CIS 680 Fall 2021, Project Report

Extension of GAN and VAE to model Multimodal Distribution

Team Members:

- Shiyani Patel; Email: `shiyani@seas.upenn.edu`
- Karan Wanchoo; Email: `kwanchoo@seas.upenn.edu`
- Varun Lalwani; Email: `varunl@seas.upenn.edu`

1 Abstract

Data for training neural network is most important, as it is the data that provides different scenarios for the network to learn against. However, it is common to lack sufficient data and so networks like Generative Adversarial Networks, GANs, are used. As per the methodology for GANs, a discriminative model D calculates the chance that a sample came from the training data rather than G, and a generative model G that reflects the data distribution [5].We want to introduce a more advanced generative model named BicycleGAN, an extension of CycleGAN which simulate a distribution of possible outcomes for a conditional generative model[15]. This model links GAN and VAE, and explicitly learns a latent distribution that encodes the uncertainty information in image-to-image translation. The objective of our network is to produce diverse set of image outputs along with the images being plausible given the conditional image input. You will find the link to our video here: **Link for our Demo Video**

2 Introduction

For our final project, we propose to implement and evaluate the performance the BicycleGAN over the edge2shoes data set, briefly discussed in section 3.2. In many image conditional synthesis tasks, we are expanding the input image signals. For example, when we translate an image for diverse output, we need to fill-in the missing colors and textures, which could have multiple plausible hypotheses. Unfortunately, the generative model, CycleGAN is useful for modelling one-to-one mapping. That's why we propose to implement BicycleGAN, which is an extention to CycleGAN. BicycleGAN is a more advanced generator that can perform one-to-many modelling tasks or multimodal image to image translation. This network aims to produce not only plausible, but also diverse set if generate outputs for a given conditional image input.

Our network has consists of two methods the Conditional Variational Autoencoder GAN, cVAE-GAN, and the Conditional Latent Regressor GAN, cLR-GAN. The cVAE-GAN are a variational generative adversarial networks. It is a general learning framework that combines a variational auto-encoder and a generative adversarial network for synthesizing images in fine-grained categories, for a wide variety of categories like faces, flowers, shoes, and others [1]. It uses a probabilistic model to model a picture as a composite of label and latent attributes [1]. This model is able to generate images in a certain category with randomly generated values on a latent attribute vector by altering the fine-grained category label fed into the generative model [1].

The other model cLR-GAN gives the generator a randomly drawn latent vector first. The output is different from the ground truth image but it is realistic [15]. The latent vector is then attempted to be recovered from the output image using an encoder. This technique is similar to InfoGAN [2] and can be considered as a conditional version of the ”latent regressor” model [3], [4]. We integrate both of these approaches to increase performance by enforcing the relationship between latent encoding and output in both directions.

Our objective is to create a model that is able to produce a distribution of potential output, that are diverse and realistic based on a single image.

3 Related Work

3.1 Generative Adversarial Networks

Generative Adversarial Networks(GAN), are generally used to train generative models [9] as to generate data. The extension of GAN and Variational Auto Encoder(VAE) for modeling mutimodal distribution requires an understanding of these two base concepts. The GANs are designed to assimilate two parallel pipelines, the generative model and the discriminative model. The generative model captures the data distribution and the discriminative model addressed the estimation of sample data being generated or true using probability. These generative and distibutive models are created via multilayer perceptrons and trained with backpropagation [5]. As for the VAEs it is an independently parameterized models, the encoder and the decoder. The encoder focuses on getting approximation for the posterior over latent random variables whereas the decoder learns the meaningful part of the data such as the labels. According to Bayes rule the encoder and decoder are inverse of each other [8]

3.2 BicycleGAN

The BicycleGAN formally the multimodal image to image translation follows the general pipeline of mapping by objective consistency between output and the latent code making the mode collapse problem irrelevant [15]. The training of such models requires a large dataset and we leverage the STL-10 dataset for our study. BicycleGAN consists of two components; Conditional Variational AutoEncoder GAN(cVAE-GAN) and Conditional Latent Regressor GAN (cLR-GAN) [15]. cVAE-GAN, encodes the ground truth image into the latent space using an encoder. Then input image and encoded ground truth image are provided into the generator which produces the output image [15]. In cLR-GAN a randomly drawn latent vector along with the input image(A) is provided to the generator. The generated output may or may not look like ground truth image. The generated output is passed through the encoder, to regain the latent vector from the output image [15]. The models for conditional image generation so far have covered in painting, adding color and filling in colors for sketches [6][7][10][13][12]. However, these models output a single subject from the described outputs, the BiCycleGANs go a step further and are able to output distribution of potential results based on single input, which offer both diverse and realistic results [10]. The network details include symmetric skip connections between the generator and discriminator with training on 20 epochs [1].

