Extending GAN and VAE to Model Multimodal Distribution

Jinxu Zhang Wei-Cheng Huang

***Abstract*—The generative ability of many models like CycleGAN is limited due to its one-to-one image mapping nature while the image-to-image translation is actually ambiguous in many cases. Therefore, we would like to implement the model BicycleGAN [1] which enforces the connection between output and latent space values to extend the output to multimodal distribution. The idea of BicycleGAN is to combine the cVAE-GAN and cLR-GAN to explicitly learn a latent distribution that encodes the uncertainty information in image-to-image translation. Through our implementation and experiments, we show that BicycleGAN can generate diverse and realistic images across a wide range of image-to-image translation problems, as shown in Figure 1.**

I. Introduction

Recent advances in deep learning have significantly improved the model's ability in conditional image generation. For instance, neural networks have been employed to fill in missing areas of an image [2, 3], convert grayscale images to color [4, 5], and create photorealistic images from sketches [6]. However, most of these approaches focus on generating a single output. In this project, we use BicycleGAN: a model that considers a distribution of possible results, to generate results that are both perceptually realistic and diverse, while still remaining faithful to the input.

A common approach to representing multimodality is learning a low-dimensional latent code, which captures aspects of the possible outputs that may not present in the input image. However, such approaches usually encounter mode collapse problem [7], where only a small number of real samples get represented in the output. In our project, we show that using BicycleGAN could effectively handle the problem.

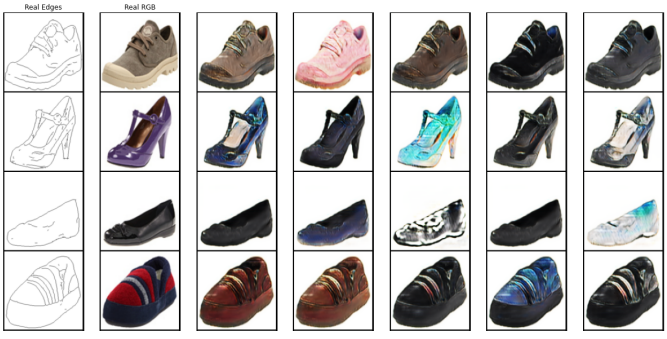


Fig. 1. Sample generated images from our BicycleGAN

II. Background and Related Work

A. Generative Modeling

Modeling the natural image distribution parametrically is a difficult problem. Variational autoencoders (VAEs) [8] offer an effective method for incorporating stochasticity within the network by reparameterizing the latent distribution during training. Generative adversarial networks (GANs) [9] address this issue by mapping random values from a simple-to-sample distribution (such as a low-dimensional Gaussian) to output images in a single feedforward pass of a network. The BicycleGAN implemented in this project is also based on variants of VAEs and GANs, whichi would be discussed in Section III.

B. Conditional Image Generation

All of the generative models can be easily conditioned. Conditional VAEs [10] and autoregressive models [11, 12] have demonstrated success in this regard, and image-to-image conditional GANs have shown the most promise. While these methods have resulted in a significant improvement in the quality of the results, they have come at the cost of multimodality, which has been shown in several studies. [6, 13]

C. Approaches for Multimodality

There are some methods to express multiple modes, one can choose to encode modes explicitly, and combine them with input image as the additional input, as shown in works like iGAN [14] and pix2pix [2], also, using a mixture of models can also be effective. However, these methods all suffer from not being able to produce continuous changes. For now, generating multimodal outputs under conditional image-to-image generation settings are still far from being satisfactory compared with unconditioned or text-conditioned settings.

III. Methodology

In this project, we followed the methodology and architecture given by the BicycleGAN paper [1]. BicycleGAN is a hybrid model that consists of two components: Conditional Variational Autoencoder (cVAE-GAN) and Conditional Latent Regressor GAN (cLR-GAN). They share the same generator and encoder but use separate discriminators. As is shown in Figure 2. Next, we will first describe the network architecture and workflow of this model. Then we will also mention the detailed choice of the generator and encoder and the dataset we target on.

