Image Colorization with conditional Deep Convolutional Generative Adversarial Networks

Jianxiang Feng[1], Chengnan Yu[1], Huiming Jin[1], and Siyue Zhou[1]

[1]Technical University of Munich

Abstract

In this project, we managed to solve the problem of automatically colorizing grayscale images using conditional Deep Convolutional Generative Adversarial Networks (cDCGANs). The inputs of our system are grayscale images. We then use cDCGANs to output the predictions of the realistic colorization of the intput images. In addition, we used 2 types of Discriminator and the results demonstrates that introducing conditions into the Discriminator will significantly improve the quality of generated images.

Methodology

- **Dataset**
  - **CIFAR-10**: 50000 training images of size $32 \times 32 \times 3$
  - **STL-10**: 8000 training images of size $96 \times 96 \times 3$
- **Architecture**
  We designed a Generative Adversarial Network (GAN) to generate the colorized imgaes. A GAN is composed of a Generator, who colorizes the grayscale images, and a Discriminator, who estimates a given image is fake (generated) or not to improve the generator.
  - **Generator** The Generator is a downsampling and upsampling network. It takes the concatenation of grayscale images and noises as inputs, then outputs the colorized RGB imgaes depends on the grayscale images. In particular, we used U-Net to implement it in this project.
  - **Discriminator** The Discriminator is a classifier network, who labels the input images as fake or real ones. In this work, we tried two types of Discriminator:
    * **non-Conditional Discriminator** It takes RGB imges as inputs.
    * **Conditional Discriminator** It takes RGB imgaes and their grayscale images (the conditions) as inputs.
- **Formulization** Let D be the estimations of the Discriminator based on the given grayscale images y, and G be the colorized images of y.
  - Discriminator loss function: $J^{(D)} = -\mathbb{E}_{x \sim p_{data}}[\log D(x|y)] - \mathbb{E}_z[\log(1 - D(G(z|y)|y))]$.
  - Generator loss fucntion: $J^{(G)} = -\mathbb{E}_z[\log(1 - D(G(z|y)|y))]$.
    In order to remit the overfitting, we added L_1 loss to it in the implementation.

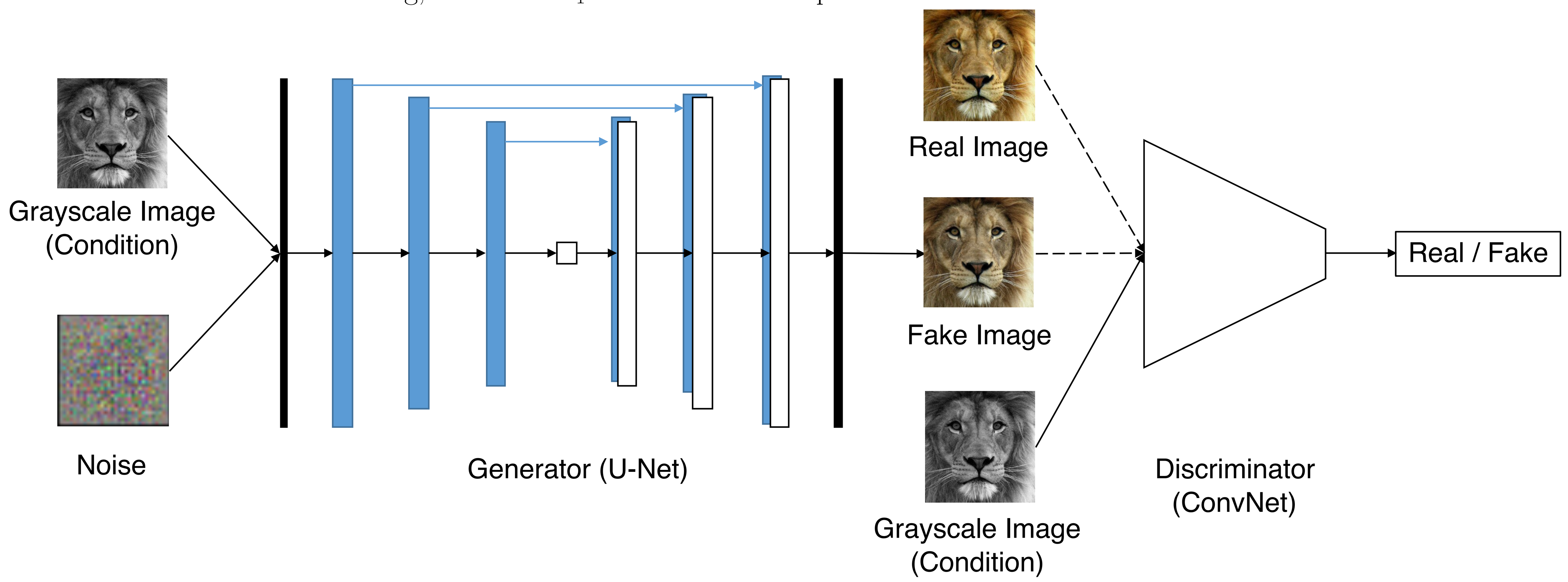

Figure 1: The architecture of the cDCGANs. The conditional Discriminators will take the grayscale images as inputs, and the non-conditional Discriminators will not.

