Information Engineering Express
International Institute of Applied Informatics
2017, Vol.3, No.4, P.55 - 66

Deep Manga Colorization with Color Style Extraction by Conditional Adversarially Learned Inference

Yusuke Kataoka [*] , Takashi Matsubara [†] , Kuniaki Uehara [‡]

Abstract

Many comic books are now published as digital books, which easily provide colored contents compared to physical books. The motivation of automatic colorization of comic books now arises. Previous studies colorize sketches with spatial color annotations or no clues at all. They are expected to reduce workloads of comic artists but still require spatial color annotations in order to produce desirable colorizations. This study introduces color style information and combines it with the conditional adversarially learned inference. Experimental results demonstrate that objects in the manga are painted with colors depending on color style information and that color style information extracted from another colored image to paint an object with the desired color.

1 Introduction

Following the popularization of digital devices, many comic books (*manga* in Japanese) are now published as digital books (e.g., [1]). The digital books can be scanned physical books or bound images originally painted digitally. The digital books easily provide color contents compared to physical books, which have often been printed in monochrome owing to printing cost. However, additional colorization process imposes considerable burdens on comic artists and many digital books are still painted in the style of conventional monochrome comics in expectation of future publications as physical books. The motivation of automatic colorization of comic books arose under such situations.

Colorization of monochrome photos and films continues to be a hot topic in computer vision (see the references [2–4] and therein). In general, monochrome photos contain brightness information, and the colorization methods are required to estimate the remaining information (e.g., hue and saturation). The deep convolutional neural networks (DCNNs) trained with the pixel-wise squared errors estimate only the mode of the ground truths [5] while DCNNs trained with the framework of the generative adversarial networks (GANs) demonstrate diversity in estimated colors [6].

[*] Graduate School of System Informatics, Kobe University, Hyogo, Japan.
[†] Graduate School of System Informatics, Kobe University, Hyogo, Japan.
[‡] Graduate School of System Informatics, Kobe University, Hyogo, Japan.

On the other hand, monochrome comics are just sketches (the outline information) with matte-black and screentones, i.e., they lack even the brightness information. The methods colorizing the sketches should estimate all the three channels of the color space while keeping the outline information. Colorization of the sketches is an image-domain translation rather than colorization of photos. The method based on the framework of the conditional GANs, translating images from a source domain X to a target domain Y (i.e., $G_Y : X \to Y$), called *pix2pix* demonstrated remarkable results in translation of the sketches to photo-realistic images [7]. Following the pix2pix, several studies have proposed the DCNN-based methods colorizing the sketches with spatial color annotations, or no clues at all [8–11].

Owing to the difficulty in obtaining both the sketches and the corresponding colored images, the previous studies employed edge detectors and used the edges extracted from the color images as the sketches for training [7–11]. They colorize the sketches without other clues or with spatial color annotations after having trained with the colored images, the extracted edges, and spatial color annotations. They are expected to reduce workloads of comic artists but still require spatial color annotations for desirable colorizations.

For example, characters who dyed their hair with vivid colors such as pink, red, and green, often appear in mangas.[1] Although their hair colors play very important roles in their identities, their hair colors are grayscaled or even binarized in the style of conventional monochrome comics. Readers find the original colors from limited color pages and cover pages (cover pages are often painted as colored images even in monochrome comics). The original colors are almost unpredictable from the monochrome pages without spatial color annotations. The same is true of any objects appearing in comics such as clothes and furniture.

To free comic artists and editors from specifying additional annotations, this study introduces color style information and proposes the conditional adversarially learned inference (cALI) [12] as an extension of the conditional generative adversarial networks (cGANs) [6, 7, 10, 11]. Unlike cGANs, the proposed network extracts color style information from a colored page and colorizes a monochrome page according to the extracted color style information. Experimental results demonstrate that the objects are painted with colors corresponding to the color style information and that the color style information extracted from a colored image contributes to painting an object with the desired color.

