Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 3, March 2019, pp. 910-918
DOI: 10.23940/ijpe.19.03.p20.910918

Colorization for Anime Sketches with Cycle-Consistent Adversarial Network

Guanghua Zhang*, Mengnan Qu*, Yuhao Jin, and Qingpeng Song

School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, 050018, China

Abstract

Coloring animation sketches has always been a complex and interesting task, but as the sketch is the first part of animation creation that neither presents gray value nor presents semantic information, the lack of real animation sketches is the biggest difficulty in current model training. It is also usually expensive to collect such data. In recent years, some methods based on generative adversarial networks (GANs) have achieved great success. They can generate colorized anime illustration on given sketches. Many existing sketch coloring tools are based on this supervised learning method, but the marking of data is particularly important for supervised learning, and much time is spent on the marking of data. To address these challenges, we propose a novel approach for unsupervised learning based on U-net and periodic consistent confrontation. Specifically, we combine the periodic consistent antagonism framework with the U-net structure and residual network, enabling us to robustly train a deep network to make the resulting images more natural and realistic. We also adopted some special data generation methods, so that our model can not only color anime sketches but also extract line drafts from colored pictures. By comparing the mainstream models of supervised learning, we show that the image processed by the proposed method can achieve a similar effect.

Keywords: anime; sketches; colorization; cycle-consistency; u-net

(Submitted on October 24, 2018; Revised on November 26, 2018; Accepted on December 28, 2018)

1. Introduction

Sketch coloring plays an important role in illustration, animation, and other artistic processes. The computer colorization of comics or animations is usually based on a sketch drawn first. Production personnel use professional software to fill the color of the line draft and express the texture of the object in the line draft by controlling the color depth, shade, and other means to give vitality to the picture. The sketch painting can be divided into the following steps: clear the line drawing, layer color, draw shadow color blocks, soften the color edges, and add material effect. Especially in the animation industry, it is referred to as a hard labor. Usually, it takes a long time for a painter to paint excellent works, and computer assistance can effectively improve the speed and quality of paintings. Therefore, there has been more and more research on animation draft simplification and animation sketch colorization. It is a challenging task to develop a quick and direct approach to produce realistic illustrations with line art.

Meanwhile, there has been a new wave for art creation in the artificial intelligence field. Several papers by Gatys et al. [1-3] put forward the concept of making the machine able to understand the ambiguity of style, marking the beginning of style transfer. Over the past five years, generative adversarial networks (GANs) [4-9] played an important role in the development of image generation. New models are constantly being proposed, and the quality of the generated images is getting higher and higher. The image-to-image model derived from these models has gradually become the focus of style transfer researchers. Almost existing models have a good effect on the style transfer of usual tasks such as face-swap, but they are insufficient in the colorization of animation sketches.

In this paper, we propose our model with U-net [10] and Cycle-Consistency. U-net can maintain a balance between input and output. This allows the output image to have low-level information of sketches, but it cannot be ignored that low-

* Corresponding author.
E-mail address: xian_software@163.com

level layers do not get any gradient when performing simple tasks. In order to solve this problem, we fixed the connection network and provided a restricted channel, so that the network can be forced to calculate the gradient.

Recently, a new model using Cycle-Consistency with unpaired image-to-image task was proposed [6]. Cycle-Consistency can solve the problems of collecting datasets, and it can learn a mapping two-way task with unpaired image datasets. This idea of using transitivity as a way to regularize structured data has a long history. In CycleGAN, it has a similar loss to push G and F to be consistent with each other. It can quickly learn the transformation between two different distribution of learning, but it is not very good at processing the image mastery of detail. For tasks like colorization for anime sketches, where it is hard to colorize color collocation, it does not perform optimally.

As a result of the above problems, the proposed model not only has a periodic consistent network structure, but also has the ability to extract details as shown by U-net. This allows our model to handle complex problems such as anime sketch colorization.

