Deep-Eyes: Fully Automatic Anime Character Colorization with Painting of Details on Empty Pupils

K. Akita, Y. Morimoto, and R. Tsuruno

Kyushu University

Abstract

Many studies have recently applied deep learning to the automatic colorization of line drawings. However, it is difficult to paint empty pupils using existing methods because the networks are trained with pupils that have edges, which are generated from color images using image processing. Most actual line drawings have empty pupils that artists must paint in. In this paper, we propose a novel network model that transfers the pupil details in a reference color image to input line drawings with empty pupils. We also propose a method for accurately and automatically coloring eyes. In this method, eye patches are extracted from a reference color image and automatically added to an input line drawing as color hints using our eye position estimation network.

CCS Concepts

• ***Computing methodologies*** → *Image processing;* • ***Applied computing*** → *Fine arts;*

1. Introduction

The colorization of illustrations is a very time-consuming process, and thus many automatic line drawing colorization methods based on deep learning have recently been proposed. For example, Ci et al.'s method [CMW*18], petalica paint [Yon17], and Style2Paints [ZJL*18, lll18] are semi-automatic colorization methods for intuitively colorizing line drawings based on color scribbles or dots input by users. These methods enable colorization accurately and in detail by inputting color scribbles or dots in regions that users want to colorize. Tag2Pix [KJPY19] is a colorization method that allows users to specify regions and colors using natural language, such as "brown hair". Drawings are finely colorized using this method except for empty pupils. Moreover, with this method, regions that have not been previously defined cannot be specified, making it difficult to color the left and right eyes with different colors.

Input can also be in the form of a color reference image, where methods apply the colors in a reference image to an input line drawing image [LZL17, FHOO17]. Zhang et al. proposed a colorization method [LZL17] that applies pre-trained VGG16/19 [SZ14] to extract features from a color reference image and inputs them into the middle layer of the network. Comicolorization [FHOO17] is a colorization system for manga images based on color reduction on a reference color image. The above methods train the network using line drawings with edges in the pupils and thus cannot paint details in empty pupils (Figs. 1 (a)-(c),(e),(f)).

The colorization method proposed by Huang et al. [HLK19] considers the semantic correspondence between the input image and a color reference image. The method can finely color line drawings, including illustrations, and paint pupil details by pasting local regions of the reference image to the corresponding regions in the input image as color hints. However, the results of this method show the pupil details of the input line drawing. Thus, the method cannot transfer pupil details from the reference image.

Here, we aim to support the colorization of illustrations in practical settings by colorizing empty pupils. The proposed system paints the details in empty pupils during line drawing colorization, making it practical for illustration support.

Style transfer is a technique that transfers the style of a reference image to an input image; it can generate the details in the reference image. The deep style transfer method [LAGB16] uses convolutional neural networks to transfer mainly painting styles to photographs. However, this method is unsuitable for coloring line drawings of characters (Fig. 1 (d)). TextureGAN [XSA*18] is a method that applies textures to regions in an input line drawing. However, it is designed for repeated patterns, making it unsuitable for regions with unique patterns. Moreover, it requires users to specify the texture for each region. The above methods cannot create semantic detail patterns in an input image.

Image completion using deep learning can create semantic details not included in an input image [GL19]. However, most such methods do not allow users to select the content for the target region. In contrast, our system allows users to specify styles using a reference image. Our main contributions can be summarized as follows:

