AnimeGAN: a novel lightweight GAN for photo animation

Jie Chen[1], Gang Liu[2], and Xin Chen[2]

[1] School of civil engineering, Wuhan university, Wuhan 430072, China
[2] School of Computer Science, Hubei University of Technology, Wuhan 430072, China
cjjjack@163.com, lg0061408@126.com, ghj9527@163.com

Abstract. In this paper, a novel approach for transforming photos of real-world scenes into anime style images is proposed, which is a meaningful and challenging task in computer vision and artistic style transfer. The approach we proposed combines neural style transfer and generative adversarial networks (GANs) to achieve this task. For this task, some existing methods have not achieved satisfactory animation results. The existing methods usually have some problems, among which significant problems mainly include: 1) the generated images have no obvious animated style textures; 2) the generated images lose the content of the original images; 3) the parameters of the network require the large memory capacity. In this paper, we propose a novel lightweight generative adversarial network, called AnimeGAN, to achieve fast animation style transfer. In addition, we further propose three novel loss functions to make the generated images have better animation visual effects. These loss function are grayscale style loss, grayscale adversarial loss and color reconstruction loss. The proposed AnimeGAN can be easily end-to-end trained with unpaired training data. The parameters of AnimeGAN require the lower memory capacity. Experimental results show that our method can rapidly transform real-world photos into high-quality anime images and outperforms state-of-the-art methods.

Keywords: Generative Adversarial Networks · neural style transfer · computer vision.

1 Introduction

Anime is a common artistic form in our daily life. This artistic form is widely used in several fields including advertising, film and children's education. Currently, the production of animation mainly relies on manual implementation. However, manually creating anime is very laborious and involves substantial artistic skills. For animation artists, creating high-quality anime works requires careful consideration of lines, textures, colors and shadows, which means that it is difficult and time consuming to create the works. Therefore, the automatic techniques that can automatically transform real-world photos to high-quality animation style images are very valuable and necessary. It not only allows the

artists to focus on more creative work and save time, but also makes it easier for ordinary people to implement their own animation.

Currently, image-to-image translation based on deep learning [25] has achieved great results. Recently, learning-based style transfer methods [5–8] have become the common image-to-image translation methods. These methods can learn the style of the reference image (style image) and apply the learned style to the input image (content image) to generate a new image which fuses the content of the content image and the style of the style image. These methods primarily uses the correlations between deep features and an optimization-based method to encode the visual style of an image.

Generative adversarial networks (GANs) [9, 28] have been applied for style transfer and achieved great results. Many researchers have proposed many GAN-based style transfer methods. CycleGAN [30] implementd collection style transfer, object transfiguration, season transfer, photo enhancement, etc. CartoonGAN [3] used the GAN framework to achieve the transformation of real photos into cartoon images.

Although these methods have achieved some success in anime style transfer, they still have many obvious problems. These important problems mainly include: 1) the generated images have no obvious animated style textures; 2) the generated images lose the content of the original photos; 3) a large number of the parameters of the network require more memory capacity.

In order to solve the above problems, we propose a novel lightweight GAN, called AnimeGAN, which rapidly transforms real-world photos into high-quality anime images. The proposed AnimeGAN is a lightweight generative adversarial model with fewer network parameters and introduces Gram matrix [16] to get more vivid style images. Our method takes a set of photos and a set of anime images for training. To produce high quality results while making the training data easy to obtain, we use the unpaired data for training, which means that the photos and anime images in the training set are not related in content.

To further improve the animation visual effects of the generated images, we propose three new simple yet efficient loss functions. The proposed loss functions are the grayscale style loss, the color reconstruction loss, and the grayscale adversarial loss. In the generative network, the grayscale style loss and the color reconstruction loss make the generated image have more obvious anime style and preserve the color of the photos. The grayscale adversarial loss in the discriminator network makes the generated image have vivid color. In the discriminator network, we also used the edge-promoting adversarial loss proposed by the literature [3] for preserving clear edges.

In addition, in order to make the generated images have the content of the original photos, we introduce the pre-trained VGG19 [26] as the perceptual network to obtain the L1 loss of the deep perceptual features of the generated images and original photos. Before AnimeGAN starts training, we perform an initialization training on the generator to make the training of AnimeGAN easier and more stable. A large number of experimental results show that our AnimeGAN

can quickly generate higher-quality anime style images and outperform state-of-the-art methods.

