CS230

Celebrity and You Synthesis Generator

Zhu Chen
Facebook Inc.
zhc047@stanford.edu

Abstract

As the development of image morphing, computer generated images can be synthesized smoothly, looking like a real image. Naturally, it leads us to explore real photo synthesis. We embed real photos into extended StyleGAN's latent space and interpolate two latent space to generate the synthesized image. We achieved to synthesis any two real human face images.

1 Introduction

Generative Adversarial Networks (GANs) [1] are successfully applied in generating realistically-looking images. Among recent advances in GAN architectures, StyleGAN [2] introduces an intermediate latent variable w, which is meant to be the "styles" of generated images, as input to the generator instead of a traditional random variable z, which achieved impressive results. This naturally leads to the question: can we encode any given image into this latent space W so that the generator could synthesize a image similar to the original input? Furthermore, we want to know if the learned latent variables are robust and meaningful enough so that two images can be synthesized smoothly by interpolating their latent variables, known as the image morphing task in computer vision.

The project is about generating pictures of real human faces synthesized with celebrities' faces, with user adjustable degree of synthesis. The inputs to our neural net are two photos, the first one is a real human face photo, the other from a celebrity. We encode input images into latent vectors in an optimization setting. Finally, we interpolate these two latent vectors and use it to generate the synthesized image. The project is interesting because many people use smart photoshop applications to redesign their pictures. This project can be an interesting feature to add on to those applications.

Our main contributions include:

- Independently implemented the model of embedding images to StyleGAN described in a very recent work by NVIDIA [3], which came out at about the same time we submitted the project proposal.
- Explored different loss functions to improve the sythesis quality and proposed to use only perceptual loss while embedding images to StyleGAN.
- Explored multiple ways of interpolating two latent space, providing insight into the structure of the extended StyleGAN's latent space.

2 Related work

Since Ian Goodfellow et al. proposed Generative Adversarial Networks [1], it has been improved a lot to generate high-quality, realistic images. Karras et al. collected the first 1024×1024 resolution human face dataset and proposed a methodology to train GANs to generate high resolution images [4]. Later, Karras et al. collected another human face dataset FFHQ, which is more diverse, and proposed

a new generator architecture inspired by style transfer literature, which makes it possible to separate high-level attributes on human faces [2]. However, the effort so far is mostly on improving the quality of generated image and operate on generated images, in this paper, we would like to talk about face morphing on real face photos. Around the same time when we had the proposal, Rameen Abdal et al. proposed to embed real images to StyleGAN latent space [3].

While pixel-wise L1 or L2 norm loss is widely used to compare the low-level similarity between two images, it could not capture the position- and scale-invariant features of images. Gatys et al. found that the activation layers of convolutional neural networks extracts the features of images so that can be used to measure the high-level similarity of images [5]. Later, Justin Johnson et al. formalized it as the perceptual loss [6]. Also, to enable style transfer in real time, Huang and Belongie proposed adaptive instance normalization (AdaIN) that can transfer arbitrary style in real-time[7]. In this paper, we explore generating images with both pixel-wise loss and perceptual loss as well as only using perceptual loss to achieve the best quality of reproducing a real image.

3 Dataset and Features

The datasets we use for this project are two existing ones:

CelebFaces Attributes Dataset HQ (CelebA-HQ) [8]: The dataset consists of more than 202,599 celebrity images of 10,177 identities at 1024×1024 resolution. It includes face images taken from different angles and in different backgrounds. Each image is annotated with 40 attributes.

Flickr-Faces-HQ (FFHQ) [2]: This dataset consists of 70,000 human face images each at 1024×1024 resolution. The faces are varied in terms of age, ethnicity, gender, image background, angle when the image was taken, etc. It also covers accessories like eyeglasses, sunglasses, hats, etc.

We randomly choose one image from CelebA and FFHQ, encode them into vectors and perform a synthesis. So, technically, there is no train/dev/test set since we use a pre-trained StyleGAN network and perform a per-image task.

4 Methods

Our overall strategy is: first embed two real images into StyleGAN latent space; then interpolate those two latent vectors to generate a synthesized picture.

