Unsupervised Diverse Colorization via Generative Adversarial Networks

Yun Cao, Zhiming Zhou, Weinan Zhang, Yong Yu
Shanghai Jiao Tong University, Shanghai, China
{yuncao, heyohai, wnzhang, yyu}@apex.sjtu.edu.cn

Abstract
Colorization of grayscale images has been a hot topic in computer vision. Previous research mainly focuses on producing a colored image to match the original one. However, since many colors share the same gray value, an input grayscale image could be diversely colored while maintaining its reality. In this paper, we design a novel solution for unsupervised diverse colorization. Specifically, we leverage conditional generative adversarial networks to model the distribution of real-world item colors, in which we develop a fully convolutional generator with multi-layer noise to enhance diversity, with multi-layer condition concatenation to maintain reality, and with stride 1 to keep spatial information. With such a novel network architecture, the model yields highly competitive performance on the open LSUN bedroom dataset. The Turing test of 80 humans further indicates our generated color schemes are highly convincible.

1 Introduction
Image colorization assigns a color to each pixel of a target grayscale image. Early colorization methods [Levin et al., 2004; Qu et al., 2006] require users to provide considerable scribbles on the grayscale image, which is apparently time-consuming and requires specialties. Later research provides more automatic colorization methods. Those colorization algorithms differ in the ways of how they model the correspondence between grayscale and color.

Given an input grayscale image, non-parametric methods first define one or more color reference images (provided by human or retrieved automatically) to be used as source data. Then, following the Image Analogies framework [Hertzmann et al., 2001], the color is transferred onto the input image from analogous regions of the reference image(s) [Welsh et al., 2002; Liu et al., 2008; Gupta et al., 2012; Chia et al., 2011]. Parametric methods, on the other hand, learn prediction functions from large datasets of color images in the training stage, posing the colorization problem as either regression in the continuous color space [Cheng et al., 2015; Deshpande et al., 2015; Zhang et al., 2016] or classification of quantized color values [Charpiat et al., 2008].

Whichever seeking the reference images or learning a color prediction model, all above methods share a common goal, i.e. to provide a colored image closer to the original one. But as we know, many colors share the same gray value. Purely from a grayscale image, one cannot tell what color of clothes the girl is wearing or what color the bedroom wall is. Those methods all produce a deterministic mapping function, thus when an item could have diverse colors, their models tend to provide a weighted average brownish color as pointed out in [Koo, 2016].

In this paper, to avoid this sepia-toned colorization, we use conditional generative adversarial networks (GANs) to generate diverse colorizations for a single grayscale image while maintaining their reality. GAN is originally proposed by [Goodfellow et al., 2014] and is composed of two adversarial parts: a generative model $G$ that captures the data distribution, and a discriminative model $D$ that estimates reality probability of a real or generated sample. Unlike many other conditional GANs like [Isola et al., 2016] using convolution layers as encoder and deconvolution layers as decoder, we build a fully convolutional generator and each convolutional layer is splinted by a concatenate layer to continuously render the conditional grayscale information and a batch normalization layer to provide internal covariate shift. Additionally, to maintain the spatial information, we set all convolution stride to 1 to avoid downsizing data. We also concatenate noise channels to first half convolutional layers of the generator to attain more diversity in the colored image generation process. As the generator $G$ would capture the color distribution, we can alter the colorization result by changing the input noise. Thus we no longer need to train an additional independent model for each color scheme like [Cheng et al., 2015].

As our goal alters from producing the original colors to producing realistic diverse colors, we conduct questionnaire surveys as a Turing test instead of calculating the root mean squared error (RMSE) comparing the original image to measure our colorization result. The feedback from 80 subjects indicates our model successfully produce high-reality colored images, yielding more than 62.6% positive feedback while the rate of ground truth images is 70.0%. Furthermore, a significance t-test between the percentages of humans rating as real image for each test case shows the $p$-value is $0.1359 > 0.05$, which indicates our generated results have no significant difference with the ground truth images.
2 Related Work

2.1 Diverse Colorization

The problem of colorization was proposed from last century, but the research of diverse colorization was not paid much attention until this decade. [Cheng et al., 2015] used additionally trained model to handle diverse colorization of a scene image particularly in day and dawn. [Zhang et al., 2016] posed the diverse colorization problem as a classification task and use class-rebalancing at training time to increase the diversity of colors in the result. [Deshpande et al., 2016] learned a low dimensional embedding of color fields using a variational auto-encoder (VAE). They constructed loss terms for the VAE decoder that avoid blurry outputs and take into account the uneven distribution of pixel colors and finally developed a conditional model for the multi-modal distribution between gray-level image and the color field embeddings.