DOI: 10.2312/egs.20201023

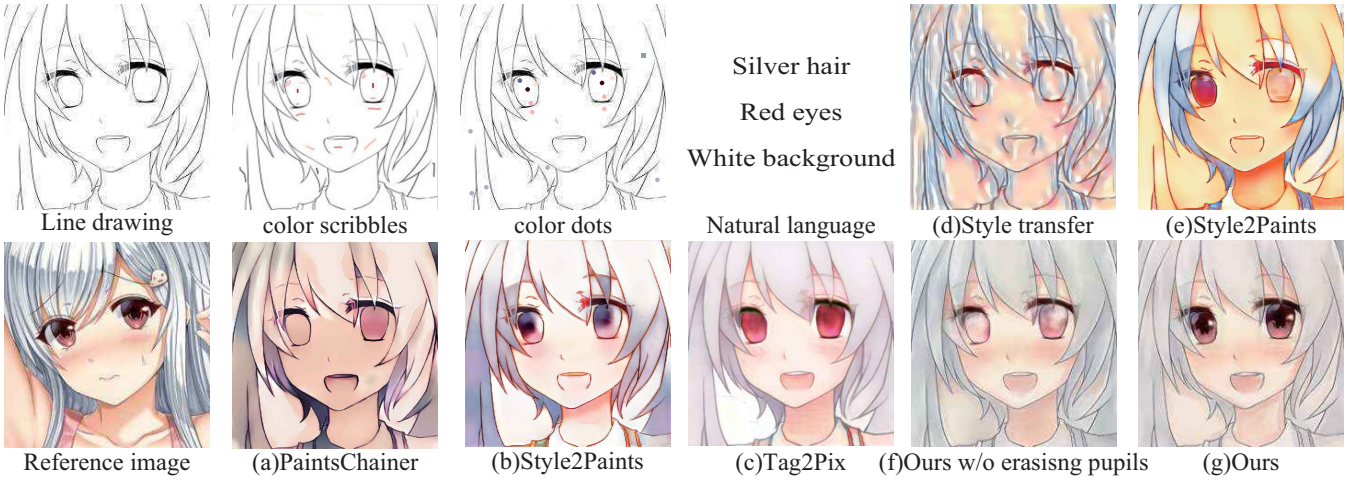


Figure 1: *Comparison between existing colorization methods [Yon17, lll18, KJPY19, LAGB16] and proposed method. The color reference inputs are (a) color scribbles, (b) color dots, (c) natural language, and (d, e, f, g) color reference images. All output images are colorized based on the reference color image. The existing methods roughly colorize the line drawing but cannot express pupil details. The proposed method accurately colorizes the line drawing and generates pupil details based on our novel dataset of line drawings with empty pupils.*

- We combine a colorization network with an eye position estimation network to automatically and accurately color eyes.
- We propose a method for creating a novel dataset for generating pupil details.

2. Overview

Figure 2 shows an overview of the proposed method. In our method, a line drawing and a color reference image (facial illustration) are given as inputs. The eye positions in the input images are first predicted by the eye position estimation network. A hint image is generated from these eye positions using image processing. Then, the hint image, input drawing, and reference image are input to the colorization network. The colorized image is the final output. We train the eye position estimation network with line drawings as the input and the eye positions as the output (see Sec. 4.1). Our colorization network is trained with line drawings and the corresponding hint images as the input and the corresponding color images as the output (see Sec. 4.2). In these line drawing datasets, regions around the eyes are erased. See Sec. 3 for our data creation method.

3. Data Creation

Here, we describe the creation of the datasets for the eye position estimation network and the colorization network. To train the eye position estimation and colorization networks, about 2.5k and 370k facial images were respectively cropped from videos [nic] and illustrations [AcBG19] for each network. To avoid low-resolution images, images with dimensions of at least 256 × 256 pixels were detected by a facial detector [Nag11]. Those larger than 256 × 256 pixels were cropped and resized to 256 × 256 pixels. To create result images of all figures, we used danbooru2018 [AcBG19] and Nico-illust Dataset [IOO16].

3.1. Line Drawing Extraction

We apply datasets of line drawings extracted from color images to train the two networks. To prevent the colorization network from overfitting to the training data [KJPY19], two kinds of line extraction are randomly applied, namely a morphological operation [Yon17] and XDoG [HWO12]. For the morphological operation, a color image is first converted to grayscale. Next, morphological processing is applied to make black lines thin. Then, the dilated image is subtracted from the grayscale image to obtain white contours. Then, the image is inverted to obtain the line drawing. XDoG is used to extract a binarized line image. To estimate the eye positions in an input color reference image, we apply XDoG to extract the line drawing (Fig. 2).

