A Style-Based Generator Architecture for Generative Adversarial Networks

> Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." *arXiv preprint arXiv:1812.04948* (2018).

Abstract

style transfer literature를 사용함으로 새로운 gan을 제안.

이는 생성된 영상 내에서 high-level의 특징들을 학습 가능함.

disentanglement륵 측정하는 정량적인 방법을 제안.

추가로 high quality의 human face dataset을 제공함

Introduction

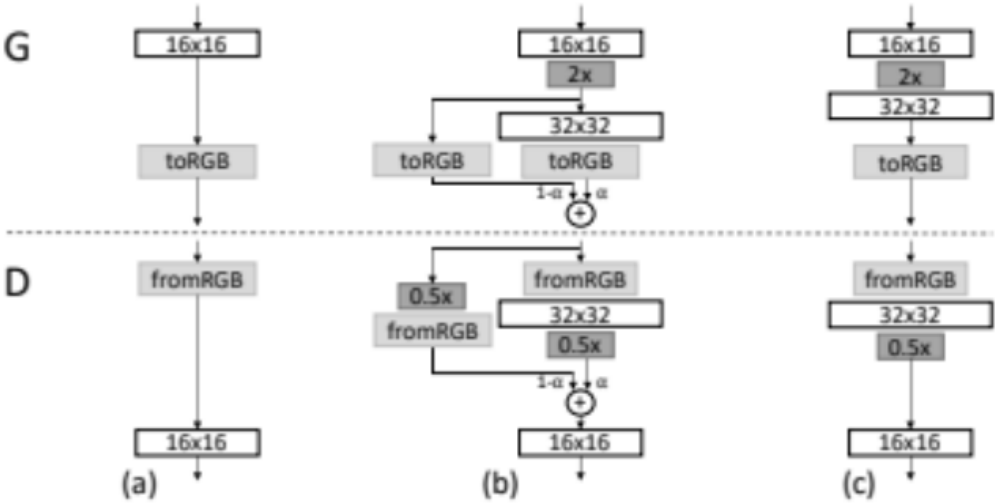
기존 gan에서의 generator는 stochastic feature에 대해 설명이 부족함.
latent space와 그에 interpolation은 generator 사이에서 비교할 수 있는 정량적 방법을 제공하지 않음.

제안한 gan의 구조는 style을 변환시키는 latent code를 기반으로 conv layer에서 image의 style을 조정하는 것이다. 이를 통해 각 다른 scale 내에서 영상의 특징 조절할 수 있다.

Progressive GAN

논문의 base line이 되는 모델로 저해상도 (4x4) 층으로 시작하여 학습이 진행됨에 따라서, generator와 discriminator에 층을 추가함으로 이미지의 해상도를 향상 시킨다.