Compared with above work, our solution uses conditional generative adversarial networks to achieve unsupervised diverse colorization in a generic way with little domain knowledge of the images.

2.2 Conditional GAN

Generative adversarial networks (GANs) [Goodfellow et al., 2014] have attained much attention in unsupervised learning research during the recent 3 years. Conditional GANs have been widely used in various computer vision scenarios. [Reed et al., 2016] used text to generate image by applying adversarial networks. [Isola et al., 2016] provided a general-purpose image-to-image translation model that handles tasks like label to scene, aerial to map, day to night, edges to photo and also grayscale to color.

Some of above work may share a similar goal with us, but our conditional GAN structure differs a lot from previous work in several architectural choices mainly for the generator. Unlike other generators which employ an encoder-like front part consisting of multiple convolution layers and a decoder-like end part consisting of multiple deconvolution layers, our generator uses only convolution layers all over the architecture, and does not downsize data shape by applying convolution stride no more than 1 and no pooling operation. Additionally, we add multi-layer noise to generate more diverse colorization, while using multi-layer conditional information to keep the generated image highly realistic.

3 Methods

3.1 Problem formulation

GANs are generative models that learn a mapping from random noise vector $z$ to output color image $x$: $G : z \rightarrow x$. Compared with GANs, conditional GANs learn a mapping from observed grayscale image $y$ and random noise vector $z$, to $x$: $G : \{y, z\} \rightarrow x$. The generator $G$ is trained to produce outputs that cannot be distinguished from “real” images by an adversarially trained discriminator $D$, which is trained with the aim at detecting the “fake” images produced by the generator. This training procedure is shown in Figure 1.

![Figure 1: The illustration of conditional GAN. Generator $G$ is given conditional information (Grayscale image) together with noise $z$, and produce generated color channels. Discriminator $D$ is trained over the real color image and the generated color image by $G$. The goal of $D$ is to distinguish real image from the fake one. Both nets are trained adversarially.](image)

The objective of a GAN can be expressed as

$$L_{GAN}(G, D) = \mathbb{E}_{x \sim P_{data}(x)} \left[ \log D(x) \right] + \mathbb{E}_{z \sim P_z(z)} \left[ \log (1 - D(G(z))) \right],$$

while the objective of a conditional GAN is

$$L_{cGAN}(G, D) = \mathbb{E}_{x \sim P_{data}(x)} \left[ \log D(x) \right] + \mathbb{E}_{y \sim P_y(y), z \sim P_z(z)} \left[ \log (1 - D(G(y, z))) \right],$$

where $G$ tries to minimize this objective against an adversarial $D$ that tries to maximize it. i.e.

$$G^* = \arg \min_G \max_D L_{cGAN}(G, D)$$

Without $z$, the generator could still learn a mapping from $y$ to $x$, but would produce deterministic outputs. That is why GAN is more suitable for diverse colorization tasks than other deterministic neural networks.

3.2 Architecture and implementation details

The high-level structure of our conditional GAN is consistent with traditional ones [Isola et al., 2016; Deshpande et al., 2016], while the detailed architecture of our generator $G$ differs a lot.

Convolution or deconvolution

Convolution and deconvolution layers are two basic components of image generators. Convolution layers are used mainly to extract exact conditional features. And additionally, many researches [Zhang et al., 2016; Isola et al., 2016; Deshpande et al., 2016] use superposition of multiple convolution layers with stride more than 1 to downsize the data shape, which works as a data encoder. Deconvolution layers are then used to upsize the data shape as a decoder of the data representation [Nguyen et al., 2016a; Isola et al., 2016; Deshpande et al., 2016]. While many other researches share this encoder-decoder structure, we choose to use only convolution layers in our generator $G$. Firstly, convolution layers are well capable of feature extraction and transmission.
Meanwhile, all the convolution stride is set to 1 to prevent data shape from downsizing, thus the important spatial information can be kept along the data flow till the final generation layer. Other work [Ronneberger et al., 2015; Isola et al., 2016] also takes this spatial information into consideration. They added skip connections between each layer and layer $n - i$ to form a “U-Net” structure, where $n$ is the total number of layers. Each skip connection simply concatenates all channels at layer $i$ with those at layer $n - i$. But our modification is more straightforward and easy to implement.

