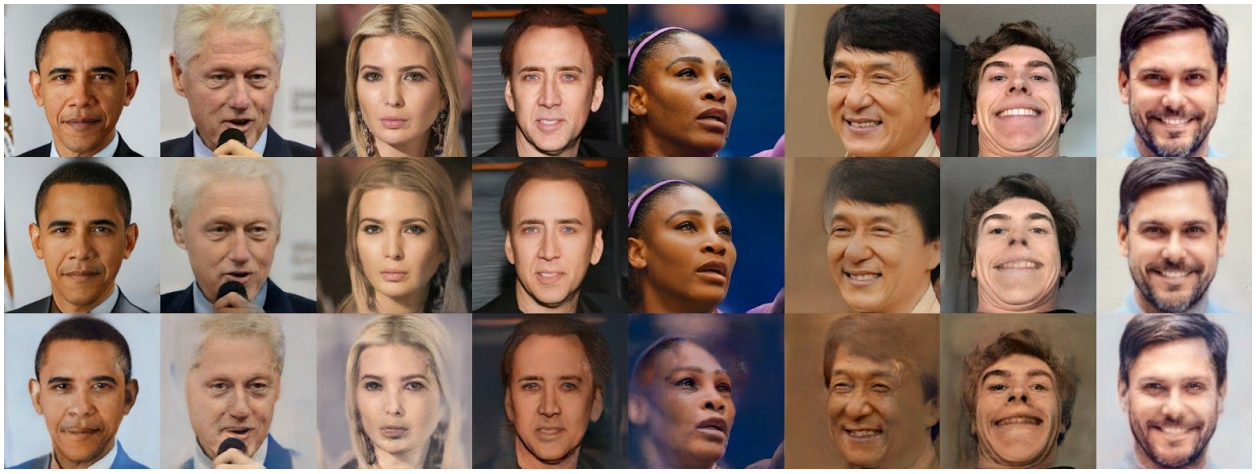
Machine Learning for the Arts
UCSD FALL 2019
FINAL PROJECT

Familiar Faces



Parker Addison
pgaddiso@ucsd.edu

Description:
In this project, real faces are transformed into uncanny, generated-looking faces. Multiple methods of achieving this were attempted, including:

1. Training CycleGAN [7] on high quality images of real faces from the FFHQ repository [6] and images of fake faces generated from ThisPersonDoesNotExist [2].
2. Running Neural Style Transfer [9] using a single real face as the Content Image, and multiple fakes faces from [2] as the Style Images.
3. Using StyleGAN-Encoder [5] to iterate towards a StyleGAN latent vector which generates output that minimizes perceptual loss with a single real face.

The StyleGAN-Encoder approach produced the best results, and this project demonstrates "GAN-ification" on the eight subjects used in the *face2fake* project: Barack Obama, Bill Clinton, Ivanka Trump, Nicolas Cage, Serena Williams, Jackie Chan, Parker Addison, and Robert Twomey.

In addition to images of each individual subject as they are independently reconstructed using the encoder, this project also showcases a video in which the encoder attempted to find a latent representation of a given subject on the condition that the optimization search had to start at the latent representation of the previously-reconstructed subject.

Concept:
Within recent years, advancements in Generative Adversarial Networks (GANs) have allowed people to generate convincing human faces given an input of random (or, as we'll soon discuss, not-so-random) noise. Cutting-edge developments such as StyleGAN [1] have greatly bolstered the believability of these generated faces, prompting the introduction of fake faces in popular culture. Spun from the StyleGAN model, there are entertainment websites which endlessly spew out high-resolution artificial visages [2], businesses which offer tailored selections of faces [3], and internet scams and criminal investigations into the abuse of particularly fool-worthy faces [4]. Despite the incredible convincingness of these faces at a quick glance or at low resolutions, the generated images are still not yet on par with what our brains have come to expect. Instead, the images appear *uncanny*. The mind can tell without much conscious thought that something about the face it's looking at just isn't right. Upon close inspection, artifacts left over by the generating process emerge, such as teeth that are melded together, eyes with mismatched pupils, or skin that is unnaturally folded and stretched. It is precisely this uncanniness that this project attempts to recreate.

