Unpaired Style-specific Colorization with Contrastive Learning

Chueng Kit Leong

Abstract

This project consider a style transfer problem from grayscale images to a specific style of colored images. I focus on the case that colored images in specific domain is expensive to obtain. Hence, we demonstrate an unpaired images translation model on colorization. The models on this project is based on CUT, a GAN-based model for image-to-image translation with contrastive learning, which is much more efficient on unidirectional translation than Cycle-GAN. I introduced three additional loss items to improve the performance of CUT on colorization task: (1) Total variance loss. (2) L1 loss of style image. (3) Color variance penalty. The improvement is showed in the result section.

1 Introduction

1.1 Background

Colorization of grayscale image is an active research area in computer vision over the last decade. The task aims to colorize the grayscale image automatically, or with limited hints to reference, by the machines. Interest in this task is usually from the artistic and aesthetics aspects. For example, a semi-automatic colorization system have been proposed for monochrome manga colorization [3]. Besides, due the rich information in colored image, there are several important applications in various domain, such as black and white photos reconstructions, medical image coloring for education purpose [11], and even as a pretext task for self-supervised feature learning [9, 23].

1.2 Paired images translation

In general, colorization task is very in favour of supervised learning. Since the training pairs can be easily obtained by transferring colored image to grayscale. The supervised learning approach for colorization can be summarized in deep neural networks (DNNs) based [2, 9, 19, 23] and generative adversarial networks (GANs) [4] based [1, 5, 6, 8, 11, 14, 18, 22, 25]. Both methods can achieve high-quality results. GANs-based model is relatively new trend on the image-to-images translation tasks such as colorization. Many extended works studied to improve the image generation result, such as conditional GAN [13], and its application on image-to-images translation [6]. All the methods I mentioned above required paired training data, which may be easily to obtained for realistic images.

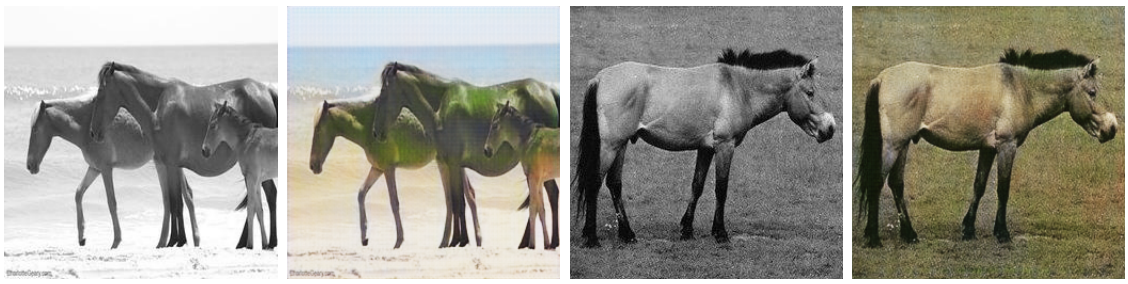


Figure 1: Input (left) and generated (right) image on *pix2pix* for colorization of zebra

1.3 Unpaired images translation

However, obtaining paired data may be difficult in some situations. It is necessary to train colorization model by unpaired data in two cases: (1) the colored images are limited or expensive, for instance, most of the medical image dataset are grayscale. (2) The target style to color is unusual. The second case can also be described as general image style transfer. An experiment in Figure 1 test the result of inputting a grayscale horse into *pix2pix* model [6] trained for coloring zebra. The desired result is reproducing the zebras' color features on horse. However, the resulted coloring on horse tended to be black or white, but failed to reproduce the pattern. This problem is known as domain shift. The paired models required an exact pairs of horse and zebra images in same content to fulfil this task, which is very difficult to obtain.

In my project, I focused on training colorization model by unpaired images. Zhu at el. proposed CycleGAN [24], which allows the learning of unpaired data for images translation. As an extended work, Park at el. proposed CUT [14], which is more faster on image translation in single direction. The colorization model I proposed is based on CUT. To summarize my contrubutions: (1) Demonstration on CUT(a GAN-based model) and its application on colorization. (2) Comparison on the result and training time on CycleGAN, CUT, and different variations of my model. (3) Demonstration and results on introducing 3 additional loss functions to CUT to improve the performance, specific to the colorization task.

