Anime Sketch Coloring with Swish-gated Residual U-net and Spectrally Normalized GAN

Gang Liu, Xin Chen and Yanzhong Hu

***Abstract*—Anime sketch coloring is to fill various colors into the black-and-white anime sketches and finally obtain the color anime images. Recently, anime sketch coloring has become a new research hotspot in the field of deep learning. In anime sketch coloring, generative adversarial networks (GANs) have been used to design appropriate coloring methods and achieved some results. However, the existing methods based on GANs generally have low-quality coloring effects, such as unreasonable color mixing, poor color gradient effect. In this paper, an efficient anime sketch coloring method using swish-gated residual U-net (SGRU) and spectrally normalized GAN (SNGAN) has been proposed to solve the above problems. The proposed method is called spectrally normalized GAN with swish-gated residual U-net (SSN-GAN). In SSN-GAN, SGRU is used as the generator. SGRU is the U-net with the proposed swish layer and swish-gated residual blocks (SGBs). In SGRU, the proposed swish layer and swish-gated residual blocks (SGBs) effectively filter the information transmitted by each level and improve the performance of the network. The perceptual loss and the per-pixel loss are used to constitute the final loss of SGRU. The discriminator of SSN-GAN uses spectral normalization as a stabilizer of training of GAN, and it is also used as the perceptual network for calculating the perceptual loss. SSN-GAN can automatically color the sketch without providing any coloring hints in advance and can be easily end-to-end trained. Experimental results show that our method performs better than other state-of-the-art coloring methods, and can obtain colorful anime images with higher visual quality.**

***Index Terms*—Anime sketch coloring, U-net, spectrally normalized GAN, swish layer, swish-gated residual blocks.**

I. Introduction

Anime sketch coloring is an important step in animation production. Its purpose is to convert the black-and-white anime sketches into the colorful anime images. At present, anime sketch coloring mainly relies on the anime painters with the professional ability. It takes a lot of time and effort to manually color the anime sketches and the coloring effect is influenced by the professional ability of the anime painters. In order to reduce the difficulties of manual coloring, it is very important to design an appropriate automatic coloring method. The automatic coloring methods can avoid the complicated work procedures generated by manual coloring, and also enable ordinary people to easily create the favorite anime images.

Recently, generative adversarial networks (GANs) [1] have been used for anime sketch coloring and some anime sketch coloring models based on GANs are proposed, such as Style2paints [2], Paintschainer [3], Auto-painter [4] and so on. These methods automatically convert the black-and-white anime sketches into the colorful anime images and the speed of coloring is faster than that of manual operation. However, GANs usually have the problems such as the difficulties of network training, unstable generating effects and non-convergence of the network. These problems lead to the poor quality of the colorful images generated by the anime sketch coloring models based on GANs, such as unreasonable color mixing, dramatic changes in colour brightness, coloring beyond the filled areas and so on. All in all, the current anime sketch coloring models based on GANs are difficult to meet the actual needs.

GANs are composed of the generator and the discriminator. For anime sketch coloring, the generator of GANs is used to generate the colorful anime images. The architecture and the loss function of the generator have direct impacts on the quality of the generated images. Therefore, designing the appropriate architecture and loss function can effectively improve the quality of the generated images. The discriminator of GANs is used to determine whether the image generated by the generator is close to the effect of manual coloring. Since the discriminator affects the training stability of GANs, the discriminator also needs further optimization to ensure the stability of the training.

In order to solve the above problems, we propose a deep learning architecture for anime sketch coloring. This new architecture is composed of the swish-gated residual U-net (SGRU) and spectrally normalized GAN (SNGAN) [5]. It is called spectrally normalized GAN with swish-gated residual U-net (SSN-GAN). The black-and-white anime sketches are input into SSN-GAN, and then the colorful anime images are output. The discriminator of SNGAN uses the spectral normalization to enhance the stability of the network training. SGRU is an improvement to the U-net [6] and is used to be the generator of SNGAN. SGRU contains the proposed swish-gated residual blocks (SGBs) and swish layers which are inspired by the swish activation function [7]. SGBs and the swish layers can filter the feature information transmitted in SGRU and improve the learning ability of the network. SGRU uses the perceptual loss [8] and the per-pixel loss as the training loss of the generator. The use of the perceptual loss can help the network to color the black-and-white animation sketches with more smooth and saturated colors and solve the coloring problems caused by the automatic coloring models based on GANs. The discriminator of SSN-GAN is used as the perceptual network to extract the perceptual features to calculate the perceptual loss. Experimental verifications are conducted on the Danbooru2017 dataset [9]. Experimental results show that SSN-GAN is significantly better than other state-of-the-art methods and the quality of the color images generated by SSN-GAN is close to or reaches the level of manual coloring.