All of the examples of fake faces mentioned above share one thing in common—the face being examined is not a real person. As [2] so aptly puts it, *"this person does not exist"*. This attribute prompts many questions.

What if the person *did* exist? How would it feel to see yourself as the output of a fake face generator? How would seeing a real person in the style of a fake person influence what we believe is real or fake in a world of blossoming artificial intelligence?

The goal of this art piece is to deliver on those questions—to explore the emotions and implications that arise with taking a real face and embedding within it the uncanniness of a generated face.

Technique:

To go about "GAN-ifying" input faces, I utilized Dmitry Nikitko's extension of StyleGAN [1] which allows real images to be discovered in the model's latent space [5]. StyleGAN in particular works by representing varying levels of details, or "styles", in the latent space.

Due to the nature of latent spaces being a compressed representation of a target, it is highly unlikely that a latent-vector could produce a perfect lossless reconstruction of a target. Instead, in the areas which the StyleGAN generator does not have information the model will fill in a blend of its similar training data. In this process, artifacts are likely to arise. Is is these artifacts are what give generated faces their distinct uncanniness, and which this project is focused on revealing.[1]

The way that StyleGAN-Encoder [5] works is by examining differences between the target real image and a given latent-vector's decoded reconstruction. Starting at the null latent-vector, the StyleGAN-Encoder can iteratively tweak the latent-vector using Gradient Descent such that perceptual loss between the target and the reconstruction is minimized. As with all Gradient Descent problems, this does not necessarily ensure that a global optimum is found, but in practice is a quick and effective way to reach satisfactory local optima given an appropriate learning rate and number of iterations.

For each of my desired targets I ran the StyleGAN-Encoder up to 1500 iterations from the null latent-vector since by this amount of iterations most of the reconstructions had reached approximate convergence. This process was repeated two additional times without much visible difference in result, so it seems that the same local optima were found by Gradient Descent during each separate attempt. The one exception to this statement is for the reconstruction of my own face. At this point it's worth noting that StyleGAN was trained on the Flickr Faces High-Quality (FFHQ) dataset [6], which contains a large variety of high resolution, closely-cropped photos of faces. Flickr consists of a community of photographers who desire to show off their photography skills. As such, photos uploaded to Flickr containing human subjects are likely to have the subject posing for the camera, or in a candid yet compositionally well-framed position. Since I purposefully chose to take a picture in an uncommon—dare I say unflattering—pose for what one would expect from photographs on Flickr, the StyleGAN-Encoder required further iterations in order to reconstruct my portrait. For the sake of uniformity, however, I still chose to run the reconstruction on my face for the same amount of iterations as the other faces.

---

[1] A quick aside about how some generative artifacts arise: For example, if a latent-vector doesn't quite manage to encode the exact crookedness of a specific target's teeth, or the amount of gap between their teeth, it is likely that the model will attempt to reconstruct teeth that are closer to the average of its training data—teeth which are relatively straight and have little gap. Continuing that example, if the latent-vector doesn't encode tooth crookedness and separation to begin with (if during training the model never associated values in the latent-vector with their respective input's teeth), then the model is likely to reconstruct teeth that are close to average of its training set—teeth which are relatively straight, have little gap, and are directly facing the camera—no matter the target.

I wanted to produce something more meaningful (and a bit flashier) than a still image, so I modified a couple of the files used by the StyleGAN-Encoder in order to produce a video for each target showcasing the evolution of the reconstruction from the null latent-vector to the final optimized latent-vector. The original code only saves a generated reconstruction after the latent-vector is fully optimized, but in order to produce a video I added in the ability to produce a generated photo every 10 epochs. I then ran each target reconstruction to 1440 epochs, such that I could take all of the generated images to create videos that were six seconds in length at 24 frames per second. Each reconstruction resulted in 144 images which I used FFmpeg to convert into a video. These videos have proven intriguing to watch, and they help reveal artifacts which remain from iteration-to-iteration even as the latent-vector is optimized.