Our contributions are:

- A cycle-consistency network to apply the colorization of a painting to a sketch.
- A residual U-net capable with scanty information quantity.
- A discriminator modified from BEGAN suitable to deal with colorization of different colors.

2. Related Work

Neural style transfer [1-3] is a method of image-to-image translation that makes a content image have the style of a style image based on matching the Gram matrix statistics of pre-trained features. The crux is that the representation of content and style can be separated in CNN. The two characterizations can be operated independently to produce new and perceptible images. Unfortunately, the neural style transfer cannot be competent for this kind of task. In fact, the result of neural style transfer on anime sketches is really grotesque, completely unlike a painting.

Generative Adversarial Nets (GANs) [4-9] were introduced by Goodfellow et al. as a new type of generative model, and they have shown impressive results in image generation. There are many models based on them, and they have become the most important part of image processing, such as Pix2Pix [11] and Auxiliary Classifier GANs [10]. They all use paired image-to-image datasets. However, for the animation sketch coloring problem, it is very difficult to obtain image data, and it is even more difficult to obtain paired data. Cycle-Consistent Adversarial Networks (CycleGAN) [6] are a new model using unpaired image-to-image datasets. Therefore, CycleGAN proposal solves this problem to some extent, but it is followed by the problem of image quality decline. The performance of the generator is tailored to the appearance of the object, which makes it difficult to successfully train some objects with more color variations.

Paintschainer [12] abandoned conditional Adversarial Nets [8], which restored to an unconditional discriminator because it easily led the generator to focus too much on the relationship between sketches and paintings and ignored the composition of a painting. Style2paints [13] is also a tool for painters that is based on enhanced residual U-net [10] and auxiliary classifier GAN [7]. It requires two pictures: the sketch and the style image. Moreover, it depends on the classification ability of the VGG 16 or 19, and the quality of the images it produces depends on many factors. In other words, it is more selective. It can produce a good effect on the positive body of the whole body, but if the input image only shows a part of the body or it has some disturbance term, the output effect is not good. Auto-painter [14] is a learning model that can automatically generate painted cartoon images from a sketch based on conditional Adversarial Nets. It uses a feed-forward deep neural network as the generator, which takes the sketch as the input and outputs a colorful image of the same resolution at pixel-level. This model uses many of the same cartoon character pictures for training in the process of practice, and the treatment effect is very good in the process. Cartoon characters such as Spongebob Squarepants were mentioned in the paper. With massive existing finished paintings, though there is a demand for a method to colorize sketches, there is no reliable and mature solution for it.

The similarity of these models is that users only need to give a sketch to get a colorized painting. In view of the above problems, we have improved the data set, network structure, and loss function, so that our model can learn better. Paintschainer is a popular tool for painters and can be increasingly powerful. We compared our results with this popular tool, and the comparisons were shown in the following.

3. Method

Our painter is an unsupervised learning model. with unpaired images dataset. We have reduced unnecessary human intervention to make it look more like we have learned this task. We can give a black-and-white sketch, and the model would generate a colorized image. We employ a deep residual block [15] U-net generator and use cycle consistency [6] to learn how to convert between sketch and colorful image.

3.1. Network Structure

Our model's structure is similar to Cycle-GAN [6], and the overall structure is in Figure 1. We are not the first to use the cycle-consistency or U-net [10]. Prior to this, Iizuka et.al [16] added the high-level layer as a high dimension vector to the global colorization network and achieved good performance in the task of coloring grayscale graphs [17]. The architecture is shown in Figure 1.