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

2 Related Work

Neural Style Transfer. A variety of Neural style transfer (NST) algorithms [5–8, 13] have been developed to synthesize a novel image by combining the content of one image with the style of another image (typically a painting) based on matching the Gram matrix statistics of deep features extracted from a pre-trained convolutional network. Gatys et al. [5–8] have achieved many impressive results in a series of style transfer tasks, but their network models often have a large number of parameters, which makes the training process quite time-consuming. Furthermore, their methods focus on the style transfer of the paintings. For other style transfer tasks, such as animation style transfer, photography style transfer, etc, they still need to be improved. Huang et al. [11] proposed a simple but effective method, called AdaIN, which aligns the mean and variance of the content image features with those of the style image features. The method does not use complex Gram matrix. Li et al. [17] proposed a novel approach called Universal Style Transfer (UST), which used Whitening and Coloring transform (WCT) to directly match the features covariance in the content image to those in the given style image. In order to achieve photorealistic image stylization, Li et al. [18] presented a novel fast photography style transfer method, which consists of a stylization step and a photorealistic smoothing step. The stylization step transfers the style of the reference photo to the content photo, the smoothing step ensures spatially consistent stylizations. Each of the steps has a closed-form solution and can be computed efficiently. But their method requires additional semantic label maps as supervision to help style transfer between corresponding semantic regions. In a word, these existing neural style transfer methods are usually only suitable for specific style transfer task. When they are used for animation style transfer, they tend to obtain unsatisfactory results.

Image-to-Image Translation with GANs. Image-to-image (I2I) translation, which is a research hotspot in the field of computer vision, refers to the task of mapping an image from a source domain to a target domain, e.g. semantic maps to real images, grayscale images to color images, low-resolution images to high-resolution images, and so on. Generative adversarial networks (GANs) have become a research focus of artificial intelligence, which is inspired by two-player zero-sum game. A GAN often comprises a generator and a discriminator that learn simultaneously and these two networks are optimized using a min-max game. Recently, the image-to-image translation methods based on GANs have achieved many results. Isola et al. [12] proposed the pix2pix to use the conditional GAN (cGAN) and U-Net neural network 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. Wang et al. [29] proposed the pix2pixHD approach on the basis of the pix2pix approach for synthesizing high-resolution photo-realistic images from semantic label maps using conditional generative adversarial networks. CycleGAN [30] is an approach for learning to translate an image from a source domain X to a target domain Y in the absence of paired examples. Almahairi et al. [2] proposed Augmented CycleGAN, which learns many-to-many mappings between domains in an unsupervised way. Their model can learn mappings which produce a diverse set of outputs for each input and can learn mappings across substantially different domains.

It can be seen that the style transfer methods based on GANs or convolutional neural networks (CNNs) [15] have been extensively studied and many excellent results have been achieved. In recent years, animation style transfer has become a new research direction. Chen et al. in the literature [3] presented style transfer method based on Generative Adversarial Networks that seems to work really well in terms of photo cartoonization problem. Maciej et al. in the literature [23] proposed a solution to transform a video into a comics. They build two stages to transform input video into a comics. Their main contribution is to propose a keyframes extraction algorithm that selects a subset of frames from the video to provide the most comprehensive video context. Their network structure is the same as the literature [3] and some other training strategies are used in their method.

As a comparison, we propose a more lightweight AnimeGAN to learn the mapping between photo and anime manifolds using unpaired training data. We also proposed three dedicated loss functions to synthesize high quality anime images.

3 Our method

3.1 AnimeGAN architecture

In this paper, we present a simpler and more efficient generative adversarial network called AnimeGAN. AnimeGAN consists of two convolution neural networks: One is the generator G which is used to transform the photos of real-world scenes into the anime images; the another is the discriminator D which discriminates whether the images are from the real target domain or the output produced by the generator. The architecture of AnimeGAN can be seen in Fig.1.

In Fig.1, the generator of AnimeGAN can be considered as a symmetrical encoder-decoder network, which is mainly composed of the standard convolutions, the depthwise separable convolutions [4], the inverted residual blocks (IRBs) [24], the upsampling and downsampling modules. In the generator, the last convolutional layer with 1×1 convolution kernels does not use the normalization layer and is followed by the tanh nonlinear activation function.

The structure of Conv-Block, DSConv and the inverted residual blocks are shown in Fig.2. Conv-Block is composed of the standard convolution with 3×3