Training

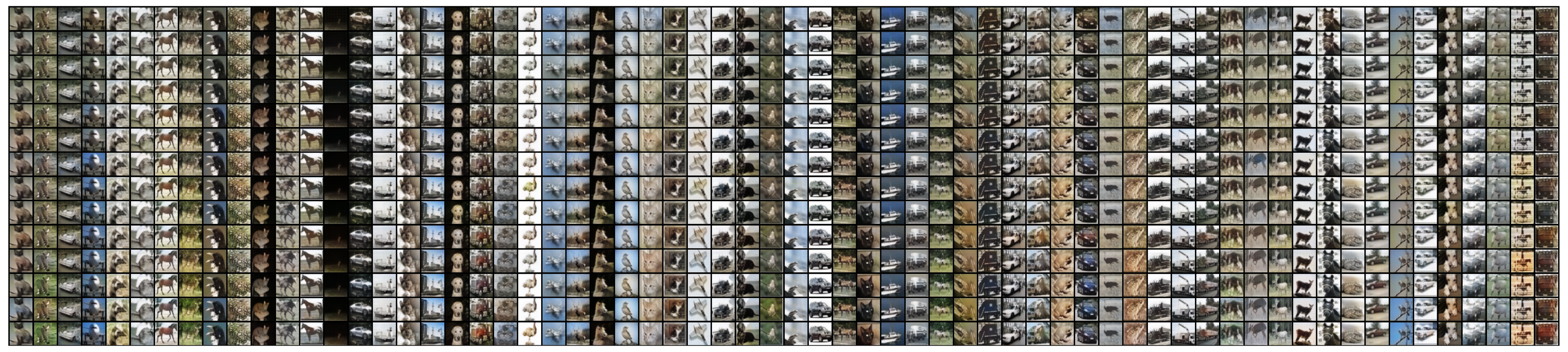

Figure 2: The generated images in training on CIFAR-10, from low epochs to high epochs.

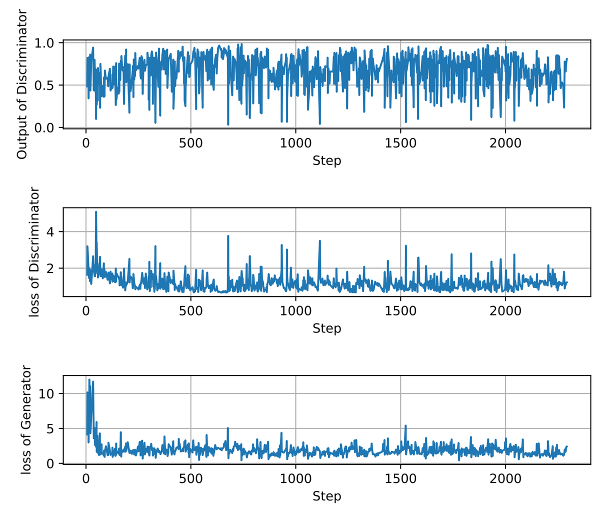

Figure 3: The traning curves of cDCGAN with non-Conditional Discriminator on STL-10.

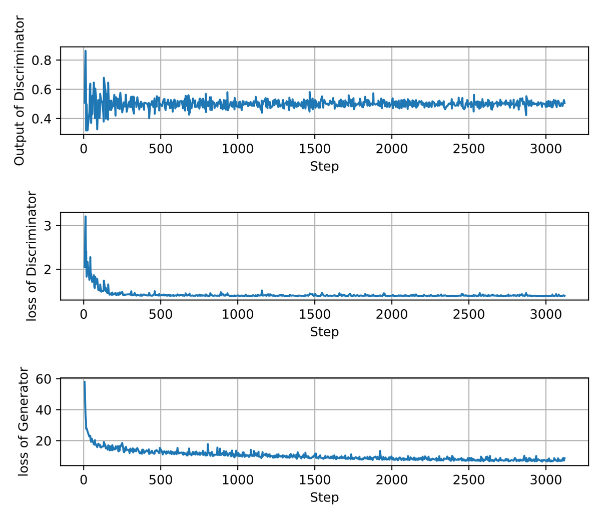

Figure 4: The traning curves of cDCGAN with Conditional Discriminator on STL-10.

