Multimedia Tools and Applications (2022) 81:26669–26683
https://doi.org/10.1007/s11042-020-10468-6

1175: IOT MULTIMEDIA APPLICATIONS AND SERVICES

Unpaired medical image colorization using generative adversarial network

Yihuai Liang[1] · Dongho Lee[1] · Yan Li[1] · Byeong-Seok Shin[1]

Received: 31 March 2020 / Revised: 10 November 2020 / Accepted: 29 December 2020 / Published online: 18 January 2021

Abstract
We consider medical image transformation problems where a grayscale image is transformed into a color image. The colorized medical image should have the same features as the input image because extra synthesized features can increase the possibility of diagnostic errors. In this paper, to secure colorized medical images and improve the quality of synthesized images, as well as to leverage unpaired training image data, a colorization network is proposed based on the cycle generative adversarial network (CycleGAN) model, combining a perceptual loss function and a total variation (TV) loss function. Visual comparisons and experimental indicators from the NRMSE, PSNR, and SSIM metrics are used to evaluate the performance of the proposed method. The experimental results show that GAN-based style conversion can be applied to colorization of medical images. As well, the introduction of perceptual loss and TV loss can improve the quality of images produced as a result of colorization better than the result generated by only using the CycleGAN model.

Keywords Generative adversarial network · Medical image colorization · TV loss · Perceptual loss

1 Introduction

Compared to grayscale images, color images contain detail and clear information so that, in many applications, when only grayscale images are generated, the grayscale image is often transformed into a color image first. Over the last decade, many studies have been done in various fields to automatically and effectively generate color images [1]. For example, automatic colorization [36] is used for grayscale satellite images to help eliminate lighting differences between multi-spectral captures, providing strong prior information for ground type classification and object detection [33]. Synthetic Aperture Radar (SAR) colorization [33]

✉ Byeong-Seok Shin
bsshin@inha.ac.kr

1 Department of Electrical and Computer Engineering, Inha University, 100 Inha-ro, Michuhol-gu, Incheon, South Korea

takes advantage of the well-established radar polarimetry theories and polarimetry approaches to interpret the vast majority of SAR images that are not full-pol. A semi-automatic colorization system takes a monochrome manga and reference images as input and generates a plausible color version of the manga [8].

Color medical imaging [29] is used in various fields, such as building virtual human models for educational purposes. A color mapping of the distribution of shear stress images [7] has been studied along the plaque surface over the last decade. However, most medical image datasets are photographed as grayscale images, so it is difficult to obtain enough color images. To solve this problem, a colorization operation for converting monochrome images into color images should be considered.

Back in early 2000s, models for colorization began. At first, researchers colorized a grayscale image by matching luminance and texture information between reference color images and the target grayscale image. Thus, the colorized results were deterministic, and relied heavily on the collection of reference images, which is referred to as a forward problem. We call this kind of method *transfer-based* [2, 5, 14, 23, 27, 37]. In addition, researchers proposed *scribble-based* colorization [8, 11, 22, 24, 31, 35, 38], which works by following users' guidance with manual intervention. For example, users first scribble regions of an image using different colors; then, the system colorizes the image based on the scribbles. This kind of method is useful in some scenarios, but not for medical image colorization. In this paper, we aim at development of *fully automatic colorization*, which takes a grayscale image as input, then outputs a colorized image directly, without any manual intervention.

A lot of work has proposed *fully automatic colorization* [3, 10, 15, 19, 28, 32, 34, 42]. Most of it was based on the deep neural network (DNN) [40, 41] and achieved high-quality image colorization. For example, Larsson et al. [19] proposed an end-to-end, fully automatic image colorization scheme by using a deep convolutional architecture, exploiting both low-level and semantic representations. Their system (at the time of writing) achieved a state-of-the-art ability to automatically colorize grayscale images. Zhang et al. [42] reported similar work on fully automatic colorization. In this paper, our proposed system falls into the category of fully automatic colorization using a DNN. However, the problem is that those previous works required each input and output pair of training examples to be both paired and aligned. But obtaining paired training data can be expensive and difficult, especially for medical images. In this paper, our proposed model can use unpaired image data for training.

With the development of generative adversarial networks (GANs) [21], Chia et al. [5] utilized a GAN to propose style conversion networks that can transfer an image style from another image, which also can be applied to transferring color to a grayscale image, or to synthesize a color image from a grayscale image [10, 15, 28, 32]. The CycleGAN [44], in particular, enables the learning of unpaired datasets by applying cycle-consistency. This advantage offers significant benefits when it is difficult to obtain a large amount of paired training data. Although the GAN has been applied to medical image processing for various purposes [39], research on its application to the colorization of medical images has not been conducted. In this paper, our work builds on the CycleGAN to colorize medical images, and achieves better performance by introducing the perceptual loss function [20] and the total variation (TV) loss function [35].

The contributions of this paper are as follows.

- We demonstrate that GAN-based style conversion can be applied to colorization of medical images.

- We demonstrate that the introduction of perceptual loss and TV loss can improve the quality of images produced as a result of colorization, and the result is better than the result generated by only using the CycleGAN model. Our model inherits the characteristics of CycleGANs that does not require the input and output training image data to be paired and aligned. This is a significant advantage when applying colorization to medical images.

This manuscript is organized as follows. In Section 2, we introduce related studies about scribble-based, transfer-based, and fully automatic colorization methods. In Section 3, we introduce our proposed unpaired GAN network and loss functions. Experimental results with a small dataset and a large set of medical images are presented in Section 4. Finally, conclusions are given in Section 5.

2 Related work

Colorization techniques [12] have been widely used in scientific illustrations, medical image processing, and old-photo colorization to enhance features and increase visual appeal. The large degree of freedom during the assignment of color information makes image colorization very challenging. Over the last decade, many approaches have been proposed to address colorization issues. Previous work falls into three broad categories, which are *scribble-based* [8, 11, 22, 24, 31, 35, 38], *transfer-based* [2, 5, 14, 23, 27, 37], and *fully automatic colorization* [19].

Scribble-based methods need users to scribble on the object grayscale image, and then put it through a coloring optimization algorithm [22]. The algorithm propagates the scribbled colors on the assumption that adjacent pixels with the same pixel values will have the same colors. The user needs to have enough knowledge about scribbling to paint enough of the color graffiti into the area. For ordinary users, it could be difficult to provide enough of the appropriate colors to achieve good results. To make the scribbling process easy and to generate better colorization results, Zhang et al. [43] proposed real-time, user-guided image colorization using a DNN. Because the colorization is performed in a single feed-forward pass, it enables real-time use.