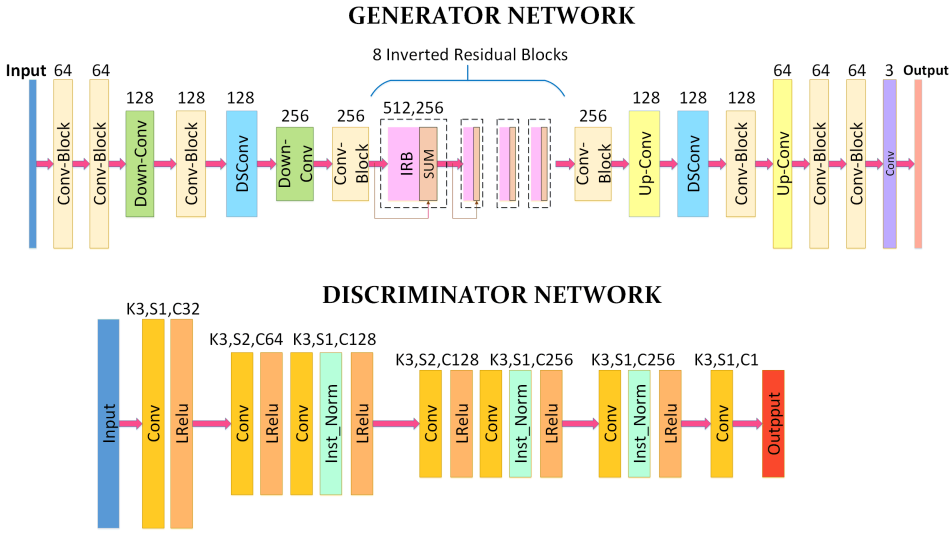

Fig. 1: The architecture of the generator and the discriminator in the proposed AnimeGAN. In the generator, the numbers on all boxes represent the number of channels and SUM means the element-wise sum. In the discriminator, "K" is the kernel size, "C" is the number of the feature maps, "S" is the stride in each convolutional layer and Inst_Norm indicates the instance normalization layer.

convolution kernels, the instance normalization layer [27] and the LRelu activation function [19]. DSConv is composed of the depthwise separable convolution with 3×3 convolution kernels, the instance normalization layer and the LRelu activation function. The inverted residual block contains Conv-Block, the depthwise convolution, the pointwise convolution and the instance normalization layer.

In order to avoid the loss of the feature information caused by max-pooling, the proposed Down-Conv is used as the downsampling module to reduce the resolution of the feature maps. The architecture of Down-Conv is shown in Fig.2. It contains the DSConv module with stride 2 and the DSConv module with stride 1. In Down-Conv, the feature maps are resized to half the size of the input feature maps. The output of the Down-Conv module is the sum of the output of the DSConv module with stride 2 and the DSConv module with stride 1.

The proposed Up-Conv is used as the upsampling module to increase the resolution of the feature maps. The architecture of Up-Conv is also shown in Fig.2. In Up-Conv, the feature maps are resized to 2 times the size of the input feature maps. The Up-Conv module is used instead of the fractionally strided convolutional layer with stride $\frac{1}{2}$ used in the literature [3]. The reason is the upsampling method used in the literature [3] can cause the checkerboard artifacts [22] in the synthesized images and affect the quality of the images.

In order to effectively reduce the number of the parameters of the generator, we use 8 consecutive and identical IRBs in the middle of the network. Compared

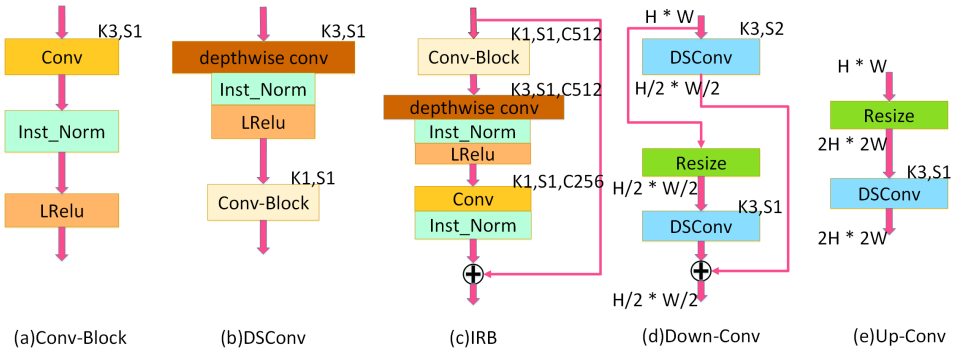

Fig. 2: The architectures of Conv-Block, DSConv, IRB, Down-Conv and Up-Conv in AnimeGAN. The "K" is the kernel size, "C" is the number of feature maps, "S" is the stride in each convolutional layer. The "H" is the height of the feature map, "W" is the width of the feature map. Resize means the interpolation method to set the size of the feature maps. The $\oplus$ indicates the element-wise addition.

with the standard residual blocks [10], IRBs can significantly reduce the number of the parameters and computational workload of the network. IRB used in the generator consists of the pointwise convolution with 512 kernels, the depthwise convolution with 512 kernels and the pointwise convolution with 256 kernels. It is worth noting that the last convolution layer does not use activation function.

Complementary to the generator network, we use the same discriminator network as in the literature [3]. The architecture of the discriminator is shown in Fig.1. All convolutional layers in the discriminator are the standard convolutions. For the weight of each convolutional layer, the spectral normalization [21] is used to make the network training more stable.

3.2 Loss function

We formulate the approach of transforming real-world photos into animation images as an image-to-image mapping model which maps the photo domain P to the animation domain A. AnimeGAN is trained using unpaired training data $S_{data}(p) = \{p_i \mid i = 1, \cdots, N\} \subset P$ and $S_{data}(a) = \{a_i \mid i = 1, \cdots, M\} \subset A$, where N and M are the numbers of the photos and the animation images in the training set, respectively.