4 Methods

BicycleGAN is an extention of CycleGAN. The model is an extention of GAN and VAE, the implementation details will be covered in 4.1.

4.1 Network Architecture and Loss Functions

Our model of BicycleGAN follows the general pipeline as shown in figures 1a and 1b for both the cVAE-GAN and cLR-GAN. We designed the generator G, that implements the U-Net architecture [11], this U-Net contains the encoder-decoder with symmetric skip connections. The baseline model has a vanilla generator [5]. The discriminator D, is vanilla discriminator for both our model and the baseline model. The input for both the networks are RGB images of shape [3x128x128]. The images have two domains denoted as

$$A \subset R^{H\times W\times 3}\text{and } B \subset R^{H\times W\times 3}$$

The cVAE-GAN begins ground truth target picture, represented in the figure 1 as B and encodes it into the latent space represented in the figure 1 as Q(z B). The generator outputs the a fake image that uses the underline method of transformation of input image A, as shown in figure 2 to a sampled from drawn latent code z back to original image B. The cLR-GAN uses random samples from the latent code from a known

[1] The current proposal is subjected to change based on future implementation liberties

prior distribution of N(0,1). It then maps A into the output $\hat{B}$ in order to reconstruct the latent code from the output.

The loss functions used for cVAE-GAN are the L1 loss shown in equation 4.1.1, Adversarial loss shown in equation 4.1.2 and KL divergent loss shown in equation 4.1.3. The purpose of these 3 losses are to constrain the transformed image G(A,z), Further, to make the generated image so that it is photo-realistic and to enforce the latent space to be a compact Gaussian distribution, respectively.The loss function used for cCLR-GAN is the L1 loss between encoded latent vector of $\hat{B}$ and prior distribution p(z) as shown in equation 4.1.4. These losses are then combined in the form shown in equation 4.1.5. For Discriminator we have used the standard BCE loss.

$$L_{1(G)}^{image} = E_{A,B \sim p(A,B),^{z} \sim \varepsilon(B)} ||B - G(A,z)||_1$$

The L1 loss between the output of the generator and the ground truth image (4.1.1)

$$L_{GAN}^{VAE} = E_{A,B \sim p(A,B),[log(D(A,B))]} + E_{A,B \sim p(A,B)^{,z \sim E(B)}} [log(1 - D(A, G(A,z)))]$$

The adversarial loss for VAE [8] with Encoder E, Discriminator D and Generator G [14] [15] (4.1.2)

$$L_{KL} = E_{B \sim p_{(B)}} [D_{KL}(E(B)||N(0,1))]$$

The L1 stands for L1 loss, D stands for discriminator loss [14] [15] (4.1.3)

$$L_1^{latent}(G,E) = E_{A \sim P_{(A)}, z \sim p(z)} ||z - E(G(A,z))||_1$$

The L1 Loss for cLR-GAN with Encoder E, and Generator G [14] (4.1.4)

$$G^{\star}E^{\star} = argmin_{G,E} max_D L_{GAN}^{VAE} + \Lambda L_1^{image}(G) + L_{GAN} + \Lambda_{latent} L_1^{latent} + \Lambda_{KL} L_{KL}$$

