DOI: 10.1111/cgf.14171
Pacific Graphics 2020
E. Eisemann, A. Jacobson, and F.-L. Zhang
(Guest Editors)

Volume 39 (2020), Number 7

Colorization of Line Drawings with Empty Pupils

K. Akita [1], Y. Morimoto[1], and R. Tsuruno[1]

[1]Kyushu University

Figure 1: *The input line drawing (left), color reference image (middle), and our result (right). Our method can paint details in empty pupils by transferring pupil details from the reference image.*

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 convolutional neural network 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 colorizing 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 pupil 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. However, it is difficult to paint details in empty pupils using these methods because the line drawings used for training are very different from actual line drawings drawn by a human. To train a network, these methods use the line drawings extracted from color illustrations drawn by illustrators. Therefore, the pupils in the extracted line drawings contain edges. In contrast, actual color character illustrations are created by painting or filling in almost empty parts of line drawings. Such line drawings are mostly contour lines and lack details. The pupils are thus empty (Fig. 2). Details are painted during the colorization of the line drawing.

Ci et al.'s method [CMW*18], Petalica Paint [Yon17], and Zhang et al.'s method [ZLW*18] are semi-automatic colorization methods for intuitively colorizing line drawings based on color scribbles or dots input by users. These methods achieve accurate and detailed colorization by having users input color scribbles or dots in regions to be colorized. Tag2Pix [KJPY19] is a colorization method that allows users to specify regions and colors using natural language, such as "brown hair". Input can also be in the form of a color reference image, where methods apply the colors in the reference image to an input line drawing image [ZJLL17, FHOO17, lll18]. These methods can automatically or semi-automatically colorize line drawings. The inputs to these methods contain edges, whereas actual color character illustrations are created from drawings that contain mostly contour lines, where pupils, the most important

Figure 2: *Line drawing for creating an actual color character illustration.*

parts of character drawings, have no edges inside. This problem is overlooked, and it is difficult to delete pupil details.

In this paper, we propose a deep learning model for semi-automatic colorization that can add pupil details to line drawings. Our method extracts line drawings from color character illustrations. Then, the network estimates the pupil positions in these line drawings. We create line drawings with pseudo empty pupils by erasing the region around the estimated pupil positions. This process creates line drawings with empty pupils (such images are traditionally difficult to obtain in large quantities) for use as a training dataset for our colorization network.

We train the colorization network with a hint image that consists of a line drawing with empty pupils and pupils cropped from a color reference image. Such hint images train the colorization network to transfer pupil details in a hint image to a line drawing. Our method can semi-automatically colorize line drawings based on an input line drawing and a color reference image (Fig. 1), which are processed by two networks, namely the pupil position estimation network and the colorization network, respectively. The colorization target of our method is character face illustrations. Our main contributions can be summarized as follows:

- We propose a method for creating a line drawing dataset with pseudo empty pupils and a hint patch for generating pupil details.
- We combine a colorization network with a pupil position estimation network to automatically and accurately colorize pupils.

For previous methods, the input line drawings are different from actual line drawings used to create a color illustration. Our method is the first reported attempt to overcome this problem.

2. Related Work

2.1. Line Drawing Colorization

Many automatic line drawing colorization methods based on deep learning have recently been proposed. Some methods use user-specified color hints in the form of added color strokes [CMW*18, Yon17, SLF*17] or dots [ZLW*18, LQWL18] in regions to be colorized. These methods achieve accurate and detailed colorization even if the regions are quite small. However, they are often complicated because users need to input many color hints for accurate colorization.

Some semi-automatic colorization methods simplify line drawing colorization by allowing natural language or color reference images to be used as color hints. Tag2Pix [KJPY19] is a colorization method that allows users to specify regions and their colors using natural language in the form of tags. Although this method is intuitive, color control by users in details is difficult because Tag2Pix is limited to colorizing learned regions. Zou et al. [ZMG*19] proposed a method similar to Tag2Pix. The main differences are that their method applies sentences as the input, whereas Tag2Pix applies words, and that their method enables the colorization of multiple objects in a scene, whereas Tag2Pix does not. However, Zou et al.'s method cannot paint details that do not appear in the input image and it is difficult to specify such painting details using sentences.

Methods that use a color reference image [ZJLL17, HLK19, FHOO17, lll18, SLWW19, LKL*20] colorize a line drawing based on colors in a reference image. These methods enable diverse and flexible colorization of line drawings if a color reference image is prepared. Comicolorization [FHOO17] and Style2Paints [lll18] colorize line drawings based on a color reference image. Users can adjust the colorization result using color dots.

The above methods train a network using line drawings with edges in the pupils and thus cannot accurately paint details in empty pupils. 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. Compared with Deep-Eyes [AMT20], which enables semi-automatic colorization, including pupil details, based on a color reference image, our colorization network enables more accurate and detailed colorization by adding feature reconstruction loss to loss functions and using a distance field image as an input image.