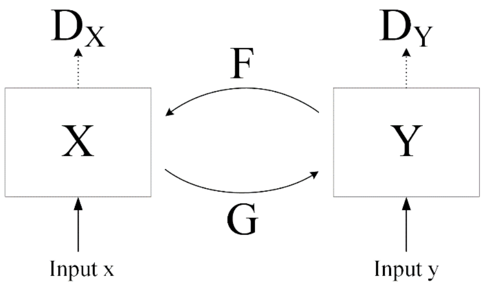
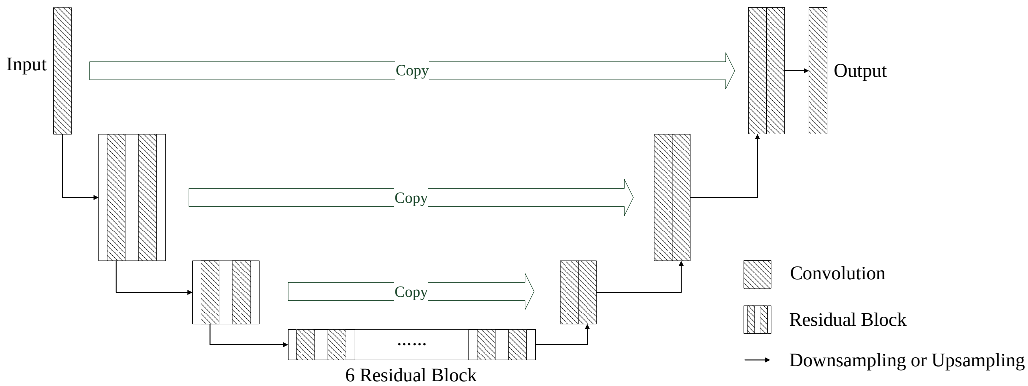


Figure 1. The architecture of our model, which contains two mapping G: $X \rightarrow Y$ and F: $Y \rightarrow X$, and two adversarial discriminators D_X and D_Y. It looks like a large autoencoder, but it has better image processing ability

In fact, if we use U-net as the generator directly, it would not work very well, because when U-net finds that the task "easy" is given, it does not get any gradient at the low-level layers [13]. Therefore, we retained the structure of U-net and adjusted accordingly. We fixed the connection layers and gave a restricted channel, the Upsampling layer, to get the residual value from the Downsampling layer at the same level. Our deep residual block U-net only has three levels, and each block has two convolutions and copies the residual value to the Upsampling layer. The architecture is shown in Figure 2. The shape in the copied feature maps with the matching segmentation map is different; for example, the copied feature map of the second level is 128×128×128, and the upsampling is 128×128×64 (Table 2) [18-20]. We set six residual blocks in the lowest-level (Table 1).



The generator is based on U-net, and we fixed the connection network between the encoder and the decoder.
The copy operation gets the residual value from each residual block.
Figure 2. The architecture of the generator

Table 1. For each residual block, there are two convolution layers with a 3×3 kernel and 1×1 stride

Operation	Kernel	Strides	Feature maps	Batch normalization	Activation function
Convolution	3×3	1×1	512	YES	LeakyReLU
Convolution	3×3	1×1	512	YES	None
Add	-	-	512	NO	None

Table 2. The generator is completely composed of the convolution layer with the specified filter size, stride, and number of filters

Operation	Kernel	Strides	Feature maps	Batch normalization	Activation function
G(*x*)/*F*(*y*) 256×256×3					
Downsampling residual block	7×7	1×1	64	-	LeakyReLU
Downsampling residual block	3×3	2×2	128	-	LeakyReLU
Downsampling residual block	3×3	2×2	256	-	LeakyReLU
Downsampling residual block	3×3	2×2	512	-	LeakyReLU
6 residual block	3×3	1×1	512	-	None
Upsampling convolution	3×3	1/2×1/2	128	YES	LeakyReLU
Add	-	-	256+128	NO	None
Upsampling convolution	3×3	1/2×1/2	64	YES	LeakyReLU
Add	-	-	128+64	NO	None
Upsampling convolution	3×3	1/2×1/2	64	YES	LeakyReLU
Add	-	-	64+64	NO	None
Convolution	7×7	1×1	3	NO	tanh
DX/DY 256×256×3					
Convolution	4×4	2×2	64	NO	LeakyReLU
Convolution	4×4	2×2	256	YES	LeakyReLU
Convolution	4×4	1/2×1/2	128	YES	LeakyReLU
Convolution	4×4	1/2×1/2	64	YES	LeakyReLU
Convolution	4×4	1×1	1	NO	None

Inspired by BEGAN [21], we designed the discriminator D in the form of an auto-encoder [22] to replace the original 70×70 PatchGANs [11,23-24]. The input of D is picture with a depth of 3, and the output is “picture” with a depth of 1 after being encoded and decoded (Figure 2). Our loss function compares the distance between the input *y* and the distribution of the generated image *G*(*x*) through the same encoder.

