An Analysis: Different Methods about Line Art Colorization

Jinhui Gao[1, a†], Ruihao Zeng[2, b†], Yuan Liang[3, c*†], Xinyu Diao[4, d*†]
[1]International School of Information Science and Engineering, Dalian University of Technology, Dalian, Liaoning 116620, China
[2]International College of Chinese Studies, Fujian Normal University, Fuzhou, Fujian 350117, China
[3]School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, Sichuan 611756, China
[4]School of Information Technology & Management, University of International Business and Economics, Beijing 100105, China
[a]1752710993@mail.dlut.edu.cn
[b]154042018010@student.fjnu.edu.cn
[c]*2018110698@my.swjtu.edu.cn
[d*]201883016@uibe.edu.cn
[†]These authors contributed equally.

ABSTRACT

We have conducted a series of studies and analyses to address the problem of line art colorization. We chose Generative Adversarial Networks (GANs), a leading neural network architecture for solving this problem, as our focus. For a large number of studies based on this architecture, we improved, applied, and analytically compared four methods, pix2pix, pix2pixHD, white-box, and scaled Fourier transform (SCFT), which can represent the mainstream problem-solving direction in the field of line colorization to the greatest extent possible. Finally, two reference quantities were introduced to quantify the results of the analysis.

Keywords-component; line art colorization; GANs; pix2pix; white-box; SCFT

1. INTRODUCTION

Line coloring has always been considered an extremely creative task that often plays a decisive role in the entire painting because different coloring styles can produce very different visual effects. With the increasing development of deep learning, it is possible to use deep algorithms for line colorization. The successful practice of automatic line colorization not only further proves the creativity of deep learning algorithms but also brings the possibility to relieve the workload of coloring for front-line artists. Meanwhile, automatic line colorization has many roles to play in many other fields, such as basic education.

As a creative network structure, Generative Adversarial Networks (GANs) [1] are well suited for solving the problem of line art colorization. In general, the framework of GANs consists of a generator and a discriminator. During training, the two components are trained on each other. The task of the generator is to generate as realistic results as possible to deceive the discriminator, while the discriminator is expected to improve the recognition of the generated products through continuous training iterations. However, the original GANs often suffer from problems such as pattern collapse or gradient disappearance. To solve this problem, conditional GAN (cGAN) [2] was introduced, which stabilized the generated data of GANs by adding supervised information. After this, more and more line coloring problems are solved using GANs, such as BicycleGAN, StyleGAN, and Tag2pix.

To better understand and use the automatic coloring technique for line art, we analyzed several algorithms through their structure, time complexity, and performances. Briefly, we make the following contributions in this work:

- We *analyzed* the latest methods of line colorization.
- We *improved* and *implemented* four different methods under the same inputs.

2nd International Conference on Artificial Intelligence, Automation, and High-Performance Computing (AIAHPC 2022), edited by Ligu Zhu, Proc. of SPIE Vol. 12348, 123483X
© 2022 SPIE · 0277-786X · doi: 10.1117/12.2641852

- We quantitatively *analyzed* the experimental results using Fréchet Inception Distance (FID) and Structure Similarity Index (SSIM).

The organization of this article is as follows. In Section 2 (Related Works), we present a detailed study of the state-of-the-art methods. In Section 3 (Methods), we introduce the basic structure of GANs and the four different approaches we improved and implemented. We discuss the details of the experimental environment and dataset as well as the quantitative comparisons of four methods in Section 4 (Experiments). Finally, in Section 5 (Conclusion), the work of this paper is summarized, and the conclusions of the experimental analysis are presented

2. RELATED WORKS

Taking the mainstream methods in the field as an example, line coloring techniques can be divided into two different research directions depending on whether human intervention is required or not, which are user-guided colorization and fully automatic colorization.

2.1 User-guided colorization

Earlier user-guided coloring methods had a great degree of dependence on the user. The most notable words are presented by Huang et al. [3] and Levin et al. [4], which used similar luminance to propagate user point control over the brush color. Such an approach is based on the theory that neighboring pixels of similar luminance tend to have similar colors. Therefore, these methods usually require very constant and close interaction with users as the users need to specify the area frequently. After this, for some specific cases of coloring needs, there are also studies to improve the similarity-metric-based approaches by different methods such as chromaticity blending [5] and subdivision of textures [6]. To reduce excessive user interaction, later Luan et al. proposed an optimized similarity metric to reduce user input [7]. In addition to the similarity metric, global optimization methods using all-pair constraints [8], deep learning methods using boosting [9], and manifold learning [10] can improve the learning efficiency for stroke colors. After this, Frans et al. [11] proposed the use of artistic lines for user-guided coloring to bring line coloring to a whole new stage. This was followed by smaller blocks of color (hint) [12] and reference drawings [13] with a certain degree of similarity for line-art colorization, which has largely freed up the amount of user interaction.

