Generating Images with FastGAN

mccb22

Abstract

This paper proposes using an extension of a generative adversarial network, FastGAN to generate images of faces from the Flickr-Faces-HQ (FFHQ) dataset at 128x128 resolution. FastGAN is a lightweight GAN architecture designed to be used with low computing power, as in this experiment. The use of a self-supervised discriminator trained as a feature encoder and a skip-layer channel-wise excitation module allow generation of clear and realistic facial images much faster than other state of the art methods.

1 Methodology

FastGAN is trained on the FFHQ thumbnails dataset to generate images.

1.1 FFHQ

The main dataset used in this paper is the Flickr-Faces-HQ dataset [2]. FFHQ contains 70,000 images of human faces all taken from Flickr, with variation in background, age, ethnicity and facial accessories. I used the FFHQ thumbnails dataset, which is identical to the original dataset but all images are resized to 128x128 resolution. This prevented me from using up unneccessary storage space as I did not need images that were the full 1024x1024 resolution. This paper will try and generate realistic images of faces that look like they belong in the FFHQ dataset, even if they are not part of the original set of images.

1.2 Generative Adversarial Networks

Generative adversarial networks or GANs involve 2 neural networks. A classifying discriminator trained with supervised learning, and a generator that generates images from random noise [1].

The discriminator is trained to classify images as real or fake, whilst the generator is trained to generate images that the discriminator will classify as real. The gradients of the discriminator can actually be used to train the generator via gradient ascent.

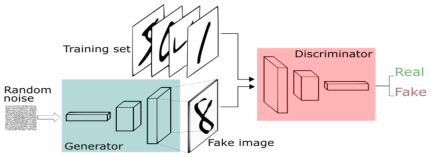

Figure 1: Architecture of a simple GAN [7]

Therefore the discriminator D is trying to maximise the loss function, and the generator G is trying to minimise it. In the paper by Liu et. Al FastGAN uses the hinge version of adversarial loss [4], however I change this to dual contrastive loss as it is shown to reduce FID when compared to hinge loss in GAN training on FFHQ [9].

$$L_{real}^{contr}(G,D) = -\mathbb{E}_{x\sim I_{real}}[\log(1+\sum_{z\sim N} e^{D(G(z))-D(x)})] \tag{1}$$

$$L_{fake}^{contr}(G,D) = -\mathbb{E}_{z\sim N}[\log(1+\sum_{x\sim I_{real}} e^{D(G(z))-D(x)})] \tag{2}$$

$$\min_G \max_D = L_{real}^{contr}(G, D) + L_{fake}^{contr}(G, D) \tag{3}$$

Where I_{real} is the dataset and N is the set of random noise that can be fed to the generator.

The discriminator wants to maximise $D(x)$ by classifying real samples as real and minimise $D(G(z))$ by classifying fake samples as fake. The generator is trying to maximise $D(G(z))$ by producing images the discriminator classes as real.

Training of the discriminator and generator are alternated; as iterations increase the generators output looks less like random noise and more like images from the original dataset. After training, feeding a random noise image to the generator that doesn't correspond to a real image will output a new image that looks like it is from the dataset, but isn't.

1.3 FastGAN

Realistic image generation is difficult under the constraints of low computing power and limited time, with state of the art GANs being trained with multiple high end GPUs for many days [3]. This leads to a high risk of overfitting and mode collapse, with the added difficulty of many regularisation techniques for the discriminator being less effective at low batch size. Furthermore, dataset distribution at high resolutions is sparse, so the model architecture needs to result in strong discriminator gradients that effectively train the generator, whilst still having a model deep enough to produce high resolution images [6].

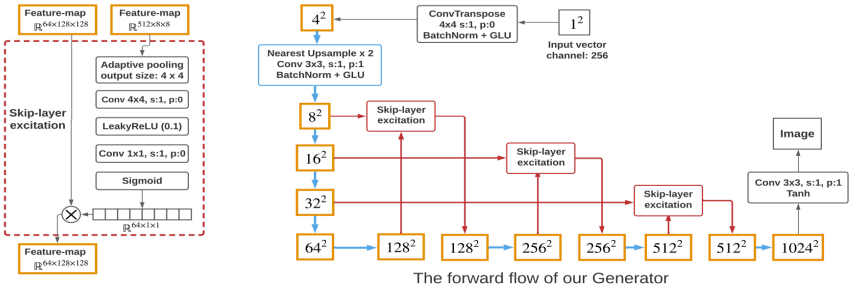

Figure 2: An example skip-layer excitation module and architecture of a generator using these modules in red. Feature map sizes are shown in yellow boxes with blue boxes/arrows showing upsampling. Note this is not the exact architecture used in my model as I adjusted the architecture to effectively produce 128x128 images [6]

