CIS680 Final Project
Extension of GAN and VAE to Model Multimodal Distribution

Zhihua Zhang, Zach Li
{zhiuaz, zachli}@seas.upenn.edu

Abstract

We investigate how to extend our previous work CycleGAN [9] that translates image to another to produce diverse images that vary in color, contrast ratio, and other characteristics. First, we adopt Conditional VAE [13] that gives generator a peek to natural input for high quality image generation, but we decide to keep the discriminator unconditional to force it to take latent code into consideration. We also include a latent regressor [4] to encourage our model to generate diverse but photo-realistic images. Finally, we creatively design a "diversity loss" that further punishes noise-generated images to not be "identical" or very similar to the real target. We run experiments to explore how different hyper-parameters and components such as KL-divergence would affect the quality of image generation. Our final BicycleGAN model can generate diverse and realistic images as shown in Fig. 1.

1. Introduction

VAE [15] and GAN [8] are old but powerful ideas, the the former aims at modeling the latent space to reconstruct images and the latter attempts to synthesize new images that looks photo-realistic. Many successful works have extended upon them for different applications. One is CycleGAN [11] that translates image in source domain to target domain, and vice versa. However, it is limited to generating very specific images for each data pair because of the "cycle" loss design. The model we implemented in this paper, BicycleGAN [20], improves upon that by combines the objectives of two GAN networks, cVAE-GAN [2] and cLR-GAN [4], in order to jointly optimize image-latent-image and latent-image-latent cycles. cVAE-GAN allows the latent code to be learned directly from ground truth data while its space remains close to a known prior distribution to make sampling at test time easier. cLR-GAN stimulates the generator to create different images that still look natural, which discourages memorization during previous cVAE-GAN training. Combining them together, BicycleGAN can generate new images in the target domain that balance both diversity and photo-realism.

Figure 1. Generated samples of BicyleGAN.

2. Related works

Autoencoders [1] are neural networks composed of both an encoder and a decoder trained to learn lower-dimensional

feature representations from unlabeled training data. Due to overfitting, the latent space of an autoencoder can be extremely irregular thus unable to define a generative process that samples a point from the latent space and feed it through the decoder to get new data.

Variational autoencoders [15] addresses this limitation by introducing latent space regularization that allows for a generative process. Instead of encoding an input as a single point, VAE encodes it as a distribution over the latent space and utilizes KL divergence loss to make the distributions returned by the encoder close to a known prior distribution like the Gaussian. However, due to the input random noise and limited reconstruction capabilities of the decoder, generated samples can often be blurry and unrealistic.

Generative Adversial Networks [8] represent another approach to generative models - they frame the task as a supervised learning problem with two networks: a generator that we train to generate new examples from sampled noise, and the discriminator that classify examples as either real or fake. By solving this min-max two-player game, we can end up with a generative model that is very good at producing distinct and realistic images.

CycleGAN [11] proposes a image-to-image translation method by extending vanilla GANs to not only learn the one way mapping from input X to target Y, but also learns the reverse $Y \rightarrow X$. Along with a cycle consistency constraint on such mappings, the network can learn the translation of an image from source domain to target domain in the absence of paired training data.

Although these generative models and their variations such as Conditional GAN [17], InfoGAN [3], DiscoGAN [14], and StyleGAN [12] have gotten very good at generating hyper-realistic samples in a target domain, one significant limitation remain: they do not work well in modeling multi-modal distributions where a single input image may correspond to multiple possible outputs.

Relying on the principles of VAE, GAN, and previous works such as VAE-GAN [16] and latent regressor models [5] [6], BicycleGAN [20] addresses the problem of mode collapse by explicitly constraining the mapping between the output and the latent code to be invertible. This approach allows the trained generator to produce more diverse samples by preventing a many-to-one mapping from the latent code to the output during training.

3. Methodology

3.1. Dataset

We used the Edges2Shoes dataset released as apart of UC Berkeley's Pix2Pix dataset [9]. This dataset consists of 50,025 images, where each image is actually a pair of images that contains the colored image of a shoe and its extracted edges or line-silhouette (see Fig. 2). The provided training set contains 49,825 images totaling to 2.3GBs while the given validation set contains 200 images totaling to 9MBs. We resized the image to 128×128 pixels from the original 256×256 for faster training and normalize the pixel value to the range $[-1, 1]$.