2.2 Fully-automatic colorization

As another direction of research in line colorization -- the fully automatic colorization methods [14,15,16], have completely abandoned the need for user interaction. Instead of requiring color information for each region, these methods can be trained on large amounts of data by Convolutional Neural Networks (CNNs), from which a direct mapping of grayscale images to up-colored images can be learned. These deep learning-based methods can simulate realistic color changes in conjunction with the underlying details of the image (e.g., structure, style, etc.). However, this class of methods has some obvious drawbacks, such as the uncertainty of coloring. The combination of user control and CNNs has been proposed by Zhang et al. [17] to control the uncertainty of this class of methods. Such an approach is also more suitable for providing services for art education. Furthermore, the concept of semantic segmentation has been introduced to the field of line coloring and even image stylization. Methods such as VGG networks [18], ResNet [19], U-Net [20], and InceptionV4 [21] are being widely used.

3. METHODS

3.1 Conditional generative adversarial networks

Even though GAN models provide powerful performance for image generation tasks and transfer brush colors well, there are still problems such as pattern collapse and gradient disappearance, which are however well addressed by the introduction of cGAN. cGAN can provide class labels for both generators and discriminators to generate conditional samples, which can well improve the quality of generated images. A state-of-the-art landmark achievement is cWGAN-GP proposed by Ci [22], which improves the generalization of contour lines using local feature networks.

3.2 Pix2pix method

Pix2pix [23] realizes image translation based on cGAN, which can guide image generation by adding conditional information. Therefore, in image translation, the input image can be taken as a condition to learn the mapping between the input and output images to obtain the specified output image. However, for other image translations based on GAN, because the generator of GAN algorithm is based on a random noise to generate images, it is difficult to control the output,

so the image generation is basically guided by other constraints, rather than using cGAN. This is the difference between pix2pix and other GAN-based image translations. And the other differences are specifically reflected in generators, discriminators, and LOSS functions.

U-Net is introduced for the generator of pix2pix, which is widely used in the field of image segmentation and can fully integrate features.

In terms of the discriminator, PatchGAN is used to output a prediction probability value for each patch of the input image, which is equivalent to the transformation from judging whether the input is true or false to judging whether the input is true or false in $N \times N$ size area.

3.3 Pix2pixHD method

The images obtained by pix2pix are not clear enough. A new generation framework, named pix2pixHD [24], enhances the original pix2pix framework using a more detailed generator, a multi-scale discriminator structure, and a robust adversarial learning objective function. We adapted the pix2pixHD to the line colorization task.

In the generator, we decomposed it into two self-networks $G1$ (global network) and $G2$ (local boosting network). These two have similar structures, and overall can be seen as a regular generator diving into another generator.

In the discriminator structure, three layers of image pyramids (e.g., $2014 \times 1024,\ 1024 \times 512,\ 512 \times 256$) are first constructed based on real and synthetic images, and a discriminator is trained for each layer separately to discriminate the images. This part is designed in the hope that the discriminator with small size promotes the synthesis of the overall image aspect and that the discriminator with large size promotes the synthesis of the detailed image aspect.

3.4 White-box method

The white-box colorization [25] is optimized from white-box cartoon representation, which aims for real image cartoonization. Under the demand of colorizing with designated color, we take the hint, named Atari, as the input of the neural network. Also, we revised several structure details to make them better suited for sketch colorization.

The principle of this method is to identify three white-box representations from images, i.e., the surface representation that contains a smooth surface of images, the structure representation that refers to the sparse color-blocks and flattens global content in the celluloid style workflow, and the texture representation that reflects the high-frequency texture and details in colorized images.

Then a GAN network using U-Net with a generator G and two discriminators D_s and D_t, which are relatively simple convolutional neural networks, are introduced to construct the network where D_s and D_t aim to distinguish between surface and texture representations extracted from model outputs and line art with Atari. Also, a pre-trained VGG-19 network is used to extract high-level features and to impose spatial constrain on global contents between extracted structure representations and outputs and between input arts and outputs. According to our problem, we modify some of the architecture as shown in Figure 1. Our work does not use a real image, so we remove the L_{tv} and $L_{content}$, because they contribute to the transformation from a real picture to a cartoon image.

