Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 19(1): 86–101.
DOI: 10.3934/mbe.2022004
Received: 02 September 2021
Accepted: 27 October 2021
Published: 08 November 2021

Research article

An improved pix2pix model based on Gabor filter for robust color image rendering

Hong-an Li, Min Zhang*, **Zhenhua Yu, Zhanli Li and Na Li**

College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China

* **Correspondence:** Email: 19208049009@stu.xust.edu.cn; Tel: +8613720707860.

Abstract: In recent years, with the development of deep learning, image color rendering method has become a research hotspot once again. To overcome the detail problems of color overstepping and boundary blurring in the robust image color rendering method, as well as the problems of unstable training based on generative adversarial networks, we propose an color rendering method using Gabor filter based improved pix2pix for robust image. Firstly, the multi-direction/multi-scale selection characteristic of Gabor filter is used to preprocess the image to be rendered, which can retain the detailed features of the image while preprocessing to avoid the loss of features. Moreover, among the Gabor texture feature maps with 6 scales and 4 directions, the texture map with the scale of 7 and the direction of 0° has the comparable rendering performance. Finally, by improving the loss function of pix2pix model and adding the penalty term, not only the training can be stabilized, but also the ideal color image can be obtained. To reflect image color rendering quality of different models more objectively, PSNR and SSIM indexes are adopted to evaluate the rendered images. The experimental results of the proposed method show that the robust image rendered by this method has better visual performance and reduces the influence of light and noise on the image to a certain extent.

Keywords: image rendering; gabor filter; pix2pix model; generative adversarial networks; penalty term; least squares loss function

1. Introduction

At present, image color rendering as a major branch of image processing has attracted much attention. With the development of deep learning, image color rendering based on neural network has gradually become a research hotspot [1–5]. Because traditional color rendering methods require manual intervention and have high requirements of reference images. Moreover, when the structure and color of the image are complex, color rendering effect is not ideal [6–10]. Color rendering methods based on deep learning can be easily deployed in the actual production environment, and the limitation of the traditional methods can be solved [11–13]. By using the neural network model and the corresponding dataset training model [14, 15], the image can be automatically rendered according to the model, without being affected by human or other factors [16–19].

Larsson [20] used the convolutional neural network to consider the brightness of the image as input, decomposed the color and saturation of the image by the super-column model, to realize color rendering. Iizuka [21] combined the low-dimensional feature and global feature of the image by using the fusion layer in the convolutional neural network, for generating the color of the image and processing images of any resolution. Zhang [22] designed an appropriate loss function to handle the multi-mode uncertainty in color rendering and maintain the color diversity. However, when the grayscale image features are extracted using the above mentioned method, up-sampling is adopted to make the image size consistent, resulting in the loss of image information. Moreover, the network structure cannot well extract and understand the complex features of the image, and the rendering effect is limited [23–25].

Isola [26] improved conditional generative adversarial networks (CGAN) to achieve the transformation between images. The proposed pix2pix model can realize conversion between different images, for example, color rendering can be realized by learning the mapping relationship between grayscale image and color image [27, 28]. But the pix2pix model based generative adversarial networks (GAN) has the disadvantage of training instability. Moreover, the current image rendering methods based on deep learning are not good at rendering robust images. Gabor filter can easily extract texture information in all scales and directions of the image, and reduce the influence of light change and noise in the image to a certain extent.

Therefore, we propose a color rendering method using Gabor filter based improved pix2pix for robust image. The contributions of this paper are mainly there-folds:

(1) The improved pix2pix model can not only automatically complete image rendering and achieve good visual effect, but also achieve more stable training and better image quality.

(2) Gabor filter was added to enhance the robustness of model rendered images.

(3) The metric data of a series of experiments show that the proposed method has better performance for robust image.

The rest of the paper is organized as follows. Section 2 introduces the previous work, including Gabor filter and pix2pix model. Section 3 describes the method and its design details. Section 4 introduces the experiment and comparison experiment, and evaluates the image quality. Section 5 conclusions the paper and outlooks the future work.

2. Related works

2.1. Gabor filter

Fourier transform is a powerful tool in signal processing, which can help us transform images from spatial domain to frequency domain, and extract features that are not easy to extract in spatial domain. However, after Fourier transform, frequency features of images at different locations are often mixed together, but Gabor filter can extract spatial local frequency features, which is an effective texture detection tool [29, 30]. The Gabor filter is derived by multiplying a Gaussian by a cosine function [31–33], it is defined as

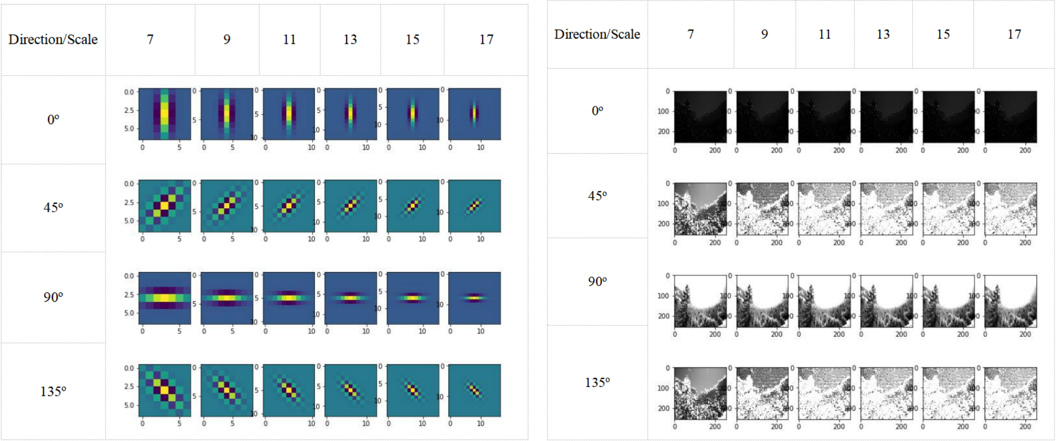
$$g(x, y, \lambda, \theta, \varphi, \sigma, \gamma) = exp(\frac{-x^{'2} + \gamma^2 y^2}{2\sigma^2})exp(i(2\pi\frac{x^{'}}{\lambda} + \varphi)) \tag{2.1}$$