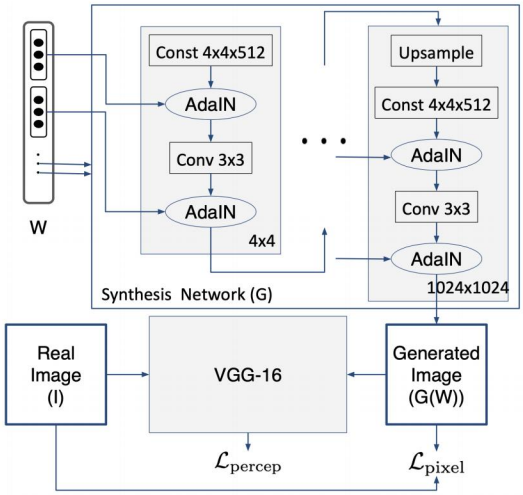

Figure 1: Image Encoding Model. W is first initialized from the training average latent vector and then input into different layers of the StyleGAN's synthesis generator (G) to output the generated image $G(W)$; the loss is calculated between the real image I and $G(W)$ in terms of pixel-wise loss and perceptual loss, which is defined as activation difference in a VGG-16 network. We optimize W to minimize the loss defined in Equation 1.

Figure 1 illustrates the overall model we use to encode images. The idea being, for any given image I, we train a latent vector W to minimize the loss

$$\mathcal{L} = \mathcal{L}_{\text{percep}}(G(W), I) + \lambda \mathcal{L}_{\text{pixel}}(G(W), I) = \sum_i \|A_i(G(W)) - A_i(I)\|_2 + \lambda \|G(W), I\|_2 \quad (1)$$

where $W \in \mathbb{R}^{\text{num of style} \times \text{latent size}}$; G is the sythesis generator of StyleGAN and $G(W)$ is the generated image; λ is the hyperparameter weighing pixel-wise loss; A_i is the i-th layer's activation of a VGG-16 net [9], and we choose 4 layers: *conv1_1*, *conv1_2*, *conv3_2* and *conv4_2*, same as [3].

For face morphing, we explore three different ways to interpolate two latent vectors W_p and W_c encoded through our above model from a person's and celebrity's image respectively:

1. Linear Interpolation
$$W_s = \lambda_{\text{interp}} W_p + (1 - \lambda_{\text{interp}}) W_c$$
where λ_{interp} is the user adjustable degree of synthesis.
2. Style Mixing
$$W_s[: \text{style\_xp}, :] = \lambda_{\text{interp}} W_p[: \text{style\_xp}, :] + (1 - \lambda_{\text{interp}}) W_c[: \text{style\_xp}, :]$$
$$W_s[\text{style\_xp} :, :] = (1 - \lambda_{\text{interp}}) W_p[\text{style\_xp} :, :] + \lambda_{\text{interp}} W_c[\text{style\_xp} :, :]$$
where style_xp is the crossover point between $[0, \text{num of styles})$.
3. Attribute Mixing
$$W_s[:, : \text{latent\_xp}] = \lambda_{\text{interp}} W_p[:, : \text{latent\_xp}] + (1 - \lambda_{\text{interp}}) W_c[:, : \text{latent\_xp}]$$
$$W_s[:, \text{latent\_xp} :] = (1 - \lambda_{\text{interp}}) W_p[:, \text{latent\_xp} :] + \lambda_{\text{interp}} W_c[:, \text{latent\_xp} :]$$
where latent_xp is the crossover point between $[0, \text{latent size})$.

The interpolated vector is finally input into the synthesis network (G) to generate the synthesis image.

5 Experiments

We conducted extensive experiments to find best model and hyperparameters; we also report our quantitative metrics and qualitative results including an user survey. Code is available on Github[1].

5.1 Loss Function

(a) $\lambda = 0$ (b) $\lambda = 0.1$ (c) $\lambda = 1$ (d) $\lambda = 10$

Figure 2: Recovered Images Using Different λ Weighing Pixel-wise L2 Loss

To generate images as similar as possible to real input images, we first tweaked the loss function. We tried out different values of λ which weights the pixel wise loss in the loss function. As shown in Figure 2, when pixel loss weighs too much, we give the neural net too hard a question to solve for regenerating an image, and that restricts its freedom to optimize. However, by only using perceptual loss, the problem becomes easier to solve, since the net now only needs to produce an image that looks perceptually similar to the real image. Therefore, we decided to use $\lambda = 0$ since it generates the image that is the best in terms of visual quality.

(a) Num of Styles $= 1$ (b) Num of Styles $= 3$ (c) Num of Styles $= 9$ (d) Num of Styles $= 18$

Figure 3: Recovered Images Using Different Numbers of Styles

5.2 Number of Styles

We would like to understand how different number of styles affects the generated images' quality. Figure 3 shows that with 18 styles, we can capture more details than with fewer styles. A deeper layer in the synthesis network captures finer features, so by having one style per layer, we can use different styles to capture different levels of features and thus recovering the input images more realistically.