In our task, the texture of the animation images instead of the color is transferred into the photo images. We use the grayscale Gram matrix to make the generated images have the texture of the anime images instead of the color of the anime images. Therefore, it is necessary to transform the animation images into the grayscale images for eliminating color interference. The color animation image a_i in $S_{data}(a)$ is transformed to the grayscale image x_i. The grayscale animation training data $S_{data}(x) = \{x_i \mid i = 1, \cdots, M\} \subset X$ corresponding to the

color animation domain $S_{data}(a)$ is obtained. In addition, as mentioned in the reference [3], the training data $S_{data}(e) = \{e_i \mid i = 1, \cdots, M\} \subset E$ is constructed by removing the edges of the animation images in the training data $S_{data}(a)$. In our method, we further process the images in $S_{data}(e)$ into the grayscale images to obtain $S_{data}(y)$. The reason is to avoid the influence of the color of the images in $S_{data}(e)$ on the color of the generated images.

In order to enable AnimeGAN to generate the higher quality images and make the training of the entire network more stable, the least squares loss function in LSGAN [20] is employed as the adversarial loss $L_{adv}(G, D)$. The loss function $L(G, D)$ used in AnimeGAN can be simply expressed as follows:

$$L(G,D) = \omega_{adv} L_{adv}(G,D) + \omega_{con} L_{con}(G,D) + \omega_{gra} L_{gra}(G,D) + \omega_{col} L_{col}(G,D) \tag{1}$$

where $L_{adv}(G, D)$ is the adversarial loss that affects the animation transformation process in the generator G, $L_{con}(G, D)$ is the content loss which helps to make the generated image retain the content of the input photo, $L_{gra}(G, D)$ represents the grayscale style loss which makes the generated images have the clear anime style on the textures and lines. Since the use of the grayscale style loss can easily cause the generated image to be displayed as the grayscale image, $L_{col}(G, D)$ is used as the color reconstruction loss to make the generated images have the color of the original photos. ω_{adv}, ω_{con}, ω_{gra} and ω_{col} are the weights to balance four given loss functions. In all our experiments, we set $\omega_{adv} = 300$, $\omega_{con} = 1.5$, $\omega_{gra} = 3$ and $\omega_{col} = 10$ which achieves a good balance of style and content preservation.

The loss functions used in the generator include the least squares loss function, the content loss function, the grayscale loss function and the color reconstruction loss function. For the content loss $L_{con}(G, D)$ and the grayscale style loss $L_{gra}(G, D)$, the pre-trained VGG19 is used as the perceptual network to extract the high-level semantic features of the images. $L_{con}(G, D)$ and $L_{gra}(G, D)$ can be expressed as:

$$L_{con}(G,D) = E_{p_i \sim S_{data}(p)}[\| VGG_l(p_i) - VGG_l(G(p_i)) \|_1] \tag{2}$$

$$\begin{aligned} L_{gra}(G,D) = E_{p_i \sim S_{data}(p)}, E_{x_i \sim S_{data}(x)}[\| & Gram(VGG_l(G(p_i))) \\ & - Gram(VGG_l(x_i)) \|_1] \end{aligned} \tag{3}$$

where $VGG_l(\bullet)$ refers to the feature maps of the lth layer in VGG and "$\bullet$" indicates the input. In our method, the lth layer is "conv4-4" in VGG. Gram represents the Gram matrix of the features. $G(p_i)$ means the generated image and p_i means the input photo. The $l1$ sparse regularization is used to calculate $L_{con}(G, D)$ and $L_{gra}(G, D)$.

In order to make the image color reconstruction better, we convert the image color in RGB format to the YUV format to build the color reconstruction loss $L_{col}(G, D)$. In $L_{col}(G, D)$, L1 loss is used for the Y channel and Huber Loss is

used for the U and V channels. Formally, $L_{col}(G, D)$ can be defined as:

$$L_{col}(G,D) = E_{p_i \sim S_{data}(p)}[\| Y(G(p_i)) - Y(p_i) \|_1 + \| U(G(p_i)) - U(p_i) \|_H + \| V(G(p_i)) - V(p_i) \|_H] \quad (4)$$

where $Y(p_i)$, $U(p_i)$, $V(p_i)$ represent the three channels of the image p_i in the YUV format, respectively, and H represents Huber Loss. Finally, the loss function of the generator can be expressed as follows:

$$L(G) = \omega_{adv} E_{p_i \sim S_{data}(p)}[(G(p_i) - 1)^2] + \omega_{con} L_{con}(G, D) + \omega_{gra} L_{gra}(G, D) + \omega_{col} L_{col}(G, D) \quad (5)$$

