Postprint of a Conditional Generative Adversarial Network-Based Method for Coloring Manga Hand-drawn Sketches

Authors: Liang Peijun, Liu Yijun

Date: 2018-05-20T00:00:00+00:00

Abstract

Coloring hand-drawn comic sketches is a time-consuming and labor-intensive yet essential task in the comic and game development industries. Therefore, this paper proposes an automatic coloring method for hand-drawn comic sketches based on Conditional Generative Adversarial Networks (CGAN). In the experiments, a generator with a U-Net architecture is adopted, L1 loss is utilized to constrain the network model, and during the adversarial training between the generator and discriminator, the model continuously learns and optimizes the mapping relationship from hand-drawn sketches to corresponding color images. Finally, the trained conditional GAN model is employed to color the hand-drawn sketches. Experimental results demonstrate that this approach can effectively and efficiently color hand-drawn comic sketches while maintaining satisfactory visual quality.

Full Text

Colorization of Manga Sketch Based on Conditional Generative Adversarial Networks

Liang Peijun^{a}, **Liu Yijun**^{a,b}

^{a}College of Computer Science, ^{b}College of Information Engineering
Guangdong University of Technology, Guangzhou 510006, China

Abstract: Colorization of manga sketches is a time-consuming and labor-intensive yet crucial task in manga and game development. To address this challenge, this paper proposes an automatic colorization method for manga sketches based on Conditional Generative Adversarial Networks (CGAN). In our experiments, we employ a U-Net structured generator and constrain the

network model using L1 loss. Through adversarial training between the generator and discriminator, the model continuously learns and optimizes the mapping relationship from sketches to corresponding color images. Finally, the trained conditional GAN model is used to colorize the sketches. Experimental results demonstrate that this approach can effectively and rapidly colorize manga sketches while maintaining satisfactory visual quality.

Keywords: manga; sketch; colorization; deep learning; conditional generative adversarial network (CGAN)

0 Introduction

With the flourishing development of the manga and digital media industries, the production speed of various manga products has gradually accelerated, and consumer demand has evolved beyond mere black-and-white line art. However, manga creation begins with artists drawing black-and-white sketches, followed by a series of complex processing steps before final colorization. Both the sketching and coloring stages are extremely time-consuming and labor-intensive. Due to this time- and labor-intensive nature, many manga products on the market today feature color only on their covers or first few pages to attract buyers' attention. Therefore, leveraging deep learning technology to automate the colorization process in an unsupervised manner would bring qualitative improvements in time and efficiency to manga production.

Previous image colorization methods [1–3] primarily rely on the continuity of grayscale regions to identify areas with identical grayscale values, thereby assisting image segmentation for colorization. However, black-and-white sketches contain only contour lines, unlike grayscale images that have continuous and hierarchically differentiated grayscale regions suitable for image segmentation. Consequently, grayscale colorization methods cannot effectively handle sketch colorization, as demonstrated by the grayscale colorization effects in Figure 1 [Figure 1: see original paper].

Gatys et al. [4–6] proposed an artistic style transfer method that leverages the separability of content and style features in convolutional neural networks. This approach extracts one feature from each of two different images and combines them to generate a novel image whose content and style originate from the two source images. However, applying this model to manga sketch colorization fails to achieve proper colorization, as style does not equate to the color attribute, nor does content correspond to the contour attribute, as shown by the style transfer effects in Figure 1.

In 2014, Goodfellow et al. [7] introduced the pioneering Generative Adversarial Network (GAN), a model comprising a generator and a discriminator. The generator captures the data distribution of samples, while the discriminator acts as a binary classifier to determine whether input data originates from the generator

or represents real data. The optimization process can be summarized as: the generator produces images to deceive the discriminator, which then judges their authenticity, and the entire network iteratively trains these two models adversarially. As training progresses, both models become increasingly capable, eventually reaching equilibrium. Through this game-theoretic optimization, GANs can generate high-quality images. However, this training approach offers excessive freedom, and applying it directly to sketch colorization produces malformed results, such as identical colors being filled in unrelated regions.

1 Methodology

This paper proposes a conditional GAN-based method for automatic manga sketch colorization. Our approach employs a U-Net [9] encoder-decoder [10] network structure with skip connections for the generator. The U-Net architecture, illustrated in Figure 2 [Figure 2: see original paper], allows low-level information to directly reach the decoder, thereby preserving the underlying information. The discriminator model evaluates image authenticity at the patch level (PatchGAN) [11], examining each generated color sketch in $N \times N$ patches and averaging the responses as the final result. Additionally, an L1 constraint term is incorporated into the network' s objective function to enhance image clarity. The network takes black-and-white manga sketches as generator input, adds noise via Dropout in the decoder, outputs colored manga images, and trains adversarially with the discriminator to optimize the mapping between color and sketch images, ultimately converging to a stable network model. The training and testing dataset was crawled and processed from Safebooru, with network parameters trained on GPU using PyTorch. The effectiveness of our method is demonstrated in Figure 1 [Figure 1: see original paper].