Figure 2. Sample images from the Edges2Shoes dataset.

3.2. Architecture

We followed the architecture and training procedures outlined in the original BicycleGAN paper [20]. The network is a hybrid model that combines the training objectives of both cVAE-GAN and cLR-GAN. They share the same generator and encoder, but uses separate discriminators. A diagram of the architecture is shown in Fig. 3.

3.2.1 Generator, Encoder, and Discriminator

We used U-Net [18] for our generator, which is an encoder-decoder architecture with symmetric skip connections that has shown to produce optimal results when spatial correspondence exists between input and output pairs. Our discriminator follows PatchGAN [10] that assesses image authenticity at fine-grained levels to better discriminate between real and fake candidates. For the encoder, we used Resnet-18 [7] to extract relevant features and and to estimate the mean and logarithmic variance of the posterior distribution of latent space.

3.2.2 cVAE-GAN

Conditional variational autoencoder GAN, or cVAE-GAN, first encodes image B from target domain onto the latent space z represented by distribution $Q(z|B)$. The generator G then uses a sampled latent code (with reparameterization trick) and image A from the input domain to generate a reconstructed image $\hat{B}$ in the target domain.

3.2.3 cLR-GAN

Conditional latent regressor GAN, or cLR-GAN, samples from a known prior distribution and then uses the generator to map image A from input domain and the sampled latent code onto generated output $\hat{B}$ in target domain. Then it attempts to reconstruct the latent code z from $\hat{B}$.

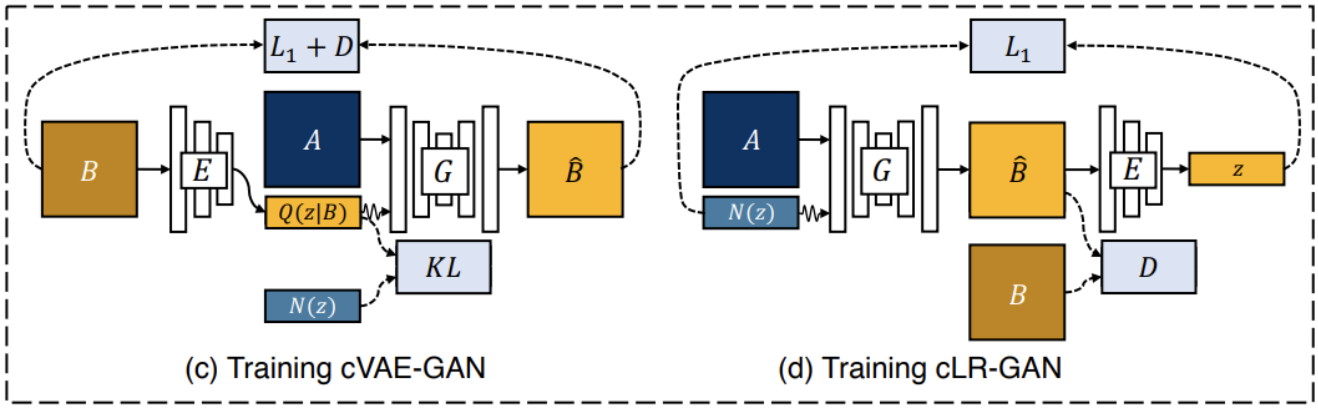

Figure 3. Architecture overview of BicycleGAN.

3.3. Training Procedure

3.3.1 Generator and Encoder

We will begin by first training the generator and encoder through cVAE-GAN and cLR-GAN and disable the gradient for their respective discriminators. After a forward pass through cVAE-GAN, we obtain a fake image $\hat{B}$ generated by G, as well as μ and σ representing the distribution of the latent space z. To enforce the generated image $\hat{B}$ to be close to the real image B in target domain, we implement a simple L_1 distance loss.

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

To encourage generated image to be photo-realistic, we run the discriminator on $\hat{B}$ and calculate a adversarial loss similar to the one in generative adversial networks.

