

PAINTING STYLE-AWARE MANGA COLORIZATION BASED ON GENERATIVE ADVERSARIAL NETWORKS

Yugo Shimizu* Ryosuke Furuta† Delong Ouyang*

Yukinobu Taniguchi* Ryota Hinami ‡ Shonosuke Ishiwatari‡

* Tokyo University of Science † The University of Tokyo ‡ Mantra Inc.

ABSTRACT

Japanese comics (called manga) are traditionally created in monochrome format. In recent years, in addition to monochrome comics, full color comics, a more attractive medium, have appeared. Unfortunately, color comics require manual colorization, which incurs high labor costs. Although automatic colorization methods have been recently proposed, most of them are designed for illustrations, not for comics. Unlike illustrations, since comics are composed of many consecutive images, the painting style must be consistent.

To realize consistent colorization, we propose here a semi-automatic colorization method based on generative adversarial networks (GAN); the method learns the painting style of a specific comic from small amount of training data. The proposed method takes a pair of a screen tone image and a flat colored image as input, and outputs a colored image. Experiments show that the proposed method achieves better performance than the existing alternatives.

Index Terms— GAN, Colorization, Comics, Manga

1. INTRODUCTION

Japanese comics, also known as manga, are read not only in Japan, but around the world, and their popularity continues to grow. Although they have been traditionally drawn and sold in monochrome format with screen tones, full color comics are now being sold in because they attract a wider range of customers. However, the colorization process is time-consuming and costly, which is a huge problem.

In order to solve the above problem, many automatic colorization methods based on machine learning have been proposed for illustrations. However, their results are not pleasant enough for practical use. In addition, they have two problems: (i) none of them pay attention to consistency in painting style between images on different pages, and (ii) most of them require a large amount of training data. Failure of a colorized comic to maintain consistency in painting style across the pages degrades the reader experience and enjoyment. A large amounts of training data are difficult to obtain in many cases due to copyright issues.

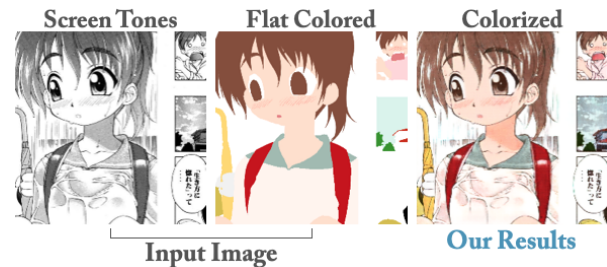


Fig. 1. Input and output images of our method.

In this paper, we propose a painting style-aware colorization method based on GANs. The proposed method takes a pair of a screen tone image and a flat colored image as input, and outputs a colored image. Because the shadow and color information are given by the screen tone and flat colored image, respectively, the generator does not need to learn or infer those for the objects in the input image. Therefore, the proposed method (i) focuses on and learns only the painting style in the training data (ii) and can be trained with small amounts of data. We use flat colored images as input because they are easy to create and contain a wealth of information as hints.

We conducted experiments on the comics in the Manga109 dataset [1] [2] and commercially available comics. The experimental results show that the proposed method can efficiently learn the painting style from a small amount of data and achieve better performance than other methods.

2. RELATED WORKS

Various colorization methods have been already proposed, for not only illustrations or manga but also natural images, as is well summarized in [3]. This paper focuses on works that relevant to illustrations and manga.

2.1. Illustration colorization

Existing colorization methods are categorized into two classes: those based on hand-crafted algorithms such as [4] [5],

and [6] and machine learning. We focus on the latter because the proposed method employs machine learning.

In order to create the dataset, in general, it is necessary to prepare a large number of colored images and their corresponding line drawings. Liu et al. [7] create their line drawings by applying a XDoG filter to the colored images. However, their method requires manual adjustments, and the painting styles output are not close to the ground truth images although their method can colorize specific regions with the correct color.