The work described in this paper was support by National Natural Science Foundation of China Foundation No.61300127. Any conclusions or recommendations stated here are those of the authors and do not necessarily reflect official positions of NSFC.

Gang Liu, Xin Chen and Yanzhong Hu are with the School of Computer Science, Hubei University of Technology, Wuhan, 430068 China e-mail: lg0061408@126.com, ghj9527@163.com and 15738443@qq.com.

The remainder of this paper is organized as follows. The related work is described in Section 2. The architecture of SSN-GAN is presented in Section 3. Experimental datasets and results are reported in Section 4. Finally, some conclusions are given in Section 5.

II. Related work

Recently, generative adversarial networks (GANs) attracts increasing attention in the field of deep learning technology (DL) [10]. A GAN often comprises a generator and a discriminator that learn simultaneously. The generator tries to capture the potential distribution of real samples, and generates new data samples. The discriminator is often a binary classifier, discriminating real samples from the generated samples as accurately as possible. These two networks are optimized using a min-max game: the generator attempts to deceive the discriminator by generating data indistinguishable from the real data, while the discriminator attempts not to be deceived by the generator by finding the best discrimination between real and generated data.

Many researchers have proposed many GANs variants for the improvement of GANs. Arjovsky [11] proposed Wasserstein GAN (WGAN) by using the Earth-Mover distance to replace the Jensen-Shannon divergence for evaluating the distribution distance between the real data and the generated data. They used a critic function that builds on Lipschitz constraint to represent the discriminator. WGAN makes significant progress towards stable training of GANs, but can still generate low-quality samples or fail to converge in some settings. Mirza [12] introduced the conditional version of generative adversarial nets. It extends the GAN framework to the conditional setting by making both the generator and the discriminator networks class-conditional. Conditional GANs have the advantage of being able to provide better representations for multi-modal data generation. Chen [13] described InfoGAN, an information-theoretic extension to the Generative Adversarial Network that is able to learn disentangled representations in a completely unsupervised manner. InfoGAN is a generative adversarial network that also maximizes the mutual information between a small subset of the latent variables and the observation. They derive a lower bound of the mutual information objective that can be optimized efficiently. Qi [14] presented the Lipschitz regularization theory and algorithms for a novel Loss-Sensitive Generative Adversarial Network (LS-GAN). Specifically, it trains a loss function to distinguish between real and fake samples by designated margins, while learning a generator alternately to produce realistic samples by minimizing their losses. The LS-GAN further regularizes its loss function with a Lipschitz regularity condition on the density of real data, yielding a regularized model that can better generalize to produce new data from a reasonable number of training examples than the classic GAN. They further presented a Generalized LS-GAN (GLS-GAN) and showed it contains a large family of regularized GAN models, including both LS-GAN and Wasserstein GAN, as its special cases. Berthelot [15] proposed BEGAN, which is a new equilibrium enforcing method paired with a loss derived from the Wasserstein distance for training auto-encoder based GANs. This method balances the generator and discriminator during training. Additionally, it provides a new approximate convergence measure, fast and stable training and high visual quality. Odena [16] proposed auxiliary classifier GAN (AC-GAN) for semi-supervised synthesis. Their objective function consists of two parts: the log-likelihood of the correct data source and that of the correct class. The key of AC-GAN is that it can incorporate label information into the generator and adjust the objective function for the discriminator. In consequence, the generation and discrimination abilities of GAN are improved. Yu [17] proposed SeqGAN to generate data sequences. Modeling the data generator as a stochastic policy in reinforcement learning (RL), SeqGAN bypasses the generator differentiation problem by directly performing gradient policy update. The RL reward signal comes from the GAN discriminator judged on a complete sequence, and is passed back to the intermediate state-action steps using Monte Carlo search. Extensive experiments on synthetic data and real-world tasks demonstrate significant improvements over strong baselines.