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

Lastly, we will use KL divergence to enforce the latent space z to follow a compact Gaussian distribution, similar to the technique used in variational autoencoders.

$$\mathcal{L}_{KL} = \mathbb{E}_{B\sim P(B)}[D_{KL}(E(B)||N(0,1))] \quad (3)$$

Moving on to forward pass of cLR-GAN, we will first obtain a fake image $\hat{B}$ from the generator and then run it through the encoder to obtain the reconstructed latent code represented by μ and σ. To regularize $\hat{B}$ so it can be encoded back to the learned latent space, we apply a L_1 loss between the sample latent vector used to generate $\hat{B}$ and the prior distribution $p(z)$, which can be represented by μ.

$$\mathcal{L}_1^{latent}(G,E) = \mathbb{E}_{A\sim P(A),z\sim p(z)}||z - E(G(A,z))||_1 \quad (4)$$

To enforce the generated image $\hat{B}$ to be photo-realistic in the cLR-GAN module as well, we will apply a adversarial loss $\mathcal{L}_{GAN}$ similar to what we did in cVAE-GAN. Note that they use separate discriminators albeit with the same architecture in our scenario. Thus the overall training objective of BicycleGAN can be expressed as the following:

$$G^*, E^* = arg \min_{G,E} \max_{D} \mathcal{L}_{GAN}^{VAE} + \lambda \mathcal{L}_1^{image}(G) + \mathcal{L}_{GAN} + \lambda_{latent}\mathcal{L}_1^{latent} + \lambda_{KL}\mathcal{L}_{KL} \quad (5)$$

4. Experiments

Dataset We performed all the experiments using the Edges2Shoes dataset described in Section 3.1.

4.1. Baseline

We trained the model for 20 epochs using $\lambda = 10, \lambda_{\text{latent}} = 0.5, \lambda_{KL} = 0.01$, batch size $bz = 32$, Adam optimizer with learning rate of $0.0002 \times \frac{bz}{32}$, and the latent dimension $nz = 8$. The training results of this baseline setting is shown in Fig. 4, where each loss is scaled by its weight λ. Fig. 5 shows sample images generated by sampled random noise and image A in source domain from the validation dataset.

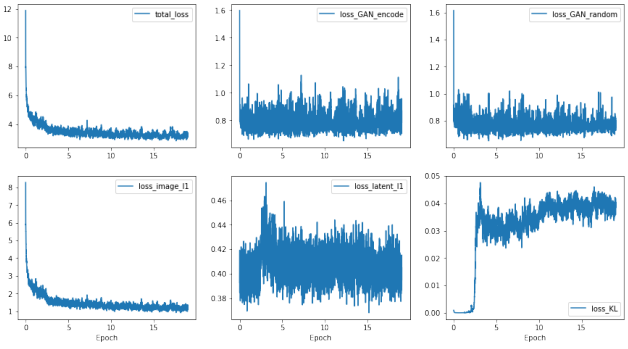

Figure 4. Baseline training curves with $\lambda_{KL} = 0.01$

Figure 5. Baseline visualization with $\lambda_{KL} = 0.01$

Figure 6. $\lambda_{KL} = 0.05$ Visualization

4.2. KL Divergence

The image quality of the baseline model is mediocre for noise-based generation. We observe that KL-divergence $\mathcal{L}_{KL}$ has a much smaller magnitude compared to other objectives, so our initial thought was that encoded latent code had a very different distribution from that of its prior, making it hard to sample valid latent code at inference time. Therefore, we decided to assign greater weight to the KL-divergence target to motivate the encoder to learn a similar distribution.

Using the same setting except $\lambda_{KL} = 0.05$ is 5x larger, we obtain the result displayed in the Fig. 6. The quality improves significantly but at the cost of losing diversity, which suggests there exists mode collapse. We attribute it to that enforcing the latent distribution to be close to standard normal, which needs not be, encourages generator to memorize the output using encoded latent code and ignore the randomly sampled one when creating images.

We then decided to try decreasing λ_{KL} to keep more valid latent information and expect generated images to look realistic but still remain diverse. The result is in Fig. 7, which looks better overall than Fig. 6. However, when compared to Fig. 5, it still encounters the reality versus diversity trade-off that it looks more natural but varies little. Because we value diversity more, we decide to stick with $\lambda_{KL} = 0.01$ and train longer to achieve better results as we can see from Fig. 4 that the loss has not fully converged.