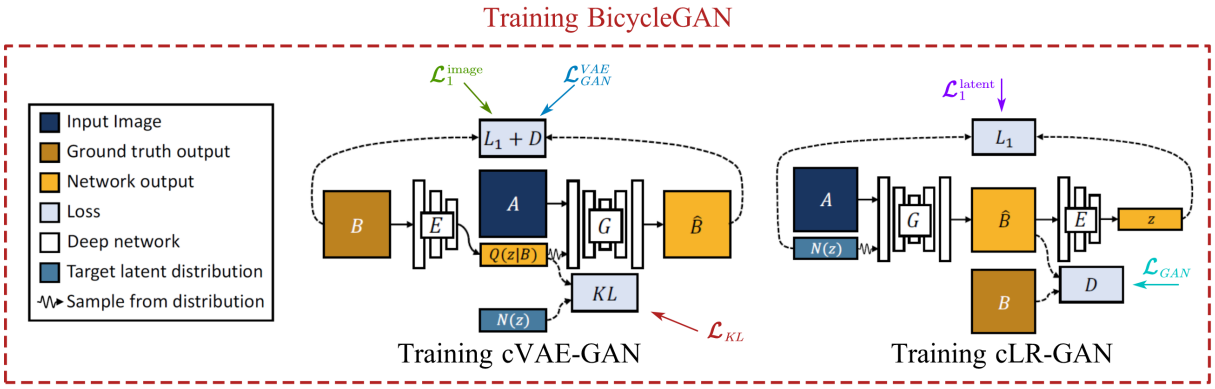


Fig. 2. Architecture of BicycleGAN: a hybrid model

A. Architecture

1) cVAE-GAN:

As shown in Figure 2 (left part), cVAE-GAN first encodes the ground truth image B into the latent space using encoder E. Then generator G takes in an input image A as well as the latent code z which is the encoded ground truth image, returning a generated output image $\hat{B}$ The flow of cVAE-GAN is $B \rightarrow z \rightarrow \hat{B}$

The objective functions associated with cVAE-GAN are the following three:

$$\mathcal{L}_1^{\text{image}}(G) = \mathbb{E}_{A,B\sim p(A,B),z\sim E(B)} \parallel B - G(A,z) \parallel_1 \quad (1)$$

$$\begin{aligned}\mathcal{L}_{GAN}^{VAE}(D,G,E) &= \mathbb{E}_{A,B\sim p(A,B)}[\log(D(A,B))] \\ &+ \mathbb{E}_{A,B\sim p(A,B),z\sim E(B)}[\log(1 - D(A,G(A,z)))]\end{aligned} \quad (2)$$

$$\mathcal{L}_{KL}(E) = \mathbb{E}_{B\sim p(B)}[\mathcal{D}_{KL}(E(B) \parallel \mathcal{N}(0,1))] \quad (3)$$

Equation (1) is an ℓ_1 loss between the target image B and the generated output image $\hat{B}$ that constraints the transformed image $G(A,z)$ to be similar to the target. In equation (2) $\mathcal{L}_{GAN}^{VAE}$ symbolizes the minimax game played between the generator and discriminator, $D(A,\cdot)$ is an indicator function ruling if the discriminator believes to be the real image associated with the input. In equation (3), $\mathcal{L}_{KL}$, enforces the latent space to be a compact Gaussian distribution, $\mathcal{L}_{KL}$ can be derived in the following way:

$$\begin{aligned}\mathcal{L}_{KL}(E) = 0.5\lambda_{KL} \sum_{\dim(z)} &\left(\exp\left(\log\left(\sigma^2(x)\right)\right)\right. \\ &\left.+\mu(x)^2 - 1 - \log\left(\sigma^2(x)\right)\right)\end{aligned} \quad (4)$$

2) cLR-GAN:

As shown in Figure 2 (right part), cLR-GAN passes the input image as well as a randomly sampled latent vector z through the generator G The generated output $\hat{B}$ is then passed through an encoder E, which tries to regain the latent vector from the output image. The flow of cLR-GAN is $z \rightarrow \hat{B} \rightarrow \hat{z}$.