To quantitatively measure the synthesis quality, we define the perceptual path length (PPL) as follows:

$$\text{PPL} = \mathbb{E}\left[\frac{1}{\epsilon^2} d(G(\text{lerp}(W_p, W_c; \lambda_{\text{interp}})), G(\text{lerp}(W_p, W_c; \lambda_{\text{interp}} + \epsilon)))\right]$$

where lerp is the interpolation function; $\epsilon = 10^{-4}$ is a small tweak to interpolation parameter; d is the perceptual length measured by L2 norm distance between VGG-16 activation. Intuitively, this metric shows the robustness of generating images from recovered latent vectors and the smaller PPL, the better. Table 1 shows the quantitative metrics of different number of styles. As we can see, 18 styles gives the best results.

Table 1: Loss and PPL Metrics of Different Number of Styles

Num of Styles	1	3	9	18
Loss	10504	8835	6737	**6037**
PPL ($\times 10^7$)	1.0997	1.2537	0.9501	**0.8592**

5.3 Synthesis Degree

We would like to adjust the degree of synthesis. So, we tuned λ_{interp} which controls the weights on using the features from W_p. As shown in Figure 4, when λ_{interp} is small, the synthesized photo looks more like the photo generated by W_c. Since we would like to get a synthesized photo that look more like the user than the celebrity, we decided to use $\lambda_{\text{interp}} = 0.7$.

5.4 Interpolation Methods

We tried different face morphing methods defined in Section 4. Figure 5 shows the output of different morphing methods. Linear interpolating means all features from one human have the same weights, so we see the images synthesized nicely with one person's features dominating. Mixing styles means one person's coarse features dominates and another person's fine features dominates because the deeper the styles in the neural network, the finer details they capture. From figure 5, we found that style mixing also produces reasonably good quality of synthesized images. However, attribute mixing works poorly. We think it is because features might be controlled by multiple attributes, so randomly choosing a cross point might put attributes that control the same feature into different weights, for examples, attributes A and B together controls eyelids, if A of image1 dominates and B of image2 dominates, then the synthesized image might be distorted on that feature.

5.5 User Survey on Synthesis Quality

We sent out a survey to 15 users. 80% users can identify which two images are used to synthesize the given image from 4 options. 46.7% users think that linear interpolating produces the best quality of

[1]Link to Github repo: `https://github.com/zhc047/cs230`

(a) $\lambda_{\text{interp}} = 0$ (b) $\lambda_{\text{interp}} = 0.3$ (c) $\lambda_{\text{interp}} = 0.7$ (d) $\lambda_{\text{interp}} = 1$

Figure 4: Synthesis Images Using Different λ_{interp} Controlling Synthesis Degree

(a) Style XP $= 4$ (b) Latent XP $= 128$ (c) Style XP $= 9$ (d) Latent XP $= 256$

Figure 5: Recovered Images Using Different Interpolation Methods

synthesized image, and 33.3% think that style mixing where crossover point is at half of total number of styles synthesized the best. 53.3% users think that when we don't use pixel wise loss, i.e. $\lambda = 0$, it does the best job of reproducing the original image.

6 Conclusion and Future Work

We embedded two real human face photos into StyleGAN's latent vectors, and interpolated them to generate a synthesized image. Perceptual loss alone reproduces images better than with pixel wise loss due to more freedom to optimize. Synthesis on styles has smoother results than on latent space because choosing cross point on latent space might separate two entities which control one feature.

We currently interpolate two latent vectors W's by using a scale preserving linear combination. However, we don't have any constraint on the distribution of W_p and W_c, so simply adding them up by adjusting some percentage of each does not seem to make sense. The next step would be considering the distribution of W when calculating loss. Another idea is to further analyze latent space, and see which features can be manipulated in group, so that we can synthesize part of the features but not all. For example, always use the background, skin color of first image, and synthesize on other attributes like face shape, eyes, mouth, etc. Besides, we can also employ some task specific techniques such as facial landmark, geometry and illumination [10, 11, 12] to further improve the system. Those techniques would allow us to better synthesize photos taken at different angles, under different lightning, and be able to detect faces from a picture. We can also try out with expression/gender transfer, etc.

7 Contributions

This is a solo project, so Zhu contributed to all the work above.

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] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

[3] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? *arXiv preprint arXiv:1904.03189*, 2019.

[4] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[5] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems*, pages 262–270, 2015.

[6] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[7] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1274–1283, 2017.

[11] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. Fsnet: An identity-aware generative model for image-based face swapping. *arXiv preprint arXiv:1811.12666*, 2018.

[12] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 168–184, 2018.