2 Related Work

2.1 Generative Adversarial Networks

Generative adversarial networks [13] is a generative model based on a pair of neural networks. The other neural network called generator G accepts a latent variable $z \in Z$ drawn from a prior distribution $p(z)$ and outputs an artificial data point $x' \in X$; $G : Z \to X$. A neural network called discriminator D accepts a data point, which can be either a real data point x drawn from a dataset X or an artificial data x' generated by the generator G, and outputs a score $d \in \mathbb{R}$; $D : X \to \mathbb{R}$. Using the output scores d, the discriminator D classifies the input data point x into the two

[1] Strictly speaking, they might not dye.

groups; real or artificial. The discriminator D is trained to classify the input data point x correctly and the generator G is trained to trick the discriminator D, i.e., to make the discriminator D classify its artificial data point x' into the real group. As a result of this two-player game, the generator G learns to generate more realistic artificial data points. The typical objective function is based on the cross-entropy for the binary classification:

$$L_{GAN} = \log D(x) + \log(1 - D(G(z))). \tag{1}$$

The output $d = D(x)$ of the discriminator D can be considered as the estimated probability that the given data point x is drawn from the given dataset X. The discriminator D is trained to maximize L_{GAN}, while the generator G is trained to minimize L_{GAN}. The original objective function minimizes the Jensen-Shannon divergence between the set X of the real data points and the set X' of the artificial data generated by the generator G (see the original study [13] for proof). By modifying the objective function, the distance between the sets X and X' can be measured using as other criteria such as Kullback-Leibler divergence and χ^2 distance [14].

The conditional generative adversarial networks (cGANs) is an extension of the GANs [15]. The generator G of a cGANs accepts a condition $m \in M$ as well as the latent variable z and generates an artificial data x' under the condition m; $G : Z \times M \to X$. The discriminator D also classifies the data point x under the condition m; $D : X \times M \to R$. The objective function of cGANs is

$$L_{CGAN} = \log D(x|m) + \log(1 - D(G(z|m)|m)). \tag{2}$$

The pix2pix is a variant of cGANs without the latent variable z. An image in the source domain works as a condition, and the generator G generates the artificial data x' related to the condition m in the target domain.

2.2 Adversarially Learned Inference

In previous studies [16], the latent variable z was inferred from a given data point x by another network E called encoder, which is trained after the training of the GANs; $E : X \to Z$. However, this two-step learning only considers the reconstruction of the latent variable z but no reconstruction of the given data point x. The adversarially learned inference (ALI) realized a simultaneous optimization of the encoder E and the generator G [12]. The generator G is sometimes called a decoder. The discriminator D accepts a pair of artificial data point x and a latent variable z and outputs a score d. The discriminator D evaluates the encoder E through scoring the latent variable z under the condition of the real data point x and evaluates the generator G in a similar way. The following objective functions are used in ALI:

$$L_G = \log(1 - D(x, E(z|x))) + \log D(G(x|z), z), \tag{3}$$

$$L_D = \log D(x, E(z|x)) + \log(1 - D(G(x|z), z)). \tag{4}$$

The generator G and the encoder E are trained to maximize L_G and the discriminator D is trained to maximize L_D.

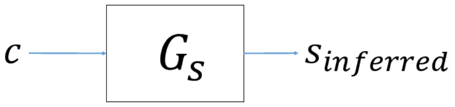

Figure 1: A diagram of the color style network G_s.

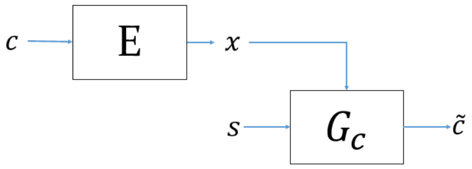

Figure 2: A diagram of the colorization network G_c.

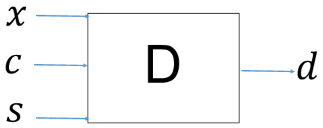

Figure 3: A diagram of the discriminator D.

2.3 Neural Style Transfer

Other notable studies related to the image domain translation are of *image style transfer* [17–19]. In general, these methods receive two images; a content image and a style image. They then transfer the style (e.g., texture) from the style image to the content image while keeping the shapes of objects in the content image intact. One of the advantages of the neural style transfer compared to the pix2pix is that a single model of the neural style transfer can transfer multiple styles.