The objective functions associated with cVAE-GAN are the following two:

$$\begin{aligned}\mathcal{L}_{GAN}(D.G) &= \mathbb{E}_{A,B\sim p(A,B)}[\log(D(A,B))] \\ &+ \mathbb{E}_{A\sim p(A),z\sim p(z)}[\log(1 - D(A,G(A,z)))]\end{aligned} \quad (5)$$

$$\mathcal{L}_1^{\text{latent}}(G,E) = \mathbb{E}_{A\sim p(A),z\sim p(z)} \parallel z - E(G(A,z)) \parallel_1 \quad (6)$$

It can also be observed from figure 2 and figure 3 that $\mathcal{L}_{GAN}$ described in equation (5) is an adversarial GAN loss similar to equation (2). It helps ensure the generated images are as photo-realistic as possible. $\mathcal{L}_1^{\text{latent}}$ is an ℓ_1 loss between the randomly sampled latent code and the Encoder output. It ensures that the generated image $\hat{B}$ can be encoded back into the learned latent space of B. This helps achieve a bijection between the output and latent space and thus encourages diversity among the output images.

3) Bicycle-GAN:

The core model of this project: BicycleGAN, essentially combine the aforementioned two approaches. For this hybrid model, the overall full loss can be written as:

$$\begin{aligned}G^*, E^* = \arg\min_{G,E}\max_{D}\ &\mathcal{L}_{GAN}^{VAE}(G,D,E) + \lambda\mathcal{L}_1^{VAE}(G,E) \\ &+\mathcal{L}_{GAN}(G,D) + \lambda_{\text{latent}}\mathcal{L}_1^{\text{latent}}(G,E) + \lambda_{KL}\mathcal{L}_{KL}(E)\end{aligned} \quad (7)$$

where the hyper-parameters λ, λ_{KL} and λ_{latent} controls the relative importance of each term.

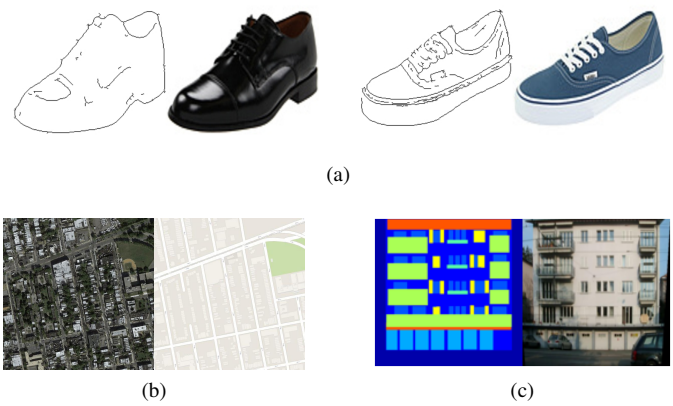
B. Choice of Generator, Encoder, and Discriminator

For the choice of the specific network, we followed the pipeline of the original paper and the code template given. For generator G, we used the U-Net [15] which is an encoder-decoder architecture with symmetric skip connections that have been shown to produce optimal results when spatial correspondence exists between input and output pairs. For the encoder E, we used Resnet-18 [16] to extract relevant features and to estimate the mean and logarithmic variance of the posterior distribution of latent space. For the discriminator D, we used PatchGAN discriminators. During the implementation, we found that U-Net generator mentioned in the original paper is not so well-performed, so we also tried the Resnet generator, which will be discussed in Section IV.

C. Dataset

For this project, we mainly used the Edges2Shoes dataset released as a part of UC Berkeley's Pix2Pix dataset [2]. The dataset consists of 50,025 images (49825 for training and 200

for validation), where each image is a pair of images that contain the colored image of a shoe and its extracted edges or line-silhouette, as shown in figure 3. For faster training, we resized the image to 128×128 pixels from the original 256×256. Though computing resources are limited, we use the whole 49825 images for training in order to achieve the best performance. To test the generalization ability of our model architecture design, we have also tried datasets including Map2Satellite and Label2Facade dataset apart from the given Edge2Shoes dataset.



(a) (b) (c)

Fig. 3. Demonstration of datasets: (a) main target data set: Edges2Shoes (b) Map2Satellite (c) Label2Facade

IV. Experiment

In order to fully test the performance of our BicycleGAN model, we have experimented with a list of different model settings to assure the model output diversity while maintain the realistic output image quality.

A. Baseline Model

We have adopted the U-Net as our model backbone for baseline generator design. To train the model, we use Adam optimizer with learning rate 0.0002 and batch size 32. The model is trained 40 epochs with $\lambda = 10, \lambda_{latent} = 0.5, \lambda_{KL} = 0.01$. The inference result for the baseline model is shown in Figure 4.