For the loss function used by the discriminator, in addition to introducing edge-promoting adversarial loss [3] to make the images generated by AnimeGAN have clearly reproduced edges, a novel grayscale adversarial loss is also used to prevent the generated image from being displayed as a grayscale image. Finally, the loss function of the discriminator is expressed as follows:

$$L(D) = \omega_{adv}[E_{a_i \sim S_{data}(a)}[(D(a_i) - 1)^2] + E_{p_i \sim S_{data}(p)}[(D(G(p_i)))^2] + E_{x_i \sim S_{data}(x)}[(D(x_i))^2] + 0.1 E_{y_i \sim S_{data}(y)}[(D(y_i))^2]] \quad (6)$$

where $E_{y_i \sim S_{data}(y)}[(D(y_i))^2]$ represents the edge-promoting adversarial loss and $E_{x_i \sim S_{data}(x)}[(D(x_i))^2]$ represents the grayscale adversarial loss. 0.1 is the scaling factor. The purpose of setting the scaling factor of the edge-promoting adversarial loss to 0.1 is to avoid the edges of the generated image being too sharp.

3.3 Training

The proposed AnimeGAN can be easily end-to-end trained with unpaired training data. Since the GAN model is highly nonlinear, with random initialization, the optimization can be easily trapped at suboptimal local minimum. The literature [3] suggests that the pre-training of the generator helps to accelerate GAN convergence. Hence, the generator network G is pre-trained with only the content loss $L_{con}(G, D)$. The initialization training is performed for one epoch and the learning rate is set to 0.0001.

For the training phase of AnimeGAN, the learning rates of the generator and discriminator are 0.00008 and 0.00016, respectively. The training epochs for AnimeGAN is 100 and the batch size is set to 4. Adam optimizer [14] is used to minimize the total loss. AnimeGAN is trained on a Nvidia 1080ti GPU using the Tensorflow [1].

4 Experiments

4.1 Data

The proposed AnimeGAN can be easily end-to-end trained to generate the high-quality anime style images with the unpaired data. The real-world photos as the

content images and the anime images as the style images are used as the training data, and the test data only includes the real-world photos. The resolution of all training images is set to 256×256. For the content images, $6,656$ real-world photos are employed for training and these photos have been used as the training data for CycleGAN [30]. For the style images, since different animation artists have their own unique animation creation styles, in order to obtain a series of the animation images with the same style, we use the key frames of the animated films drawn and directed by the same artist as the style images. In our experiments, 1792 animation images from the movie "The Wind Rises" directed by Miyazaki Hayao are used for training the Miyazaki Hayao style model, 1650 animation images from the movie "Your Name" directed by Makoto Shinkai are used for training the Makoto Shinkai style model and 1553 animation images from the movie "Paprika" directed by Kon Satoshi are used for training the Kon Satoshi style model.

4.2 Results

We first compare AnimeGAN with the two state-of-the-art anime style transfer methods, namely, CartoonGAN [3] and ComixGAN [23]. It is worth noting that CartoonGAN and ComixGAN have the same network structure and loss function. The difference between them is that ComixGAN uses different training strategies and applies animation style transfer to video. Furthermore, the architecture of the discriminator used by AnimeGAN is the same as that used by CartoonGAN. The discriminator used by AnimeGAN is the lightweight convolutional neural network with a model size of $4.30M$. For different generators, the comparisons between them in term of model size, the number of the parameters and computational cost are shown in Table 1. In Table 1, the size of each input photo is 256×256 for inference.

Table 1: Comparisons of the performance of the different generators

Network	Params	Model Size	FLOPs	Inference Time
CartoonGAN	12253152	46.74M	108.98B	51ms/image
AnimeGAN	**3956096**	**15.09M**	**38.66B**	**43ms/image**

As can be seen from Table 1, compared with CartoonGAN, AnimeGAN significantly reduces the number of parameters and computational cost, and has the faster inference speed.

The qualitative results of CartoonGAN, ComixGAN and AnimeGAN are shown in Fig.3. It can be clearly seen that the three methods can effectively capture the anime style. However, the image generated by CartoonGAN produces obvious colorful artifacts in the local areas and loses the color of the original content images. The images generated by ComixGAN are easily excessively stylized

in the local areas, which causes the generated images to lose the content of the original photos. The images generated by AnimeGAN can effectively preserve the content of the photos and the color of the corresponding areas in the photos. Compared with CartoonGAN and ComixGAN, AnimeGAN can produce the higher quality animated visual effects.