Transfer-based methods rely on the availability of reference images in order to transfer those colors to the target grayscale image. It works well if there are enough similar reference images, and also requires less skill and effort from the user, compared to scribble-based methods. But sometimes a unique grayscale input image cannot match any similar reference images, which leads to a non-ideal result. Also, this requires building a large repository to store reference images at test time. A transfer-based colorization system uses a strategy that establishes mapping between source and target images by using correspondence between local descriptors [2, 27, 37], or in combination with the user's guidance [5, 14].

Fully automatic colorization is our goal, in contrast to the above methods. Grayscale images are the input for the system, which outputs the corresponding color images directly, without manual intervention. The fully automatic image colorization systems [3, 19, 34, 42] achieve high-quality performance that generates high-quality colorized images owing to the emergence and development of DNN technology. A lot of work has also tried to apply a Generative Adversarial Network (GAN) [9] to colorization, focusing on improving training stability and making robust color image synthesis in large multi-class image datasets [10, 15, 28, 32]. Nazeri et al. [28] applied a GAN to image colorization by redefining the generator cost function proposed in the first GAN model [9], and they also adopted the idea from the conditional generative adversarial network [26] to allow grayscale images to be the input for the generator, rather than randomly generated noise data. Sharma et al. [32] proposed robust image colorization using a self-attention–based progressive GAN, which consists of a residual encoder–decoder network and a self-attention–based progressive generative network in cascaded form to achieve robust image colorization. Górriz et al. [10] used conditional GAN, and improved training stability to enable better generalization in large multi-class image datasets. Isola et al. [15] proposed a framework, named *pix2pix*, which uses a conditional generative adversarial network to learn mapping from input to output images. Those works have a common drawback in that each input and output pair of training examples needs to be paired and aligned. However, obtaining paired training data can be expensive and difficult. Zhu et al. [44] proposed a CycleGAN built on *pix2pix* to learn the mapping without having paired training examples. Mehir et al. [25] proposed an architecture to colorize near infrared images. It learns the color channels on unpaired dataset, requires less computation times, converges faster and generates high quality samples. The advantage that uses unpaired training dataset offers significant benefits, given that it is difficult to obtain enough paired training examples, especially in medical images. Our proposal in this paper builds on the CycleGAN, but we achieve better performance by introducing the perceptual loss function and the TV loss function (Table 1).

3 Method

We introduced perceptual loss and TV loss into CycleGAN to colorize medical images. On one hand, our method inherits a main advantage from CycleGAN in that it can learn to translate an image from a source domain, I_{gray}, to a target domain, I_{color}, in the absence of paired examples while training the model. This characteristic is very practical if there is a lack of paired training data, especially in medical images. On the other hand, the experiment results show that our method outperforms the CycleGAN and Perceptual-based models. In this section, we first present the overall structure of our proposal. Then, we present in detail the objective functions of adversarial loss, cycle-consistency loss, perceptual loss, and TV loss. Finally, we present the final objective function of our proposal.

3.1 Network structure

We present the overall structure of the proposed technique in Fig. 1 (a). The technique is based on the structure of CycleGAN that enables learning of unpaired datasets, i.e., the domains of

Table 1 Overview of existing work in three broad categories. Our proposal belongs to unpaired *fully automatic colorization*

	Scribble-based	Transfer-based	Fully automatic colorization	
Existing works	[8, 11, 22, 24, 31, 35, 38]	[2, 5, 14, 23, 27, 37]	[3, 10, 15, 19, 28, 32, 34, 42]	[25, 44], Ours
Training Data	Paired	Paired	Paired	Unpaired (or paired)
Testing	Testing image	Testing + reference images	Testing image	Testing image
Automation	Semi-automatic	Automatic	Automatic	Automatic

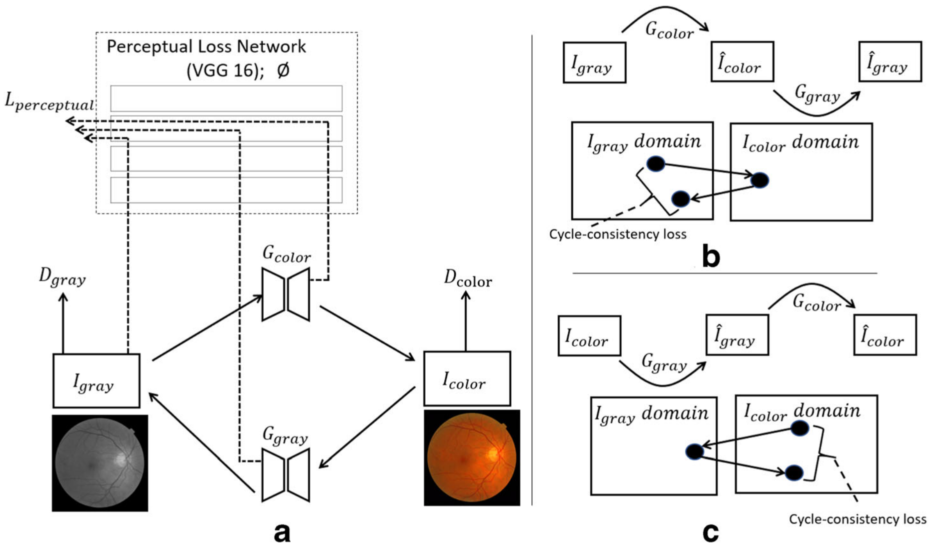

Fig. 1 **a** The overall structure of the proposed technique contains two map functions, $G_{color}: I_{gray} \rightarrow I_{color}$ and $G_{gray}: I_{color} \rightarrow I_{gray}$, associated with adversarial discriminators, $D_{gray}(I_{gray})$ and $D_{color}(I_{color})$, and a Perceptual Loss Network. With $\varnothing_i$, we indicate the feature map obtained by the i-th convolution (after activation) before the max-pooling layer within the VGG16 network. **b** Forward cycle-consistency loss: $I_{gray} \rightarrow G_{color}(I_{gray}) \rightarrow G_{gray}(G_{color}(I_{gray})) \approx I_{gray}$. **c** Backward cycle-consistency loss: $I_{color} \rightarrow G_{gray}(I_{color}) \rightarrow G_{color}(G_{gray}(I_{color})) \approx I_{color}$. We adopted (**b**) and (**c**) from [44]

I_{gray} and I_{color} are unpaired. The loss function of CycleGAN consists of the sum of the adversarial loss, which determines whether the input image is composite or not, and the cycle-consistency loss between the reconstruction image, which is created by restoring the composite image to the original image. The parameters of the perceptual loss network in Fig. 1(a) are pre-trained and untrainable. The functionality of the perceptual loss network is to generate features for three kinds of data. The first is I_{gray}, denoting the source domain or gray medical image to be colorized. The second is $G_{color}(I_{gray})$, and the third is $G_{gray}(G_{color}(I_{gray}))$, where G_{gray} is a generator network to generate gray images, and G_{color} is another generator network to generate color images (Fig. 2).