Our study considers the information about colors of a colored image as an extractable and transferable information like the style of the neural style transfer.

2.4 Comparison between the Proposed Method and Previous Works

The comparison between previous works and the proposed method (cALI) is summarized in Table 1.

Standard GANs [13, 16] generate realistic images from scratch given latent variables, but cannot be applied to colorization of existing images. ALI [12] is an extension of GANs: ALI can infer the latent variables given the real images, but it is also unable to colorize existing images. Conditional GANs (cGANs) [7, 15] are GANs given additional information as the condition. When the additional information is edge images or monochrome images, cGANs can colorize these images instead of

Table 1: Qualitative Comparison between the Proposed Methods and Existing Methods.

Model	Colorization	Random colorization	Extracting color	Adjusting color
GANs [13,16]				
cGANs [7,15]	✓	✓		
Image Style Transfer [17–19]	✓		✓	
ALI [12]				
cALI(Proposed method)	✓	✓	✓	✓

generating images from scratch. These cGANs determine colors automatically and require spatial color annotations to change colors, not reducing workloads of manga artists well.Image Style Transfer [17–19] is not based on adversarial learning and can colorize images by transferring color style (i.e., statistics of colors and edges) of reference color images. Image Style Transfer always requires references and cannot adjust colors.

On the other hand, our proposed method is based on cALI, which can colorize images like cGANs. We consider that the latent variables extracted using ALI correspond to color information of real color images, and thus, our proposed method can transfer color style information like Image Style Transfer. The color style information is expected to follow a prior distribution $p(s)$ and can be drawn from it, unlike Image Style Transfer. Hence, our proposed method colorizes images without any reference images like cGANs and can adjust color style information manually unlike any other competitive methods.

3 Proposed Method

3.1 Conditional Adversarial Learned Inference

In this section, we propose conditional adversarially learned inference (cALI) for colorization of manga: This is an extension of ALI. The conceptual diagrams are depicted in Figs. 1–3. First, we introduce *color style extractor* G_s, which corresponds to the encoder E of the ALI (see Fig. 1). The color style extractor G_s accepts a colored image $c \in C$ and infers its style $s_{inferred} \in S$; $G_s : C \to S$. We expect that the style s to indicate the color style information of the original colored image c. The proposed cALI also has a network called *painter* G_c, which accepts a monochrome image $x \in X$ and a style s and, according to them, it generates an artificial colored image $\tilde{c}$; $G_c : X \times S \to C$ as shown in Fig. 2. The style s can be a sample $s_{sampled}$ from a prior distribution $p(s)$. Ideally, the monochrome image x is a sketch or a draft related to the colored image c. Owing to the difficulty in obtaining such pairs of images, we used the edges extracted from the colored images c as the corresponding monochrome images x. The edge extractor is denoted by E; $E : C \to X$. The discriminator D shown in Fig. 3 evaluates the accepted set of a monochrome image x, a colored image c, and a style s. Then, the edge x works as a condition, and the color style extractor G_s and the painter G_c work as the encoder and the decoder of the original ALI, respectively.

3.2 Training Procedure

We introduce our training procedure in detail in this section. First, a colored image c is sampled from a dataset X and is given to the edge detector E:

$$x = E(c), x \sim X. \tag{5}$$

The colored image c is also given to the color style extractor G_s and its style $s_{inferred}$ is extracted:

$$s_{inferred} = G_s(c). \tag{6}$$

As a result, a pair of the factors c, x, and $s_{inferred}$ is obtained. Next, a style $s_{sampled}$ is sampled from a prior distribution $p(s)$, which is a multivariate standard normal distribution:

$$s_{sampled} \sim p(s). \tag{7}$$

The painter G_c colorizes the edge x according to the sampled style $s_{sampled}$ and generates a colorized image $\tilde{c}_{painted}$:

$$\tilde{c}_{painted} = G_c(x, s_{sampled}). \tag{8}$$

As a result, another pair of the factors $\tilde{c}_{painted}$, x, and $s_{sampled}$ is obtained. The first pair has a real colored image c and a generated style $s_{inferred}$, while the second pair has an artificial colored image $\tilde{c}_{painted}$ and a style $s_{sampled}$ following the prior distribution $p(s)$. According to these pairs, the following objective functions are introduced.