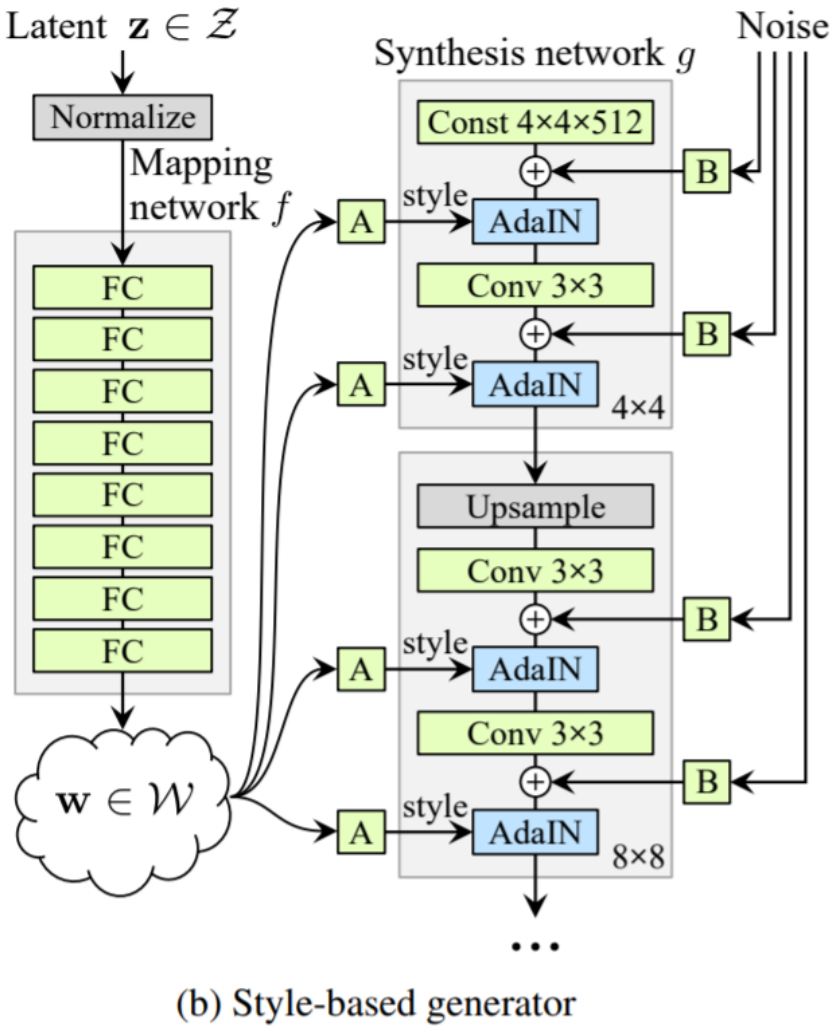
층을 곧바로 추가한다면, 학습이 전혀 안된 층의 영향으로 학습된 저해상도 층에 영향을 끼칠 수 있으니, smooth fading in을 활용하여 추가한다.



위 그림처럼 새로운 층을 추가할 때, 학습된 층과, 새로운 층의 가중치를 1-a,a (a는 0에서 점진적으로 증가하여 1까지)로 두어 학습을 진행한다.

Style-based generator

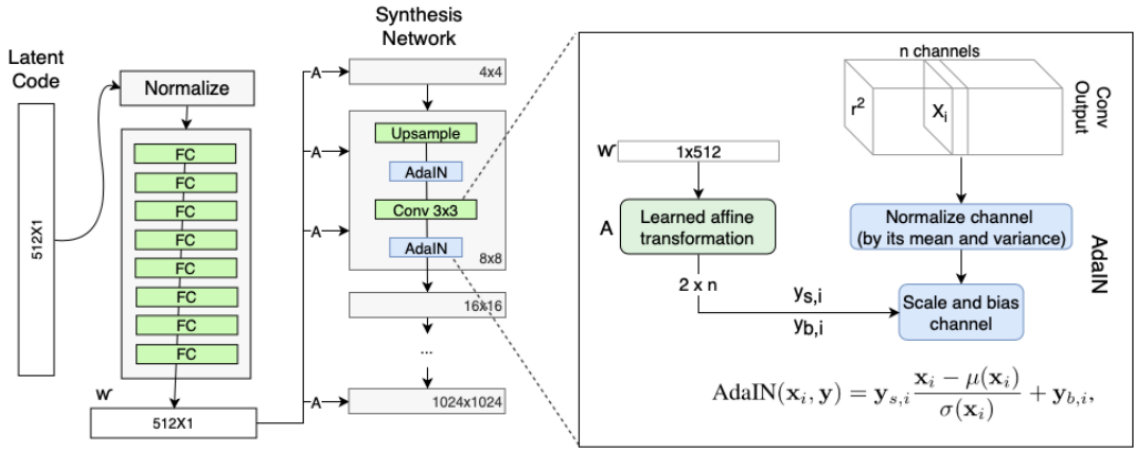
generator의 구조는 아래와 같다.



(b) Style-based generator

style gan에서는 latent space Z 내의 latent code z 가 있을 때, 이를 style인 W에 mapping 하는 network f(8-layer mlp) 를 구축한다.