Zhang et al. [8] proposed a method based on Pix2Pix, where user hints are added in addition to line drawings as their input. They use the danbooru dataset [9] as their training data. Although their method can successfully learn the colorization for illustrations, directly adopting their method to comics has several drawbacks. In the colorization of comics, each comic has a different painting style. However, because model training uses a large number of colored images from different artists, it cannot learn the painting style of a particular comic. In addition, unlike our method, which uses screen tone images as input, the position of shadows cannot be specified in their method.

Recently, Ren et al. [10] proposed a two-stage sketch colorization method, and Akita et al. [11] proposed a colorization method that uses a reference image. Zou et al. [12] proposed a language-based colorization method, where users can control colorization by using text descriptions as input. Different from the above methods, as discussed in Section 1, we propose a style-aware colorization method for manga by taking a pair of a screen tone image and a flat colored image as input.

Alternatives include some attempts to base illustration on machine learning [13], [14], [15], and [16]. Different from ours, their target is learning for lighting effects or shadowing, not colorization.

2.2. Manga colorization

Hensman and Aizawa [17] proposed an automatic manga colorization method. In this method, the conditional GAN (cGAN) is trained online using a single reference image. This method has the advantage that a large amount of training data is not needed. However, it is sometimes difficult to obtain reference images similar to the target image (e.g., some characters appear only a few times in a comic).

Xie et al.’s method [18] enables the bidirectional translation between screen tone images and colored images. However, this method requires a large amount of training data, which is difficult to obtain in most cases.

Silva et al. [19] proposed a hint-based semi-automatic manga colorization method that uses cGAN as its model. The difference between ManGAN and our method is that ManGAN takes a pair of line-art and color-hint images as input, whereas ours takes a pair of screentones and flat-colored images. Moreover, the network architectures are different.

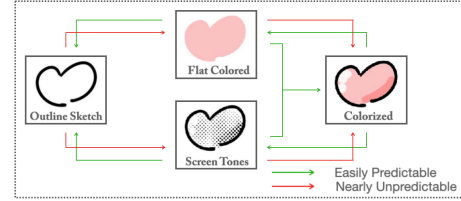


Fig. 2. Name and Relationships between manga states.

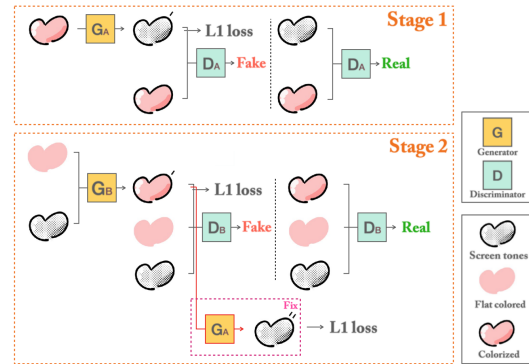


Fig. 3. Structure of our proposed method.

3. PROPOSED METHOD

3.1. Observations

Before detailing the proposed method, we discuss the states of manga images and their relationships. Fig. 2 shows the name of each state and the relationships between them. The green arrows indicate the conversions between the states that are easy to predict, while the red arrows indicate the nearly unpredictable conversions (due to insufficient information). Because an outline sketch does not contain shading or color information, it is difficult to determine the direction of the light and the appropriate color uniquely (i.e., it is difficult to predict the colorized image from the outline sketch as shown in Fig. 2). Zhang et al. [8] solved this problem by learning from a large amount of training data. However, such training data is not always available. This is especially true when we want to learn the painting style of a particular colorist because it is difficult to create a large-scale dataset that contains only the artworks drawn by the particular person. Similarly, due to lack of information, to predict the colored image from either flat colored or screen tone image is not ideal. As shown in Fig. 2, the direction of the light and the appropriate color can be uniquely determined by using both flat colored and screen tone images, which makes it possible to predict the colorized image. Therefore, in this paper, we propose a method that predicts the colorized image by using both flat colored and screen tone images as input.

Because the screen tone images represent the monochrome manga itself, it incurs no additional costs to obtain the screen

tone images. The user only needs to prepare the flat colored images. Compared to the actual coloring process, preparing flat color images requires no special skills and much less time and effort, and thus anyone (not just colorists) can perform the task.