The CGAN training process is illustrated in Figure 3 [Figure 3: see original paper]. The generator G receives conditional information (black-and-white sketches) and random noise vector z to generate color images. The discriminator D trains on both generated and original color images, aiming to distinguish real images from generated ones. The two networks optimize through adversarial training, constructing a mapping from sketches to color images, ultimately enabling unsupervised colorization functionality.

1.1 CGAN Principles

A GAN is a generative model comprising a generator G and a discriminator D. The original GAN objective function can be expressed as:

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y\sim p_{data}(x,y)}[\log D(x,y)] + \mathbb{E}_{x\sim p_{data}(x),z\sim p_z(z)}[\log(1-D(x,G(x,z)))]$$

where G strives to minimize the objective function while D attempts to maximize it. Previous conditional GANs have found it beneficial to combine the GAN

objective with traditional loss functions such as L2 distance [12]. While the discriminator's task remains unchanged, the generator must not only deceive the discriminator but also approximate the real output based on L2 distance. Using L1 distance instead of L2 produces sharper results. The complete objective function becomes:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y\sim p_{data}(x,y),z\sim p_z(z)}[||y - G(x,z)||_1]$$

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G)$$

Even without random noise z, the network can still learn the mapping from x to y, but it would produce deterministic outputs and thus fail to match distributions beyond delta functions. Previous conditional GANs have addressed this by providing Gaussian noise z and input image x to the generator. In our implementation, noise is supplied via dropout applied to several generator layers during both training and testing. This enables the model to produce stochastic outputs and capture the full entropy of the distribution. Besides dropout noise, minimal randomness can also be observed in the network' s final output.

1.2 Network Architecture and Implementation Details

Our model comprises a generator and a discriminator. We denote Ck as a layer with k filters, processing input through convolution, batch normalization, and ReLU activation [13] in sequence. CDk denotes a layer with a dropout rate of 0.5, processing input through convolution, batch normalization, dropout, and ReLU activation. All convolutions use 4×4 kernels with a stride of 2. The downsampling factor is 2 for convolutions in the encoder and discriminator, while the upsampling factor is 2 for convolutions in the decoder.

1.2.1 Generator Architecture A key characteristic of manga sketch colorization is the mapping from high-resolution input grids to high-resolution output grids. Although inputs and outputs differ in color representation, they share the same underlying structure—the same contours. Therefore, the generator architecture is designed around these properties.

Many image colorization solutions employ an encoder-decoder network, as shown in Figure 4 [Figure 4: see original paper]. In such networks, the encoder progressively downsamples the input through each layer, requiring all information to flow through the bottleneck. For colorization tasks, substantial low-level information is shared between input and output, such as the positional information of prominent edges in sketch colorization. Therefore, it is desirable to allow this information to bypass the bottleneck.

To enable shared information to bypass the bottleneck in the generator, skip connections are incorporated, creating a U-Net structure. Figure 5 [Figure

5: see original paper] illustrates the generator architecture. The encoder consists of C64-C128-C256-C512-C512-C512-C512-C512, while the decoder comprises CD512-CD1024-CD1024-C1024-C1024-C512-C256-C128. Following the final decoder layer is a convolution to match output channels, followed by a tanh function. A special handling is required: batch normalization is not applied to the C64 layer in the encoder. All ReLU activations in the encoder use LeakyReLU with a slope of 0.2, while the decoder uses standard ReLU.

Let n be the total number of layers. The U-Net architecture is identical except for skip connections between each layer i in the encoder and layer $n-i$ in the decoder. These connections concatenate the activations from layer i to layer $n-i$, simultaneously adjusting the channel dimensions in the decoder.

1.2.2 Discriminator Architecture As shown in Equation (5), relying on the L1 term enhances low-frequency correctness, constraining the discriminator to model only high-frequency structures and increasing the clarity of generated color images. To model high frequencies, the discriminator adopts PatchGAN, which penalizes structure only at the patch scale. The discriminator distinguishes whether each $N \times N$ patch in an image is real, then averages all responses as the final result. Figure 6 [Figure 6: see original paper] illustrates the discriminator architecture. Following the final layer is a convolution to match one-dimensional output, succeeded by a Sigmoid function. Notably, the C64 layer does not use batch normalization. All ReLU activations in the discriminator use LeakyReLU with a slope of 0.2.

2 Experiments

2.1 Experimental Platform

Convolution operations require extensive matrix computations during training. Graphics Processing Units (GPUs) significantly accelerate training speed compared to Central Processing Units (CPUs), making them more suitable for this task. Our experimental platform consists of Ubuntu 16.04 64-bit OS, Intel i5 CPU, 8 GB RAM, and an NVIDIA GTX 1050Ti GPU. We use the PyTorch deep learning framework for implementation.