2 Related Works

2.1 Semi-automatic Colorization

Before the trend of deep neural networks, colorization system is usually required the aid of human instruction. An effective scribble-based coloring algorithm have been proposed by Levin et al. [10]. In their method, user only needs to indicate the color of image by scribbling. The image will be fully colorized by the neighbour pixels and their similarity in luminance. The method is speeded up by the later work and applied in video coloring [21]. Semi-auto colorizing is still useful nowadays to ensure high-quality results, such as colorizing manga characters by reference images [3].

2.2 Neural Colorization Models

Thanks to the large-scale dataset of colored images, DNNs can be directly applied on image classification task. Cheng et al. proposed a multilayer NN with pixel-wise L2 loss to learn the gray-to-color mapping [2]. Also, convolutional NNs (CNNs) are widely used as an image feature extractor. Zhang et al. [23] showed CNNs is capable to give more plausible colorization results. However, L2 loss is not robust for the colorization due to the multimodal uncertainty [23]. Most of the extended works make effort to modify the loss function, such as predicting the color histogram [9] and preforming color propagation [19].

2.3 Conditional GANs

The first GAN paper proposed adversarial loss [4], which can be described as a zero-sum game between generator and discriminator. The original model generated image from random noise, and it is flexible to be used in different settings. GANs in conditional settings (cGAN) [13] are found to be useful for images translation task, since the generator learns the mapping from input x and noise z to output y, $G : \{x, z\} \rightarrow y$. The loss function can be combined with adversarial loss and L1/L2 loss to further restrict the output. Based on the cGAN, *pix2pix* [6] made a huge contributions on images translation. They proposed the U-Net based generator [15] and the PatchGAN discriminator, which became the popular setting of GAN for images translation in recent. Lots of extended works are focus on image colorization, such as adapting multi-scale discriminators [5], reducing the uncertainty of coloring by learning the mapping between noise and output representation [25].

2.4 CycleGAN and Contrastive Unpaired Translation (CUT)

As I mentioned that *pix2pix* required paired training data for images translation. CycleGAN [24] allows training with two collections of images X, Y, without paired data. The model aims to learn two mappings between collections $G : X \rightarrow Y$ and $F : Y \rightarrow X$ by two losses, adversarial loss and new proposed cycle consistency loss. Mathematically, $F(G(x)) \approx x$. Cycle consistency loss is the L1 loss on $F(G(x))$ and x. It can also be applied on the counterpart $G(F(y)) \approx y$. Figure 2 illustrated the model structure. DiscoGAN [8] and DualGAN [22] are the concurrent works with CycleGAN and shared the same idea. In recent works, CycleGAN is applied on colorizing medical images [11] with modification in loss function.

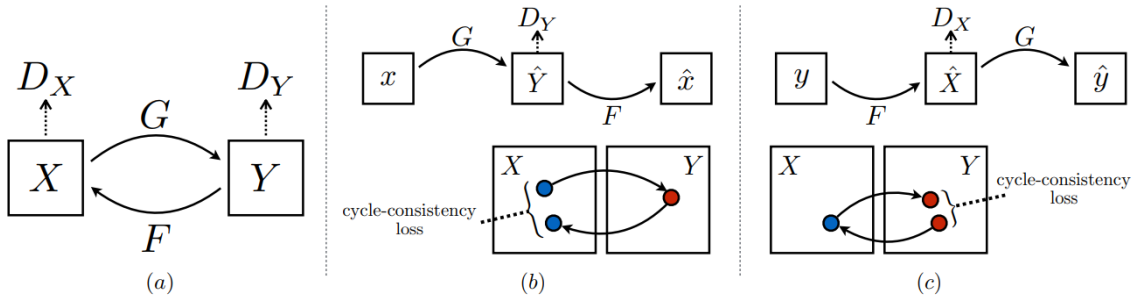


Figure 2: (a) The CycleGAN architecture. (b) The cycle-consistency loss on x (c) The counterpart on y. The figure is retrieved from the original CycleGAN paper [24].