3.2 Objective functions

In this section, we present the details of the loss functions used in our proposed scheme. Those functions are essential, and they significantly affect the performance of the colorization system. There are four kinds of loss functions: adversarial loss, cycle-consistency loss, perceptual loss, and total variation loss, as described below. Finally, we combine these functions to get our full objective function.

Adversarial loss We adopted the adversarial loss from [9] and applied it to both mapping functions. For the mapping functions $G_{color}: I_{gray} \rightarrow I_{color}$ and its discriminator, D_{color}, we present the loss function as

$$\begin{aligned} L_{GAN}\left(G_{color}, D_{color}, I_{gray}, I_{color}\right) &= E[\log(D_{color}(I_{color}))] \\ &+ E\left[\log\left(1 - D_{color}\left(G_{color}\left(I_{gray}\right)\right)\right)\right] \end{aligned} \tag{1}$$

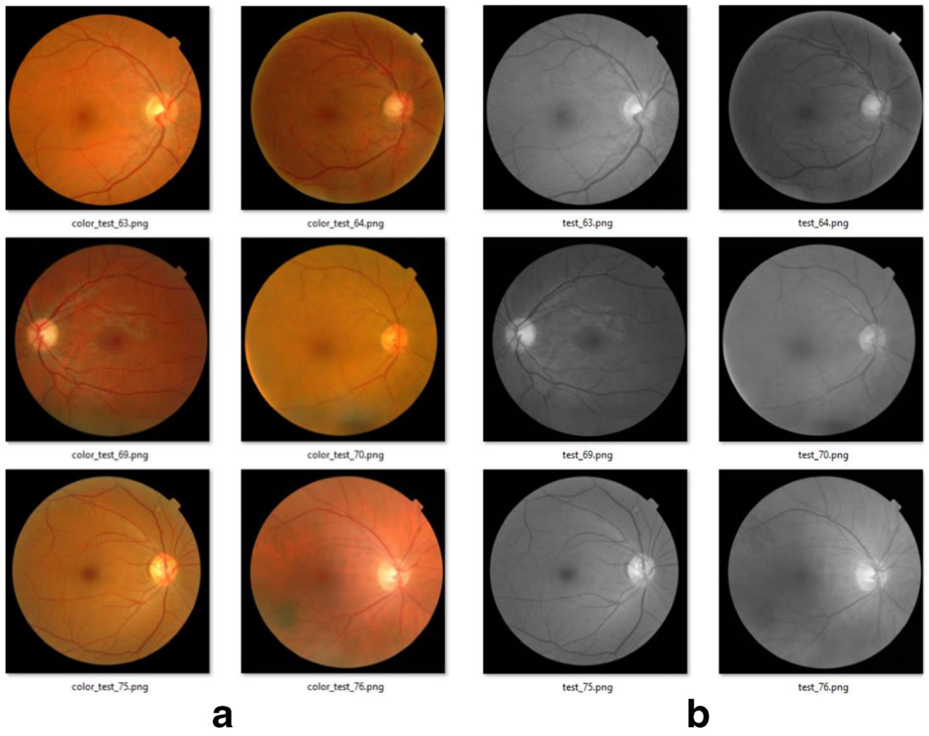

Fig. 2 Image samples of the Messidor dataset. **a** Colorful samples. **b** Gray samples

where G_{color} tries to generate images $\widehat{I}_{color} = G_{color}(I_{gray})$ that look similar to images from domain I_{color}, while D_{color} aims to minimize this objective against an adversary, D, that tries to maximize it, i.e., $min_{G_{color}}\ max_{D_{color}}\ L_{GAN}\ (G_{color}\ , D_{color}\ , I_{gray}\ , I_{color})$. We introduce a similar adversarial loss for mapping function G_{gray}: $I_{color} \rightarrow I_{gray}$ and its discriminator, D_{gray}, as well; i.e., $min_{G_{gray}}\ max_{D_{gray}}\ L_{GAN}\ (G_{gray}\ , D_{gray}\ , I_{color}\ , I_{gray})$.

Cycle-consistency loss We apply cycle-consistency loss [44] to transfer images from the I_{gray} domain to the I_{color} domain, and vice versa. The advantage from cycle-consistency loss is that I_{gray} and I_{color} can be unpaired, which is a very practical and useful characteristic when it is costly to label or mark the training data.

$$\begin{aligned} L_{cyc}(G_{color}, G_{gray}) = & E\left[\| G_{gray}\ (G_{color}\ (I_{gray})) - I_{gray} \|_1\right] \\ & + E\left[\| G_{color}(G_{gray}(I_{color})) - I_{color} \|_1\right] \end{aligned} \tag{2}$$

Perceptual loss Ledig et al. [20] used perceptual loss to improve the quality of the resulting image. Perceptual loss [17] introduces a pre-learned VGG16 model to an additional feature extraction network to extract the features of the composite image and the original image, and it then calculates the difference between them. In this experiment, calculation of the perceptual loss between the reconstructed image and the original image was added to the learning process.

$$L_{perceptual\_invert} = E\left[\| \varnothing(G_{color}(I_{gray})) - I_{gray} \|_1\right] \tag{3}$$

$$L_{perceptual\_cycle} = E\left[\left\|\varnothing\left(G_{gray}\left(G_{color}\left(I_{gray}\right)\right)\right)-I_{gray}\right\|_1\right] \tag{4}$$

where $\varnothing_i$ indicates the feature map obtained by the i-th convolution (after activation) before the max-pooling layer within the VGG16 network.