The surface representation imitates the painting style. To smooth the images and meanwhile keep the global semantic structure, the differentiable guided filter is adopted for edge-preserving filtering, which is denoted as F_{gdf}. Structure representation emulates flattened global content. To enhance the contrast of images and reduce the hazing effect, it uses an adaptive coloring algorithm F_{st} while the loss function uses the VGG network for pre-training. The last texture representation is produced by a random color shift algorithm F_{rcs} that converts the RGB map to a single-channel map. Loss functions of surface, structure, and texture are displayed in Equation 1.

$$\begin{cases} \mathcal{L}_{surface}(G, D_s) = logD_s(\mathcal{F}_{dgf}(I_c, I_c)) \\ \mathcal{L}_{structure} = \left\| VGG_n\left(G(I_p)\right) - VGG_n\left(\mathcal{F}_{st}\left(G(I_p)\right)\right) \right\| \\ \mathcal{L}_{texture}(G, D_t) = logD_t(\mathcal{F}_{rcs}(I_c)) + \log\left(1 - D_t(\mathcal{F}_{rcs}(G(I_p)))\right) \end{cases} \tag{1}$$

So, we can get the total loss function by adding up all the loss functions with parameter λ, as shown in Equation 2.

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{sur} + \lambda_2 \mathcal{L}_{tex} + \lambda_3 \mathcal{L}_{struc} + \lambda_4 \mathcal{L}_{cont} + \lambda_5 \mathcal{L}_{tv} \tag{2}$$

Based on this, the weight for each component can be adjusted in the loss function for us to adapt to the sketch colorization case we are facing.

3.5 SCFT method

SCFT-based method [26] mainly uses a spatially corresponding feature transfer (SCFT) module to transfer information from the encoded images generated by augmented-self reference to the encoded images extracted by outline extractor, inspired by the idea of the self-attention mechanism of the transformer. Then, based on the visual mappings from the SCFT module, context features are passed through several residual blocks and a U-Net-based decoder to train the discriminator.

Appearance and spatial transformations are applied to generate a reference color image using augmented-self reference generation. Some random noises are added to each RGB channel in every iteration. Also, this output image will be the ground truth of the whole model at test time. Afterward, the thin-plate splines (TPS) transformation, a non-linear spatial transformation operator, is applied to the result in the final reference image, preventing bringing the color in the same position.

After the images are generated, both are feed to the SCFT module to decide from which part of the reference image to bring information and from which part of a sketch image to transfer the information. Also, it is important that each encoder consists of L convolutional layers, producing L activation maps $(f^1, f^2, \cdots, f^L)$ including intermediate outputs and forming the final activation map V as shown in Equation 3.

$$V = [\varphi(f^1); \varphi(f^1); \cdots; \varphi(f^{l_p})] \tag{3}$$

As for the objective function, the total loss is considered as the aggregation of similarity-based triplet loss, including L1 loss, adversarial loss, perceptual loss, and style loss.

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{tr} + \lambda_2 \mathcal{L}_{rec} + \lambda_3 \mathcal{L}_{adv} + \lambda_4 \mathcal{L}_{perc} + \lambda_5 \mathcal{L}_{style} \tag{4}$$

4. EXPERIMENTS

4.1 Dataset description

We use a dataset that consists of richly tagged and labeled artwork depicting characters from Japanese anime, and they are collected from two imageboards Danbooru and Moeimouto. All images in the dataset have been tagged as SFW (non-explicit). The dataset file has a subset of 300,000 images that are in normalized size format of 512×512 px. The total amount of data is about 45.34 GB. In this study, we only took some data for experiments.

4.2 Experimental settings

We implement the GANs methods with PyTorch using Colab Pro powered by Tesla P100 GPU. The ADAM optimizer is used in each network. Learning rate and batch size are set to 1×10^{-4} and 16 during training. Considering the high efficiency, we trained the models by 3000 iterations.

4.3 Line Extraction

The quality of the line is also an important factor that affects the result of the line art colorization. With access to cleaned images, our priority is to process edge detection and sketch extraction. We first adopted the XDoG method, which is a general image processing algorithm. It uses the difference of two Gauss operators to remove low-frequency information and noise in the middle and high frequencies from the full frequency spectrum. We also used HED, which results from weighted loss generated by the simultaneous training of six layers of fully convolutional networks based on VGG-16. To avoid overfitting, lines extracted from both two kinds of edge detection methods are used as the input of the neural network.