$$L_G = \log(1 - D(x, c, s_{inferred})) + \log D(x, \tilde{c}_{painted}, s_{sampled}), \tag{9}$$

$$L_D = \log(D(x, c, s_{inferred})) + \log(1 - D(x, \tilde{c}_{painted}, s_{sampled})). \tag{10}$$

The color style extractor G_s and the painter G_c are trained to maximize L_G, while the discriminator D is trained to maximize L_D.

3.3 Reconstruction Error

For more robust training, we introduce a pixel-wise reconstruction error. An edge x and a style $s_{inferred}$ were extracted from the same colored image c. This pair is expected to contain all the information to reconstruct the original colored image c. The painter G_c re-colorized the extracted edge x according to the extracted style $s_{inferred}$:

$$s_{inferred} = G_s(c), \tag{11}$$

$$\tilde{c}_{recolored} = G_c(x, s_{inferred}). \tag{12}$$

Then, the pixel-wise root-mean-squared-error is defined as

$$L_{pixelwise} = RMSE(c, \tilde{c}_{recolored}), \tag{13}$$

where the $RMSE$ denotes the pixel-wise root-mean-squared-error. This objective function directly indicates the relationship between the colored image c and the corresponding style $s_{inferred}$.

Finally, the objective function to be minimized for the generators G_s and G_c is

$$\min L = L_{pixelwise} - L_G. \tag{14}$$

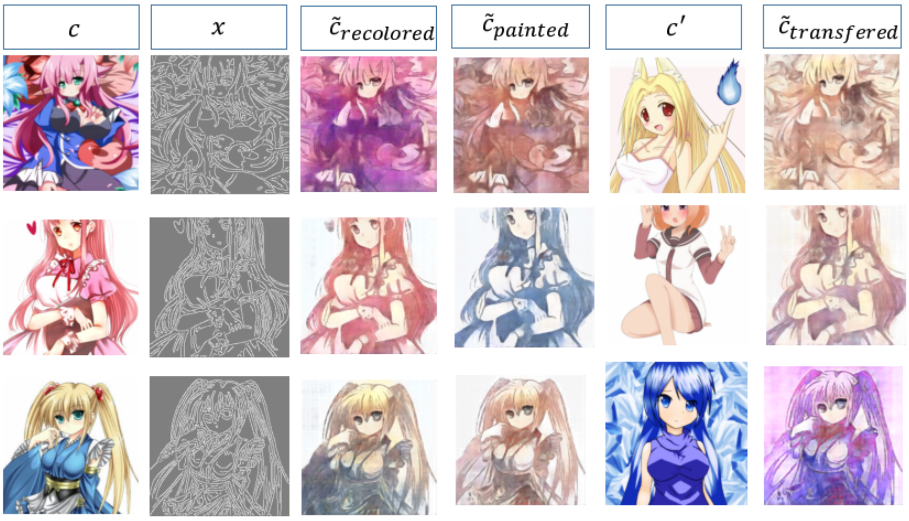

Figure 4: Colored images and their recolored versions.

4 Experiments and Results

4.1 Experimental Setting

Complying their terms of uses, we crawled around 3,000,000 colored illustrations from various social networking services, online communities, and personal websites of manga artists. We do not provide the list of the websites to prevent undesirable influences on them. We used them as the training dataset. We also used Nico-illust dataset[2] as our testing dataset. Each image was resized randomly so that the length of the shortest side was 256–512 pixels and then it was cropped at the random position so that the final size was 256×256 pixels. We used the canny edge detector as the edge detector E. Each test image was examined by the following experiments:

1. colorizing an edge $x = E(c)$ according to the style $s_{inferred}$ extracted from the original colored image c. The result is denoted as $\tilde{c}_{recolored}$.