The final perceptual loss function is

$$L_{perceptual}\left(G_{color}, G_{gray}\right) = L_{perceptual\_invert} + L_{perceptual\_cycle} \tag{5}$$

Total variation loss TV loss [35], which has a smoothing effect, was added to prevent excessive contrast in the resultant image:

$$L_{TV}(Z) = \sum_{w,\ h}\left[\left(Z_{w,h+1}-Z_{w,h}\right)^2\left(Z_{w+1,h}-Z_{w,h}\right)^2\right]^{\frac{B}{2}} \tag{6}$$

where $Z = G_{color}(I_{gray})$, $B = 1$.

Full objective function Hence, we add all loss functions mentioned above and get our final loss function of the proposed technique, as follows:

$$\begin{aligned} L\left(G_{color}, D_{color}, G_{gray}, D_{gray}\right) = {} & L_{GAN}\left(G_{color}, D_{color}, I_{gray}, I_{color}\right) \\ & + L_{GAN}\left(G_{gray}, D_{gray}, I_{color}, I_{gray}\right) + L_{cyc}\left(G_{color}, G_{gray}\right) \\ & + L_{perceptual}\left(G_{color}, G_{gray}\right) + L_{TV}(Z) \end{aligned} \tag{7}$$

where $Z = G_{color}(I_{gray})$.

4 Experiment

4.1 Dataset description and preprocessing

Drive DRIVE [6] only contains 40 digital color retinal images. We divided them into two average sets, named *ColorA* and *ColorB*. Then, we converted all images in *ColorA* to grayscale and stored them in another set, named *GrayA*. We use *GrayA* and *ColorB* to train our model. When validating the trained model, we input *GrayA* to the model to get the predicted resulting images, which were used to compare with the images in *ColorA* to calculate performance metrics. Because of the small data size, we used *GrayA* to both train and validate the model. Ideally, the training set has no intersection with the test or validation sets. Therefore, to overcome this disadvantage, we also used another bigger dataset as follows.

Messidor The Messidor Database [15] contains 1200 digital color retinal images. We first cropped all images to make them square (1488 × 1488) and resized them to 256 × 256. Then, we shuffled them before dividing them into three sets (400 images each) named *TrainA*, *TrainB*, and *ValColor*. We converted the images in *TrainA* to grayscale, and copied all *ValColor* images, converting the copies into a grayscale dataset named *ValGray*. We use *TrainA* and *TrainB* to train our model, and used *ValColor* and ValGray to calculate performance metrics after training. Note that, while training the model, inputs I_{gray} and I_{color} are unpaired, which means not only is the color different, but the image texture is not aligned.

Also, training sets *TrainA* and *TrainB* had no intersection with validation sets *ValGray* or *ValColor*.

4.1.1 Training details

With the DRIVE dataset, we chose training parameter *epoch num* = *200* and decided that the learning rate scheduler would be *torch.optim.lr_scheduler.CosineAnnealingLR* from PyTorch, which reduced the learning rate over the training epochs and finally reached zero. These decisions were made based on experiments explained as follows. The DRIVE dataset only contains 40 images, which is very few. Thus, we needed to take care of the overfitting problem while training the model. First, we set up the parameters so that *batch size* = 1, *epoch num* = 2000, and constant *learning rate* = 0.002, and we got Fig. 3(c) and (d), which show that the values of the loss functions were very unstable over the training time, and while *epoch num* > 400, the values of the loss functions almost reached the lowest point. Secondly, we set the parameters *batch size* = 1 and *epoch num* = 200, and used a cosine-annealing kind of learning rate to get Fig. 3(a) and (b), which show that the values of the loss functions became much more stable when the training epoch was around 200. We also manually checked the predicted (generated) images after finishing the model training, finding that a model trained for 200 epochs generates color images with some additional, strange components, which means the model overfits. Therefore, we set the parameters as mentioned in the first sentence above, and applied them to all experiments with the DRIVE dataset.

On the Messidor dataset, we used Adam [18] with a learning rate of 2×10^{-3}, along with a cosine-annealing kind of scheduler to reduce the learning rate over the training epochs and

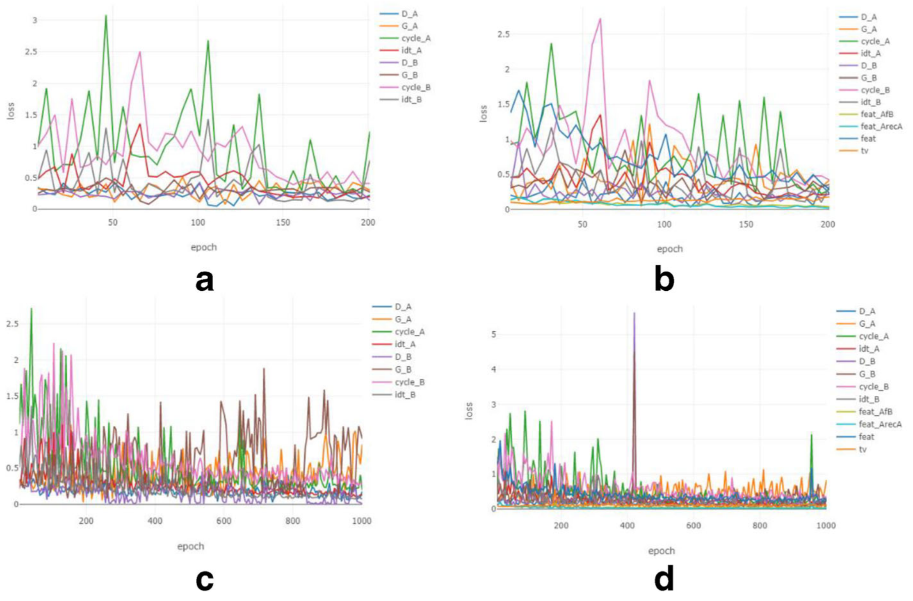

Fig. 3 With the DRIVE dataset, the training loss over time with respect to each loss function inside the relative model. **a** Epochs training the proposed model, **b** 200 epochs training the proposed model, **c** 1000 epochs training the proposed model, **d** 1000 epochs training the proposed model

finally reached zero. We resized all images to 256 × 256 and trained the networks with *batch size* = 2 for 200 iterations, giving 50 epochs over the training data. We did not use weight decay or a dropout technique. Our implementation used PyTorch [30] and cuDNN [4], taking roughly 1.5 h on a single GTX 1080 Ti GPU.