2.2 Experimental Data and Processing

The manga image dataset used for training was randomly crawled from the popular manga material website Safebooru using a web scraper. After filtering out unsuitable images, approximately 27,000 manga images remained. These images were batch-resized to 256×256 dimensions using ImageMagick. Figure 7 [Figure 7: see original paper] shows some original images.

All manga images were then batch-converted to sketch renderings through contour extraction using OpenCV. The sketch images are shown in Figure 8 [Figure 8: see original paper]. The first 3,000 images were selected as test data, with the

remainder used for training. To facilitate data reading and comparison, original images and their processed counterparts were concatenated pairwise for training and testing, as shown in Figure 9 [Figure 9: see original paper].

2.3 Model Training

During training, gradient descent was alternately executed between the discriminator and generator using minibatch stochastic gradient descent with the Adam solver [14]. The batch size was set to 1, a configuration proven effective for image generation tasks in prior work [15]. The learning rate was set to 0.0002 with 200 training iterations, requiring 3 days of training time. A Visdom server was employed to visualize training progress.

2.4 Experimental Results

To evaluate whether the trained model can colorize black-and-white manga sketches, we tested it on 3,000 sketch images from the test set, achieving an average colorization time of approximately 0.8 seconds per sketch. The unsupervised colorization results are shown in Figure 10 [Figure 10: see original paper]. The results demonstrate that our proposed network model satisfies the two key requirements: preservation of underlying contour information and colorization rationality, while maintaining relatively clear and reasonable overall effects. However, some colorized results exhibit blurriness and relatively monochromatic color schemes.

Beyond evaluating manga sketch colorization, we also tested the model's capability on natural grayscale images. The training and test data for this experiment comprised 10,000 and 1,000 dog images from ImageNet, respectively. Preprocessing and training procedures were similar to the manga sketch experiment. The colorization effects are shown in Figure 11 [Figure 11: see original paper], revealing that while the model demonstrates feasibility for natural grayscale images, some details exhibit incorrect color filling.

3 Conclusion

This paper proposes an unsupervised colorization method for manga sketches based on conditional GANs. We construct a conditional adversarial generative network model for anime sketch colorization, adding an L1 constraint to the objective function to enhance result clarity. The U-Net structured generator preserves low-level contour information between sketches and color images, while adversarial training between generator and discriminator constructs the mapping relationship between sketches and color images. The final model can colorize general black-and-white sketches, achieving unsupervised manga sketch colorization. Experiments demonstrate that this method can reasonably colorize sketches while maintaining satisfactory clarity. Future work will further investigate improvements in colorization clarity and color palette diversity.

References

[1] Zhang R, Isola P, Efros A A. Colorful image colorization [C]// Proc of European Conference on Computer Vision. [S. l.]: Springer International Publishing, 2016: 649-666.

[2] Levin A, Lischinski D, Weiss Y. Colorization using optimization [J]. ACM Trans on Graphics, 2004, 23(3): 689-694.

[3] Cao Yun, Zhou Zhiming, Zhang Weinan, et al. Unsupervised diverse colorization via generative adversarial networks [J]. arXiv preprint arXiv: 1702.06674, 2017.

[4] Gatys L A, Ecker A S, Bethge M. A neural algorithm of artistic style [J]. arXiv preprint arXiv: 1508.06576, 2015.

[5] Gatys L A, Ecker A S, Bethge M. Image style transfer using convolutional neural networks [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. [S. l.]: IEEE Computer Society, 2016: 2414-2423.

[6] Johnson J, Alahi A, Li F F. Perceptual losses for real-time style transfer and super-resolution [C]// Proc of European Conference on Computer Vision. 2016: 694-711.

[7] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets [C]// Advances in Neural Information Processing Systems. 2014: 2672-2680.

[8] Mirza M, Osindero S. Conditional generative adversarial nets [J]. arXiv preprint arXiv: 1411.1784, 2014.

[9] Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation [C]// Proc of International Conference on Medical Image Computing and Computer-Assisted Intervention. 2015: 234-241.

[10] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks [J]. Science, 2006, 313(5786): 504-507.

[11] Li C, Wand M. Precomputed real-time texture synthesis with Markovian generative adversarial networks [C]// Proc of European Conference on Computer Vision. [S. l.]: Springer International Publishing, 2016: 702-716.

[12] Pathak D, Krahenbuhl P, Donahue J, et al. Context encoders: feature learning by inpainting [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2536-2544.

[13] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C]// Proc of International Conference on Machine Learning. 2015: 448-456.

[14] Kingma D, Ba J. Adam: a method for stochastic optimization [J]. arXiv preprint arXiv: 1412.6980, 2014.

[15] Ulyanov D, Vedaldi A, Lempitsky V. Instance normalization: the missing ingredient for fast stylization [J]. arXiv preprint arXiv: 1607.08022, 2016.

Note: Figure translations are in progress. See original paper for figures.