4.4 Quantitative comparisons

We conducted experiments with each of the four methods under the same data set and were able to obtain the following results, as shown in Fig. 1 to Fig. 4.

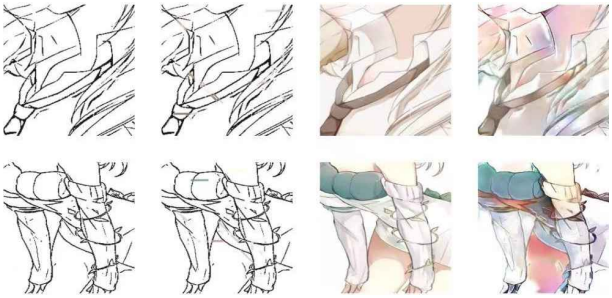

Figure 1. pix2pix

Figure 2. pix2pixHD

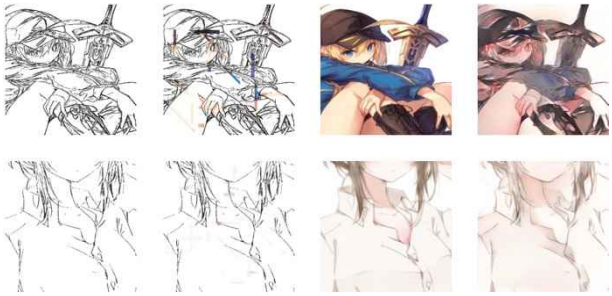

Figure 3. white-box

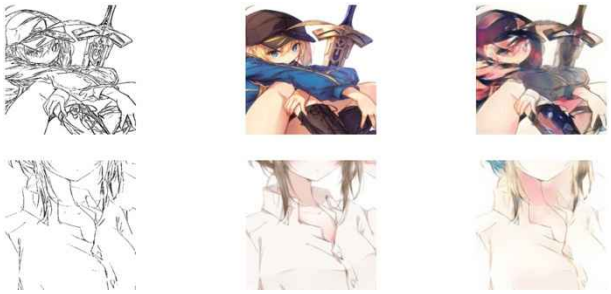

Figure 4. SCFT

In order to be able to compare the coloring results quantitatively, we introduced two performance metrics:

- Fréchet Inception Distance (FID)

FID [27] calculates the distance between the real and generated samples in the feature space. The Inception network is first used to extract the features, then the feature space is modeled using a Gaussian model, and then the distance between two features is solved. A lower FID means that the two distributions are closer to each other, which means that the resulting images are of higher quality and better diversity. If there is only one kind of image, the distance of FID will be quite high. Therefore, FID is more suitable to describe the diversity of GANs.

- Structure Similarity Index (SSIM)

SSIM [28] is a perceptual metric based on error sensitivity that can quantify the degradation of image quality caused by processing. It requires two images from the same image sample, i.e., a reference image and a generated image. By comparing the brightness and contrast of the images, SSIM can derive the structural differences between the two images. Therefore, the higher the SSIM value, the more similar the two images are and the better the colorization is.

We perform 10 experiments on each of the four algorithms with the same data set and take the average results for comparison. The results are presented in Table 1.

Table 1 Average indexes

parameter	FID	SSIM
pix2pix	95.4	0.74
pix2pixHD	81.2	0.79
white-box	57.3	0.80
SCFT	78.2	0.77

From the coloring results and the two indicators, we can see that the white-box and pix2pixHD methods are relatively good coloring results, and pix2pix is less effective.

5. CONCLUSION

Overall, we have compared the strengths and the weakness of different coloring techniques under the same data set by modifying and applying existing theories. Besides, two parameters, FID and SSIM, were introduced to quantify the coloring results for comparison, which concluded that the white-box algorithm and pix2pixHD algorithm have better performance in terms of coloring quality.

Also, we found that in many of the tests, the shading of hair and clothing was not very accurate. In future work, we will improve the robustness and accuracy of the methods and find the cause from the perspective of interpretability. We know that the interpretable GAN is very difficult, as there is much rich semantic information in GAN latent space. Many previous studies often train attribute classifiers using classical machine learning algorithms, such as SVM, to find different attributes in the latent space of the hyperplane interface and discover semantic information. However, this requires a lot of advanced assumptions and definitions of the target attributes and a lot of time to train the classifier. We found a closed trial solution algorithm to make the potential semantic discovery by directly decompressing pre-trained weights through reading. With this fast implementation, the method is comparable to supervised learning algorithms in finding semantically associated vectors in latent space and finds many more rich concepts than some attribute classifiers.