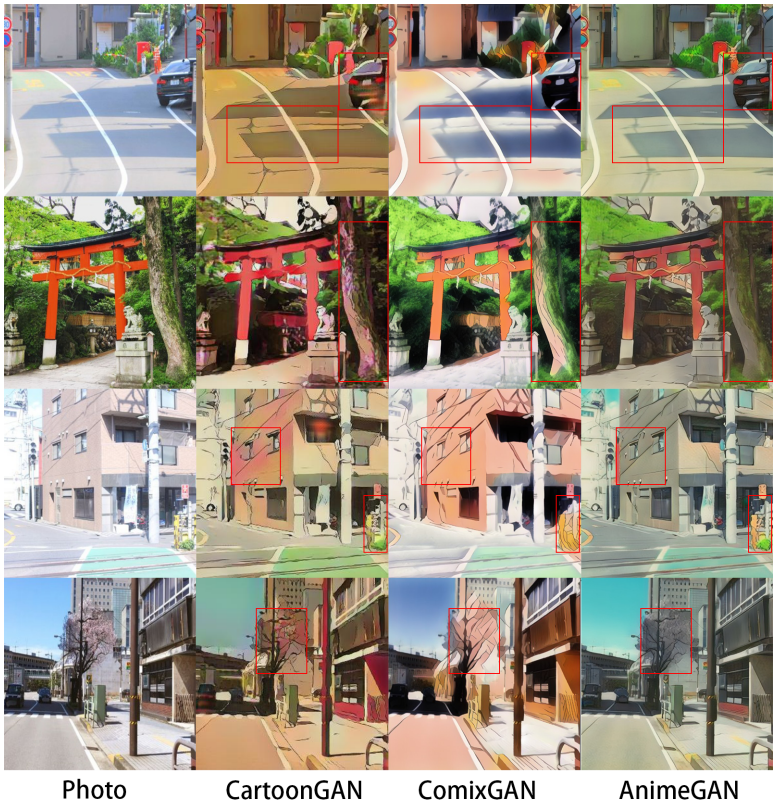

Fig. 3: Qualitative comparison of CartoonGAN, ComixGAN and AnimeGAN.

It is worth noting that the adversarial loss of AnimeGAN is much smaller than the content loss and the grayscale style loss. In order to balance the influence of different losses on the performance of the network, we need to reasonably weight different losses. If the weight of the content loss is too large, the generated images will be very close to the real photo. If the weight of the grayscale style loss is too large, the generated images will lose the content of the original photos. In the experiments, the weights of the content loss and grayscale style loss are set to 1.5 and 3.0, respectively. These two losses are larger than other losses and must be given the small weights.

To further investigate the effect of the weights on the quality of the generated images, we compared the effects of the weight of the adversarial loss and the weight of the color reconstruction loss on the generated images. The comparison results are shown in Fig.4. Compared to AnimeGAN with $\omega_{col} = 10$ and $\omega_{adv} = 300$, the images generated by AnimeGAN with $\omega_{col} = 10$ and $\omega_{adv} = 250$ are closer to the input photo, and the generated images have no obvious animation style. Although the images generated by AnimeGAN with $\omega_{col} = 10$ and $\omega_{adv} = 350$ have obvious animation style, the hue of the generated images will be affected by the style of the training data. As shown in the third column of Fig.4, the images generated by AnimeGAN with $\omega_{col} = 10$ and $\omega_{adv} = 350$ have obvious dark green.

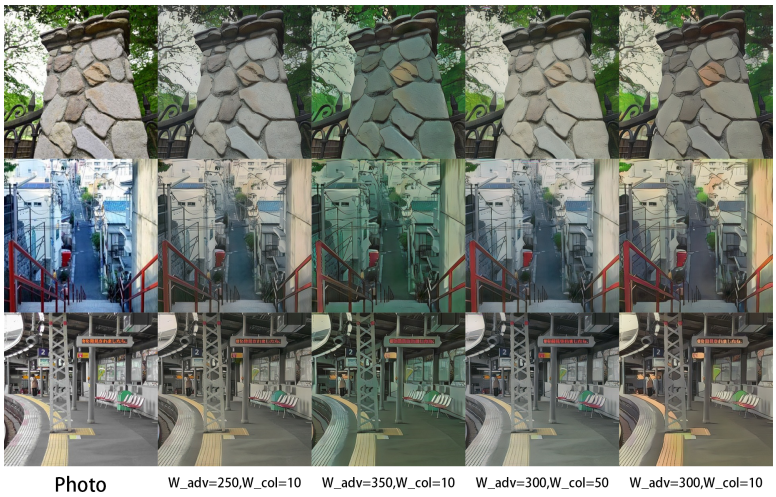

Fig. 4: Qualitative comparison of different weights.

As can be seen from the last two columns in Fig.4, compared to AnimeGAN with $\omega_{col} = 10$, the images generated by AnimeGAN with $\omega_{col} = 50$ have more realistic content but the animation style of the images is not obvious. Therefore, when $\omega_{col} = 10$ and $\omega_{adv} = 300$, the images generated by AnimeGAN have the satisfactory animated visual effects.

The color reconstruction loss is essentially the comparison between the pixels of the generated images and the pixels of the input photos. If its weight is set too large, the generated images are closer to the input photos.