B. Baseline Model with ResNet Generator

Based on the encoder and discriminator setting from the baseline model, we have experimented with different generator architecture design using ResNet as backbone and keep all other hyperparameters unchanged. The major difference in ResNet backbone is the introduced residual block where there is no skip connections across layers as in U-Net, but we are still following the general similar workflow where images is first down-sampled and then up-sampled with CNN layers.

1) ResNet Generator with varied λ_{KL}: With the ResNet as our generator backbone, one problem we encounter is to keep the balance between photo-realistic and diversity of the generated image. With the default KL divergence weight factor 0.01 we observe the model fails to diversify the output as

Fig. 4. Inference result from Baseline model

Fig. 5. Inference result from ResNet generator model

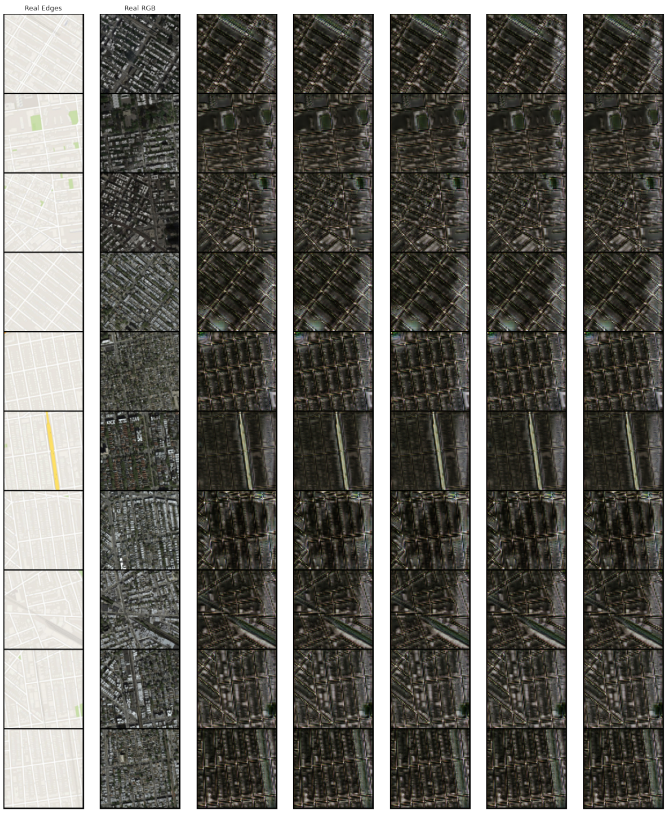


Fig. 6. Inference result in map dataset

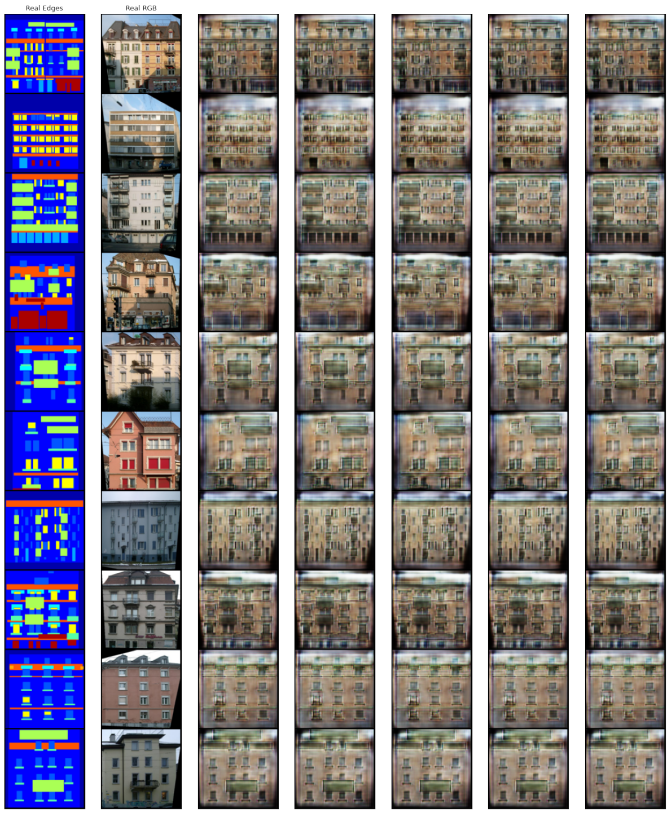


Fig. 7. Inference result in facade dataset

the color doesn't vary much. To deal with this problem, we have experimented with different KL divergence setting with 0.05 and 0.005. Also, we have tried different latent variable setting to increase its dimension from 8 to 32 and it has been observed no significant improvements are made. Therefore, we have chosen the setting of $\lambda_{KL} = 0.05$ as our optimal ResNet generator design. The inference result for the ResNet model with $\lambda_{KL} = 0.05$ is shown in Figure 5.

