Mangan: Assisting Colorization Of Manga Characters Concept Art Using Conditional GAN

Conference Paper *in* Proceedings / ICIP ... International Conference on Image Processing · August 2019

DOI: 10.1109/ICIP.2019.8803667

Felipe Coelho Silva
Georgia Institute of Technology

Paulo André Lima de Castro
Instituto Tecnologico de Aeronautica

Ernesto Marujo
ITA Instituto Tecnológico de Aeronáutica, Brazil

MANGAN: ASSISTING COLORIZATION OF MANGA CHARACTERS CONCEPT ART USING CONDITIONAL GAN

Felipe Coelho Silva[†‡], *Paulo André Lima de Castro*[‡], *Hélio Ricardo Júnior*[†], *Ernesto Cordeiro Marujo*[‡]

[†]Data Science Team, Itaú Unibanco, São Paulo, Brazil
[‡]Aeronautics Institute of Technology (ITA), São José dos Campos, Brazil.

ABSTRACT

Colorization is a challenging task that has recently been tackled by deep learning. Line art colorization is particularly difficult because there is no grayscale value to indicate the color intensities as there is in black-and-white photograph images. When designing a character, concept artists often need to try different color schemes, however, colorization is a time-consuming task. In this article, we propose a semi-automatic framework for colorizing manga concept arts by letting concept artists try different color schemes and obtain colorized results in fashion time. Our approach uses Conditional Generative Adversarial Networks (cGAN) and outperforms current hint-based line-art colorization techniques by providing natural-looking arts with only minor coloring mistakes.

Index Terms— Colorization, Deep Learning, GANs

1. INTRODUCTION

Colorization is a very important process in art creation. In particular, when designing a character, it is common to try different color schemes to see which one is the best fit. This process is very time-consuming and challenging because creating expressive art requires applying texture and shading correctly over the drawing.

Generative Adversarial Networks (GAN) [1] have recently been used in a variety of image-to-image problems successfully [2], including black-and-white image colorization applications [3, 4, 5, 6]. Colorization in Manga Sketches has different properties and challenges when compared to photographs. Sketches do not present information about color intensity in internal regions of the drawing while photographs have different gray-scale values for every pixel in the image which facilitates the problem. While when dealing with photographs the goal is to achieve realistic images, the goal when dealing with line-arts is to achieve natural artistic colorization.

In this scenario, this paper seeks to present a solution to assist in the character design process by allowing designers make quick color experiments with their sketches and observe how the color palettes provided as hints work with the provided sketch.

[1]Any opinions, findings, and conclusions expressed in this manuscript are those of the authors and do not necessarily reflect the views, official policy or position of Itaú-Unibanco.

2. RELATED WORKS

Qu [7] proposed a colorization technique for Mangas that propagates colors over regions based on pattern-continuity. Their approach is successful in propagating static colors throughout regions. Our approach, in contrast, does not require explicit color information for every region type and enables non-static colorization, that is, shading and lighting.

There are successful approaches on Style Transfer for Anime Sketches using U-Net-like architecture and an Auxiliary Generative Adversarial Network [8], however, this kind of approach requires previous colorized art examples which are not available when designing a character from scratch.

Different from the mentioned papers, our approach aims to assist character design from scratch. We used two related works for result comparison. The first one is PaintsChainer [9], a publicly available software that is used for user-guided colorization and is based on deep learning image segmentation to define different parts of an image and then colorize it. The second one is DeepColor [10], which uses Tandem Networks to process the color hint and predict what the color schema should be in the entire image and then colorize it. Our approach is similar to DeepColor, however with a modified architecture and improvements on training techniques and color-hint generation.

3. METHODOLOGY

In this section, we describe the procedure for building the dataset, the Generative Adversarial network architecture that was developed for the application, what techniques were considered when training the architecture and explain how we evaluated the produced results.

3.1. Dataset

No specific dataset for manga character colorization (pairing line arts and colored arts) could be found in the literature and related work datasets were not available for use as well, which demanded the need of building a dataset to test our approach. We have crawled safebooru[1] [11] website, that has plenty of colored manga/anime character arts. After cleaning the dataset by removing duplicates and uncolored images, we ended up with 13,000 images.