$$g_{real}(x, y, \lambda, \theta, \varphi, \sigma, \gamma) = exp(\frac{-x^{'2} + \gamma^2 y^2}{2\sigma^2})\cos(i(2\pi\frac{x^{'}}{\lambda} + \varphi)) \tag{2.2}$$

$$g_{imag}(x, y, \lambda, \theta, \varphi, \sigma, \gamma) = exp(\frac{-x^{'2} + \gamma^2 y^2}{2\sigma^2})\sin(i(2\pi\frac{x^{'}}{\lambda} + \varphi)) \tag{2.3}$$

where, $x^{'} = x\cos\theta + y\sin\theta, y^{'} = -x\sin\theta + y\cos\theta$. Where, x, y represent the coordinate position of the pixel, λ represents the wavelength of the filter, θ represents the tilt degree of the Gabor kernel image, φ represents the phase offset, σ represents the standard deviation of the Gaussian function, and γ represents the aspect ratio.

In order to make full use of the characteristics of Gabor filters, it is necessary to design Gabor filters with different directions and scales to extract features. In this study, the Gabor filter extracts the texture features of the image in 6 scales and 4 directions. Namely, the Gabor scales are 7, 9, 11, 13, 15 and 17. The Gabor directions are 0°, 45°, 90° and 135°, as shown in Figure 1(a). Extract effective texture feature sets from the output results of the filter. The extracted texture feature sets are shown in Figure 1(b), with 24 texture feature maps in total.



(a) Gabor filter

(b) Texture feature sets

Figure 1. Gabor filter.

2.2. *pix2pix model*

At present, image rendering based on generative adversarial networks [34] attracts much attention because it can directly generate color images by using mapping relations. Therefore, it is widely used in image processing, text processing, natural language processing and other fields. pix2pix model [26] is a model for image-to-image conversion based generative adversarial networks. It can better synthesize image or generate color image. The following are the main features of the pix2pix model.

(1) Both the generator and discriminator structure use the convolution unit of Conv-Batchnorm-ReLU, namely, convolutional layer, batch normalization and ReLU Loss are used.

(2) The input of the pix2pix model is the specified image, such as the label image to the real image, the input is the label image, the input is the grayscale image to the color image, and the input is the grayscale image. The grayscale image as the input of the generator, and the input and output of the generator as the input of the discriminator, so as to establish the corresponding relationship between the input image and the output image, realize user control, and complete image color rendering.

(3) PatchGAN was used as discriminator for pix2pix model. Specifically, the image is divided into several fixed-size blocks, and the authenticity of each block is determined. Finally, the average value is taken as the final output. A network structure similar to U-net is adopted as a generator, and skip connections are added between i and $n - i$ at each layer to simulate U-net, where n is the total number of layers of the network. Not only can the path be shrunk for context information, but the symmetric extension path can be positioned precisely.

(4) The loss function of the pix2pix model is as follows, which is composed of L1 loss and Vanilla GAN loss. Where, let x be the input image, y be the expected output, G be the generator, and D be the discriminator:

$$G^* = argmin_G max_D L_{cGAN}(G, D) + \lambda L_{L1}(G) \tag{2.4}$$

$$L_{cGAN}(G, D) = E_{x,y}(logD(x, y)) + E_x(log(1 - D(x, G(x)))) \tag{2.5}$$

$$L_{L1}(G) = E_{x,y}(\| y - G(x)_1 \|) \tag{2.6}$$

3. Proposed method

3.1. Method framework

In view of the detail problems existing in the generative adversarial networks based image color rendering method in complex scenes, this paper proposes an image color rendering method using Gabor filter based improved pix2pix for robust image. The network framework is shown in Figure 2. The rendering process is shown in Figure 3. After selecting the data set for training, the trained generator is used for color rendering.

Firstly, we preprocessed the image with Gabor filter, and extracted the texture feature set of the image as input for training and verification. By comparing 24 Gabor texture feature maps with 6 scales and 4 directions, the texture map with 7 scales and 0° direction has the best color rendering effect. Secondly, this paper utilizes the existing pix2pix model architecture for image transformation to perform color rendering by learning the mapping relationship between grayscale image and color image. Finally, although the pix2pix model solves some problems existing in the generative adversarial networks, it still has the instability problem of training on large-scale image dataset. Therefore, the least square loss in LSGAN [35] is used in the objective function of pix2pix model, and the penalty term similar to WGAN_GP [36] is added. We improve the overall model framework, it is shown that the proposed method has a better performance on the rendering of robust images by a series of comparison experiments.