2.2. Neural Style Transfer

Neural style transfer [GEB16, HB17, LW16, LYY*17, PL19] uses a convolutional neural network (CNN) to transfer the style of a reference image to an input image; it can generate the details in the reference image. Neural style transfer is effective for photographs and pictures. However, to apply neural style transfer to line drawings, the reference image and input image must have almost the same features (e.g., character poses and image structures). In contrast, the proposed method can colorize a line drawing and transfer pupil details in a reference image to the line drawing even if the structures of the line drawing and color reference image are different because it trains a colorization network using a hint image that consists of pupil patches of a color reference image (these patches are arranged at the pupil positions of the line drawing).

2.3. Image Synthesis

Recently, an image synthesis method for line drawings that uses a generative adversarial network (GAN) has been proposed [GPAM*14]. Pix2Pix [IZZE17], LinesToFacePhoto [LCWZ19], and SketchyGAN [CH18] are image synthesis methods that take a line drawing or sketch as the input. These methods can generate

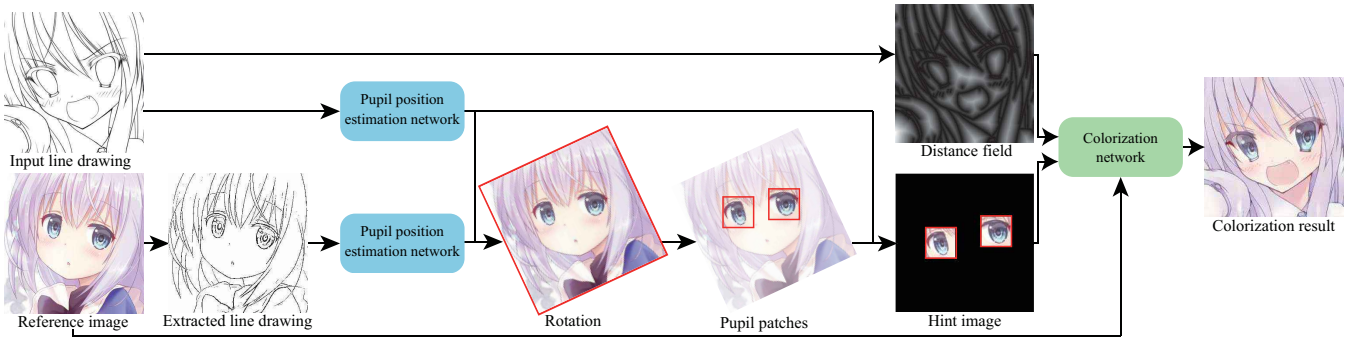

Figure 3: *Test phase of the proposed system.*

details inside the contours of a line drawing, but users cannot specify these details and their colors. In contrast, with our method, users can specify the pupil details and their colors with a color reference image.

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.

3. Overview

Figure 3 shows an overview of our system in the test phase. A line drawing and a color reference image (face illustration) are given as inputs. The pupil positions in the input images are first estimated by the pupil position estimation network. A hint image is generated from these pupil positions using image processing. Then, the distance field generated from the input line drawing, the hint image, and the reference image are input to the colorization network. We train the pupil position estimation network with line drawings as the input and the pupil positions as the output (Fig. 4, top). Our colorization network is trained with the distance field image and the corresponding hint images as the input and the corresponding color images as the output (Fig. 4, bottom). In these line drawing datasets, regions around the pupils are erased. See Sec. 4 for our data creation method.

4. Data Creation

Here, we describe the creation of the datasets for the pupil position estimation network and the colorization network. We used Danbooru2018 [AcBG19], which is a large-scale character illustration image database, to create dataset images to train the networks. We used 2275 and 256 images to train and validate the pupil position estimation network, respectively, and about 420000 images cropped from color illustrations to train the colorization network. To avoid low-resolution images, images with dimensions of at least 256 × 256 pixels were detected by a face detector [Nag11]. Images larger than 256 × 256 pixels were cropped and resized to 256 × 256 pixels. The images were randomly cropped to 224 × 224 pixels. The line drawings and color illustrations shown in the figures are from Danbooru2018.

4.1. Line Drawing Creation

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 types of line extraction are randomly applied, namely a morphological operation [Yon17] and extended difference-of-Gaussians (XDoG) [WKO12]. To estimate the pupil positions in an input color reference image, we apply XDoG to extract the line drawing (Fig. 3). We attached the source code as reference material for those interested in the details.

4.2. Dataset Creation for Pupil Position Estimation Network

The pupil position estimation network estimates the pupil 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.

Then, line drawings with pseudo empty pupils are automatically created as follows. The areas around pupils at the ground truth positions in the extracted line drawings are filled with white ellipses. The height and width of the ellipses are randomly set to values in the ranges of 15 to 30 pixels and 8 to 30 pixels, respectively. The ground truths of the pupil positions are created in advance by manually specifying the centers of the right and left pupils. The color image dataset for this network is transformed into line drawings, 80% of which are transformed to pseudo empty pupils. The mixed line drawing dataset comprises images with and without pseudo empty pupils. These images are the line drawing dataset for this network. We train the pupil position estimation network described in Sec. 5.1 using this mixed line drawing dataset.