3.2. Dataset Creation for Eye Position Estimation Network

The eye position estimation network predicts the eye positions in the input line drawing and the color reference image. Here, the input line drawing has no edges in the pupils whereas the line drawing generated from the reference image has edges. This network works for cases with or without pupil details. The network is trained using line drawings with and without pupil edges. We create a dataset of line drawings without pupil edges because it is difficult to collect large quantities of illustrations without pupil details. We create line drawings with pseudo empty pupils by filling areas around eye positions in extracted line drawings with white ellipses. The height and width of the ellipses respectively range from 15 to 30 pixels and from 8 to 30 pixels. The color image dataset for this network is transformed to line drawings, 80% of which are transformed to pseudo empty pupils in the above processing. The mixed line drawing dataset comprises images with and without the pseudo empty pupils. This dataset is processed as below to increase estimation accuracy. Half of the images are rotated by a randomly selected angle in the range of -30 to 30 degrees in intervals of 5 degrees. Then,

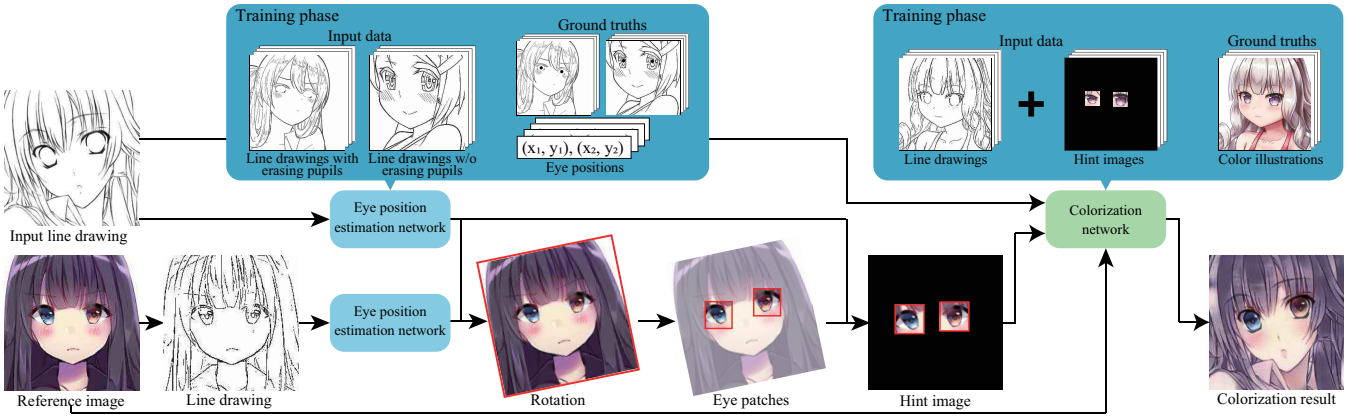


Figure 2: *Overview of proposed method. Line drawings with and without edges in pupils are used to train the eye position estimation network and line drawings without edges in pupils are used to train the colorization network.*

they are randomly cropped to 224 × 224 pixels in the bounding box of the rotated images. These images are the line drawing dataset for this network.

We train the eye position estimation network described in Sec. 4.1 using this mixed line drawing dataset. We create the ground truths of eye positions by manually specifying the positions of the right and left eyes.

3.3. Dataset Creation for Colorization Network

We train the colorization network using the pseudo empty dataset and color hint images as the input data and the corresponding color images for the dataset as the ground truths. The line drawings and color images are randomly cropped to 224 × 224 pixels inside these images to increase robustness. When creating a hint image, if the eye patches are in the original scale, the coloring results will include areas that are unpainted or colored beyond the edges when the eye size is different between the input line drawing and the reference image. To minimize these artifacts, the height and width of the reference image are independently scaled in the range of 0.5 to 2 times, and then the eye patches with a size of 48 × 48 pixels are cut out around the eyes. The cropped right and left patches are randomly arranged in one of four combinations (left-right, right-left, left-left, or right-right). We train the colorization network with the above-described hint images.