The work of coloring black-and-white anime sketches is similar to neural style transfer (NST) [18], [19], [20]. However, unlike NST, anime sketch coloring does not require the style reference image in advance so that it is more challenging. Recently, the automatic coloring models based on deep learning mainly uses the architecture of GANs. Sangkloy [21] proposed a deep adversarial image synthesis architecture that is conditioned on sketched boundaries and sparse color strokes to generate realistic cars, bedrooms, or faces. They demonstrated a sketch based image synthesis system which allows users to scribble over the sketch to indicate preferred color for objects. The network can then generate convincing images that satisfy both the color and the sketch constraints of user. The network is feed-forward which allows users to see the effect of their edits in real time. The Pix2Pix method proposed in the literature [22] used the conditional GAN (cGAN) to achieve general image-to-image transfer. They demonstrated that this approach is effective at synthesizing photos from label maps, reconstructing objects from edge maps, and colorizing images, among other tasks. Zhang [2] integrates residual U-net to apply the style to the black-and-white sketch with auxiliary classifier generative adversarial network (AC-GAN). In fact, Style2paints is the style transfer model and it needs to provide the color reference image in advance when converting the animation sketch into the color image. Liu [4] proposed a model called auto-painter which can automatically generate compatible colors given a sketch. Wasserstein distance is used in training cGAN to overcome model collapse and enable the model converged much better. The new model is not only capable of painting hand-draw sketch with compatible colors, but also allowing users to indicate preferred colors. Experimental results on different sketch datasets show that the auto-painter performs better than other existing image-to-image methods.

It can be seen that much research has been done in GANs to enhance the performance of GANs and GANs has also achieved outstanding results in anime sketch coloring.

III. Our method

A. Architecture of generator

1) Swish layer and swish-gated residual blocks: This paper proposes a novel type of residual blocks, which is

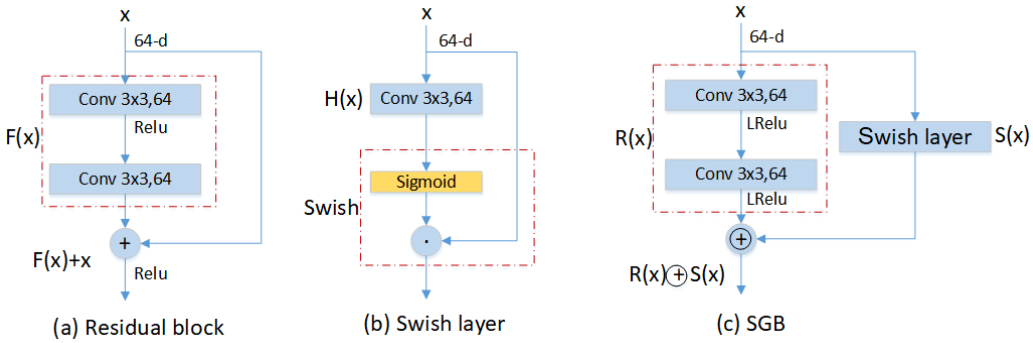

Fig. 1. The structures of the residual block, swish layer and SGB. Each blue box represents the corresponding data operation in the network. The arrows indicate data flow.

an improvement on the residual blocks in Resnet [23]. The novel residual blocks are called the swish-gated blocks (SGBs). SGBs are composed of the proposed swish layers and the residuals. The proposed swish layer contains the convolutional layer and the swish activation function [7]. The structure of the residual block, the swish layer and SGB are shown in Fig.1. In Fig.1, x represents the input data and $F(x)$ represents the residual. $F(x) + x$ is the output of the residual block. The $H(x)$ represents the output of the convolution layer in the swish layer. The "$\cdot$" denotes the element-wise multiplication and "$+$" means the element-wise addition. The $R(x)$ represents the output of the convolution layers using the nonlinearity function LReLU [24] in SGB. The $S(x)$ represents the output of the swish layer. The "$\oplus$" indicates the concatenation operation of the feature maps. The $R(x) \oplus S(x)$ represents the output of SGB.

As shown in Fig.1, x are added directly to the residuals in the residual block without any processing. In order to process x, the swish layer is proposed to control the propagation of x. In fact, the swish layer can be considered as the swish-inspired adaptive gating mechanism. Compared to the residual block, SGB uses the swish layer to filter x. The purpose of SGB is to control the data sent to the higher layers through the shortcut connection [23]. Formally, the swish layer can be defined as:

$$S(x) = x \cdot \sigma(H(x)) \tag{1}$$

where x and $S(x)$ are the input and output of the swish layer. The $H(x)$ represents the output of the convolution layer in the swish layer. The "σ" denotes the sigmoid function. The "$\cdot$" denotes the element-wise multiplication.