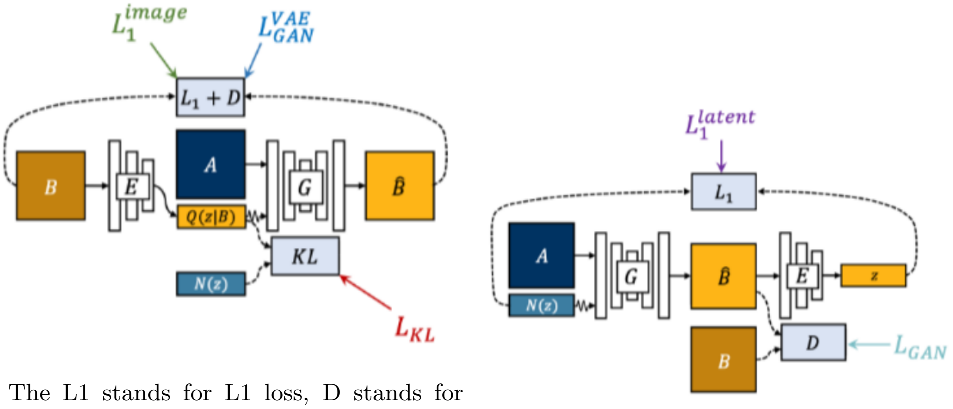
The collective loss for Generator $G^{\star}$ and Encoder $E^{\star}$ [14] (4.1.5)

5 Experiments and Results

This section will cover the quantitative in 5.1 and qualitative in 5.2 results for our model. The hyperparameters used are defined in table 1. We have designed our model to have a generator based on U-Net architecture with symmetric skip connection while keeping the baseline model's generator vanilla.

Network	Encoder	Generator	Discriminator cVAE	Discriminator cCLR
Learning rate	5e−6	8e−4	4e−7	4e−7

Table 1: The table shows the learning rates used for training different components of our model



(a) The L1 stands for L1 loss, D stands for discriminator loss, together making the loss for VAE GAN and KL stands for KL divergence for cVAE-GAN [14]

(b) The L1 stands for L1 loss, D stands for discriminator loss for cLR-GAN [14]

Figure 1: Brief overview of BicycleGAN [14], where figure 1a is overview of cVAE-GAN and figure 1b is an overview of cLR-GAN