3.2. Loss Function

The task of our model is similar to colorization but has many differences, such that a grayscale image has more information than a simple sketch. Because of the great difficulties of this task, we need more constraints to make our model perform better. The CycleGAN learns mapping functions between two domains X and Y, and there are two mappings $G: X \rightarrow Y$ and $F: Y \rightarrow X$. The full objective function can be expressed by Equation (1).

$$\min_{G,F} \max_{D_X, D_Y} L(G, F, D_X, D_Y) \quad = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \_L_{cyc}(G, F) \tag{1}$$

Where λ controls the relative importance of the two objectives. It seems like training two autoencoders as one autoencoder $F \circ G: X \rightarrow X$ with another $G \circ F: Y \rightarrow Y$. The loss function $G: X \rightarrow Y$ and its discriminator D_Y can be expressed by Equation (2).

$$L_{GAN}(G, D_Y, X, Y) \quad = \mathbb{E}_{y\sim p_{data}(y)}[log D_Y(y)] + \mathbb{E}_{y\sim p_{data}(y)}[log(1 - D_Y(G(x))] \tag{2}$$

Where G tries to generate images $G(x)$ that look similar to the image from domain Y, while D_Y aims to distinguish between translated samples $G(x)$ and real samples y. In order to obtain more stability during training and generate high quality results, we replaced the negative log objective by a least-squares loss [9]. For the loss function $L_{GAN}(G, D, X, Y)$, we train the G to minimize

$$\mathbb{E}_{x\sim p_{data}(x)}[(D(G(x)) - 1)^2]$$

and the D to minimize

$$\mathbb{E}_{y\sim p_{data}(y)}[(D(y) - 1)^2] + \mathbb{E}_{x\sim p_{data}(x)}[(D(G(x))^2]$$

It can be expressed by Equations (3) and (4).

$$L_G(G, D, X) \quad = \mathbb{E}_{x\sim p_{data}(x)}\left[\left(D\big(G(x)\big) - 1\right)^2\right] \tag{3}$$

$$L_D(G,D,X,Y) \quad = \mathbb{E}_{y\sim p_{data}(y)}[(D(y)-1)^2] + \mathbb{E}_{x\sim p_{data}(x)}[D(G(x))^2] \tag{4}$$

We can use cycle consistency loss to improve the backward cycle consistency, as shown in Equation (5).

$$L_{cyc} = \mathbb{E}_{x\sim p_{data}(x)}\left[\left\|F\big(G(x)\big)-x\right\|_1\right] + \mathbb{E}_{x\sim p_{data}(x)}\left[\left\|G\big(F(x)\big)-y\right\|_1\right] \tag{5}$$

There is an adversarial loss between $F(G(x))$ and x and between $G(F(y))$ and y. It ensures that images and input images are as close as possible.

4. Experiments

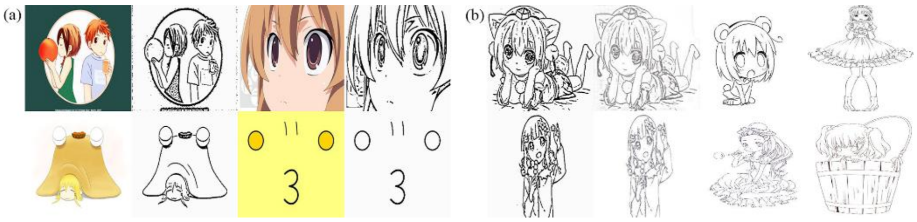
4.1. Dataset

In order to train this model, we collect many colorful anime pictures from the Internet with a crawler. To be more practical in high resolution, our training images are all resized to 256×256. However, it is difficult to find enough sketch images, so we use the boundary detection filter XDoG [25] to extract sketches (see Figure 3(a)). By adjusting the parameters γ of the XDoG, we can get different details of sketches. We finally include a type of sketch in training with the parameter γ set to 0.97.

