CS 398 PRG 4

Ben van Oostendorp
Affiliation

Problem

For the final project for this class, I wanted to work with GANs. The main purpose was to generate images and also acheive a more stable result, since the most recent attempt I had at working with GANs resulted in a generator that would fall apart over time. I sought out some new types in order to see what would address this by looking at the GAN Zoo (GAN zoo). From here, I read about many different GANs, a lot of them more complicated than I had time to sit down and dissect, but found an elegant solution proposed. The version I ended up implementing is called a Least Squares GAN (LSGAN) where instead of minimizng loss by using binary cross entropy, we instead us Mean Squared Error (MSE) which leads to more stable GANs with the same quality of output.

Dataset

The dataset I ended up using is called "Cat faces 64x64 (For generative models)." This dataset is taken from the CAT Dataset and zoomed into just the faces of each cat (dataset). This dataset seemed really interesting since it had a lot of images (15,747) and they were already preprocessed.

Implementation

Following the paper, I built basically the same GAN that I did for a previous project (with some tuning since that one needed a lot of work) and moved from using a Deep Convolutional GAN (DCGAN) into a GAN that uses upsample blocks with convolution layers. These differ in because DCGANs use Conv2dTranspose layers instead, which "work backwards" through a covolution. In my implementation, we upsample and then have a regular convolution after. This approach can be subjective to some noise since the upscaling might not be perfect, but it leads to decent results which will be discussed later. One other interesting thing with upsampling is that we can change the method of upsampling to increase visual performance or compute performance. In my case I went with bicubic upsampling, which has great quality but significantly reduces performance time.

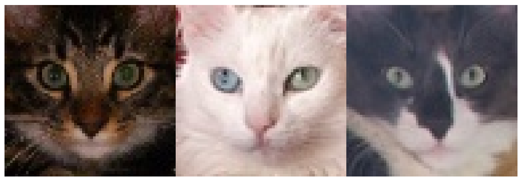

Figure 1. Examples from the dataset

Results

After running my LSGAN twice, once on my computer and once using a multi-gpu setup with the school computer, it generated pretty great looking cats. The images are still noisier than I would like them to be, but it's definitely a good step. One thing I look to try and do would be using the GAN to do the upscaling since I think that's a decent amount of where the noise comes from. For example, transitioning back into using a DC Generator or maybe another method. However, I'm definitely happy with the results, and the biggest change from the previous attempt and this implementation is that the GAN did not fall apart during training. There's a few reasons this might be the case, but it definitely seems that the Least Squares method holds to be really good. One reason might be because of the size of the dataset, while the other is perhaps the MSE did greatly affect its stability.

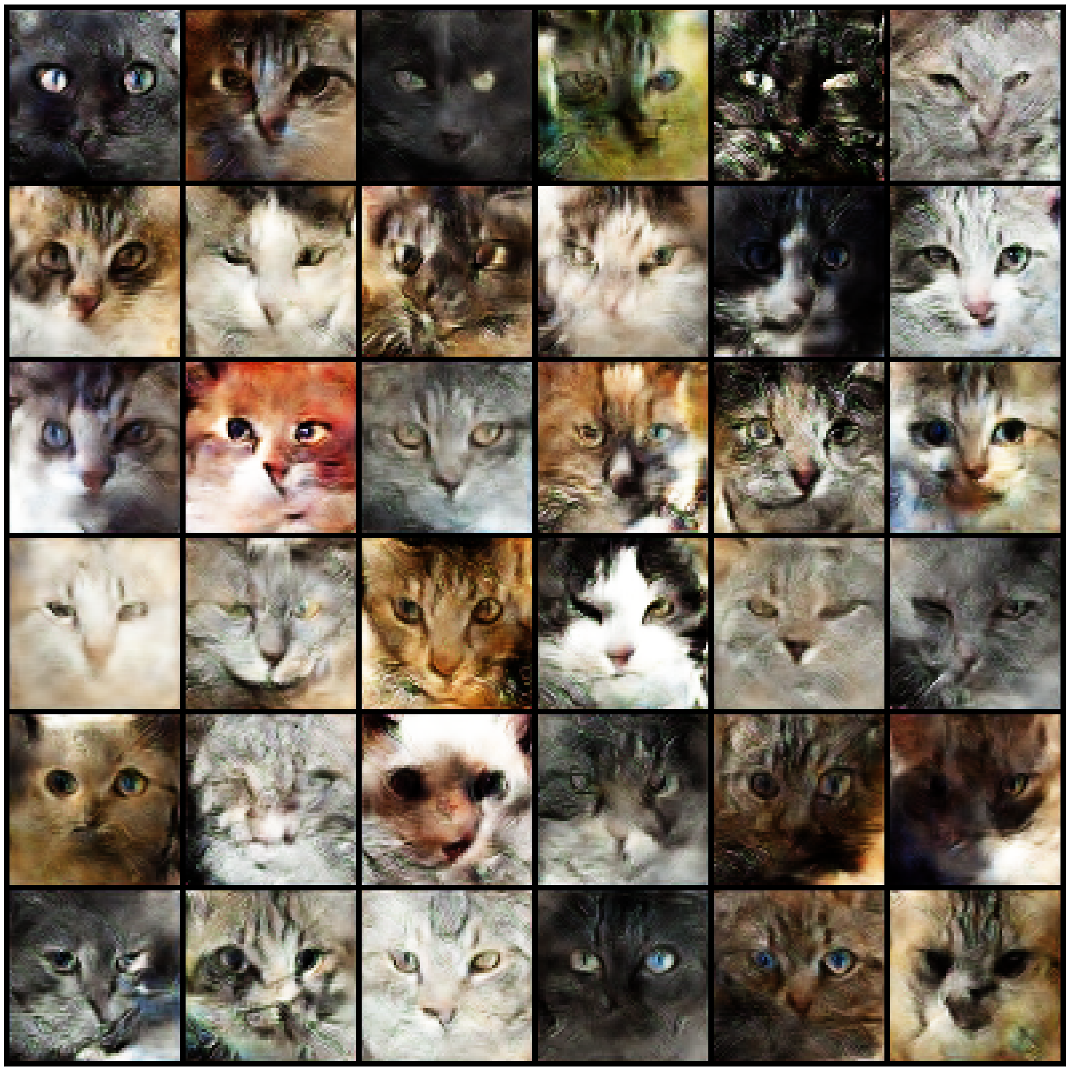

Figure 2. Results from training the GAN