (2016) 2
21. Li, Y., Wang, N., Liu, J., Hou, X.: Demystifying neural style transfer. arXiv preprint arXiv:1701.01036 (2017) 2
22. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Diversified texture synthesis with feed-forward networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3920–3928 (2017) 2
23. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2794–2802 (2017) 6
24. Mechrez, R., Talmi, I., Zelnik-Manor, L.: The contextual loss for image transformation with non-aligned data. In: Proceedings of the European conference on computer vision (ECCV). pp. 768–783 (2018) 2
25. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015) 3
26. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. arXiv preprint arXiv:1505.04597 (2015) 3
27. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018) 3
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 2
29. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016) 2
30. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision **104**, 154–171 (2013) 8
31. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016) 4
32. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6924–6932 (2017) 2
33. Wang, X., Yu, J.: Learning to cartoonize using white-box cartoon representations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8090–8099 (2020) 2, 3, 9
34. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018) 2, 6
35. Zhao, Y., Ren, D., Chen, Y., Jia, W., Wang, R., Liu, X.: Cartoon image processing: a survey. International Journal of Computer Vision **130**(11), 2733–2769 (2022) 1
36. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017) 2, 3, 10