From the baseline model inference result we can observe that both the generated image photo-realistic quality and diversity are presented. In Figure 4, the first column represents the edge corresponding to the real shoes image in second column. The generated images sampled from the multi-modal distribution are shown from third column to seventh column. Similarly, the inference result from the ResNet based generator model is shown in Figure 5. It can be observed that the generated image from ResNet generator model present more diversities in color and texture variation. However, some of the color filling is inferior to the baseline model, for instance the pink color shoes is embellished with discrete black shade which seems to be fake artifact.

C. Generalization to Different Dataset

In order to test our model generalization ability, we have also experimented with the Map2Satellite and Label2Facade dataset. With the optimal setting from the ResNet generator model and $\lambda_{KL} = 0.05$, some of the inference results are shown in Figure 6 and Figure 7. It can observed that the model is learning to extract some key latent variable representation and reproduce the map or facade patterns from the multi-modal distribution with similar texture and style. However, the image photo-realistic quality is far from perfection compared with the Edge2Shoes dataset. Another problem is about mode collapse since there is almost no diversity among the generated images. We propose one way to mitigate this effect by adopting the Wasserstein loss such that the vanishing gradient from the discriminator can be resolved. Unfortunately, we didn't manage to implement this loss into our model due to limited time.

V. Evaluation

A. Loss curve

To evaluate the training performance of the ResNet based generator model we have visualized the training loss curve of a list of core terms described from Equation 7 in Figure 8.

B. Quantitative Evaluation

1) FID Score: To evaluate the photo-realistic quality, we have chosen to evaluate the FID score between 200 generated images and real images in the validation set. The FID score is defined as

$$FID(r, g) = ||\mu_1 - \mu_2||_2^2 + Tr(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{\frac{1}{2}})$$

