Learning to Synthesize Realistic Fine-grained Images by Object Parts

Zixuan Huang
Department of Computer Science
University of Wisconsin-Madison
NetID: zhuang356
zhuang356@wisc.edu

Yufei Wang
Department of Computer Science
University of Wisconsin-Madison
NetID: ywang2395
ywang2395@wisc.edu

Yang Fang
Department of Electrical & Computer Engineering
University of Wisconsin-Madison
NetID: fang65
fang65@wisc.edu

Briefly explain what problem you are trying to solve.

Learning controllable and interpretable generative model has been one of the main pursuit of image synthesis, yet the controllability and interpretability can happen at multiple faucets. Consider the examples in Fig. 1. If we want to generate these images, what is a reasonable way to generate them in a controllable fashion? We believe at least one possible solution is to synthesize these images by object parts (e.g., the head, neck, belly, legs, wings of the birds and the background). Such an approach not only provides an explicit way to synthesize fine-grained details according to each specific object part, but also brings a controllability over the synthesis procedure via independently manipulating different parts of generated images.

On the other hand, Generative Adversarial Network (GAN) [1] are tremendously successful for image generation. Current state-of-the-art methods along this direction are already able to generate photo-realistic images. By combining our idea with GANs, we hope the resulting method is able to generate both photo-realistic and controllable image by object parts.

Figure 1: Four randomly sampled images from CUB-200 [2]. Assume they are synthesized images, what is a reasonable way to generate them in a controllable fashion?

How will you evaluate the performance of your solution? What results and comparisons are you eventually planning to show? Include a time-line that you would like to follow.

We will first evaluate the performance of W-GAN quantitatively and qualitatively on CUB-200 datasets as our baseline. After implementing our own approach, we will compare it with the baseline. We plan to conduct the qualitative evaluation by showing sample outputs, and demonstrate quantitative results through commonly used Inception Score [3] or Frechet Inception Distance [4].

We will roughly follow the time-line below:

2.15 - 3.1: Finish an overall literature review and design concrete approaches to improve the baseline.
3.1 - 3.15: Implement the baseline on CUB-200 and evaluate the baseline.
3.15 - 4.15: Implement our approach and compare with the baseline. If the results are great, also compare with other state-of-the-art methods.
4.15 - 4.29: Wrap up everything and prepare for the final presentation.

What did we do in the past two months?

We roughly followed our timeline:

- First, we finished an overall literature review. We found that W-GAN [5] could be regarded as a baseline. Rather than using the CUB-200 dataset, we decided to use Celeba [6] instead, as it is larger and more commonly used by other GAN models.
- We designed our concrete approaches to achieve our goal. Our basic idea is from Bi-GAN [7]. In a general GAN, there is one generator and one discriminator: the generator is to generate images from the latent code, while the discriminator is to distinguish between the real images (images from the training datasets) and the images generated by the generator. In Bi-GAN, we have one more encoder, which is used to convert the generated image back to the latent code. To make this encoder sensitive to different parts of the object in the image, a new structure as well as a regularization term can be used. In this way, the generated latent code will be meaningful and could be used to generate new images through the generator part by part.
- We implemented the baseline, which is a Bi-GAN with the loss using Wasserstein distance. We also evaluate this baseline qualitatively on Celeba and the results are attached in the next section.
- We began to implement our new method and decided to rewrite the encoder by combining the graph projection unit proposed in [8]. The projection unit extracts meaningful parts from the images via clustering the CNN feature maps. We hope our encoder can also learn such part representation for latent code and thus the generator can synthesis the fine-grained images by object parts. We are still working on this and will get more results in the next few weeks.

What results did we get?

We implemented our baseline, which is the Bi-GAN with Wasserstein distance on the Celeba dataset. The results below illustrates our training results.