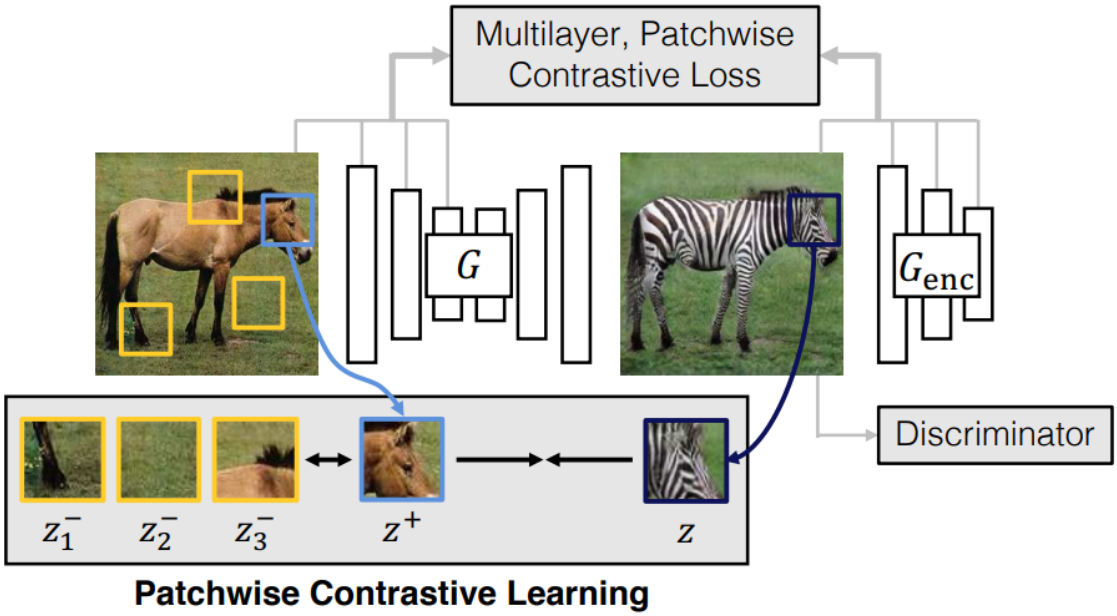


Figure 3: The model architecture of CUT. The figure is retrieved form the original CUT paper [14].

CycleGAN is capable to learn a bidirectional mappings between X and Y. However, colorization is a unidirectional process. Cycle-consistency loss is costly in training time. Park et al. proposed CUT [14], which allowed one directional unpaired images translation and reduce the training time. Their method only used adversarial loss to encourage the correct mapping between two domains. In addition, they used a patch-wise contrastive loss to encourage the the output and input share content. Compared to CycleGAN, CUT is faster, less memory-intensive, and also sufficient to fulfil our task. The details in architecture of CUT and the difference with my model will be presented in the next section.

3 Method

I presented CUT model and contrastive loss. Since my model inherited the major components of CUT, I will describe the detailed model architecture of CUT and my model in this section. Afterwards, I will state the formulation and objectives of different loss functions in my models.

3.1 Model Architecture

Figure 3 illustrated the model structure of CUT. The input distribution (grayscale images) is denoted by $\mathcal{X}$ and the target distribution is denoted by $\mathcal{Y}$. The objective is learning the mapping $G : \mathcal{X} \to \mathcal{Y}$. Different from CycleGAN, CUT only learn one generator in the whole model.

To preserve the content of input image, for example, the shape of horse, CUT introduced patch-wise constrative learning. The objective is maximizing the similarities of patch information between input and output images, which is formatted as a classification problem. The patch information of output images is compared to the corresponding patch of input as a positive example, and non-corresponding