SGBs combine the residual blocks and the swish layers. In SGBs, the swish layers are used as the learnable gating mechanism which can filter the transmitted information. The x is filtered by the swish layer and the filtered information is concatenated to $R(x)$ to obtain the output of SGBs. SGB can be expressed as:

$$y = R(x) \oplus S(x) \tag{2}$$

where $R(x)$ can be considered as the residuals of SGB and $S(x)$ is the output of the swish layer in SGB. The "$\oplus$" denotes $R(x)$ and $S(x)$ are concatenated together.

2) Architecture of generator: In this paper we propose spectrally normalized GAN with swish-gated residual U-net (SSN-GAN) which is composed of the swish-gated residual U-net (SGRU) and spectrally normalized GAN (SNGAN) for coloring the black-and-white anime sketches into the colorful anime images. The generator of SSN-GAN is SGRU, which is the U-net [6] with the swish-gated residual blocks. The structure of SGRU is shown in Fig.2. The network has 6 different resolution levels. With the increase of the serial number of the level, the resolutions of the feature maps decrease gradually.

In the vertical direction of SGRU, the swish layers are embedded between two adjacent levels. The swish layers and the stacked convolutional layers in the left (right) branches constitute SGBs. The structures inside the dashed boxes in Fig.2 are SGBs. There are 10 SGBs in SGRU. In the horizontal direction of SGRU, the swish layers are embedded in each skip connection between the left and right branches to filter the information passed from the encoding path to the decoding path. The swish layers in the skip connections can improve the performance of the network.

Except for the last convolutional layer used in the output of SGRU, all other convolutional layers use layer normalization [25] and nonlinearity LReLU. In SGRU, the input of SGBs in the ith level is the output of the 1×1 convolutions in the $(i-1)th$ level and the output of SGBs in the ith level is concatenated to the input of the 1×1 convolutions in the $(i+1)th$ level. From the $1st$ level to the $6th$ level, the number of convolution kernels in each convolutional layer of SGB in ith level is the same as the number of the 1×1 convolutional kernels in the $(i-1)th$ level. Like the U-net, the upsampling still uses the deconvolution method.

The last convolution layer in the first level of the right branch converts the feature maps into the color image. It does not use the normalization operations and the activation functions. SGRU contains 6 levels, the number of the convolution kernels in each convolution layer from the first level to the last level is 96, 192, 288, 384, 480, 512, respectively.

SGBs and the swish layers used in SGRU have the following obvious advantages: 1) the swish layers as the learnable gating mechanism in SGBs or SGRU can be used to filter the feature information transmitted from the lower layer to the higher layer or between the same layer. The swish layer can intelligently filter the transmitted information to enhance the important information in the feature maps; 2) SGB is an improvement for the general residual block, which is more able to improve the learning ability of the network than the general residual block; 3) the use of SGBs can effectively

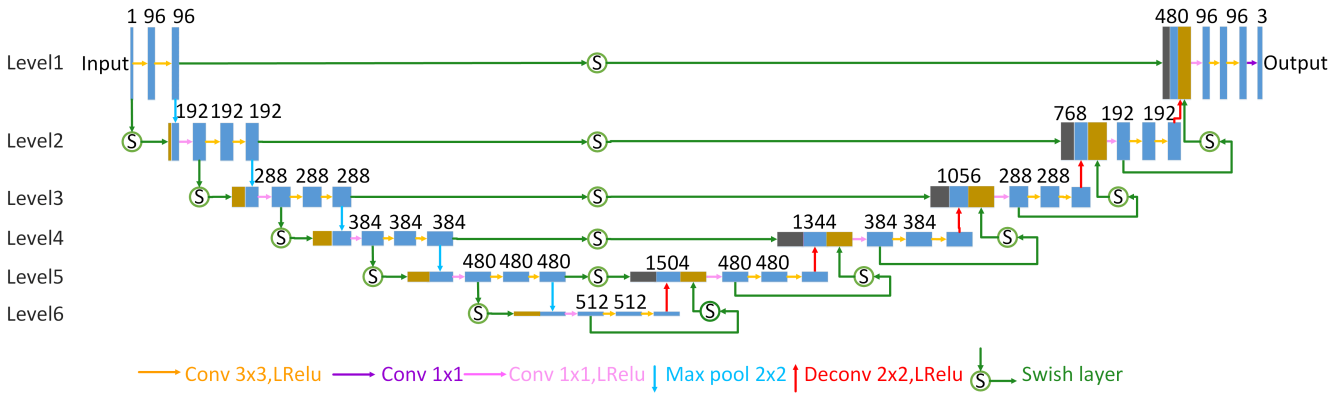

Fig. 2. The architecture of SGRU. Each blue box represents a multi-channel feature map. Each brown box represents a multi-channel feature map output by the swish layer. Each black box represents the copies of the feature maps of the left branch. The number on the box indicates the number of channels. The arrows denote the different operations. The S indicates the swish layer. From the $1st$ level to the $6th$ level, the resolution of the feature maps is halved in turn.