Figure 2: 64×64 generation results

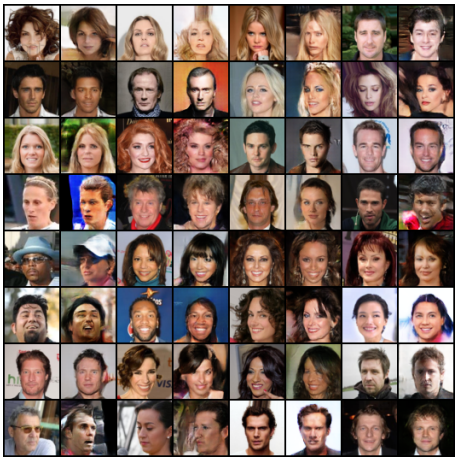

Figure 3: 64×64 reconstruction results

The left figure shows generation results, where all images are directly generated from the randomly sampled latent code. The right figure demonstrates reconstruction results by pairs, where the columns with the odd indexes (i.e. the first, third, fifth, and the seventh columns) are the images in the training set, and the columns with the even indexes are reconstructed image. This reconstruction process first

converts the images to the latent code by encoder, and then reconstruct images with the encoded code via generator.

The generated and reconstructed images looks realistic. And the reconstruction results demonstrates that the encoder of Bi-GAN indeed project image to its corresponding code. Then if our methods can enable the encoder to learn part-based representation, we will be able to synthesis the images by object parts.

What problems did we encounter?

The first problem we encountered is the size of the training images. During training, all images in Celeba dataset are resized to 64×64, but this scale might be too small for finding object parts in the images. For instance, for human faces in the Celeba dataset, if we rescale all images to 64×64, small object parts like eyes may only contain few pixels. This makes the learning of part representation hard as some significant parts are missing due to the small resolution. We tried to enlarge the size of the training images to 128×128, however, the results seem to be worse as illustrated below.

Figure 4: 128×128 generation results

Figure 5: 128×128 reconstruction results

Another problem we encountered is that our new encoder does not fit quite well with the original Bi-GAN structure. Right now our code is runnable but the reconstruction process fails (see Fig. 7). We are still trying to solve this problem.

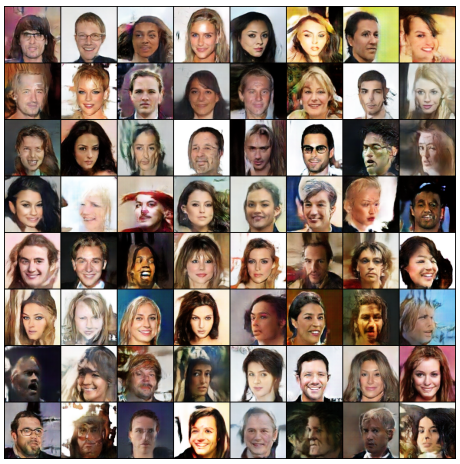

Figure 6: 128×128 generation results with our new method

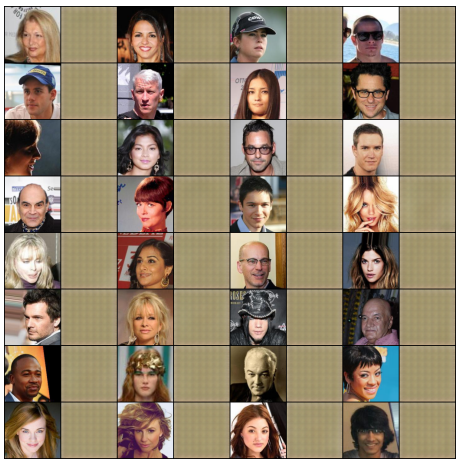

Figure 7: 128×128 reconstruction results with our new method

How will we evaluate the performance of our approach? What results and comparisons are you eventually planning to show?

We roughly stick to our original evaluation in the proposal. Specifically, We will first evaluate the performance of Wasserstein Bi-GAN on CelebA dataset. After getting our own approach work (hopefully), we will compare it with the Wasserstein Bi-GAN baseline. We will mainly conduct the qualitative evaluation by showing sample outputs, and may also demonstrate quantitative results through commonly used Inception Score [3] or Frechet Inception Distance [4].

What are we planning to do?

Right now we are still on schedule. We will try to solve the problems listed above and finish implementing our new methods by 4.15, and then start to write our final report and design the final project website. See the timeline section for more details.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[2] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[3] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[6] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15:2018, 2018.

[7] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[8] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. In *Advances in Neural Information Processing Systems*, pages 9225–9235, 2018.