3.2. Line-art Extraction

To get the line-art from the colored image, we initially converted the colored image into grayscale, and then applied adaptative thresholding [12], a technique that has been proved to succeed in image segmentation tasks [13]. This approach works in this case because the artist line-arts are supposed to be one of the darkest values among its neighbors. Since the images have different lighting conditions in different areas, using an adaptative thresholding is beneficial to capture the edges.

Fig. 1. Left: Original. Right: Edge Detection Results

Figure 1 shows the technique results. While the produced image is not a real line-art, it is suitable for this application since we would be able to extract similar-looking images from an uncolored example.

3.3. Color Hints

In order to implement a hint-based colorization scheme, we have to provide color-hints alongside the line-arts. It would be impractical to manually provide example hints for each image, so we came up with an automatic solution to aid in training. In this application, color hints are supposed to be an indication of color to be used in a region, that is, just a generic color tip for an image region, and is also supposed to be incomplete, as many regions will not have color hints at all.

To fulfill these requirements, we randomly remove patches throughout the image by replacing the selected regions by white, which is equivalent to no-hint, making the final color hint contain less information, which is beneficial for the training process. Finally, we applied a median blur in the color hint to make the color details fade away and, this way, provide a more generic color hint.

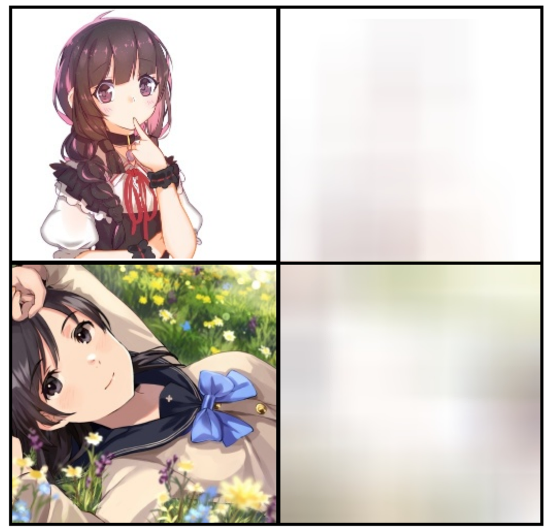

Fig. 2. Left: Original. Right: Automatic Color Hint

The results of this process can be observed on Figure 2. We can see that the resulting color hint barely describes what the image colors should be like. In particular, it could not be used to discover the exact color of small and complex regions. This characteristic is particularly useful for our application domain because we do not want to specify the color to be used in every region, which would be time consuming. At the same time, this also makes the problem harder, since the colorization model will have to paint regions without local color information.

3.4. GAN Architecture

A Generative Adversarial Network [1] is a framework for estimating generative models via an adversarial process. In particular, it is composed by two elements: i) a generative model G, that captures the data distribution from input noise variables (z); ii) a discriminative model D, responsible for estimating what is the probability of a sample to be real or fake. Both models interact as in a two-player minmax game.

Originally, Generative Adversarial Networks (GAN) is an unconditioned generative model, that is, there is no control over modes of the data being generated. However, in our application domain, we are looking to generate colored images

[1]The original art images presented in this work are not owned by the authors. These images were retrieved from http://safebooru.org, where they are properly accredited.