prevent the problem of gradient disappearance during deep network training. In addition, SGBs can effectively improve network performance and enable our model to generate the images with higher visual quality.

B. Architecture of discriminator

Generally, the role of the discriminator is to distinguish between the generated images and the ground-truth images. In this paper, the discriminator is a convolutional neural network. The architecture of the discriminator is shown in Fig.3. The discriminator is mainly composed of 5 convolutional layers and 1 fully connected layer. It should be noted that there is only one neuron in the fully connected layer. The spectral normalization operations are used for the weight parameters of each convolution layer and the fully connected layer. The instance normalization [26] operations are performed on the feature maps output by the convolutional layers using the activation function LReLU. The specific structural details of the discriminator are shown in Table I.

TABLE I
THE SPECIFIC STRUCTURAL DETAILS OF THE DISCRIMINATOR.

layer	output size	filter size	stride
input image	256x256x3	-	-
Conv1	128x128x64	5x5	2
Conv2	64x64x128	5x5	2
Conv3	32x32x256	5x5	2
Conv4	32x32x512	3x3	1
Max_pool1	16x16x512	2x2	2
Conv5	16x16x1024	3x3	1
Max_pool2	8x8x1024	2x2	2

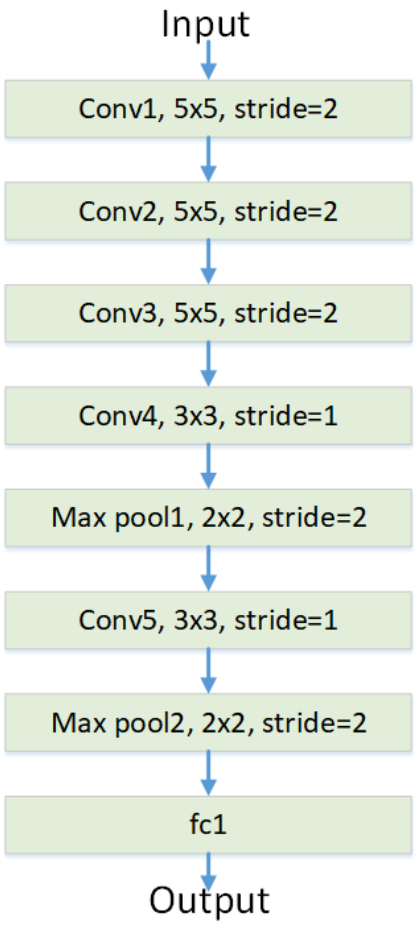

Fig. 3. The architecture of the discriminator.

In our work, the discriminator has two tasks: 1) it is used to discriminate between the generated images and the ground-truth images like the general discriminator; 2) it is used as the perceptual network to extract the perceptual features of both the generated images and the ground-truth images for calculating the perceptual loss.

In SSN-GAN, the purpose of spectral normalization is to enhance the stability of network training. Spectral normalization controls the Lipschitz constant of the discriminator function by literally constraining the spectral norm of each layer. It normalizes the spectral norm of the weight matrix W of the discriminator so that it satisfies the Lipschitz constraint $\sigma(W) = 1$, where $\sigma(W)$ is the spectral norm of the matrix W which is equivalent to the largest singular value of W.

C. Training

SSN-GAN is trained end-to-end with the supervised method. SSN-GAN uses the training pair $D = \{S, G\}$ as the input training dataset. A black-and-white anime sketch S is used as the input and the corresponding reference color image G as the output label. For the coloring tasks, simply

comparing the pixel colors of the generated image and the reference color image can severely penalize the quality of the output image because the transformation from the anime sketch to the color image is not a one-to-one transformation. In fact, anime sketch coloring is a one-to-many transformation. For example, the hair color in the reference color image is red, but the hair color in the output color image may be black or silver. These hair colors are reasonable for anime. But the color of two kinds of hairs would have huge differences as measured by the per-pixel loss and the per-pixel losses do not capture perceptual differences between the output images and the reference color images. In this situation, the perceptual features of the color of these two kinds of hairs may be similar. Therefore, in addition to the per-pixel loss, the perceptual loss is also employed in the generator of SSN-GAN to measure the high-level perceptual differences between the output images and the reference color images.