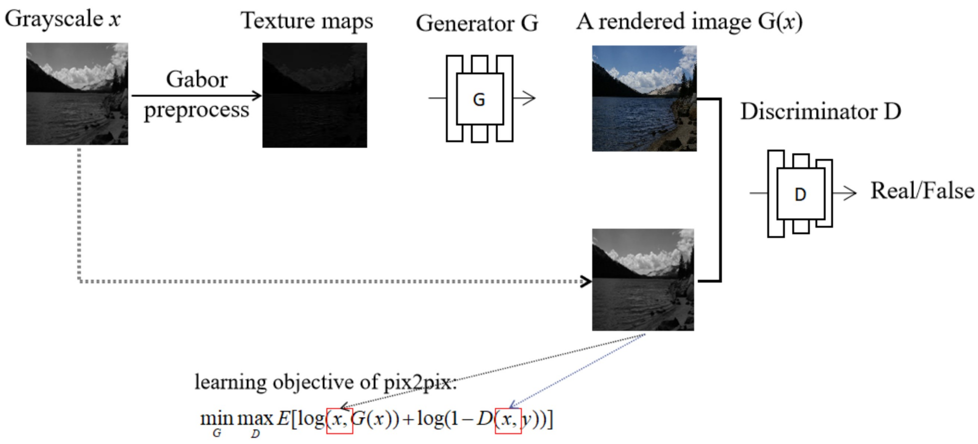


Figure 2. Method framework.

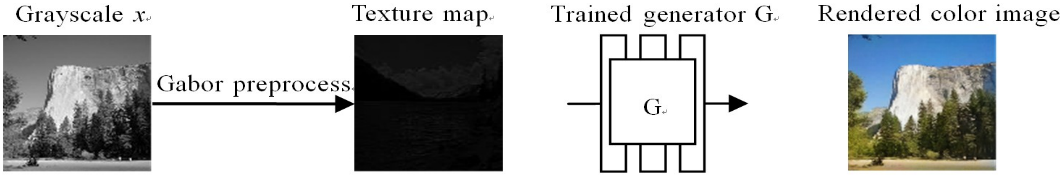


Figure 3. Rendering procedure.

3.2. Improved pix2pix model

The generator in generative adversarial networks hopes that the output data distribution can be more close to the distribution of the real data. Meanwhile, the discriminator of generative adversarial networks needs to make a judgment between the real data and the output data by the generator to find the real data and the fake data. The loss function can generate more real data through the Lipschitz constraint generative adversarial networks. The traditional generative adversarial networks uses the cross entropy loss or Vanilla GAN loss as the loss function. The classification is correct, but gradient dispersion occurs when the generator is updated [36,37]. LSGAN uses the square loss as the objective function, and the least square loss function penalizes the samples (fake samples) that are in the discriminant true but far away from the decision boundary, and drags the false samples far away from the decision boundary into the decision boundary, to improve the quality of the generated image.

Therefore, compared with the traditional generative adversarial networks, the image generated by LSGAN has higher quality and a more stable training process. So the least square loss function is adopted in the framework of this paper.

$$\begin{cases} min_D V_{LSGAN}(D) = \frac{1}{2} E_{x\ P_{data}(x)[(D(x)-b)^2]} + \frac{1}{2} E_{z\ P_z(z)[(D(G(z))-a)^2]} \\ min_D V_{LSGAN}(D) = \frac{1}{2} E_{z\ P_z(z)[(D(G(z))-c)^2]} \end{cases} \tag{3.1}$$

where, the input image is x, expected output is y, generator is G, discriminator is D, noise is z, labels of generated sample and real sample are a and b, respectively. c is the value set by the generator to let the discriminator think the generated image is real data.

Generative adversarial networks can generate better data distribution, but it has the problem of training instability. Improving the training stability of generative adversarial networks is a hot topic in deep learning. Wasserstein generative adversarial networks (WGAN) [38] uses Wasserstein distance to generate a value function with better theoretical properties than JS divergence in order to constrain the Lipschitz constant of the discriminator function, which basically solves the problems of generative adversarial networks training instability and model collapse and ensures the diversity of generated samples [39]. WGAN_GP continues to improve on WGAN, and its penalty term is derived from the Wasserstein distance, where the penalty coefficient is 10.

The objective function of WGAN_GP is as follows, adding the original critic loss and the gradient penalty term of WGAN_GP.

$$L = \underset{\widetilde{x}\sim P_g}{E}[D(\widetilde{x})] - \underset{\widetilde{x}\sim P_r}{E}[D(x)] + \lambda \underset{\widetilde{x}\sim P_{\widehat{x}}}{E}[(\| \nabla_{\widehat{x}} D(\widehat{x} \|_2 - 1)^2] \tag{3.2}$$

where, $\underset{\widetilde{x}\sim P_g}{E}[D(\widetilde{x})] - \underset{\widetilde{x}\sim P_r}{E}[D(x)]$ is the original critic loss, $\lambda \underset{\widetilde{x}\sim P_{\widehat{x}}}{E}[(\| \nabla_{\widehat{x}} D(\widehat{x} \|_2 - 1)^2]$ denotes the gradient penalty term of WGAN_GP, $\widehat{x} = t\widehat{x} + (1-t)x$, $0 \le t \le 1$, and λ is the penalty coefficient.

4. Experiments

To verify the effectiveness and accuracy of the proposed method, we conducted extensive experiments on summer dataset [40], with 1231 pieces of train set, and 309 pieces of test set. Experiment 1 is conducted to test the effect of application of the Gabor filter and different objective functions in the pix2pix model environment. Experiment 2 is performed to test the rendering effect when different Gabor texture feature maps are given as input. Experiment 3 is conducted to test whether the penalty term should be added in the discriminator. Experiment 4 is the rendering effect of low-quality or robust images was tested by adding noise and dimming the brightness of the image for assessing the robustness of this model.

Training parameters: The experiment was performed on a PC with Intel(R) Core(TM) i7-9750H CPU @ 2.6 GHz 2.59 GHz, a graphics card NVIDIA GeForce GTX 1650, and CUDA+Cudnn for acceleration training. The proposed method is implemented based on Python 3.7 and Pytorch framework. The number of experimental training iterations is 200, optimizer is Adam, batch_size is 1, learning rate is 0.0002, and number of processes is 4.