REFERENCES

[1] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, *35*(1), 53-65.

[2] Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

[3] Huang, Y. C., Tung, Y. S., Chen, J. C., Wang, S. W., & Wu, J. L. (2005, November). An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on Multimedia* (pp. 351-354).

[4] Levin, A., Lischinski, D., & Weiss, Y. (2004). Colorization using optimization. In *ACM SIGGRAPH 2004 Papers* (pp. 689-694).

[5] Yatziv, L., & Sapiro, G. (2006). Fast image and video colorization using chrominance blending. *IEEE transactions on image processing*, *15*(5), 1120-1129.

[6] Qu, Y., Wong, T. T., & Heng, P. A. (2006). Manga colorization. *ACM Transactions on Graphics (TOG)*, *25*(3), 1214-1220.

[7] Luan, Q., Wen, F., Cohen-Or, D., Liang, L., Xu, Y. Q., & Shum, H. Y. (2007, June). Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques* (pp. 309-320).

[8] Xu, K., Li, Y., Ju, T., Hu, S. M., & Liu, T. Q. (2009). Efficient affinity-based edit propagation using kd tree. *ACM Transactions on Graphics (TOG)*, *28*(5), 1-6.

[9] Li, Y., Adelson, E., & Agarwala, A. (2008, June). ScribbleBoost: Adding Classification to Edge-Aware Interpolation of Local Image and Video Adjustments. In *Computer Graphics Forum* (Vol. 27, No. 4, pp. 1255-1264). Oxford, UK: Blackwell Publishing Ltd.
[10] Chen, X., Zou, D., Zhao, Q., & Tan, P. (2012). Manifold preserving edit propagation. *ACM Transactions on Graphics (TOG)*, *31*(6), 1-7.
[11] Frans, K. (2017). Outline Colorization through Tandem Adversarial Networks.
[12] Liu, Y., Qin, Z., Wan, T., & Luo, Z. (2018). Auto-painter: Cartoon image generation from sketch by using conditional Wasserstein generative adversarial networks. *Neurocomputing*, *311*, 78-87.
[13] Zhang, L., Ji, Y., Lin, X., & Liu, C. (2017, November). Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. In *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)* (pp. 506-511). IEEE.
[14] Cheng, Z., Yang, Q., & Sheng, B. (2015). Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 415-423).
[15] Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2016). 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 (ToG)*, *35*(4), 1-11.
[16] Zhang, R., Isola, P., & Efros, A. A. (2016, October). Colorful image colorization. In *European conference on computer vision* (pp. 649-666). Springer, Cham.
[17] Zhang, R., Zhu, J. Y., Isola, P., Geng, X., Lin, A. S., Yu, T., & Efros, A. A. (2017). Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*.
[18] Sengupta, A., Ye, Y., Wang, R., Liu, C., & Roy, K. (2019). Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in neuroscience*, *13*, 95.
[19] Targ, S., Almeida, D., & Lyman, K. (2016). Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*.
[20] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... & Rueckert, D. (2018). Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*.
[21] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.
[22] Ci, Y., Ma, X., Wang, Z., Li, H., & Luo, Z. (2018, October). User-guided deep anime line art colorization with conditional adversarial networks. In *Proceedings of the 26th ACM international conference on Multimedia* (pp. 1536-1544).
[23] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).
[24] Kim, H., Jhoo, H. Y., Park, E., & Yoo, S. (2019). Tag2pix: Line art colorization using text tag with secat and changing loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 9056-9065).
[25] Wang, X., & Yu, J. (2020). Learning to cartoonize using white-box cartoon representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8090-8099).
[26] Lee, J., Kim, E., Lee, Y., Kim, D., Chang, J., & Choo, J. (2020). Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5801-5810).
[27] Heinonen, M., Rantanen, A., Mielikäinen, T., Kokkonen, J., Kiuru, J., Ketola, R. A., & Rousu, J. (2008). FiD: a software for ab initio structural identification of product ions from tandem mass spectrometric data. *Rapid Communications in Mass Spectrometry: An International Journal Devoted to the Rapid Dissemination of Up-to-the-Minute Research in Mass Spectrometry*, *22*(19), 3043-3052.
[28] Wang, L. T., Hoover, N. E., Porter, E. H., & Zasio, J. J. (1987, October). SSIM: A software levelized compiled-code simulator. In *Proceedings of the 24th ACM/IEEE Design Automation Conference* (pp. 2-8).