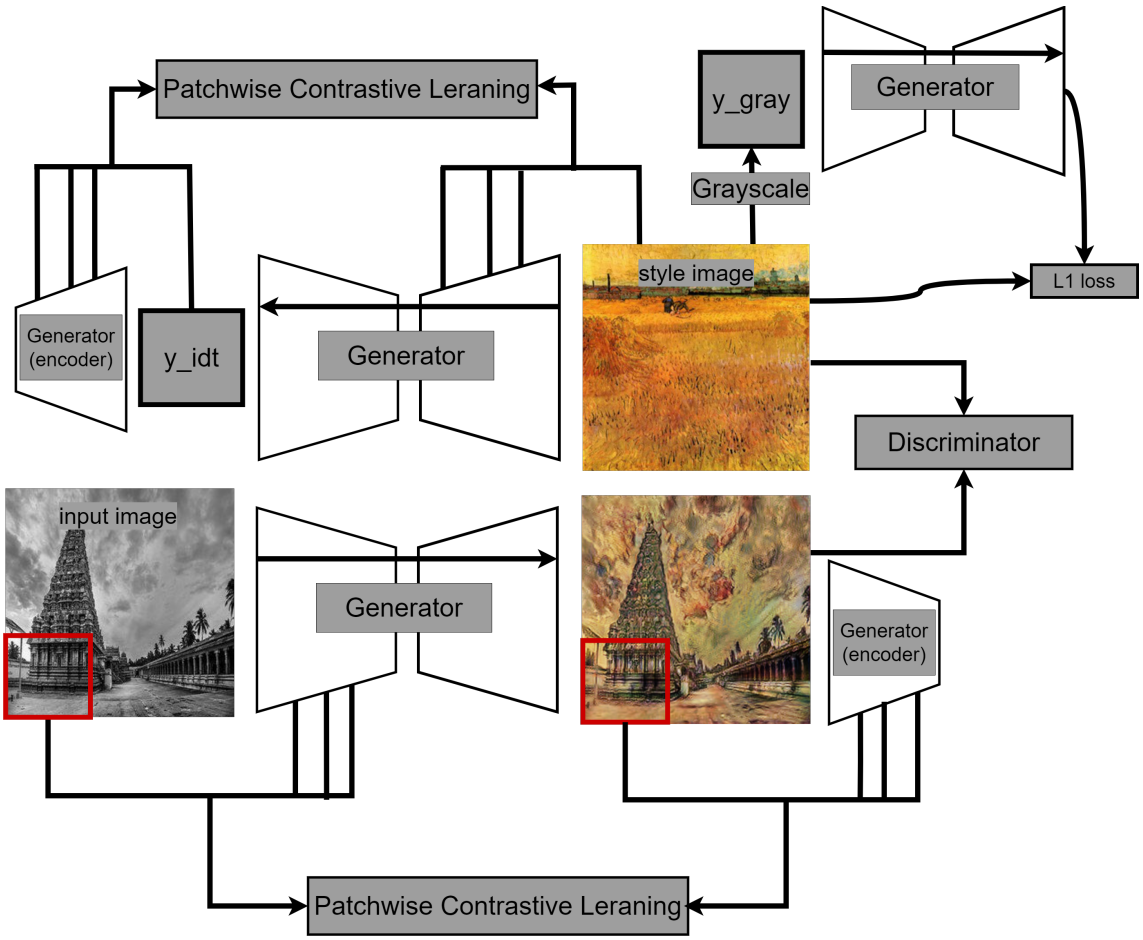


Figure 4: The full model architecture of this project.

patch of input as a negative examples. The cross-entropy loss is calculated:

$$l(\boldsymbol{v}, \boldsymbol{v}^+, \boldsymbol{v}^-) = -\log\left[\frac{\exp\frac{\boldsymbol{v}\cdot\boldsymbol{v}^+}{\tau}}{\exp\frac{\boldsymbol{v}\cdot\boldsymbol{v}^+}{\tau} + \sum_{n=1}^{N}\exp\frac{\boldsymbol{v}\cdot\boldsymbol{v}^-}{\tau}}\right]$$

where $\tau = 0.07$ is a scaling hyperparameters, $\boldsymbol{v}$ is the query patch in the output, $\boldsymbol{v}^+$ is the corresponding patch in the input and $\boldsymbol{v}^-$ is the non-corresponding patch in the input.

To define the patch information, CUT use the embeddings in the encoder of the generator G. Embeddings from the different layers represented different patch size, and are passed to a small MLP layer, denoted by H, to produce a stack of features. The features with the same patch size is compared by the above formula. I omit the detail formulation of patch-wise contrastive loss, which is denoted by $\mathcal{L}_{NCE}(G, H, X)$ and can be found in the original CUT paper[14].