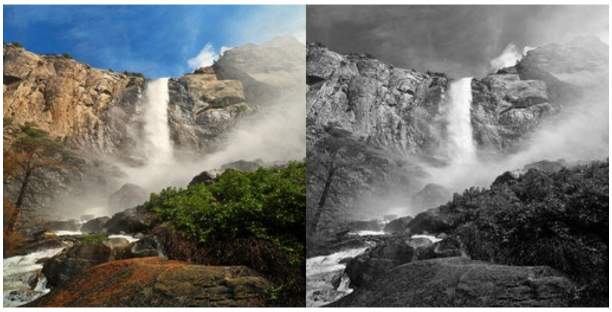


Figure 4. Spliced input graph with 512 × 256.

Network structures and implementation details: All the models we train are designed to 256 × 256 images. The input image of the model is 512 × 256; the left one is the original color image, and the right one is the texture feature map processed by the Gabor filter, as shown in Figure 4. By default, the pix2pix model uses a generator similar to U-net, PatchGAN, and Vanilla GAN loss.

Evaluation Metrics: To reflect image color rendering quality of different models more objectively, peak signal to noise ratio (PSNR) and structural similarity (SSIM) indexes are adopted to evaluate the rendered images [41, 42]. These two indexes are often used in the evaluation metrics of image processing. PSNR is an objective standard to evaluate the quality of the color image produced. The calculation formula is as follows:

$$PSNR = 10log_{10}\frac{(2^n - 1)^2}{MSE} \tag{4.1}$$

$$MSE = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}[X(i, j) - Y(i, j)]^2 \tag{4.2}$$

where, H and W represent the width and height of the image respectively, (i, j) represents each pixel point, and n represents the number of bits of the pixel, X and Y represent two images respectively.

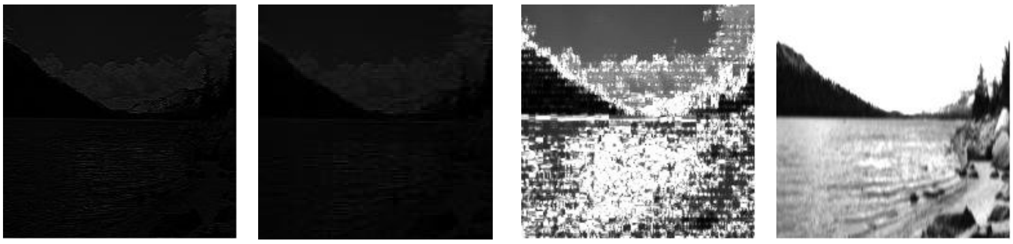
Because PSNR index also has its limitations, it cannot completely reflect the consistency of image quality and human visual effect, so SSIM index is used for further comparison. SSIM is a metric to measure the similarity of two images. By comparing the image rendered by the model with the original color image, the effectiveness and accuracy of this algorithm are demonstrated. The calculation formula is as follows:

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2\mu_y^2 + c_1)(\sigma_x^2\sigma_y^2 + c_2)} \tag{4.3}$$

where, μ_x and μ_y respectively represent the average value of the real image and the generated image, σ_x^2 and σ_y^2 respectively represent the variance of the real image and the generated image, σ_{xy} represents the covariance of the real image and the generated image, $c_1 = (k_1, L)^2$ and $c_2 = (k_2, L)^2$ are constants that maintain stability, and L is the dynamic range of pixel value, $k_1 = 0.01$, $k_2 = 0.03$.

4.1. Using Gabor filters and different objective function

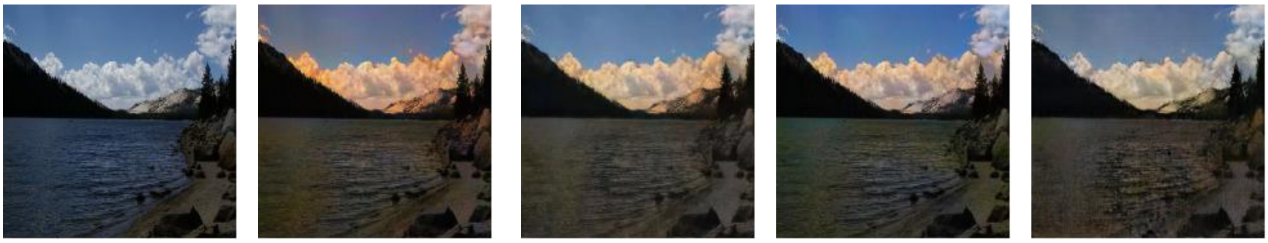
In this study, the Gabor filter extracts the texture features of the image in 6 scales and 4 directions. For convenience, according to the texture feature set shown in Figure 1(b), the images are numbered from left to right and from top to bottom. The direction is assumed to be d and the size is s, as shown in Figure 5. For example, G1 means "$s = 7$, $d = 0°$". So the direction is 0° and the scale is 7. G6 means "$s = 17$, $d = 0°$". So the direction is 0° and the scale is 17. By default, the pix2pix model uses Vanilla GAN loss. Based on pix2pix model, the model using least squares loss function is called LSpix (least squares pix2pix). Based on Gabor filter, the model using Gabor texture maps is called pixGn (pix2pix Gabor n), n = 1, 6, 7, 13.



(a) G1 : s = 7, d = 0° (b) G6 : s = 17, d = 0° (c) G7 : s = 7, d = 45° (d) G13 : s = 7, d = 90°

Figure 5. Texture feature map with Gabor.