4.2. Train Details

In the actual experiment, we adopted two techniques to improve our training effect. One of them is obtained through the XDoG algorithm after the dataset, and we obtained some real animation sketches through the crawler (about 1K). We first trained a CycleGAN model by using these real animation sketches and artificial fake sketches made by the XDoG algorithm and then produced a batch of new data sets for our model training (see Figure 3(b)).

The purpose of this is to make the original distribution of our training data be more similar to the distribution in reality. Another purpose is that during the training, we take some data (about 20%) to train to over-fitting and then train all the data. This can improve the training speed of the model to a certain extent. The trend of loss during training is shown in the figure.



(a) (b)

For each group, the left picture is the raw picture and the right picture is the sketch from the right, using XDoG
Comparison of samples using XDoG generated sketch (left), images processed by CycleGAN (middle), and real sketch images (right).
Figure 3. Generated black-white sketches using the XDoG filter [14]

For the experiment, we used about 10K datasets total and trained ten epochs with NVIDIA GeForce 940M, and we set all models λ = 10 and used the Adam optimizer with learning rate 0.0002. The leaky values were set to 0.2.

4.3. Comparisons

We compare our method with four methods:

- Canna: Canna, the default method of PaintsChainer [12].
- Satsuki: Satsuki, the second method of PaintsChainer.
- Tanpopo: Tanpopo, the third method of PaintsChainer.
- Deepcolor [26]: The method proposed by Kevin Frans.

Qualitative Comparisons Figures 4 and 5 show the comparisons of the same samples, and our model shows obvious characteristics and advantages. In terms of color alone, our model can add more comfortable colors to the sketch. One of the major differences between our model and other models is that the images generated by our model dilute the original sketch, and the images processed by other models retain the original thicker lines. An unexpected benefit was found in the training process. The highlights were added to the images in our model. We think this is because the U-net network structure helps to extract the small structural features in the image. Adding a U-net structure to the generated model can pay attention to detail when generating a colorized image.

We selected ten images to test on CycleGAN and our model; the middle pictures are from CycleGAN and the right pictures are from our model. All models have been trained 200K iterations. The results from our model show suitable colorization.

Figure 4. Comparisons of testing for CycleGAN and our model

In terms of detail processing, our model can generate smoother images of lines and automatically repair some defects in the sketch (Figure 6). These models in the comparison are overly dependent on the input image, which further magnifies the defect of the input image. Our model not only has strong generalization ability but also does not rely too much on the input image. In particular, our model can generate many colorful results for the same sketch (see Figure 7).

Quantitative Comparisons Evaluating the quality of synthesized images is an open and difficult problem; as mentioned in [17], there is no good numerical metric to evaluate image inpainting result due to the existence of many possible solutions. Unlike the colorization of grayscales, anime sketches have too little information to use traditional metrics such as PSNR. Therefore, we use the mean opinion score test to judge the quality of generated images. This method uses people's subjective feelings to score images, and the average score reflects the effect of the method.