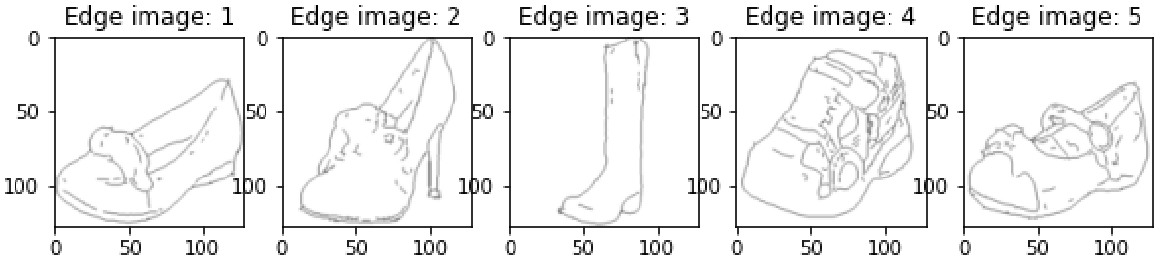


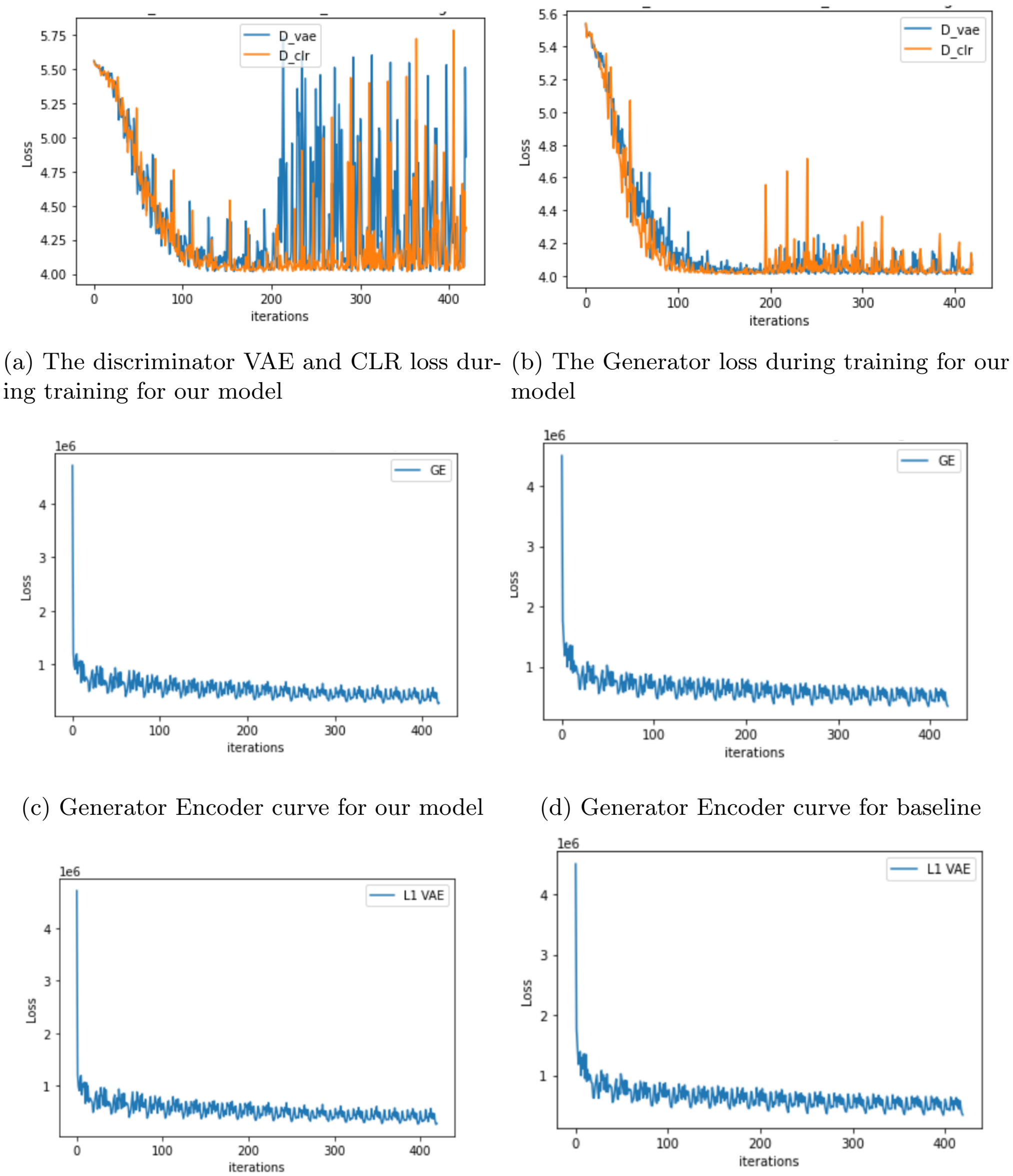
Figure 2: The RGB image input for the Network

5.1 Quantitative Comparisons

We show the difference in the loss curve for our model with U-Net generator and the Baseline with vanilla Generator.The figure 3a and 3b shows the difference between discriminator loss for cVAE-GAN and cLR-GAN on both our model and the baseline. The loss starts to increase for our model as our outputs generated are increasingly realistic based on increasing iterations compared to the baseline. The figure 3c and 3d shows the generator loss for our model and the baseline. The difference in the curves for L1 Clr-GAN loss is decreasing overtime for our model, figure 4a, while is almost steady for baseline, figure 4b as it is approach to capture image mode in latent space and enforce consistency. Another drastic difference is in the KL Loss for cVAE-GAN for our model, figure 4c, and baseline 4d as the cVAE-GAN is unable to make hard decision boundaries. Finally as expected the Generator losses for our model figure 4e and figure 4g are getting more inconsistent with increasing iterations while for baseline, figure 4f and 4h is comparatively steady. The reason for this inconsistency is due to the fact that our model is producing more realistic outputs. The Frechet Inception Distance (FID) scores for the validation set with validation set is $-4.876e{-5}$ and the between the validation set and the generated images for our model is **117.10**. The averaged pairwise Learned Perceptual Image Patch Similarity (LPIPS) score for our model on the validation dataset is **0.1627**.