To test the effect of application of the Gabor filter and different objective functions in the pix2pix model environment, we divided the experiment into adding Gabor filters (Figures 6(c),(e)), not adding Gabor filters (Figures 6(b),(d)), using least squares loss (Figures 6(d),(e)) or Vanilla GAN loss (Figures 6(b),(c)). By comparing the images in Figure 6, it can be confirmed that the rendering effect preprocessed by least square loss or Gabor filter is better, which is the LSpixG1 model. This is because Gabor can preprocess images and obtain multi-scale and multi-direction features of images, so as to achieve good and fast feature extraction and learning during network model learning. Moreover, compared with other loss functions, the least square loss function only reaches saturation at one point, which is not easy to cause the problem of gradient disappearance.



(a) ground truth (b) pix2pix (c) pixG1 (d) LSpix (e) LSpixG1

Figure 6. Effect of Gabor filter and use of different objective functions.

Table 1. PSNR index of different models (dB).

Network	MAX PSNR	MIN PSNR	AVE PSNR
pix2pix	29.204	11.126	23.024
pixG1	28.225	10.477	19.981
LSpix	**32.795**	11.003	**24.107**
pixG6	27.874	9.883	20.012
LSpixG6	32.616	10.632	21.409
LSpixG1	32.524	**11.238**	21.354

Note: Bold font is the best value for each column.

Tables 1 and 2 compare the distortion and structural similarity between the rendered image and the ground truth, show the maximum, minimum, and average indexes. This is an additional interpretation of Figure 6. The LSpix model has the highest score in the maximum and average PSNR, which is 3.591dB and 1.083dB higher than that of the pix2pix model. Meanwhile, the LSpix model has the highest score in SSIM, which is 1.618%, 15.649% and 3.848% higher than that of the pix2pix model, respectively. This proves that our model is closer to ground truth in structure, and the colors are more reductive.

Table 2. SSIM index of different models (%).

Network	MAX SSIM	MIN SSIM	AVE SSIM
pix2pix	92.888	52.474	82.163
pixG1	86.101	36.592	69.145
LSpix	**94.506**	**68.123**	**86.011**
pixG6	85.625	33.117	68.845
LSpixG6	91.312	56.897	78.387
LSpixG1	91.757	54.785	78.485

Note: Bold font is the best value for each column.

4.2. Inputing of different Gabor texture maps

In order to test the rendering effect when different Gabor texture feature maps are input, we use different feature images as input. Figure 7 shows how different Gabor texture images are rendered when Vanilla GAN loss is the target function of the pix2pix model. Figures 5(c),(d), that is, the direction is the scale is 7 and 45° or 90°, contain incomplete details of the original image, resulting in incomplete input texture features. Therefore, the generated image is blurred, as shown in Figures 7(a),(b). Although the 7th and 13th texture images were considered as training sets (pixG7+G13 model) with a total of 1231 × 2 images taken together, the rendering effect was not significantly improved, as shown in Figure 7(b). Evidently, by comparing the images in Figure 7, it can be found out that the visual effect of Figures 7(c)–(e) is good and not blurred. And Table 3 and 4 show the evaluation indexes after the input of different feature maps. The data show that incomplete input of texture feature map is not desirable.

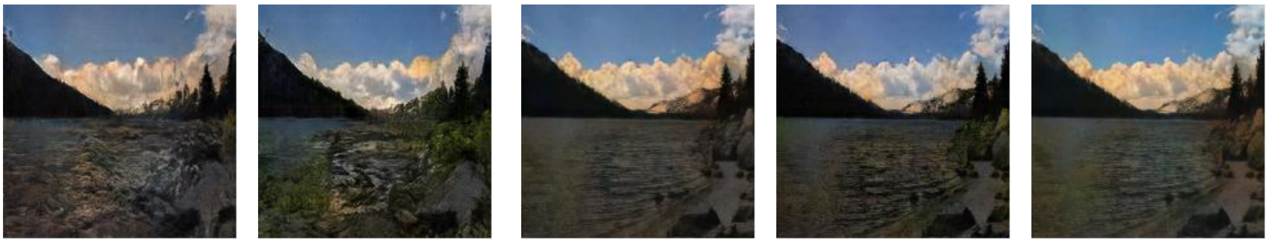

(a) pixG7 (b) pixG7+G13 (c) pixG1 (d) pixG1+G13 (e) pixG6

Figure 7. Effect of inputting different Gabor texture images.

And to compare the operation efficiency of input different texture maps, the training time is shown in Table 5, in hours. Regardless of whether the Gabor filter was used, which texture map was entered, the operation time was around 9 hours. However, if two texture maps are used for training, such as G1 and G13 are used in pixG1+G13 model, the training set doubles and the pre-training time doubles. Even though the results shown in Figure 7(d) are good, the method is not desirable. This is because when we use filtering, we need to extract multi-scale and multi-direction features and remove redundant information. Once the important information is removed, it will certainly have a certain impact on the results, resulting in blurred images.

Table 3. PSNR index of different models (dB).

Network	MAX PSNR	MIN PSNR	AVE PSNR
pixG1	28.225	**10.477**	19.981
pixG6	27.874	9.883	20.012
pixG7	27.565	9.232	17.600
pixG1+G13	**28.960**	9.947	**20.682**

Note: Bold font is the best value for each column.

Table 4. SSIM index of different models (%).