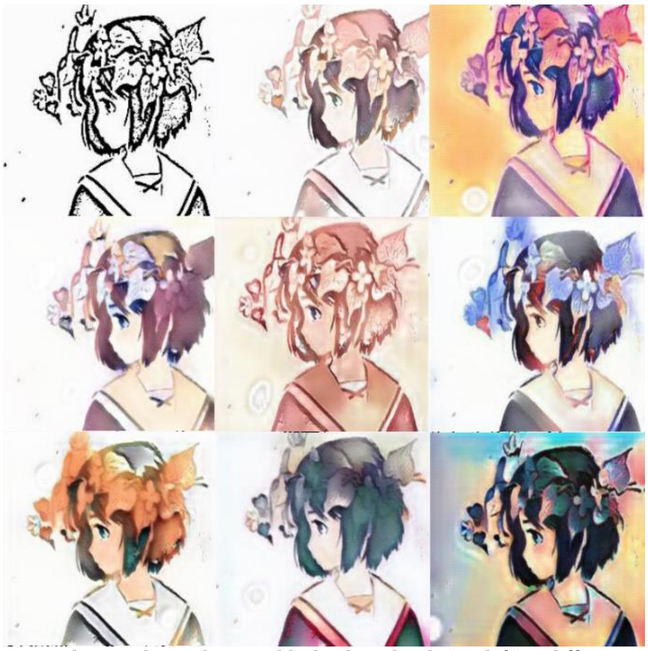
We confirm the performance of our model using Mean Opinion Score (MOS) testing. We have asked 24 human raters to assign a score from 1 (bad quality) to 5 (excellent quality) to quantify the ability of different approach of painting. In total, 1296 ratings were obtained, where each rater rated 54 images from five methods: Deepcolor, PaintsChainer's Canna, Tanpopo, Satsuki, and our method. Images were presented in a completely randomized fashion without any indication of the approach. The experimental results of the conduct MOS tests are summarized in Table 3. The score of our model is very close to that of other models of supervised learning, which proves that our model is comparable to current popular models in the case of no supervision.

We selected seven images to test on PaintsChainer and our model and compared the test results.
On the left is the input image, followed by the images processed by Deepcolor, Canna, Tanpopo, Satsuki, and our model.
Figure 5. Comparisons of testing on each sketch

The result of our model (left) is smooth lines and even coloring, while the images processed by Deepcolor (middle) and PaintsChainer (right) still have rough serrated edges.
Figure 6. Contrast image details

Table 3. Performance of different methods for automatic colorization on our sketches dataset. Our model has a slightly higher MOS than the other methods

Training Configuration	MOS
PaintsChainer (Canna)	3.646
PaintsChainer (Tanpopo)	2.953
PaintsChainer (Satsuki)	3.372
Deepcolor	3.528
Ours	3.632



Our approach can colorize the same black-white sketch (top-left) in different styles.
Some others would colorize the same sketch in a single style, such as Paintschainer.
Figure 7. Comparisons of many-times testing results with the same picture

5. Conclusions

We have presented the results of our method and comparison with other popular tools. To the best of our knowledge, this is the first time that unpaired image-to-image unsupervised learning has been applied to coloring animation sketches. We have made some progress in the results of model training, including details processing, color matching, and other aspects. Our model can complete the task of coloring the black-white sketch through its own learning, and the quality of the image generated is not inferior to other supervised learning models. However, it needs to be stressed that the images generated by these machines are not ready for commercial use, as some limitations were worth noticing such as image resolution and clarity.

Acknowledgements

All the pictures for training come from Pixiv and Konachan. The work is supported by the National Key Research and Development Program of China (No. 2016YFB0800703), the National Natural Science Foundation of China (No. 61572255), and the Science & Technology Research Project of Hebei Higher Education Institutions (No. ZD2018236).