5.2 Qualitative Results

The difference in the image generation for our model with U-Net on epoch 0, epoch 10 and epoch 20 are shown with image generation for baseline in figure 5. At epoch 0 our model is already able to pick up on



(a) The discriminator VAE and CLR loss during training for our model

(b) The Generator loss during training for our model

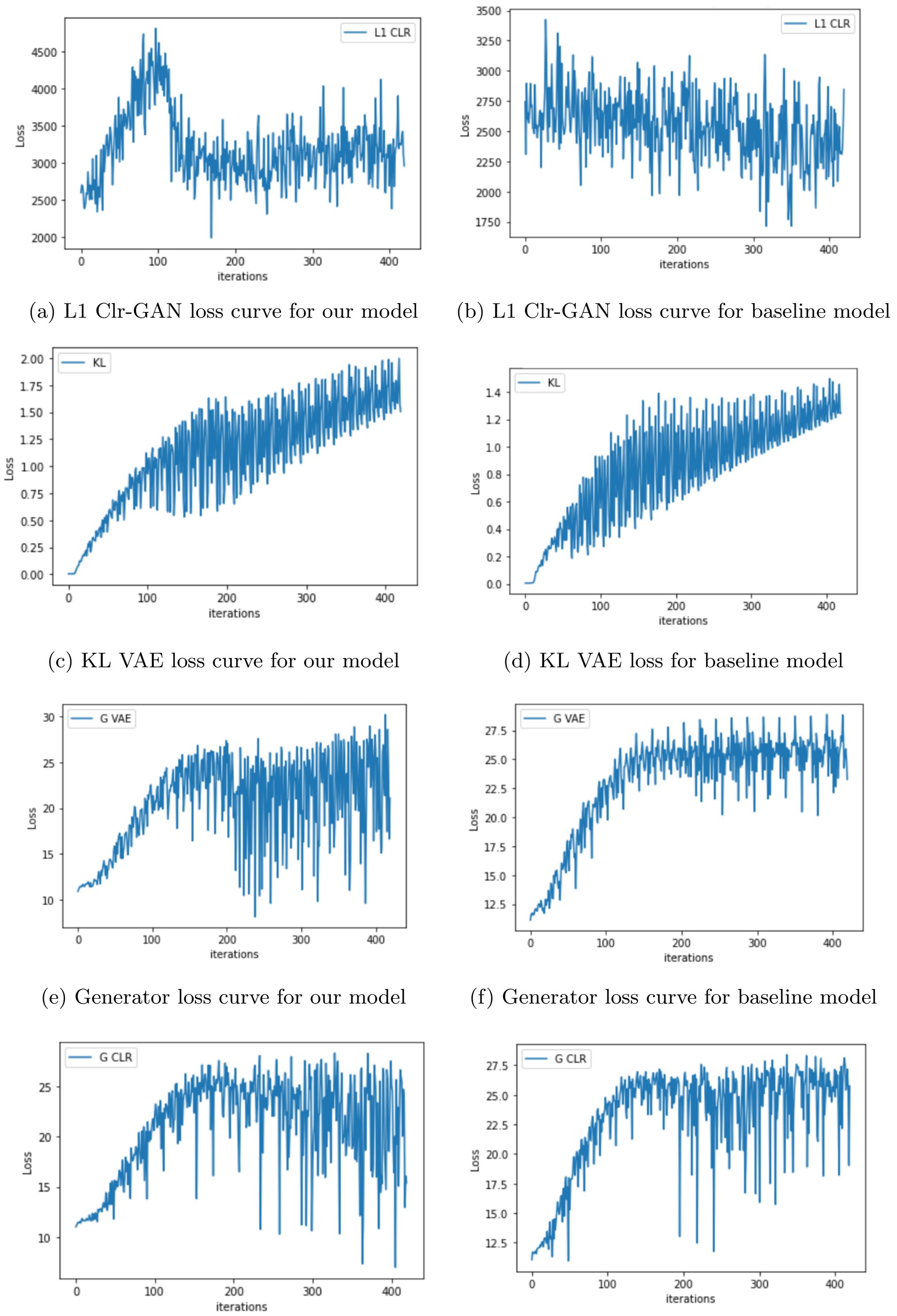
(c) Generator Encoder curve for our model

(d) Generator Encoder curve for baseline

(e) L1 VAE loss for our model

(f) L1 VAE loss for baseline

Figure 3: Various Loss curves for our model with U-Net generator and baseline model