**YUV or RGB**

A color image can be represented in different forms. The most common representation is RGB form which splits a color pixel into red, green, blue three channels. Most computer vision tasks use RGB representation like [Deshpande et al., 2015; Isola et al., 2016; Nguyen et al., 2016b] due to its generality. Other kinds of representations are also included like Lab [Charpiat et al., 2008; Zhang et al., 2016; Limmer and Lensch, 2016], YUV (or Y’CrCb) [Cheng et al., 2015]. In colorization tasks, we have grayscale image as conditional information, so there is convenience using YUV representation, because the Y channel or so called luminance channel represents exactly the grayscale information. So while using YUV representation, we can just predict 2 channels and then concatenate with the grayscale channel to give a full color image. Additionally, if you use RGB as image representation, all result channels are predicted, thus to keep the grayscale of generated color image consistent with the original grayscale image, we need to add an additional loss Eq. (4) as a controller to make sure Gray($x$) ≈ $y$:

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

where function $Y = \text{Gray}(RGB)$ is the well-known psychological formulation:

$$
Y = 0.299 R + 0.587 G + 0.114 B.
$$

Note that Eq. (4) can still remain colorization diversity. Then the objective function will be modified to:

$$
G^* = \arg \min_{G} \max_{D} \mathcal{L}_{\text{GAN}}(G, D) + \lambda \mathcal{L}_{L1}(G)
$$

But no matter how we change the parameter to balance two loss parts, the L1 part will not reach zero, so there will always be blurry unless the model is overfitting. We do comparison experiments on both RGB and YUV representations, results are shown in Section 4.2.

**Multi-layer noise**

In the work of [Isola et al., 2016], they mentioned noise ignorance while training the generator. To handle this problem, they provide noise only in the form of dropout, applied on several layers of the generator at both training and testing time. We also noticed this problem. Traditional GANs and condition GANs receive noise information at the very start layer, during the continuous data transformation through the network, the noise information attenuated a lot. To overcome this problem and make the colorization results more diversified, we concatenate the noise channel onto the first half of the generator layers (the first three layers in our case). We do comparison experiments on both one-layer noise and multi-layer noise representations, with results shown in Section 4.2.

**Algorithm 1** Training phase of our conditional GANs, with the default parameters $k_D = 1, k_G = 1, m = 64, s_z = 100, s = 64$.

1: for the number of training iterations do
2: for $k_D$ steps do
3:  Generate minibatch of $m$ randomly sampled noise \{$z^{(1)}, \ldots, z^{(m)}$\} each of size $[s_z]$.
4:  Sample minibatch of $m$ grayscale images \{$y^{(1)}, \ldots, y^{(m)}$\} each of shape $[s, s, 1]$.
5:  Get the corresponding minibatch of $m$ color images \{$x^{(1)}, \ldots, x^{(m)}$\} from data distribution $p_{\text{data}}(\mathbf{x})$.
6:  Update the discriminator $D$ by:
7:  \[\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(\mathbf{x}^{(i)}) + \log (1 - D(G(y^{(i)}, z^{(i)})))]
\]
8: end for
9: for $k_G$ steps do
10:  Generate minibatch of $m$ randomly sampled noise \{$z^{(1)}, \ldots, z^{(m)}$\}.
11:  Sample minibatch of $m$ grayscale images \{$y^{(1)}, \ldots, y^{(m)}$\}.
12:  Update the generator by:
13: \[\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log (1 - D(G(y^{(i)}, z^{(i)})))
\]
14: end for

**Multi-layer conditional information**

Other conditional GANs usually add conditional information only in the first layer, this is because the layer shape of previous generators changes along their convolution and deconvolution layers. But due to the consistent layer shape of our generator, we can apply concatenation of conditional grayscale information throughout the whole generator layers which can provide sustained conditional supervision. Though the “U-Net” skip structure of [Isola et al., 2016] can also help posterior layers receive conditional information, our model modification is still more straightforward and convenient. Illustration of details of our model structure is shown in Figure 2.

**3.3 Training and testing procedure**

The training phase of our conditional GANs is presented in Algorithm 1. While to keep the BatchNorm layers work correctly, one cannot feed an image batch of same images to test various noise response. So we use multi-round testing with same batch and rearrange them to test different noise responses of each image, which is described in Algorithm 2.