References

1. L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer using Convolutional Neural Networks," in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414-2423, IEEE, 2016
2. L. A. Gatys, A. S. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," arXiv Preprint arXiv: 1508. 06576, 2015
3. L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman, "Preserving Color in Neural Artistic Style Transfer," arXiv Preprint arXiv: 1606. 05897v1, 2016
4. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., "Generative Adversarial Nets," in *Proceedings of International Conference on Neural Information Processing Systems*, pp. 2672-2680, MIT Press, 2014
5. A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," arXiv preprint arXiv: 1511. 06434, 2015
6. J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 2242-2251, IEEE Computer Society, 2017
7. A. Odena, C. Olah, and J. Shlens, "Conditional Image Synthesis with Auxiliary Classifier GANs," arXiv Preprint arXiv: 1610. 09585v4, 2016
8. M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," in *Proceedings of International Conference on Neural Information Processing Systems*, pp. 2672-2680, MIT Press, 2014
9. X. Mao, Q. Li, H. Xie, R. Y.K. Lau, Z. Wang, and S. P. Smolly, "Least Squares Generative Adversarial Networks," arXiv Preprint arXiv: 1611. 04076v3, 2016

10. O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234-241, Springer, Cham, 2015
11. P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5967-5976, IEEE Computer Society, 2017
12. “Paintschainer,” (https://github.com/pfnet/PaintsChainer, accessed March 29 2017)
13. L. Zhang, Y. Ji, and X. Lin, “Style Transfer for Anime Sketches with Enhanced Residual U-Net and Auxiliary Classifier GAN,” arXiv Preprint arXiv: 1706. 03319v2, 2017
14. Y. Liu, Z. Qin, T. Wan, and Z. Luo, “Auto-Painter: Cartoon Image Generation from Sketch by using Conditional Generative Adversarial Networks,” arXiv Preprint arXiv: 1705. 01908, 2017
15. K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, IEEE Computer Society, 2016
16. S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Let There Be Color: Joint End-to-End Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification,” *ACM Transactions on Graphics*, *ACM*, Vol. 35, No. 110, 2016
17. R. Zhang, P. Isola, and A. A. Efros, “Colorful Image Colorization,” in *Proceedings of European Conference on Computer Vision*, pp. 649-666, Springer, Cham, 2016
18. J. Yu, Z, Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Generative Image Inpainting with Contextual Attention,” arXiv Preprint arXiv: 1801. 07892, 2018
19. K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” arXiv Preprint arXiv: 1409. 1556, 2014
20. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context Encoders: Feature Learning by Inpainting,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536-2544, IEEE Computer Society, 2016
21. D. Berthelot, T. Schumm, and L. Metz, “BEGAN: Boundary Equilibrium Generative Adversarial Networks,” arXiv Preprint arXiv: 1703. 10717, 2017
22. D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” arXiv Preprint arXiv: 1312. 6114, 2013
23. C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, et al., “Photo-Realistic Single Image Super-Resolution using a Generative Adversarial Network,” arXiv Preprint arXiv: 1609. 04802, 2016
24. C. Li and M. Wand, “Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks,” in *Proceedings of European Conference on Computer Vision*, pp. 702-716, Springer, Cham, 2016
25. H. Winnemöller, “XDoG: Advanced Image Stylization with Extended Difference-of-Gaussiansm,” in *Proceedings of ACM Siggraph/Eurographics Symposium on Non-Photorealistic Animation and Rendering*, pp. 147-156, ACM, 2011
26. “Deepcolor: Automatic Coloring and Shading of Manga-Style Lineart,” (http://kvfrans.com/coloring-and-shading-line-art-automatically-through-conditional-gans/, accessed March 1, 2017)

Guanghua Zhang received his B.S. degree from Hebei Normal University, China, in 2002. He received his M.S. degree and Ph.D. from Xidian University, China, in 2005 and 2014 respectively. He is currently a vice professor in the College of Information Science and Engineering, Hebei University of Science and Technology, China. His main research interests include wireless security, trust management, and cooperative spectrum sensing.

Mengnan Qu is an undergraduate student in the School of Information Science and Engineering, Hebei University of Science and Technology. His research interests include machine learning and digital image processing.

Yuhao Jin is an undergraduate student in the School of Information Science and Engineering, Hebei University of Science and Technology. His research interests include machine learning and information security technology.

Qingpeng Song is an undergraduate student in the School of Information Science and Engineering, Hebei University of Science and Technology. His research interests include machine learning.