Figure 7. $\lambda_{KL} = 0.002$ Visualization

4.3. Diversity loss

In the above experiments, we realized that there is a trade-off between natural appearance and diversity. Rather than compromising one to make the other better, we want to think of a way to improve the overall quality of image

generation. Motivated by how adversarial attacks can fool a specific model but fails to generalize to fool other well-performed models, we decide to add a "diversity loss" (6) to our objective

$$\mathcal{L}_{\text{diversity}} = -\|B - G(z)\|_1, z \sim p(z) = \mathcal{N}(0, I) \quad (6)$$

Here, B is the real target image and $G(z)$ is our noise generated image. Minimizing this diversity loss is equivalent to applying adversarial attack to the generator by pushing its generated image away from the real target. We expect it to work for the following reasons

1. The generator should be punished if there exists mode collapse that it always generates identical or very similar images to its real target.
2. The generator is encouraged to explore a variety of realistic images in the nature's distribution. They can be far away from our observed training data in terms of pixel values. For example, an edge can correspond to either a white shoe or a black shoe.
3. With a powerful generator opponent, the discriminator is also motivated to learn a better decision boundary to make correct classifications.

Our full objective becomes

$$\begin{aligned} L(G, E) =& \mathcal{L}_{GAN}^{VAE} + \lambda \mathcal{L}_1^{image}(G) + \mathcal{L}_{GAN} + \lambda_{latent}\mathcal{L}_1^{latent} \\ &+ \lambda_{KL}\mathcal{L}_{KL} - \lambda_{diversity}\mathcal{L}_1^{image}(G(z)) \end{aligned} \quad (7)$$

We experimented with different weight of the diversity loss and found $\lambda_{diversity} = 0.05$ performs the best in terms of the diversity and reality of random noise-generated images. The complete evaluation will be reported in the next section.

4.4. Other Modifications

We also tried other changes to seek improvement.

1. Dividing training samples in half and separately feeding them to cVAE-GAN and cLR-GAN stabilizes training and improves the final performance.
2. Passing latent code to only the input or all intermediate layers of the U-Net generator produced similar results.
3. Increasing latent space dimension to nz={16,32} did not lead to significant improvement.

5. Evaluation

5.1. Training Visualizations

Training loss curves and evaluation metrics for our final model trained over 50 epochs is shwon in in Fig. 8. Generated images from intermediate epochs can be found in the appendix.

5.2. Quantitative Evaluation

5.2.1 FID Score

To evaluate the photo-realistic quality of our generated images, we calculate the FID score that measures the distance between the distributions of the generated images and real images in the validation set.

$$FID(r, g) = ||\mu_1 - \mu_2||_2^2 + Tr(\sum_1 + \sum_2 - 2(\sum_1 \sum_2)^{0.5})$$

The features are obtained by from the final average pooling layer of InceptionNet V3 for both sets of images. The final FID score is 107.12 and its change over training is shown in Fig. 8.

5.2.2 LPIPS Metric

To quantify the diversity of our multi-modal generation, we computed the average pairwise perceptual similarity score between 10 samples for each conditional inputs across the entire validation set using the Learned Perceptual Image Patch Similarity (LPIPS) metric [19]. The final LPIPS score is 0.143 and the change over training is shown in Fig. 8.

5.3. Qualitative Evaluation

We generated 8 randomly sampled results as well as the image generated using the encoded latent code for 20 images in the validation set. The output is shown in Fig. 1.

6. Conclusion

Conclusion: In this work, we implemented an extension of CycleGAN that is capable of modeling multi-modal distribution and generate diverse images. We have tried and evaluated several methods to balance the trade-off between mode collapse where model fixates on encoded latent code and ignores samples from prior, and the distortion of generated images. We find that a strict restriction on KL-divergence leads to mode collapse due to loss of valid latent information, while it being too loose can result in blurry outputs because random samples come from an inaccurate distribution. Other experimental settings such as batch size and latent dimension did not have significant effects. To improve the overall results, we propose a "diversity loss" that encourages noise-generated images to be different from the real images in target domain and successfully obtained generated samples that are diverse and photo-realistic.

Future work: There are many promising improvements that could be done. One is to refine the design of "diversity loss". Currently, we are too greedy for outputs to be different, which can make training unstable. Motivated by the effectiveness of the label smoothing technique, we can include a margin to allow some degree of similarity.