4.3. Dataset Creation for Colorization Network

We train the colorization network using a dataset of images with our pseudo empty pupil dataset and color hint images as the input data and the corresponding color images for the dataset as the ground truths. A pseudo empty pupil dataset is created using the procedure described in Sec. 4.2 to generate distance fields. The pupil positions are estimated by our estimation network. Here, the dataset

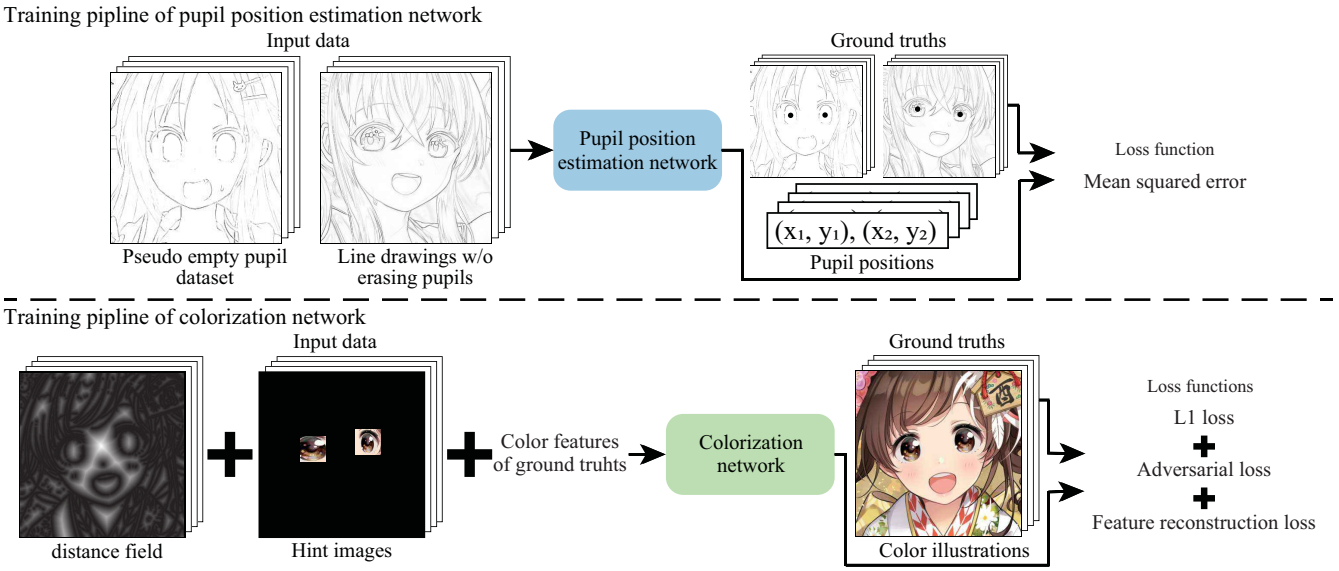

Figure 4: *Training of networks. Line drawings with and without edges in pupils are used to train the pupil position estimation network.*

with pseudo empty pupils is transformed into distance field images as input for this network because distance field images are a dense representation, which includes the distance from drawing lines, for training the colorization network [SZC*20, CH18, LCWZ19]. A distance field representation is more robust than line drawings against noise and variations in input images. It enables the colorization of a given region with the same color and the generation of shadows and highlights in hair and skin regions.

When automatically creating a hint image to train the network, if the pupil patches are on the original scale, the colorization results will include areas that are unpainted or colorized beyond the edges because the pupil sizes in the input line drawing and the reference image are different. In such cases, the network generates pupils on the original scale regardless of the pupil size in the input line drawing. 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 pupil patches with a size of 48 $\times$ 48 pixels are cut out around the pupils. The cropped right and left patches are randomly arranged in one of four combinations (left-right, right-left, left-left, or right-right). Then, the pupil patches are arranged at the pupil positions in the input line drawing in an empty (black) image to generate the hint image. We train the colorization network with these hint images.

4.4. Hint Image Creation during Testing

The pupil angle in a reference image is usually different from that in an input line drawing. Therefore, the reference image is rotated during testing so that the pupils are at the same angle as that of those in the line drawing to avoid shifting the pupil area in the output image. The preprocessing of our method deals with only the rotation of the hint image during testing. The rotation is automatically applied as follows. The angle between pupils in the reference image and pupils in the input line drawing is obtained from linear functions derived from the pupil positions output by ResNet. The reference image is rotated so that this angle is 0 degrees. Pupil patches around the pupils with a size of 48 $\times$ 48 pixels are generated, as done in the training of the colorization network.

5. Network and Training

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

5.1. Pupil Position Estimation Network

Generally, character illustrations have various pupil shapes and textures. To deal with the various pupil shapes, we use a CNN for pupil position estimation. Object detection using a CNN [GDDM14, WSC*19, TPL20] can be used for the detection of regions of a specified object; however, many data are required to train the CNN, which is thus very time-consuming. The proposed method uses a relatively simple CNN based on ResNet-34 [HZRS16] to estimate the centers of pupils. This network can be trained using a small dataset, it very accurately estimates positions, and its training is very fast.