The patch-wise contrastive loss can also be adopt by inputting style image to the generator, to enhance the content preserving ability of the generator. This loss item is denoted by $\mathcal{L}_{NCE}(G, H, Y)$, which is also used in CUT but omitted on Figure 3. Figure 4 illustrated my model structure and show the usage of $\mathcal{L}_{NCE}(G, H, Y)$. Also, to learn the color distribution of style image, L1 loss of the colorization on style image is added in the loss item, which is showed by figure 4 also. The formulation on this loss item will be stated on next subsection.

The generator used is a ResNet-based generator with downsamapling and upsampling operations from [7], and the discriminator is a 70×70 PatchGAN discriminator

[6].

3.2 Loss Functions

Based on the CUT model, several extra loss items are applied to improve the performance. The first loss item is **total variance loss** [11, 16, 17], which is widely used in image generation to obtain a smooth image.

$$\mathcal{L}_{TV}(Z) = \sum_{i,j} \left((Z_{i,j+1} - Z_{i,j})^2 + (Z_{i+1,j} - Z_{i,j})^2 \right)^{\frac{B}{2}}$$

where $B = 2$ is a hyperparameter. The TV loss is computed of the output image, i.e. $Z = G(X)$.

The second item is the L1 loss on passing grayscale style image into the generator. This loss item is preformed as the loss function in paired image colorization. I assumed the mapping learned by this loss can be approximate to the mapping $\mathcal{X} \to \mathcal{Y}$, and the effect is showed on the next section.

$$\mathcal{L}_{l1}(G, Y) = \|G(Y_{\text{gray}}) - Y_{\text{gray}}\|_1$$

where Y_{gray} is the style image after grayscale preprocessing.

The lass item is a simple penalty to alleviate the mode collapse problem. I found that the model tends to colorize the input image by similar color, which should be the mean color of the dataset. There are many approach increase the diversity of generated image, such as compare the loss in color distribution histogram [1, 18], or enhance the effect on noise in cGAN [20]. However, most of the approaches required a more complex model architecture and significant increase in training time. I proposed a simple **color variance** penalty, to encourage the model use more various color on output.

$$\mathcal{P}_{Var}(Z) = \min\{\mathbb{V}_Z[a], \mathbb{V}_Z[b]\}$$

where $Z = G(X)$. The output image is transform from RGB color space to CIELAB color space. $\mathbb{V}_Z[a], \mathbb{V}_Z[b]$ represented the variance of a and b value on the output image.

The effect of above loss items are presented in the next section. We summarize all the loss item, two NCE losses and the GAN loss. The GAN loss $\mathcal{L}_{GAN}$ I used is the LSGAN loss [12], which give the most stable training process when comparing with different GAN loss function. The final objective in one training pairs (X, Y) (no correspondence) is:

$$\mathcal{L}_{\text{NCE\_both}} = \frac{1}{2}(\mathcal{L}_{NCE}(G, H, X) + \mathcal{L}_{NCE}(G, H, Y))$$

$$\mathcal{L} = \lambda_{GAN}\mathcal{L}_{GAN} + \lambda_{NCE}L_{\text{NCE\_both}} + \lambda_{TV}\mathcal{L}_{TV}(Z) + \lambda_{l1}\mathcal{L}_{l1}(G, Y) - \lambda_{Var}\mathcal{P}_{Var}(Z)$$

4 Data

The CycleGAN paper [24] provided a number of datasets that is well-fitted for images translation task. The dataset used in this project is *vangogh2photo*. I used it in inverse direction, transform grayscale photo to vangogh color style. The dataset

Figure 5: The examples of the *vangogh2photo* dataset.

contains over 7000 images for each classes. Some examples are shown on figure 5. Besides grayscale processing, only random flip is applied for preprocessing. The image size are 256×256 and resize and crop are not applied.

All datasets can be obtained from https://people.eecs.berkeley.edu/~taesung_park/CycleGAN/datasets/. These datasets can be directly used in colorization after simple grayscale processing.