The latter of these problems is commonly addressed with residual blocks that use skip connections, but these also increase computation time. Liu et. Al propose the skip-layer channel-wise excitation (SLE) module, that instead of adding activations, performs channel-wise multiplication of activations. This not only reduces the number of convolutions needed but allows skip connections between feature maps of different resolutions.

$$y = F(x_{low}, [W]_i) \cdot x_{high} \tag{4}$$

Where x_{low} is the lower resolution feature map that undergoes F operations with weights W_i. Giving a tensor with shape [batch size,channels in x_{high},1,1] that is multiplied by x_{high}.

The result is stronger gradients to train the generator with far less extra computation than residual connections [6].

To reduce risk of mode collapse, Liu et. Al use a regularization approach for the discriminator independent of batch size. The discriminator is trained as an autoencoder. As shown in figure 3 the discriminator has an encoder structure and is trained alongside small decoders.

Decoders are trained with reconstructive loss on real samples. This is added to the discriminator loss function shown earlier.

$$L_{recons} = -\mathbb{E}_{f \sim D_{encode}(x), x \sim I_{real}}[||G(f) - T(x)||] \tag{5}$$

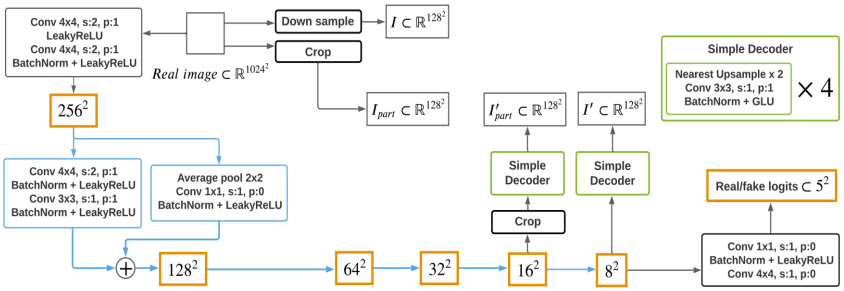

Figure 3: Example discriminator architecture. Blue boxes/arrows represent residual downsampling, green boxes represent decoders [6]. I altered this for 128x128 images in my model

$$\max_D = L_{real}^{contr}(G, D) + L_{fake}^{contr}(G, D) + L_{recons} \tag{6}$$

Where f are feature maps from the discriminator processed by decoder G, and T is processing on real images. $G(f) - T(x)$ represents perceptual loss, a measure of difference in image style. This approach encourages the discriminator to learn important features of images at high composition level and low texture level so that decoders reconstruct images similar to the original [6].

My implementation of FastGAN is based off code from [5]. As this code is only compatable with images of a minimum size 256x256 I altered it to work with 128x128 images. This involved altering convolutions, skip connections and encoders so the model is able to produce 128x128 images yet still learns to extract important high and low level image features at reduced resolution. I also swapped the hinge loss with dual contrastive loss as implemented in [8] as this is shown to give improved performance [9].

2 Results

Overall images are clear and look like faces, indicating the model has learnt a good composition for the overall structure and lower level textures. Images are very diverse with a range of gender, ethnicity and accessories. Under close inspection there are artifacts in many of the images, the model seems to have a difficult time discerning hair from backgrounds. Although most images could be easily spotted by a human, the results look fairly realistic.