Network	MAX SSIM	MIN SSIM	AVE SSIM
pixG1	86.101	36.592	69.145
pixG6	85.625	33.117	68.845
pixG7	91.188	4.964	39.057
pixG1+G13	**87.615**	**42.562**	**71.630**

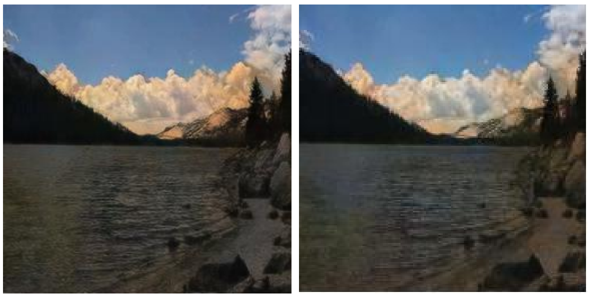
Note: Bold font is the best value for each column.

Table 5. Pre-training time of inputing different texture maps (h).

Model	pix2pix	pixG1	pixG6	pixG7	pixG7+G13	pixG1+G13
Time	8.72	9.00	8.76	8.43	15.27	16.52

4.3. Add the penalty item

Figure 8 shows the performance on whether or not to add a penalty item in the discriminator based on the pixG1 model. Figure 8(a) is the effect of not adding penalty items, and Figure 8(b) is the effect of adding penalty items. Obviously, Figure 8(b) has less error in detail and better visual effect. Penalty term, that is, gradient punishment is carried out by interpolation method to make the model satisfy Lipschitz constraint. The addition of punishment terms similar to WGAN_GP basically solves the problems of training instability and model collapse in the GAN model and ensures the diversity of generated samples.



(a) pixG1 (b) pixG1_GP

Figure 8. Adding the effect of the penalty item.

Tables 6 and 7 show the evaluation indexes whether or not to add a penalty item. With the addition of penalty term, the LSpix_GP model achieved the highest score in the minimum PSNR, which was 0.904dB higher than that of the original pix2pix model. Evidently, in the texture map extracted based on Gabor filter, the image with scale of 7 and direction of 0° has the best training effect. Furthermore, when the objective function is least squares loss, the average SSIM and performance are improved. When penalty term is added, the score of maximum and average SSIM is the highest, which is 1.753% and 1.083% higher than that of the pix2pix model. Therefore, the image rendered by the LSpixG1_GP model is better than that of the original model.

Table 6. PSNR index of different models (dB).

Network	MAX PSNR	MIN PSNR	AVE PSNR
pix2pix	29.204	11.126	23.024
pixG1	28.225	10.477	19.981
pixG6	27.874	9.883	20.012
LSpix	**32.795**	11.003	**24.107**
LSpixG1	32.524	11.238	21.354
LSpix_GP	31.859	**12.030**	24.019
LSpixG1_GP	32.342	11.514	21.290
LSpixG6	32.616	10.632	21.409
LSpixG1_GP	32.113	11.067	21.384

Note: Bold font is the best value for each column.

Table 7. SSIM index of different models (%).

Network	MAX SSIM	MIN SSIM	AVE SSIM
pix2pix	92.888	52.474	82.163
pixG1	86.101	36.592	69.145
pixG6	85.625	33.117	68.845
LSpix	94.506	**68.123**	**86.011**
LSpixG1	91.757	54.785	78.485
LSpix_GP	**94.641**	67.250	85.967
LSpixG1_GP	90.772	54.308	78.067
LSpixG6	91.312	56.897	78.387
LSpixG6_GP	90.941	52.740	78.236

Note: Bold font is the best value for each column.

To compare the operating efficiency of different objective functions given as input and increase the punishment items, the running time is listed in Table 8 in hours. For example, LSpixG6_GP represents using the least squares loss, adding the penalty item, the direction is 0° and the scale is 17. Regardless of whether Gabor filter was used, which texture map was input, whether Vanilla GAN loss or least square loss was the target function, the training time was approximately 9 h. Although the algorithm efficiency of adding the filter alone is basically the same, the time of using the filter after adding the penalty term will be increased by 2–3 h. Therefore, the algorithm in this study adopts LSpixG1_GP model, namely Gabor texture map with model input scale of 7 and direction of 0°, least squares loss and penalty term.

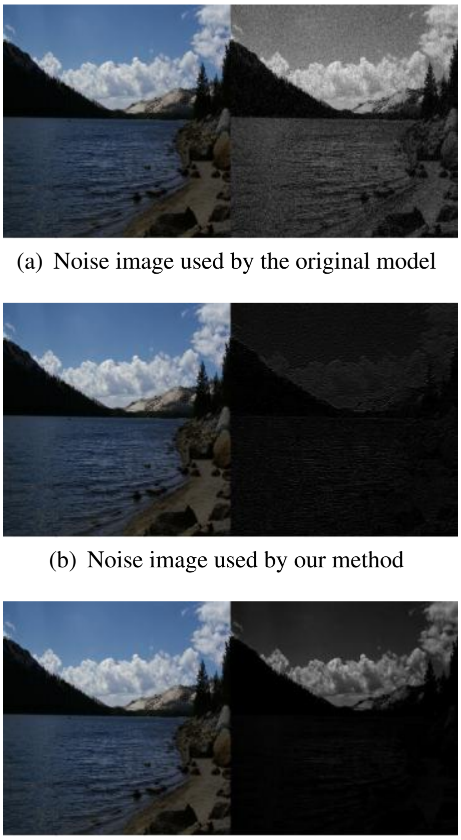
Table 8. Pre-training time of using different models (h).

Model	pix2pix	LSpix	LSpixG1	LSpix_GP	LSpixG1_GP	LSpixG6	LSpixG6_GP
Time	8.72	**8.66**	**8.66**	8.72	11.17	8.72	11.72

Note: Bold font is the best value for each row.