3.2. Framework

The training process is shown in Fig. 3.

The proposed method has two generators. One converts a colorized image into a screen tone image, while the other generates a colorized image from a pair of flat colored and screen tone images. These two generators are trained separately. The task of converting a colorized image to a screen tone image is called “stage 1”, and the generation of a colorized image from a pair of flat colored and screen tone images is called “stage 2”.

The training data consists of the triplets of colorized, screen tones, and flat colored images (x, y, z). First, in “stage 1”, generator G_A learns how to generate screen tone images from colorized images. This process removes color information from colorized images and predicts the position and patterns of the corresponding screen tones. As shown in Fig. 2, colorized images contain sufficient information to predict screen tone images. The training procedure follows that of Pix2Pix [20].

3.2.1. training stage 1

We adopted the UNet architecture for generator G_A . Let x denote a colorized image and y denote a screen tone image. The discriminative loss of G_A is expressed as:

$$\mathcal{L}_{G_A}(G_A, D_A) = E_{x,y}[\log D_A(x, y)] + E_x[\log(1 - D_A(x, G_A(x)))], \quad (1)$$

where generator G_A learns how to fool the discriminator D_A . Conversely, discriminator D_A learns to discriminate between fake and real tuples.

In addition to the discriminative loss in Eq. 1, we use L1 loss between the ground truth screen tone image y and the generated image $G_A(x)$. Because pix2pix uses L1 loss in order to suppress blurring, we used the same L1 loss term in the proposed method.:

$$\mathcal{L}_{L_1}(G_A) = E_{x,y}[\|y - G_A(x)\|_1]. \quad (2)$$

Our final objective of G_A is:

$$G_A^* = \arg \min_{G_A} \max_{D_A} \mathcal{L}_{G_A}(G_A, D_A) + \lambda_1 \mathcal{L}_{L_1}(G_A). \quad (3)$$

3.2.2. training stage 2

After completion of “stage 1” training, we move on to “stage 2”. In “stage 2”, the input is a pair of flat colored and screen

tone images. The generator G_B learns how to generate colorized images from flat colored and screen tone images. The generator model is an extension of “U-Net” [21]. In order to gain one output from two inputs, the model has a two stream structure.

Let x, y , and z be a colorized image, a screen tone image, and a flat colored image, respectively. The discriminative loss of G_B is expressed as:

$$\mathcal{L}_{G_B}(G_B, D_B) = E_{x,y,z}[\log D_B(y, z, x)] + E_{y,z}[\log(1 - D_B(y, z, G_B(y, z)))], \quad (4)$$

where the generator G_B learns to fool the discriminator D_B . In contrast, the discriminator D_B learns to classify between the fake and real triplets. We also use L_1 loss to increase the quality of the output:

$$\mathcal{L}_{L_1}(G_B) = E_{x,y,z}[\|x - G_B(y, z)\|_1], \quad (5)$$

Furthermore, in order to retain cycle consistency[22], the colorized image generated by G_B is input to the fixed G_A . The L_1 distance between the fake screen tone image from the fixed G_A and the ground truth screen tone image is calculated (the second term in Eq. 6). This idea, based on [23] [22], contributes to increasing the quality of the output. Our final objective in stage 2 is:

$$G_B^* = \arg \min_{G_B} \max_{D_B} \mathcal{L}_{G_B}(G_B, D_B) + \lambda_2 \mathcal{L}_{L_1}(G_A) + \lambda_3 \mathcal{L}_{L_1}(G_B). \quad (6)$$

3.2.3. Testing

At the time of inference, we can obtain the colorized image by inputting a pair of screen tone image and its corresponding flat colored image into generator G_B .

3.3. Discussions

Because flat colored images and screen tone images contain color and shading information, respectively, there is no need to predict the color and shading of the objects in the input. The proposed model can focus on and learn only the painting style in the training data. Therefore, our method can be trained with only a small amount of data and can learn the painting style of a particular colorist.