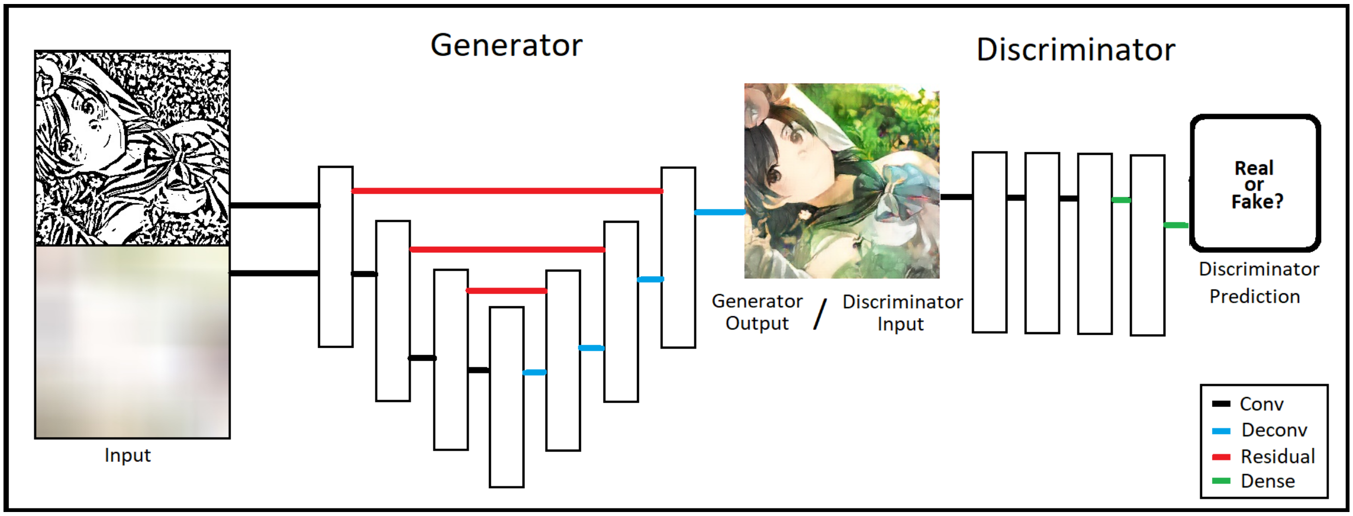

Fig. 3. Representation of our conditional Generative Adversarial Network architecture

that are subject to line-arts and color hints. To incorporate this conditioning, we use Conditional Generative Adversarial Nets (cGAN) [14]. This kind of network was introduced to make the generator learn to produce fake samples following a condition (in our case, the line-arts and color-hints).

We designed our generator using fully convolutional layers followed by deconvolutional layers, as this kind of architecture has seen great success in image-to-image, such as in U-Net [15], with some adaptations described below and in the next section. For the discriminator, we have used a sequence of fully convolutional networks followed by fully connected layers. The discriminator classifies whether an image is real or fake). The described network is represented by Figure 3, and the full implementation and hyperparameters are available in a public code repository [2].

3.5. GAN Training

Additionally, some improved techniques to train GANs have proven to be successful [16], some of which were used in our approach, listed below:

- One-sided Label Smoothing: This technique smooth the positive labels in the GAN process, reducing the vulnerability of neural networks to adversarial examples [17].
- Batch Normalization: As a regularization technique, this also allows the use of much higher learning rates and to be less careful about initialization [18].
- Leaky ReLUs: Using a Leaky Rectified Linear Unit is useful as has the benefits of a regular rectified linear unit and prevent the gradient from not flowing when the function input is lower than zero [19].

[2]Source code available at https://github.com/Lodur03/ManGAN

3.6. Evaluation

In order to evaluate our approach, only images that were not used during training are considered. How good, how natural or how artistic an image looks is subjective and hard to measure quantitatively. In order to capture this subjectivity, we set up an experiment with randomly selected users. In the experiment the user was presented with 4 different versions of an image: the original image and the colored version by DeepColor, PaintsChainer and our (ManGAN) technique, all using the same color hint. The user was instructed to order the images by their preference considering how natural each art looks.

For each user, this procedure was repeated 8 times with different images, randomly selected and without informing the user about the origin of each image. Each time the images were presented to the user they were in different, random, order to alleviate position bias when displaying the images. The forms were filled by 32 different persons and the results will be analyzed in the next section.

4. RESULTS

The survey described in the last section was filled by 32 different persons, and each one evaluated different images. This led to a total of 256 user-created rankings, each one indicating their preference towards the survey images. In particular, we are interested in their impression of our creations, so we decided to analyze how our colorization did when compared to DeepColor, PaintsChainer and the Original colored image.