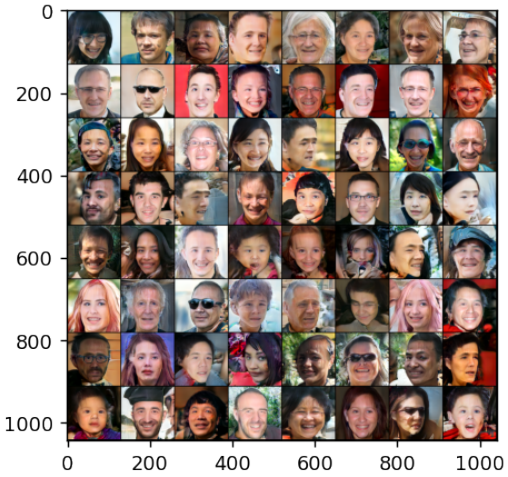

Figure 4: A non cherry picked batch of 64 images generated by the model

Interpolations between pairs in the latent space are shown in figure 5. There is a clear progression from images on the left to the right, and most of the points in the middle look fairly realistic. Images next to each other are very similar, but all seem to show some progression to the image on the other side of the row.

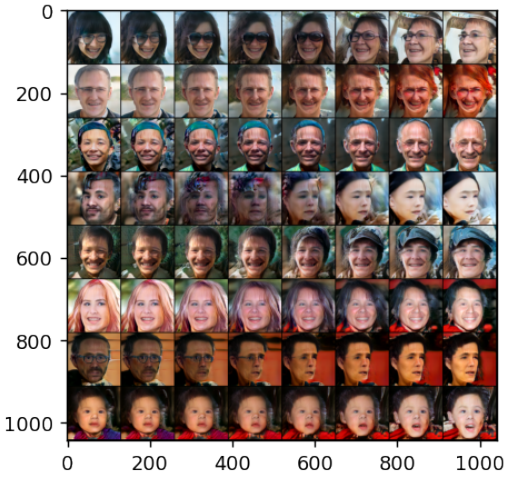

Figure 5: Interpolations between 8 pairs of images

Finally, some of the best samples generated by the model are shown below in figure 6. These images may be able to pass as actual people if someone wasn't looking too closely.

Figure 6: Cherry picked samples

3 Limitations

Although results resemble faces, many generated images still show significant artifacts that stop the results looking really realistic. Maybe altering the position of skip connections and the feature maps of the discriminator that decoders work on could help the model distinguish more higher level features like hair and background. However, in my experiments I was unable to fix these merging artifacts.

It may be the case that the model used simply did not have the capacity to learn to generate perfect faces. Despite the model being impressive under time and computing power constraints it contains many less parameters than the state of the art GANs [3]. Furthermore, to ensure the model learnt fast, many techniques known to boost GAN performance were not used. It would be interesting to see how well this model compared to StyeGAN2 with increased capacity and using further regularisation techniques, more skip connections etc.

Bonuses

This submission has a total bonus of -2 marks (a penalty), it is trained on FFHQ at 128x128 resolution (+2) but is a GAN (-4).

References

[1] Ian Goodfellow et al. "Generative Adversarial Networks". In: *arXiv:1406.2661 [cs, stat]* (2014). URL: http://arxiv.org/abs/1406.2661.

[2] Terro Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *arXiv:1812.04948 [cs, stat]* (2019). URL: http://arxiv.org/abs/1812.04948.

[3] Terro Karras et al. "Analyzing and Improving the Image Quality of StyleGAN". In: *arXiv:1912.04958 [cs, eess, stat]* (2020). URL: http://arxiv.org/abs/1912.04958.

[4] Jae Hyun Lim and Jong Chul Ye. "Geometric GAN". In: *arXiv:1705.02894 [cond-mat, stat]* (2017). URL: http://arxiv.org/abs/1705.02894.

[5] Bingchen Liu. *A Fast and Stable GAN for Small and High Resolution Imagesets - pytorch*. URL: https://github.com/odegeasslbc/FastGAN-pytorch. (accessed: 09.02.2022).

[6] Bingchen Liu et al. "Towards Faster and Stabilized GAN Training for High-fidelity Few-shot Image Synthesis". In: *arXiv:2101.04775 [cs]* (2021). URL: http://arxiv.org/abs/2101.04775.

[7] Thalles Silva. *An intuitive introduction to Generative Adversarial Networks (GANs)*. URL: https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/. (accessed: 12.24.2021).

[8] Phil Wang. *lightweight-gan*. URL: https://github.com/lucidrains/lightweight-gan. (accessed: 09.02.2022).

[9] Ning Yu et al. "Dual Contrastive Loss and Attention for GANs". In: *arXiv:2103.16748 [cs]* (2021). URL: http://arxiv.org/abs/2103.16748.