To further study the effect of the grayscale images on the results, we conduct the ablation experiment to analyze the effectiveness of the grayscale adversarial loss and the blurred grayscale edge images $S_{data}(y)$ used for the edge-promoting adversarial loss. In Fig.5, "A" indicates the inference results generated by AnimeGAN without using the grayscale adversarial loss, "B" indicates the inference results generated by AnimeGAN using the blurred color edge images in $S_{data}(e)$

for the edge-promoting adversarial loss, and "C" indicates the inference results generated by AnimeGAN. The results in A and B are similar. The difference between C and B is that the edge-blurred color images in $S_{data}(e)$ are processed into the grayscale images in $S_{data}(y)$ to avoid the color effects of the edge-blurred images on the generated results. Compared with C, the colors in A are not rich enough and saturated. Hence, the grayscale adversarial loss can promote the generated images to have a more saturated color.

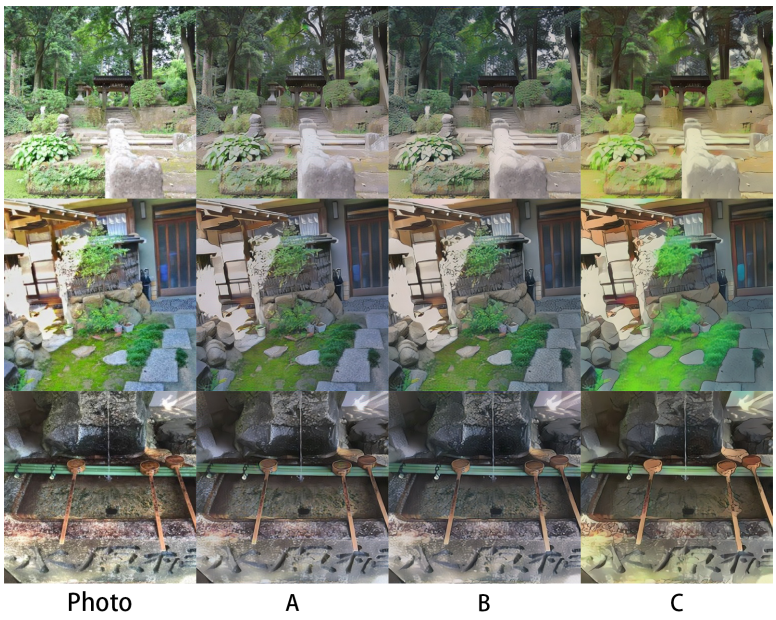

Fig. 5: Qualitative comparison of the grayscale adversarial loss and the different edge-promoting adversarial losses.

Compared with B, the use of the grayscale data from $S_{data}(y)$ for the edge-promoting adversarial loss can not only enable the generated images to have clear edges but also make the generated images have more saturated color.

In order to fully verify the ability of AnimeGAN for photo animation, a large number of experiments were conducted on three animation style datasets. The experimental results are shown in Fig.6. It can be seen that AnimeGAN successfully learns the animation styles of three different artists and generates high-quality animation visual images.

5 Conclusions

In this paper, we proposed AnimeGAN which is the lightweight generative adversarial network to fast transform real-world photos to the high-quality anime

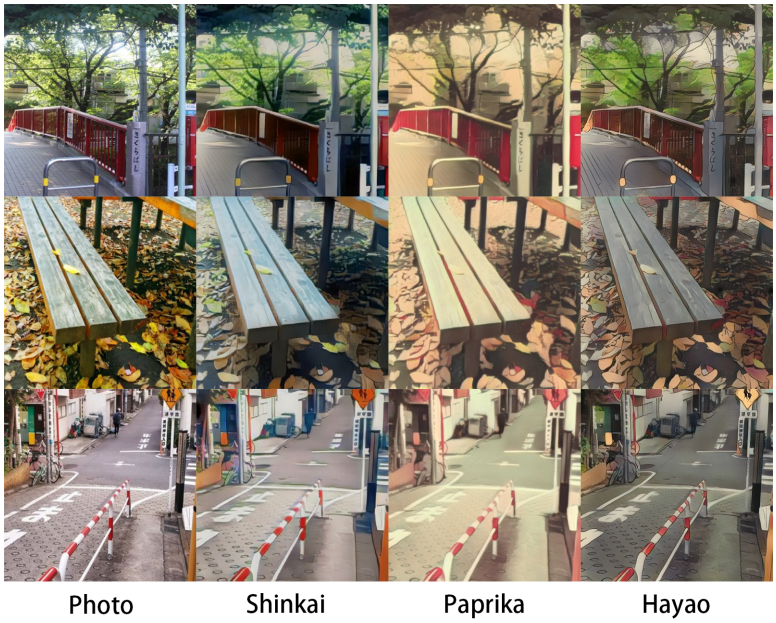

Fig. 6: The results produced by AnimeGAN for 3 different styles.

style images. The main contributions of this paper are as follows:1) the novel grayscale style loss for transforming the anime style textures and lines; 2) the novel color reconstruction loss to preserve the color of the content images; 3) the novel grayscale adversarial loss for preventing the generated images from being displayed as the grayscale images; 4) a lightweight generator using depthwise separable convolutions and inverted residual blocks to achieve faster transfer. The experiments show that AnimeGAN can transform photos of real-world scenes to anime style images with high-quality animated visual effects and significantly outperforms the state-of-the-art stylization methods. In the future work, in order to achieve a faster animation style transfer, we will further study the more lightweight GAN to achieve real-time applications on small mobile devices. Furthermore, it is also considered to integrate the inference phase of AnimeGAN into the video processing pipeline to achieve the animation style transfer of videos.