4.2 Results

4.2.1 Performance results on the DRIVE dataset

Experiment results in Table 2 compare performance among the CycleGAN and Perceptual models against our proposal. For each model, we performed the experiment three times, and each time, we used 20 ground truth images, and generated images to calculate metrics from NRMSE, PSNR, and SSIM. Lower NRMSE scores but higher PSNR and SSIM scores indicate better performance. Table 2 shows the mean values and the standard deviations of these metrics. CycleGAN applied GAN loss, cycle loss, and identity loss. The source code of its implementation in the PyTorch framework can be found in [13]. We added perceptual loss to CycleGAN, and kept other conditions unchanged, to be the second method in Table 2. The experiment results show that the Perceptual method has better performance than CycleGAN. Furthermore, we added perceptual loss plus TV loss to CycleGAN to be our proposal. The results show that the performance from our proposal is better than the other two.

4.2.2 Performance results with the Messidor dataset

Table 3 shows the NRMSE, PSNR, and SSIM performance metrics of the CycleGAN and Perceptual methods against our proposed model with the Messidor dataset, which indicates that the performance of our method is better than the other two. We chose *batch size* = 2, *epoch num* = 50, and initial *learning rate* = 0.002, using *torch.optim.lr_scheduler.CosineAnnealingLR* from PyTorch, which reduced the learning rate over the training epochs, and finally reached zero. Figure 4 shows the values of the loss functions over the training time, which indicate that the values of loss functions are not so stable, and around epoch 50, these loss values hardly improved at all. While training the models, we saved them at the end of each epoch. Thus, after finishing the training, we chose the best epoch that can colorize the best images but that does not overfit.

Table 2 A DRIVE dataset comparison of mean and standard deviations from NRMSE, PSNR, and SSIM

Method		NRMSE	PSNR	SSIM
CycleGAN	1	0.2535 (±0.1278)	24.6306 (±1.1259)	0.6966 (±0.0368)
	2	0.1820 (±0.0860)	24.6463 (±1.4262)	0.7056 (±0.0397)
	3	0.1781 (±0.0957)	25.8381 (±1.7402)	0.7164 (±0.0272)
	Avg	**0.2045** (±0.1032)	25.0383 (±1.4308)	0.7062 (±0.0346)
Perceptual	1	0.1866 (±0.1007)	25.7041 (±1.5534)	0.7221 (±0.0263)
	2	0.2115 (±0.1077)	25.3563 (±1.1254)	0.6942 (±0.0309)
	3	0.2065 (±0.1228)	26.8807 (±1.3953)	0.7148 (±0.0529)
	Avg	0.2015 (±0.1104)	25.9804 (±1.3580)	0.7104 (±0.0367)
Our proposal	1	0.2043 (±0.1137)	26.5571 (±1.2251)	0.7452 (±0.0387)
	2	0.1990 (±0.1247)	27.6569 (±1.2279)	0.7636 (±0.0350)
	3	0.1700 (±0.0943)	26.7060 (±1.2459)	0.7464 (±0.0436)
	Avg	0.1911 (±0.1109)	**26.9733** (±1.2330)	**0.7517** (±0.0391)

The maximum average value of each metrics (i.e., column) is presented in bold font

Table 3 With the Messidor dataset, a comparison of mean and standard deviations from NRMSE, PSNR, and SSIM. We added perceptual loss to CycleGAN to get the Perceptual (Third row). Furthermore, we added TV loss to Perceptual to get our proposal

Method	NRMSE	PSNR	SSIM
CycleGAN	**0.2302** (±0.0804)	22.1244 (±3.4041)	0.7382 (±0.0645)
Perceptual	0.2204 (±0.0732)	22.4707 (±3.4396)	0.7994(±0.0497)
Our proposal	0.1895 (±0.0932)	**24.1719** (±3.7804)	**0.8209** (±0.0547)

The maximum average value of each metrics (i.e., column) is presented in bold font

We present samples of the colorized results in Fig. 5. The first column is the gray input images to be colorized by the models. Columns 2 to 4 are the colorized results from the CycleGAN and Perceptual models, and our proposal, respectively. The final column is ground truth images. We trained these three models under the same conditions, including the same hyperparameters. We used the trained models from epoch number 50 to generate color images, and show some of them in Fig. 5. Two obvious disadvantages can be found in the second column, generated by CycleGAN. One is that there are rings, indicated with arrows. This is because the size of the retina varies in proportion to different training images. The second disadvantage is that the retina texture is blurred, and the color of the whole image is quite different from ground truth. The images in the third column in Fig. 5 show the results of the Perceptual model. The retinal texture is clear, and the color of the whole image looks like the ground truth image. However, the model generates slight and imperceptible rings, which is still much better than CycleGAN. Besides, it generated some strange components that do not belong in the training images. Except for that, when zooming in the bottom image in the third

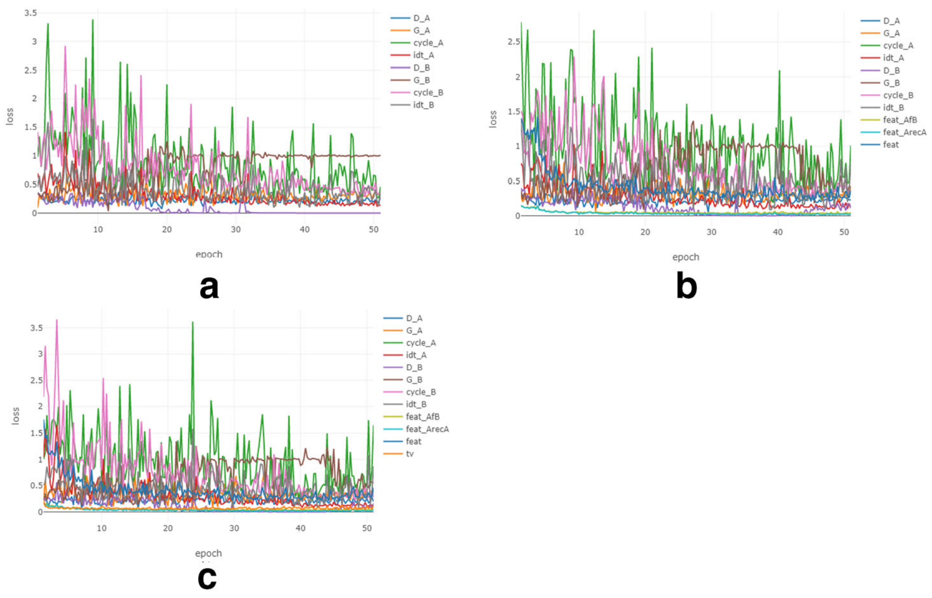

Fig. 4 With the Messidor dataset, values of loss functions of each model over the training time. **a** CycleGAN loss over time, **b** Perceptual loss over time, **c** Our proposal's loss over time