**4 Experiments**

**4.1 Dataset**

There are various kinds of color image datasets, and we choose the open LSUN bedroom dataset\(^1\) [Yu et al., 2015] to conduct our experiment. LSUN is a large color image dataset generated iteratively by human labeling with automatic deep

---

\(^1\)LSUN dataset is available at http://lsun.cs.princeton.edu
neural classification. It contains around one million labeled images for each of 10 scene categories and 20 object categories. Among them we choose indoor scene bedroom because it has enough samples (more than 3 million) and unlike outdoor scenes in which trees are almost green and sky is always blue, items in indoor scenes like bedroom have various colors, as shown in Figure 3. This is exactly what we need to fully represent the capability of our condition GAN model.

In experiments, we use 503,900 bedroom images randomly picked from the LSUN dataset. We crop a maximum center square out of each image and reshape them into \(64 \times 64\) pixel as preprocessing.

4.2 Comparison Experiments

YUV and RGB

The generated colorization results of same grayscale images using \(YUV\) representation and \(RGB\) representation with additional \(L_1\) loss are shown in Figure 4. Focus on the results in red boxes, we can see \(RGB\) representation suffers from structural miss due to the additional \(L_1\) loss. Moreover, the model using \(RGB\) representation shall predict 3 color channels while \(YUV\) representation only predicts 2 channels, which makes the \(RGB\) model training much more unstable.

Single-layer and multi-layer noise

The generated colorization results of same grayscale images using single-layer noise model and multi-layer noise model are shown in Figure 5. We can see from the result that multi-layer noise leads to much more diversity.

Single-layer and multi-layer condition

The generated colorization results of same grayscale images using single-layer condition model and multi-layer condition model are shown in Figure 6. We can see from the results that the multi-layer condition model give the generator more structural information and thus the results of multi-layer condition model are more stable while the single-layer condition model suffers from colorization derivation. More results and discussion of our final model will be shown in the next section.
5 Results and Evaluation

5.1 Colorization Results

A variety of image colorization results by our conditional GANs are provided in Figure 7. Apparently, our fully convolutional (without stride) generator with multi-layer noise and multi-layer condition concatenation produces various kinds of colorization schemes while maintaining good reality. Almost all color parts kept within correct components without deviation.

5.2 Evaluation via Human Study

Previous methods share a common goal to provide a color image closer to the original one. That is why many of their models [Jung and Kang, 2016; Perarnau et al., 2016; Deshpande et al., 2016] take image distance like RMSE (Root Mean Square Error) and PSNR (Peak Signal-to-Noise Ratio) as their measurements. And others [Iizuka et al., 2016; Isola et al., 2016] use additional classifiers to predict if colorized image can be detected or still correctly classified. But our goal is to generate diverse colorization schemes, so we cannot take those distance as our measurements as there exist reasonable colorizations that diverge a lot from the original color image. Note that some previous work [Cheng et al., 2015; Dong et al., 2017; Nguyen et al., 2016b; Nguyen et al., 2016a] do not provide quantified measurements.

Therefore, just like some previous researches [Isola et al., 2016; Zhang et al., 2016], we provide questionnaire surveys as a Turing test to measure our colorization results. We ask each of the total 80 participants 20 questions in which we display 5 color images, one of them is the ground truth image, the others are our generated colorizations of the grayscale image of the ground truth, and ask them if any one of them is of poor reality. We add ground truth image among them as a reference in case of participants do not think any of them is real. And we arrange all images randomly so that participants won’t find the ground truth from their positions. The feedback from 80 participants indicates more than 62.6% of our generated color images are convincible while the rate of ground truth images is 70.0%. Furthermore, we run significance t-test between the percentages of humans rating as real image for each test case. The p-value is 0.1359 > 0.05, indicating our generated results have no significant difference with the ground truth images. Also we calculate the average credibility rank (higher ranking of a image means higher percentage of participants think it is real) of the ground truth image among those generated images is only 2.5 out of 5, which means we successfully fooled about half of the participants.

6 Conclusion

In this paper, we proposed a novel conditional generative adversarial network architecture that will automatically generate diverse colorization schemes for a grayscale image while maintaining their reality. We introduce a novel generator architecture which consists of fully convolutional (without stride) structure with multi-layer noise and multi-layer condition concatenation. With this structure, our model successfully generated diversified high-quality colored images for each input grayscale image. We performed a questionnaire survey as a Turing test to measure our colorization result. The feedback from 80 participants indicates our generated colorization schemes are quite convincible.