(a) L1 Clr-GAN loss curve for our model

(b) L1 Clr-GAN loss curve for baseline model

(c) KL VAE loss curve for our model

(d) KL VAE loss for baseline model

(e) Generator loss curve for our model

(f) Generator loss curve for baseline model

(g) Generator loss curve for Clr-GAN for our model

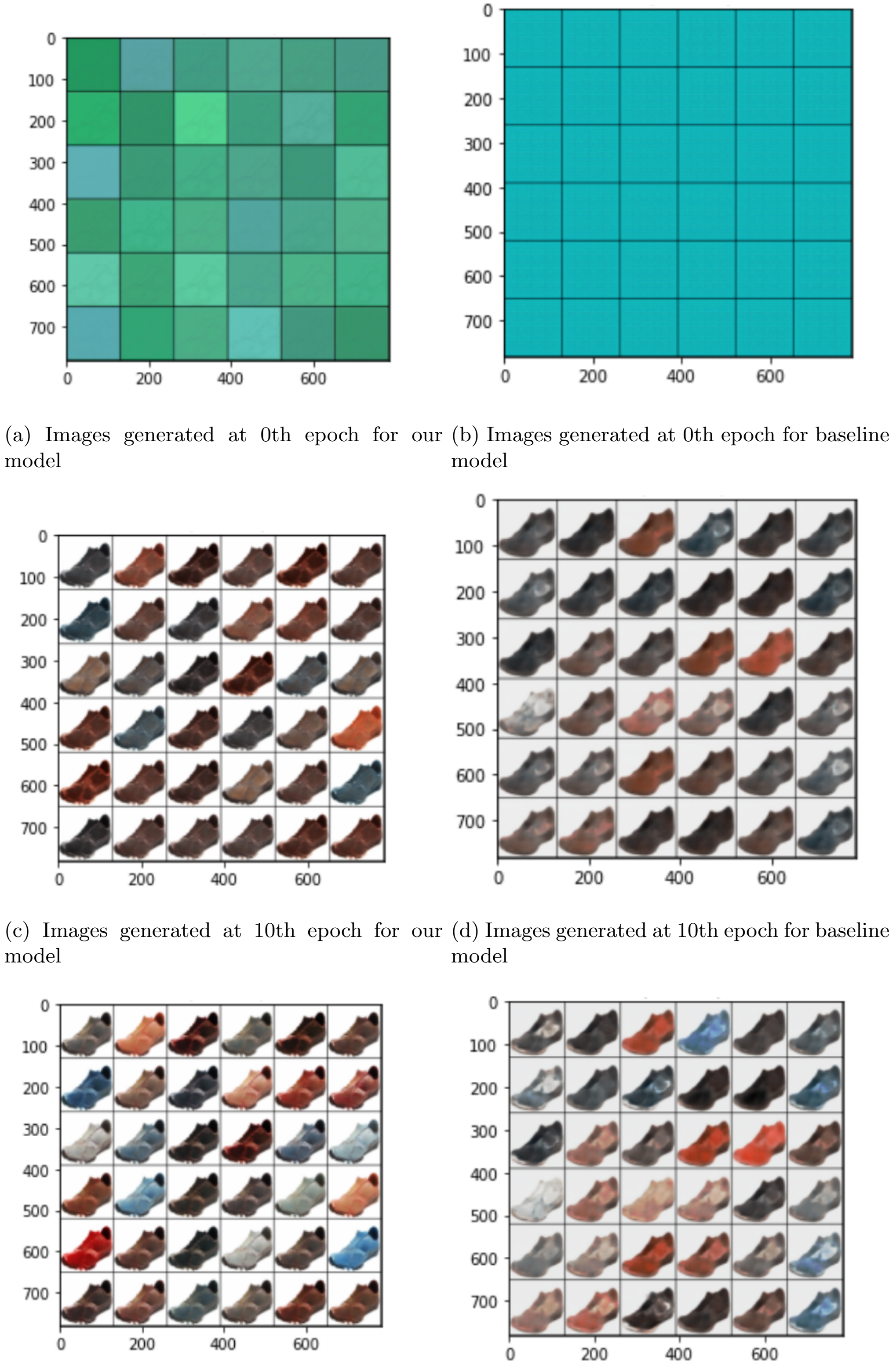
(h) Generator loss curve for Clr-GAN for baseline model

Figure 4: Loss curves for our model and the baseline

the underlying features as shown in figure 5a, while the baseline model is not, shown in figure 5b. The 20th epoch shows the drastic difference in realistic image output for our model, shown in figure 5e, while the baseline model has grainy output as shown in figure 5f. The figure 6 shows the 10 set of outputs rendered by our model that shows how realistic our generated images look.