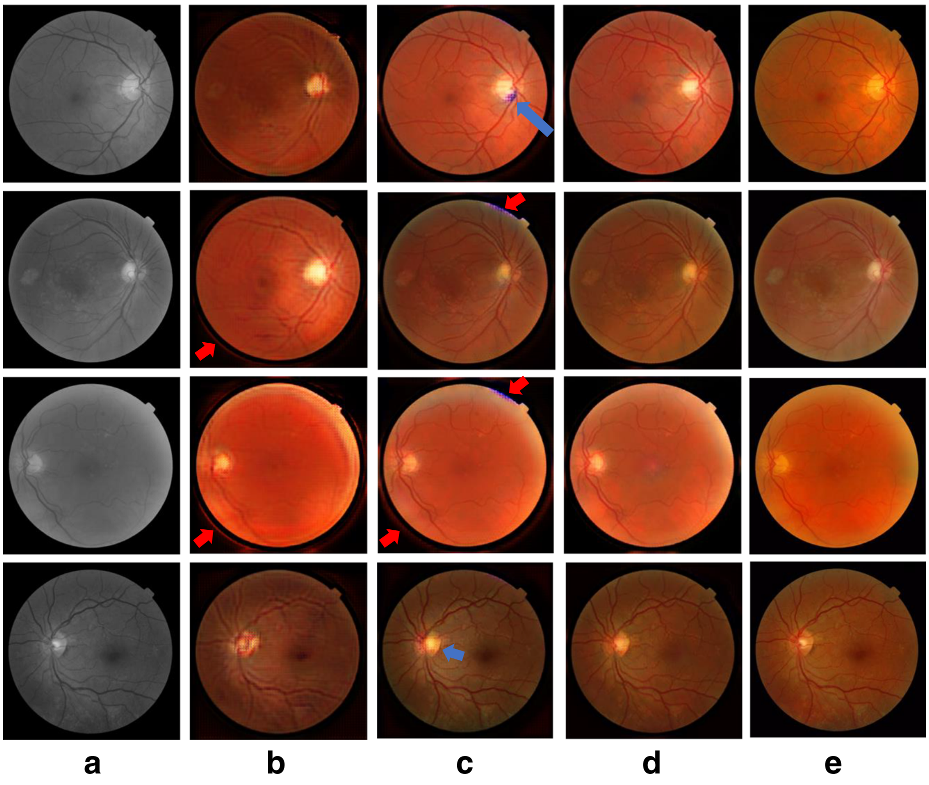

Fig. 5 From the Messidor dataset, colorization results of the CycleGAN, Perceptual methods, and the proposed method. **a** Gray input, **b** CycleGAN, **c** Perceptual, **d** Proposal, **e** Ground truth

column, we find that the area indicated by the arrow is not smooth enough, compared with ground truth and our proposal. We can see that the results generated by our proposed method are much better with regard to image color, retina texture, and smoothness.

4.2.3 Effect of identity loss

The implementation of CycleGAN applies identity loss [16] even though the paper on CycleGAN does not mention it, which specifically are $L_{idt\_gray} = \| G_{gray}(I_{gray}) - I_{gray} \|_1$ and $L_{idt\_color} = \| G_{color}(I_{color}) - I_{color} \|_1$. The identity loss indicates that if the input of G_{color} is a color image, its output should also be a color image that is identical to the input. (Generally, the job of G_{color} is to transform gray images into color images.) A similar job is true for G_{gray} with color images. We performed experiments to verify the performance contribution of identity loss. Without using the identity loss function in our proposed method, we performed the experiment three times to measure the NRMSE, PSNR, and SSIM metrics shown in Table 4. The results illustrate that the metric No Identity Loss is not better than With Identity Loss, but they are close. We furthermore manually checked their generated images, and show samples in Fig. 6. We found that the images for No Identity Loss are fragile, which tends to introduce some strange areas that do not exist in the ground truth images. Examples are in the second column and are indicated by the arrows. Intuitively, this is because identity loss can force the generated images to look like the input images.

Table 4 With the Messidor dataset, a comparison of mean and standard deviations from NRMSE, PSNR, and SSIM

Method		NRMSE	PSNR	SSIM
No Identity Loss	1	0.2102 (±0.1007)	23.2432 (±3.4917)	0.8140 (±0.0552)
	2	0.2105 (±0.1113)	23.3458 (±3.2989)	0.8010 (±0.0556)
	3	0.2115 (±0.0828)	23.0316 (±3.7527)	0.7871 (±0.0686)
	Avg	**0.2107** (±0.0983)	23.2069 (±3.5144)	0.8007 (±0.0598)
Our Proposal (With Identity Loss)	1	0.1950 (±0.1061)	24.0395 (±3.7498)	0.8193 (±0.0577)
	2	0.1875 (±0.0785)	24.0956 (±3.7618)	0.8117 (±0.0503)
	3	0.1860 (±0.0951)	24.3806 (±3.8297)	0.8318 (±0.0561)
	Avg	0.1895 (±0.0932)	**24.1719** (±3.7804)	**0.8209** (±0.0547)

The maximum average value of each metrics (i.e., column) is presented in bold font

4.2.4 Symmetric and asymmetric structures of perceptual loss

The structure of CycleGAN is symmetric, and our proposal was built upon it by introducing perceptual loss and TV loss. Thus, a natural thought is about the performance if the structure of perceptual loss is also symmetric. Specifically, we determine perceptual loss with Eq. (8), as seen below. Furthermore, we add the asymmetric part, $L_{perceptual\_invert}$, from Eqs. (3) to (8) to obtain Eq. (9).

$$\begin{aligned} L_{perceptual\_sym}\left(G_{color}, G_{gray}\right) &= L_{perceptual\_cycle} + E\left[\|\varnothing\left(G_{color}\left(G_{gray}\left(I_{color}\right)\right)\right) - I_{color}\|_1\right] \\ &= E\left[\|\varnothing\left(G_{gray}\left(G_{color}\left(I_{gray}\right)\right)\right) - I_{gray}\|_1\right] + E\left[\|\varnothing\left(G_{color}\left(G_{gray}\left(I_{color}\right)\right)\right) - I_{color}\|_1\right] \end{aligned} \quad (8)$$