For future work, we plan to explore more application scenarios and to achieve a more general purpose unsupervised diverse generative model.
Figure 7: Example results of our conditional GAN on LSUN bedroom data. 20 groups of images, each consists of the leftmost ground truth color image and 4 different colorizations generated by our conditional GANs given the grayscale version of the ground truth image. One can clearly see that our novel structure generator produce various colorization schemes while maintaining good reality.
References

[Charpiat et al., 2008] Guillaume Charpiat, Matthias Hofmann, and Bernhard Scholkopf. Automatic image colorization via modal predictions. In Computer Vision - ECCV 2008, 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part III, pages 126–139, 2008.

[Cheng et al., 2015] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In ICCV 2015, Santiago, Chile, December 7-13, 2015, pages 415–423, 2015.

[Chia et al., 2011] Alex Yong Sang Chia, Shaojie Zhuo, Raj Kumar Gupta, Yu-Wing Tai, Siu-Yeung Cho, Ping Tan, and Stephen Lin. Semantic colorization with internet images. ACM Trans. Graph., 30(6):156:1–156:8, 2011.

[Deshpande et al., 2015] Aditya Deshpande, Jason Rock, and David A. Forsyth. Learning large-scale automatic image colorization. In ICCV 2015, Santiago, Chile, December 7-13, 2015, pages 567–575, 2015.

[Deshpande et al., 2016] Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh, and David A. Forsyth. Learning diverse image colorization. CoRR, abs/1612.01958, 2016.

[Dong et al., 2017] Hao Dong, Paarth Neekhara, Chao Wu, and Yike Guo. Unsupervised image-to-image translation with generative adversarial networks. CoRR, abs/1701.02676, 2017.

[Goodfellow et al., 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 2672–2680, 2014.

[Gupta et al., 2012] Raj Kumar Gupta, Alex Yong Sang Chia, Deepu Rajan, Ee Sin Ng, and Zhiyong Huang. Image colorization using similar images. In Proceedings of the 20th ACM Multimedia Conference, MM ‘12, Nara, Japan, October 29 - November 02, 2012, pages 369–378, 2012.

[Hertzmann et al., 2001] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David Salesin. Image analogies. In SIGGRAPH 2001, Los Angeles, California, USA, August 12-17, 2001, pages 327–340, 2001.

[Iizuka et al., 2016] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 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(4):110:1–110:11, 2016.

[Isola et al., 2016] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. CoRR, abs/1611.07004, 2016.

[Jung and Kang, 2016] Miyoun Jung and Myungjoo Kang. Variational image colorization models using higher-order mumford-shah regularizers. J. Sci. Comput., 68(2):864–888, 2016.

[Koo, 2016] Stephen Koo. Automatic colorization with deep convolutional generative adversarial networks. 2016.

[Levin et al., 2004] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. ACM Trans. Graph., 23(3):689–694, 2004.

[Limmer and Lensch, 2016] Matthias Limmer and Hendrik P. A. Lensch. Infrared colorization using deep convolutional neural networks. In ICMLA 2016, Anaheim, CA, USA, December 18-20, 2016, pages 61–68, 2016.

[Liu et al., 2008] Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung, and Pheng-Ann Heng. Intrinsic colorization. ACM Trans. Graph., 27(5):152:1–152:9, 2008.

[Nguyen et al., 2016a] Tung Duc Nguyen, Kazuki Mori, and Ruck Thawonmas. Image colorization using a deep convolutional neural network. CoRR, abs/1604.07904, 2016.

[Nguyen et al., 2016b] Van Nguyen, Vicky Sintunata, and Terumasa Aoki. Automatic image colorization based on feature lines. In VISIGRAPP 2016 - Volume 4: VISAPP, Rome, Italy, February 27-29, 2016., pages 126–133, 2016.

[Perarnau et al., 2016] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. Invertible conditional gans for image editing. CoRR, abs/1611.06355, 2016.

[Qu et al., 2006] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. Manga colorization. ACM Trans. Graph., 25(3):1214–1220, 2006.

[Reed et al., 2015] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In ICML 2016, New York City, NY, USA, June 19-24, 2016, pages 1060–1069, 2016.

[Ronneberger et al., 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III, pages 234–241, 2015.

[Welsch et al., 2002] Tomihisa Welsh, Michael Ashkhmin, and Klaus Mueller. Transferring color to greyscale images. In SIGGRAPH 2002, San Antonio, Texas, USA, July 23-26, 2002, pages 277–280, 2002.

[Yu et al., 2015] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. CoRR, abs/1506.03365, 2015.

[Zhang et al., 2016] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III, pages 649–666, 2016.