2. colorizing an edge $x = E(c)$ according to a random style $s_{sampled}$ sampled from the prior distribution $p(s)$. The result is denoted as $\tilde{c}_{painted}$.

3. colorizing an edge $x = E(c)$ according to the style $s_{transfered}$ extracted from another colored image c'. The result is denoted as $\tilde{c}_{transfered}$

Experiment (1) was also performed in the training phase to calculate reconstruction error. This experiment evaluates whether the extracted style $s_{inferred}$ represents color information of the original colored image c. Experiment (2) confirms whether the edge s can be colorized with various colors according to arbitrary color styles

[2] https://nico-opendata.jp/

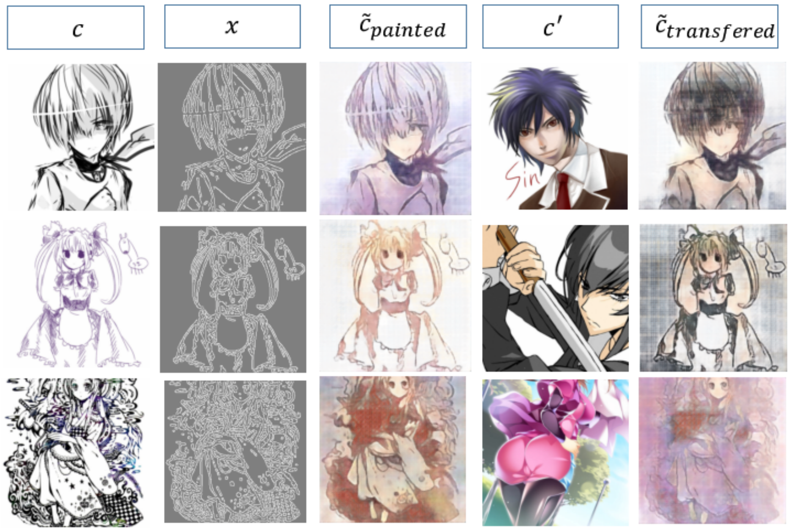

Figure 5: Monochrome images and their colorization.

$s_{sampled}$. Experiment (3) was performed as:

$$s_{transfered} = G_s(c'), \tag{15}$$

$$\tilde{c}_{transfered} = G_c(x, s_{transfered}). \tag{16}$$

The extracted style $s_{transfered}$ is expected to represent the color style of the source colored image c' and to be independent of objects in the source colored images c'. Experiment (3) evaluates whether the extracted style $s_{transfered}$ is applicable to another colored image c universally.

4.2 Recolorization of Colored Images

Figure 4 shows results on re-colorization; the original colored images c, the extracted edges x, the results $\tilde{c}_{colorred}$ on Experiment (1), the results $\tilde{c}_{painted}$ on Experiment (2), the source images c' providing the styles $s_{transfered}$, and the results $\tilde{c}_{transfered}$ on Experiment (3), from left to right respectively. The recolored images $\tilde{c}_{recolored}$ were filled with colors similar to the original colored images c. The images $\tilde{c}_{painted}$ were filled with different colors. At the middle row, the character wears a red dress in the original colored image c and in recolored image $\tilde{c}_{recolored}$ and she wears a blue dress in the painted image $\tilde{c}_{painted}$. The character in the recolored image $\tilde{c}_{recolored}$ has the same skin tone as the one in the original colored image c. Therefore, our proposed approach can change the coloring of style-dependent areas semantically while keeping style-independent areas as it was. The colored images $\tilde{c}_{transfered}$ painted with the styles $s_{transfered}$ extracted from source images c' have similar features to the source images c'; at the middle row, the character in the colored image $\tilde{c}_{transfered}$ has the orange hair and the purple dress just like the source image c'. The extracted styles $s_{transfered}$ contain color style information related to the original colored images c and such information contribute to semantic colorization of other images x.