Finally, I wanted to create a looping experience, so I modified the code again to allow the search for the optimum latent-vector to start at the latent-vector of the previous face. I assigned an order to the target faces, and proceeded to run each target reconstruction to 1440 epochs twice: 1. starting the first face at the null-latent vector and every subsequent face at the previous face's optimized latent-vector, and 2. starting the first face at the last face's latent vector from the first run. I ran each reconstruction to 1440 epochs, constructed videos using the same process as above, and then concatenated the eight videos to form a single clip. Since the optimized latent-vector of the last face served as the starting latent-vector of the first face, this video is seamlessly loopable.

Process:

I attempted three different approaches towards GAN-ifying real faces.

The first approach was a continuation of my work on a previous project entitled *face2fake*. This approach made use of CycleGAN [7], a model capable of unpaired image-to-image translation. In *face2fake*, I attempted to translate between real faces and fake faces using CycleGAN trained on a dataset of 2000 real faces from Labeled Faces in the Wild (LFW) [8] and 2000 StyleGAN-generated faces from ThisPersonDoesNotExist [2]. However, time constraints and my choice to use LFW restricted me to using 250×250px images. This model ended up suffering mode-collapse, which was an interesting result but wasn't what I set out to achieve. Additionally, the low resolution meant that subtle artifacts which contribute to the overall uncanniness of generated face were often covered up by pixelation. As I began working on this project, I started by doubling the size of the dataset to 4000 images in each category, and quadrupling the resolution to 1024 pixels in an attempt to train an HD CycleGAN between real and fake faces. I downloaded 4000 high-resolution real faces from the FFHQ dataset and used a script to download 4000 unique generated faces from ThisPersonDoesNotExist, and stored these in a PersistentVolume on the Nautilus HyperCluster. I then ran a job which reserved a V100 GPU and ran a custom script following my implementation of *face2fake*, saving checkpoints and generated results to the PersistentVolume. Since CycleGAN requires four networks to be loaded—two generators and two discriminators—I had to set the crop size to 512 pixels during training. This meant that during training the model would not rescale the image, but instead grab random square 512 pixel patches for each training batch. I hoped that by looking at subsets of a high resolution image, the model should have been capable of discovering subtle differences between the real and fake faces. This was not the case.

Throughout the 50 epochs of training (each epoch taking roughly 47 minutes), I grew increasingly worried that the model had simply learned the identity transformation. After each epoch, the model would save a sample translation from Real to Fake, but the vast majority of these translations only appear to differ in saturation and contrast as opposed to in content. The one area which the model appeared to be making some progress was in the eyes, as can be seen in the translation sampled after epoch 38. However, only a few samples after epoch 38 exhibited any modification to the eyes, and no other samples exhibited any substantial change to the face. All hope was lost for this model.

My second approach was a short-lived exploration into Neural Style Transfer. I briefly entertained the idea of style transfer onto a real face using multiple fake faces at the style sources. Using Cameron Smith's Neural Style repository [9], I attempted this with a single content image of Nicolas Cage and ten style images from ThisPersonDoesNotExist. I played around with Style/Content weight and a couple other parameters, but the output resulted in a painterly effect even though no paintings were part of the style images! This is likely due to how the model had been pre-trained, and perhaps could have been addressed if I attempted to fine-tune the model. Overall, neural style transfer on the entire image would likely not ever produce good results for the GAN-ification that I was trying to achieve, since the effect on the face would be far too dependent on the background of the fake faces, as opposed to the faces themselves. However, if it were possible to carry out masked style transfer such that the mask is applied not only to the content image, but also to the style images, then perhaps this approach would have found success.

Finally, I realized, *what better way is there to make an image look like it was generated by a GAN than to actually generate it with a GAN*? The third and final approach was to use an encoder for StyleGAN such that a latent-vector could be found which would produce an output perceptually similar to a target image, but still possessing any artifacts that StyleGAN creates when generating faces. This approach was quick, easy, and produced great results, albeit the process was a fair bit less poetic.

