Generative Adversarial Networks

Objective: -

To improve the quality of computer-generated images and bring about changes in various features by tweaking the latent vector.

Dataset: -

The Flickr-Faces-HQ (FFHQ) image dataset is used which was originally created as a benchmark for generative adversarial networks (GAN). This dataset consists of 70k+ high-quality (1024x1024) PNG images at 1024×1024 resolution and contains considerable variation in terms of age, ethnicity and image background. It also has good coverage of accessories such as eyeglasses, sunglasses, hats, etc.

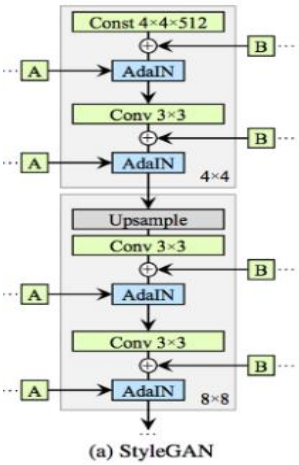
(This dataset was imported onto collab using the git repository)

Literature Survey: -

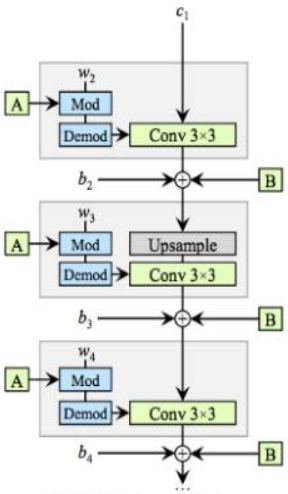
Implemented **StyleGAN2** to improve the image quality from the previous review which had results from GAN. StyleGAN is improved version of the predecessor DCGAN and ProgressiveGAN. StyleGAN used 3 main aspects to improve the image quality which are the FFHQ dataset, weight demodulation and adaptive instance normalization. The above conclusions are derived from "**Analyzing and improving image quality by StyleGAN**" by Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten and Timo Aila.

NVIDIA researchers used normalization layers for image synthesis applications. StyleGAN uses adaptive instance normalization to control the influence of the source vector on the resulting generated image.

Adaptive Instance Normalization is a normalization layer derived from research into achieving faster Neural Style Transfer. Neural Style Transfer demonstrates a remarkable disentanglement between low-level "style" features and high-level "content" features.



(a) StyleGAN



(d) Weight demodulation

Results from StyleGAN2 implementation and its working: -

1. StyleGAN generates seeds which in turn develops the 512 dimensional latent vector.
2. These seeds are expanded and each of these vectors represents a slight feature variation.
3. Images are generated from the Nvidia StyleGAN code.

Image (a)

Image(b)

NOTE: Here, even though the created images are high quality and indistinguishable from the real ones at the first glance, but the background and ears give it away. As it can be clearly seen that in image (a) that the background is kind of mutated structure and in image (b) the ear size and height are a mismatch.

What did I do: -

I came across a Kaggle competition question which read – "**Implement a StyleGAN to depict one people with and without glasses**". Upon digging deeper, the process to do this was to modify a single GAN latent vector since that's what contains the various features. I understood that this latent vector is like a genetic sequence.

After handpicking the images with spectacles/glasses; iterate a single pic through 500 at a batch rate of 10, to find the nominal dimensional range wherein the glasses disappear from the image. The final result is as follows: -

From the above two images, **at closer observation**, we can see that there are various differences like change in hair length, a bit of facial feature differences etc. But, all in all at first glance they look like the same person with and without glasses.

Performance analysis: -

Two metrics are the most commonly features while analyzing GAN images, which are-

1. Inception Score or IS for short, is an objective metric for evaluating the quality of generated images, specifically synthetic images output by generative adversarial network models.
   IS Score = 9.46 **IS score of state of art model=11.93**

2. The Fréchet Inception Distance score, or FID for short, is a metric that calculates the distance between feature vectors calculated for real and generated images. [Lower the FID score, better the model]
   FID Score=3.15 **FID score of state of art=2.2**

Conclusion: -

StyleGAN2 is already released by NVIDIA and further enhancements are on route. So, the next step is to either improve the performance metrics or indulge into further varieties of GAN like SketchyGAN.

Future Work: -

Understand the intricacies of edge mapping to further implement SketchyGAN and enhance its implementation.