The input and ground truths for training the pupil position estimation network are the mixed line drawing dataset (Sec. 4.2) and the corresponding pupil positions, respectively. The input and output layers of our network are different from those of ResNet-34. The input layer has one channel for inputting the grayscale line drawing. The output layer has four channels for outputting the x and y coordinates of the right and left pupils. The loss function of the network is the mean squared error. The Adam optimizer [KA15] is used with the parameters $lr = 0.0002$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. We updated the weights on two GPUs (NVIDIA RTX 2080 Ti) and

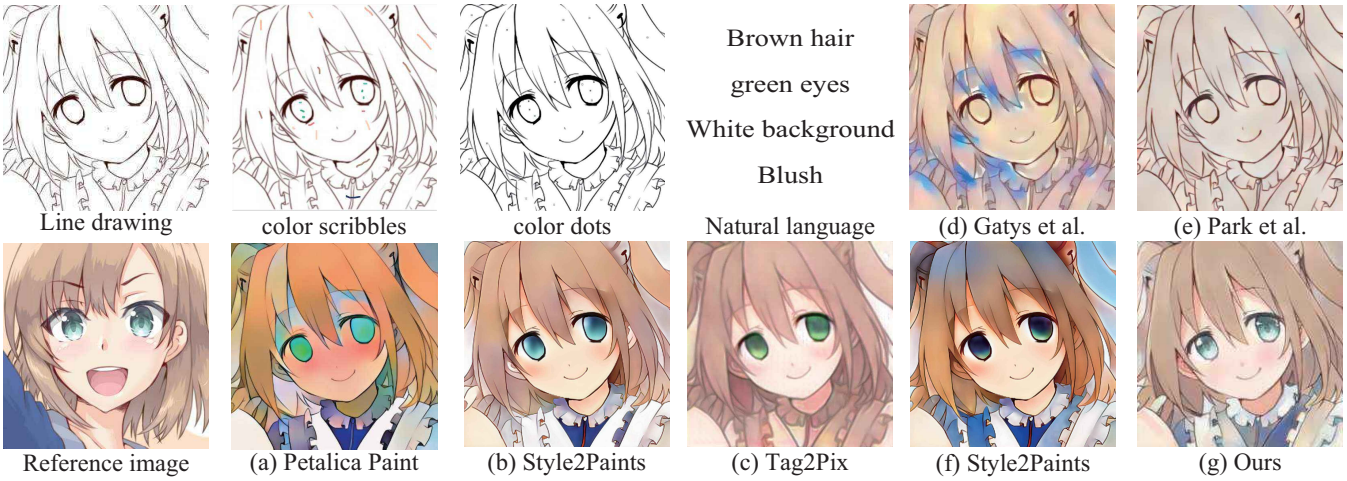

Figure 5: *Comparison between existing colorization methods [Yon17, lll18, KJPY19, GEB16, PL19] and proposed method. The color reference inputs are (a) color scribbles, (b) color dots, (c) natural language, and (d, e, f, g) color reference images. All output images are colorized based on the reference color image. The existing methods roughly colorize the line drawing but cannot generate pupil details. The proposed method accurately colorizes the line drawing and generates pupil details based on our dataset of line drawings with empty pupils.*

trained the model for 1325 epochs with a batch size of 128 per GPU. The time required for training was about 1 hour 40 min.

5.2. Colorization Network

We use the distance field dataset, which contains images with pseudo empty pupils (Sec. 4.4), and the corresponding color hints of pupils as the inputs, and 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 [ZLW*18]. Unlike the drafting stage, 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. A diagram of the network architecture is shown in the supplemental material. The input layer has four channels for inputting the grayscale line drawing and the hint image of pupils. The loss functions of the network are L1 loss (L_1), adversarial loss [GPAM*14] (L_{adv}), and feature reconstruction loss [HLK19] (L_F). Feature reconstruction loss is a loss function used to match high-level similarities between the output and ground truth using image feature layers ($relu3\_2$) from VGG-19 [SZ15]. Using this loss function, the network can generate more detailed pupils. L_F is expressed as follows:

$$L_F = \mathbb{E}[\|\Phi(G(x, x_{hint}, v_c)) - \Phi(y)\|_1] \quad (1)$$

where G is the generator and x and x_{hint} are the distance field image and hint image, respectively. v_c is the color feature vectors and y is the ground truth. $\Phi(\cdot)$ indicates feature extraction. The loss function of the colorization network adds these loss functions (Eq. 2).

$$L_{color} = \alpha L_1 + \beta L_{adv} + \gamma L_F \quad (2)$$

where α, β, and γ are the weights for the three loss functions. In our method, $\alpha = 1.0$, $\beta = 0.001$, and $\gamma = 0.1$. The Adam optimizer uses the parameters $lr = 0.0002$ for the generator, $lr = 0.00002$ for the discriminator, and $\beta_1 = 0.9$ and $\beta_2 = 0.99$ for both. We updated the weights on two GPUs (NVIDIA RTX 2080 Ti) and trained the model for 11 epochs with a batch size of 16 per GPU. The time required for training was about 34 hours.

6. Results and Evaluation

6.1. Comparison of Results

Figure 5 shows a comparison between our results and those obtained using existing methods. Figure 5 (f) shows the result of Style2Paints [lll18] obtained using a reference image. Using only the reference image, the method can transfer the colors of the reference image to the line drawing, but often cannot colorize regions with specified colors. For example, in Fig. 5 (f), the blue color of the clothes is reflected in the hair. Tag2Pix [KJPY19] can colorize specified regions with specified colors using natural language as color hints (Fig. 5 (c)) . However, these methods cannot accurately paint pupil details (Figs. 5 (c) and (f)).