AdaIN



The generator's Adaptive Instance Normalization (AdaIN)

저번 dcgan paper에서 batch normalization을 적용할 때 sample oscillation등의 문제의 개선 방안으로 Adaptive Instance Normalization 적용함.

$$AdaIN(x_i, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y)$$

AdaIN은 데이터를 normalization하는 역할을 하는데, channel-wise로 normalization 이후 y(style)의 mean, variance에 맞는 분포로 변환시킨다. 이를 통해 각 style에 맞는 분포로 normalization이 가능하다.

style gan의 generator에서는 latent code w의 shape이 1,512 이므로 AdaIN에 적용할 수 있는 shape을 맞추기 위해 affine transform을 사용하였다.

위의 그림처럼 3x3 conv 이후 adain을 활용해 style에 맞는 distribution으로 바꾸어 줌으로 영상에 대해 style을 입힐 수 있다고 한다.

	Method	CelebA-HQ	FFHQ
A	Baseline Progressive GAN [30]	7.79	8.04
B	+ Tuning (incl. bilinear up/down)	6.11	5.25
C	+ Add mapping and styles	5.34	4.85
D	+ Remove traditional input	5.07	4.88
E	+ Add noise inputs	**5.06**	4.42
F	+ Mixing regularization	5.17	**4.40**

Table 1. Fréchet inception distance (FID) for various generator designs (lower is better). In this paper we calculate the FIDs using 50,000 images drawn randomly from the training set, and report the lowest distance encountered over the course of training.

A. base line(pggan)

B. bilinear interpolation, long training, tuned hyperparameters

C. mapping network, adain

D. const (4 x 4 x 512)

E. add noise input

F. mixing regularization

style mixing

F에서의 mixing regularization으로 방법은 아래와 같다.

1. latent code z1,z2 를 mapping network를 사용하여 w1,w2를 구한다.
2. w1을 사용하여 synthesis network에 적용한다.
3. 랜덤으로 선정된 layer에 w2를 적용한다.

	Mixing regularization	Number of latents during testing 1	2	3	4
E	0%	4.42	8.22	12.88	17.41
	50%	4.41	6.10	8.71	11.61
F	90%	**4.40**	**5.11**	6.88	9.03
	100%	4.83	5.17	**6.63**	**8.40**

적용 결과로 latent code의 수에 따라 fid가 작아지는 regularization과 같은 효과를 낼 수 있다.

Stochastic variation

주근깨, 피부 모공, 머릿결 등의 경우 안경 유무, 인종 등의 특징들에 비해 생성된 이미지의 겉모습에는 큰 영향을 끼치지 않는다.

기존의 generator의 경우는 input으로만 이미지 생성이 진행되기 때문에, 반복되는 패턴을 생성하게 된다는 문제점이 있다.

convolution output에 pixel당 noise를 더함으로써 위와 같은 문제점을 해결할 수 있다고 한다.

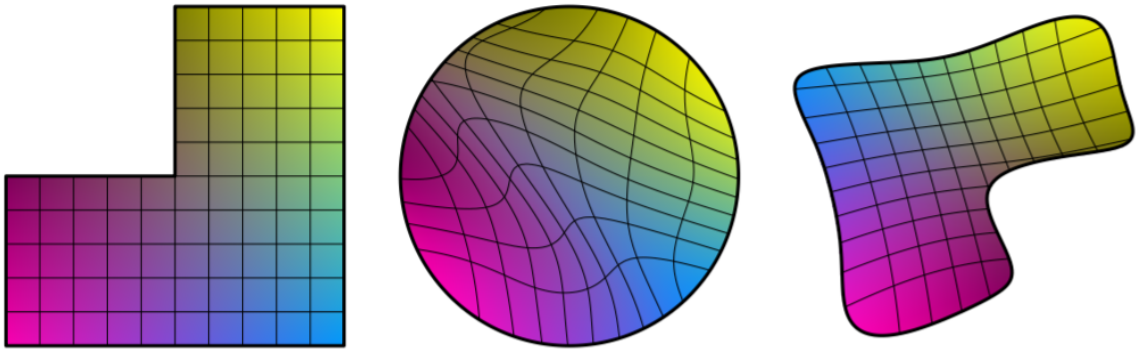
(a) Generated image (b) Stochastic variation (c) Standard deviation