5 Experiment

5.1 Training details

Due to the limitation of my local machine, I trained each models with the training set with about 6000 images in 30 epochs only. The training loss is possible to further converge if train the models for a longer time. The initial learning rate is set as 5×10^{-3} and reduce to 0 finally by a cosine annealing scheduler, which is

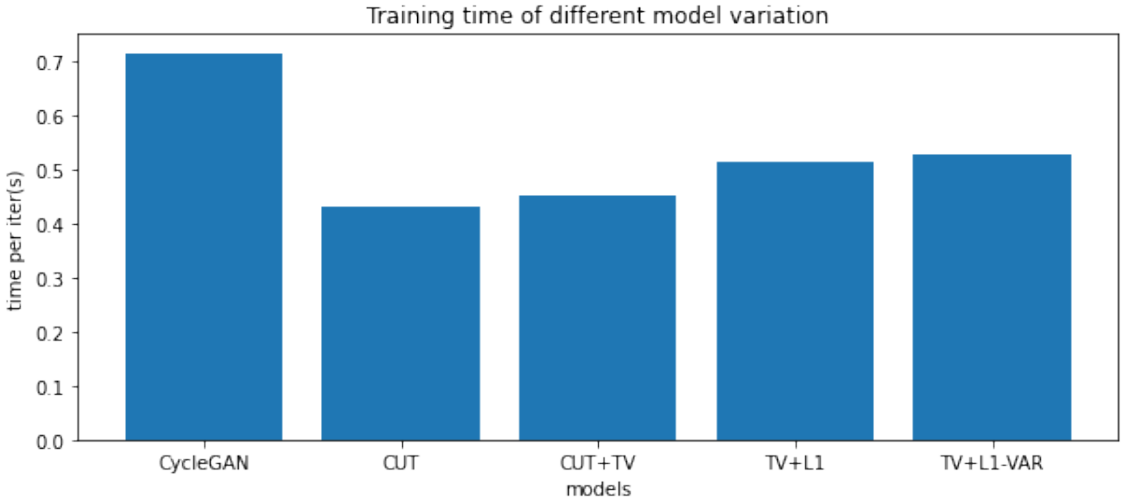
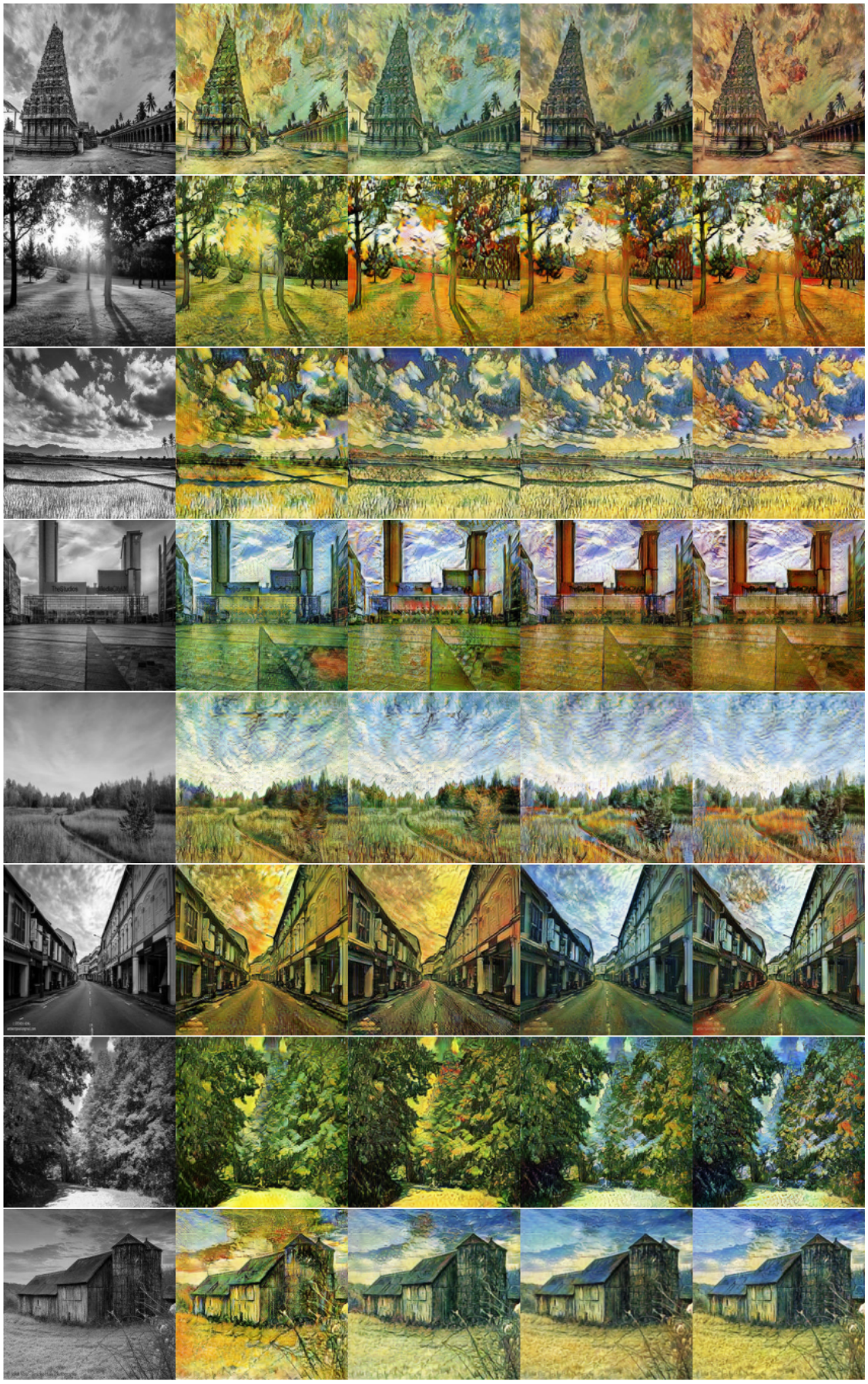