Figure 5 (b) shows the image corrected by adding color dots to 5 (f) with Style2Paints. In such cases, the results often become more vivid than those obtained using the proposed method. Style2Paints with a reference image and color dots can paint pupil details based on color gradation, but it cannot generate the high-frequency components of pupil details. In addition, users need to input many color hints to generate pupil details. Petalica Paint [Yon17] enables colorization based on color hints, but it cannot generate pupil details and the sclera. Neural style transfer cannot transfer the style of a color reference image to a line drawing if the characters and their poses in the input color reference image are different from those in the line drawing (Figs. 5 (d) and (e)).

Figure 6: *Comparison of multiple colorized line drawings.*

Table 1: *Comparison of proposed method with previous methods.*

Method	FID	PSNR	SSIM
Tag2Pix	89.86	\	\
Style2Paints	\	12.28	0.5320
Ours	**30.59**	**15.46**	**0.6268**

In contrast to these methods, our method can paint pupil details and colorize hair and skin with colors of the corresponding regions in a color reference image without additional input (Fig. 5 (g)). In addition, our method shows high reproducibility of reference colors when colorizing multiple input line drawings using a given reference image (Fig. 6), and enables highly accurate colorization of the pupils, even when the left and right pupil colors of the reference image are different (Fig. 6, right).

To evaluate our method, we quantitatively compared it with previous methods. We selected Tag2Pix and Style2Paints because they allow easy specification of colors. We used the Fréchet inception distance (FID) [HRU*17] to compare our method and Tag2Pix, and the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) to compare our method and Style2Paints using only a reference image. FID is a metric for evaluating the performance of a GAN [GPAM*14]. It represents the difference between the distribution of ground truth images and that of generated images. A lower FID value indicates higher similarity between the ground truth images and the generated images. PSNR evaluates image reproducibility; it indicates how similar an output image is to a ground truth image. SSIM is similar to PSNR, but is more sensitive to local differences. For both PSNR and SSIM, higher values indicate higher similarity between an output image and a ground truth image.

To calculate FID, we used 43000 color images as the test dataset and 235 line drawings drawn by a human. We obtained all color images of the test dataset from Danbooru2018 [AcBG19] , 203 line drawings from Danbooru2018, and 32 line drawings from online sources. The same test dataset and line drawings were used for Tag2Pix and our method. To colorize face line drawings using Tag2Pix, we created tags for the hair, eyes, blush, and background as color hints. These tags are described below.

Figure 7: *Representation of images in the questionnaire.*

Table 2: *User study results.*

Categories	Style2Paints	w/o erasing pupils	Ours
Color reproducibility	1.75	3.3	**4.1**
Pupil details	1.26	2.08	**4.28**
Details except for pupils	1.85	3.15	**3.73**
Overall quality	1.64	2.72	**3.77**

We used 180 combinations of hair tags (15 colors) and eye tags (12 colors) as color hints. Then, a blush tag (pink or transparent) and background tags (13 colors) was selected randomly. These tags were added to the 180 combinations of hair and eye tags. We colorized a line drawing using 180 tag hints, creating 42300 colorized images. Our method colorized a line drawing using 180 reference images, creating 42300 colorized images.

We calculated PSNR and SSIM for the results generated by colorizing an actual line drawing using semi-automatic colorization methods and actual illustrations. We obtained 28 sets from online sources because it is difficult to collect a lot of actual line drawings as input and the corresponding color illustrations created by a human. We colorized the line drawings using the corresponding color images, creating 28 colorized images using our method and 28 colorized images using Style2Paints. Table 1 shows the FID, PSNR, and SSIM values for each method. Our method has the highest value for each metric.

6.2. User Study

To evaluate our method, we conducted an online questionnaire with 12 participants. Using reference color images, we compared our method to Style2Paints using only the reference color images [lll18] and our network trained without erasing pupils. We colorized 14 line drawings randomly selected from 235 actual line drawings with 10 reference color images randomly selected from 103 color images, using three colorization methods. Of the 140 (14 line drawings $\times$ 10 reference images) images created for each method, we randomly selected 10 images for evaluation. Each participant evaluated 30 images in terms of four criteria. Each evaluation criterion was scored on a five-point Likert scale. The evaluation criteria were as follows:

- **Color reproducibility**
  Closeness of the colors in the generated image to those in the reference image.