In our work, the discriminator of SSN-GAN is used as the visual perception network to extract the perceptual features of the images output by the generator SGRU and the reference color images. The loss function of the generator includes the per-pixel loss and the perceptual loss between the output images and the reference color images. Let φ be the discriminator. The φ_l represents a collection of layers in the network φ. The total loss function of the generator SGRU can be represented as:

$$L_g = \sum_l \lambda_l ||\varphi_l(T) - \varphi_l(G)||_1 \tag{3}$$

where T and G represents the output color image and the reference color image, respectively. $\varphi_l(T)$ and $\varphi_l(C)$ represent the feature maps output from the lth layer in the discriminator when T and G are respectively input to the discriminator. $l \in \{0, 1, 2, 3, 4, 5\}$ and φ_1 to φ_5 represent the convolutional layers (conv1, conv2, conv3, conv4 and conv5) that are selected to calculate the perceptual loss in the discriminator. When $l = 0$, $\varphi_0(T)$ and $\varphi_0(G)$ represent the original input T and G. The hyperparameter λ_l is used to balance the contribution of the lth layer to the total loss L_g and $\lambda_l = \{0.88, 0.79, 0.63, 0.51, 0.39, 1.07\}$. Adam optimizer [27] is applied to minimize the total loss.

The discriminator is essentially a binary classifier and its role is to distinguish between the images output by the generator SGRU and the label images. The loss function of the discriminator used is expressed as follows:

$$L_d = -E[\log(\sigma(D(G)) + \log(1 - \sigma(D(T)))] \tag{4}$$

where $D(T)$ and $D(G)$ represent the output of the discriminator with T and G as the input, respectively. The E represents mathematical expectation. The $\sigma(.)$ represents the sigmoid function.

IV. Experiments

A. Datasets and evaluation metrics

We performed a lot of experiments on the Danbooru2017 dataset [9] to verify the performance of the proposed network. The Danbooru2017 dataset is a large-scale crowdsourced and tagged anime illustration dataset. In Danbooru2017 dataset, we selected $18,560$ color anime images for training. These color images and their corresponding black-and-white sketches obtained from preprocessing (resize, truncate to squared images and extract sketches) are used as the training dataset. In our experiments, the resolution of all training images is adjusted to 256×256.

It is well-known that evaluating the quality of a generated image is an open and difficult problem [28]. For the task of coloring, when the same black-and-white anime sketch is used as the input to get the corresponding color image, the different networks may color different colors at the same position. In addition to the difference in color, the color anime images generated by different networks may also have great differences in the content (texture, brightness, shadow, etc.) and the image visual quality. Therefore, we used several standard quantitative measures of the image visual quality to evaluate and compare the proposed methods and other existing methods. The standard quantitative evaluation metrics used in our experiments included peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [29], feature similarity (FSIM) [30] and FSIMc (FSIM incorporates image chrominance information) [30]. In order to evaluate the visual quality of the images generated by SSN-GAN and compare to other networks that use the U-net as the generator, Frechet inception distance (FID) [31] is also used as the evaluation protocol to quantify the quality of our results.

B. Results

The qualitative results of the three coloring models on the Danbooru2017 dataset are shown in Fig. 4. It can be clearly seen that compared with SNGAN-Unet the generator of which is U-net, the color images generated by SSN-GAN are more vivid and saturated. Specifically, the color gradient is smoother and the shadow distribution is reasonable. In addition, when the discriminator is not used as the perceptual network to calculate the perceptual loss, the model is called SSN-GAN without perceptual loss. To verify the validity of the perceptual loss, we compared SSN-GAN and SSN-GAN without perceptual loss. As can be clearly seen from Fig. 4, the color images generated by SSN-GAN without perceptual loss are not rich enough and have fewer changes in color gradation. Moreover, the color saturation is lower and there is no obvious boundary between foreground and background colors. Therefore, the perceptual loss is important to improve the effect of coloring. Compared with the color images generated by other two methods, the images generated by SSN-GAN are richer in texture details and smoother in color transition.