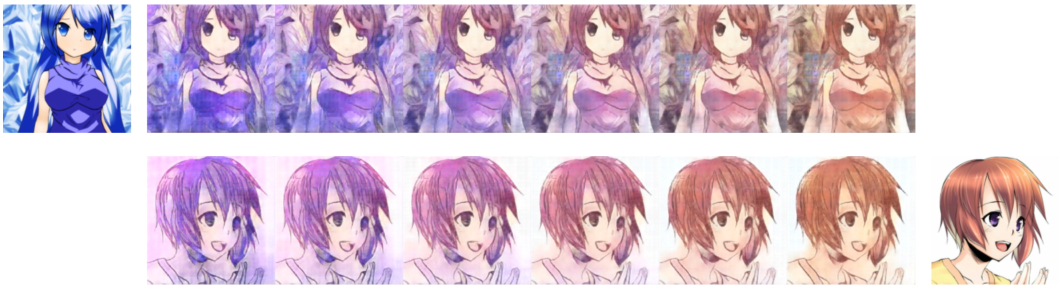

Figure 6: The color transition of colored images.

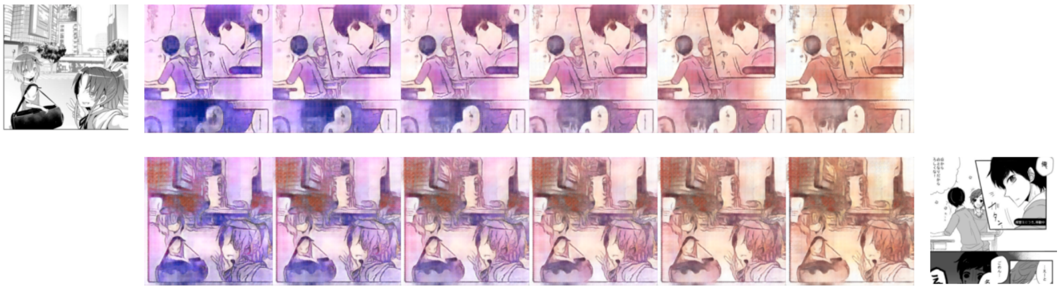

Figure 7: The color transition of monochrome images.

4.3 Colorization of Monochrome Images

Figure 5 shows the results on colorization of monochrome images; the original monochrome images c, the extracted edges x, the results $\tilde{c}_{painted}$ on Experiment (2), the source images c' providing the styles $s_{transfered}$, and the results $\tilde{c}_{transfered}$ on Experiment (3), from the left to the right. At the top row, the original monochrome image c shows a character: His hair has a bright color but its hue and saturation are unknown. With randomly sampled style $s_{sampled}$, his hair was painted with a light purple color (see the third column). When the style $s_{transfered}$ was extracted from the source image c', his hair was painted black as the character in the source image c'. The colorized images $\tilde{c}_{painted}$ and $\tilde{c}_{transfered}$ in the third and fifth columns demonstrate that the monochrome images can also be colorized by our approach.

4.4 Color Transition by Walking Color Spaces

We prepared two colored images c_1 and c_2 shown in the top-left corner and the bottom-right corner of Fig. 6. We extracted their styles s_1 and s_2 and interpolated them linearly; $s_i = \alpha s_1 + (1-\alpha)s_2$ for $\alpha \in [0,1]$. Then, we recolored the edges with the interpolated styles s_i, summarized in Fig. 6: The horizontal axis denotes the direction from the style s_1 to the style s_2 and the vertical axis denotes the content of colored images. The hairs and dresses of the characters are blue on the leftmost column and become brown gradually from the left toward the right while the skin tone is kept as they are. The style s is a continuous variable in the color space and the interpolated style s_1 is on the line connecting the styles s_1 and s_2 in the color space.

We also colorized monochrome images with the same interpolated styles s in Fig. 6, summarized in Fig. 7. As is the case with colored images, the monochrome images are recolored with the styles s interpolating between the styles s_1 and s_2. Unfortunately, when an image contains many objects, our proposed approach cannot colorize all of them appropriately. This is the common issue shared by colorization methods for manga [8–11] and provides an opportunity for future work.