4. EXPERIMENTS

4.1. Dataset

Manga109: Manga109 [1] [2] is a dataset that consists of 109 volumes of Japanese manga drawn by professional manga artists. We randomly selected 10 pages from manga “Nekodama ©Ebifly”. We used five pages as the training data, and the remaining pages as the test data. Images in the training



Fig. 4. Ablation study of the proposed method.

data were colorized and flat colored manually by a proficient colorist using CLIP STUDIO PAINT EX.

Commercial dataset A: We used “Sgt. Frog”, a commercially available comic. Because the colorized data is available only for the cover pages, we used them as ground truth. We generated screen tone images from those cover pages by passing them through the LT filter. The flat coloring process was done manually. We used 30 cover pages from “Sgt. Frog”, and used the first ten pages as the training data.

Commercial dataset B: We also used commercial comics available in both monochrome and colored versions. In this dataset, because we have both colorized and screen tone images, we prepared only the flat colored images manually. We randomly selected 10 pages from “Gotobun no Hanayome”, and used the first two pages as the training data.

We report only the quantitative results in the commercial datasets A/B due to copyright issues.

4.2. Compared methods

We compare the proposed method with the following four methods in terms of performance.

(i).Pix2Pix (Screen Tones): We trained Pix2Pix, which takes a screen tone image as input and outputs a colored image.

(ii).Pix2Pix (Flat Colored): We trained another Pix2Pix model. Different from (i), this model takes a flat colored image as input.

(iii).cGAN-Based Manga Colorization: This method requires a reference image. For details, see [17].

(iv).Two-Stage Sketch Colorization [8]: We used a third party implementation of [8]¹. This model was trained on the danbooru dataset [9].

(i) and (ii) were trained on the same training sets as the proposed method.

4.3. Results

Fig. 4 shows the ablation results of the proposed method by comparing the results w/ and w/o stage 1 and G_A in stage 2 (see Fig. 3). We observe that by incorporating them, the

¹<https://github.com/adamz799/Paints>



Fig. 5. Qualitative comparisons.

Table 1. PSNR on Commercial datasets A and B.

		(i) [20]	(ii) [20]	(iv) [8]	Ours
dataset A	Ave.	12.99	15.22	15.19	26.71
	Max	15.27	19.28	17.46	27.38
	Min	8.47	9.86	13.68	25.71
dataset B	Ave.	17.95	16.17	13.26	24.47
	Max	20.64	18.47	16.66	27.02
	Min	14.39	14.28	11.00	21.73

proposed method successfully learned the painting style of the ground truth image created by the colorist.

Fig. 5 shows qualitative comparisons of the proposed method and the other methods. Compared to cGAN-based method [17] and Two-stage method [8], the proposed method yielded much more pleasing results. We observe that Pix2Pix (Screen Tones) and Pix2Pix (Flat Colored) failed to predict the colors and shadow, respectively, due to the small amount of training data, and the insufficient information in their inputs. In contrast, the proposed successfully learned the painting style by taking both screen tones and flat colored images as input.

Table 1 shows the comparisons of PSNR on the commercial datasets A and B. We observe that our method achieved significantly better results than the alternative methods.

5. CONCLUSION

In this paper, we proposed a style-aware colorization method for manga. The proposed method can learn unique painting styles from a small amount of training data by taking a pair of screen tones and flat colored images as input. The outputs of our method exhibit very high visual quality because the generator efficiently learns the painting style from a particular colorist. Because the flat coloring process does not require any painting skill, anyone can perform this task, which means our method can replace the traditional cumbersome steps.

As a practical workflow, the colorist only needs to colorize the first few pages manually. We use these images as the training data for the proposed method. The method automatically colorizes the remaining pages by preparing the corresponding flat colored images and inputting them to trained generator G_B . The proposed workflow makes it possible to reduce time and money overheads.

In future work, we will tackle automatic generation of flat colored images, which makes our method more practical.