To further investigate our approach, we use the quantitative criteria to evaluate the quality of the generated images. We used Frechet inception distance (FID) to evaluate three coloring models. FID compares the inception activation value between the real image and the generated image. Since automatic coloring is a one-to-many conversion, FID is mainly used in this paper to evaluate the visual quality of the generated images. Table II reports FID of the different networks. The best results are shown in **boldface**. SSN-GAN without perceptual loss is marked as SSN-GAN-wpl in Table II. As can be seen from Table II, our method outperforms the other methods and can generate the color images with higher visual quality.

Fig. 4. Qualitative comparison of the generated images on the Danbooru2017 dataset.

TABLE II
COMPARISON OF FID OF THE DIFFERENT NETWORKS ON THE DANBOORU2017 DATASET.

	SSN-GAN-wpl	SNGAN-Unet	SSN-GAN
FID	119.59	112.85	**106.55**

In order to compare the performance of 3 methods, PSNR, SSIM, FSIM and FSIMc are conducted. Table III shows the quantitative results of the three methods. The best results are shown in **boldface**. SSN-GAN has achieved the best results on all metrics. Moreover, the results of SSN-GAN without perceptual loss are lower than all methods. It indicates that the perceptual loss plays an important role in anime sketch coloring. Our proposed generator SGRU has better performance than the U-net. Therefore, the quality of the color anime images generated by SSN-GAN is better than SNGAN-Unet and SSN-GAN without perceptual loss.

TABLE III
QUANTITATIVE COMPARISON OF THREE COLORING METHODS ON THE DANBOORU2017 DATASET.

	PSNR	SSIM	FSIM	FSIMc
SSN-GAN-wpl	16.616	0.831499	0.845807	0.826905
SNGAN-Unet	17.077	0.844272	0.853128	0.834270
SSN-GAN	**17.485**	**0.849160**	**0.856360**	**0.837966**

The comparison results of SSN-GAN, Style2paints [2] and Paintschainer [3] are shown in Fig. 5. As shown in Fig. 5, the colorful images generated by SSN-GAN have higher visual quality than other two state-of-the-art coloring models. Compared with Paintschainer and Style2paints, SSN-GAN is more adept at processing the texture details which makes the coloring smoother and more natural. SSN-GAN can effectively avoid the problems existing in Style2paints and Paintschainer, such as the color clutter caused by irregular

Fig. 5. Qualitative comparison of the state-of-the-art coloring models.

mixing of multiple colors, the sharp changes in colors brightness, coloring beyond the filled areas and so on. Therefore, the coloring effect of SSN-GAN is superior to the state-of-the-art coloring methods.

Table IV shows the quantitative comparison results of SSN-GAN, Style2paints and Paintschainer. The best results are shown in **boldface**. In Table IV, SSN-GAN outperforms Style2paints and Paintschainer on all metrics. It indicates that the visual quality of the color anime images generated by SSN-GAN is better than Paintschainer and Style2paints. Among the three methods, SSN-GAN can get the best coloring effect.

TABLE IV
QUANTITATIVE COMPARISON OF SSN-GAN, STYLE2PAINTS AND PAINTSCHAINER ON THE DANBOORU2017 DATASET.

	PSNR	SSIM	FSIM	FSIMc
Paintschainer	14.516	0.784078	0.799761	0.785066
Style2paints	14.802	0.785283	0.791618	0.772001
SSN-GAN	**19.127**	**0.844990**	**0.873481**	**0.857569**

V. CONCLUSIONS

This paper presents spectrally normalized GAN with swish-gated residual U-net (SSN-GAN) which is composed of the swish-gated residual U-net (SGRU) and spectrally normalized GAN (SNGAN) for automatically coloring the black-and-white anime sketches into the color anime images. In SSN-GAN, the discriminator uses the spectral normalization to make the training more stable, and the generator SGRU uses the swish-gated residual blocks (SGBs) and

the swish layers to improve the quality of the generated images. SSN-GAN can be easily trained end-to-end with the perceptual loss and the per-pixel loss while the discriminator is used as the perceptual network. On the Danbooru2017 dataset, the experimental results show that the proposed method has obvious advantages over other state-of-the-art automatic coloring models in the complicated sketch coloring tasks. Compared with other state-of-the-art methods, the output images colored by SSN-GAN have the higher visual quality. Future work includes implementing the conditional black-and-white anime sketch coloring and generating larger resolution color images.