4.4. Rendering robust images

In order to evaluate the robustness of the model for rendering robust image, the rendering effect of low-quality images was tested by adding noise and dimming the image brightness, as shown in Figure 9. When testing the noise image, the Gaussian noise image with mean value of 0 and variance of 10 is added. When testing low-illumination images, power operation is performed on the pixels of the image, and the power is set to 2.5 to generate low-illumination images.



(a) Noise image used by the original model

(b) Noise image used by our method

(c) Low-illumination image

Figure 9. Test images.

We use PSNR evaluation metric to evaluate the rendering results of each model for low-quality images. As shown in Table 9, the image rendered by the LSpix model is of higher quality when rendering noisy images. As shown in Table 10, images rendered by Gabor filter models are generally of good quality for low-illumination images. After the Gabor filter, the objective function is least square loss and the penalty term is added, the image quality of the LSpixG1_GP model is higher than that of the original model. This is because the method in this paper uses Gabor filter to avoid the interference of noise to the image to a certain extent. And when extracting features, the depth information of the image can be extracted to avoid the influence of light on the image. Clearly, the proposed method in this paper is robust to color rendering of low-quality images.

Table 9. PSNR index of noise image (dB).

Network	MAX PSNR	MIN PSNR	AVE PSNR
pix2pix	29.489	10.770	22.528
pixG1	25.657	10.562	18.665
LSpix	**29.655**	11.805	**22.528**
LSpixG1	27.650	**12.409**	19.942
LSpix_GP	29.516	11.950	22.504
LSpixG1_GP	27.306	11.548	19.966

Note: Bold font is the best value for each column.

Table 10. PSNR index of low-illumination image (dB).

Network	MAX PSNR	MIN PSNR	AVE PSNR
pix2pix	21.977	8.723	12.535
pixG1	**26.158**	7.864	**14.441**
LSpix	21.457	9.119	12.579
LSpixG1	24.565	7.946	14.171
LSpix_GP	21.948	**9.334**	12.563
LSpixG1_GP	24.337	7.886	14.127

Note: Bold font is the best value for each column.

5. Conclusions

We proposed a novel image color rendering method based on using Gabor filter based improved pix2pix for robust image and demonstrate its feasibility and superiority for a variety of tasks. It enables automatically render robust images and has good robustness with low-quality image rendering. The experimental results on summer dataset demonstrate that the proposed method can achieve high-quality performance with image color rendering. At present, the image resolution of image processing based on deep learning is limited, which leads to the limitation in the practical application of rendering method. In the future, we will focus on improving the resolution of network model input images.

Acknowledgments

This work were partially supported by the National Natural Science Foundation of China (No. 62002285 and No. 61902311).

Conflict of Interest

The authors declare there is no conflict of interest.

References

1. M. Wang, G. W. Yang, S. M. Hu, S. T. Yau, A. Shamir, Write-a-video: Computational video montage from themed text, *ACM Trans. Graphics*, **38** (2019), 1–13. doi: 10.1145/3355089.3356520.
2. R. Yi, Y. J. Liu, Y. K. Lai, P. L. Rosin, Apdrawinggan: Generating artistic portrait drawings from face photos with hierarchical gans, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 10743–10752.
3. T. Yuan, Y. Wang, K. Xu, R. R. Martin, S. M. Hu, Two-layer qr codes, *IEEE Trans. Image Process.*, **28** (2019), 4413–4428. doi: 10.1109/TIP.2019.2908490.
4. H. Li, Q. Zheng, J. Zhang, Z. Du, Z. Li, B. Kang, Pix2pix-based grayscale image coloring method, *J. Comput. Aided Comput. Graphics*, **33** (2021), 929–938.
5. H. Li, M. Zhang, K. Yu, X. Qi, J. Tong, A displacement estimated method for real time tissue ultrasound elastography, *Mobile Netw. Appl.*, **26** (2021), 1–10. doi: 10.1007/s11036-021-01735-3.
6. T. Welsh , M. Ashikhmin, K. Mueller, Transferring color to greyscale images, *ACM Trans. Graph.*, **21** (2002), 277–280. doi: 10.1145/566570.566576.
7. Y. Jing, Z. J. Chen, Analysis and research of globally matching color transfer algorithms in different color spaces, *Comput. Eng. Appl.*, (2007), 45–54.
8. S. F. Yin, C. L. Cao, H. Yang, Q. Tan, Q. He, Y. Ling, et al., Color contrast enhancent method to imprrove target detectability in night vision fusion, *J. Infrared Milli. Waves*, **28** (2009), 281–284.
9. M. W. Xu, Y. F. Li, N. Chen, S. Zhang, P. Xiong, Z. Tang, et al., Coloration of the low light level and infrared image using multi-scale fusion and nonlinear color transfer technique, *Infrared Techn.*, **34** (2012), 722–728.
10. Z. P, M. G. Xue, C. C. Liu, Night vision image color fusion method using color transfer and contrast enhancement, *J. Graphics*, **35** (2014), 864–868.
11. R. Zhang, J. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, et al., Real-time user-guided image colorization with learned deep priors, preprint, arXiv:1705.02999.
12. Z. Cheng, Q. Yang, B. Sheng, Deep colorization, preprint, arXiv:1605.00075.
13. K. Nazeri, E. Ng, M. Ebrahimi, Image colorization using generative adversarial networks, in *International Conference on Articulated Motion and Deformable Objects*, (2018), 85–94. doi: 10.1007/978-3-319-94544-69.
14. H. Li, Q. Zheng, W. Yan, R. Tao, X. Qi, Z. Wen, Image super-resolution reconstruction for secure data transmission in internet of things environment, *Math. Biosci. Eng.*, **18** (2021), 6652–6671. doi: 10.3934/mbe.2021330 .
15. H. A. Li, Q. Zheng, X. Qi, W. Yan, Z. Wen, N. Li, et al., Neural network-based mapping mining of image style transfer in big data systems, *Comput. Intell. Neurosci.*, **21** (2021), 1–11. doi: 10.1155/2021/8387382.
16. C. Xiao, C. Han, Z. Zhang, J. Qin, T. Wong, G. Han, et al., Example-based colourization via dense encoding pyramids, *Comput. Graph. Forum*, **12** (2019), 20–33. doi: 10.1111/cgf.13659 .