6. REFERENCES

- [1] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa, “Sketch-based manga retrieval using manga109 dataset,” *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21811–21838, 2017.
- [2] Kiyoharu Aizawa, Azuma Fujimoto, Atsushi Otsubo, Toru Ogawa, Yusuke Matsui, Koki Tsubota, and Hikaru Ikuta, “Building a manga dataset “manga109” with annotations for multimedia applications,” *IEEE MultiMedia*, vol. 27, no. 2, pp. 8–18, 2020.
- [3] Saeed Anwar, Muhammad Tahir, Chongyi Li, Ajmal Mian, Fahad Shahbaz Khan, and Abdul Wahab Muzaffar, “Image colorization: a survey and dataset,” *arXiv e-prints*, p. arXiv:2008.10774, Aug. 2020.
- [4] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng, “Manga colorization,” *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 1214–1220, 2006.
- [5] Daniel Sýkora, John Dingliana, and Steven Collins, “Lazybrush: Flexible painting tool for hand-drawn cartoons,” *Computer Graphics Forum*, vol. 28, no. 2, pp. 599–608, 2009.
- [6] Bin Bao and Hongbo Fu, “Scribble-based colorization for creating smooth-shaded vector graphics,” *Computers and Graphics*, vol. 81, pp. 73–81, 2019.
- [7] Yifan Liu, Zengchang Qin, Zhenbo Luo, and Hua Wang, “Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks,” *Neurocomputing*, vol. 311, pp. 78–87, 2018.
- [8] Lvmin Zhang, Chengze Li, Tien-Tsin Wong, Yi Ji, and Chunping Liu, “Two-stage sketch colorization,” *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 261:1–261:14, 2018.
- [9] “Danbooru2017:a large-scale crowdsourced and tagged anime illustration dataset,” DanbooruCommunity, 2018.
- [10] Hui Ren, Jia Li, and Nan Gao, “Two-stage sketch colorization with color parsing,” *IEEE Access*, vol. 8, pp. 44599–44610, 2020.
- [11] Akita Kenta, Morimoto Yuki, and Tsuruno Reiji, “Colorization of line drawings with empty pupils,” *Computer Graphics Forum*, vol. 39, no. 7, pp. 601–610, 2020.
- [12] Zou Changqing, Mo Haoran, Gao Chengying, Du Ruofei, and Fu Hongbo, “Language-based colorization of scene sketches,” *ACM Transactions on Graphics*, vol. 38, no. 6, pp. 233:1–233:16, 2019.
- [13] Matis Hudon, Mairead Grogan, Rafael Pages, and Aljosa Smolic, “Deep normal estimation for automatic shading of hand-drawn characters,” in *ECCV Workshops*, 2018.
- [14] Wanchao Su, D. Du, X. Yang, Shizhe Zhou, and Hongbo Fu, “Interactive sketch-based normal map generation with deep neural networks,” in *ACM on Computer Graphics and Interactive Techniques*, 2018, vol. 1, pp. 22:1–22:17.
- [15] Lvmin Zhang, Edgar Simo-Serra, Yi Ji, and Chunping Liu, “Generating digital painting lighting effects via rgb-space geometry,” *ACM Transactions on Graphics*, vol. 39, no. 2, 2020.
- [16] Qingyuan Zheng, Zhuoru Li, and Adam Bargteil, “Learning to shadow hand-drawn sketches,” in *CVPR*, 2020.
- [17] Hensman Paulina and Aizawa Kiyoharu, “cGan-based manga colorization using a single training image,” in *ICDRA Workshops*, 2017.
- [18] Minshan Xie, Chengze Li, Xueting Liu, and Tien-Tsin Wong, “Manga filling style conversion with screentone variational autoencoder,” *ACM Transactions on Graphics*, vol. 39, no. 6, pp. 226:1–226:15, 2020.
- [19] Felipe Coelho Silva, Paulo André Lima de Castro, Hélio Ricardo Júnior, and Ernesto Cordeiro Marujo, “Mangan: Assisting colorization of manga characters concept art using conditional gan,” in *ICIP*, 2019, pp. 3257–3261.
- [20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, “Image-to-image translation with conditional adversarial networks,” in *CVPR*, 2017, pp. 5967–5976.
- [21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *MICCAI*, 2015, vol. 9351, pp. 234–241.
- [22] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *ICCV*, 2017, pp. 2242–2251.
- [23] Taeksoo Kim, Moon-su Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim, “Learning to discover cross-domain relations with generative adversarial networks,” in *ICML*, 2017, vol. 70, pp. 1857–1865.
- [24] Paszke et al., “Pytorch: An imperative style, high-performance deep learning library,” in *NeurIPS*, 2019, pp. 8024–8035.