Result:

Ultimately, using the StyleGAN-Encoder, I was able to accomplish what I set out wanting to achieve: I transformed real faces into their GAN-ified counterparts. The results starting from the null latent-vector are delightfully subtle and uncanny for some such as Nicolas Cage, eerily unnatural for others such as Ivanka (it appears that her real photo is just as unnatural looking as the generated one!), and downright uncomfortable for others such as myself.

These results aren’t quite noticeable in such a small format!

The results starting from the previous face's latent vector produce almost painterly styled outputs, especially during the first few hundred iterations, and it almost appears as if these outputs could pass for artist renditions of a subject.

Obama reconstructed from Twomey's latent-vector, at about 400 epochs. Looks like a painterly rendition.

Then, strangest of all, it's oddly pleasant (in a non-creepy way, I assure you) to watch the faces be formed from a generic person into an identifiable one. It's captivating to see subtleties gradually emerge. It's intriguing to ponder at what point the face becomes identifiable as its target, instead of simply a person that does not exist. This seems to be achieved very quickly, within the first few hundred iterations.

And, finally, it is worth noting that the videos do a great job of illustrating where StyleGAN artifacts reside. For example, the video of Jackie Chan's reconstruction nicely exemplifies the teeth artifacts discussed in the Technique section (1). Notice that throughout the entire reconstruction process, the teeth are directly facing the camera, despite the orientation of the head moving from facing right to facing left.

Notice that the teeth stay in more or less the same position the whole time.

Reflection:

I am pleased with my results, but have three qualms I would have wanted to address if given the time.

1. I wanted to run more iterations. Though many of the faces converged within the 1500 epochs I ran, I would have liked to see all of the reconstructed be just subtly incorrect, and avoid any glaring inconsistencies with real human faces.
2. I wanted to run this process on more people. Every face seemed to result in an interesting progression, revealing its own story. For example, comparing the rate of convergence between faces allowed me to uncover bias in the training set—the reconstruction converges a lot quicker for young, white faces than it does for other ethnicities.
3. Now, I really want to run this process with StyleGAN2. Taking a look at the repository, it appears that they now have latent-space projection built in to their developed code. The "droplet" artifact bugged me when I'd sit and watch the loop—it would be nice to get rid of that!

REFERENCE:

[1] Tero Karras, Samuli Laine, Timo Aila, and Nvidia. *A Style-Based Generator Architecture for Generative Adversarial Networks*. December 2018. https://github.com/NVlabs/stylegan

[2] Phillip Wang. *This Person Does Not Exist*. February 2019. https://thispersondoesnotexist.com

[3] Ivan Braun et al. and Generated Media Inc. *100K Faces project*. https://generated.photos

[4] AP News. *"Experts: Spy used AI-generated face to connect with targets"*. June 2019. https://apnews.com/bc2f19097a4c4fffaa00de6770b8a60d

[5] Dmitry Nikitko. *StyleGAN-Encoder*. February 2019. https://github.com/Puzer/stylegan-encoder

[6] Tero Karras, Samuli Laine, Timo Aila, and Nvidia. Flickr Faces High Quality dataset. February 2019. https://github.com/NVlabs/ffhq-dataset

[7] Jun-Yan Zhu*, Taesung Park*, Phillip Isola, Alexei A. Efros. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. November 2017. https://junyanz.github.io/CycleGAN/

[8] Gary B. Huang, Manu Ramesh, Tamara Berg, Erik Learned-Miller. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. University of Massachusetts. October 2007. http://vis-www.cs.umass.edu/lfw/

[9] Cameron Smith. *Neural Style Transfer*. November 2016. https://github.com/cysmith/neural-style-tf

CODE:

See https://github.com/ucsd-ml-arts/ml-art-final-parker-final

RESULT:

See `index.html` (you may need to enable GitHub Pages on the repository then go to https://ucsd-ml-arts.github.io/ml-art-final-parker-final/, or clone the repository and open the html file locally)

See `images/generated_images/*` for individual generated photos and videos.