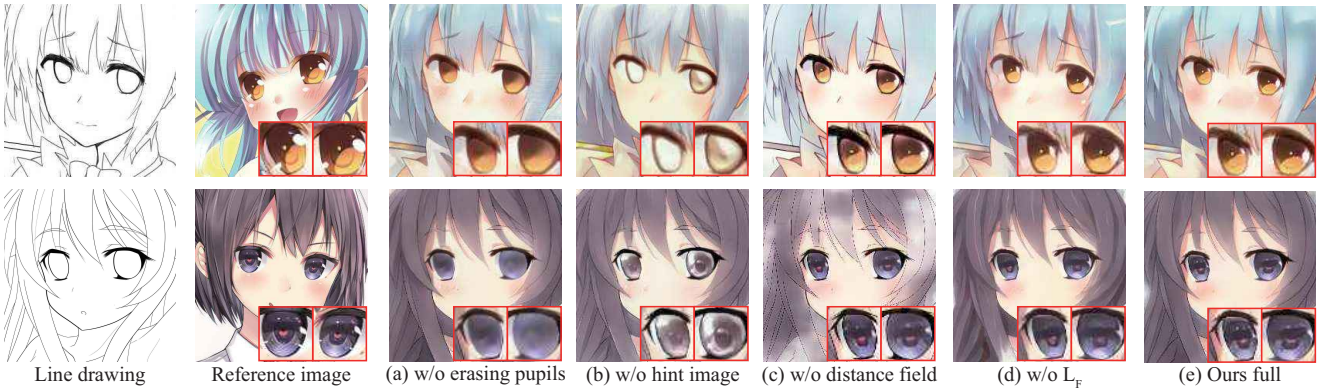

Figure 8: *Ablation study results of colorization network.*

- **Pupil details**
  Transfer of pupil details in the reference image to the generated image.
- **Non-pupil details**
  Transfer of non-pupil details in the reference image to the generated image.
- **Overall quality**
  Quality of the color character illustration.

Table 2 shows that our method outperformed the existing methods in all evaluation metrics. All scores for Style2Paints are low. Style2Paints tends to colorize line drawings more vividly than the reference image, and often includes hair color in skin. Our method without erasing pupils obtained a low score for pupil details. This shows that the colorization network cannot transfer pupil details without erasing pupils. The overall quality of this method is also low, indicating that pupil details are important in a color illustration.

6.3. Analysis of Proposed Method

We analyzed our data creation, input image, and loss function. We conducted ablation studies of the colorization network to confirm the effectiveness of the data creation, distance field image, and feature reconstruction loss. Figure 8 shows the generated images. The colorization network trained using line drawings often failed to colorize hair regions and sometimes did not paint shadows (Fig. 8 (c)), compared with our result using distance field images (Fig. 8 (e)). The colorization network trained without hint images sometimes could not paint pupil details and users cannot specify the pupil details as an input (Fig. 8 (b)). The network trained without erasing pupil details could not generate the high-frequency components of pupils in the reference image; it could only transfer pupil colors (Fig. 8 (a)).

We evaluated the results in terms of FID, PSNR, and SSIM. Then, we calculated the PSNR and SSIM of images, 48×48 pixels in size, cropped at the estimated pupil positions because the focus is the painting of pupil details. Table 3 shows the values of FID, PSNR for the whole face, SSIM for the whole face, PSNR for only the pupils, and SSIM for only the pupils. In the table, the lowest scores

Figure 9: *Comparison of results obtained with and without rotation of the reference image. Without rotation, the angle of pupil details is different from that of the face in the generated image.*

Figure 10: *Example of generated image when the estimated pupil position is outside the pupil.*

of each metric during training in 11 epochs are shown. The dataset used for evaluation was the same as that described in Sec. 6.1. The method without L_F had the highest FID and PSNR (face) values. The full method had the highest PSNR (pupils), SSIM (face), and PSNR (pupils) values. Figures 8 (d) and (e), and these scores show that the colorization network can paint more pupil details when it uses feature reconstruction loss; however, the colorization network trained without this loss enables colorization that is closer to the ground truth images. For the method without feature reconstruction loss, the FID value is the highest but the lines of input line drawing are often deleted (Fig. 8 (d)).

The reference image is rotated so that the pupils are at the same angle as that of those in the line drawing. We thus compared the cases with and without the rotation of the reference image. When the face angles of the images are different, the generated image becomes unnatural because the face angle is different from the pupil angle (Fig. 9).

Table 3: *Evaluation of various versions of the proposed methods in terms of FID, PSNR, and SSIM.*

Method	FID	PSNR (face)	PSNR (pupils)	SSIM (face)	SSIM (pupils)
(a) w/o erasing pupils	36.31	15.27	16.84	0.6144	0.6590
(b) w/o hint image	38.69	14.59	13.35	0.5973	0.5070
(c) w/o distance field	69.67	13.92	17.55	0.6202	0.7546
(d) w/o L_F	**28.42**	**15.51**	18.64	0.6003	0.7470
(e) Ours Full	30.59	15.46	**19.23**	**0.6268**	**0.7924**