6 Conclusion

We are able to show that we have designed a model called the BicycleGAN with U-Net generator is able to produce a distribution of potential output that are diverse and realistic based on single image. Our model has substantially performed better than the base model and so we can build upon this network to create a more robust network for image generation. For future we want to add the patch GAN for our discriminator while keeping the architecture for our model's generator to be U-Net, as well as test on a different dataset. Finally to evaluate our work we will do quantitative checks with Learned Perceptual Image Patch Similarity (LPIPS) score and the FID scores.



(a) Images generated at 0th epoch for our model

(b) Images generated at 0th epoch for baseline model

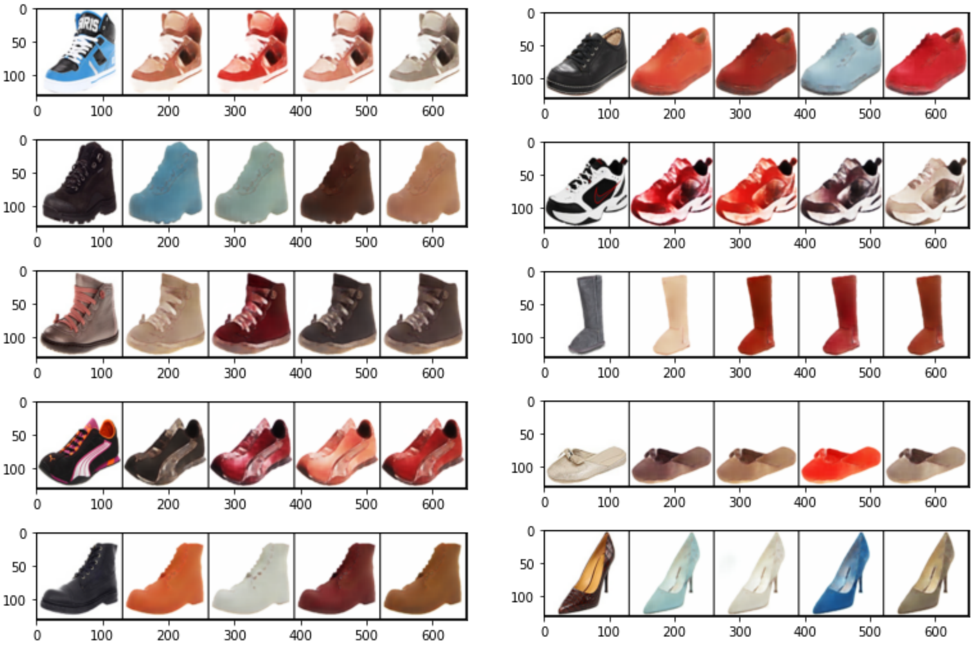
(c) Images generated at 10th epoch for our model

(d) Images generated at 10th epoch for baseline model

(e) Images generated at 20th epoch for our model

(f) Images generated at 20th epoch for baseline model

Figure 5: The images generator by our model(left) and the images generated by the baseline(right)



(a) Images generated by our model

(b) Images generated by our model

Figure 6: The images generated by our model, where the first column for both sets of images 6a and 6b are the input images and the rest of the columns are genertaed images

References

[1] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: Fine-grained image generation through asymmetric training. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

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

[3] Jeff Donahue, Philipp KrÃ€henbÃŒhl, and Trevor Darrell. Adversarial feature learning, 2017.

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

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[6] 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 Transactions on Graphics (Proc. of SIGGRAPH 2016)*, 35(4):110, 2016.

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

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

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

[10] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting, 2016.

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

[12] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis, 2017.

[13] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization, 2016.

[14] Lukas Zhornyak, Wen Jiang, and Jianbo Shi. Final project 2021, 2021.

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