3.4. Hint Image Creation

A hint image is generated from the input reference image during colorization. The eye positions of the line drawing extracted from the color reference image are first predicted by the eye position estimation network. Then, the reference image is rotated so that the eyes are at the same angle as that of those in the line drawing to avoid shifting the pupil area in the output image. Eye patches around the pupils with a size of 48 × 48 pixels are generated, as done in the training of the colorization network. Then, the eye patches are arranged at the eye positions of the input line drawing in an empty (black) image to generate the hint image.

4. Network

Our method uses two types of network, namely the eye position estimation network and the colorization network. In this section, the structure and training of each network is described.

4.1. Eye Position Estimation Network

The input and ground truths for training the eye position estimation network are the mixed line drawing dataset (Sec. 3.2) and the corresponding eye positions, respectively. Except for the input and output layers, the network structure is based on ResNet-34 [HZRS16]. The input layer has one channel to input the grayscale line drawing. The output layer has four channels to output the x and y coordinates of the right and left eyes. The output layer has no activation function because the network outputs coordinates.

The loss function of the network is the mean squared error. The Adam optimizer [KA15] is used with the parameters $lr = 2e-4$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. We trained the model for about 15k iterations with a batch size of 256.

4.2. Colorization Network

We use the line drawing dataset with pseudo empty pupils (Sec. 3.3) and corresponding color hints of pupils as the inputs, and we use the corresponding color image as the ground truths to train the colorization network. The structures of the generator and discriminator of this network are almost the same as those for the drafting stage [ZJL*18]. Unlike the drafting stage, batch normalization [IS15] is used in the middle layer of our generator, and spectral normalization [MKKY18] is used in the middle layer of our discriminator. In addition, the reference image is transformed into color features using a histogram model [FHOO17]. These color features are input into the middle layer of the network. The input layer has four channels to input the grayscale line drawing and the hint image of pupils.

The loss function of the network is L1 loss and adversarial loss [GPAM*14], and optimizer is the Adam optimizer [KA15] with

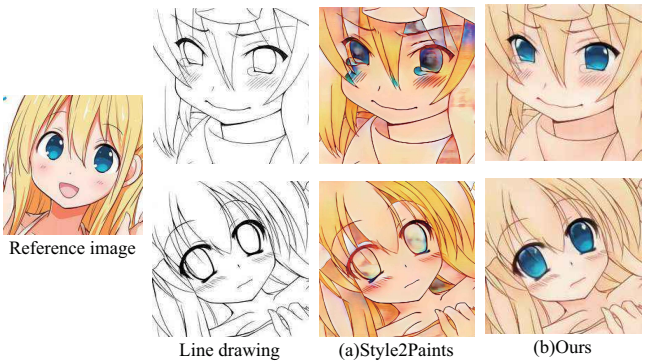
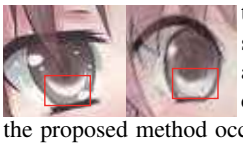


Figure 3: *Comparison of colorizing multiple line drawing inputs.*

the parameters $lr = 2e-4$ for the generator, $lr = 2e-5$ for the discriminator, and $\beta_1 = 0.9$ and $\beta_2 = 0.99$ for both. These losses were weighted as L1:adversarial = 1:0.001. We trained the model for about 110k iterations with a batch size of 32. To adapt the network to a real-world line drawing, we trained it for about 30k additional iterations with a brightness control technique [SLF*17].

5. Results and Limitations

Figure 1 shows comparisons between our results and those obtained using existing methods. We briefly described these methods in Sec. 1. Our method shows high reproducibility of reference colors when coloring multiple input line drawings using one reference image (Fig. 3). Moreover, it is difficult to accurately colorize small regions based on a reference image using most existing methods. In contrast, the proposed method enables highly accurate colorization of the eyes, even when the left and right eye colors of the reference image are different (Fig. 2).



However, our method cannot accurately transfer details such as highlights and shadows of regions other than the eyes. In addition, our method sometimes unevenly colorizes an image. Even for the pupils, the proposed method occasionally does not transfer small details, such as those in the red frames of the above eye images (the left is a reference, the right is a result).