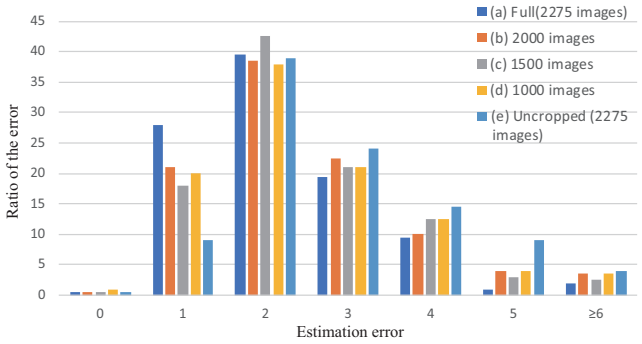

Figure 11: *Estimation accuracy of pupil positions in extracted line drawing.*

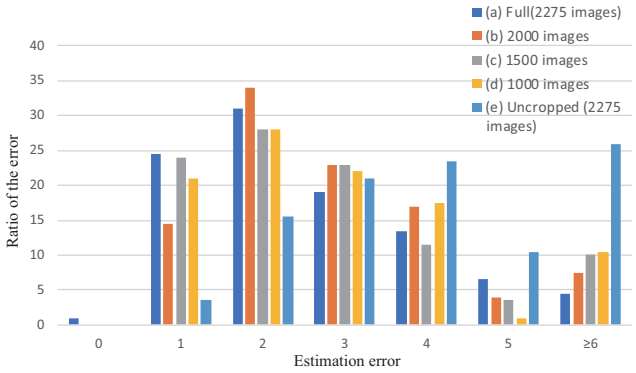

Figure 12: *Estimation accuracy of pupil positions in actual line drawing.*

In our method, cropped patches around the estimated pupil positions are given to the colorization network as hints. When the estimated positions are outside the pupils, the colorization network cannot accurately paint the pupils (Fig. 10). Therefore, we calculated the error between the ground truth positions and the estimated positions. We used 200 actual line drawings and extracted line drawings from color images for this test. We used the Euclidean distance to calculate the errors because the network estimates the centers of pupils. The results were rounded to the nearest integer. If the estimated position of even one pupil had a large deviation, our method could not transfer the pupil details. Therefore, we calculated the error between the ground truth positions and the estimated positions for each pupil. The largest value of the two pupils was taken as the estimation error. Figures 11 and 12 show comparisons of the estimation error for the pupil position estimation

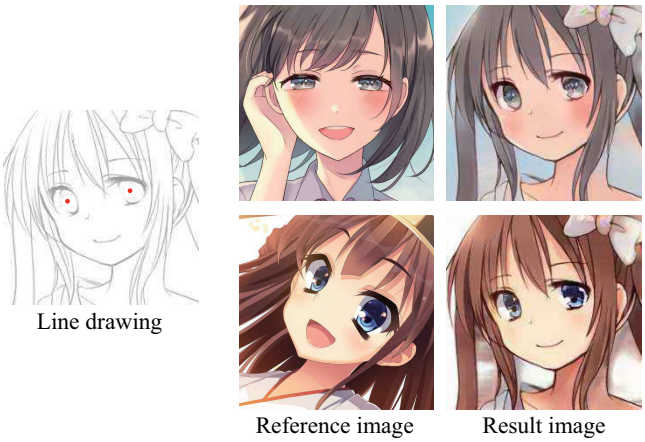

Figure 13: *Example of colorized image with an estimation error of 4 pixels. The red dots in the line drawing show the estimated pupil positions.*

network trained with 2275, 2000, 1500, and 1000 cropped images, and 2275 uncropped images. For training with 2275 cropped images (a), the estimation error was less than 4 pixels for extracted line drawings from color illustrations for 98% of the color images and 89% of actual line drawings. In contrast, for training with 2275 uncropped images (e), the estimation accuracy is relatively poor. This empirically shows that random cropping increases estimation accuracy. For errors of less than 4 pixels, the estimated pupil positions are mostly located within the pupils even for small pupils, and the colorization network can generate images with little distortion (Fig. 13).

Comparing with the cases trained with fewer images (b, c, d), the errors of less than 4 pixels were obtained for more than 90% and 80% of line drawings extracted from illustrations and the actual line drawings, respectively, even for the dataset that included only 1000 images. These results show that our network trained with a small dataset can estimate pupil positions with an error of less than 4 pixels.

6.4. Limitations

When the estimated positions are outside the pupils, our method transfers pupil details to these incorrect positions (Fig. 10). The pupil position estimation network is trained with line drawings that include both right and left pupils. Therefore, for line drawings with only one pupil, the network fails to estimate the pupil position and cannot transfer pupil details. In the future, these problems will be

addressed by developing a graphical user interface. In addition, the proposed method cannot generate details in areas other than pupils, such as hair regions.

7. Conclusion

In this paper, we proposed a network 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 estimating pupil positions. To enable the painting of details on empty pupils, edges around pupils are erased in the line drawing. For more accurate colorization, the colorization network uses distance field images as input images. This network generates more pupil details by adding feature reconstruction loss to the loss functions.