Table 1 display the survey summary, we can observe that the results produced by our technique were preferred against PaintsChainer and DeepColor results in almost all cases, which indicates that our method outperformed their results in most cases. Additionally, when comparing the images

Fig. 4. Image colorization results by different methods. From left to right: Line-art, Raw color-hint, DeepColor, PaintsChainer, ManGAN (Ours), Original

produced artificially by our method against the original arts, almost every time the original art was preferred.

Table 1. Survey Results of Image Preference

Comparison	ManGAN Preferred
Original	8.98%
DeepColor	88.67%
PaintsChainer	81.64%

Some image generation results are presented by Figure 4. We can see that the results of our technique are richer in details when compared to DeepColor results, however, it still fails in some spots, especially when the region is part of a more complex object or pattern. The results of PaintsChainer are mainly overcolored and present color leakage between regions, whereas our method does not overcolor as much.

In addition, some artifacts can be observed in our results. More specifically, the painted characters often present heterochromia, which could be a characteristic developed in response to the collected dataset, that had a couple of characters with this effect. Moreover, since the edge detection results produce thick lines, the method learned to smooth the lines, making the final art more delicate but also losing some details when they are very small, such as mouths and noses.

Finally, our method was able to guess some colors that fitted the art naturally, even without specifying them in the color hint. This can be particularly observed on the first and last image of Figure 4, in which the ties were colorized with colors not present in the color hint schema.

5. CONCLUSIONS AND FUTURE WORK

The presented methodology proved to be efficient by synthesizing colors, and creating natural-looking colorization from an incomplete and generic user-defined color-hint, despite minor defects. This indicates the success of our hint-generation, architecture and training. Our results were preferred against others in the literature (DeepColor and PaintsChainer), however, there is room for improvements.

As future work, introducing a post-processing user-guided color correction could be useful to make the results more realistic and eliminate artifacts. Another potential improvement spot is on the color hints since we had to generate the color hints for training artificially, the method would probably perform better if there was a database with real user-defined hints, that should be more natural and diverse than our produced hints, for each professionally colorized images.

6. REFERENCES

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017.

[3] Richard Zhang, Phillip Isola, and Alexei A Efros, "Colorful image colorization," in *European Conference on Computer Vision*. Springer, 2016, pp. 649–666.

[4] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros, "Real-time user-guided image colorization with learned deep priors," *ACM Transactions on Graphics (TOG)*, vol. 9, no. 4, 2017.

[5] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi 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 (TOG)*, vol. 35, no. 4, pp. 110, 2016.

[6] Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh, Min Jin Chong, and David Forsyth, "Learning diverse image colorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6837–6845.

[7] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng, "Manga colorization," *ACM Transactions on Graphics (SIGGRAPH 2006 issue)*, vol. 25, no. 3, pp. 1214–1220, July 2006.

[8] Lvmin Zhang, Yi Ji, and Xin Lin, "Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan," *CoRR*, vol. abs/1706.03319, 2017.

[9] Taizan Yonetsuji, "Paintschainer," 2018.

[10] Kevin Frans, "Outline colorization through tandem adversarial networks," *CoRR*, vol. abs/1704.08834, 2017.

[11] "safebooru," 2018.

[12] Derek Bradley and Gerhard Roth, "Adaptive thresholding using the integral image," *Journal of graphics tools*, vol. 12, no. 2, pp. 13–21, 2007.

[13] Payel Roy, Saurab Dutta, Nilanjan Dey, Goutami Dey, Sayan Chakraborty, and Ruben Ray, "Adaptive thresholding: a comparative study," in *Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on*. IEEE, 2014, pp. 1182–1186.

[14] Mehdi Mirza and Simon Osindero, "Conditional generative adversarial nets," 2014.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, p. 234–241, 2015.

[16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[17] David Warde-Farley and Ian Goodfellow, "11 adversarial perturbations of deep neural networks," *Perturbations, Optimization, and Statistics*, p. 311, 2016.

[18] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[19] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, 2013, vol. 30, p. 3.