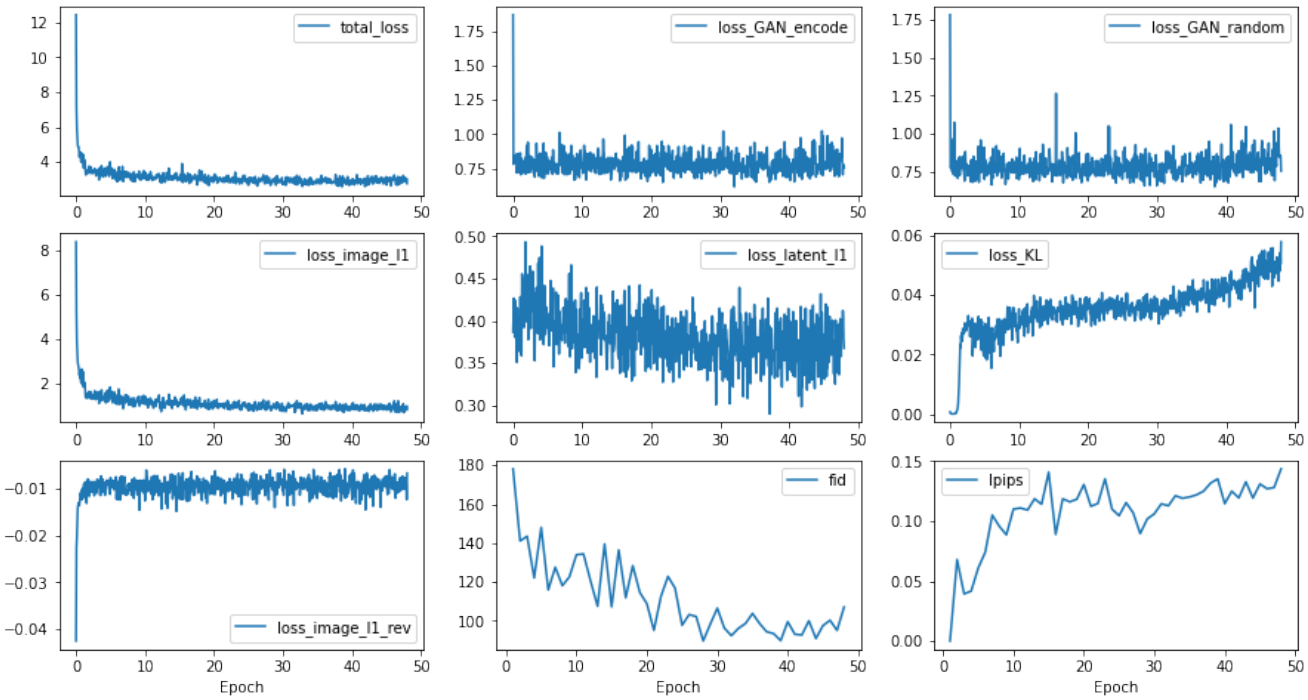

Figure 8. Model loss curves over training.

References

[1] Dana H. Ballard. Modular learning in neural networks. In *AAAI*, 1987. 1

[2] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: Fine-grained image generation through asymmetric training, 2017. 1

[3] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets, 2016. 2

[4] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning, 2017. 1

[5] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning, 2017. 2

[6] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference, 2017. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2

[8] Mehdi Mirza Bing Xu David Warde-Farley Sherjil Ozair Aaron Courville Ian J. Goodfellow, Jean Pouget-Abadie and Yoshua Bengio. Generative Adversarial Networks. In *NeurIPS*, 2014. 1, 2

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018. 1, 2

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018. 2

[11] Phillip Isola Jun-Yan Zhu, Taesung Park and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks . In *CVPR*, 2020. 1, 2

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 2

[13] Xinchen Yan Kihyuk Sohn, Honglak Lee. Learning structured output representation using deep conditional generative models. In *NIPS*, 2015. 1

[14] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks, 2017. 2

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *arXiv preprint*, 2013. 1, 2

[16] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric, 2016. 2

[17] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. 2

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 2

[19] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 5

[20] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation, 2018. 1, 2

Appendix

A. Video link

Please check out this link for our video presentation.

