Generating Images using StyleGAN2 + ADA

Anonymous author

Abstract

This paper proposes using a StyleGAN2 architecture in combination with an adaptive discriminator augmentation (ADA) mechanism trained on the FFHQ dataset. This mechanism passes the images shown to the discriminator through an augmentation pipeline and dynamically adjusts the strength of augmentation. We produce realistic interpolation and achieve a reasonable balance between diversity and quality of images.

1 Methodology

1.1 StyleGAN

The StyleGAN [1] is an extension of conventional GAN architecture. It proposes empirically verified changes to the generator model inspired by style transfer literature [2] which we discuss.

Conventionally an image is generated using a GAN, by sampling a latent factor z from a normal or uniform distribution. Conversely, the StyleGAN applies an eight layered fully-connected neural network (the mapping network) that converts the latent factor z into an intermediate latent factor w which is then used to generate the image. The goal of this mapping network is to create untangled features that are easy to render by the generator and avoid feature combinations that do not happen in the training dataset.

Secondly, in a conventional GAN, the latent factor z is an input to the first network layer only. This is disadvantageous as its role diminishes in the deeper layers. In the style-based generator, we apply a separately learned affine operation A to transform w in each layer.

In addition to these main changes, the authors empirically verify the addition of 5 more changes that when added to a baseline Progressive GAN improve the FID score.

Firstly, nearest-neighbour up and downsampling is replaced with bilinear sampling in both the generator and discriminator architectures. The second improvement is replacing Pixel-Norm so that adaptive instance normalisation (AdaIN) applies styling to the spatial data instead. To do this, for each layer, a pair of style values are computed from w. We use these to scale and add bias to the spatial feature map i to apply the style.

$$\text{AdaIN}(x_i, y) = y_{s,i} \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}$$

Thirdly, the input to the first layer is replaced by a learned constant matrix with dimension 4×4×512 as experiments verified that there was no benefit in adding a variable input. The fourth improvement is the introduction of noise into the spatial data to create stochastic variation. The final improvement is mixing regularisation. Previously, we only used a single latent factor z in deriving the styles. With mixing regularization, after reaching a certain spatial resolution we switch from the latent factor z_1 to a different latent factor z_2 (which are both passed through the mapping network to create w_1 and w_2). The final architecture with these discussed changes can be seen in part b) of Figure 1.

1.2 StyleGAN2

StyleGAN achieved a state-of-the-art performance however sometimes images are plagued by artifacts. StyleGAN2 [3] circumvents these through specific design changes to the StyleGAN.

The authors attribute the blob-like artifacts to the AdaIN operation. To circumvent this, they redesign the normalisation. To facilitate their redesigned normalisation, they adapt some of the generator architecture. First, they simplify how the constant c_1 is fed in at the beginning by removing any processing (e.g. removing the additive bias and noise). They also remove the mean in normalising features and move the noise module outside the style module. This can be seen in Figure 1 part c.

They redesign the normalisation by replacing the instance normalization design with demodulation that is applied to the convolution feature map weights. To do this, we first combine modulation and convolution by scaling the convolution weights:

$$w'_{ijk} = s_i \cdot w_{ijk}$$

Where s_i is the scale corresponding to the i^{th} input feature map, and j and k are the output feature maps and spatial footprint of the convolution. To remove the effect of s_i from the statistics of the convolution's output feature maps we then demodulate as follows:

$$w''_{ijk} = \frac{w'_{ijk}}{\sqrt{\sum_{i,k} {w'_{ijk}}^2 + \epsilon}}$$

StyleGAN2 further supplements this performance by empirically-verified design choices. Firstly, R1 regulation is only applied once every 16 mini-batches. This reduces the computational cost at no expense to the performance. Secondly an additional path regularization term is introduced which is given by:

$$\mathbb{E}_{\mathbf{w},\mathbf{y}\sim\mathcal{N}(0,\mathbf{I})} \left(||\mathbf{J}_{\mathbf{w}}^T \mathbf{y}||_2 - a\right)^2 \tag{1}$$

Where $\mathbf{J}_{\mathbf{w}} = \partial g(\mathbf{w})/\partial \mathbf{w}$. It is advantageous that the same displacement in the latent space should yield the same magnitude change in the image space, regardless of the value of the latent factor. This regularisation term adds cost when the change in the image space is different from the ideal expected displacement. This contributes to interpolations appearing so realistic.

Thirdly, the progressive growing architecture is revised as the authors attribute 'phase' artifacts to it. This is where certain features take a fixed location regardless of the orientation of the objects. StyleGAN2 explores exploiting a skip connection design and other residual concepts similar to ResNet. It finds that using skip connections for the generator architecture and a residual network for the discriminator is most effective. The final changes can be seen in Figure 1 part d.