5 Discussion

In general, data-driven methods including deep neural networks require very large data sets (typically data sets composed of more than a million images). It is impossible to collect numerous images of characters relevant to each target character. Moreover, our final goal is to construct a model colorizing general characters, but not a model colorizing a specific character. Hence, we trained the DNNs using a general data set composed of numerous character images, and thus, the DNNs can colorize various character images even when the target is a new character and has no relevant characters. When a character image should be colorized using a known default color set, we can extract the color set from an existing color image as demonstrated in Section 4.

In previous works on image generation [21], Turing test was used to evaluate the quality of the generated images. Salimans and others [22] suggested that the output of a trained DNN to classify natural images can be used to evaluate the generated images. However, we cannot use this methodology to evaluate automatic colorization because our collected images have no class labels. Algorithmic and quantitative evaluation is still an open question.

In general, images contain various types of objects other than persons (e.g., animals, furniture, and buildings). In colorization of natural images, many objects have been colorized successfully [2–4]. However, colorization of manga by our proposed method and other competitive methods (e.g., [10, 11]) is almost limited to the object related to persons (e.g., human bodies, hairs, clothes, etc), as shown in Fig. 7. We consider that this is because the training dataset is relatively small and persons appear in manga extremely frequently compared with other objects. Our future work includes collecting much more images and adjustment of the imbalance in appearances of objects.

6 Conclusion

This study proposed a novel neural network architecture called the conditional adversarially learned inference (cALI). The painter network accepts an image and a style and, according to them, it generates an artificially colored image accordingly. The style used by our painter can be either sampled from a prior distribution or extracted from another image via the color style extractor. The generated images and the extracted styles were evaluated by the discriminator, and the networks were trained like adversarially learned inference. An image colorized with the extracted style inherits that color style.

Future works include end-to-end colorization without edge extraction [20], generation of finer colored images, appropriate colorization of images showing many objects.

Acknowledgments

This study was partially supported by the JSPS KAKENHI (16K12487), and SEI Group CSR Foundation.

References

[1] SHUEISHA, "Jump Book Store." url: http://plus.shonenjump.com/

[2] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 415–423, 2015.

[3] R. Zhang, P. Isola, and A. A. Efros, "Colorful Image Colorizations," in *European Conference on Computer Vision*, 2016.

[4] S. Iizuka, "Let there be Color !: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification," in *SIGGRAPH*, 2016.

[5] A. B. L. Larsen, S. K. Sønderby, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *International Conference on Machine Learning*, 2016.

[6] Y. Cao *et al.*, "Unsupervised Diverse Colorization via Generative Adversarial Networks," *arXiv*, 2017.

[7] P. Isola *et al.*, "Image-to-Image Translation with Conditional Adversarial Networks," *arXiv*, p. 16, 2016.

[8] P. Sangkloy *et al.*, "Scribbler: Controlling Deep Image Synthesis with Sketch and Color," *arXiv*, 2016.

[9] Taizan, "PaintsChainer." url: https://github.com/pfnet/PaintsChainer

[10] K. Frans, "Outline Colorization through Tandem Adversarial Networks," *arXiv*, 2017.

[11] Y. Liu *et al.*, "Auto-painter: Cartoon Image Generation from Sketch by Using Conditional Generative Adversarial Networks," *arXiv*, 2017.

[12] V. Dumoulin *et al.*, "Adversarially Learned Inference," *arXiv*, no. 1606.00704, 2016.

[13] I. J. Goodfellow *et al.*, "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[14] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization," *arXiv*, no. 1606., 2016.

[15] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *arXiv*, p. 1411.1784, 2014.

[16] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” *International Conference on Learning Representations*, 2016.

[17] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image Style Transfer Using Convolutional Neural Networks,” *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.

[18] R. Yin, “Content-Aware Neural Style Transfer,” pp. 1–15, 2016.

[19] L. A. Gatys *et al.*, “Preserving Color in Neural Artistic Style Transfer,” *arXiv*, pp. 1–8, 2016.

[20] J. Y. Zhu *et al.*, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” *arXiv*, 2017.

[21] E. Denton *et al.*, “Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks,” *arXiv*, 2015.

[22] T. Salimans *et al.*, “Improved Techniques for Training GANs,” *arXiv*, 2016.