ACKNOWLEDGMENT

The work described in this paper was support by National Natural Science Foundation of China Foundation No.61300127. Any conclusions or recommendations stated here are those of the authors and do not necessarily reflect official positions of NSFC.

REFERENCES

[1] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA Journal of Automatica Sinica.*, vol. 4, no. 4, pp. 588–598, 2017.

[2] L. Zhang, Y. Ji, and X. Lin, "Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier GAN," *CoRR*, vol. abs/1706.03319, 2017. [Online]. Available: http://arxiv.org/abs/1706.03319

[3] T. Yonetsuji, "Paintschainer," https://paintschainer.preferred.tech/index_en.html.

[4] Y. Liu, Z. Qin, T. Wan, and Z. Luo, "Auto-painter: Cartoon image generation from sketch by using conditional wasserstein generative adversarial networks," *Neurocomputing.*, vol. 311, no. 15 October 2018, pp. 78–87, 2018.

[5] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *CoRR*, vol. abs/1802.05957, 2018. [Online]. Available: https://arxiv.org/abs/1802.05957

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'2015)*, Munich, Germany, Oct. 2015, pp. 234–241.

[7] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *CoRR*, vol. abs/1710.05941, 2017. [Online]. Available: http://arxiv.org/abs/1710.05941

[8] J. Johnson, A. Alahi, and F.-F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. 14th European Conference on Computer Vision (ECCV'2016)*, Amsterdam, Netherlands, Oct. 2016, pp. 694–711.

[9] Anonymous, the Danbooru community, G. Branwen, and A. Gokaslan, "Danbooru2017: A large-scale crowdsourced and tagged anime illustration dataset," https://www.gwern.net/Danbooru2017.

[10] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, no. January 01, 2015, pp. 85–117, 2015.

[11] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th International Conference on Machine Learning, ICML 2017*, Sydney, NSW, Australia, Aug. 2017, pp. 298–321.

[12] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014. [Online]. Available: http://arxiv.org/abs/1411.1784

[13] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. 30th Annual Conference on Neural Information Processing Systems, NIPS 2016*, Barcelona, Spain, Dec. 2016, pp. 2180–2188.

[14] G. Qi, "Loss-sensitive generative adversarial networks on lipschitz densities," *CoRR*, vol. abs/1701.06264, 2017. [Online]. Available: http://arxiv.org/abs/1701.06264

[15] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: boundary equilibrium generative adversarial networks," *CoRR*, vol. abs/1703.10717, 2017. [Online]. Available: http://arxiv.org/abs/1703.10717

[16] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proc. 34th International Conference on Machine Learning, ICML 2017*, Sydney, NSW, Australia, Aug. 2017, pp. 4043–4055.

[17] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proc. 31st AAAI Conference on Artificial Intelligence, AAAI 2017*, San Francisco, CA, United states, Feb. 2017, pp. 2852–2858.

[18] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stylebank: An explicit representation for neural image style transfer," in *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, United states, Jul. 2017, pp. 2770–2779.

[19] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *CoRR*, vol. abs/1508.06576, 2015. [Online]. Available: http://arxiv.org/abs/1508.06576

[20] ——, "Image style transfer using convolutional neural networks," in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, Las Vegas, NV, United states, Jun. 2016, pp. 2414–2423.

[21] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, United states, Jul. 2017, pp. 6836–6845.

[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, United states, Jul. 2017, pp. 6836–6845.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, Las Vegas, NV, United states, 2016, pp. 770–778.

[24] ——, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. 2015 IEEE International Conference on Computer Vision (ICCV'2015)*, Santiago, Chile, 2015, pp. 1026–1034.

[25] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016. [Online]. Available: https://arxiv.org/abs/1607.06450v1

[26] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016. [Online]. Available: http://arxiv.org/abs/1607.08022

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. the 3rd International Conference for Learning Representations (ICLR'2015)*, San Diego, CA, United states, May 2015, pp. 1–15.

[28] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," *CoRR*, vol. abs/1603.08511, 2016. [Online]. Available: http://arxiv.org/abs/1603.08511

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing.*, vol. 13, no. 4, pp. 600–612, 2004.

[30] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "Fsim: A feature similarity index for image quality assessment," *IEEE Transactions on Image Processing.*, vol. 20, no. 8, pp. 2378–2386, 2011.

[31] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a nash equilibrium," *CoRR*, vol. abs/1706.08500, 2017. [Online]. Available: http://arxiv.org/abs/1706.08500