Figure 6: The training time of different models



Input CUT CUT+TV TV+L1 TV+L1-Var

Figure 7: Model result

found to produce the most stable training process. I used Adam optimizer without weight decay. The batch size is set to 1 and the model is trained on single RTX 2060 GPU. The whole training process for CUT-based models took roughly 16 hours.

5.2 Results

Figure 6 plot the training time of different models. We can observed that CUT-based model is much more faster than CycleGAN since CUT only train one pair of generator and discriminator. Also, the addition of loss items made a minor increase in training time. L1 loss gives a relatively great increase since it requires one more forward pass to the generator. TV loss and color variation penalty are computational efficient, and almost make no increase in training time obviously.

FIgure 7 listed the image generation results of different variation of my model. In this subsection, I will examine the observations on each addition loss item.

The vanilla CUT model give a desire result on preserving the input content. Unlike traditional image colorization model, my approach does not preserve the luminance information of the input image, but generate a completely new RGB image. However, the contrastive learning successfully generate image with almost the same content as the input, which is proven to be capable for image classification task. Also, a small change on the image content produced a desire effect on style transfer. The generated images have a texture of oil painting, which is inherited from the style image in the dataset.

The unwanted noise in the output is a usual problem in image generation, which is also appeared in the result of vanilla CUT model. Noise can also be observed in upper part of the third and fifth input sample in figure 7. By adding TV loss (the column of CUT+TV), which give a smoothing effect to reduce the noise in the output. In the last input sample in figure 7, an effect of color correction by the neighbor pixels can also be observed. In my experiment, the weight of TV loss is set to be 0.1 of the total loss. Since the blurring effect of TV loss, large weight easily lower the image quality.

Mode collapse issue can be observed in both vanilla CUT and CUT+TV model, the output images tend to be yellow and green. It is probably because the mean color of the dataset is yellow and green. L1 loss is significant to improve the model, to output different color based on the image content. In figure 7, we can observed that the model learn to color the sky blue, but may failed without the L1 loss.

Color variation penalty is added to further alleviate the model collapse issue by encouraging the model out different color in an image. In the second, fourth and the last input samples, the output is quite desirable since it looks like more colorful. However, some unwanted effect, such as color bleeding and unnatural coloring may be observed. We can conclude that color variation penalty give an unrobust result but indeed increase the colorfulness of the output. Considering its low cost in computation, it may be a possible loss item in some situation.