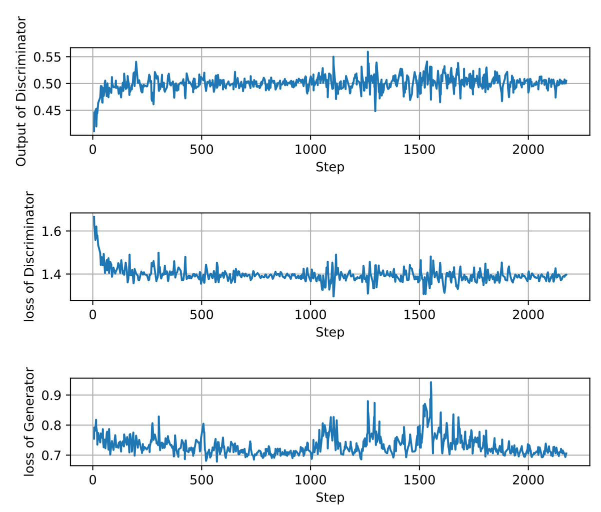

Figure 5: The traning curves of cDCGAN with non-Conditional Discriminator on CIFAR-10.

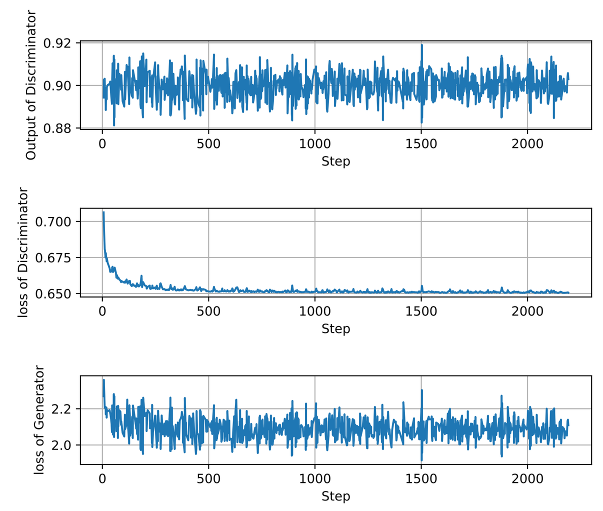

Figure 6: The traning curves of cDCGAN with Conditional Discriminator on CIFAR-10.

Generalization Results

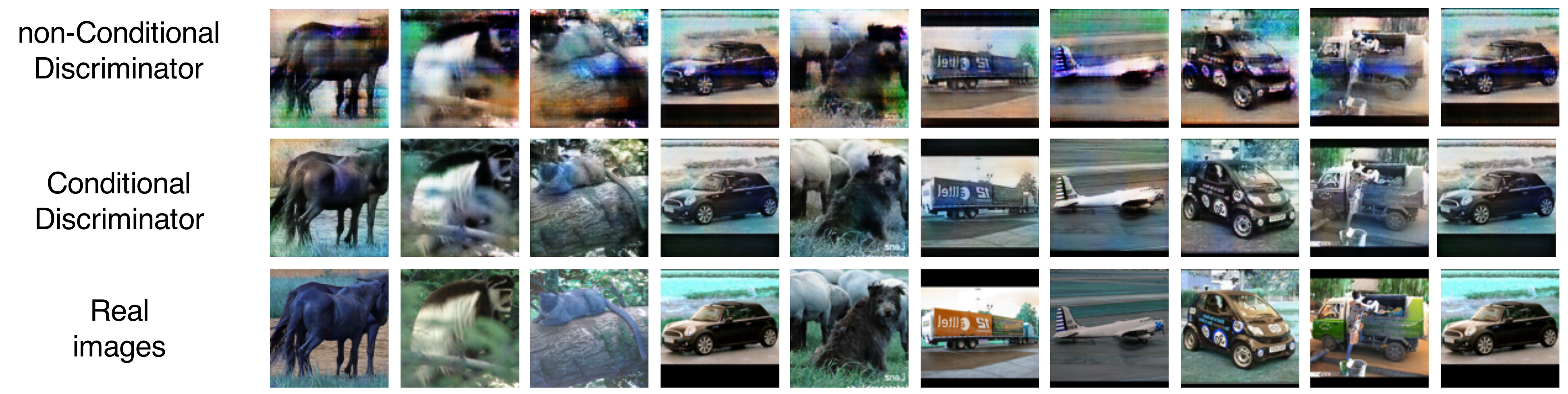

Figure 7: The comparison of cDCGAN with non-Conditional Discriminator and with Conditional Discriminator on STL-10.

Conclusion

- The results demonstrate that cDCGAN can get acceptable outputs, according to the experiments on CIFAR-10 and STL-10.
- Through comparing the training on Dataset CIFAR-10 and STL-10, we can know that small size problem can be solved comparably more easily via GAN.
- According to results of two different structures based on STL-10, the prior conditional knowledge, the gray images, play an important role on the stability of the training and the performance of the generalization of GAN.