StyleGAN and StyleGAN2 for Face Morphing Applications

Jason Kuo

March 2021

Abstract

1 Introduction

Face recognition is an important feature in many systems today, including unlocking personal devices and verifying identity. One particular application of note is an eGate system which is used to automatically permit authenticated users into a restricted area such as for boarding a plane. Face recognition, along with other biometric verification systems, are used by comparing the known data about an individual (such as their face on a passport image) to new data at the gate (for example, by taking a live picture with a webcam).

Face morphing poses a potential security threat to systems like these because if a morph is successfully submitted as a passport photo, it could authenticate multiple users with the same image. There are several methods for generating these morphs. Traditionally, landmark-based morphing methods have been used to detect important facial landmarks and interpolate between the corresponding landmarks for each face contributing to the morph (usually only 2). The best versions of these methods require manual placement of landmarks, which is extraordinarily slow and labor intensive. This process can be automated, but many of the current fully automatic landmark-based algorithms produce easily seen morphing artifacts.

A more modern approach is to use Generative Adversarial Networks (GANs) to produce morphs. In general, GANs attempt to generate new data from the underlying distribution of input data. They do this by constructing two neural networks, a generator and a discriminator, and setting them against each other in a zero-sum game. The generator's goal is to learn to generate data samples similar to the input data for the purpose of fooling the discriminator, whereas the discriminator's goal is to learn to distinguish which samples come from the actual input and which come from the generator. After training, the generator with its learned model should be able to produce data similar to the input, but not exactly copying the input. For the specific example of face images, GANs usually train two convolutional neural networks. Most GANs produce the output image starting with some latent vector representation, meaning that we can think of a GAN as mapping an input vector to a face

image and training the GAN as defining that mapping. For a given input face, if we can find the representation such that the output after GAN processing is as close to the input face as possible, then we can simply interpolate between two such representations to generate morphs between the two faces that those vectors represent. Interpolation is meaningful in the context of GANs because they attempt to learn a smooth mapping from the latent space of representations to the output images.

2 StyleGAN

StyleGAN [1] is a GAN architecture from NVIDIA which is able to generate high quality and high resolution images. It is inspired by style transfer literature which emphasizes the importance of meaningful interpolation properties. Thus, it is very useful for morphing applications. By default, as well as in our pipeline, StyleGAN is trained on the FFHQ dataset consisting of 70,000 high-quality face images at 1024x1024 resolution collected from Flickr. However, one problem with the original StyleGAN is that it does not provide a means for finding the latent vector that corresponds to a given input image. That is, the inverse mapping that converts from a face image to a latent vector is not learned with NVIDIA's implementation.

A popular implementation that does provide this functionality is the stylegan-encoder repository [2]. It uses a pre-trained VGG16 network to calculate the feature vectors for the given input image and an initial StyleGAN generated image. A loss function is defined as the difference between these two vectors. Then, it minimizes this loss by changing the generated image to obtain the encoded latent representation for the input image.

3 StyleGAN2

More recently, NVIDIA released an improved version of StyleGAN named StyleGAN2 [3]. It builds on the original by focusing on removing normalization artifacts which take the form of blurry blobs in the image. It also improves the image quality and provides built in projection functionality for mapping from an image to its latent representation. Their projector mainly differs from the third-party stylegan-encoder by focusing on finding latent representations that the StyleGAN2 generator could have produced. In constrast, the details of the stylegan-encoder's methodology ends up extending the latent space, which allows encoding arbitrary images that would not otherwise have a latent representation. This crucial difference propagates to our morph results as we will see.

4 Experimental Methods

Our goal is to evaluate to what extent a face recognition algorithm is fooled by GAN-generated morphs. To this end, we set up a pipeline for generating morphs, feeding them into a face recognition algorithm, and analyzing the recognition scores produced.

4.1 Setup

We implemented this pipeline on a CentOS 7.6 machine with 2 Intel Xeon CPUs and 2 Tesla K40m NVIDIA GPUs. We tested using the Face Research Lab London Dataset [4]. This dataset was separated into male and female faces so that morphs would only be generated within gender. Then, using both the StyleGAN-encoder and the StyleGAN 2 projector, 50-50 morphs (equal contribution morphs generated using 0.5 as the interpolation coefficients) were generated between each pair of faces within each gender group.

These morphs were evaluated using the `face_recognition` Python module which is a wrapper around Dlib's face recognition features. This module produces a face distance d which is then inverted with $1 - d$ to produce a similarity score for our evaluation. We compared each morph with the two images from which it is generated to produce the morph distribution. We also generate a genuine distribution by comparing between the neutral and smiling versions of each face and an imposter distribution by comparing between neutral images of different people.

4.2 Results

5 Discussion

6 Related Work

7 Future Work

8 Conclusion

References

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

[2] Dmitry Nikitko. Stylegan - encoder for official tensorflow implementation, 2019.

[3] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020.

[4] Lisa DeBruine and Benedict Jones. Face research lab london set, May 2017.