6. Conclusion

In this paper, we proposed a network model that can colorize an illustration in a way that reflects the details in reference image pupils. Our method accurately colorizes small areas of pupils by predicting eye positions. To enable the painting of details on empty pupils, edges around pupils are erased in the line drawing. Eyes can be colorized with different colors and details can be painted on empty pupils; these tasks are not possible using most existing methods. We would like to extend the proposed approach to other areas, such as hair, in future work.

References

[AcBG19] ANONYMOUS, COMMUNITY D., BRANWEN G., GOKASLAN A.: Danbooru2018: A large-scale crowdsourced and tagged anime illustration dataset. https://www.gwern.net/Danbooru2018 , 2019. 2

[CMW*18] CI Y., MA X., WANG Z., LI H., LUO Z.: User-guided deep anime line art colorization with conditional adversarial networks. *MM* (2018), 1536–1544. 1

[FHOO17] FURUSAWA C., HIROSHIBA K., OGAKI K., ODAGIRI Y.: Comicolorization: Semi-automatic manga colorization. *SIGGRAPH Asia Technical Briefs* (2017). 1, 3

[GL19] GUO J., LIU Y.: Image completion using structure and texture gan network. *Neurocomputing* (2019). 1

[GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. *NIPS* (2014). 3

[HLK19] HUANG J., LIAO J., KWONG T. W. S.: Semantic example guided image-to-image translation. *arXiv* (2019). arXiv:1909.13028 . 1

[HWO12] HOLGER WINNEMÖLLER J. E. K., OLSEN S. C.: Xdog: An extended difference-of-gaussians compendium including advanced image stylization. *Computers & Graphics* (2012). 2

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. *CVPR* (2016). 3

[IOO16] IKUTA H., OGAKI K., ODAGIRI Y.: Blending texture features from multiple reference images for style transfer. *SIGGRAPH Asia Technical Briefs* (2016). 2

[IS15] IOFFE S., SZEGEDY C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML* (2015). 3

[KA15] KINGMA D. P., ADAM B. J.: Adam: A method for stochastic optimization. *ICLR* (2015). 3

[KJPY19] KIM H., JHOO H. Y., PARK E., YOO S.: Tag2pix: Line art colorization using text tag with secat and changing loss. *ICCV* (2019). 1, 2

[LAGB16] L. A. GATYS A. S. E., BETHGE M.: Image style transfer using convolutional neural networks. *CVPR* (2016). 1, 2

[lll18] LLLYASVIEL: style2paints v3. https://github.com/lllyasviel/style2paints , 2018. 1, 2

[LZL17] LVMIN ZHANG YI JI X. L., LIU C.: Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. *ACPR* (2017). 1

[MKKY18] MIYATO T., KATAOKA T., KOYAMA M., YOSHIDA Y.: Spectral normalization for generative adversarial networks. *ICLR* (2018). 3

[Nag11] NAGADOMI: lbpcascade_animeface. https://github.com/nagadomi/lbpcascade_animeface , 2011. 2

[nic] NICONICO:. https://www.nicovideo.jp/ . 2

[SLF*17] SANGKLOY P., LU J., FANG C., YU F., HAYS J.: Scribbler: Controlling deep image synthesis with sketch and color. *CVPR* (2017). 4

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *arXiv* (2014). arXiv:1409.1556 . 1

[XSA*18] XIAN W., SANGKLOY P., AGRAWAL V., RAJ A., LU J., FANG C., YU F., HAYS J.: Texturegan: Controlling deep image synthesis with texture patches. *CVPR* (2018). 1

[Yon17] YONETSUJI T.: petalica paint. https://petalica-paint.pixiv.dev/index_en.html , 2017. 1, 2

[ZJL*18] ZHANG L., JI Y. I., LI C., WONG T.-T., JI Y., LIU C.: Two-stage sketch colorization. *ACM Trans. Graph 37*, 6 (2018). 1, 3