위와 같은 방법으로 noise를 추가한 실험 결과로 a는 생성한 이미지, b는 noise를 추가한 이미지로 머릿결의 형태가 약간 변한걸 확인할 수 있고, c는 100개의 다른 noise로 이미지 생성한 이후 표준편차들을 시각화 한 결과이다. 눈,코,입에 비해 머릿결이 편차가 크게 나타남을 확인할 수 있다.

이를 통해 noise에 따라서 안경, 인종 들의 큰 특징이 아닌 머릿결 등의 작은 특징들이 변한 것을 확인 가능하다.

Disentanglement studies

Disentanglement는 latent space의 subspace가 linear함을 의미한다.



(a) Distribution of features in training set (b) Mapping from $\mathcal{Z}$ to features (c) Mapping from $\mathcal{W}$ to features

예를 들어 z의 특정 값만 바꾸었을 때, 생성되는 이미지의 하나의 특성만 영향을 끼치면, disentanglement라고 한다.

기존의 generator의 경우 z에 의해서 이미지를 생성하기 때문에 entanglement representation을 기반으로 이미지를 생성한다고 볼 수 있으며, 이보단 mapping network인 f(z)를 기반으로 나온 w(disentanglement representation)을 기반으로 학습한다면 현실적인 이미지를 생성하는 것이 더 쉬울 것이다.

하지만 disentanglement를 측정하는 metrics가 필요하기 때문에 2가지 방법을 제안한다.

Perceptual path length

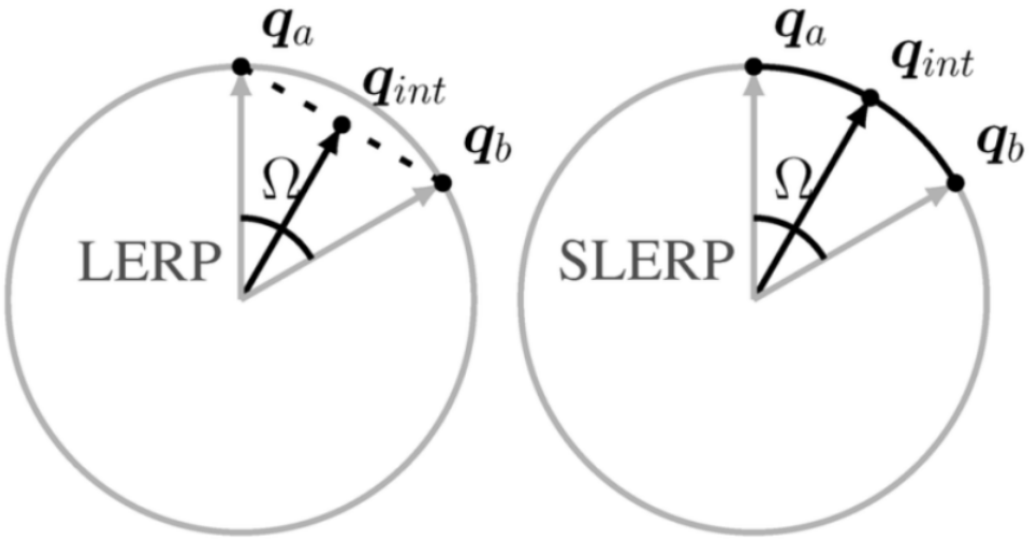
metrics의 수행 방법은 아래와 같다.