B. Intermediate Image Generation

Figure 9. BicycleGAN epoch=10 visualization (Validation set)

Figure 10. BicycleGAN epoch=20 visualization (Validation set)

Figure 11. BicycleGAN epoch=30 visualization (Validation set)

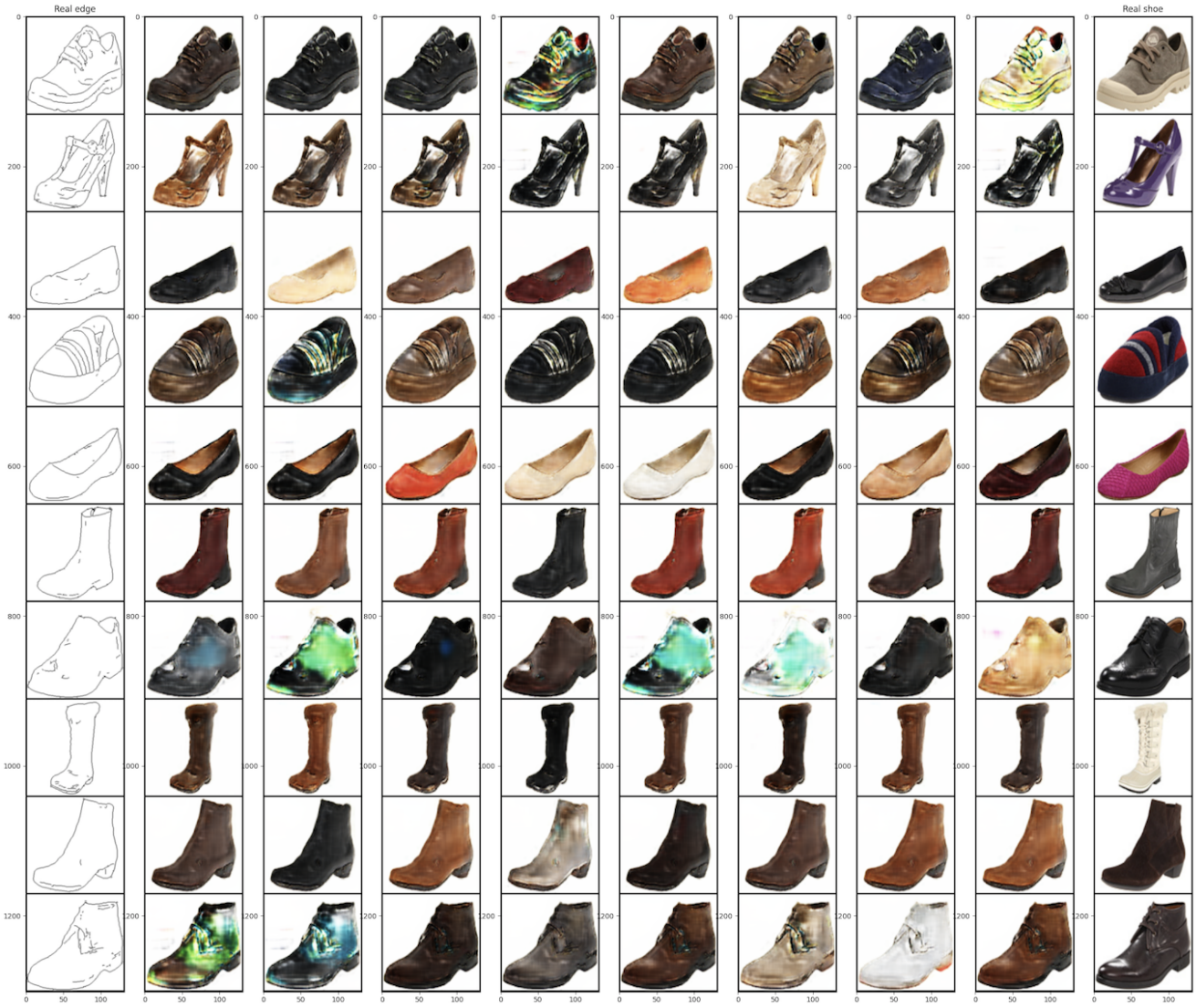

Figure 12. BicycleGAN epoch=40 visualization (Validation set)

Figure 13. BicycleGAN epoch=50 visualization (Validation set)