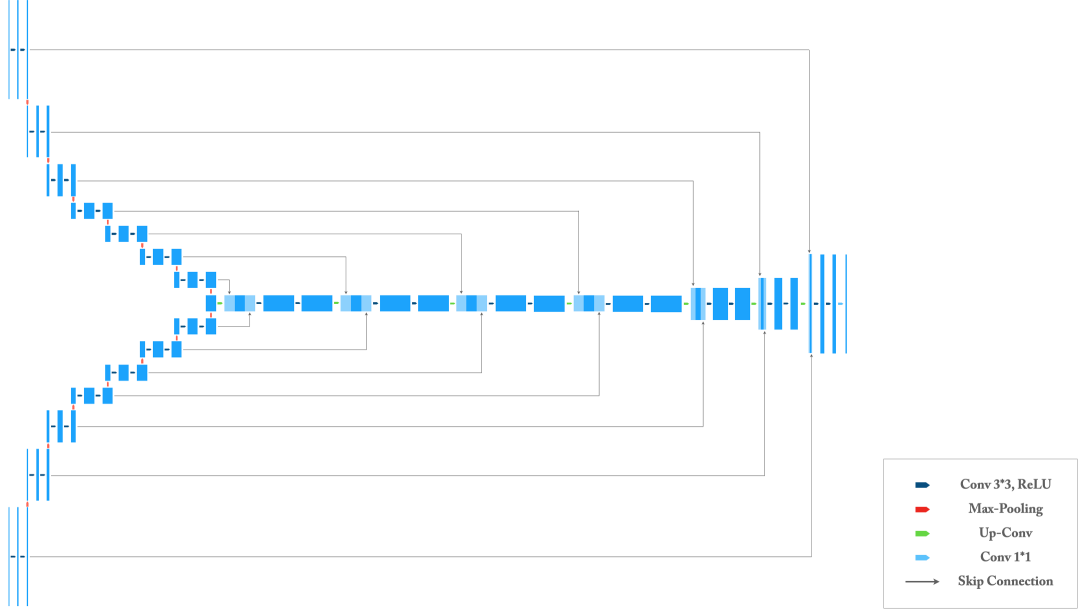


Fig. 6. Network structure of generator G_B .



Fig. 7. Images from Manga109 colored by our method.

7. SUPPLEMENTAL MATERIAL

7.1. Implementation details

The generator model is an extension of “U-Net” [21] as shown in Fig. 6. In order to gain one output from two inputs, the model has a two stream structure. Each input image is pro-

Table 2. Time required for manually preparing flat colored images and colorized images.

	Flat colored (<i>min</i>)	Colorized (<i>min</i>)
Illustrator A	73	148
Illustrator B	80	172
Illustrator C	160	320
Illustrator D	75	180
Average	97	205

cessed by different encoders, and the feature maps are extracted by convolution. Then, the feature maps from the two encoders are concatenated and convolved. The resulting feature map is used as the input of the decoder. Similar to U-Net, this model also has skip connections between some corresponding layers in the encoder and the decoder.

We implemented the proposed method using the PyTorch library [24]. We used Adam optimizer and set the learning rate to 0.001. The number of training epochs was 150, 100, and 100 for the Manga109 dataset, commercial dataset A, and B, respectively. Each page was resized to 1024×724 and 640×453 , and then randomly cropped to 256×256 during the training. We set $(\lambda_1, \lambda_2, \lambda_3)$ as (100, 50, 50).