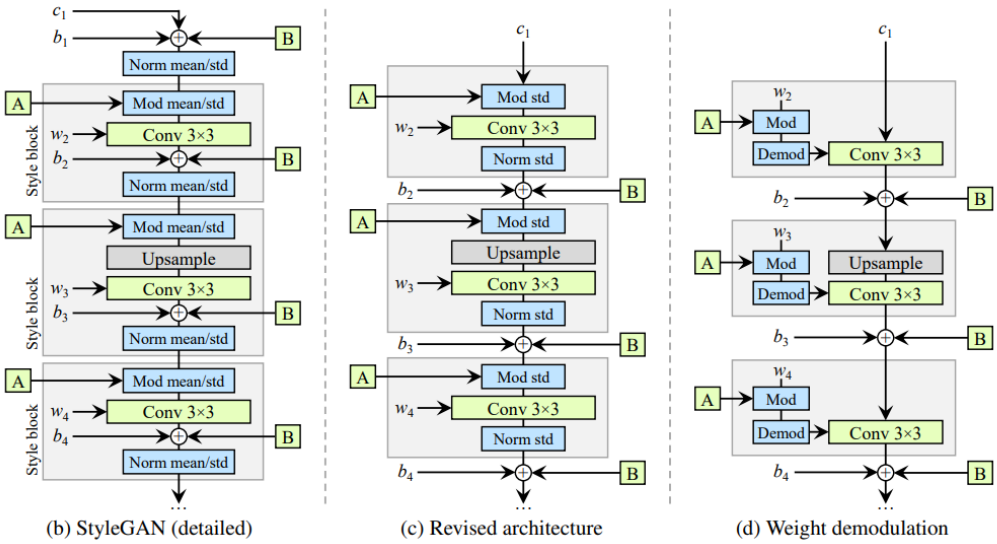


Figure 1: The evolution of the StyleGAN architecture taken from [3]

To circumvent the high computational expense and long training times involved with training a StyleGan2, we trained on a smaller dataset. With a smaller dataset, we can expect the

discriminator to overfit yielding meaningless feedback to the generator [4]. To mitigate this issue we considered different dataset augmentations that have proven very effective in other areas of deep learning.

There is a recent body of work that all focuses on non-leaky discriminator augmentations: T. Karras et al. [5] (ADA) Z. Zhao et al. [6], Tran et al. [7], and S. Zhao et al. [8]. ADA proposes the most maximally diverse pipeline, which has shown to be advantageous in other data augmentation literature in deep learning [9]. Moreover, ADA demonstrates that an optimal augmentation strength varies across training iterations indicating that a varying augmentation strength is superior to any set of fixed augmentation parameters which all the other literature use.

Certain regions in the latent space w may not have enough training data to learn it accurately. To avoiding sampling from these regions we truncate the latent space w. A lower truncation coefficient (around 0.7) yielded extremely realistic but not very diverse samples whereas no truncation yielded diverse but not very realistic samples. The final samples used a truncation coefficient of 0.8.

To train the generator and discriminator, we set them up to play the following mini-max game, where the generator tries to fool the discriminator and the discriminator tries to maximise its differentiation power between real and generated:

$$\min_G \max_D \mathcal{L}_{GAN}(D, G)$$

We use the non-saturated loss for the generator so we maximise the log of the discriminator probabilities for generated images instead of minimizing the log of the inverted discriminator probabilities. This loss, across m images, is given by:

$$\nabla_{\theta g} \frac{1}{m} \sum_{i=1}^{m} \log(D(G(z^{(i)})))$$

We also include R_1 regularisation and 1 path regularisation term. The discriminator loss, across m images is given by:

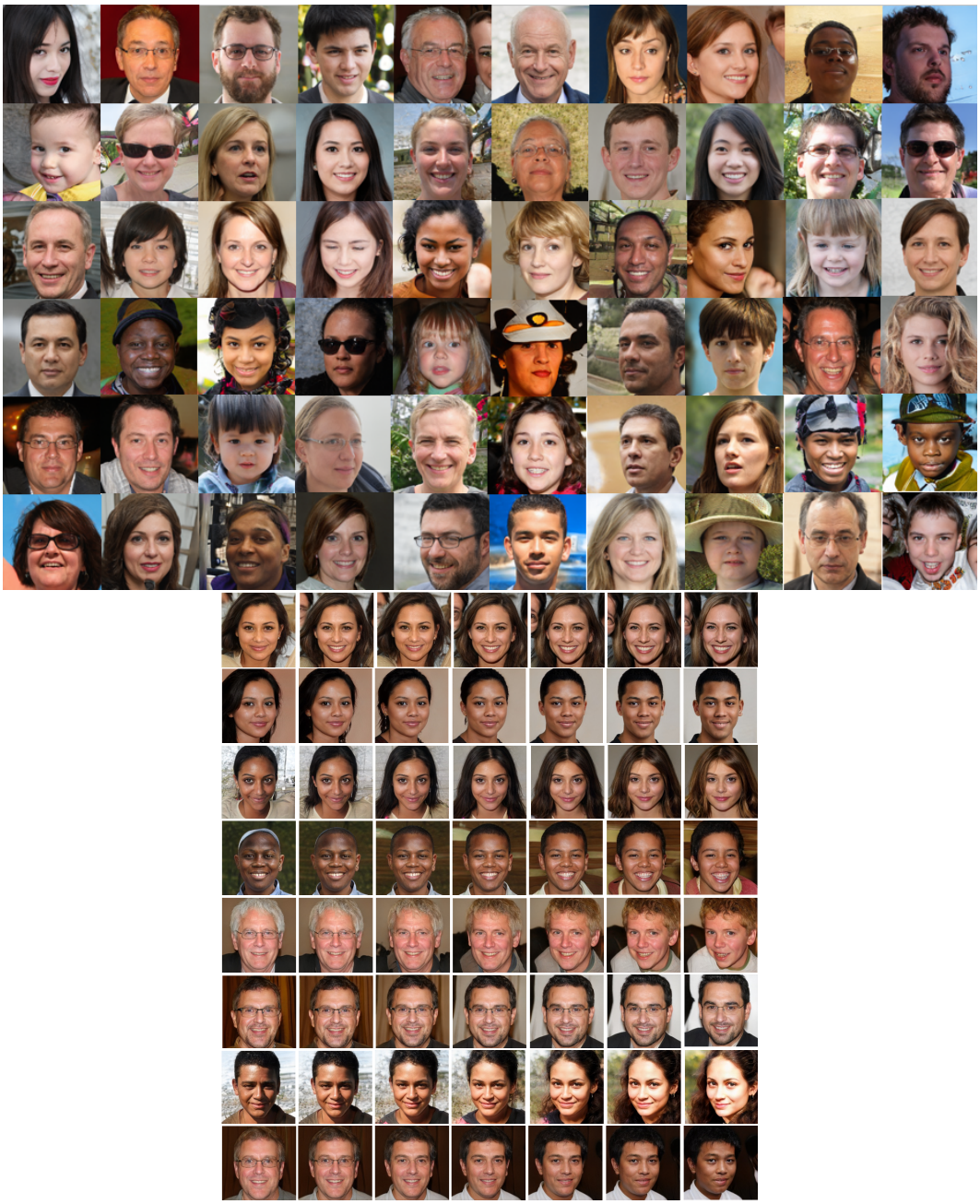
$$\nabla_{\theta d} \frac{1}{m} \sum_{i=1}^{m} \log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))$$

We train both the discriminator and the generator iteratively via backpropagation. We apply a pipeline of 18 non-leaky transformations. Each image is augmented by applying these transformations in a fixed order with a probability p. This probability p is adapted depending on whether the discriminator is over or under fitting. These augmented images are shown to the discriminator and the discriminator is evaluated using only augmented images. As the augmentations are also executed when the generator is training, we necessitate them to be differentiable.

We trained the StyleGAN2 + ADA on the FFHQ dataset. We trained at the 256 x 256 resolution to be able to inherit the successful hyperparameters that the StyleGAN2 + ADA paper uses on the FFHQ dataset.

2 Results

We present our results. We show 22 of the best images generated, a sample of 60 images generated with 60 random seeds and images generated by linearly interpolating between points in the w latent space where we have used the best images.



3 Limitations and Further Work

The lowest FID score this paper achieved was 5.08, which is slightly higher than the StyleGAN2 + ADA. Moreover, the diversity could be stronger. To mitigate these issues, we could train for longer and truncate the latent space less. Finally, we notice that it struggles with hats and other people in the background so we could introduce more samples like this in the dataset.

Bonuses

This submission has a total bonus of 0 marks. (-4 for GAN, +4 for training FFHQ at 256)

References

[1] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410.

[2] Xun Huang and Serge Belongie. "Arbitrary style transfer in real-time with adaptive instance normalization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1501–1510.

[3] Tero Karras et al. "Analyzing and improving the image quality of stylegan". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8110–8119.

[4] Martin Arjovsky and Léon Bottou. "Towards principled methods for training generative adversarial networks". In: *arXiv preprint arXiv:1701.04862* (2017).

[5] Tero Karras et al. "Training generative adversarial networks with limited data". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 12104–12114.

[6] Zhengli Zhao et al. "Image augmentations for gan training". In: *arXiv preprint arXiv:2006.02595* (2020).

[7] Ngoc-Trung Tran et al. "On data augmentation for gan training". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 1882–1897.

[8] Shengyu Zhao et al. "Differentiable augmentation for data-efficient gan training". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7559–7570.

[9] Ekin D Cubuk et al. "Randaugment: Practical automated data augmentation with a reduced search space". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 702–703.