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. Accessed: 2020-06-30. 3, 6

[AMT20] AKITA K., MORIMOTO Y., TSURUNO R.: Deep-Eyes: Fully Automatic Anime Character Colorization with Painting of Details on Empty Pupils. In *Eurographics 2020 - Short Papers* (2020). 2

[CH18] CHEN W., HAYS J.: SketchyGAN: Towards diverse and realistic sketch to image synthesis. In *Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 9416–9425. 2, 4

[CMW*18] CI Y., MA X., WANG Z., LI H., LUO Z.: User-guided deep anime line art colorization with conditional adversarial networks. In *ACM International Conference on Multimedia (MM)* (2018), pp. 1536–1544. 1, 2

[FHOO17] FURUSAWA C., HIROSHIBA K., OGAKI K., ODAGIRI Y.: Comicolorization: Semi-automatic manga colorization. In *SIGGRAPH Asia Technical Briefs* (2017). 1, 2, 5

[GDDM14] GIRSHICK R., DONAHUE J., DARRELL T., MALIK J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)* (2014), pp. 580–587. 4

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. pp. 2414–2423. 2, 5

[GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *Neural Information Processing Systems (NIPS)* (2014), pp. 2672–2680. 2, 5, 6

[HB17] HUANG X., BELONGIE S.: Arbitrary style transfer in real-time with adaptive instance normalization. In *International Conference on Computer Vision (ICCV)* (2017), pp. 1510–1519. 2

[HLK19] HUANG J., LIAO J., KWONG T. W. S.: Semantic example guided image-to-image translation. *arXiv* (2019). arXiv:1909.13028 . 2, 5

[HRU*17] HEUSEL M., RAMSAUER H., UNTERTHINER T., NESSLER B., HOCHREITER S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. p. 6629–6640. 6

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778. 4

[IZZE17] ISOLA P., ZHOU T., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 5967–5976. 2

[KA15] KINGMA D. P., ADAM B. J.: Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)* (2015), CoRR. 4

[KJPY19] KIM H., JHOO H. Y., PARK E., YOO S.: Tag2Pix: Line art colorization using text tag with secat and changing loss. In *International Conference on Computer Vision (ICCV)* (2019). 1, 2, 3, 5

[LCWZ19] LI Y., CHEN X., WU F., ZHA Z.-J.: Linestofacephoto: Face photo generation from lines with conditional self-attention generative adversarial networks. In *ACM International Conference on Multimedia (MM)* (2019), p. 2323–2331. 2, 4

[LKL*20] LEE J., KIM E., LEE Y., KIM D., CHANG J., CHOO J.: Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 5801–5810. 2

[lll18] LLLYASVIEL: style2paints v4.5. https://github.com/lllyasviel/style2paints , 2018. Accessed: 2020-06-30. 1, 2, 5, 6

[LQWL18] LIU Y., QIN Z., WAN T., LUO Z.: Auto-painter: Cartoon image generation from sketch by using conditional wasserstein generative adversarial networks. *Neurocomputing 311* (2018), 78 – 87. 2

[LW16] LI C., WAND M.: Combining markov random fields and convolutional neural networks for image synthesis. In *Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2479–2486. 2

[LYY*17] LIAO J., YAO Y., YUAN L., HUA G., KANG S. B.: Visual attribute transfer through deep image analogy. *ACM Trans. Graph. 36*, 4 (2017). 2

[MKKY18] MIYATO T., KATAOKA T., KOYAMA M., YOSHIDA Y.: Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)* (2018). 5

[Nag11] NAGADOMI: lbpcascade_animeface. https://github.com/nagadomi/lbpcascade_animeface , 2011. Accessed: 2020-06-30. 3

[PL19] PARK D. Y., LEE K. H.: Arbitrary style transfer with style-attentional networks. In *Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 5873–5881. 2, 5

[SLF*17] SANGKLOY P., LU J., FANG C., YU F., HAYS J.: Scribbler: Controlling deep image synthesis with sketch and color. In *Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 6836–6845. 2

[SLWW19] SUN T.-H., LAI C.-H., WONG S.-K., WANG Y.-S.: Adversarial colorization of icons based on contour and color conditions. In *ACM International Conference on Multimedia (MM)* (2019), p. 683–691. 2

[SZ15] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)* (2015). 5

[SZC*20] SHI M., ZHANG J.-Q., CHEN S.-Y., GAO L., LAI Y.-K., ZHANG F.-L.: Deep line art video colorization with a few references, 2020. arXiv:2003.10685 . 4

[TPL20] TAN M., PANG R., LE Q. V.: Efficientdet: Scalable and efficient object detection. In *Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 10781–10790. 4

[WKO12] WINNEMÖLLER H., KYPRIANIDIS J. E., OLSEN S. C.: Xdog: An extended difference-of-gaussians compendium including advanced image stylization. *Computers & Graphics 36*, 6 (2012), 740–753. 3

[WSC*19] WANG J., SUN K., CHENG T., JIANG B., DENG C., ZHAO Y., LIU D., MU Y., TAN M., WANG X., LIU W., XIAO B.: Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019). 4

[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. In *Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 8456–8465. 3

[Yon17] YONETSUJI T.: petalica paint. https://petalica-paint.pixiv.dev/index_en.html , 2017. Accessed: 2020-06-30. 1, 2, 3, 5

[ZJLL17] ZHANG L., JI Y., LIN X., LIU C.: Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. *2017*

4th IAPR Asian Conference on Pattern Recognition (ACPR) (2017), 506–511. 1, 2

[ZLW*18] ZHANG L., LI C., WONG T.-T., JI Y., LIU C.: Two-stage sketch colorization. *ACM Trans. Graph 37*, 6 (2018). 1, 2, 5

[ZMG*19] ZOU C., MO H., GAO C., DU R., FU H.: Language-based colorization of scene sketches. *ACM Trans. Graph 38*, 6 (2019). 2