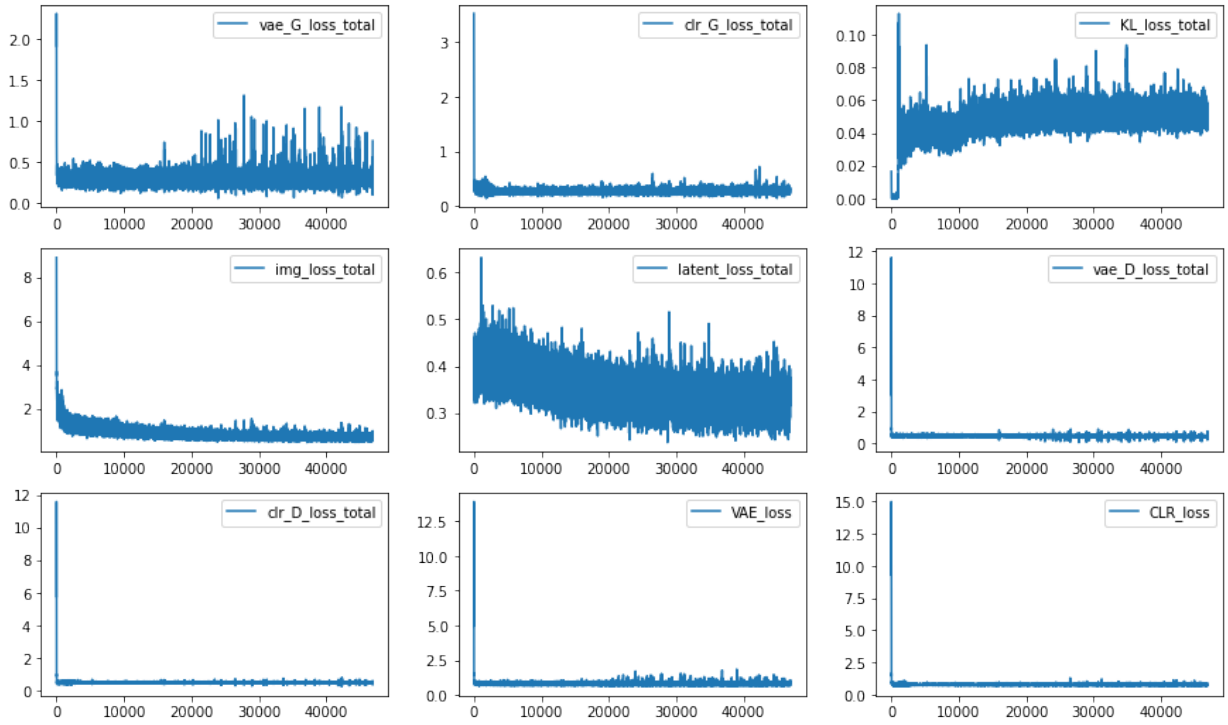


Fig. 8. Training loss curve

which measures the distance between real image set and generated image set. By embedding both the real image and generated image into a feature space given by the output of final avgpool layer of Inception Net v3 model, we can extract the image distribution mean and variance statistics. Then we can calculate the Fréchet distance between these two Gaussian distributions. Our model has reached a FID score of 88.00.

2) LPIPS metric: To evaluate the generated image diversity from our BicycleGAN model, we have adopted the Learned Perceptual Image Patch Similarity (LPIPS) metric [18] which computes the distance between generated image patches. According to our ResNet based generator model it can achieve a LPIPS score of 0.158.

C. Qualitative Evaluation

To evaluate the qualitative performance of the model, we have randomly selected 10 sample images from the validation dataset and present 5 corresponding generated images for each of them. The result is shown in Figure 4 for baseline model and Figure 5 for ResNet generator model.

VI. Summary

In conclusion, we have implemented an extended version of GAN and VAE model to map the multimodal image-to-image translation between two image domains. By combining multiple objectives for encouraging a bijective mapping between latent and output spaces, the generated image can achieve both diversity and photo-realistic quality in conditional generation. With the cVAE-GAN model, we can enforce the latent vector to be meaningful such that it encapsulates the ambiguous aspects of the output mode which are not present in the input image. The cLR-GAN model assures the latent vector can be recovered from the encoder for later distribution reproduce. By combining both cVAE-GAN and cLR-GAN objective together, we can achieve the objective of extending to multimodal image translation and eliminate the mode collapsing problem in conditional GAN image synthesis. To select the optimal model backbone architecture, we have tried both U-Net and ResNet as our model generator and get different generated image quality in terms of both photo-realistic and diversity. To enforce the image diversity, we have experimented with different λ_{KL} and latent vector dimension and select the optimal settings for different backbone model. To test our model generalization ability, we have also explored other datasets including Map2Satellite and Label2Facade. Although the generated images is inferior to the Edge2Shoes dataset in image photo-realistic and diversity, the model still manages to capture part of the common underlying information and reproduce part of the texture and pattern.

For the future work, we think it would help improve the model generalization on other dataset by incorporating Wasserstein loss as the discriminator objective function to deal with the vanishing gradient and mode collapse problem so there will be more diversities in the map and facade generated images.

References

[1] J-Y Zhu, R. Zhang, D. Pathak, T. Darrell,A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In NIPS, 2017.

[2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In CVPR, 2017.
[3] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. In CVPR, 2016.
[4] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. SIGGRAPH, 35(4), 2016.
[5] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In ECCV, 2016.
[6] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. In CVPR, 2017.
[7] I. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160, 2016.
[8] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In ICLR, 2014.
[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014.
[10] K. Sohn, X. Yan, and H. Lee. Learning structured output representation using deep conditional generative models. In NIPS, 2015.
[11] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. PMLR, 2016.
[12] A. v. d. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelcnn decoders. In NIPS, 2016.
[13] W. Xian, P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Texturegan: Controlling deep image synthesis with texture patches. In arXiv preprint arXiv:1706.02823, 2017.
[14] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In ECCV, 2016.
[15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI, pages 234–241. Springer, 2015.
[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.
[17] R. Tylecek, and R. Sára. "Spatial Pattern Templates for Recognition of Objects with Regular Structure." In GCPR, 2013.
[18] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In arXiv preprint arXiv:1801.03924, 2018.

APPENDIX

Our final presentation video is included in this link.