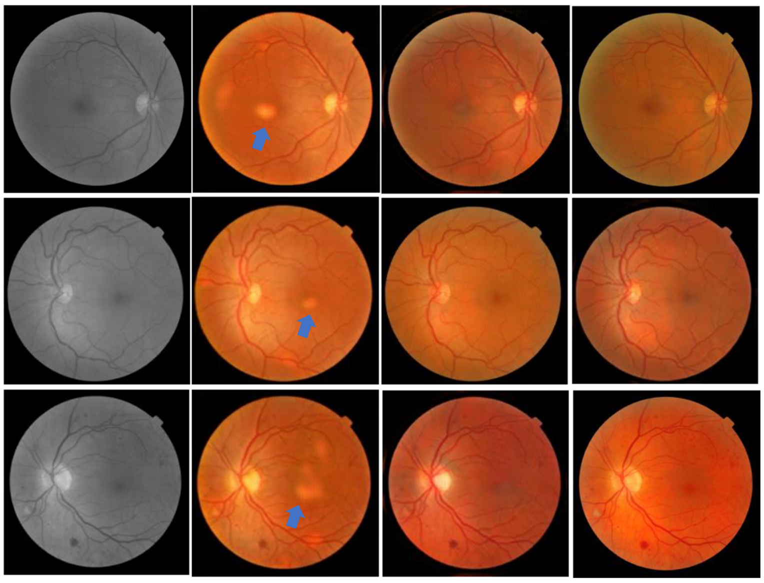

Fig. 6 The colorized results: with and without identity loss

Table 5 With the Messidor dataset, a comparison of mean and standard deviations from NRMSE, PSNR, and SSIM

Method		NRMSE	PSNR	SSIM
$L_{perceptual\_sym}$ (Eq. 8)	1	0.2049 (±0.0935)	23.3483 (±3.4961)	0.8089 (±0.0556)
	2	0.2214 (±0.0741)	22.4469 (±3.5212)	0.8036 (±0.0518)
	3	0.1923 (±0.0738)	23.7612 (±3.9130)	0.8292 (±0.0478)
	Avg	**0.2062** (±0.0804)	23.1855 (±3.6434)	0.8139 (±0.0518)
$L_{perceptual\_sym\_2}$ (Eq. 9)	1	0.1921 (±0.0809)	23.9011 (±3.8722)	0.8328 (±0.0526)
	2	0.1968 (±0.1094)	24.0546 (±4.0310)	0.8278 (±0.0527)
	3	0.2052 (±0.0770)	23.2302 (±3.7651)	0.7942 (±0.0466)
	Avg	0.1980 (±0.0891)	23.7287 (±3.8894)	0.8183 (±0.0506)
Our proposal ($L_{perceptual}$) (Eq. 5)	Avg	0.1895 (±0.0932)	**24.1719** (±3.7804)	**0.8209** (±0.0547)

The maximum average value of each metrics (i.e., column) is presented in bold font

$$L_{perceptual\_sym\_2}\left(G_{color}, G_{gray}\right) = \mathrm{L}_{perceptual\_sym}\left(G_{color}, G_{gray}\right) + L_{perceptual\_invert} \tag{9}$$

Currently, three options can be the perceptual loss in our models: Eqs. (5), (8), and (9). We performed experiments to verify the performance for each of them. Table 5 shows the results from the NRMSE, PSNR, and SSIM performance metrics. The performance is close, but our proposal is the best. Additionally, we generated color images with a trained model by using $L_{perceptual\_sym}$, and we manually checked the quality of the colorized images, from which we find that most of them have redundant spots that do not exist in the corresponding ground truth images. Figure 7 shows samples of these images, with the abnormal areas indicated by the arrows. Although the performance of $L_{perceptual\_sym\_2}$ and $L_{perceptual}$ are close, based on Table 5, Occam's razor suggests it is better to not use more than is necessary. Therefore, $L_{perceptual}$ is better because it is simpler.

5 Conclusion

A DNN model for colorization is data-driven, and training millions of parameters of the model needs a large amount of labeled or paired image training data. In this paper, we focused on solving the problems that 1) The colorized medical image should have the same features as the input image; 2) It is difficult and expensive to obtain many paired training data of medical images. We demonstrated that our model could use unpaired medical images to train a colorization system to synthesize high-quality color medical images. Our model is built on CycleGAN, introducing perceptual loss and TV loss. Our model has almost the same training

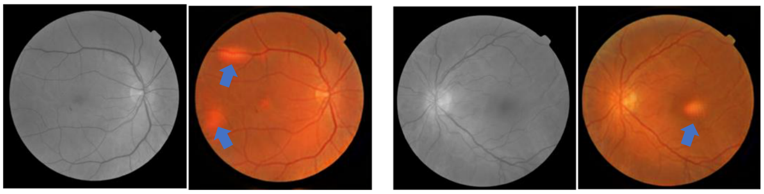

Fig. 7 Colorized results of Symmetric Perceptual Loss. The gray images are the input, while the color images are the results of the model using Symmetric Perceptual Loss

and testing process complexity, but the experiment results show our model has better performance than CycleGAN in colorizing medical images.

Acknowledgments This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korea government (No.NRF-2019R1A2C1090713). This research was supported by the BK21 Four Program (Pioneer Program in Next-generation Artificial Intelligence for Industrial Convergence, 5199991014250) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF).