6 Conclusion

In this project, I focused to demonstrate a colorization GAN-based model, which allow to specify a color style by a collection of style images that is unpaired to the input images. My work is based on CUT model, with modification in loss function. In my result, I showed that my modification improved the result produced by original

CUT in colorization task without sacrificing a lot of training time, and study the effect of each loss item I made.

Because of the limitations of my local machine, my result is able to be improved further by longer training process and choosing a better loss function. Also, in the original CUT paper, Part et al. also produced sinCUT, which allows style transfer by giving a single style image. It can be easily implemented on my model as a further work.

References

[1] M. Afifi, M. A. Brubaker, and M. S. Brown, *Histogan: Controlling colors of gan-generated and real images via color histograms*, 2020. eprint: arXiv:2011.11731.

[2] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 415–423. DOI: 10.1109/ICCV.2015.55.

[3] C. Furusawa, K. Hiroshiba, K. Ogaki, and Y. Odagiri, *Comicolorization: Semi-automatic manga colorization*, 2017. eprint: arXiv:1706.06759.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial networks*, 2014. eprint: arXiv:1406.2661.

[5] M. Górriz, M. Mrak, A. F. Smeaton, and N. E. O'Connor, *End-to-end conditional gan-based architectures for image colourisation*, 2019. eprint: arXiv:1908.09873.

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-image translation with conditional adversarial networks*, 2016. eprint: arXiv:1611.07004.

[7] J. Johnson, A. Alahi, and L. Fei-Fei, *Perceptual losses for real-time style transfer and super-resolution*, 2016. eprint: arXiv:1603.08155.

[8] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, *Learning to discover cross-domain relations with generative adversarial networks*, 2017. eprint: arXiv:1703.05192.

[9] G. Larsson, M. Maire, and G. Shakhnarovich, *Learning representations for automatic colorization*, 2016. eprint: arXiv:1603.06668.

[10] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004, ISSN: 0730-0301. DOI: 10.1145/1015706.1015780. [Online]. Available: https://doi-org.easyaccess2.lib.cuhk.edu.hk/10.1145/1015706.1015780.

[11] Y. Liang, D. Lee, Y. Li, and B.-S. Shin, *Unpaired medical image colorization using generative adversarial network*, 2021. eprint: https://doi.org/10.1007/s11042-020-10468-6.

[12] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, *Least squares generative adversarial networks*, 2016. eprint: arXiv:1611.04076.

[13] M. Mirza and S. Osindero, *Conditional generative adversarial nets*, 2014. eprint: arXiv:1411.1784.

[14] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, *Contrastive learning for unpaired image-to-image translation*, 2020. eprint: arXiv:2007.15651.

[15] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, 2015. eprint: arXiv:1505.04597.

[16] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992, ISSN: 0167-2789. DOI: https://doi.org/10.1016/0167-2789(92)90242-F. [Online]. Available: https://www.sciencedirect.com/science/article/pii/016727899290242F.

[17] Y. Taigman, A. Polyak, and L. Wolf, *Unsupervised cross-domain image generation*, 2016. eprint: arXiv:1611.02200.

[18] P. Vitoria, L. Raad, and C. Ballester, *Chromagan: Adversarial picture colorization with semantic class distribution*, 2019. eprint: arXiv:1907.09837.

[19] S. Wan, Y. Xia, L. Qi, Y.-H. Yang, and M. Atiquzzaman, "Automated colorization of a grayscale image with seed points propagation," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1756–1768, 2020. DOI: 10.1109/TMM.2020.2976573.

[20] D. Yang, S. Hong, Y. Jang, T. Zhao, and H. Lee, *Diversity-sensitive conditional generative adversarial networks*, 2019. eprint: arXiv:1901.09024.

[21] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1120–1129, 2006. DOI: 10.1109/TIP.2005.864231.

[22] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2868–2876. DOI: 10.1109/ICCV.2017.310.

[23] R. Zhang, P. Isola, and A. A. Efros, *Colorful image colorization*, 2016. eprint: arXiv:1603.08511.

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, 2017. eprint: arXiv:1703.10593.

[25] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017, pp. 465–476. [Online]. Available: http://papers.nips.cc/paper/6650-toward-multimodal-image-to-image-translation.pdf.