We use the same notation as [20]. Ck denotes a Convolution-BatchNorm-ReLU layer with k filters, and CDk denotes a Convolution-BatchNorm-Dropout-ReLU layer with a dropout rate of 0.5. The encoder architecture of generator G_A is C64-C128-C256-C512-C512-C512-C512, and their decoder architecture is CD512-CD512-CD512-CD512-CD256-CD128-CD64. All convolutions are 3×3 spatial filters applied with

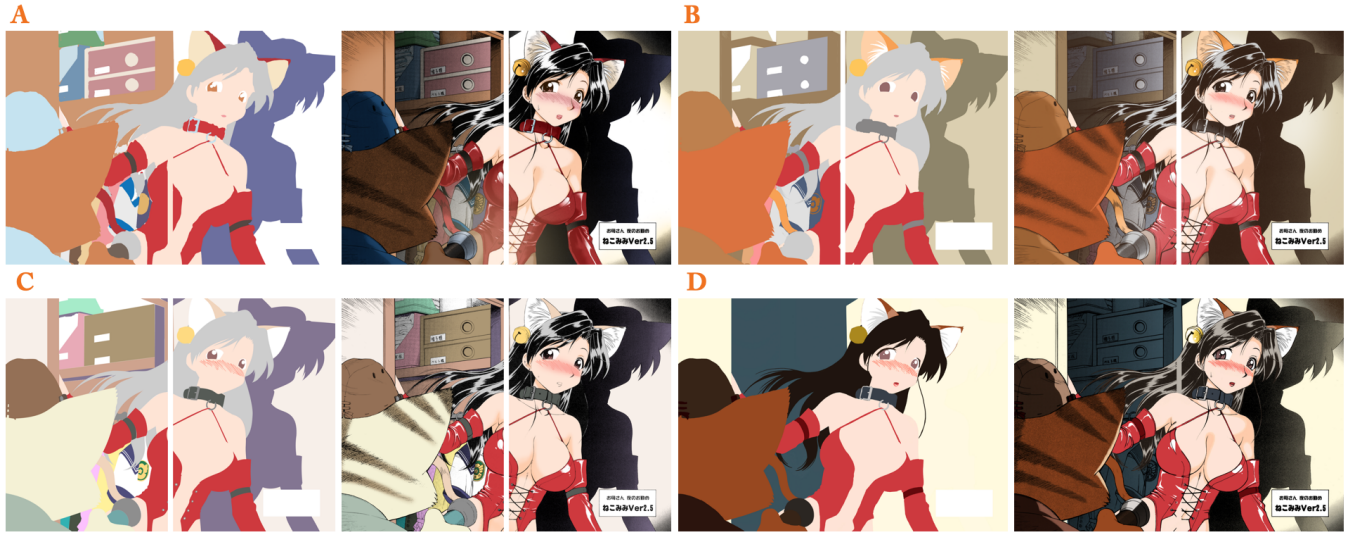


Fig. 8. Images colored by four professional illustrators.

stride 1. Generator G_B has the same architecture as G_A except that it has two stream encoders. Similar to [20], there are some exceptions with regard to BatchNorm and ReLU in some layers. The first C64 layer does not have BatchNorm, and all ReLUs in the encoder are leaky with slope 0.2 while ReLUs in the decoder are not leaky.

The discriminator architecture is C64-C128-C128. Similar to the original paper of Pix2Pix [20], after the last layer, a convolution is applied to map to a 1-dimensional output, followed by a Sigmoid function. The first C64 layer does not have BatchNorm, and all ReLUs are leaky with slope 0.2.

7.2. Additional results

7.2.1. Qualitative results

Fig. 7 shows the input images and the output results of the proposed method on the Manga109 dataset. We observe that the proposed method yielded high quality results.

7.2.2. Comparison of time for manual colorization

Although the proposed method requires flat colored images as input, they are easier to create than colored images as mentioned in Sec. 1. In order to ensure this premise, we conducted the user study to measure the time required for manually preparing flat colored images and colored ones, respectively. We asked four professional illustrators to manually create them from a screen tone image. We used a page randomly chosen from “Nekodama ©Ebify”. Table 2 and Fig. 8 show the time and result images, respectively. We observe that flat coloring process can save 53% of time on average.