17. S. S. Huang, H. Fu, S. M. Hu, Structure guided interior scene synthesis via graph matching, *Graph. Models*, **85** (2016), 46–55. doi: 10.1016/j.gmod.2016.03.004.

18. Y. Liu, K. Xu, L. Yan, Adaptive brdf mriented multiple importance sampling of many lights, *Comput. Graph. Forum*, **38** (2019), 123–133. doi: 10.1111/cgf.13776.

19. S. S. Huang, H. Fu, L. Wei, S. M. Hu, Support substructures: Support-induced part-level structural representation, **22** (2015), 2024–36. doi: 10.1109/TVCG.2015.2473845.

20. G. Larsson, M. Maire, G. Shakhnarovich, Learning representations for automatic colorization, in *European Conference on Computer Vision, Springer International Publishing*, (2016), 577–593. doi: 10.1007/978-3-319-46493-035.

21. I. H. Iizuka S, Simo-Serra E, Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification, *ACM Trans. Graph.*, **35** (2016), 577–593. doi: 10.1145/2897824.2925974.

22. R. Zhang, P. Isola, A. A. Efros, Colorful image colorization, *Comput. Vision Pattern Recogn.*, **9907** (2016), 649–666. doi: 10.1007/978-3-319-46487-940.

23. C. Li, J. Guo, C. Guo, Emerging from water: Underwater image color correction based on weakly supervised color transfer, *IEEE Signal Proc. Lett.*, **25** (2018), 323–327. doi: 10.1109/LSP.2018.2792050.

24. R. Zhou, C. Tan, P. Fan, Quantum multidimensional color image scaling using nearest-neighbor interpolation based on the extension of frqi, *Mod. Phys. Lett. B*, **31** (2017), 175–184. doi: 10.1142/s0217984917501846.

25. E. Reinhard, M. Adhikhmin, B. Gooch, P. Shirley, Color transfer between images, *IEEE Comput. Graph. Appl.*, **21** (2001), 34–41. doi: 10.1109/38.946629.

26. P. Isola, J. Zhu, T. Zhou, A. A. Efro, Image-to-image translation with conditional adversarial networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 1125–1134. doi: arxiv-1611.07004.

27. L. Tao, Review on gabor expansion and transform, *J. Anhui Univ.*, **41** (2017), 2–13.

28. R. Yi, Y. J. Liu, Y. K. Lai, P. L. Rosin, Unpaired portrait drawing generation via asymmetric cycle mapping, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 8214–8222. doi: 10.1109/CVPR42600.2020.00824.

29. Z. H. Wang, Z. Z. Wang, Robust cell segmentation based on gradient detection, Gabor filtering and morphological erosion, *Biomed. Signal Proces. Control*, **65** (2021), 1–13. doi: 10.1016/j.bspc.2020.102390.

30. V. Kouni, H. Rauhut, Star DGT: a robust gabor transform for speech denoising, preprint, arXiv: 2104.14468.

31. Y. Chen, L. Zhu, P. Ghamisi, X. Jia, G. Li, L. Tang, Hyperspectral Images Classification With Gabor Filtering and Convolutional Neural Network, *IEEE Geosci. Remote Sens. Lett.*, **14** (2020), 2355–2359. doi: 10.1109/LGRS.2017.2764915.

32. H. W. Sino, Indrabayu, I. S. Areni, Face recognition of low-resolution video using gabor filter and adaptive histogram equalization, in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, (2019), 417–421. doi: 10.1109/ICAIIT.2019.8834558.

33. X. Lin, X. Lin, X. Dai, Design of two-dimensional gabor filters and implementation of iris recognition system, *Telev. technol.*, **35** (2011), 109–112.

34. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial networks, *Adv. Neural Inform. Proc. Sys.*, **3** (2014), 2672–2680, .

35. X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, S. P. Smolley, Least squares generative adversarial networks, in *Proceedings of the IEEE International Conference on Computer Vision*, (2016), 2813–2821.

36. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein gans, preprint, arXiv: 1704.00028.

37. Z. Zhang, M. R. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, in *32nd Conference on Neural Information Processing Systems (NeurIPS)*, (2018), 1–14.

38. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, preprint, arXiv:1701.07875.

39. F. Duan, S. Yin, P. Song, W. Zhang, H. Yokoi, Automatic welding defect detection of x-ray images by using cascade adaboost with penalty term, *IEEE Access*, **7** (2019), 125929–125938. doi: 10.1109/ACCESS.2019.2927258.

40. CycleGAN/datasets, *Summer2winter*, 2000. Available from: https://people.eecs.berkeley.edu/ taesung park/CycleGAN/datasets.

41. Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment : From error visibility to structural similarity, *IEEE Trans. Image Process.*, (2004), 600–612. doi: 10.1109/TIP.2003.819861.

42. A. Horé, D. Ziou, Image quality metrics: Psnr vs. ssim, in *International Conference on Pattern Recognition*, (2010), 2366–2369. doi: 10.1109/ICPR.2010.579.