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. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P.A., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zhang, X.: Tensorflow: A system for large-scale machine learning. CoRR **abs/1605.08695** (2016), http://arxiv.org/abs/1605.08695
2. Almahairi, A., Rajeswar, S., Sordoni, A., Bachman, P., Courville, A.: Augmented cyclegan: Learning many-to-many mappings from unpaired data. In: Proc. 35th International Conference on Machine Learning, ICML 2018. pp. 300–309. Stockholm, Sweden (2018)
3. Chen, Y., Lai, Y.K., Liu, Y.J.: Cartoongan: Generative adversarial networks for photo cartoonization. In: Proc. 31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018. pp. 9465–9474. Salt Lake City, UT, United states (2018)
4. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. pp. 1800–1807. Honolulu, HI, United states (2017)
5. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. CoRR **abs/1508.06576** (2015), http://arxiv.org/abs/1508.06576
6. Gatys, L.A., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. In: Proc. 29th Annual Conference on Neural Information Processing Systems, NIPS 2015. pp. 262–270. Montreal, QC, Canada (2015)
7. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proc. 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016. pp. 2414–2423. Las Vegas, NV, United states (2016)
8. Gatys, L.A., Ecker, A.S., Bethge, M., Hertzmann, A., Shechtman, E.: Controlling perceptual factors in neural style transfer. In: Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. pp. 3730–3738. Honolulu, HI, United states (2017)
9. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proc. 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014. pp. 2672–2680. Montreal, QC, Canada (2014)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016. pp. 770–778. Las Vegas, NV, United states (2016)
11. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proc. 16th IEEE International Conference on Computer Vision, ICCV 2017. pp. 1510–1519. Venice, Italy (2017)
12. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. pp. 5967–5976. Honolulu, HI, United states (2017)
13. Johnson, J., Alahi, A., Li, F.F.: Perceptual losses for real-time style transfer and super-resolution. In: Proc. 14th European Conference on Computer Vision (ECCV'2016). pp. 694–711. Amsterdam, Netherlands (Oct 2016)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proc. the 3rd International Conference for Learning Representations (ICLR'2015). pp. 1–15. San Diego, CA, United states (May 2015)

15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proc. 26th Annual Conference on Neural Information Processing Systems 2012, NIPS 2012. pp. 1097–1105. Lake Tahoe, NV, United states (2012)
16. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Diversified texture synthesis with feed-forward networks. In: Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. pp. 266–274. Honolulu, HI, United states (2017)
17. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. In: Proc. 31st Annual Conference on Neural Information Processing Systems, NIPS 2017. pp. 386–396. Long Beach, CA, United states (2017)
18. Li, Y., Liu, M.Y., Li, X., Yang, M.H., Kautz, J.: A closed-form solution to photorealistic image stylization. In: Proc. 15th European Conference on Computer Vision, ECCV 2018. pp. 468–483. Munich, Germany (2018)
19. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: ICML Workshop on Deep Learning for Audio, Speech and Language Processing (2013)
20. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: Proc. 2017 IEEE International Conference on Computer Vision, ICCV 2017. pp. 2813–2821. Venice, Italy (2017)
21. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. CoRR **abs/1802.05957** (2018), http://arxiv.org/abs/1802.05957
22. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016). https://doi.org/10.23915/distill.00003, http://distill.pub/2016/deconv-checkerboard
23. Pesko, M., Svystun, A., Andruszkiewicz, P., Rokita, P., Trzcinski, T.: Comixify: Transform video into a comics. CoRR **abs/1812.03473** (2018), http://arxiv.org/abs/1812.03473
24. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proc. 31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018. pp. 4510–4520. Honolulu, HI, United states (2018)
25. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Networks **61**(January 01, 2015), 85–117 (2015)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014), http://arxiv.org/abs/1409.1556
27. Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Instance normalization: The missing ingredient for fast stylization. CoRR **abs/1607.08022** (2016), http://arxiv.org/abs/1607.08022
28. Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., Wang, F.Y.: Generative adversarial networks: Introduction and outlook. IEEE/CAA Journal of Automatica Sinica. **4**(4), 588–598 (2017)
29. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proc. 31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018. pp. 8798–8807. Salt Lake City, UT, United states (2018)
30. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proc. 16th IEEE International Conference on Computer Vision, ICCV 2017. pp. 2242–2251. Venice, Italy (2017)