1. z1과 z2를 t, t+eps(1e-4)를 기준으로 선형보간(lerp)나, 구면 선형보간(slerp)를 사용하여 z1과 z2 사이에 있는 지점을 구한다.
2. 1에서 구한 2 지점을 기준으로 generator를 사용하여 이미지를 생성한다.
3. 생성된 이미지들을 vgg16 network를 사용하여 embedding한다
4. embedding된 vector 2개 사이의 거리(pairwise distance)를 구한다.

$$l_{\mathcal{Z}} = \mathbb{E}\left[\frac{1}{\epsilon^2} d\left(G\left(slerp\left(\mathbf{z_1}, \mathbf{z_2}; t\right)\right), G\left(slerp\left(\mathbf{z_1}, \mathbf{z_2}; t+\epsilon\right)\right),\right)\right]$$

$$l_{\mathcal{W}} = \mathbb{E}\left[\frac{1}{\epsilon^2} d\left(G\left(lerp\left(\mathbf{z_1}, \mathbf{z_2}; t\right)\right), G\left(lerp\left(\mathbf{z_1}, \mathbf{z_2}; t+\epsilon\right)\right),\right)\right]$$

lerp, slerp



z는 normalization이 되어있고, w는 그렇지 않아서 z는 slerp를 사용하고, w에는 lerp를 사용하였다고 한다.

Linear separability

만약 latent space가 효율적으로 disentangle하다면, 특성을 나누는 direction vector를 찾을 수 있을 것이다.

이를 수행한 방법으로는 여성,남성 등의 특징들을 이진분류하는 auxiliary classification network를 활용하여 생성된 이미지에 label(Y)을 부여한다. (논문에서는 40개의 특징들을 사용, 200000개 생성된 이미지중 신뢰도가 높은 100000개를 사용)

이러한 label을 분류할수 있는 linear hyperplane을 찾기 위해서 linear svm을 수행한다(X).

이를 conditional cross entropy를 계산한다.

$$exp(\sum_i H(Y_i|X_i))$$

즉 auxiliary classification network로 분류된 label을 linear svm에서 잘 예측 한다면, latent space가 disentangle하다고 볼 수 있다.

	Method	Path length full	Path length end	Separability
B	Traditional generator $\mathcal{Z}$	412.0	415.3	10.78
D	Style-based generator $\mathcal{W}$	446.2	376.6	3.61
E	+ Add noise inputs $\mathcal{W}$	**200.5**	**160.6**	3.54
	+ Mixing 50% $\mathcal{W}$	231.5	182.1	**3.51**
F	+ Mixing 90% $\mathcal{W}$	234.0	195.9	3.79

Table 3. Perceptual path lengths and separability scores for various generator architectures in FFHQ (lower is better). We perform the measurements in $\mathcal{Z}$ for the traditional network, and in $\mathcal{W}$ for style-based ones. Making the network resistant to style mixing appears to distort the intermediate latent space $\mathcal{W}$ somewhat. We hypothesize that mixing makes it more difficult for $\mathcal{W}$ to efficiently encode factors of variation that span multiple scales.

	Method	FID	Path length full	Path length end	Separability
B	Traditional 0 $\mathcal{Z}$	5.25	412.0	415.3	10.78
	Traditional 8 $\mathcal{Z}$	4.87	896.2	902.0	170.29
	Traditional 8 $\mathcal{W}$	4.87	324.5	212.2	6.52
	Style-based 0 $\mathcal{Z}$	5.06	283.5	285.5	9.88
	Style-based 1 $\mathcal{W}$	4.60	219.9	209.4	6.81
	Style-based 2 $\mathcal{W}$	4.43	**217.8**	199.9	6.25
F	Style-based 8 $\mathcal{W}$	**4.40**	234.0	**195.9**	**3.79**

Table 4. The effect of a mapping network in FFHQ. The number in method name indicates the depth of the mapping network. We see that FID, separability, and path length all benefit from having a mapping network, and this holds for both style-based and traditional generator architectures. Furthermore, a deeper mapping network generally performs better than a shallow one.