References

1. Anwar S, Tahir M, Li C, Mian A, Khan FS, Muzaffar AW (Nov. 2020) Image colorization: a survey and dataset, arXiv:2008.10774 [cs, eess]. Accessed: Nov. 09, 2020. [Online]. Available: http://arxiv.org/abs/2008.10774.
2. Charpiat G, Hofmann M, Schölkopf B (2008) Automatic image colorization via multimodal predictions. in European conference on computer vision:126–139
3. Cheng Z, Yang Q, Sheng B (2015) Deep colorization. in Proceedings of the IEEE International Conference on Computer Vision:415–423
4. Chetlur S et al. (2014) cudnn: Efficient primitives for deep learning. arXiv preprint arXiv:1410.0759
5. Chia AY-S, Zhuo S, Gupta RK, Tai YW, Cho SY, Tan P, Lin S (Dec. 2011) Semantic colorization with internet images. ACM Trans Graph 30(6):1–8. https://doi.org/10.1145/2070781.2024190
6. DRIVE - Grand Challenge, grand-challenge.org. https://drive.grand-challenge.org/ (accessed Jul. 27, 2020).
7. Fukumoto Y, Hiro T, Fujii T, Hashimoto G, Fujimura T, Yamada J, Okamura T, Matsuzaki M (2008) Localized elevation of shear stress is related to coronary plaque rupture: a 3-dimensional intravascular ultrasound study with in-vivo color mapping of shear stress distribution. J Am Coll Cardiol 51(6):645–650
8. Furusawa C, Hiroshiba K, Ogaki K, Odagiri Y (2017) Comicolorization: semi-automatic manga colorization. in SIGGRAPH Asia 2017 Technical Briefs:1–4
9. Goodfellow I et al (2014) Generative adversarial nets. Adv Neural Inf Proces Syst:2672–2680
10. Górriz M, Mrak M, Smeaton AF, O'Connor NE (2019) End-to-end conditional GAN-based architectures for image colourisation. In: 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP), 27–29 Sept 2019, Kuala Lumpur, Malaysia, pp 1–6
11. Huang Y-C, Tung Y-S, Chen J-C, Wang S-W, Wu J-L (2005) An adaptive edge detection based colorization algorithm and its applications. in Proceedings of the 13th annual ACM international conference on Multimedia:351–354
12. Iizuka S, Simo-Serra E, Ishikawa H (2016) Let there be color! Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM Transactions on Graphics (ToG) 35(4):1–11
13. Image-to-Image Translation in PyTorch, https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix, Accessed 26 July, 2020
14. Ironi R, Cohen-Or D, Lischinski D (2005) Colorization by example. in Rendering Techniques:201–210
15. Isola P, Zhu J-Y, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. in Proceedings of the IEEE conference on computer vision and pattern recognition:1125–1134
16. Jing Y, Yang Y, Feng Z, Ye J, Yu Y, Song M (2019) Neural style transfer: a review. IEEE transactions on visualization and computer graphics
17. Johnson J, Alahi A, Fei-Fei L (2016) Perceptual losses for real-time style transfer and super-resolution. in European conference on computer vision:694–711
18. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980

19. Larsson G, Maire M, Shakhnarovich G (2016) Learning representations for automatic colorization. in European conference on computer vision:577–593
20. Ledig C et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. in Proceedings of the IEEE conference on computer vision and pattern recognition:4681–4690
21. Lee D-H, Li Y, Shin B-S (Apr. 2020) Generalization of intensity distribution of medical images using GANs. Human-centric Computing and Information Sciences 10(1):17. https://doi.org/10.1186/s13673-020-00220-2
22. Levin A, Lischinski D, Weiss Y (2004) Colorization using optimization. in ACM SIGGRAPH 2004 Papers: 689–694
23. Li B, Lai Y, John M, Rosin PL (Sep. 2019) Automatic example-based image colorization using location-aware cross-scale matching. IEEE Trans Image Process 28(9):4606–4619. https://doi.org/10.1109/TIP.2019.2912291
24. Luan Q, Wen F, Cohen-Or D, Liang L, Xu Y-Q, Shum H-Y (2007) Natural image colorization. in Proceedings of the 18th Eurographics conference on Rendering Techniques:309–320
25. Mehri A, Sappa AD (Jun. 2019) Colorizing near infrared images through a cyclic adversarial approach of unpaired samples, in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, pp. 971–979. https://doi.org/10.1109/CVPRW.2019.00128.
26. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784
27. Morimoto Y, Taguchi Y, Naemura T (2009) Automatic colorization of grayscale images using multiple images on the web. in SIGGRAPH 2009: Talks:1–1
28. Nazeri K, Ng E, Ebrahimi M (2018) Image colorization using generative adversarial networks. in International conference on articulated motion and deformable objects:85–94
29. Park Y-S, Lee J-W (2020) Class-labeling method for designing a deep neural network of capsule endoscopic images using a lesion-focused knowledge model. J Inf Process Syst 16(1):171–183
30. PyTorch. https://www.pytorch.org (accessed Jul. 27, 2020).
31. Qu Y, Wong T-T, Heng P-A (2006) Manga colorization. ACM Transactions on Graphics (TOG) 25(3): 1214–1220
32. Sharma M et al (2019) Robust image colorization using self attention based progressive generative adversarial network. in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW):2188–2196
33. Song Q, Xu F, Jin Y-Q (2017) Radar image colorization: converting single-polarization to fully polarimetric using deep neural networks. IEEE Access 6:1647–1661
34. Su J-W, Chu H-K, Huang J-B (2020) Instance-aware image colorization. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition
35. Taigman Y, Polyak A, Wolf L (2016) Unsupervised cross-domain image generation. arXiv preprint arXiv: 1611.02200
36. Wan S, Xia Y, Qi L, Yang Y, Atiquzzaman M (Jul. 2020) Automated colorization of a Grayscale image with seed points propagation. IEEE Transactions on Multimedia 22(7):1756–1768. https://doi.org/10.1109/TMM.2020.2976573
37. Welsh T, Ashikhmin M, Mueller K (2002) Transferring color to greyscale images. in Proceedings of the 29th annual conference on Computer graphics and interactive techniques:277–280
38. Yatziv L, Sapiro G (2006) Fast image and video colorization using chrominance blending. IEEE Trans Image Process 15(5):1120–1129
39. Yi X, Walia E, Babyn P (2019) Generative adversarial network in medical imaging: a review. Med Image Anal 58:101552
40. You SD, Liu C-H, Chen W-K (Nov. 2018) Comparative study of singing voice detection based on deep neural networks and ensemble learning. Human-centric Computing and Information Sciences 8(1):34. https://doi.org/10.1186/s13673-018-0158-1
41. Yu N, Yu Z, Gu F, Li T, Tian X, Pan Y (Apr. 2017) Deep learning in genomic and medical image data analysis: challenges and approaches. Journal of Information Processing Systems 13(2):204–214
42. Zhang R, Isola P, Efros AA (2016) Colorful image colorization. in European conference on computer vision:649–666
43. Zhang R, et al. (2017) Real-time user-guided image colorization with learned deep priors. arXiv preprint arXiv:1705.02999
44. Zhu J-Y, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. in Proceedings of the IEEE international conference on computer vision:2223–2232